Microsoft

OFFICIAL MICROSOFT LEARNING PRODUCT

20778C

Analyzing Data with Power BI

Product Number: 20778C

Part Number (if applicable): X22-11274

Released: 06/2019

MICROSOFT LICENSE TERMS
MICROSOFT INSTRUCTOR-LED COURSEWARE

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. Please read them. They apply to your use of the content accompanying this agreement which includes the media on which you received it, if any. These license terms also apply to Trainer Content and any updates and supplements for the Licensed Content unless other terms accompany those items. If so, those terms apply.

BY ACCESSING, DOWNLOADING OR USING THE LICENSED CONTENT, YOU ACCEPT THESE TERMS. IF YOU DO NOT ACCEPT THEM, DO NOT ACCESS, DOWNLOAD OR USE THE LICENSED CONTENT.

If you comply with these license terms, you have the rights below for each license you acquire.

1. DEFINITIONS.

a. "Authorized Learning Center" means a Microsoft IT Academy Program Member, Microsoft Learning Competency Member, or such other entity as Microsoft may designate from time to time.

b. "Authorized Training Session" means the instructor-led training class using Microsoft Instructor-Led Courseware conducted by a Trainer at or through an Authorized Learning Center.

c. "Classroom Device" means one (1) dedicated, secure computer that an Authorized Learning Center owns or controls that is located at an Authorized Learning Center's training facilities that meets or exceeds the hardware level specified for the particular Microsoft Instructor-Led Courseware.

d. "End User" means an individual who is (i) duly enrolled in and attending an Authorized Training Session or Private Training Session, (ii) an employee of a MPN Member, or (iii) a Microsoft full-time employee.

e. "Licensed Content" means the content accompanying this agreement which may include the Microsoft Instructor-Led Courseware or Trainer Content.

f. "Microsoft Certified Trainer" or "MCT" means an individual who is (i) engaged to teach a training session to End Users on behalf of an Authorized Learning Center or MPN Member, and (ii) currently certified as a Microsoft Certified Trainer under the Microsoft Certification Program.

g. "Microsoft Instructor-Led Courseware" means the Microsoft-branded instructor-led training course that educates IT professionals and developers on Microsoft technologies. A Microsoft Instructor-Led Courseware title may be branded as MOC, Microsoft Dynamics or Microsoft Business Group courseware.

h. "Microsoft IT Academy Program Member" means an active member of the Microsoft IT Academy Program.

i. "Microsoft Learning Competency Member" means an active member of the Microsoft Partner Network program in good standing that currently holds the Learning Competency status.

j. "MOC" means the "Official Microsoft Learning Product" instructor-led courseware known as Microsoft Official Course that educates IT professionals and developers on Microsoft technologies.

k. "MPN Member" means an active Microsoft Partner Network program member in good standing.

l. "Personal Device" means one (1) personal computer, device, workstation or other digital electronic device that you personally own or control that meets or exceeds the hardware level specified for the particular Microsoft Instructor-Led Courseware.

m. "Private Training Session" means the instructor-led training classes provided by MPN Members for corporate customers to teach a predefined learning objective using Microsoft Instructor-Led Courseware. These classes are not advertised or promoted to the general public and class attendance is restricted to individuals employed by or contracted by the corporate customer.

n. "Trainer" means (i) an academically accredited educator engaged by a Microsoft IT Academy Program Member to teach an Authorized Training Session, and/or (ii) a MCT.

o. "Trainer Content" means the trainer version of the Microsoft Instructor-Led Courseware and additional supplemental content designated solely for Trainers' use to teach a training session using the Microsoft Instructor-Led Courseware. Trainer Content may include Microsoft PowerPoint presentations, trainer preparation guide, train the trainer materials, Microsoft One Note packs, classroom setup guide and Pre-release course feedback form. To clarify, Trainer Content does not include any software, virtual hard disks or virtual machines.

2. USE RIGHTS. The Licensed Content is licensed not sold. The Licensed Content is licensed on a ***one copy per user basis***, such that you must acquire a license for each individual that accesses or uses the Licensed Content.

2.1 Below are five separate sets of use rights. Only one set of rights apply to you.

a. **If you are a Microsoft IT Academy Program Member:**

i. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.

ii. For each license you acquire on behalf of an End User or Trainer, you may either:

1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User who is enrolled in the Authorized Training Session, and only immediately prior to the commencement of the Authorized Training Session that is the subject matter of the Microsoft Instructor-Led Courseware being provided, **or**
2. provide one (1) End User with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
3. provide one (1) Trainer with the unique redemption code and instructions on how they can access one (1) Trainer Content,

provided you comply with the following:

iii. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,

iv. you will ensure each End User attending an Authorized Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Authorized Training Session,

v. you will ensure that each End User provided with the hard-copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,

vi. you will ensure that each Trainer teaching an Authorized Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Authorized Training Session,

vii. you will only use qualified Trainers who have in-depth knowledge of and experience with the Microsoft technology that is the subject of the Microsoft Instructor-Led Courseware being taught for all your Authorized Training Sessions,
viii. you will only deliver a maximum of 15 hours of training per week for each Authorized Training Session that uses a MOC title, and
ix. you acknowledge that Trainers that are not MCTs will not have access to all of the trainer resources for the Microsoft Instructor-Led Courseware.

b. **If you are a Microsoft Learning Competency Member**:
   i. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.
   ii. For each license you acquire on behalf of an End User or Trainer, you may either:
      1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User attending the Authorized Training Session and only immediately prior to the commencement of the Authorized Training Session that is the subject matter of the Microsoft Instructor-Led Courseware provided, **or**
      2. provide one (1) End User attending the Authorized Training Session with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
      3. you will provide one (1) Trainer with the unique redemption code and instructions on how they can access one (1) Trainer Content,

   provided you comply with the following:
   iii. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,
   iv. you will ensure that each End User attending an Authorized Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Authorized Training Session,
   v. you will ensure that each End User provided with a hard-copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,
   vi. you will ensure that each Trainer teaching an Authorized Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Authorized Training Session,
   vii. you will only use qualified Trainers who hold the applicable Microsoft Certification credential that is the subject of the Microsoft Instructor-Led Courseware being taught for your Authorized Training Sessions,
   viii. you will only use qualified MCTs who also hold the applicable Microsoft Certification credential that is the subject of the MOC title being taught for all your Authorized Training Sessions using MOC,
   ix. you will only provide access to the Microsoft Instructor-Led Courseware to End Users, and
   x. you will only provide access to the Trainer Content to Trainers.

c. **If you are a MPN Member**:

i. Each license acquired on behalf of yourself may only be used to review one (1) copy of the Microsoft Instructor-Led Courseware in the form provided to you. If the Microsoft Instructor-Led Courseware is in digital format, you may install one (1) copy on up to three (3) Personal Devices. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.

ii. For each license you acquire on behalf of an End User or Trainer, you may either:

1. distribute one (1) hard copy version of the Microsoft Instructor-Led Courseware to one (1) End User attending the Private Training Session, and only immediately prior to the commencement of the Private Training Session that is the subject matter of the Microsoft Instructor-Led Courseware being provided, **or**
2. provide one (1) End User who is attending the Private Training Session with the unique redemption code and instructions on how they can access one (1) digital version of the Microsoft Instructor-Led Courseware, **or**
3. you will provide one (1) Trainer who is teaching the Private Training Session with the unique redemption code and instructions on how they can access one (1) Trainer Content,

provided you comply with the following:

iii. you will only provide access to the Licensed Content to those individuals who have acquired a valid license to the Licensed Content,

iv. you will ensure that each End User attending an Private Training Session has their own valid licensed copy of the Microsoft Instructor-Led Courseware that is the subject of the Private Training Session,

v. you will ensure that each End User provided with a hard copy version of the Microsoft Instructor-Led Courseware will be presented with a copy of this agreement and each End User will agree that their use of the Microsoft Instructor-Led Courseware will be subject to the terms in this agreement prior to providing them with the Microsoft Instructor-Led Courseware. Each individual will be required to denote their acceptance of this agreement in a manner that is enforceable under local law prior to their accessing the Microsoft Instructor-Led Courseware,

vi. you will ensure that each Trainer teaching an Private Training Session has their own valid licensed copy of the Trainer Content that is the subject of the Private Training Session,

vii. you will only use qualified Trainers who hold the applicable Microsoft Certification credential that is the subject of the Microsoft Instructor-Led Courseware being taught for all your Private Training Sessions,

viii. you will only use qualified MCTs who hold the applicable Microsoft Certification credential that is the subject of the MOC title being taught for all your Private Training Sessions using MOC,

ix. you will only provide access to the Microsoft Instructor-Led Courseware to End Users, and

x. you will only provide access to the Trainer Content to Trainers.

d. **If you are an End User:**

For each license you acquire, you may use the Microsoft Instructor-Led Courseware solely for your personal training use. If the Microsoft Instructor-Led Courseware is in digital format, you may access the Microsoft Instructor-Led Courseware online using the unique redemption code provided to you by the training provider and install and use one (1) copy of the Microsoft Instructor-Led Courseware on up to three (3) Personal Devices. You may also print one (1) copy of the Microsoft Instructor-Led Courseware. You may not install the Microsoft Instructor-Led Courseware on a device you do not own or control.

e. **If you are a Trainer.**

i. For each license you acquire, you may install and use one (1) copy of the Trainer Content in the form provided to you on one (1) Personal Device solely to prepare and deliver an Authorized Training Session or Private Training Session, and install one (1) additional copy on another Personal Device as a backup copy, which may be used only to reinstall the Trainer Content. You may not install or use a copy of the Trainer Content on a device you do not own or control. You may also print one (1) copy of the Trainer Content solely to prepare for and deliver an Authorized Training Session or Private Training Session.

ii. You may customize the written portions of the Trainer Content that are logically associated with instruction of a training session in accordance with the most recent version of the MCT agreement. If you elect to exercise the foregoing rights, you agree to comply with the following: (i) customizations may only be used for teaching Authorized Training Sessions and Private Training Sessions, and (ii) all customizations will comply with this agreement. For clarity, any use of "*customize"* refers only to changing the order of slides and content, and/or not using all the slides or content, it does not mean changing or modifying any slide or content.

2.2 **Separation of Components.** The Licensed Content is licensed as a single unit and you may not separate their components and install them on different devices.

2.3 **Redistribution of Licensed Content**. Except as expressly provided in the use rights above, you may not distribute any Licensed Content or any portion thereof (including any permitted modifications) to any third parties without the express written permission of Microsoft.

2.4 **Third Party Notices**. The Licensed Content may include third party code tent that Microsoft, not the third party, licenses to you under this agreement. Notices, if any, for the third party code ntent are included for your information only.

2.5 **Additional Terms**. Some Licensed Content may contain components with additional terms, conditions, and licenses regarding its use. Any non-conflicting terms in those conditions and licenses also apply to your use of that respective component and supplements the terms described in this agreement.

3. LICENSED CONTENT BASED ON PRE-RELEASE TECHNOLOGY. If the Licensed Content's subject matter is based on a pre-release version of Microsoft technology ("**Pre-release**"), then in addition to the other provisions in this agreement, these terms also apply:

a. **Pre-Release Licensed Content.** This Licensed Content subject matter is on the Pre-release version of the Microsoft technology. The technology may not work the way a final version of the technology will and we may change the technology for the final version. We also may not release a final version. Licensed Content based on the final version of the technology may not contain the same information as the Licensed Content based on the Pre-release version. Microsoft is under no obligation to provide you with any further content, including any Licensed Content based on the final version of the technology.

b. **Feedback.** If you agree to give feedback about the Licensed Content to Microsoft, either directly or through its third party designee, you give to Microsoft without charge, the right to use, share and commercialize your feedback in any way and for any purpose. You also give to third parties, without charge, any patent rights needed for their products, technologies and services to use or interface with any specific parts of a Microsoft technology, Microsoft product, or service that includes the feedback. You will not give feedback that is subject to a license that requires Microsoft to license its technology, technologies, or products to third parties because we include your feedback in them. These rights survive this agreement.

c. **Pre-release Term**. If you are an Microsoft IT Academy Program Member, Microsoft Learning Competency Member, MPN Member or Trainer, you will cease using all copies of the Licensed Content on the Pre-release technology upon (i) the date which Microsoft informs you is the end date for using the Licensed Content on the Pre-release technology, or (ii) sixty (60) days after the commercial release of the technology that is the subject of the Licensed Content, whichever is earliest ("**Pre-release term**"). Upon expiration or termination of the Pre-release term, you will irretrievably delete and destroy all copies of the Licensed Content in your possession or under your control.

4. **SCOPE OF LICENSE**. The Licensed Content is licensed, not sold. This agreement only gives you some rights to use the Licensed Content. Microsoft reserves all other rights. Unless applicable law gives you more rights despite this limitation, you may use the Licensed Content only as expressly permitted in this agreement. In doing so, you must comply with any technical limitations in the Licensed Content that only allows you to use it in certain ways. Except as expressly permitted in this agreement, you may not:
   - access or allow any individual to access the Licensed Content if they have not acquired a valid license for the Licensed Content,
   - alter, remove or obscure any copyright or other protective notices (including watermarks), branding or identifications contained in the Licensed Content,
   - modify or create a derivative work of any Licensed Content,
   - publicly display, or make the Licensed Content available for others to access or use,
   - copy, print, install, sell, publish, transmit, lend, adapt, reuse, link to or post, make available or distribute the Licensed Content to any third party,
   - work around any technical limitations in the Licensed Content, or
   - reverse engineer, decompile, remove or otherwise thwart any protections or disassemble the Licensed Content except and only to the extent that applicable law expressly permits, despite this limitation.

5. **RESERVATION OF RIGHTS AND OWNERSHIP**. Microsoft reserves all rights not expressly granted to you in this agreement. The Licensed Content is protected by copyright and other intellectual property laws and treaties. Microsoft or its suppliers own the title, copyright, and other intellectual property rights in the Licensed Content.

6. **EXPORT RESTRICTIONS**. The Licensed Content is subject to United States export laws and regulations. You must comply with all domestic and international export laws and regulations that apply to the Licensed Content. These laws include restrictions on destinations, end users and end use. For additional information, see www.microsoft.com/exporting.

7. **SUPPORT SERVICES**. Because the Licensed Content is "as is", we may not provide support services for it.

8. **TERMINATION.** Without prejudice to any other rights, Microsoft may terminate this agreement if you fail to comply with the terms and conditions of this agreement. Upon termination of this agreement for any reason, you will immediately stop all use of and delete and destroy all copies of the Licensed Content in your possession or under your control.

9. **LINKS TO THIRD PARTY SITES**. You may link to third party sites through the use of the Licensed Content. The third party sites are not under the control of Microsoft, and Microsoft is not responsible for the contents of any third party sites, any links contained in third party sites, or any changes or updates to third party sites. Microsoft is not responsible for webcasting or any other form of transmission received from any third party sites. Microsoft is providing these links to third party sites to you only as a convenience, and the inclusion of any link does not imply an endorsement by Microsoft of the third party site.

10. **ENTIRE AGREEMENT.** This agreement, and any additional terms for the Trainer Content, updates and supplements are the entire agreement for the Licensed Content, updates and supplements.

11. **APPLICABLE LAW.**
    a. United States. If you acquired the Licensed Content in the United States, Washington state law governs the interpretation of this agreement and applies to claims for breach of it, regardless of conflict of laws principles. The laws of the state where you live govern all other claims, including claims under state consumer protection laws, unfair competition laws, and in tort.

b. Outside the United States. If you acquired the Licensed Content in any other country, the laws of that country apply.

12. LEGAL EFFECT. This agreement describes certain legal rights. You may have other rights under the laws of your country. You may also have rights with respect to the party from whom you acquired the Licensed Content. This agreement does not change your rights under the laws of your country if the laws of your country do not permit it to do so.

13. DISCLAIMER OF WARRANTY. THE LICENSED CONTENT IS LICENSED "AS-IS" AND "AS AVAILABLE." YOU BEAR THE RISK OF USING IT. MICROSOFT AND ITS RESPECTIVE AFFILIATES GIVES NO EXPRESS WARRANTIES, GUARANTEES, OR CONDITIONS. YOU MAY HAVE ADDITIONAL CONSUMER RIGHTS UNDER YOUR LOCAL LAWS WHICH THIS AGREEMENT CANNOT CHANGE. TO THE EXTENT PERMITTED UNDER YOUR LOCAL LAWS, MICROSOFT AND ITS RESPECTIVE AFFILIATES EXCLUDES ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NON-INFRINGEMENT.

14. LIMITATION ON AND EXCLUSION OF REMEDIES AND DAMAGES. YOU CAN RECOVER FROM MICROSOFT, ITS RESPECTIVE AFFILIATES AND ITS SUPPLIERS ONLY DIRECT DAMAGES UP TO US$5.00. YOU CANNOT RECOVER ANY OTHER DAMAGES, INCLUDING CONSEQUENTIAL, LOST PROFITS, SPECIAL, INDIRECT OR INCIDENTAL DAMAGES.

This limitation applies to

- anything related to the Licensed Content, services, content (including code) on third party Internet sites or third-party programs; and
- claims for breach of contract, breach of warranty, guarantee or condition, strict liability, negligence, or other tort to the extent permitted by applicable law.

It also applies even if Microsoft knew or should have known about the possibility of the damages. The above limitation or exclusion may not apply to you because your country may not allow the exclusion or limitation of incidental, consequential or other damages.

Please note: As this Licensed Content is distributed in Quebec, Canada, some of the clauses in this agreement are provided below in French.

Remarque : Ce le contenu sous licence étant distribué au Québec, Canada, certaines des clauses dans ce contrat sont fournies ci-dessous en français.

EXONÉRATION DE GARANTIE. Le contenu sous licence visé par une licence est offert « tel quel ». Toute utilisation de ce contenu sous licence est à votre seule risque et péril. Microsoft n'accorde aucune autre garantie expresse. Vous pouvez bénéficier de droits additionnels en vertu du droit local sur la protection dues consommateurs, que ce contrat ne peut modifier. La ou elles sont permises par le droit locale, les garanties implicites de qualité marchande, d'adéquation à un usage particulier et d'absence de contrefaçon sont exclues.

LIMITATION DES DOMMAGES-INTÉRÊTS ET EXCLUSION DE RESPONSABILITÉ POUR LES DOMMAGES. Vous pouvez obtenir de Microsoft et de ses fournisseurs une indemnisation en cas de dommages directs uniquement à hauteur de 5,00 $ US. Vous ne pouvez prétendre à aucune indemnisation pour les autres dommages, y compris les dommages spéciaux, indirects ou accessoires et pertes de bénéfices.
Cette limitation concerne:

- tout ce qui est relié au le contenu sous licence, aux services ou au contenu (y compris le code) figurant sur des sites Internet tiers ou dans des programmes tiers; et.
- les réclamations au titre de violation de contrat ou de garantie, ou au titre de responsabilité stricte, de négligence ou d'une autre faute dans la limite autorisée par la loi en vigueur.

Elle s'applique également, même si Microsoft connaissait ou devrait connaître l'éventualité d'un tel dommage. Si votre pays n'autorise pas l'exclusion ou la limitation de responsabilité pour les dommages indirects, accessoires ou de quelque nature que ce soit, il se peut que la limitation ou l'exclusion ci-dessus ne s'appliquera pas à votre égard.

EFFET JURIDIQUE. Le présent contrat décrit certains droits juridiques. Vous pourriez avoir d'autres droits prévus par les lois de votre pays. Le présent contrat ne modifie pas les droits que vous confèrent les lois de votre pays si celles-ci ne le permettent pas.

Revised July 2013

Welcome!

Thank you for taking our training! We've worked together with our Microsoft Certified Partners for Learning Solutions and our Microsoft IT Academies to bring you a world-class learning experience—whether you're a professional looking to advance your skills or a student preparing for a career in IT.

- **Microsoft Certified Trainers and Instructors**—Your instructor is a technical and instructional expert who meets ongoing certification requirements. And, if instructors are delivering training at one of our Certified Partners for Learning Solutions, they are also evaluated throughout the year by students and by Microsoft.

- **Certification Exam Benefits**—After training, consider taking a Microsoft Certification exam. Microsoft Certifications validate your skills on Microsoft technologies and can help differentiate you when finding a job or boosting your career. In fact, independent research by IDC concluded that 75% of managers believe certifications are important to team performance[1]. Ask your instructor about Microsoft Certification exam promotions and discounts that may be available to you.

- **Customer Satisfaction Guarantee**—Our Certified Partners for Learning Solutions offer a satisfaction guarantee and we hold them accountable for it. At the end of class, please complete an evaluation of today's experience. We value your feedback!

We wish you a great learning experience and ongoing success in your career!

Sincerely,

Microsoft Learning
www.microsoft.com/learning

Microsoft | Learning

[1] *IDC,* Value of Certification: Team Certification and Organizational Performance, *November 2006*

Acknowledgements

Microsoft Learning would like to acknowledge and thank the following for their contribution towards developing this title. Their effort at various stages in the development has ensured that you have a good classroom experience.

Aaron Johal – Content Developer

Aaron Johal is a Microsoft Certified Trainer who splits his time between training, consultancy, content development, contracting and learning. Since he moved into the non-functional side of the Information Technology business. He has presented technical sessions at SQL Pass in Denver and at sqlbits in London. He has also taught and worked in a consulting capacity throughout the UK and abroad, including Africa, Spain, Saudi Arabia, Netherlands, France, and Ireland. He enjoys interfacing functional and non-functional roles to try and close the gaps between effective use of Information Technology and the needs of the Business.

Rachel Horder – Content Developer

Rachel Horder graduated with a degree in Journalism and began her career in London writing for The Times technology supplement. After discovering a love for programming, Rachel became a full-time developer, and now provides SQL Server consultancy services to businesses across a wide variety of industries. Rachel is MCSA certified, and continues to write technical articles and books, including What's New in SQL Server 2012. As an active member of the SQL Server community, Rachel organizes the Bristol SQL Server Club user group, runs the Bristol leg of SQL Relay, and is a volunteer at SQLBits.

David Coombes – Content Developer

David Coombes is a Microsoft subject matter expert and professional content developer at Content Master—a division of CM Group Ltd. As a content developer David has been working with LeX on a number of courses in the Azure and Office 365 workspace, and more recently has been working with the Azure Gov. team on a series of technical whitepapers.

Geoff Allix – Technical Reviewer

Geoff Allix is a Microsoft SQL Server subject matter expert and professional content developer at Content Master—a division of CM Group Ltd. As a Microsoft Certified Trainer, Geoff has delivered training courses on SQL Server since version 6.5. Geoff is a Microsoft Certified IT Professional for SQL Server and has extensive experience in designing and implementing database and BI solutions on SQL Server technologies, and has provided consultancy services to organizations seeking to implement and optimize database solutions.

Lin Joyner – Technical Reviewer

Lin is an experienced Microsoft SQL Server developer and administrator. She has worked with SQL Server since version 6.0 and previously as a Microsoft Certified Trainer, delivered training courses across the UK. Lin has a wide breadth of knowledge across SQL Server technologies, including BI and Reporting Services. Lin also designs and authors SQL Server and .NET development training materials. She has been writing instructional content for Microsoft for over 15 years.

Contents

About This Course

This section provides a brief description of the course, audience, suggested prerequisites, and course objectives.

Course Description

This three-day instructor-led course provides students with the knowledge and skills to analyze data with Power BI.

Audience

The primary audience for this course is BI professionals who need to analyze data utilizing Power BI.

The secondary audiences for this course are technically proficient business users.

Student Prerequisites

In addition to their professional experience, students who attend this training should already have the following technical knowledge:

- Basic knowledge of the Microsoft Windows operating system and its core functionality.
- Basic knowledge of data warehouse schema topology (including star and snowflake schemas).
- Some exposure to basic programming concepts (such as looping and branching).
- An awareness of key business priorities such as revenue, profitability, and financial accounting is desirable.
- Familiarity with Microsoft Office applications – particularly Excel.

Course Objectives

After completing this course, students will be able to:

- Describe self-service BI.
- Describe the Power BI suite of products.
- Connect to data sources and optimize data models.
- Shape and combine data from different sources.
- Model data.
- Create reports and manage solutions.
- Connect to data stores using Power BI.
- Describe the Power BI developer API.
- Describe the Power BI mobile app.

Course Outline

The course outline is as follows:

- Module 1: 'Introduction to Self-Service BI Solutions' describes what self-service BI is and introduces software products that you can use to implement a BI solution.
- Module 2: 'Introducing Power BI' introduces the Power BI suite of products.
- Module 3: 'Power BI Data' describes the data sources available to Power BI and how to optimize your data model.

- Module 4: 'Shaping and Combining Data' introduces the tools that are available for preparing and transforming data.
- Module 5: 'Modeling Data' describes how to shape and enhance data.
- Module 6: 'Interactive Data Visualizations' describes how to create reports and manage solutions.
- Module 7: 'Direct Connectivity' describes how to connect directly to data stores, particularly cloud data and Analysis Services.
- Module 8: 'Development with Power BI' describes the Power BI developer API and custom visuals.
- Module 9: 'Power BI Mobile' introduces the Power BI mobile app and how to use it.

Course Materials

The following materials are included with your kit:

- ***Course Handbook:*** a succinct classroom learning guide that provides the critical technical information in a crisp, tightly-focused format, which is essential for an effective in-class learning experience.
  - **Lessons**: guide you through the learning objectives and provide the key points that are critical to the success of the in-class learning experience.
  - **Labs**: provide a real-world, hands-on platform for you to apply the knowledge and skills learned in the module.
  - **Module Reviews and Takeaways**: provide on-the-job reference material to boost knowledge and skills retention.
  - **Lab Answer Keys**: provide step-by-step lab solution guidance.

Additional Reading: Course Companion Content on the http://www.microsoft.com/learning/en-us/companion-moc.aspx_ Site*:* searchable, easy-to-browse digital content with integrated premium online resources that supplement the Course Handbook.

- **Modules**: include companion content, such as questions and answers, detailed demo steps and additional reading links, for each lesson. Additionally, they include Lab Review questions and answers and Module Reviews and Takeaways sections, which contain the review questions and answers, best practices, common issues and troubleshooting tips with answers, and real-world issues and scenarios with answers.
- **Resources**: include well-categorized additional resources that give you immediate access to the most current premium content on TechNet, MSDN®, or Microsoft® Press®.

Additional Reading: Student Course files on the http://www.microsoft.com/learning/en-us/companion-moc.aspx_ Site: includes the Allfiles.exe, a self-extracting executable file that contains all required files for the labs and demonstrations.

- ***Course evaluation:*** at the end of the course, you will have the opportunity to complete an online evaluation to provide feedback on the course, training facility, and instructor.

- To provide additional comments or feedback on the course, send email to mcspprt@microsoft.com. To inquire about the Microsoft Certification Program, send an email to mcphelp@microsoft.com.

Virtual Machine Environment

This section provides the information for setting up the classroom environment to support the business scenario of the course.

Virtual Machine Configuration

In this course, you will use Microsoft® Hyper-V™ to perform the labs.

Note: At the end of each lab, you must revert the virtual machines to a snapshot. You can find the instructions for this procedure at the end of each lab

The following table shows the role of each virtual machine that is used in this course:

Virtual machine	Role
20778C-MIA-DC	MIA-DC1 is a domain controller.
20778C-MIA-SQL	MIA-SQL has SQL Server 2016 installed
MSL-TMG1	TMG1 is used to access the internet

Software Configuration

The following software is installed on the virtual machines:

- Windows Server 2012 R2
- SQL2016
- Microsoft Office 2016
- SharePoint 2013SP1

Course Files

The files associated with the labs in this course are located in the D:\Labfiles folder on the 20778C-MIA-SQL virtual machine.

Classroom Setup

Each classroom computer will have the same virtual machine configured in the same way.

Course Hardware Level

To ensure a satisfactory student experience, Microsoft Learning requires a minimum equipment configuration for trainer and student computers in all Microsoft Learning Partner classrooms in which Official Microsoft Learning Product courseware is taught.

- Intel Virtualization Technology (Intel VT) or AMD Virtualization (AMD-V) processor
- Dual 120-gigabyte (GB) hard disks 7200 RM Serial ATA (SATA) or better
- 16 GB of random access memory (RAM)
- DVD drive

- Network adapter
- Super VGA (SVGA) 17-inch monitor
- Microsoft mouse or compatible pointing device
- Sound card with amplified speakers

Additionally, the instructor's computer must be connected to a projection display device that supports SVGA 1024×768 pixels, 16-bit colors.

Module 1

Introduction to Self-Service BI Solutions

Contents:

Module Overview

Business intelligence (BI) is a term that has become increasingly common in recent years. Along with big data, data mining, predictive analytics, data science, and data stewards, BI is now very much part of business vocabulary. Much of the impetus behind this is the need for organizations to cope with ever-increasing datasets. It's normal to have databases that contain millions of rows, requiring gigabytes, terabytes, or even petabytes, of storage space. Data is no longer confined to an on-premises server room—it's hosted in the cloud, feeds are taken from third-party providers, public datasets are freely available, and social media interactions generate ever-expanding datasets.

Reporting and analysis is certainly not a new concept to business, but the difference between how data analysis is done today, compared with five or 10 years ago, is immense. Nowadays, organizations need BI to see not only what was done in the past, but also more of what's to come. There's an overwhelming amount of data to gather and compose into reports. There's also an increasing need for data to offer up-to-the-minute numbers, so business can react faster to changing trends in markets and industries. Businesses that react fast and predict near-term trends to provide products and services where there is consumer demand have the best chance of survival in a modern and highly competitive world. With the rise of big data, there's an increasing need for data analysts who can take this data, and find the critical points within a plethora of information.

Objectives

After completing this module, you will be able to:

- Describe the trends in BI.
- Describe the process of data analysis in Power BI.
- Use the key visualizations in Power BI.
- Describe the rationale for self-service BI.

- Describe considerations for self-service BI.
- Understand how you can use Microsoft® products to implement a BI solution.

Lesson 1
Introduction to business intelligence

This lesson introduces you to the concepts that comprise BI. You will explore scenarios for using BI and how current trends affect the use of BI, project roles and data models.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand BI scenarios.
- See how trends in data and reporting solutions have affected BI.
- Describe the project roles within BI.
- Explain how enterprise BI data models work.

Business intelligence scenarios

Big data is big news—since the rise of the internet, social media, and the rapid growth of e-commerce, more and more data is being generated, gathered, and analyzed. Supermarkets and retail outlets offer store cards, loyalty cards, or reward cards—depending on how they want to label it—because they need to track spending habits and use this data to sell you more. They gather data, analyze what you like to buy, and then offer incentives to entice you to buy more of the same, or similar. Meanwhile, your online habits are monitored by cookies and advertisements show up on websites, tempting you to buy something you might have searched for earlier.

- Big data is the result of data generated by the internet, social media, and e-commerce:
  - Data is constantly being gathered for commercial use
  - Data is constantly growing in size
- Reporting:
  - Extracting data and presenting it to enable decision-making
  - Show metrics for organizational performance
- Analysis:
  - Evaluating data to discover insights
  - Data should answer questions, but quickly becomes outdated
- Collaboration:
  - Business analysts need to share data for decision-making

Reporting

Extracting data from your company's database and presenting it in reports is certainly not a new phenomenon. Most organizations, whatever their size, use some form of reporting, as a reflection of performance within their sector. Until recently, most organizations were happy with end-of-month and annual reports, as a backward reflection of their performance. Modern reporting still needs this, but it should also look to the future to predict where and how to sell more, thereby increasing turnover and reducing the bottom line.

Traditionally, reports have been compiled by department heads, and then given to directors to guide their decision-making. Organizational data, or business intelligence, was the privilege of a few. For example, reports show metrics—how much did we sell last month? How many new customers have we acquired this year? How many mentions did our latest promotion receive on social media? A report provides the answers to questions that the organization needs to make decisions. Reports can be contained in spreadsheets, or created using a visual tool, and distributed on a daily, weekly, or other regular schedule. Reflecting on past performance is a worthwhile task, but modern reports must also be forward looking.

Analysis

Analysis is the process of evaluating data to find insights. Data analysis should answer questions, and offer guidance in decision-making. Data is extracted from source, then cleaned, modeled, and transformed until it's presented appropriately in a report. The report can be a simple table in a spreadsheet, or a visual and dynamic, colorful solution. How the data is presented affects the analysis and the conclusions drawn. For example, you can present data in a column chart, but not notice patterns in that data until you use a different type of chart—such as a map or scatter chart—and discover clusters of behavior as a result of geographic location, or outliers that are skewing results.

With so much data to analyze and constant changes in consumer and market trends, modern data has a limited lifespan. Data quickly becomes outdated, so the process of analysis is ongoing. However, with bigger data to analyze, more questions are asked. With an increase in publicly available datasets, including population changes, socioeconomic data, weather patterns, and climate change, you analyze corporate data against a backdrop of relevant statistics.

Collaboration

Data is generated and consumed ubiquitously—it's no longer retained and controlled by a handful of decision makers in an organization. Instead, data is used at all levels, meaning colleagues can react to it, and change the course of their work. Information is critical to companies of all sizes and across industries, with information workers needing to collaborate and share data and results. Microsoft Excel® has long been the dominant tool of the business user—spreadsheets are created, shared, published, altered, emailed, printed, saved, and distributed without version control or adherence to security policies. As spreadsheets are shared and changed, and shared again, analysts work from different datasets, see different results, and reach different conclusions. To collaborate and work cohesively, analysts must be able to synchronize their teamwork.

Trends in business intelligence

The possibilities for analysis grow in line with the increasing number of data sources, and expanding volumes of data. With Microsoft SQL Server® offering in-memory analytics, data doesn't have to be moved outside of the database, and organizations can perform real-time operational analytics. The BI trend is moving away from only analyzing past data, to analyzing real-time data, and using historical data to predict the future.

- BI trend is moving away from analyzing historical data, towards real-time analytics and predictions:
  - Self-service reporting and analysis:
    - Self-service has existed since the invention of spreadsheets
    - Widespread adoption of Excel and the use of power tools
    - Enables independence from IT, quick to produce reports
  - Increasing adoption of BI:
    - Organizations of all sizes gathering data and statistics
    - Essential to react to trends and remain competitive
  - Availability of out-of-the-box solutions:
    - Solutions from Tableau, Qlik, Microsoft, Salesforce, and so on
    - Some have large license fees and may require trained report developer

Self-service reporting and analysis

It could be argued that self-service BI has been around since spreadsheets first entered the software market, enabling users to crunch numbers at their desks. The almost universal adoption of Excel has enabled this trend to continue. With the integration of the four power tools—Power Pivot, Power Query, Power View, and Power Map—Excel users acquire data from a myriad of sources then model, transform, and present that data in sophisticated visualizations. The attraction of Excel and its power tools is the independence it offers to business users. When users access the data they need, they immediately begin shaping and formatting that data, and designing reports to their own specification.

Using a more sophisticated reporting solution generally requires a dedicated report developer, and a lengthy process to submit a feature requirement to IT then wait for the report to be developed and published—only to find it doesn't deliver the correct data. So begins another lengthy process of

submitting a change request, and waiting for the report developer to make the changes. Giving users access to the data means they see what is available for analysis and decide what is useful. The delay in waiting for a report not only frustrates users and holds back their work, but also delays decision-making and the ability for organizations to react to changing circumstances.

Increasing adoption of BI by a wider range of organizations

BI is no longer the reserve of big organizations with large budgets to throw at data warehousing projects. Any business operating on the web gathers information about their customers' spending habits, the products they viewed, and their buying decisions. It seems that our online presence, enhanced through the proliferation of mobile devices, is continuously monitored, with all our moves and preferences stored for analysis. To be more efficient, and therefore more competitive, organizations of all sizes must gather data. However, gathering this data is no use unless it's converted to actionable information. Along with increasing volumes of data, the availability of cheaper, easier to use solutions has helped drive the market, meaning organizations with even the smallest of budgets devote some level of resource to BI.

Availability of out-of-the-box solutions

Organizations can license sophisticated BI solutions from the major vendors in the market, including Tableau, Qlik, Salesforce, Microsoft, Oracle, IBM, SAP, SAS, and more. You can use these solutions to create highly visual reports. With the ability to connect to a variety of data sources, you can create reports and dashboards. However, depending on the vendor, some of the major solutions require expensive server and client licenses, in addition to trained users to create the reports.

Business intelligence project roles

Developing a BI solution requires much upfront planning and designing to ensure the project stays on target and comes to fruition without major issues. The BI project team comprises a number of roles. If it's a new project, the program manager might hire and instruct a data architect and a technical architect—after much of their planning is complete, BI developers will be hired. This depends on the organization, how many projects are in the pipeline, and if contract staff are to provide extra resource.

- Developing BI solutions requires upfront planning
- Each role in the project performs a vital function:
  - Program manager
  - Data architect
  - Technical architect
  - BI developer

Program manager

The program manager is responsible for the organizational BI strategy and delivery, often coordinating multiple projects at a time. The program manager is the overall leader of the BI department and, while the role is nontechnical, it does require an understanding of the subject matter, the business requirements, and a comprehension of technical terminology. The main role of a program manager includes:

- Acquiring funding for projects.
- Creating budgets.
- Engaging with stakeholders to determine requirements.
- Analyzing the impact of the project going into production.
- Communicating vision to end users and stakeholders.
- Being responsible for building teams and hiring new employees.

- Undertaking risk assessment.
- Setting standards and ensuring these are met.
- Establishing project priorities, and creating deadlines.
- Managing the expectations of both users and stakeholders.
- Providing status updates.
- Measuring performance.

Data architect

Like the program manager, the data architect is responsible for multiple projects, combining business and technical knowledge to shape the BI solutions. The data must be architected and presented in a design that the organization understands. The main role of the data architect includes:

- Developing the data architecture of the organization.
- Analyzing data requirements and planning for future change requirements.
- Performing logical data modeling.
- Implementing databases.
- Resolving issues between different systems and different data sources.
- Managing master data and liaising with the data steward.

Technical architect

The technical architect must communicate with the BI developers and the operations team to ensure the BI environment is configured correctly. This role is less hands-on than the BI developer, but requires deep technological understanding. The main role of the technical architect includes:

- Assessing the existing BI environment.
- Evaluating development technologies.
- Deciding on appropriate development technologies and justifying the decisions to the program manager.
- Designing the architecture of the extract, transform, and load (ETL) processes.
- Developing the disaster recovery (DR) plan.
- Interfacing with operations and DBA teams.

BI developer

The BI developer role can comprise ETL, data warehouse (DW), and report development. Depending on the size of the organization and the structure of the team, a developer might specialize in one aspect, or may perform one or more roles, but there is likely to be an overlap between at least two. The main role of a BI developer includes:

- Designing ETL packages to load data into a staging area.
- Building ETL packages that perform data transformations in the staging area.
- Writing ETL packages to load the transformed data into the data warehouse.
- Creating and managing ETL job schedules.
- Monitoring the ETL process for performance issues and failures.
- Debugging issues in the ETL process.

- Developing the data warehouse database.
- Resolving data issues.
- Building cubes.
- Designing and developing reports.
- Writing code to extract data from the data warehouse.
- Creating a schedule for publishing and distributing reports.

Enterprise BI data models

- Create a consistent view of data elements and their relationships in the organization
- Set standards and use naming conventions
- Comprise a logical and physical model
- Semantic model gives meaning to the data

Enterprise data modeling is the creation of a consistent view of data elements and their relationships in the organization. When more than one data modeler is working on the model, it's important that standards and naming conventions are created and adhered to. Data might be imported from different systems, so naming conventions are likely to vary across sources. This inconsistency should be addressed during the modeling process. If the model comprises a data warehouse, naming conventions should be used for fact, lookup, and history tables. Conventions can also be applied to columns to denote keys, codes, and identifiers. The model might consist of a number of subject areas, reflecting different departments in the organization.

Data modeling

A data model is a visual representation of how the data will be structured in a database. In an OLTP database, the data will be normalized to reduce repeating values and ensure an entity only has the attributes that belong to it. This leads to the best performance for random, small, and isolated transactions. A data warehouse denormalizes the data, so the database performs optimally for reporting.

A data model comprises a logical design and a physical design. There are two approaches to data modeling: a top-down approach, or a bottom-up approach. In a top-down approach, the model is created by gaining an understanding of the business requirements. The bottom-up approach creates a model from existing databases. A model is only a representation of the database, so it will contain objects such as tables, columns, and relationships that can be visualized. A database developer uses the model to develop the physical database.

Semantic models

A semantic model is a data model that includes information to give meaning to the data. The semantic information should enable the model to describe itself. Semantic models help to create consistency. The dataset of a semantic model uses inherent structures; in a database, the context of data is defined through its relationships with other data. Semantic data models give representation to real-world entities such as a Customer, Store, or Employee. A relational model breaks entities into parts, whereas the semantic model uses the entity to represent itself fully.

> **Question:** How does your organization approach BI? Is this a major part of the corporate strategy? What BI solutions does your organization use? Is Excel used as a self-service tool? What do you think are the major issues with your organization's approach to BI?

Lesson 2
Introduction to data analysis

This lesson breaks down the components of data analysis. It looks at using queries to extract data from a variety of data sources, using transformations to make imported data easier to work with, and using visualizations to present data.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe how to use data sources in BI.
- Understand how to use queries for extracting data from data sources.
- Explain why transformations are needed.
- Use visualizations to present data.

Data sources

A data source is the location or repository for the data you import into your data warehouse or reporting tools. In a traditional data warehousing scenario, an ETL package extracts the changed data from the operational database and loads it into a staging area, before applying transformations to prepare the data for loading into the data warehouse. Online transactional processing (OLTP) databases are designed for random access and are extremely fast for small transactions. They perform much less well at aggregations, whereas a data warehouse is designed to make this a faster process. Extracting data from operational systems, remodeling, transforming, and applying aggregations in the data warehouse is a lengthy process that requires considerable funding and resources in an organization of any size. In-memory data and real-time operational analytics have the advantage that the data does not need to be extracted to a secondary location—in-memory processing is designed for optimal performance and can better handle aggregations.

- The location, or repository, of the data for your BI solution
- Traditionally used ETL process, now held:
  - On-premises
  - In the cloud
  - In files

However, the data an organization wants to analyze is typically not confined to an on-premises database server. The online world in which third-party services and publicly available datasets interact with business operations is now very much part of the regular data landscape. The boundaries of data have expanded to disparate locations in the cloud. Data sources you are likely to add to your reports, include:

- **On-premises databases**

  Despite the current trend of moving databases to the cloud, most organizations hold some data on-premises. These might include your Microsoft SQL Servers, such as SQL Server Analysis Services (SSAS), Active Directory® (AD), Exchange, and Access® databases. Your organization might also use other main industry databases, including Teradata, Oracle, MySQL, Sybase, IBM DB2, SAP HANA, and PostgreSQL.

- **Cloud databases**

  Cloud is an increasingly popular choice, with Microsoft offering a wide range of Azure cloud services. These include Azure® SQL Database, Azure SQL Data Warehouse, Azure Marketplace, Azure HDInsight®, Azure Blob Storage, Azure Table Storage, Azure DocumentDB, and Azure Data Lake Store.

- **Software as a service providers**

  Organizations are increasingly turning to software as a service (SaaS) providers as a more cost effective option than the development of in-house solutions. Your organization might use third-party solutions such as Facebook, Marketo, and MailChimp, alongside Bing®, Google Analytics, GitHub, and Zendesk. Having the ability to use the data generated from these services is important for gathering a complete picture of activity in your data.

- **Files**

Most organizations hold data in spreadsheets and are likely to have data stored in Excel or CSV format. JSON and XML are popular languages for exchanging data between systems and should be supported by your BI solution as a data source. In addition, business users might have data stored in text format, which requires importing into the BI solution.

Queries

- Commands you run against the data source to specify the data to extract:
  - Return entire tables or run a query against the source
  - Use stored procedures against SQL Server databases
  - Only return the data that you need
- Expressions used to transform data:
  - M Query Language:
    - Use in Power Query Editor
    - Generate using menu options or edit query directly
  - DAX:
    - Use in Power BI Desktop
    - Derived from MDX and Excel formulas
    - Straightforward to use but very powerful

You use queries to extract data from your data sources. If you have connected to a database, queries specify the tables and columns that you want to export into your BI solution. Your BI solution might offer the choice of importing entire tables, or writing a query to specify the columns you want. If you are connecting to a database such as SQL Server, then using stored procedures to query the data is preferable. A stored procedure is a query that's stored on the server. Stored procedures are more efficient than specific, one-off queries, because SQL Server creates an execution plan, which it reuses each time the procedure is called. This plan works out the optimal way to retrieve the data, resulting in the fastest possible return of results. Stored procedures can also be used by other colleagues—sharing code prevents duplication of effort.

Depending on your role within the organization, you might be dependent on a database developer to write the queries or stored procedures for you and, for security reasons, you might not have access to all objects in the database. It's important that you only return rows and columns from the database that you intend to use in your reports. Not only does importing unnecessary data create additional network traffic, it also makes larger datasets more cumbersome to work with.

After you've imported your data into Power BI Desktop, you might still need to perform further customizations on the dataset.

- Using M in Power Query Editor

In Power Query Editor, you can use menu options or the M query language to transform your data. Transformations that you make using the menu options are written into your dataset query in M. This is particularly useful when you are unable to use queries to determine the data to import into your original dataset. For example, when you have limited access to the data source or are importing data from flat files.

- Using Data Analysis Expressions (DAX) in Power BI Desktop

You also use DAX in Power BI Desktop to transform the data in your reports. For example, your dataset might contain first name and last name fields—you can use DAX to concatenate them into a name field.

If you're an advanced Excel user who is familiar with Excel formulas, you will find Data Analysis DAX to be very much the same. Whereas Excel formulas operate at a row level, DAX is used with relational datasets. You might have already used DAX in Power Pivot or SQL Server Analysis Services tabular models. This powerful formula language has evolved from the Multidimensional Expression (MDX) language used for querying cubes, and has been merged with Excel functions. DAX offers a library of more than 200 functions, operators, and constraints that mean you can perform sophisticated transformations on your datasets.

If you're using Power BI Desktop for self-service BI solutions, you can use DAX to enhance the data you import, without having to depend on developers to do this. If you're importing data that cannot be altered until after it has been imported, DAX again comes in useful.

The following code uses DAX to concatenate the FirstName and LastName fields to create a new column called FullName:

Concatenate the FirstName and LastName Fields Using DAX

```
FullName = [FirstName] & " " & [LastName]
```

Data transformations

After you've loaded your data into Power BI, you might need to transform it into a different format for your BI reports. This is often the case when using multiple data sources but, in even the most straightforward systems, it's likely that some transformations are required. To accurately report on the data, you must ensure values are consistent if you intend to use them for filtering.

The following transformations are typically applied to data:

- Data must be transformed from its form in the data source into a compatible format for your reports:
  - Cleaning
  - Formatting

Cleaning

Before applying any transformations to your data, it's a good idea to clean, or cleanse, the data first. This process corrects dirty data or removes it to another area for investigation. You want the quality of your data to be as high as possible. Typical cleansing operations might include:

- Detecting dirty data as it is loaded and either applying a transformation to data that can be cleaned or filtering the dirty data into a separate table for further investigation into why it's incorrect in the source system.

- Removing duplicate rows.
- Eliminating incomplete rows.
- Performing logic tests to check date fields, such as seeing if a date is earlier or later than should be possible—for example, if the **Ship Date** is before the **Order Date**, then this data is dirty.
- Checking address and postal code fields are correct.
- Performing character pattern testing to ensure phone numbers and email addresses are in the correct format.
- Logging missing values.
- Checking that data matches the business rules. For example, only one Sales Person manages a single customer.

Formatting

When the data is clean, you apply formatting to ensure the source data is in a useful format for your end users. Typical formatting operations include:

- Concatenating columns. For example, combining **First Name** and **Last Name** into a **Full Name** column, or concatenating **Address1**, **Address2**, **City**, **Country**, and **Postal Code** into a **Full Address** column.
- Replacing shorthand values with full words to enable better filtering. For example, you could change **M**, **F**, and **U** values to **Male**, **Female**, and **Unknown**, or **S**, **M**, **D**, **W**, to **Single**, **Married**, **Divorced**, and **Widowed**. **True** and **False** values are frequently stored as **1** and **0** values in the source database, and should be converted.
- Changing the casing on text values. You might want to ensure country or state codes are all uppercase, and names and address all have title case, with the first letter of each word in uppercase, the rest in lowercase.
- Dates might need to be formatted to full date time values to enable filtering at a low level of granularity. The format of dates generally varies quite widely across systems, with no consistency, so you need to be aware of formats and ensure that datetime values are converted to the same format and locale.
- Currency and number fields should be formatted and handled carefully. Ensure decimal columns that undergo any rounding up or down do not skew figures and produce unexpected results. If accuracy is critical, you must ensure that values are entered correctly into the destination database. If decision makers are not concerned about precision and are happy with an approximate figure in aggregations, you will have more freedom to apply some formatting.

Visualization

- Human eye recognizes patterns
- Easier to see anomalies in charts and maps than tables
- Visualizations reveal patterns, clusters, and outliers
- Help make fast decisions about data
- Eliminates the need for the brain to process raw numbers

Evolution has given humans the ability to recognize patterns—this means we instantly read and deal with dangerous situations, helping us to survive. We very quickly identify irregularities, which means we recognize when a situation is no longer regular—something has changed and could be life threatening. Although we are no longer presented with the same dangers that early humankind endured, we have retained the ability to visually assess and make judgements within incredibly small timeframes. In the modern world of information, you might apply this innate ability to different scenarios, primarily including the reading of data.

The way in which data is presented affects how quickly and efficiently you can process and understand it. If you're presented with a table of numbers in a spreadsheet, it's likely that you would need to reorder the data and take time to work out the highest and lowest values; you might not notice clustering, outliers, or other patterns within the data. If you present the data on a map, or in a column or scatter chart, you might instantly see the high and low values, such as customers who spend most on products within a particular category live by the coast—or that males over 45 are the most popular return customers. The context within which you place the data affects its interpretation.

The power tools within Excel have no doubt increased its popularity as a data analysis tool. This is because users quickly take data that's in a table format and difficult to comprehend, and convert it into colorful charts and maps, which become instantly readable. Tables of data, even when ordered so values run from high to low or vice versa, still require us to read the numbers and compare rows of values. For example, when we view a colored pie chart, we can instantly see how the values are distributed by the size of the portions. Initially, we don't need to know the values behind the portions; we can make an instant assessment, and then start drilling down to obtain further detail. Visualizations are vital for helping us make fast decisions about business data. They effectively eliminate the need for the human brain to process raw numbers, search for patterns, or dig for outliers by manipulating the data.

Demonstration: Importing data with Power BI Desktop

In this demonstration, you will see how to:

- Import data warehouse data into Power BI Desktop.
- Remove columns.
- Format a column.
- Create a new column using a DAX expression.

Question: How much data does your organization gather? Have you noticed an increase in the volume of data that you have to work with? Do you have a mix of data sources, such as on-premises databases, cloud services, and SaaS providers?

Lesson 3
Introduction to data visualization

Data visualizations bring data to life, using colors and shapes to present data that would otherwise remain as text and numbers. This lesson explores how visualizations help you discover insights into your data that you would not otherwise find. The chart types in this lesson focus on the charts available in Power BI Desktop; however, the principles of charting components are generally standard across BI solutions and vendors.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the different types of chart available for presenting data.
- Use cards to display data.
- Use maps to show the spread of data in a geographic area.
- Use tables to organize data.
- Explain how the tree map works.
- Format charts.

Charts

You use the chart visuals in Power BI Desktop, to quickly create visually stunning and interactive reports and dashboards. You can select a chart from the **Visualizations** pane to add to the report canvas, or drag a data field onto the report to automatically create a table visual—that can then be converted to another chart type. For example, you could drag the **Categories** field onto the report, which automatically creates a table. You could then drag **Total Sales** onto the table, to add another column. Then you could click one of the chart icons in the **Visualizations** pane, and quickly switch between a bar or pie chart.

- Power BI Desktop includes a wide range of all the common chart types used in data analysis:
  - Bar and column charts
  - Line and area charts
  - Line and column charts
  - Funnel charts
  - Scatter charts
  - Bubble charts
  - Pie charts
  - Donut charts

Bar and column charts

Stacked bar and column charts are identical, except that the bars on a stacked bar chart span horizontally, rather than vertically, as in a column chart. Each chart accepts an axis field, such as Sales Person, and a Value; for example, Sales YTD.

Clustered bar and column charts are similar to stacked charts, but they include two data fields for the Value, which results in two bars or columns for each axis.

Again, 100 percent stacked bar and column charts are similar to stacked and clustered charts, except the bars and columns stretch the width or length of the chart area, and display the progress of each axis against a value. You add two data fields to the Value, such as Sales YTD and Sales Quota. If you need to display progress in attempting to meet a target figure, 100 percent stacked charts are useful.

Line and area charts

The line and area charts are fundamentally the same. However, the area chart is filled in, so the area below the line values appears as a solid block. Line and area charts are useful for displaying data over a period of time—such as financial data.

Line and column charts

The line and stacked column chart combines columns and lines. The columns and lines share the same data field for the axis—for example, Year. The column value could be Gross Sales, with a line value for Share Price. You might include multiple lines on a line and stacked column chart. You use the line and clustered column chart to include multiple columns for each shared axis.

Note: If your data creates a large number of data points—for example, hundreds of bars on a bar chart—the scrollbar will adjust so that it does not become too small. Instead, as you scroll to the end, more data is loaded, but the scrollbar remains a viewable size.

Other chart types

Power BI Desktop provides other types of graphical chart, including:

- Funnel charts
- Scatter charts
- Bubble charts
- Pie charts
- Donut charts

Cards

When you present data in a report or dashboard, you should take care to ensure the most important information is easy to find. If your audience normally reads from left to right, top to bottom, then displaying the most critical data in the top left, flowing through to less important content at the bottom right, is helpful. If you have important figures that need to be presented clearly, so that they are easily read, then the card and multirow card charts suit this purpose.

- Present most important data first:
  - If users normally read left to right and top to bottom, show most important data in top left
  - Use card, multirow card, and KPI visuals to present important figures clearly and efficiently
- Card chart:
  - Displays a single numeric value, such as Total Sales
  - Optionally displays data label and title
- Multirow card chart:
  - Shows multiple numeric values, useful for small datasets, such as Main Category and Total Sales
  - Optionally include the data labels and a chart title
- KPI
  - Visualize a business objective and show progress towards the goal

Card chart

The card chart displays a single value and a description. The numeric column values are aggregated to show the total value, such as Total Sales; the data label is the name of the field. Before using the card chart, ensure that the field to be aggregated is formatted correctly, especially if this represents financial data. If the Value column is not specified as a currency data type, then it shows only a number without the currency symbol. This should be included to make clear that it is a monetary figure. The data label can be turned off, but unless it's entirely clear what the figure refers to, this is best included. You rename the field by right-clicking it in the Fields pane, then selecting Rename. Again, be as clear as possible as to what this refers to. If you cannot change the name of the field, you can hide the data label and add a title instead. You format the card to change the

background color and transparency, format the card border, and change the font properties of both the data value, and the label and title.

Multirow card chart

The multirow card chart is a useful way to clearly present numbers, without using the format of a table or matrix chart—which are difficult to digest. Like tables, the multirow card chart works best for smaller data sets; otherwise, there is too much data and text to read. For example, a multirow card chart is useful for displaying main categories, and sales. You can also add a title to the multirow card chart and turn off the category label. Use the Format options to customize all aspects of the card, including adding a border, changing the background color, modifying font properties, and adding a back color to each data value.

KPIs

Key Performance Indicators enable companies to measure their progress towards a business objective or goal. You create KPIs at a high level to measure the overall performance of the company; you can also set KPIs at lower levels, such as by departments—for example, sales, call center, or warehouse. You add a KPI visual to your report in Power BI to track progress towards a target. Similar to the card visual, the KPI displays a single value such as TotalSales for the current year—this is the Indicator. The Target value is the goal, such as TargetSales. Add a data value such as Year to the Trend axis to display how well the target is being met. This is represented as a filled line chart and Power BI automatically colors the filled area using green, yellow, or red to show if progress is good, neutral, or bad. You use the Format options to change these colors.

Maps

Power BI Desktop includes a map chart and a filled map chart. You use these charts to map your data visually, both regionally and globally. Power BI integrates with Bing maps to find default coordinates for locations, based on a string value, in a process known as geo-coding. This integration means you do not need to provide longitude and latitude coordinates in your data—this is optional, because Bing makes a best guess at the location.

- Power BI integrates with Bing to identify location
- Map chart:
  - Represents data as proportionally sized, color-coded bubbles
  - Good for data based on cities
- Filled map chart:
  - Uses shading across a region; darker shades for higher numbers, or rather, high density
  - Useful for demographic data
- ArcGIS map chart:
  - Uses points, areas, clusters, heat maps
  - Can analyze your data against demographic layers

Map chart

The map chart accepts data for the Legend, Longitude, Latitude, Values, and Color saturation. The Legend property accepts fields such as City, County, and Province, and the Values property accepts numeric values such as Total Sales, or Number of Customers. The numeric values are presented as colored bubbles on the applicable location specified in the Legend property. The bubbles are sized proportionally to the data they represent within the field in the dataset; that is, the bigger the value, the bigger the bubble. The map chart is useful for presenting data based on cities, rather than wide areas.

Filled map chart

The filled map chart (also known as a choropleth) uses a slightly different visualization to represent the data. This chart uses shading, tinting, or patterns to represent the data value across a geographic area. The darker the color, the higher the value; the lighter the color, the smaller the value. This is particularly useful for presenting socioeconomic or demographic data, because it provides a visual overview of data across a wide area, such as all the states in the United States.

ArcGIS map

The ArcGIS map chart uses the ArcGIS technology provided by Esri. Before creating or viewing an ArcGIS map, the report builder or user must accept the Esri terms and privacy policy. You use ArcGIS maps to create informative map visualizations by using points, areas, clusters, or heat maps alongside demographic and reference layers.

Tables

You use table and matrix charts to add data fields to create columns and build up a table. Each numeric column is automatically aggregated, with a total at the bottom of the column. Using a table or matrix is useful when you want to display the actual numbers, such as for financial data.

- Display data in columns and rows:
  - Useful for displaying numeric data, such as financial
  - Each numeric column is aggregated
- Table:
  - Best for small datasets
  - Includes very little visual formatting
  - Data must be read to be understood
  - Consumes a lot of space on the report canvas
- Matrix:
  - Can add rows, columns, and values
  - Can enable drilldown

Tables are best used for smaller sets of data because, as the number of rows or columns increases, the information becomes harder to assimilate. To help alleviate this, the table chart includes the option to apply predefined styles, which makes the data easier to read. You set styling, such as alternate row highlighting, and use the predefined styles or select custom colors for the alternate rows, to format the table to your exact requirements.

Consider the following table, which would appear much the same in a Power BI Desktop report. The chart displays the total sales by category and country. It is consuming a lot of space, and requires you to read each of the values in the **Sales Territory Country** column, and then the figures in the **Total Sales** column. Furthermore, the values in the **Sales Territory Country** column are ordered alphabetically, which determines the order of the **Total Sales** column, making it difficult to compare the sales figures. You might be able to order by each column, but not by **Total Sales** within the **Accessories** category only.

Category Name	Sales Territory Country	Total Sales
Accessories	Australia	$81,309.16
Accessories	Canada	$59,758.93
Accessories	France	$37,421.30
Accessories	Germany	$36,908.60
Accessories	United Kingdom	$43,481.35
Accessories	United States	$148,170.91
Bikes	Australia	$2,440,928.44
Bikes	Canada	$581,424.73
Bikes	France	$870,221.82
Bikes	Germany	$1,025,888.91

Category Name	Sales Territory Country	Total Sales
Bikes	United Kingdom	$1,148,585.76
Bikes	United States	$3,095,275.19
Clothing	Australia	$41,646.69
Clothing	Canada	$32,444.55
Clothing	France	$14,535.92
Clothing	Germany	$14,093.26
Clothing	United Kingdom	$18,219.16

There is little difference in displaying data in a table in Power BI Desktop compared to Excel, or even a SQL Server Reporting Services report. You'll see that the table consumes space and takes time to read.

Visually, the table and matrix charts look quite similar; however, the matrix chart provides more functionality than a table. When using the matrix chart, you add rows, columns, and values to your data, in addition to implementing drilldown capabilities. Displaying the example table as a matrix would enable users to group the information by category or country, simplify the layout and help end users to better understand the data to make more informed business decisions based upon that data.

Conditional formatting

You customize the background color of a cell depending on its value, including the ability to use gradient colors. After you create a table in Power BI, in the **VISUALIZATIONS** pane, in the **Fields** bucket, right-click the field that you want to colorize. From the menu, select **Conditional formatting** and then select whether you want to color the text or the background, or use data bars. You then select the minimum and maximum colors and set the values to be that of the lowest and highest values in the data—or manually set the values. You can optionally add a center, or middle, value and color, by clicking the **Diverging** box.

Tree maps

The tree map might not physically represent a tree; however, the principle behind its function is representative of a tree with larger data scaling through to smaller data, as if the data were branches scaling down to twigs. For example, in Power BI Desktop, add the **Country** data field to **Group**, add **Territory** to **Details**, and **Total Sales** to **Values**. Each country is represented by a rectangle that is proportionate to the number of sales, so the countries with the most sales have the largest rectangles. Each country rectangle is subdivided into territory rectangles, with their size again being proportional to the value of sales in that territory. This style of representing data is classed as hierarchical.

- The tree map functionality represents a tree, even though it doesn't look like one:
  - Data represented as a rectangle or branch
  - Branch can be further divided into nested rectangles, or leaves of the branch
- Represents data hierarchically
- Efficient use of space
  - Flattens data to show two layers—for example, sales by country, with each county broken into territories
  - No need to drill down to see this data

Unlike a table or matrix chart, the tree map is more efficient in how it uses the space it consumes in a report. By showing both Country and Territory in the tree map, it has effectively flattened the data—you don't need to drill down to see categories for each territory.

Formatting charts

The visuals in Power BI include extensive options for customizing how your data is displayed. Some of the options available will depend on the type of chart.

- All charts can be customized with colors and borders:
  - Show or hide a chart title, change font color and size
  - Set X and Y position, width and height of each chart
  - Show or hide axis, data labels, or legends
  - Set colors of data points—for example all columns—or by each value
- Add shapes, text boxes, and images:
  - Use shapes to group related visuals
  - Use text boxes to add headers or create hyperlinks
  - Add corporate logos, pictures, or photos to enhance report
- Right-click bar or line: drill down to underlying records
- Customize tooltips by adding extra fields
- Quick measures quickly change the aggregation on a field
- Add trend, constant, and dynamic reference lines to charts

Settings

Each visual can be customized with colors and other settings using **Format**, so you can easily use corporate colors to ensure your Power BI reports match the look and feel of business-specific colors. This is particularly useful if you use the embedding tools to include visuals within your own custom applications or websites.

The title of each visual can be customized. Included by default, you turn off the title to hide it completely, or change the text, font color, text size, and background color, and set the alignment of the text to left, right, or center. You can also choose to lock the aspect of the visual. Under the **General** settings, you can configure the X Position and Y Position of each visual, and specify Width and Height, ensuring your visuals are of a consistent size in your reports. All visuals enable you to add a border, which is not included by default, and you can change the color of the border to suit your design requirements. Each group of settings includes a **Revert to default** button to reset the visual and remove any formatting you have applied.

Other settings include the ability to show or hide axis, data labels, or legends, and set the colors of data points. With a column chart, you change the color of all columns, or set them individually, based on each data value. This is helpful if your report shows consistent data, such as sales by department or category, where the department or category is represented by color. For a supermarket, fresh fruit and veg might be represented with green, frozen food with blue, pet food with brown, and so on. For each visual, click **Format** to see the available options.

Shapes, text boxes, and images

In addition to data-bound visuals such as column charts and maps, you can also add static features to further format and customize your reports.

You use shapes to highlight or group items in a report. From the **Insert** group on the **Home** tab, click **Shapes**, and choose from Rectangle, Oval, Line, Triangle, or Arrow. For example, you could use the rectangle shape to group one set of visuals that contain data pertaining to sales, and another to group visuals referring to product returns. You might also use the line shape to divide the report into sections using horizontal and vertical lines. The arrow shape can be used to point to a spot on the report to which you want to draw your colleague's attention. Each shape can be customized, and you can change the border and background colors, or add a title.

Including a text box in your report is a useful feature for adding titles or extra headings to visuals. For example, you could add a main heading to the report, and then a subheader to a group of visuals. To add a text box, from the **Insert** group on the **Home** tab, click **Text box**. Type the text into the main box then

you can format the face and size of the font, set bold, underline, italic formatting, and alignment—or add a background color. You can also create a hyperlink using one or more words in a text box.

You add images from the **Insert** group on the **Home** tab, by selecting **Image**. Browse to the image you want to add and click **Open**. The image appears on the report canvas and you can add a title, turn on the background color and set the transparency, and add a border. This is useful for adding logos to your reports so they adhere to corporate design. You can also add a photo to a report.

Drill through

Power BI visuals automatically include the ability to click a data point, such as a bar, line, or portion of a donut chart, and it will display the underlying records. For example, right-click a bar in a bar chart and click **See Records** to show a list of the underlying data, or click **See Data** to display both the visual and the aggregations for each bar. This is available in both Power BI Desktop and the Power BI service.

Customizable tooltips

By default, visuals will display a tooltip that includes the data point's value and category. You add other fields to the tooltip by dragging a field from the **FIELDS** pane, to the **Tooltip** bucket on the **VISUALIZATIONS** pane. Right-click the field in the bucket list to choose from additional aggregations that you apply to the field.

Quick measures

Use the quick measures feature to quickly change the aggregation that is applied to the data in a visual. The default aggregation function is Sum, but you change this by right-clicking the **Values** field in the VISUALIZATIONS pane, and choosing a different function, such as Average, Minimum, Maximum, or Count. You can also show the values as percentages of subtotals or grand totals already included in the dataset.

Reference lines

Use the **Analytics** pane to create trend, constant, and dynamic reference lines on selected visuals. A constant reference line is located at the value you specify—for example, 10 million on a sales bar chart—regardless of the underlying data. You use dynamic reference lines to add lines based on minimum, maximum, or average, which change dynamically depending on the underlying data. You can also have multiple lines on one chart, including more than one constant line. You customize each line by changing the color, transparency, and dash type—and whether the line sits in front or behind the data points. The lines that you can add depend upon the visual that you use.

The following visuals can include all lines:

- Area chart
- Line chart
- Scatter chart
- Clustered column chart
- Clustered bar chart

The following visuals can only include a constant line:

- Stacked area
- Stacked bar
- Stacked column
- 100 percent stacked bar

- 100 percent stacked column

The following visuals can only include a trend line:

- Nonstacked line
- Clustered column chart

Demonstration: Visualizing data with Power BI Desktop

In this demonstration, you will see how to:

- Add visualizations to a Power BI report.
- Apply basic formatting to the visualizations.

Check Your Knowledge

Question	
Which of the following is not a real chart type?	
Select the correct answer.	
	100 percent stacked bar chart
	Line and column chart
	Multirow card chart
	Donut chart
	Pie and line chart

Lesson 4
Overview of self-service BI

This lesson describes how the recent growth in data has driven the need for self-service BI solutions, and compares managed enterprise BI to self-service BI solutions.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand how prolific data growth has affected and driven the BI market.
- See how managed enterprise BI solutions limit users.
- Explain why self-service BI has become such a popular choice.

Data explosion

The term "big data" was recently plunged into the limelight to describe the vast quantities of unstructured data being generated in our technology-driven world. It is now a common term used not only by the CTOs and by CIOs in the boardrooms of major global organizations—such as Microsoft, Amazon, and Facebook—but also by organizations in all sectors and of all sizes. It seems that big data is unavoidable. Big data is too large for traditional software programs to capture, store, and manage, and presents a challenge to businesses wanting to analyze this data.

- Big data is high-volume, unstructured data
  - Generated as a result of a technology-driven world
  - Characteristics:
    - Volume
    - Variety
    - Velocity
    - Variability
    - Veracity
- BI data: structured in DW, is useful for measures, and KPIs
- Big data: reveals relationships

Big data is described using the following characteristics:

- **Volume**: this is the quantity of data generated and stored. The data must be large enough to be considered big data, and the size is also a determining factor of the value—and whether insights can be gained from it.
- **Variety**: this refers to the type of data. For example, data taken from a Facebook feed would gather text, photos and images, and video.
- **Velocity**: this is the speed at which data is generated and processed. Big data can be available in real time, using in-stream technology to view it as it's in motion.
- **Variability**: this refers to the consistency of the data; that is, how much does it vary? Inconsistency causes issues with data processing and management.
- **Veracity**: this is the quality of the captured data. The higher the quality, the better the results.

Organizations already have a lot of data, and the volume is constantly growing, with big data expanding from terabytes to petabytes. It is not easy for business to cope, especially if an organization considers all data to be valuable—and does not know how to separate any data that is not useful. However, big data does have a shelf life, and before too long, becomes worthless. In addition, there is a cost associated with storing and managing the data.

Difference between big data and BI data

BI data is extracted from operational systems and processed using ETL. The staging area enables the data to be highly structured, consistent, and organized, ready for loading into the data warehouse. The data is highly dense, trends can be highlighted, and data can be measured. Because of its size and unstructured format, big data requires a new approach when it comes to processing and analyzing. The data is not dense, but is a patchwork of clustered information. Rather than using measures and KPIs, the nonlinear format of big data reveals relationships and dependencies, and predicts behaviors.

Cause of big data

The Internet of Things (IoT), and social media—with their usage facilitated by mobile devices—are major contributors to the generation of big data that is unstructured and difficult to process. The IoT is a network of objects embedded with software, electronics, and sensors. Built-in network connectivity enables devices, buildings, and vehicles to communicate and exchange data. Increasingly, IoT technology is entering our homes, built in areas such as fridges, thermostats, fitness wristbands, and AV equipment. Not only are these devices gathering data, we can also often control them remotely. Social media websites such as Facebook, Yammer, Twitter, and LinkedIn, all operate on the connection of interpersonal relationships, generating data containing a variety of text, images, photos, hyperlinks, and video.

Limitations of managed enterprise BI

The nature of software development—for example, web applications, database development, or report creation—means a project can take a long time to come to fruition. IT departments are frequently overloaded with user requests for new features, or changes that need to be made to existing systems. This can be obstructive to users wanting to do their work, because they are waiting on a developer as an available resource to complete the task. IT departments, especially development teams, often have a backlog of work.

- Development work is generally a slow process
- IT departments frequently have a backlog of requirements
- Main limitations of managed enterprise BI include:
  - Time
  - Budget
  - Developer cost
  - Lack of business knowledge
  - Changing requirements

The main limitations of managed enterprise BI include:

- **Time**: one of the biggest factors in managed enterprise BI is the time taken to develop the ETL system, build the data warehouse and cubes, write code to query the data, and design, develop, and publish reports. Even in a small organization, this is not a quick process—it requires planning, and a team of skilled developers. Much of the work is often centered on transforming the data in the staging database after extracting it from the source systems. This is ongoing work, because anomalies that arise from the source systems must be continuously monitored and fixed. Furthermore, the design and development of reports can be a slow process, especially if there is a lot of detailed information over several pages.
- **Budget**: the budget is linked to the time it takes to build the BI infrastructure and associated code base. The amount of work required up front before anything tangible is delivered is often a concern for stakeholders. Developers might be working hard creating the ETL and data warehouse, but until reports are delivered, stakeholders and users do not see that anything is actually being done. This can be difficult for nontechnical users to understand—why must they wait so long for what they consider a straightforward report? The cost of hiring BI developers is also expensive, especially if contract staff are required solely for the length of time it takes to deliver the project.

- **Developer cost versus business user cost**: in many instances, the cost of employing a report developer is costlier than that of business users. It makes financial sense, therefore, to empower the business user to create their own reports.
- **Lack of developer knowledge**: while a report developer might be highly technically skilled, they do not necessarily possess an understanding of the business, or the data. If this is the case, the developer is unlikely to produce a report that details exactly what the user needs. This can be frustrating when a user has been waiting for a developer to be available to create the report, only to find it is not what they need. A request for change is then submitted, and the user must wait for this work to be done. However, a further request does not guarantee that the developer will produce what they need.
- **Changing requirements**: in addition to user requests to change reports that do not actually give the user the data they need, developers must cope with new requirements, and increasing volumes of data. For example, with sales, marketing, finance, and support departments all using SaaS data sources—requiring publicly available datasets to be included in their analysis, and statistics from customer data and internet usage—the developer must continuously integrate new data.

However, even if an organization handed over all report development to the business users, there would still be a requirement to build the ETL and data warehouse, provide access to the databases, ensure security is properly implemented, and assist users with complex queries.

Self-service BI trend

Nowadays, big data is less about it being big, and more about an organization's ability to extract useful insights from it, to improve company performance. Many SaaS providers, such as MailChimp and Google Analytics, already offer some level of data analysis to their customers. However, this usually involves the customer logging into the SaaS portal to view the data. Having the ability to download data from MailChimp, Twitter, and Facebook—and combine this with a marketing campaign created in Marketo—offers more cohesive insights. Being able to analyze data, and react to it quickly, requires a quick turnaround time for processing data. Dependency on an IT department delays this considerably, so business analysts utilizing a self-service BI approach have greater gains from their data.

- Big data:
  - Less about being big, more about an organization's ability to extract useful insights
  - Users need to combine data from various sources
  - Data analysis needs to be done quicker
- Self-service BI:
  - Business users can access corporate data and perform analysis without possessing technical skills
- Popularity driven by:
  - Excel power tools
  - Increase in affordable solutions from software vendors, such as Tableau and Qlik

What is self-service BI?

Self-service BI is an environment in which business users access corporate data to produce their own reports, without dependency on IT. Until quite recently, BI was held tightly in the realm of specialists, who were highly skilled in the use of the tools on offer. Now, with modern self-service BI tools, users do not need to have IT skills in writing complex database query code, developing data warehouses, reports, or data mining. Self-service BI tools do most of the hard work, enabling the user to quickly produce data that's suitable for analysis and can be shared with colleagues.

Why is self-service BI so popular?

Using a self-service BI tool frees up IT departments, and means business users can generate reports exactly how they want them. When thinking of self-service BI, Microsoft Excel initially comes to mind. Its popularity as a spreadsheet program, ideal for day-to-day number crunching, was boosted by the inclusion of the four additional power tools—Power Query, Power Pivot, Power View, and Power Map.

These tools take data from a tabular format that is difficult to read, enabling external data connections, data formatting and manipulation, and a whole host of charts and maps to present the data, and perform deeper analysis. Adding these tools into a program with which millions of users were already familiar, takes BI from the boardroom, and gives the power of analysis to the business user.

Furthermore, a wide range of tools are on offer in the self-service BI solutions marketplace, ranging from Microsoft's Power BI suite of tools—which is available on a free license—to solutions from popular vendors such as Tableau, and Qlik, that are priced considerably higher. Yet, while the license fees may initially appear steep, return on investment (ROI) of this initial financial cost is recouped when compared to the time cost of employing a report developer to manually create equivalent reports, and manage them. These tools can also deal with unstructured data better than spreadsheets, which need data in tabular format before any visualizations are applied. With the ubiquity of big data in business, it is fast becoming a requirement that a BI tool should cope with the challenge.

Question: Given what you have learned so far in this module, regarding the limitations of managed BI and the uptake of self-service BI with all its advantages, do you think there is a future for managed BI?

Lesson 5
Considerations for self-service BI

This lesson looks at some of the important aspects to consider when planning a self-service BI solution. This includes issues users might have when accessing data, the importance of data source reliability, how users require analysis skills, and how a data steward can help.

Lesson Objectives

After completing this lesson, you will be able to:

- Explain issues that arise when accessing data in a managed and a self-service BI solution.
- Understand why the reliability of data sources is important.
- Describe how users need some expertise in data analysis.
- Explain the role of the data steward.

Data access

By using self-service BI, users connect to a wide variety of data sources, including on-premises databases and data warehouses, local files, cloud services, SaaS hosted solutions, and public datasets. While managed BI solutions tend to be more highly controlled by policies maintained by IT, self-service opens up the possibilities for importing data from anywhere, outside the control of IT.

- Self-service BI enables users to connect to a wide range of data sources:
  - On-premises databases and data warehouses:
    - Can easily control access
  - Local files:
    - Difficult to restrict access—files easily transferred and shared
  - Cloud:
    - Can secure own cloud databases
  - Public data:
    - No control over access
- Data traffic increased due to one-off queries

On-premises data

Self-service access to on-premises data is generally controlled by IT. Data can be controlled in how it is shared with users, through database security rules to restrict access to sensitive data. For example, users of SQL Server databases are given access to data views, which provide selective fields, without giving full access to other sensitive data. It is imperative that data is protected and controlled, and that business users have access to the data they need to do their job.

Data from files such as Excel, CSV, text, and XML, are emailed, shared, and imported into a self-service BI solution. It is harder for IT to control and secure access to this data, because it is easily transferable, both within the organization and externally.

Cloud and public data

Self-service BI enables business users to take advantage of publicly available data. Data repositories, such as Microsoft Azure Marketplace, Amazon Web Services, and Wikipedia, all provide datasets, some of which are free. These can easily be incorporated into a self-service BI solution, by downloading the data, or by connecting directly to the source using a URL from within the self-service BI solution. This provides quick and easy access to very useful data that enhances the analysis of existing corporate data.

Databases stored in the cloud, including Microsoft Azure SQL Database and Microsoft Azure SQL Data Warehouse solutions, can be managed by IT with the same security principles applied. Users connecting to cloud-based data stored by SaaS providers require a username and password.

Data traffic

Enabling users to access large datasets and transfer data by sharing reports can cause issues with the volume of data moving around the network. With many users accessing data in a specific, one-off fashion, the load on the network increases. IT needs to monitor the performance of servers and networks to prevent bottlenecks. For managed BI solutions, this is less of an issue because precompiled queries executed against the database provide better performance, and data is cached.

Data reliability

Data reliability refers to the condition of data, and whether it is complete and sufficiently free of errors, so that it's fit for purpose. This is particularly relevant to data imported from public sources. To be complete, the data fields must be sufficiently populated. A dataset with a sparse population of data across many fields and rows cannot provide suitable results. The data need not be entirely free of errors, but the errors that do exist must not be severe enough to make the user doubtful of the results and question their validity. The data within each field should accurately represent the field, be correct, and of the correct data type. This ensures that the data is analyzed with confidence.

- The condition of the data:
  - Complete, error free, and fit for purpose
  - Most relevant to publicly available datasets
  - Data fields should be densely populated to be useful
  - Errors should not be severe enough to cause doubt
- Risk analysis :
  - Will the data be used for critical decision-making?
  - Will it influence policy-making or legislation?
  - Is the risk of using it high, medium, or low?
- Question the data source, frequency of refresh, the data owner, connection, and structure

Risk analysis

Risk analysis is a useful and often essential exercise to perform on data that is imported from sources external to the organization. If you need to make serious decisions after analyzing the data, then consideration must be given to the reliability of the data. In such circumstances, the following questions should be considered:

- Is the data to be used for critical decision-making by an organization or individual?
- Will the figures be used to influence policy-making or legislation?
- Is the risk of using the data high, medium, or low?
- Is the data of a sensitive nature?
- Will the results of the data be made available publically?

When performing risk analysis to determine the reliability of the data, the following questions should be answered as part of the assessment:

- **Data source**: where has the data come from? Is the data provided by a reputable organization?
- **Data refresh**: how often is the data refreshed? Does the analysis that uses the data require it to be kept up to date, for the reported results to be useful and accurate?
- **Data owner**: who owns the data? Does the organization require any permission to use the data? Is it permissible to publish reports that include the data?
- **Connection**: are there likely to be any issues connected with the data? What is the up time of servers on which the data is stored? Will the data always be available, or is there a time limit?
- **Structure**: will the structure of the data change, thereby requiring the dataset to be reimported?

Data from on-premises databases that store corporate information do not need to undergo such extensive risk assessment. Data should already be qualified, especially if it is derived from a data warehouse that has been designed and managed in-house.

User expertise

Self-service BI solutions require less technical knowledge than is needed to produce a managed BI solution. The purpose is for users to create reports as quickly as possible, with the least amount of effort, so that time and energy can be spent on analyzing the results of the reports. However, having knowledge of the business, formatting data, and understanding which visualizations best display the data, are useful for making the most of the BI solution.

- Self-service BI solutions require less technical knowledge than developing a managed solution:
  - Designed for least effort and quick to create
  - Enable users to concentrate on analyzing the results
- Accessing data—users should know where the data is located and how to access it
- Formatting data—skills are needed to clean, concatenate, format, filter, and exclude data
- Displaying data—users should be able to choose the correct chart type to accurately display data

Accessing data

Users need to know where data is located, and how to access it. Data stored in on-premises databases or data warehouses are supported by IT, so there is likely to be scope for a developer to provide queries, or offer advice on exporting data. External guidance might be required for accessing data held by third parties, including SaaS providers, and publicly available datasets.

Formatting data

Transforming and formatting data is an important step in the process of building reports. If the data is not right, then the results will not be right. Users must understand the principles and structures of data that is sourced from a relational database, a data warehouse, or an unstructured big data source, such as a social media site. Skills are required to:

- **Perform data cleaning**: remove duplicate rows, handle dirty data, and errors.
- **Concatenate data**: create new columns by combining existing columns.
- **Format data types**: ensure currency, number, and datetime columns have the correct data type.
- **Apply adequate filtering**: ensure data can be filtered to the expected granularity. How do sales need to be measured? Do "days" represent a fine enough granularity or does the report need to show online sales by the hour?
- **Exclude redundant columns**: columns and rows that are not needed in the dataset should be removed, to make the dataset easier to manage and understand.

Displaying data

Users should be familiar with all the major chart types and understand how to use them to display data most effectively so that decisions can be made. For example, geographic data is best presented using a map chart; a scatter chart should be used to show overlaps in data, clusters, and outliers. Financial data, such as a share price, is best displayed using a line chart. There are plenty of free internet resources that show examples of all the chart types and how they can be used. This will help self-service BI users quickly become familiar with chart types. Users should also understand how to create and use measures and KPIs.

Data stewards

The data steward role is aligned more with the business than with IT. A data steward ensures that the quality of data in an organization is high and is responsible for data governance. With the proliferation of data in organizations, a data steward is considered less of a luxury, and more of an essential role. A data steward has a varied role in managing data, and is responsible for:

- More business focused than IT focused:
  - Ensures quality data in the organization is high
  - Responsible for data governance
  - Manages data
  - Has skills/understanding of:
    - Business knowledge
    - Technical writing
    - Data modeling
    - Relational database management systems
    - Data warehousing
    - Nonrelational database systems
    - Programming
- Big data presents fresh challenges to this role

- Master data management.
- Ensuring the consistency of data between systems.
- Mapping data between different systems.

The data steward is responsible for managing data in the following ways:

- Removing duplicate data, particularly lookup data, or data that should be stored once.
- Removing unused, out of date data; for example, a product category that is never used.
- Removing ambiguous data.
- Checking data is fit for purpose.
- Securing data to ensure only authorized users can make amendments.
- Documenting metadata.
- Ensuring the organization adheres to data-related legislation.
- Determining data security requirements.
- Monitoring the quality of data.
- Developing data definitions.
- Establishing naming standards and conventions.
- Documenting business rules.

The data steward should either possess skills in, or a thorough understanding of, the following areas:

- **Business expertise**: the role of the data steward sits more with the business side than with IT. It is crucial that a data steward understands how the business functions and has departmental knowledge of all business areas, such as finance, marketing, sales, enterprise resource planning (ERP), manufacturing, retail, and supply chain.
- **Technical writing**: the data steward is responsible for documenting the metadata and should be able to write clearly, and with accuracy. The documentation spans multiple departments within the organization and must be clear to all who read it.
- **Data modeling**: although data modeling experience is not necessary, the data steward works closely with the technical architect and, at the very least, needs an understanding of terminology.
- **Relational database systems**: an understanding, or preferably first-hand experience, of relational database management systems is vital for the data steward. This role works closely with database developers, so knowledge is crucial.

- **Data warehousing**: understanding data warehouse concepts, including ETL, is also essential for the data steward to communicate effectively with BI developers.
- **Nonrelational database systems**: the emergence and pervasiveness of big data requires an understanding of unstructured, large volume datasets, the issues of managing them, and the technology required to process them.
- **Programming skills**: understanding programming and being able to directly manage data in the database is a useful skill for the data steward.

Data that is managed by a data steward will be of a higher quality than data that is not. This quality will be reflected in the data that is presented to customers, and data used in reporting and analysis. The growth of data provides continuous challenges to the data steward—the rise of big data demands another element of management that is less easy to apply, given the size and lack of structure.

Question: Discuss the role of the data steward. Does your organization have a data steward? If not, do you think one is necessary? Discuss some of the issues your organization faces, that your existing data steward manages, or that the addition of one could solve.

Lesson 6
Microsoft tools for self-service BI

This lesson reviews the different self-service BI solutions currently offered by Microsoft, and looks at the benefits and restrictions of using each solution.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the main features of SQL Server Reporting Services.
- Understand how Excel is currently used as a self-service BI solution.
- Explain how SharePoint® Online is used for sharing and collaborating.
- See the benefits of using Power BI Desktop as your self-service BI solution.
- Understand the capabilities of Power BI Report Server.

SQL Server Reporting Services

SQL Server Reporting Services (SSRS) is part of the SQL Server family, comprising the reporting component of the Microsoft BI stack. SSRS was introduced in 2004 as an add-in to SQL Server 2000. Since then it has grown to be a popular reporting choice for organizations running SQL Server. Reporting Services is generally installed as a stand-alone instance, as report generation requires much hardware resource, and SSRS works best on a dedicated server. Servers exist on-premises and security is managed using Windows authentication and Active Directory (AD).

- Part of the SQL Server family:
- Reporting element of the Microsoft BI stack
- Installed on stand-alone, dedicated server
- Secured using Windows Authentication/AD
- Reports created by:
  - SSRS developers using Report Designer in Visual Studio
  - Business users using Report Builder
- Data cached on server to speed report generation
- Users subscribe to report schedules

Developing reports

Organizations using Reporting Services usually have a dedicated report developer to create and update the organizational reports. The developer will have skills to query the relational database (OLTP) system, and the data warehouse if one is used. Report Designer in Visual Studio® is the main development environment for creating reports for SSRS. Usually, the developer accepts user requests to create a report based on a specification.

Reports can span multiple pages, and SSRS reports are particularly adept at managing data tables that expand to fit the size of the data, which might be unknown at design time. SSRS also supports HTML5 rendering and mobile reports, in addition to a wide range of charts, including sunburst and tree maps.

Business users with more advanced skills can create their own reports using Report Builder.

Deploying reports

After developing a report, it is deployed to the Report Server. The dataset is deployed alongside the report, and the data can be cached for faster report generation. This is useful when multiple users access the report, but the data is not frequently updated, because it delivers a faster experience.

Report subscriptions

By subscribing to scheduled reports, users receive an email message with a report attached. With the right permissions, users generate reports using the Report Manager portal, and by subscribing to report subscriptions. Reports are delivered as soon as data is updated, or can be emailed after the data warehouse load has completed overnight, so that the report is available at the start of the business day. Reports can be sent to users outside of the organizational domain.

Excel

Microsoft Excel has a loyal following of business users, and its leadership in the spreadsheet software market has long remained unchallenged. The addition of the power tools—Power Pivot, Power Query, and Power View—moved Excel to new heights, bringing self-service BI to its massive fan base. A key driver in the recent uptake of self-service BI was the enabling of business users to analyze and report on data without dependency on a managed BI solution. These power tools have liberated business users, and reduced the workload on IT to develop and manage a time-consuming BI solution.

- The addition of the three power tools to Excel was the key driver in growing the self-service BI trend:
  - **Power Pivot**: work with millions of rows, model data with DAX, create relationships, measures and KPIs
  - **Power Query**: renamed Get & Transform in Excel 2016. Import data from external data sources, including local files, on-premises and cloud databases, SaaS providers, and Hadoop. Transform, format, and combine data. Share queries using Power BI Data Catalog
  - **Power View**: create interactive visualizations, drill down into data, create new relationships, and KPIs

Power Pivot

Power Pivot was launched as an add-in to Excel in 2010, but since Office 2016, this is now included as part of the standard installation. This feature enables advanced data modeling, and data analysis—much of Power Pivot's strength lies in its ability to handle large datasets that have been imported from different data sources. You use Power Pivot to convert raw data into useful, visual charts and maps, helping you discover business insights, and trends. Using Power Pivot, you can:

- Import millions of rows of data from different data sources, including external sources.
- Model data using DAX functions.
- Create relationships between tables of data, including tables from different sources.
- Integrate with the other power tools to create charts, pivot tables, maps, and interactive Power View visualizations.
- Add measures and KPIs to your data model.

Note: To use Power Pivot in Office 2016, open Excel, on the **File** menu, point to **Options**, and then click **Add-ins**. In the **Manage** dialog box, click **COM Add-ins**, and then click **Go**. Select **Microsoft Power Pivot for Excel**, and click **OK**.

Power Query

Since Excel 2016, Power Query is known as **Get & Transform**, and the tools are located on the **Data** tab in Excel. You can use Get & Transform to search for data sources, connect to the data source and import the data, and then shape the data ready for visualizing. With Get & Transform, you can:

- **Connect**: you connect to local files including an Access database, CSV, or Excel file, data stored in the cloud, and data located on the internet, in addition to on-premises SQL Server, and SQL Server Analysis Services databases, Oracle, and MySQL. You can also connect to SaaS providers such as Facebook, and Salesforce, and big data sources including Hadoop.
- **Transform**: you can transform data using the Query Editor. Transformations enable you to shape your data so it is in the structure and format required to fulfill your reporting and analysis objectives. You can create new columns, remove columns and rows, and split columns. You alter data types to ensure number and currency values are aggregated and displayed correctly. Text data can be cleaned and trimmed, and the case changed to upper, lower, or title. You can also write your own transformations using the M Language.
- **Combine**: you combine rows from different tables to create a new table, and you can append rows from one table to the end of the rows in another table.
- **Share**: rather than saving your workbooks and distributing them to colleagues using email, you can share the queries in your workbooks to the Power BI Data Catalog. You can also Merge and Append queries.

Note: The data sources you connect to depend on the license you have. Some sources are only available to Professional, and Professional Plus license holders.

Each of the steps you perform as part of Get & Transform records in the Query Editor, enabling you to undo, redo, reorder, and even modify steps using the M Language.

Power View

Power View is an interactive visualization tool that you can use to quickly build a model, using the drag-and-drop interface. You can use advanced pie charts, maps, and data hierarchies that enable drill-down into your data. In addition, you can create new relationships and add KPIs based on these new relationships.

Note: To use Power View in Office 2016, open Excel, and on the **File** menu, point to **Options**, and then click **Add-ins**. In the **Manage** dialog box, click **COM Add-ins**, and then click **Go**. Select **Microsoft Power View for Excel**, and click **OK**.

SharePoint Online

In addition to using SharePoint data in your Power BI reports, you can also share Power BI reports in SharePoint Online.

- Share Power BI reports in SharePoint Online:
  - Publish Power BI report
  - Get URL for published report
  - Add Power BI web part to SharePoint page
  - Set report link property to report URL
- Power BI security settings maintained in SharePoint Online

After publishing a report to the Power BI Service, you locate the URL for that report. When you add a Power BI web part on SharePoint Online, you specify the URL of the report, which embeds the report into your SharePoint page. Only users who have been granted access to the report in the Power BI Service are able to view the report, so security is maintained in Power BI and there's no risk of unintended access being given.

The Power BI web part requires the user to have a Power BI Pro license. If they do not have one, when they access the page containing the web part, they will be directed to the Power BI website to purchase a license or sign up for a trial.

Note: You can only use the Power BI web part in SharePoint Online, not on-premises SharePoint Server.

Power BI Desktop

- Share many Excel power tool features
- Data sources: include files, on-premises databases, cloud data sources, and SaaS providers
- Transformation: apply same transformations and formatting in Power Query Editor as with Excel
- Reports: create stunning reports for publication
- Dashboards: create dashboards using tiles from different reports and share them with colleagues
- Power BI Mobile: app for iOS and Android

Power BI Desktop shares many of the features offered by the Excel power tools, so business users will find transitioning between the two tools to be a straightforward process. Power BI Desktop is a stand-alone tool that enables you to import data, model and apply transformations to your data, and then create stunning, interactive reports.

You can download the 32-bit or 64-bit version of Power BI Desktop from **https://www.microsoft.com/en-us/download/details.aspx?id=45331**. It is updated monthly with new features and capabilities. A message in the user interface will notify you when an update is available to install directly from the product.

The original Power BI Desktop application enables you to upload reports to the online Power BI service, where colleagues share reports, and create dashboards. Power BI is available on a Free license, or a Professional license that offers extra features, and supports a higher volume of data for a small monthly fee.

Power BI Desktop, optimized for Power BI Report Server, provides the same report creation functionality, but enables you to upload reports to the on-premises Power BI Report Server. Power BI Report Server is available through either a Power BI Premium or SQL Server Enterprise Edition with Software Assurance purchase. For more information about Power BI Desktop optimized for Power BI Report Server, see *Install Power BI Desktop optimized for Power BI Report Server* in the Power BI documentation.

Install Power BI Desktop optimized for Power BI Report Server

http://aka.ms/AA52x7l

Data sources

From Power BI Desktop, you can connect to a wide range of data sources including:

- **Files**: you can import from Excel, CSV, XML, Text, JSON, a folder, or a SharePoint folder.
- **Databases**: all the main industry databases are supported—SQL Server, Access, SQL Server Analysis Services, Oracle, IBM DB2, MySQL, PostgreSQL, Sybase, Teradata, SAP HANA, Amazon Redshift, Impala, Google BigQuery and Snowflake.
- **Azure**: Microsoft Azure SQL Database, Microsoft Azure SQL Data Warehouse, Microsoft Azure HDInsight, Microsoft Azure Blob Storage, Azure HDInsight Spark, Microsoft Azure Cosmos DB, Microsoft Azure Data Lake Storage.
- **Online services**: the main SaaS providers are supported, including Dynamics 365, Facebook, Google Analytics, Salesforce, GitHub, MailChimp, Marketo, QuickBooks Online, Webtrends, and Zendesk.
- **Others**: you can also import from a webpage, an OData feed, Hadoop, Active Directory, Microsoft Exchange, ODBC, and R Script.

Power BI Desktop supports DirectQuery, which you can use to query the data source, rather than importing the data. This is helpful when analyzing very large datasets.

Transformations

You can use Power BI Desktop to transform your data, and the Power Query Editor feature includes the same functionality as Get & Transform in Excel. With DAX for Power BI, you can choose from more than 200 functions, constants, and operators, to help shape data exactly how you need it. DAX for Power BI is slightly different to Excel, as it works at the column, rather than the row level. You can also create calculated columns, calculated tables, and measures, in addition to using the measures in your functions.

Reports

After importing and transforming your data, you can drag visuals or fields onto the report designer, to begin building reports. The visuals are customized with colors, titles and text, and other settings applicable to each type of chart. You can also create or download custom visuals for your reports.

Dashboards

One of the most powerful features of Power BI is the ability to quickly and easily share reports, dashboards, and datasets. After publishing a report, you can use the report items, known as tiles, to create a new dashboard, even combining charts, maps, and KPIs from different reports. With the Power BI Service, Professional license holders can create content packs. A content pack is a bundle of reports, dashboards, and datasets that make it easy for colleagues to share their work. Users on a Free license can download and view content packs. Reports are published to the Power BI service, or Pyramid Analytics.

Power BI Mobile

Power BI offers a mobile app for iOS and Android devices. Reports and dashboards automatically adjust their size to fit the screen of the device, so you need not worry about creating mobile versions of your work. The apps are free to download, and reports and dashboards are fully interactive.

Power BI Report Server

- On-premises report server for in-house reporting
- Uses SSRS reporting tools
- Functionality of SSRS, and more
- Hosts:
  - Power BI reports
  - Paginated reports
  - Mobile reports
- Requires an instance of SQL Server during configuration stage

Power BI Report Server is an on-premises report server that integrates with Power BI Desktop to create an easy to use in-house reporting platform. In addition to hosting reports with Power BI Report Server, you can use the reporting tools from SSRS to track reporting data. You use Power BI Report Server to do everything that SSRS does—and more. You can migrate your SSRS instance to Power BI Report Server to take advantage of this.

Power BI Report Server provides a web portal through which users access your Power BI reports (.pbix), paginated reports (.rdl), and mobile reports. Because users often want to view reports on demand, Power BI Report Server enables them to access their reports in a web browser, on a mobile device, or directly from an email.

Installing Power BI Report Server

Power BI Report Server is supported only on x64 processors. It uses an instance of SQL Server to host the report server databases, although no checks are made for this during the installation process.

Configuring Power BI Report Server

The final stage of the installation wizard enables you to start the report server configuration process using SQL Server Reporting Services Configuration Manager. During configuration, you need to specify the instance of SQL Server on which to create the report server database. If this is a remote instance, you will need to perform additional steps to enable remote access to this database.

Licensing Power BI Report Server

The report server is available with SQL Server Enterprise Edition with Software Assurance or as an extension to Power BI Premium. When using the Power BI Premium license, you create deployments spanning on-premises and hosted cloud.

Demonstration: Publishing a report to the Power BI Service

In this demonstration, you will see how to:

- Publish a report to the Power BI Service.
- Create a dashboard.

Check Your Knowledge

Question	
Which of the following is not an Excel power tool?	
Select the correct answer.	
	Power Chart
	Get & Transform
	Power Pack
	Power Pivot
	Power View

Lab: Exploring an Enterprise BI solution

Scenario

Adventure Works employees are increasingly frustrated by the time it takes to implement managed BI services. The existing managed BI infrastructure, including a data warehouse, enterprise data models, and reports and dashboards, are valued sources of decision-making information. However, users increasingly want to explore relationships with other, currently unmanaged data—and it takes too long for the IT department to incorporate these requirements into the corporate BI solution.

As a BI professional, you have been asked to explore ways in which Adventure Works can empower business users to augment their managed enterprise BI solution with self-service BI.

Objectives

After completing this lab, you will be able to:

- View reports in SharePoint Server.
- Create a Power BI report.
- Create a Power BI dashboard.

Note: Because of updates to Microsoft Power BI, the lab steps for this course change frequently. Microsoft Learning regularly updates the lab steps, so they are not available in this manual – but you can access them on GitHub.

Lab Setup

Estimated Time: 60 minutes

Virtual machine: **20778C-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa55w.rd**

All the lab steps are contained in 20778C_LAB_01.md.

Exercise 1: Viewing reports

Scenario

You have been asked to compare Excel Services in SharePoint with Power BI Desktop and Power BI Service to see which offers the best self-service BI solution. You will share an Excel file on SharePoint to determine how user friendly this experience is.

The main tasks for this exercise are as follows:

1. Prepare the lab environment
2. View reports in SharePoint Server

▶ Task 1: Prepare the lab environment

▶ Task 2: View reports in SharePoint Server

Results: At the end of this exercise, the Adventure Works Sales workbook will be published on SharePoint.

Exercise 2: Creating a Power BI report

Scenario

You have published an Excel workbook to SharePoint, and you next need to see how this compares to Power BI. You will create a report and add data, and then add visualizations to the report.

The main tasks for this exercise are as follows:

1. Import data into Power BI Desktop
2. Add visualizations to the report

▶ **Task 1: Import data into Power BI Desktop**

▶ **Task 2: Add visualizations to the report**

Results: At the end of this exercise, you will have a new Power BI report.

Exercise 3: Creating a Power BI dashboard

Scenario

Your Power BI report is ready to be published to the Power BI Service. Next, you will publish the report and create a dashboard, and then use the Natural Query Language to ask questions of your data.

The main tasks for this exercise are as follows:

1. Create a Power BI dashboard
2. Ask questions of your data

▶ **Task 1: Create a Power BI dashboard**

▶ **Task 2: Ask questions of your data**

Results: At the end of this exercise, you will have published a report to create a dashboard.

Question: Discuss using Power BI Desktop and Power BI Service, compared to using Excel and Excel Services in SharePoint. Which do you think is the best, and why?

Question: Has your organization started using Power BI? If not, how easy do you think it will be to implement, and convert existing business users from Excel, or other BI solutions? If you have already started using it, how do users find the experience compared to the previous solution?

Module Review and Takeaways

In this module, you have learnt about the basics of BI and data analysis. You have considered the emergence of self-service BI and looked at the tools available for creating self-service BI solutions.

Module 2

Introducing Power BI

Contents:

Module Overview

Self-service Business Intelligence (BI) has rapidly grown in popularity because of its ability to empower users to generate reports, process data, perform analysis, and more—all without having to depend on a report developer. The self-service BI trend is driven by Microsoft's commitment to improving Excel® and Power BI, both products having seen many enhancements in recent years. However, despite Microsoft enabling deeper data analysis with the four tools added to Excel—Power Pivot, Power View, Power Query, and Power Map—they are not fully integrated into the Excel interface. Instead, they exist in separate windows. Add to this the complexity of publishing to SharePoint® to share reports with colleagues, and it all becomes a time-consuming effort.

Using Power BI eliminates complications and barriers with a simple integrated user interface, and has the ability to publish rapidly to either a cloud-based or an on-premises portal to share reports easily. This module introduces Power BI, and explores the features that enable the rapid creation and publication of sophisticated data visualizations.

Objectives

After completing this module, you will be able to:

- Develop reports using the Power BI Desktop app and use report items to create dashboards on the Power BI portal.
- Understand the components of the Power BI Service, including licensing and tenant management.

Lesson 1
Power BI

In this lesson, you'll learn about the main features of Power BI that will help you create and publish reports.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the features and architecture of Power BI.
- Understand the main functionality of the PowerBI.com portal.
- Understand the functionality of Power BI Report Server.
- Download and use Power BI Desktop.
- Create reports using Power BI Desktop.
- Use report items on the Power BI portal to create dashboards.

What is Power BI?

- The Power BI suite comprises Power BI Desktop, Power BI Service, Power BI Report Server, and Power BI Mobile. Use it to:
  - Quickly create reports using the app or portal
  - Import data from files, on-premises databases, and SaaS providers
  - Combine multiple data sources into one report
  - Publish reports in the cloud or on-premises
  - Create dashboards on the portal from report items
  - Share dashboards with colleagues
  - View reports and dashboards with the Power BI Mobile app for iOS and Android devices
  - Gain insights in your data with Power BI Q&A

Microsoft has demonstrated a commitment to its Power BI suite of tools by producing a monthly software release of fixes and new features. As a data visualization tool, Power BI Desktop is quickly maturing and, with its cohesive user interface and ability to integrate with Office 365®, it is an obvious choice for the rapid creation of reports.

Power BI comprises the Power BI Desktop app, the Power BI Service, Power BI Report Server, and Power BI Mobile. You import data and create reports using the desktop app, transforming your data into rich, interactive visualizations. Using Power BI Desktop, you can connect to a wide range of data sources, and combine data from multiple sources within one report. You connect to, but are not limited to, Microsoft SQL Server, Microsoft Azure SQL Database, Excel, Oracle, and MySQL. You can also connect to software as a service (SaaS) providers, such as Facebook, Salesforce, MailChimp, and Google Analytics.

You can then publish your reports and datasets to the Power BI Service portal or Power BI Report Server to create and share dashboards with your colleagues. You do not have to use the desktop app to create reports; you can also sign in to the portal, import data, and create reports online. The report items are then used in dashboards. You can also view and interact with reports and dashboards using the Power BI Mobile app for iOS and Android mobile devices.

You can use the natural query language to ask questions from your data through Power BI Q&A. This interactive service quickly finds the answers within your data.

PowerBI.com

The PowerBI.com web portal, part of the Power BI Service, is the cloud location for publishing your reports, creating dashboards, and sharing data with others in your organization. Microsoft is applying weekly updates to the portal, continuously enhancing the service. Furthermore, if you require functionality that is not on the portal, you use the feedback facility to request a new feature and send ideas, rate the service, and vote on which features you think Microsoft should add next.

- The Power BI Service is a web portal for sharing reports, data, and dashboards
- My Workspace comprises:
  - Dashboards
  - Reports
  - Workbooks
  - Datasets
- App workspaces:
  - Package contents into an app
  - Share app with other users
  - More granular security than traditional workspaces

My Workspace

When you sign in to the portal, you have a personal workspace, called **My Workspace**. This workspace is comprised of the following areas:

- **Dashboards**. You create dashboards from your reports by pinning report items such as bar or pie charts, to new or existing dashboards.
- **Reports**. All the reports you have published from Power BI Desktop are listed alphabetically in this section.
- **Workbooks**. You import Excel workbooks into your workspace to access the data and reports they contain.
- **Datasets**. When you add a dataset to a report and publish it, the datasets used in the report are published to the portal, and listed alphabetically. You use these datasets to create new reports while signed in to the portal.

App workspaces

The app workspaces area enables you to share dashboards and reports with colleagues. You add dashboards and reports to an app and then publish that app to other people in your organization.

The original group workspaces implementation uses Office 365 groups, but the updated app workspace technology enables you to work with security groups, distribution lists, and individuals, in addition to Office 365 groups. You add users to admin, member, or contributor roles in a new workspace to control the access and capabilities they are allowed. This increased granularity provides a more flexible approach to securing your data.

Power BI Report Server

Power BI Report Server is the on-premises location for publishing your reports, creating dashboards and apps, and sharing data with others in your organization. It's designed for those organizations that have strict data protection policies that require them to keep all their data in house.

- On-premises enterprise reporting solution
- Combines the functionality of SSRS with Power BI
- Licensed with:
  - Power BI Premium
  - SQL Server Enterprise Edition with Software Assurance
- Download from Microsoft Download Center
- Configure using Report Server Configuration Manager

There are two ways to purchase Power BI Report Server:

- Purchasing Power BI Premium provides you with a Power BI Report Server license for on-premises reporting, in addition to capacity for cloud-based reporting.
- Purchasing SQL Server Enterprise Edition with Software Assurance enables you to use your Enterprise licenses to run Power BI Report Server for on-premises reporting.

To use Power BI Report Server, you need to install and configure the product, and then install the version of Power BI Desktop optimized for the version of Power BI Report Server that you are running. You download Power BI Report Server and the matching version of Power BI Desktop from the Microsoft Download Center.

Microsoft Download Center

https://aka.ms/Xa0153

On the Download page, in the search box, type **Microsoft Power BI Desktop (Optimized for Power BI Report Server)**, and then click **Search**. On the results page, click the link for the newest version that includes downloads for both Power BI Report Server and Microsoft Power BI Desktop (optimized for Power BI Report Server).

After running the installation program for Power BI Report Server, you use the Report Server Configuration Manager to configure the server. At this stage, you need access to an instance of SQL Server to host the report server database. After configuration is complete, you run the installation program for Power BI Desktop, and then start using the products.

Note: At the time of writing, you can only connect to SQL Server Analysis Services or Azure Analysis Services data sources when you publish to Power BI Report Server.

The Power BI Desktop (optimized for Power BI Report Server) user interface is very similar to Power BI Desktop. You use it to create reports that users then access in a browser or in a Power BI mobile app.

When you save a report, you save it directly to the Power BI Report Server—if you later need to make changes to the report, you open it directly from the server and update it there. You also upload reports from this version of Power BI Desktop to Power BI Service.

Power BI Desktop

- Power BI Desktop:
  - Combines Microsoft's Power Query engine, with data modeling, and visualizations
  - Free, stand-alone application for creating reports
  - Download from Microsoft website or Power BI portal
  - Report files can be saved with a .pbix extension
- Create reports using three-step process:
  1. Connect to data sources
  2. Shape the data to create the model
  3. Create reports to share with colleagues
- Workspace views: Report, Data, Model

Power BI Desktop combines the Microsoft Power Query engine, also known as M, with data modeling and visualizations, to provide data analysts with a flexible tool for quickly creating interactive reports.

Power BI Desktop is a stand-alone Windows® app, which you download from the Microsoft website, or from the Power BI portal. The Power BI Desktop app can be downloaded free of charge. You use this powerful tool to connect to a plethora of data sources, to create datasets and reports that can be shared. You save report files in the Power BI Desktop format, with a **.pbix** extension. Although you can save reports locally, or to a file share location, a trusted way to share data is by publishing reports and datasets to the Power BI portal.

There is a straightforward three-step process to creating reports:

1. Connect to your data sources.
2. Shape the data by using queries to create the data model.
3. Create reports that can be shared with, and enhanced by, others.

Download Power BI Desktop

http://aka.ms/C0fbvk

The Power BI Desktop features a workspace for creating reports. It comprises three key views in which you work:

1. **Report view**. This is your main workspace for adding report items, such as bar charts, maps, and pie charts, and displaying data using these report items.
2. **Data view**. You use the data view to see imported datasets, in addition to shaping the data using transformations and M expressions.
3. **Model view**. Power BI autodetects relationships from structured data sources, such as SQL Server or Microsoft Access®. Autodetection might not work for flat files, but after you have imported data, you can create relationships, and set the cardinality and cross-filter properties of the relationships.

Signing in to Power BI

When you first launch Power BI Desktop, the start screen gives you the option to sign in to your Power BI account. If you choose not to sign in at this point, you can optionally sign in later using the **Sign in** link in the top right corner of the screen. You can also use this link to switch accounts when signed in. To sign out, on the **File** menu, click **Sign out**.

Reports

You create multipage reports using Power BI Desktop or the PowerBI.com portal, but the Power BI Desktop app is likely to be your main tool for designing reports. The first step in creating a report is to connect to your data. Power BI Desktop supports a wide range of database, file, and SaaS connections and, along with the monthly software updates, new compatible data sources are continuously added. Data is imported into datasets, which can be transformed before using in visualizations.

- Create multipage reports in Power BI Desktop
  - Load datasets into a report, or use DirectQuery to query the data source and always return latest data
  - DirectQuery is useful for large datasets with long load time
  - Report view:
    - Add visualizations and additional report pages
    - Publish reports to the portal
  - Data view:
    - Shape data using transformations and Power Query Editor tools
  - Model view:
    - Manage relationships between datasets
    - Relationship autodetection runs by default
  - Use templates to share and reuse shaped data and formatted visuals

You can choose to load the data into the report—either refresh manually or on a schedule—or you can use DirectQuery, which does not import any data. After you import data, it is used as you create and customize your visualizations. If you use DirectQuery, the tables and columns are visible in the **Fields** list; as you work with the fields, Power BI queries the data source so that you always see the latest data. If you choose DirectQuery, remember that each time the data is queried the performance is dependent on the data source system, and how fast the data source system responds to the data request. DirectQuery is useful if you have very large datasets, and want to create your visualizations without loading large volumes of data. However, DirectQuery is not without its limitations, so you should shape data before you create your dataset. Note that you can only use tables from a single data source.

The Report view

After opening the Power BI Desktop app, you see the Report view. This workspace is initially blank, unless you have clicked a **.pbix** file to open the app. The **External data** ribbon menu is your main starting point for adding data. You click **Get Data** to choose a new data source, or **Recent Sources** to connect to data sources that you previously created. This includes data sources used in previous reports, as Power BI retains a list for future reference.

You add pages to your report from the **Insert** ribbon menu, which gives you the option of **Blank Page**, or **Duplicate Page**. Report pages are added and deleted using the tab at the bottom of each page. You can also right-click a tab to duplicate or rename the report page. After you add a dataset, the **Calculations** menu is activated, and enables you to create measures and add columns. The **Publish** option on the **File** menu prompts you to sign in to your Power BI account, so you can upload reports to the portal.

The Data view

You use the Data view to perform transformation operations on your imported datasets, so you can shape data appropriately for the reports you are producing. Click a dataset to view the imported rows and see the data you want to work with. You right-click any column to refresh the data, set the sort order of the data to either ascending or descending, rename a column, add or delete a column, and add a new measure. For more sophisticated transformation tools, right-click any column and click **Edit query**, which will open the **Power Query Editor** window. This opens the **Power Query Editor** window where you can split columns, apply statistical functions, pivot and unpivot columns, and more. From the **View** menu, you can open the **Advanced Editor** which displays a code view of the query.

You can also transform your data before you import it. Connect to your data source and, after you select the data you want to import, choose **Edit** rather than **Load**. This opens the **Power Query Editor** window where you shape your data.

The Model view

Power BI Desktop autodetects relationships in your data when the data is structured in a format in which the relationships can be adequately established. The Model view enables you to manage and create relationships. You can set the cardinality to **Many to one (*:1)**, **One to one (1:1)**, **One to many (1:*)**, or **Many to Many (*:*)**. The cross filter can be switched between **Both** or **Single**. You can also delete relationships.

Creating report templates

After creating a report, you can optionally save it as a template. Templates are useful for reusing data that has already been shaped, and visuals that have been customized using corporate colors. If you are producing several reports that share data, visuals, and formatting, templates are a useful feature for avoiding the duplication of work, while ensuring consistency across reports.

You edit an existing template and resave the file as a **.pbit** template, or edit and save as a standard **.pbix** report file. To create a template file, design the report you want to use as the basis for the template, then on the **File** menu, click **Save As**, and then select **Save as type: Power BI template files (*.pbit)**. Alternatively on the **File** menu, click **Export**, and then click **Power BI template**. To open an existing template, on the **File** menu, click **Import**, and then **Power BI template**, or on the **File** menu**,** click **Open**, and then go to the location of the template file, selecting **Power BI template files (*.pbit)** from the file type list.

Dashboards

After you have created the reports, you publish them to the PowerBI.com portal so they can be used to create dashboards. By sharing your reports with colleagues, you enable them to create their own dashboards and data insights. To publish a report, open the report in Power BI Desktop, and click **Publish**. You might be prompted to sign in to Power BI. After your credentials are confirmed, the report is published. If the report already exists on the portal, you are prompted to confirm the overwriting of any existing datasets that have changed.

- Power BI dashboards are created by pinning visuals to a new or existing dashboard
- Pin Live Page creates a dashboard tile from a report page, including all items in the report
- Pin from one dashboard to another dashboard for easy duplication
- Dashboard sharing with other users for a read-only view
- Full Screen Mode displays the dashboard without menus or browser—ideal for presentations or TVs
- Last Refresh Time can be enabled for each tile
- Favorite dashboards for most frequently used
- Featured dashboard to return to most used first

Creating dashboards

A Power BI user can create personalized dashboards using the reports and data that are available. Dashboards are an easy and effective method for combining data from disparate sources and reports. Any chart or item (known as a visual) from one or more reports can be intermixed on a dashboard. With this flexibility, users build profiles of data and search for trends or answers to questions. Dashboards are created by pinning visuals to a new or existing dashboard. These visuals are created as tiles on the dashboard.

Pin Live Page

You can pin a complete report page to a dashboard as a single tile item. A page can be pinned on its own, or combined with other tile items. Changes to the report appear in the dashboard whenever the page is refreshed. To pin a report, click the one you want to pin, and then, on the horizontal menu bar at the top of the webpage, click **Pin Live Page**. This provides the option to add the page to an existing dashboard, or create a new one.

Pin from dashboard

You can pin a tile from one dashboard straight onto another. Click **More options** (the ellipsis) on a tile, and then click **Pin tile**. This opens the **Pin to dashboard** window with the option to pin to an existing dashboard, or create a new one. This works in the same way as pinning a report visual to a dashboard.

Dashboard sharing

You can share, or unshare, a dashboard with other users in a group. After a colleague accepts an invitation to share a dashboard, it appears in their **My Workspace** menu, along with the reports associated with the dashboard. The dashboard is read-only for the recipient of the shared invitation. To share a dashboard from Power BI, right-click the name of the dashboard in the **My Workspace** menu, and then click **SHARE** to open the **Share dashboard** window. You then enter one or more email recipients, along with a message to describe the dashboard.

Focus mode

To view a tile in greater detail, use the focus mode feature. In the top right corner of the tile you want to view, click **More options**, and then click **Open in focus mode**. This expands the tile so it is the only tile in view. You then begin filtering and drilling through your data as appropriate. Click **Exit Focus mode** to return to the dashboard.

This works equally within reports. Click **Focus mode** to expand the visual, and then click **Back to report** to return to the main report view. When used in your dashboards, this differs from the **Full Screen Mode**, as the focus mode retains menus and controls to enable you to filter the data.

Full Screen Mode

Power BI dashboards can be displayed in full screen mode, which is ideal for presentations, or TV screens. The browser and Power BI menu are hidden from view, and the dashboard expands to fill the screen. By moving the mouse over tile elements, text pop-ups continue to show. To enter the presentation mode, click **Enter Full Screen Mode** from the dashboard toolbar in Power BI. Press **Esc** or click **Exit Full Screen Mode** to return to Power BI.

The **Fit to Width** and **Fit to Screen** buttons in full screen mode improve a dashboard that does not have enough tiles to fill the full screen, and has excess background space. For example, if there are only a few small charts on a dashboard, the buttons zoom in to enlarge the items and fill as much width or space as possible—this makes the charts easier to read and improves the presentation of the dashboard.

Last Refresh Time

Items that you add to a dashboard can now display the last updated date and time. This is useful for checking when data was last loaded, and ensuring users have the most up-to-date figures. The Last Refresh Time, which is visible in **Full Screen Mode**, is enabled at an individual tile level by using the **Tile details** menu.

Favorite dashboards

You make a dashboard a favorite so you can access it from anywhere within the Power BI Service. To do this, select a dashboard from the navigation pane on the left. When the dashboard loads, on the toolbar, click **Favorite**. Click **Favorites** on the navigation pane to see all your favorite dashboards. To remove a dashboard from your favorites, hover your mouse over the dashboard tile to bring up the icons, and click **Unfavorite**. Alternatively, open the dashboard from the navigation menu to view it, and then on the toolbar, click **Unfavorite**.

Featured dashboard

A featured dashboard is like a favorite dashboard, but is given the status of being the first dashboard you see when you log in. You can also view it immediately by clicking **Featured Dashboard** on the navigation pane. To make a dashboard the featured dashboard, select it from the **My Workspace** pane to open the dashboard, and then click **Set as featured**. You can now view this dashboard from the navigation pane.

Designing reports and dashboards

One of the most attractive features of Power BI is the stunning visualizations you use to create reports and dashboards. Along with these visuals, you can apply various techniques to make your reports and dashboards easier to consume. In addition to looking great, important information is conveyed quickly.

- Use techniques to design your reports and dashboards to make them easy to digest:
  - **Customize visuals**: use colors, labels, borders, and titles to enhance and group related visuals together
  - **Positioning**: most import information at top of report or dashboard, especially for viewing on mobile device
  - **Audience**: what metrics are most useful to end users? Think about displaying on TV screen or large monitor
  - **Storytelling**: only show relevant and related data, use multipage reports to break up data into context/subject
  - **Choosing a formatting a visual**: bigger visuals for more important data; try out charts to find best for scenarios

Customize visuals

You can fully customize each visual, with colors, labels, borders, and titles. You change colors so they match corporate colors or, when creating reports for departments, colors could be used to distinguish departments—for example, blue for finance, yellow for sales, red for marketing. Labels and titles enhance the descriptive text given to a visual—you can also include a text box next to a visual to add a lengthier description where appropriate. Visuals that are related, or work with a slicer, can be grouped together using shapes. Rectangle and line shapes help to contextually group or partition visuals.

Positioning

Published reports might not be editable to the end user, so users cannot move the visuals around on the canvas if they are placed inappropriately. When a user creates a dashboard, they can move pinned tiles around on the canvas. When creating a report or dashboard, the most important information should be presented first, in the top left corner of the screen. This is particularly important when designing for mobile devices; a user will not be able to move pinned items, so it's vital to have the most important visual at the top—so it is visible first on a small mobile phone screen.

Audience

Think about the person who will consume your reports and dashboards. What metrics are important to them? Are specific key performance indicators (KPIs) needed for them to measure the effectiveness of their department, or their role? A salesperson might want to see how close they are to their sales target, whereas the sales director will want to see how each salesperson is progressing.

You might also consider where the dashboard will be viewed. If it is to be displayed on a TV or large monitor, then you include more content than you would for displaying on a mobile phone. Furthermore, displaying your dashboard in full screen mode removes menu bars and other distractions.

Storytelling

Reports and dashboards should not be cluttered, and show only relevant and related data. When creating reports, use multiple pages to group related visuals by department or subject. Rename the tabs at the bottom of the screen to help users quickly find data. Try to avoid having so many visuals on a report or dashboard that make the user scroll across or down.

Choosing and formatting a visual

The most important information should not only be displayed first, but also have the biggest visual suitable for presenting it. You should size visuals so that important information is displayed in bigger visuals, and less important information in smaller visuals. This guides the user to interpret and digest the report or dashboard more efficiently. A card at the top of the screen showing sales means executives can immediately see organizational performance. Consider the following design principles when choosing a visual to represent your data:

- Look at both the fields and the data values you want to present in a chart. If the data includes geographic fields, a map chart might be best. However, if values need to be displayed proportionately, then perhaps a pie or tree map would be appropriate. Would a constant or reference line add value to a chart? If you are unsure, try a few chart types, then you'll see how the data is presented and what works best.
- Your charts should be consistent, both in terms of design and axes. Ensure scales on axes and the order of dimensions are consistent, and be aware of how you use colors.
- When displaying numbers, avoid using too many numerals, as this makes it difficult to read. Rather than displaying a card with $145,000,000, present the data as $145m or $145 million, because this is quicker and easier for the mind to interpret.
- Charts that present data over time should also be consistent, especially if you apply filtering. For example, don't have one chart that displays data for the last quarter next to a chart showing data for April last year.
- In addition to avoiding showing different time precision, apply the same principle to measures, and avoid mixing measures of different scales. Showing one scale in millions and another in thousands makes them difficult to compare.
- How you sort charts can make a difference to how well the data is interpreted. If you want the user's attention to be drawn to the highest or lowest number, sort the chart by that measure.
- Avoid using pie charts when you have many categories. When the number exceeds about seven or eight categories, choose another visual such as a bar or column chart. If there are too many, this makes it difficult to compare in a pie chart.

Demonstration: Creating a report with Power BI Desktop

In this demonstration, you will see how to:

- Create a new report in Power BI Desktop.
- Connect to the AdventureWorksLT Azure SQL Database.
- Add a chart to the report using AdventureWorksLT data.

Check Your Knowledge

Question	
Which of the following statements is false?	
Select the correct answer.	
	You can import data and create reports with the Power BI Desktop app.
	You can create and share dashboards on the PowerBI.com online portal.
	You can use SQL Server Standard Edition licenses to run Power BI Report Server for on-premises reporting.
	Data can be imported from an on-premises SQL Server or Azure SQL Database.
	Data can be imported from Facebook.

Lesson 2
The Power BI Service

In this lesson, you will learn about the licensing structure of Power BI, and explore the many options available when creating datasets. You will also be introduced to apps, learn how the natural query language answers questions about your data, and understand tenant management.

Lesson Objectives

At the end of this lesson, you will be able to:

- Explain the different Power BI licensing options.
- Understand tenant management.
- Describe how to incorporate datasets into Power BI reports.
- Explain how to create and use apps.
- Describe the benefits of the natural query language.

Licensing

Power BI offers a straightforward licensing model, with a choice of per-user or capacity-based licenses.

- Per-user licensing:
  - Choice of Power BI Pro or free licenses
  - Only Power BI Pro accounts can publish shared content
- Capacity licensing:
  - Power BI Premium licenses
  - Dedicated capacity to improve performance
  - Enable free users to access shared content

Per-user licenses are Power BI Pro licenses and free licenses. Both of these can do the following:

- Connect to 70-plus data sources.
- Create reports and datasets by using Power BI Desktop.
- Publish content to the web.
- Export content to PowerPoint, Excel, and CSV files.

Only Power BI Pro users can publish to, and consume content from, app workspaces, share dashboards, and subscribe to content. The free license only allows users to consume shared content that is published to Power BI Premium capacity. A free account requires a work or school email address, so personal domains such as Gmail, Hotmail, or Yahoo, are not permissible.

Capacity-based licenses are Power BI Premium licenses. By purchasing these licenses, organizations have access to their own dedicated capacity to improve performance and support higher volumes of data than shared capacity allows. Both Power BI Pro and free users can access content published to Power BI Premium capacity.

Organizations can have a mix of free and Power BI Pro accounts and use Power BI Premium capacity. For full service details and local pricing, see *Power BI pricing* in the Power BI documentation:

Power BI pricing

http://aka.ms/Qz9yz8

Tenant management

- Power BI uses a self-service sign-up model:
  - Users can sign up without dependency on an Office 365 account, or organizational Office 365 administrator
  - When a user signs up, a tenant is created for the domain, or the user joins the tenant—for example, contoso.com
  - Users within a tenant can collaborate and share content
- Office 365 admins sign up using the Power BI portal or Office 365 Admin portal:
  - Users can be assigned a license, or sign up and join the tenant and acquire a license
  - Qualifying organizations receive 1 million licenses, and can request more from Microsoft

Power BI uses a self-service sign-up model so that users can create an account without dependency upon either an Office 365 administrator, or an Office 365 subscription. When an individual from an organization signs up to Power BI, a tenant is created automatically. A tenant is a domain within your organization; for example, contoso.com. If another user from the same organization signs up, that user is added to the existing tenant. All users within the same tenant become part of the same network; this means they share reports, dashboards, and datasets. In this situation, the agreement is between Microsoft and the user, so no organization intervention or responsibility is required. Users can also reset their password directly from Microsoft, using an email verification process.

Administrator sign-up

Administrators sign up to Power BI via the PowerBI.com website, or through the **Purchase Services** section within the **Office 365 Admin Portal**. Administrators then assign licenses to users within the tenant. In addition, users can still sign up individually, and be automatically assigned an available Power BI license. If the user does not already have an Office 365 account, an account is also created for them.

For more information on managing tenants, including the prevention of users joining a tenant, see *Power BI licensing in your organization* in the Power BI Documentation:

Power BI licensing in your organization

https://aka.ms/M2nhu6

Organizations without Office 365

If your organization does not have an Office 365 environment, users can still create accounts, but the organization will not be able to collectively administer the accounts. The Power BI Service is built on the Microsoft Azure platform, so these accounts are created in a cloud-only user directory—that your organization can claim—to manage the tenant and users.

Acquiring Power BI licenses for an Office 365 tenant

Qualifying organizations with an Office 365 tenant receive 1 million licenses. Licenses are provided free of charge for using the Power BI free service. If your organization requires more than 1 million licenses, you should contact Microsoft. When a user within the organizational domain signs up for Power BI, they are assigned one of these available licenses. Administrators can also assign licenses through the portal.

For more information on the Power BI architecture and Power BI security, see *Power BI Security* in the Power BI documentation:

Power BI security

http://aka.ms/Bk38nc

Datasets

A dataset is created when you import data into Power BI Desktop, or through the Power BI portal. The dataset can be used across multiple reports; you can shape and combine the data in your datasets.

- Create a dataset by importing data into Power BI Desktop or the PowerBI.com portal
  - Import data from data sources including on-premises or cloud databases, files, SaaS connectors
  - Scrape data from a webpage into Power BI tables
  - Copy and paste data from Excel into a Power BI table
- Load data into Power BI, or transform it first
  - Datasets in the Data view and Power Query Editor
  - Power Query Editor offers transformations such as column splits, rounding, aggregations, statistical operations
- Refresh datasets in Power BI Desktop and portal
  - Schedule the refresh of datasets on the portal

Data sources

In Power BI Desktop, you have a wide choice of sources to import from, including database, file, and SaaS connections, as described here.

Database connectors

Power BI supports the main industry database and file connections for importing data from on-premises sources. Database connectors include:

- SQL Server
- SSAS tabular and multidimensional models
- Microsoft Access
- Oracle
- IBM Db2
- MySQL
- SAP HANA
- PostgreSQL
- Sybase
- Teradata

File connectors

You can import from a single file, or choose a folder to select multiple files to import. This is useful when you have a folder location used to store files created on a schedule. File connectors include:

- Microsoft Excel
- CSV
- XML
- Text
- JSON

SaaS connectors

An increasing number of connectors to Power BI Desktop make it easy to connect to external SaaS applications for analyzing data such as web traffic, sales, marketing, financial, and social media. SaaS connectors include popular services such as:

- Google Analytics
- QuickBooks Online
- MailChimp
- Facebook

- Dynamics 365 for Customer Insights
- Salesforce
- GitHub

Users connect to SaaS applications and import the data to create reports and dashboards. Due to its flexibility, Power BI combines multiple sources of data from disparate SaaS vendors into one central reporting space. For example, figures from Salesforce can be combined with a recent marketing campaign that was delivered using MailChimp, alongside marketing data from Facebook.

Other data sources

You can also connect to any webpage to scrape the data into tables within the dataset. You might not be able to determine the table names or structure of the data, but you can perform some operations to rename fields and tables after you have imported the data into Power BI Desktop.

You can quickly create a table by copying and pasting data directly from an Excel or text file. On the **Home** ribbon, click **Enter Data** to open the **Create Table** window. Right-click in the table and then click **Paste** to paste data from another file. You work with the contents of this table within your dataset, just as you would with data from any other source.

Working with datasets

You import data by connecting to a data source, such as SQL Server, or Excel. To begin, choose **Get Data** or **Recent Sources** from the **Home** ribbon, and then select your data source from the list. The **Navigator** window shows all the tables, views, or worksheets you can import. You preview and select the data you want to import. From here, you click **Load** to pull in the data as it is, or click **Edit** to make transformations. If you choose to edit the data, it opens in the Power Query Editor window, so you have access to the full range of transformations. This is a useful step if you have a large dataset, but want to reduce the amount of data that you import by excluding columns or filtering rows. Conversely, if you choose to load the data, all columns and rows are imported before you can apply transformations.

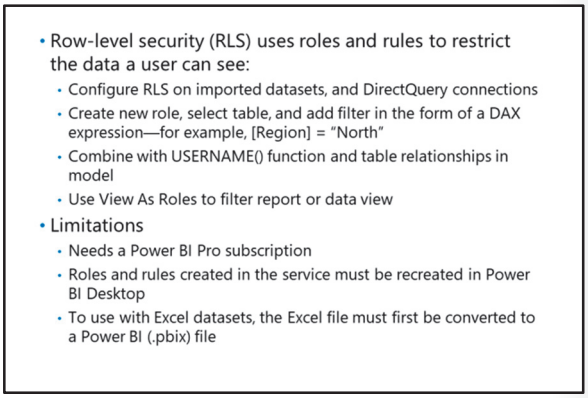
You can perform some basic operations on your datasets in the Report view. In the Fields pane, you can add or delete columns, rename the table and columns, refresh the data, and create a new measure. However, most of the work you perform on your datasets will be in the Data view window or the Power Query Editor. The Power Query Editor offers more complex transformations than the Data view—such as column splits, rounding, aggregations, statistical and scientific operations.

Refreshing data

When you publish a report to the Power BI portal, the datasets are published too. You can use the Power BI portal to refresh the data within your datasets. In the left pane, click the ellipsis next to a dataset to open the dataset menu. You can choose to **Refresh Now** or **Schedule Refresh**. If you want to schedule a data refresh, you should follow the instructions for downloading the **Power BI Gateway**.

You can also refresh your data in Power BI Desktop by clicking **Refresh** on the **Home** ribbon. When viewing a dataset in the Power Query Editor window, you might see a message such as "This preview may be up to 35 days old". Click the **Refresh** button to update the data, though the data might not have been altered. This applies to each table within your dataset, so you have control over exactly which tables to update.

Row-level security



Row-level security (RLS) enables you to restrict the data that a user can view, based on filters. These filters work at the row level to control what data is returned to the user, and can be managed using roles. In addition to configuring RLS on your data models in the Power BI Desktop, you can also configure datasets using DirectQuery. In Power BI Desktop, roles cannot be defined for SQL Server Analysis Services live connections; this must be done within the Analysis Services model. If you enable the option **Enable cross filtering in both directions** in Power BI Desktop, this applies the cross filter and the security in both directions.

To configure RLS, you start by defining roles and rules within Power BI Desktop, and then publish these to the Power BI Service:

1. Import data into Power BI Desktop, or configure a DirectQuery connection.
2. From the **Security** group on the **Modeling** ribbon, click **Manage Roles**.
3. In the **Manage roles** dialog box, click **Create**.
4. When the text box appears, type in a name for the role.
5. From the **Tables** list, select a table to apply the filter.
6. The filter will be a DAX rule that returns true or false; for example, [Region] = "South West". In the Table filter DAX expression box, type in the DAX expression, and then click the tick button to validate the expression.
   Alternatively, you right-click the table name and click Add filter. You then insert a column from the table into the DAX filter text box.
7. Repeat steps 5 and 6 to create further filters on other tables.
8. Click **Save**.

In Power BI Desktop, you cannot assign users to roles, because this is done in the Power BI Service. However, you can use the USERNAME() function alongside table relationships to dynamically restrict data to the current user. After creating a role, you can then test it:

1. From the **Security** group on the **Modeling** ribbon, select **View As Roles**.
2. Select one or more roles, and click **OK** to apply the filtering. Additionally, select **Other user**, and type in the name of another user for whom you want to test: for testing as it would appear in the Power BI Service, type the user principal name (UPN), such as jane@contoso.com.
3. You can use this in the Report or Data view to see the restricted results.
4. Click the **Stop viewing** button at the top of the view to remove the applied filters.

Limitations

There are limitations to using row-level security that you need to be aware of:

- To use RLS, you need to have a Power BI Pro subscription.
- Roles and rules created in the Power BI Service need to be recreated in Power BI Desktop.

- RLS can only be defined on datasets created in Power BI Desktop. If you want to use RLS with datasets created in Excel, you must first convert the Excel file to a Power BI (.pbix) file.
- Only imported data and DirectQuery connections are supported. Live connections to SSAS are handled in the on-premises data model.
- Cortana and Q&A do not support RLS.

Apps

- Power BI apps are packaged reports, dashboards and datasets
  - Can be shared with other Power BI users
  - Can be customized for different users
  - Created in app workspaces
  - Give access to specific groups, or entire organizations
  - Add title, description and image or company logo
  - Can automatically install
- Import apps from SaaS providers, such as MailChimp, Insightly, Marketo, and Twilio

Apps are packaged reports, dashboards, and datasets, which can be shared with other Power BI users in your organization. When you install an app on the PowerBI.com portal, the report items are merged into your workspace lists. Users with a free Power BI account can only view apps that are published to Power BI Premium capacity—they cannot create them. You can create apps to customize reports or dashboards for users in different departments within your organization. For example, you could create a set of reports with targeted visuals for finance, sales, and manufacturing, because each department is likely to want different data with which to measure performance.

You create apps in app workspaces in Power BI Desktop. Because changes that you make in the app workspace are not propagated to the app until you publish or update it, you can use the app workspace as a staging area when designing or revising your app.

When you publish an app, you choose who you want to give access to. You can choose specific groups, such as Sales, or Human Resources, or you can give access to your entire organization. You customize the app with a title, and a description to help users determine if it is applicable to their needs. You can also upload an image or company logo for the app and assign a background color to match your corporate or departmental identity. You choose the reports, dashboards, and datasets you want to include; however, when you choose a report or dashboard, it automatically includes any required datasets, and these cannot be excluded. The app is then available in your organization's content gallery. Users who have access to the app can create new dashboards from the contents.

In addition to choosing who can access the app, you can also automatically install it for end users. Provided you have been granted permission to do this, when you publish your app you can check the **Install app automatically** box and Power BI pushes the app to all the users or groups you have selected to access your app. Those end users will then find your app when they refresh their apps list without having to search for it. When using this functionality, remember that installing apps can take time and bandwidth, so it is best to do so out of office hours.

If you need to remove an app from users, in the app workspace, right-click the ellipsis, and then click **Unpublish app**. Users will no longer have access to the app and will lose any personal bookmarks, comments, or subscriptions associated with the content in the app. After unpublishing an app, the app workspace is still available for you to work in and you can republish the app later.

In addition to creating your own apps, you can also import apps from SaaS providers such as Adobe Analytics, Alpine Metrics Sales Prediction, Insightly, Marketo, and Twilio. To add an app from an SaaS provider that you have an account with, in PowerBI.com, click **Get Data** then under **Services**, click **Get**. In the **AppSource** window, under **Apps**, either click the SaaS logo to view provider details, or click **Get it**

now. You will be prompted to enter your customer details for the service. After authentication, you can import an app with reports and dashboards designed to visualize your data.

Natural language queries

Finding answers to questions can be difficult if your organization has many data sources, and users do not know which data to use. Also, if existing reports do not slice data the right way, or do not show up-to-date aggregations, or enough data, users cannot find the answers they need. This becomes particularly arduous when users frequently have questions that need an immediate answer—but it takes time for the report developer to create and publish the report. With Power BI, you use the Q&A feature to ask questions using a natural language, just as you would by using a search engine. With Q&A, anyone in the organization with access to Power BI can quickly find answers, because no additional programming skills are needed.

- Power BI Q&A helps you ask questions about your data using natural query language
  - Anyone who has access to the data in Power BI can ask a question and receive a quick response
  - Users asks questions, just as they would with a search engine
  - Q&A helps you phrase your question, uses auto-complete, restates questions, and corrects spelling
  - Terminology for names, date keywords, date ranges, aggregations, equality, sort order, and verbs
  - Searches are done with datasets used by the dashboard
  - Pin the answers to your dashboard for future reference
  - Answer can be presented in chosen chart type—for example, a map

Q&A box

The Q&A box sits at the top of the screen when viewing dashboards. When you click in the box, Q&A displays a prebuilt list of suggestions to help you get started. This list comprises the questions that were used to create the tiles that were pinned to the dashboard, in addition to the names of the tables in the datasets that were used to build the report. You select any of the suggestions from the list, or type in your own question. Q&A helps you phrase your question, using auto-complete, restating your questions, and using appropriate textual or visual aids. It also corrects spelling and dims the color of words it does not understand.

Terminology

Q&A automatically recognizes the following keywords and terms:

- **Names**. If a column in the dataset contains a phrase such as "name", for example FirstName, then Q&A knows the column values are names. You can phrase a question using the search for a particular name.
- **Tenses**. “Sell” and “sold” are treated the same.
- **Possessives**. “What is the total of Pamela’s sales”?
- **Plurals**. “Opportunity” and “opportunities” are treated the same.
- **Date keywords**. This month, last year.
- **Date ranges**. Before, after.
- **Aggregations**. Minimum, maximum, count of, average, less than, between, before.
- **Equality keywords**. Equal, more than, less than, between.
- **Sort order**. Ascending, descending, alphabetical.
- **Display verb**. Show, what is, are, what are.

How Q&A finds the answer

Q&A searches for the answer to your question using any of the datasets that have a tile on the dashboard on which you are asking the question. It returns the best answer it can from the available data. If you remove tiles from a dashboard, be aware that the underlying datasets are also removed, so you cannot use this data for your Q&A. This is particularly important if you pin the visualization answer to your dashboard.

Visualizing the answer

Power BI Q&A decides on the best visualization to present the answer. In addition to requesting the data you need, you can also ask for it be presented using a specific visualization, such as a chart or map. For example, you could ask "show sales by store as a map", or "show sales by territory as a tree map".

Demonstration: Creating an App

In this demonstration, you will see how to:

- Publish a report to the Power BI Service.
- Use the report to create a dashboard.
- Create an app using the dashboard and dataset.

Question: Discuss the benefits of using Power BI in an organization looking to create reports to analyze their data.

Lab: Creating a Power BI dashboard

Scenario

Adventure Works employees are increasingly frustrated by the time it takes to implement managed BI services. The existing managed BI infrastructure, including a data warehouse, enterprise data models, and reports and dashboards, are valued sources of decision-making information. However, users increasingly want to explore relationships with other, currently unmanaged data—and it takes too long for the IT department to incorporate these requirements into the corporate BI solution.

As a BI professional, you have been asked to explore ways in which Adventure Works can empower business users to augment their managed enterprise BI solution with self-service BI.

Objectives

After completing this lab, you will be able to:

- Connect to an on-premises SQL Server database from Power BI Desktop, create a new report, and publish it to the Power BI portal.
- Create a Power BI dashboard.

Note: Because of updates to Microsoft Power BI, the lab steps for this course change frequently. Microsoft Learning regularly updates the lab steps, so they are not available in this manual – but you can access them on GitHub.

Lab Setup

Estimated Time: 60 minutes

Virtual machine: **20778C-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa55w.rd**

All the lab steps are contained in 20778C_LAB_02.md.

Exercise 1: Connecting to Power BI data

Scenario

You are a business analyst for Adventure Works who will be creating reports in Power BI Desktop using the corporate database stored in SQL Server 2017. You have been given a set of business requirements for data and will now connect to the database from Power BI Desktop. You will publish your report to the Microsoft Power BI portal, and use the reports to create a dashboard.

The main tasks for this exercise are as follows:

1. Prepare the lab environment
2. Connect to SQL Server from the Power BI Desktop
3. Add charts to the report
4. Publish the report to the Power BI portal

- ▶ **Task 1: Prepare the lab environment**
- ▶ **Task 2: Connect to SQL Server from the Power BI Desktop**
- ▶ **Task 3: Add charts to the report**
- ▶ **Task 4: Publish the report to the Power BI portal**

Results: After this exercise, a report will be published on the Power BI portal.

Exercise 2: Create a Power BI dashboard

Scenario

You have created a management report showing Adventure Works sales data, and have published this to the Microsoft Power BI portal. Next, you will create a dashboard on the portal, so managers can use this to bring data together in one place.

The main tasks for this exercise are as follows:

1. Create a new dashboard
2. Add chart items to the dashboard
3. Customize the dashboard
4. Display the dashboard in Full Screen Mode

- ▶ **Task 1: Create a new dashboard**
- ▶ **Task 2: Add chart items to the dashboard**
- ▶ **Task 3: Customize the dashboard**
- ▶ **Task 4: Display the dashboard in Full Screen Mode**

Results: After this exercise, a dashboard will be created on the Power BI portal.

Question: Self-service BI empowers business users with the ability to use corporate data to compile reports without the dependency on an IT department, or a dedicated report developer. Giving users access to live data means they gain insights into the most up-to-date transactions. Real-time analysis is critical to organizations in certain industry sectors. While this is advantageous to the users, you must consider the security and performance of your on-premises databases. What tools can you use to ensure the safety and performance of your databases?

Question: Discuss the different SaaS providers that your organization uses, and how this data could be used in Power BI dashboards. How could this data be combined with data from production databases to create greater insights into data?

Module Review and Takeaways

Using Power BI eliminates complications and barriers with a simple integrated user interface, and has the ability to publish rapidly to a cloud-based portal to share reports easily. This module introduced Power BI and explored the features that enable the rapid creation and publication of sophisticated data visualizations.

Module 3

Power BI Data

Contents:

Module Overview

Power BI offers a straightforward approach to report creation, and the ability to create and share dashboards without dependency on a report developer, or the need for Microsoft® SharePoint®. Microsoft Excel® has long been the tool of choice for data analysts who work in a self-service style. However, Excel does not offer a quick and easy way to share reports without the use of either SharePoint, or the creation of multiple copies of spreadsheets that quickly become out of date, or exist outside source control.

In recent years, power tools have been added to Excel: Power View, Power Query (known as Get & Transform in Excel 2016), and Power Pivot. Power BI brings much of this power into an integrated environment in the form of Power BI Desktop. Previously, Excel users were inconvenienced by needing to transition between the different power tools, but Power BI Desktop brings the tools together. This means that Power BI is fast becoming an obvious choice for the analysis and sharing of data. However, analysts are likely to continue working with Excel for the near future. Power BI easily cooperates with Excel, and many other data sources. It's this ability to create reports rapidly, by using data from a combination of sources, which really puts the power into Power BI.

Objectives

After completing this module, you will be able to:

- Connect to Excel files and import data.
- Describe the data model and know how to optimize your data within the model.
- Use on-premises and cloud Microsoft SQL Server databases as data sources, along with the R script data connector.
- Take advantage of the features of the Power BI service by using Q&A to ask questions in natural query language and create apps.

Lesson 1
Using Excel as a data source for Power BI

In this lesson, you will learn how to connect to Excel from Power BI and import data. You will also learn how to update and refresh data.

Lesson Objectives

After completing this lesson, you will be able to:

- Connect to files from the Power BI service and Power BI Desktop.
- Import data from Excel.
- Publish data from Power BI to Excel.
- Update files in Power BI.
- Refresh Excel data in Power BI.

Connecting to files

In Power BI, you can connect to various file formats. In addition to Excel, you import data from comma-separated values (CSV), XML, text, or JavaScript Object Notation (JSON) files, or a folder that contains multiple files in one of these formats. You can also import a Power BI report file that has the .pbix extension. When you import data directly into the Power BI service, the maximum size for any file format is 250 megabytes (MB).

- Connect to files from Power BI desktop or service:
  - Compatible file types include Excel, CSV, XML, and JSON
- From the Power BI service:
  - Connect to a local file, OneDrive Business or OneDrive Personal, or SharePoint – Team Sites
  - Maximum file size to upload to the Power BI service is 250 MB
- From Power BI Desktop:
  - On the **External data** menu, click **Get Data** from, and then choose file location from your local computer, or OneDrive
  - Connect to a folder to import multiple files
- Folder locations can contain different file formats

You import files from your local computer, or connect to files on Microsoft OneDrive® Personal, OneDrive for Business, or SharePoint – Team Sites. Data that is imported from OneDrive or SharePoint into Power BI is automatically updated if the source file changes. For example, if additional rows are added to a table in a workbook, the changes are reflected in any reports and dashboards in Power BI, usually within an hour.

When importing CSV files, it's best to use a comma-delimited format, and include a header row. Fixed width CSV and text files are also supported. After selecting the file for import, the preview enables you to select the delimiter type, including comma, colon, semicolon, tab, fixed width, or a custom value.

Connecting to files from the Power BI Service

To connect to a file in the Power BI service, click **Get Data**. In the **Create new content** section, under **Files**, click **Get**. You then select from one of the following:

- **Local File**. Browse to a file that is stored on your local computer. Click **Open** to upload the data to Power BI.
- **OneDrive – Business** or **OneDrive – Personal**. Browse to the file that you want to upload, and then click **Connect**. Power BI creates a connection to the file; updates to the file are reflected automatically in Power BI.

- **SharePoint – Team Sites**. Click **SharePoint – Team Sites** to open the **Connection** dialog box. Either enter the URL of the SharePoint server and click **Connect**, or click **Connect** to view content that is available to you at the root level.

Connecting to files from Power BI Desktop

When you initially start Power BI Desktop, the splash screen gives you options to connect to data. If you have selected **Show this screen on startup**, you can click **Get data** to open the Get Data window. Alternatively, on the **External data** menu, click **Get Data**. This presents you with a list of the most common sources, or you can click **More** to open the Get Data window and view the full list of compatible data sources. The Get Data window breaks the data source connections into **All**, **File**, **Database**, **Power BI**, **Azure**, **Online Services**, and **Other**. Click **File** to view the list of compatible file formats, or select **Folder** to import a collection of files. When you select a file format, such as Excel, you can select a file from your local computer, or from a OneDrive location.

Note: When you are using a folder location to import multiple files, you can include different file formats in the folder. After selecting the folder location, Power BI displays a list of the files that are stored in the folder. This includes any incompatible formats such as .jpg or .docx. When you click **Load** to import the data, Power BI ignores the files that it recognizes as not being data files.

Importing Excel files

If Excel is widely used in your organization, you can combine reports that have been created in Excel with the visualizations and sharing capability of Power BI, without losing the effort that went into creating the Excel workbooks in the first place. There are two approaches to importing Excel files:

1. Connect to an Excel workbook (.xlsx) and use the contents as datasets for your Power BI reports and dashboards.
2. Import a whole Excel workbook and explore the whole file, in the same way that you would by using Excel Online.

- Two approaches to importing an Excel workbook:
  - Connect and use contents as datasets:
    - File size limitations
    - Data must be formatted as a table in Excel
    - Load the data and use Power Query Editor to apply transformations
  - Import an entire workbook, including Power Pivot and Power View, and explore as you would in Excel Online:
    - Import any .xlsx or .xlsm file to explore features
    - Power Pivot models are imported to created datasets
    - Power View content is imported as reports that can immediately be pinned to dashboards
    - Data source connections are imported for scheduled data refresh

Importing Excel content as a dataset

When you import Excel content as a dataset, the workbook can consist of a data model and the core worksheet contents, but the file that you choose to upload can be no larger than 250 MB. Within the 250 MB limit, the core worksheet can be up to 10 MB, with the remainder of space used by the data model. If your workbook meets these criteria, you save the file to OneDrive for Business and connect to it from Power BI, in addition to viewing it in Excel Online.

There are several ways to reduce the size of the core workbook in a file that you want to import. Images and clip art elements can increase the size of the file, so remove these if possible. Remove cell shading and sheet background colors to reduce the size further. If the report contains a data model, you can move data from the worksheet to the data model. Furthermore, ensure that you exclude columns that are not necessary to the analysis that you want to perform. If your data originated from a data warehouse, it might include metadata columns that were added during the extract, transform, and load (ETL) process, such as Last Run Date, or Create Date. Look out for the inclusion of these columns and remove them

where necessary. For more information about creating an efficient data model, see *Create a memory-efficient Data Model using Excel and the Power Pivot add-in* in the Office documentation:

Create a memory-efficient Data Model using Excel and the Power Pivot add-in

http://aka.ms/Ca9lsv

To import data from Excel into a Power BI dataset, the data must first be formatted as a table:

1. To convert columns of data into a table in Excel, first highlight the rows and columns that you want to include. Then, on the **Insert** menu, click **Table**.
2. After you have formatted your Excel workbook, return to Power BI, click **Get Data**, and then click **Excel**.
3. The navigator displays a list of worksheets and tables within the workbook. You select the worksheets and tables that you want to import then click **Load** to import these immediately; or you click **Edit** to open the Power Query Editor to apply transformations.
4. After you have loaded the worksheets into Power BI, you can begin working with them as Power BI datasets.

Working with a whole Excel workbook

Power BI can import any Excel .xlsx or .xls file, enabling you to explore features as if you were using Excel Online. If you have created data models by using Power Pivot, Power BI imports your tables, calculated columns, measures, and hierarchies. Power View sheets are imported and created as reports. As soon as the reports are created, you can begin pinning the visualizations to dashboards. Be aware that not all Power View visuals are supported in Power BI.

Note: If you import an Excel workbook that uses Get & Transform or Power Pivot to connect to an external data source, you can set up a scheduled data refresh. After the import has completed, Power BI uses the connection information to make a direct connection to the data source. The data is then queried and refreshed, and visualizations are updated.

The process for importing Excel files that contain Power Pivot or Power View content is the same as for a standard data worksheet. You can import the content into Power BI Desktop or upload it to the Power BI service from your local computer, or from OneDrive.

Publishing to Power BI from Excel

You can also use Excel to publish workbooks straight to the Power BI service, where you create reports and dashboards, and then share visuals with your colleagues.

- Limitations
- Two options for publishing a workbook:
  - Upload your workbook to Power BI:
    - The workbook is displayed as it is in Excel Online
    - It cannot be edited in Power BI, only in Excel
    - Use this option for workbooks containing data and no visuals
  - Export workbook data to Power BI:
    - Use this option for workbooks that use Power View, Get & Transform, and Power Pivot
    - All tables, the data model, and visualizations are exported

Limitations

There are several limitations that you must consider before publishing to Power BI from Excel:

- You must save the workbooks that you want to publish to OneDrive for Business.

- You must use the same account for Microsoft Office, OneDrive for Business, and Power BI.
- Before you publish a workbook, it must contain content that is supported in Power BI; you cannot publish an empty workbook.
- Encrypted or password-protected workbooks, or workbooks that have Information Protection Management, cannot be published.
- Modern authentication must be enabled. The **Publish** option is not available on the **File** menu if modern authentication is set to disabled.

Publishing a workbook

If necessary, save your Excel workbook to OneDrive for Business, open it from this location, click **File**, and then click **Publish**. This gives you two options for uploading your file to Power BI:

1. **Upload your workbook to Power BI**. If you choose this option, your workbook is displayed as it is in Excel Online, but you can still pin visuals in your worksheets to dashboards. You will not be able to edit your workbook in Power BI, but you can click **Edit** to open the workbook for editing in Excel Online, or on your computer. The changes are saved to the version on OneDrive. Uploading your workbook does not create any datasets in Power BI. Workbooks uploaded to Power BI have an Excel icon, to indicate that they are uploaded workbooks. This is the best option if you only have data in your workbooks, or PivotTables and PivotCharts that you want to view in Power BI. This option is similar to the Manage and View Excel in Power BI feature, in the Power BI service. Click **Get Data**, under **File**, click **Get**, click **OneDrive - Business**, and then click **Connect**.
2. **Export workbook data to Power BI**. Choose this option if you have a workbook that uses Get & Transform or Power Pivot to load data into a data model, or if the workbook contains Power View visualizations that you want to view in Power BI. Unlike the upload option, this option exports any supported tables and data models into new datasets in Power BI. Power View sheets are converted to Power BI reports, so you can instantly create dashboards from the visualizations. Furthermore, you can continue to edit your workbook in Excel. When you save changes, they are synchronized with the Power BI datasets, usually within an hour. For more immediate results, you can click **Publish** again to update the content without having to wait. Reports and dashboards that use the visualizations are updated. This option is similar to the Export Excel data into Power BI feature in the Power BI service. Click **Get Data**, under **File**, click **Get**, and then click **OneDrive - Business**.

When you click **Publish**, and choose the upload or export option, Excel signs in to your Power BI account by using the credentials for your Office account, and then publishes the workbook. The **Publishing to Power BI** status bar displays the progress of the operation.

Updating files in Power BI

If you upload a local file to Power BI to use as a dataset in your reports and dashboards, you can make changes to the file and upload it again. Providing the file name is the same, Power BI can update the file. This applies to Excel, CSV, and Power BI Desktop files. Several limitations apply:

- The file names must have the same name and the same type. If you have an Excel file named Finance, it will not be replaced with a Power BI Desktop file named Finance.

- Update local files that you have uploaded to Power BI to use in reports and dashboards:
  - Includes Excel, CSV, and Power BI Desktop files
  - The name of the file must be the same as the dataset
  - File type must be the same as the previous one
  - Keep the data structure the same
  - Power BI ignores format changes to columns
  - New columns are added to the dataset
  - Whole Excel files on OneDrive for Business or SharePoint – Team Sites are updated automatically
  - Only one dataset can exist with same name as the file

- The structure of the data must stay the same. Renaming or deleting columns that are used in a report or dashboard will break the dependent visuals.
- Power BI ignores any format changes to columns so, for example, you can change a value from 75 percent to 0.75.
- New columns are added to the dataset, but they are ignored until they are used in a visual.
- When you import whole Excel files from OneDrive for Business or SharePoint – Team Sites, the changes to the file are automatically reflected in Power BI.

To update a file in the Power BI service:

1. Click **Get Data**, under **File**, click **Get**, and then click **Local File**.
2. Browse to the file that you want to replace, and then click **Open** to upload the file.
3. Click **Import** to connect to the data and Power BI displays a message to say that you already have a dataset with that name.
4. Click **Replace it** to upload the updated file.

Note: If more than one dataset has the same name as the file that you're updating, Power BI cannot update the dataset until you rename the dataset that is not sourced from the file. There must be only one dataset with the same name as the file that you want to update.

Data refresh

The way in which data refresh works in Power BI depends on the subscription service that you are using, and the type of data source.

- Data refresh options depend on Power BI account:
  - Power BI (free): datasets can be refreshed daily
  - Power BI Pro: schedule hourly refresh, up to eight times a day
- Data source types have different refresh options:
  - Power BI to SaaS uses live connection
  - Datasets can consist of multiple data sources, such as Excel and SQL Server
- Three data refresh options:
  - Automatic refresh
  - Refresh now and scheduled refresh
  - Live connection with DirectQuery

Subscription types

The options that are available depend on whether you have a Power BI subscription, which is free of charge, or a Power BI Pro subscription:

- **Power BI (free)**. Datasets can be scheduled to refresh daily, with a maximum of 10,000 rows per hour for streaming data in dashboards and reports by using the Microsoft Power BI REST application programming interface (API), or Microsoft Azure Stream Analytics.
- **Power BI Pro**. Using a Power BI Pro account, you can schedule an hourly refresh, with up to 1 million rows per hour for streaming data in dashboards and reports by using the Microsoft Power BI REST API, or Stream Analytics. You can have up to eight hourly data refreshes per day. Furthermore, Pro accounts include data refresh for live data sources with full interactivity (Azure SQL Database, Azure SQL Data Warehouse, Spark on Azure HDInsight®), on-premises data sources that require a Power BI gateway, and on-premises SQL Server Analysis Services that require the Analysis Services Connector.

Data source types

The type of data source from which you are extracting the data determines how the data is refreshed. Software as a service (SaaS) data is automatically refreshed, so you don't need to do anything to update it.

Database connections in SQL Server Analysis Services use a live connection, which means that they always display the latest data.

After you have created a dataset in the Power BI service, it appears in the **Datasets** list in the **My Workspace** pane. You can use the ACTIONS list to refresh the data or schedule a refresh. During a refresh, Power BI connects to the data source by using the credentials that are stored in the dataset. The dataset data is refreshed, and the reports and dashboards that use this dataset reflect the changes immediately.

A dataset might consist of multiple data sources. For example, in Power BI Desktop, if you acquire data from an on-premises server running SQL Server, and other data from an Excel workbook, a single dataset is created when you publish to the Power BI service. However, this dataset contains two data sources that have connection information to both SQL Server and Excel. Be aware that, when you choose to refresh a dataset, Power BI connects to all of the data sources in the dataset so that it can refresh the data. This ensures that all data within reports and dashboards is consistently up to date.

Data refresh types

You can refresh most datasets in Power BI, but the type of data from which the dataset was created, and the data sources to which the dataset connects, determine whether you need to update it. Power BI has the following refresh options:

- **Automatic refresh**. Power BI configures the data refresh settings for data sources that benefit from an automatic refresh. For example, for files that are loaded from OneDrive, the data that does not come from an external source is refreshed approximately every hour. Although you can schedule a refresh to occur more frequently, it is unlikely that this would be necessary.
- **Refresh now and scheduled refresh**. Refresh now manually refreshes a dataset, or you can configure a schedule to run on a regular basis. Use this option for Power BI Desktop (.pbix) files, and Excel workbooks that connect to on-premises and external online data sources.
- **Live connection with DirectQuery**. If you use DirectQuery, a live connection exists between Power BI and the data source, such as a database in Azure SQL Database. You always see the latest data from the source and no manual configuration is required.

For more information about data sources and the refresh options that are available to each type, see *Data refresh in Power BI* in the Power BI documentation:

Data refresh in Power BI

http://aka.ms/Bq486n

Demonstration: Importing files from a local folder

In this demonstration, you will see how to:

- Import data from an Excel file.
- Import data from a CSV file.

Check Your Knowledge

Question	
Which of the following file formats is not a compatible data source in Power BI?	
Select the correct answer.	
	CSV
	TXT
	XML
	SQL
	JSON

Lesson 2
The Power BI data model

This lesson explores the Power BI data model. You will learn how you can use features within the data model to manage your data.

Lesson Objectives

At the end of this lesson, you will be able to:

- Describe what a data model is.
- Create and manage relationships in your data.
- Optimize the data model so your data is ready to use in visualizations.
- Understand how hierarchies can help you analyze your data by drilling down through levels.
- Create hierarchies that support your reporting and analytical requirements.

What is a data model?

- Data model typically associated with relational database
- In Power BI, connect to multiple different data sources and import into data model
- Shape optimized data ready for using in reports
- No need to flatten data
  - **Data types**: ensure data uses correct data type
  - **Fact and dimension tables**: create star schema in model
  - **Cross filtering**: bi-directional to flatten tables
  - **Reduce size of dataset**: exclude columns or rows for very large databases, or omit sensitive data

A data model is typically associated with a relational database, such as Microsoft SQL Server. In Power BI Desktop, you can connect to multiple different data sources using queries, and bring the data together in the data model. You create relationships between the tables imported from the various sources. There's no need to flatten the data you import into separate tables, you leave it in the original table structure, and create relationships. You then use these related tables for calculations or measures, and enrich the data in the model by creating calculated columns. It is best to design your model to help you build visualizations within your reports. Within the data model, you create calculated tables and columns, relationships, hierarchies, change data types, defaults, and properties. If you get all this right in your model, the creation of reports is a much smoother process, and this produces results that are more accurate.

Data types

Data that comes in from a database system such as Microsoft SQL Server is likely to be correctly typed. However, if you import from other sources, such as CSV, or the web, the data types might need to be changed. It is a good idea to correct any data types in the model so you can then begin applying the correct formatting. For example, you apply formatting to a datetime column to display the data appropriately for your region, and then use the split column function to extract the day, week, or year part into a new calculated column.

A wrong data type might cause incorrect results within your visuals. When importing numbers, be accurate and determine whether you need precise rounding. If you have financial data that you don't want to be rounded down, you use the Fixed Decimal Number data type, which includes four digits to the right of the decimal point. This is the same as the Currency type in Power Pivot. Furthermore, if you intend to include maps within your reports, ensure that address columns have the correct geo category types—for example, set Country, State, or Region to the corresponding data. After changing any data types that

need altering, it's good practice to then check that columns that Power BI has set as the default for sort orders, or aggregating, are correctly determined.

Fact and dimension tables

When we record information about an event, it is known as a **fact** in business intelligence (BI) terms. This data is stored in **fact tables**, excluding any descriptive details about that event. The details of the event are held in **dimension tables**, which are also referred to as lookup tables in a relational database. A data warehouse stores data in fact and dimension tables, and we have the same concept in Power BI. A fact table is generally on the many end of a one-to-many relationship—for example, one customer might place many orders. One fact table is usually related to many dimension tables, which creates a **star schema**, with the fact table in the center of the model, surrounded by the related dimension tables. You use dimension tables to slice and dice your data—for example, to see sales by customer, region, age range, marital status, or time.

Cross filtering

When you turn on **bi-directional cross filtering** in your relationships, this enables the tables in your star schema to operate as if held in a single table, and you can join and aggregate values between dimension tables. There is no need for you to flatten your tables, as you can achieve this by changing the cross-filter type. When you have multiple fact tables, initially start with single direction cross filtering to relate the tables correctly, before adding bi-directional cross filtering. This method ensures you have your basic relationships working first, and prevents confusion and ambiguity in your visualizations. You can also use bi-directional cross filtering to overcome Power BI's lack of support for many-to-many relationships.

However, you might have a problem creating the relationships you need when you have two fact tables joined to one or more dimension tables. In this case, you create calculated tables using existing dimension tables and then create new relationships. To do this, delete the relationships from one of the fact tables to the dimension tables, and then create new calculated tables using the existing dimension data. You can then create relationships between these new tables to the fact table from which you deleted the relationships, and set the cross-filter direction to be bi-directional. The calculated table works the same as any other table, and is updated when the model is updated, though it will increase the size of your model.

Reducing the dataset size

While you can use bi-directional cross filtering and avoid the need to flatten your data, in some instances you might want to do some flattening of your data before import. If you have a very large database, it might be beneficial to join tables, or apply filtering within the query to reduce the volume of data you include in your model. Furthermore, your database might contain sensitive data that you don't want to import, so for security reasons, you might choose to exclude such data in the query.

Managing data relationships

Relationships might be created automatically by Power BI or created manually—a relationship describes how one table relates to another. Columns are related using key fields, comprising primary and foreign keys.

- Relationships are either created automatically by Power BI, or you create them manually
  - **Primary keys**: uniquely identify each row in a table; surrogate key based on nonbusiness data, such as an incrementing whole number
  - **Foreign keys**: join to the primary key table to create relationship; ensure data integrity and prevent deletions in the primary key table
  - **Creating relationships**: Power BI creates any apparent relationships when data is imported
  - **Viewing relationships**: view and manage relationships in the data mode in a diagrammatic view

Primary keys

A primary key column uniquely identifies each row within a table. At the end of a relationship, one of the tables must contain unique values. A primary key is frequently created using a surrogate key, particularly in a database.

A surrogate key is usually a sequential number, starting at 1, and incrementing by 1, for each row that is added to the table. A surrogate key uses information that is not actually business data. However, business data can be used and an **Employee** table, for example, might have a primary key column based on the social security number, which uniquely identifies each employee, as shown in the following table:

SocialSecurityNo	FirstName	LastName
123 456 ABC	Lucinda	Smith
987 321 ZYX	Elizabeth	Jones

If another employee named Lucinda Smith joins the organization, she can be uniquely identified because the data in the **SocialSecurityNo** column will differ from the data stored for the existing Lucinda Smith. In many cases, data will not have a natural key. If you have a **Category** table, the data is likely to be a list of category names, which will not necessarily be unique. To make them unique, we add a surrogate primary key column named **CategoryID**, as shown in the following table:

CategoryID	Category
1	Frozen Food
2	Pet Supplies
3	Dairy

Each time a new category is inserted into the table, the CategoryID number is incremented. The primary key forces uniqueness within the table, and becomes the basis for foreign keys.

Foreign keys

Foreign keys enforce data integrity by ensuring that the data table in one table is correctly related to data in another table. Furthermore, it prevents rows in the primary key table from being deleted, leaving orphaned records in the foreign key table. For example, a **SubCategory** table has a **SubCategoryID** primary key column to uniquely identify each subcategory; it also includes a **CategoryID** foreign key column to the **Category** table, which acts as the parent within the relationship. This relationship can be seen in the following table:

SubCategoryID	CategoryID	SubCategory
1	1	Ice Cream
2	1	Sorbet
3	2	Dog Treats
4	2	Dog Food
5	2	Cat Food
6	3	Cheese

The relationship between the tables enables you to work with the data as if it was one table:

SubCategoryID	CategoryID	Category	SubCategory
1	1	Frozen Food	Ice Cream
2	1	Frozen Food	Sorbet
3	2	Pet Supplies	Dog Treats
4	2	Pet Supplies	Dog Food
5	2	Pet Supplies	Cat Food
6	3	Dairy	Cheese

The **Frozen Food** category cannot be deleted from the **Category** table, because there is at least one related table in the **SubCategory** table. If the category were deleted, the child records in the **SubCategory** table would have incomplete data that would not make for useful reporting. By using relationships, this prevents repeated data. In the above table showing the data joined from the Category and SubCategory table, each Category name is repeated for the corresponding row in the SubCategory table; however, it only exists once in the Category table. This is important when data is updated, as only one row requires altering.

Creating relationships

You use relationships to work with data as if all the related tables were a single table. For example, if your **Product** table is related to the **SubCategory** table using a **SubCategoryID** column, you can display the product name from the Products table alongside the subcategory name from the SubCategory table. The SubCategory table then joins to the Category table making the data appear seamless in your reports, and enabling users to slice data.

When you import data into the data model, Power BI detects existing relationships. When you connect to a relational database such as SQL Server, the data is most likely to be related already, and Power BI detects the existing connections between the tables to create the relationships. However, if you import two tables with a common column but without a relationship, Power BI is likely to detect the commonality and still create a relationship. For example, if you have two tables named **Department** and **Manager**, each containing a **DepartmentID** column, even though there is no relationship between the tables, Power BI works out that the columns and data match, and automatically creates the relationship.

If you drag fields onto the report canvas to combine with other data in a visual, Power BI checks to see if the data is related and can be combined. If it can't relate the fields, a warning message appears, and you can click to choose whether you want Power BI to autodetect the relationships between the fields, or to manually create a new relationship.

Viewing relationships

You view and manage the relationships within your data model by clicking **Model** on the left side of the Power BI Desktop window. This displays your tables diagrammatically, and you instantly see how one table relates to another. Furthermore, you can move the tables around while maintaining the links between them, so they are easier to understand. If you have many tables within your model, you use the zoom feature to zoom in or out and see a closer view of the tables and columns. The **Fit to screen** button

lays out the tables so you see all tables and relationships in a single view on the screen. After moving your tables around, you use the **Reset layout** button to return to the default layout.

Before you begin adding visuals to your report and hooking up the data, ensure you have correctly established the relationships, because this facilitates the ability to present accurate data.

Optimizing the model for reporting

Data is frequently in a raw format when you import it into Power BI, especially if you have taken it directly from a database. Furthermore, you might have fields you don't need if you have not been able to use a query to reduce columns in the dataset you have imported. When you combine data from different data sources into your data model, it is highly probable that the different source systems or files have applied different formats or data types. There's a few things you can do to optimize your data, and make it more consistent. This helps you to work with your data more efficiently, focusing on the information you need. It is also helpful to colleagues or anyone with whom you might share the data.

- Imported data from a database is often very raw:
  - Data in the model might not be formatted or optimized
  - Inconsistency in data types for data from different sources
- Optimize data in model:
  - **Hide fields**: hide fields not used in visuals in report; makes model easier to use, useful for large tables
  - **Sort data**: display data in correct order in visuals, such as ordering by day name or month name
  - **Format data**: change data types and formatting; especially useful for datetime and currency fields

Hiding fields

It's a good idea to hide fields that you know you are not going to use in your visuals. Hiding a field removes the field name from the **Fields** pane, and neither the column nor the underlying data is deleted. To hide a field, in **Report** view, right-click the field in the **Fields** pane, and click **Hide**. Right-click anywhere on a table or field to open the menu and choose **View hidden**, so hidden fields reappear in the list of fields, and show in their original position in the order of the fields. Hidden fields are shown in grey text to indicate their hidden status. You can also select **Unhide all** to make hidden fields visible again. If you switch to the data view, you will see that hidden columns also display with the column header and data in grey text.

Sorting data

If you drag a column such as **DayOfWeek**, or **MonthName** to a column chart visual to display sales figures, the values are automatically ordered alphabetically. However, it's unlikely you want the data to be ordered in this way, and require the columns to be in day or month order, rather than by name. In **Report** view, in the **Fields** pane, click the column to sort and then, from the **Sort** group on the **Modeling** ribbon, click **Sort by Column**. When you select a column from the Sort by Column list, you will see that Power BI has selected one column as the default for the sort, so when this best guess is incorrect, it is quick and easy to change. Choose another column, such as **DayNumber**, or **MonthNumber**, so the columns are ordered correctly. For example, rather than showing as April, August, December, and so on, month names now appear as January, February, March, and so on. This makes data analysis much easier as the user can read the data in the correct time order.

Note: This ability to sort data is also applicable to other items you usually refer to in a specific order, such as product categories, projects, or departments. When working with visuals, try to present data in the optimum way for enabling the end user to quickly digest the presented information.

Formatting data

Changing data types and formatting data are good ways of optimizing your data. When you drag a datetime or currency field onto the report canvas, you might find the style of the date is not easy to read, or not in your local formatting, or that sales figures display without a currency symbol. From the **Formatting** group on the **Model** tab, you select the **Data type** and **Format** menus to customize your data. Power BI enables menu formatting items that are relevant to the data type of the column that you highlight in the **FIELDS** pane. You might wish to change a data type first, and then change the available formatting options. You can format how datetime data types are displayed, in addition to setting financial columns to always include your local currency symbol. This presents the data with clarity in your reports and dashboards.

What are hierarchies?

You use hierarchies to drill down into your data. A hierarchy is a set of related fields grouped together that you can use to drill from one level in the hierarchy to the next. Each level within the hierarchy is contained within the next level—it cannot exist independently. Power BI will automatically create a hierarchy for you on datetime fields; however, you can also create your own hierarchies within the model to suit your requirements for analysis. The following is an example of an address hierarchy you could create in Power BI and use to drill through from top to bottom:

- Country
- State/Province
- City/Town
- Street
- House Number or House Name

- Hierarchies enable drill-down into your data
- A hierarchy is a set of related fields grouped together
- Each level is contained within the next level and does not exist independently
  - Example: Country, State, City
- Time intelligence: Power BI automatically creates drill-down on date columns for:
  - Year, Quarter, Month, Day
- Can also use DAX time intelligence functions to aggregate date for specific time periods

Time intelligence

If you have date columns in your data that you have used in a report in Power BI, you might have noticed that the ability to drill through is created for you. Power BI automatically splits date data into:

- Year
- Quarter
- Month
- Day

If you drag a date column to the axis field bucket of a visual, you see this in action, because you can immediately start drilling into the data over time. You use DAX time intelligence functions to aggregate your data over different time periods. DAX is covered in detail in Module 5 of this course, but to find out more information, see *Timeintelligence functions* in Microsoft Docs:

Time-intelligence functions

https://aka.ms/umjsqc

Creating hierarchies

- Create hierarchies in Power BI **FIELDS** pane:
  - Select column and click **New hierarchy**
  - Right-click and click **Rename** to specify new name
  - Click ellipsis of another column, and **Add to hierarchy**
  - Repeat to add columns; can also move, delete, rename
- Drag hierarchy to **Axis** field bucket of a visual
  - Creates navigable hierarchy in the visual
  - Click the button **Click to turn on Drill Down**
  - Click data points to drill down into data
  - Expand to see all data in a level
  - Use filters in any level to exclude data from visual

You use hierarchies to drill down into your data. By creating custom hierarchies in Power BI Desktop, you can drill through levels applicable to your data, to support your exact analytical requirements. Furthermore, you can add multiple hierarchies to a table in your data model. Use the following steps to create a new hierarchy:

1. In the **FIELDS** pane, expand the table where you want to create the hierarchy.
2. Click the ellipsis on the column you want to use as the top level of your hierarchy, and then click **New hierarchy**. This creates a new field below the column you selected. By default, this is given the name of the column—for example, **Product Hierarchy**. You right-click the new hierarchy and click **Rename** to give it another description.
3. Click the ellipsis on the column where you want to add to the hierarchy at the next level down. Click **Add to hierarchy**, and then click the name of the hierarchy. If you have already created a hierarchy, it will be listed in the menu.
4. Repeat step 3 for each column you want to add to the hierarchy.
5. If you have added a column incorrectly, you click it in the hierarchy and click **Delete**, or drag the column and move it up or down in the hierarchy to reposition it. To rename a field in the hierarchy, right-click the field and then click **Rename**.

After you have added all columns to complete the hierarchy, you drag the hierarchy to the Axis field bucket, such as on a column chart. You will see the data aggregated for the values in the top level of your hierarchy. To begin drilling down into the lower layers, click the **Click to turn on Drill Down** arrow icon in the top right corner of the visual to enable drill-down capability. When you click a data point, the visual drills into the next level in the hierarchy. In the left corner of the visual, the title shows the levels in the hierarchy the data is displayed for—for example, using an address hierarchy that might be **Sales by Country and State**. When you drill into the next level, this becomes **Sales for Country, State and City**. You will see which filters have been applied in the **FILTERS** list. This is particularly useful for hierarchies with deeper levels.

Above the title are further buttons you can use to traverse the data. Click the **Drill Up** button to move up one level in the hierarchy. Use the button with two down arrows, **Go to the next level in the hierarchy**, to go to the next level down, without filtering on data, so you don't have to click a data point and drill into one value within a level. When the data cannot be drilled into any further, the button is disabled and displays **At the lowest level of data**.

The **Expand all down one level of the hierarchy** button enables you to expand the hierarchy: for example, with a hierarchy of Country, State, City, at the State level, rather than clicking into each state to see all sales for every city, you can expand to see all values at the next level. So if the visual currently displays all states within the United States that have sales data, rather than clicking into each state to see all the cities, use the **Expand all down one level of the hierarchy** button to show sales for every state and every city.

In the **Axis** field bucket, click the **X** icon to remove a level in the hierarchy; for example, click to remove **State**, and go from **Country** directly to **City**. To add levels back into the hierarchy, click the arrow alongside the hierarchy name in the Axis field bucket, and then click **Show all levels**. You will achieve the same result by clicking the check boxes in the hierarchy in the **FIELDS** pane. You can also right-click a

data point and select **Exclude** to remove it from the visual. You can include or exclude values using **FILTERS**.

Furthermore, you can add filters to the columns included in the hierarchy, hide columns, and create new measures and columns, so the hierarchy is fully customized for your reporting requirements.

Demonstration: Creating a hierarchy

In this demonstration, you will see how to:

- Create a hierarchy.
- Use the hierarchy to navigate data.

Question: Discuss some of the different data sources that you might use in your organization to import data into the data model in Power BI. What problems would you need to overcome? How easy would it be to relate the data in tables from different sources? How easy do you think it would be to use data from the web within your model?

Lesson 3
Using databases as a data source for Power BI

In this lesson, you will learn how to connect to on-premises and cloud instances of SQL Server, SaaS connectors, the R script data connector, and other data sources.

Lesson Objectives

At the end of this lesson, you will be able to:

- Connect to SQL Server databases.
- Import data from other databases, web pages, and files.
- Use the R script data connector to import predictive data.

SQL Server

Microsoft SQL Server is a popular relational database management system (RDBMS). Unlike Microsoft Access, which is designed for a single user or very small company, SQL Server can handle multiple user connections, high volumes of transactions, and scales from the smallest to largest of databases. SQL Server can run in the cloud, but on-premises instances remain widespread, particularly in medium to large enterprises. The steps for connecting to SQL Server from Power BI are much the same as connecting to other database systems, such as Oracle, MySQL, and IBM DB2.

- SQL Server is a relational database management system (RDBMS):
  - Unlike Access, SQL Server can handle multiple users and transactions
  - Scalable from smallest to largest size databases
  - Cloud and on-premises versions
- Connect from Power BI Desktop:
  - Connect using **Get data**; enter the name of the server instance and optionally the name of database
  - Use a query or select tables and views
  - Load into Power Query Editor or straight into data model

Connecting from Power BI Desktop

To connect from Power BI Desktop:

1. Open Power BI Desktop, and click **Get data**.
2. Click **SQL Server database** and then click **Connect**. This opens the SQL Server database connection dialog.
3. Type the name of the SQL Server instance into the **Server** text box. If more than one instance is running on the server, you might need to type in **<servername>\<instancename>**.
4. Optionally, you can type the name of the database you want to connect to in the **Database** text box. If you don't include this, you will connect using the default database associated with your account. However, if you expand **Advanced options** and want to use a query to return data, you will need to specify the name of the database in the Database text box.
5. You can choose to **Import** the data into Power BI, or use **DirectQuery**, which does not import the data and queries the underlying data source as you create and interact with a visualization. DirectQuery is useful for very large databases that are likely to import big datasets that would be slow to work with. If you are uncertain, leave **Import** selected, and click **OK**.
6. If you have chosen to use a query, you will see a sample of the results in the next window. Otherwise, you can select tables and views from the list to see a preview of the data. To import data, select each table and preview what you want to include.

7. Optionally, you can click **Select Related Tables** to import data from any tables that have a relationship to the one you have selected. This feature is particularly useful for fact tables that comprise multiple foreign key relationships to dimension tables.
8. After selecting the tables that you want to import, click **Edit** to load the data and begin working on it in the Power Query Editor; or click **Load** to load the data and return to the main Power BI window.

After your data has been loaded into the Power BI data model, you can begin shaping and transforming the data, and applying other optimizations.

Other data sources

Power BI offers a wide choice of compatible data sources that you can use for creating datasets in your reports and dashboards. You have more choice of data sources when you use Power BI Desktop than when you use the Power BI service. After you have imported data into Power BI Desktop from a source that you cannot directly connect to by using the Power BI service, you upload the dataset to work with it on the Power BI portal.

- Connect to a wide range of data sources from Power BI Desktop:
  - More data source connections than the Power BI service
  - Combine data from multiple SaaS providers into one report or dashboard
  - SaaS providers include Bing, Google Analytics, Facebook, Salesforce, Marketo, GitHub, Microsoft Dynamics, and Exchange
  - Supports industry database providers such as Access, Oracle, IBM DB2, MySQL, Sybase, and Teradata
  - Connect to any webpage to scrape structured data
  - Copy and paste from an Excel or text file to create a new table in the dataset

SaaS connections

You connect to an increasing number of SaaS providers to import data from the third-party online solutions that your organization uses. From Power BI Desktop, you import data from different SaaS providers and combine the data in reports and dashboards. For example, you could create a report that showed marketing data from Facebook and MailChimp campaigns, combined with the resulting sales that used data from Salesforce. SaaS providers include, but are not limited, to:

- Google Analytics
- QuickBooks Online
- MailChimp
- Facebook
- Dynamics 365
- Microsoft Exchange
- Active Directory®
- Salesforce
- Marketo
- GitHub
- Zendesk

Other databases

Power BI includes support for the main industry databases for importing data. Database connectors include:

- Microsoft Access®
- Oracle
- IBM Db2
- MySQL
- SAP HANA
- PostgreSQL
- Sybase
- Teradata

Web page data

From Power BI Desktop, you can connect to any webpage to extract the data. Power BI scrapes the data into tables in the dataset. Depending on the webpage that you are scraping, you might not be able to determine table names or the structure of the data, but you can perform operations to rename the fields and tables after importing the data into Power BI Desktop. To do this:

1. On the ribbon menu, click **Get Data**.
2. In the **Get Data** window, click **Other**, click **Web**, and then click **Connect**.
3. Type or paste the webpage address into the **URL** box, and then click **OK**. If you have previously created a parameter, Power BI gives you the option to use a parameter value for the URL. Power BI imports the structured data, and ignores page titles and text.

Copy and paste

You can quickly create a table in Power BI by copying and pasting data directly from Excel or from a text file:

1. Copy the data you want to use to the Windows clipboard.
2. On the **Home** ribbon, click **Enter Data** to open the Create Table window.
3. Right-click in the first cell and then click **Paste** to paste the data from the Windows clipboard. The table is created within your dataset and you can work with it just as you do with other tables. If you include column headers, Power BI detects these and uses them as the column headers in the new table. You can also manually enter data and add columns.
4. In the **Name** box, type the name of the table, and then click **Load**.

R script data connector

The highly popular statistical R programming language has been integrated into the Transact-SQL language so that data scientists can develop predictive applications in R and deploy them in a SQL Server production environment. This feature was introduced with SQL Server 2016 and is known as SQL Server R Services. This service enables you to run R scripts in Power BI Desktop and import the results into a Power BI Desktop data model. You create reports by using this data that is then uploaded to the Power BI service.

- Run R scripts from Power BI Desktop:
  - Import results of R script into datasets to create reports. Publish to the Power BI service to use in dashboards
  - Must install Microsoft R Open prior to running scripts
  - Write R scripts in local environment, and test to ensure they run successfully before using in Power BI Desktop
  - Limitations include:
    - Only data frames are imported, so include all required data
    - Time-out period is limited to 30 minutes
    - N/A values are converted to NULL values
    - Complex and Vector type columns not imported, error in table
    - Working directory of R script must be full path, and not relative
  - Manage your R installations in **Options** dialog box

Installing R

To run R scripts from Power BI Desktop, you must install a local instance of R. For further information about downloading and installing R Services, see *Set up a data science client for R development on SQL Server* in Microsoft Docs:

Set up a data science client for R development on SQL Server

http://aka.ms/r2r8xh

Running R scripts from Power BI

After installing R on your local workstation, you can begin running R scripts to import data and create reports. You must first write and test the scripts in your local development environment, to ensure that the scripts run successfully. There are several limitations that should be observed before you run a script:

- Only data frames are imported, so all of the data that you want should be included in the data frame.
- The time-out for the query is limited to a maximum of 30 minutes. The script stops executing if it has to wait for user input.
- N/A values are converted to NULL values.
- Complex and Vector type columns cannot be imported, and will be replaced with error values in the table.
- When you set the working directory of the R script, you must use a full path, not a relative path.

To run R scripts from Power BI:

1. Open Power BI Desktop, and then on the ribbon menu, click **Get Data**.
2. In the Get Data window, click **Other**, click **R Script**, and then click **Connect**.
3. Type or paste your script into the script box, and then check that the location where the R script is installed is correct—for example, C:\Program Files\Microsoft\MRO\R-3.2.2.
4. If you have multiple installations, select the one that you want or explicitly provide the full location, and then click **OK**.

R script options

You can also manage your R installations in the **Options** dialog box:

1. In Power BI Desktop, click **File**, click **Options and settings**, and then click **Options**.
2. Click **R scripting**, select your R home directory from the list, and then click **OK**. The option that you choose here is then used as the default in the R script data connector.

Demonstration: Importing data from SQL Server

In this demonstration, you will see how to:

- Connect to SQL Server from Power BI Desktop.
- Import data into the Power Query Editor.

Verify the correctness of the statement by placing a mark in the column to the right.

Statement	Answer
True or false? You can use the Power BI Q&A natural language to ask questions of your data when using DirectQuery.	

Lesson 4
The Power BI Service

This lesson explores some of the advanced features in the Power BI Service—including how to use natural query language with Power BI Q&A to ask questions of your data and how to create apps to share collections of dashboards and reports.

Lesson Objectives

After completing this lesson, you will be able to:

- Configure your data to use the Q&A feature to ask natural language questions in dashboards.
- Create apps.
- Define workspace roles to share apps with colleagues.

Configuring your data for Q&A

The Q&A box appears at the top of your dashboards, and enables you to ask questions of your data by using natural query language. Q&A recognizes the words in your questions, and works out where in your dataset it can find an answer. Furthermore, Q&A helps you to formulate your question by using autocomplete, restatement, and dimming of words that it does not understand. Q&A displays the answer as an interactive visualization. Unless you specify the type of visual that you want, Q&A uses the one that it determines is most appropriate. For example, if you asked, "What were last year's sales by territory?" Q&A would know to use a map. However, you could ask, "What were last year's sales by salesperson as a pie chart?" so that you specify the exact visual that you want to represent your answer.

- Power BI Q&A:
  - The Q&A text box appears at the top of every dashboard
  - Enables users to ask questions of their data by using the natural query language
  - Q&A returns answers based on the dataset in the dashboard, using an appropriate or user-specified visual
- Q&A depends on entity names for searches:
  - Can use structured data and uploaded Excel workbooks
  - Upfront data cleaning and optimizations boost the performance of Q&A to deliver better results
  - Tables, columns, and calculated fields should be named appropriately, by using words you would search on

For more information about asking questions by using Power BI Q&A, see *Get started with Power BI Q&A* in the Power BI Documentation:

Get started with Power BI Q&A

http://aka.ms/AA4yc31

Q&A searches structured data, and can work on any Excel workbook that you upload. However, upfront data cleaning and optimizations can help to boost the performance of Q&A, and deliver the answers that you need. Consider the names that you give to your entities. For example, if you have a table named **Internet Sales**, columns named **Category**, **Product**, **Units Sold**, **Cost Price**, **Gross Sales**, **Month**, and **Year**, and a calculated column named **Profit**, you see how easy it is to find answers to questions such as, "What were the sales last year by category and month?" Q&A also understands how to filter, sort, aggregate, and group data, which you can include in your question, so you could ask, "What were last year's sales by month sorted by profit?" By being clear in your naming conventions, you see how Q&A can more easily deliver answers to your questions.

Note: The Power BI Q&A natural language only works with cloud-based datasets that have been uploaded to the service, so you cannot use it with an on-premises tabular model in SQL Server Analysis Services.

Creating apps

- Power BI content packs are packaged reports, dashboards, and datasets:
  - Can be shared with other Power BI users
  - After importing, the contents are merged into the My Workspace lists
  - Packs can be customized for different users
  - Give access to specific groups, or entire organizations
  - Add title, description, and image or company logo
  - Datasets for the selected reports and dashboards cannot be excluded
- Import content packs from SaaS providers such as Bing, MailChimp, Insightly, Marketo, and Twilio

Apps are packaged reports, dashboards, and datasets that can be shared with other Power BI users in your organization. When you install an app on the Power BI portal, the report items are merged into your workspace lists. Users who have a free Power BI account can only view apps that are published to Power BI Premium capacity—they cannot create them. You can create apps to customize reports or dashboards for users in different departments within your organization. For example, you could create a set of reports with targeted visuals for finance, sales, and manufacturing, because each department is likely to want different data with which to measure performance.

When you publish an app, you choose who you want to give access to. You choose specific groups, such as sales or human resources, or you give access to your entire organization. The app can be customized with a title and a description to help users to determine whether the app is applicable to their needs. Furthermore, you can upload an image or company logo for the app. You choose the reports, dashboards, and datasets that you want to include, but when you choose a report or dashboard, it automatically includes any required datasets—these cannot be excluded. The app is then available in your organization's content gallery. Users who have access to the app can create new dashboards from the contents.

Importing content packs

You can install apps from SaaS providers such as Adobe Analytics, Alpine Metrics Sales Predictions, Insightly, Marketo, and Twilio. To add an app from an SaaS provider with whom you have an account:

1. Sign in to the Power BI service, click **Get Data**, and then under **Services**, click **Get**.
2. Click the provider's SaaS logo, and then click **GET IT NOW**. You will be prompted to enter your login details for the service.
3. After you have been authenticated, you can import an app that contains reports and dashboards that have been designed to visualize your data without you needing to do any work.

Note: Only users who have a Power BI Pro subscription can create and share apps. You do not need a Power BI Pro account to view apps from your organization or from SaaS providers.

Defining workspace roles

When you create an app workspace, you grant access to it by adding user groups or individuals to workspace roles. These roles define which users can view, create, and share information in the app, and which users can administer the app workspace.

Three roles are available: Admin, Member, and Contributor. The information in the following table describes the capabilities assigned to users in each role.

- Grant users access to app workspaces using roles:
  - Contributor
    - Publish reports
    - Create, edit, and delete workspace content
  - Member
    - Share items and apps
    - Publish and update apps
    - Add members to workspace
  - Admin
    - Add or remove users
    - Update or delete the workspace

Capability	Contributor	Member	Admin
Publish reports to the workspace	Yes	Yes	Yes
Create, edit, and delete content in the workspace	Yes	Yes	Yes
Share an item or an app		Yes	Yes
Publish or update an app		Yes	Yes
Add members to the workspace		Yes	Yes
Add/remove users, including admins			Yes
Update or delete the workspace			Yes

To add users to the workspace:

1. Sign in to the Power BI service, and then click the arrow next to **Workspaces**.
2. In the expanded pane, click the ellipsis next to your workspace name, and then click **Workspace Access**.
3. In the **Access** pane, in the **Enter email addresses box**, type an email address or security group name, in the dropdown list select **Member**, **Contributor**, or **Admin**, and then click **Add**.
4. When you have finished adding all the required users, click **Close**.

Demonstration: Querying data by using Q&A

In this demonstration, you will see how to:

- Ask a question by using Q&A.
- Pin the answer to a question to an existing dashboard.
- Ask a question and specify the visual to represent the data.

Question: How could your organization use apps? What are the major advantages of apps?

Lab: Importing data into Power BI

Scenario

Adventure Works employees are increasingly frustrated by the time that it takes to implement managed BI services. The existing managed BI infrastructure, including a data warehouse, enterprise data models, and reports and dashboards, are valued sources of decision-making information. However, users increasingly want to explore relationships with other, currently unmanaged data, and it takes too long for the IT department to incorporate these requirements into the corporate BI solution.

As a BI professional, you have been asked to explore ways in which Adventure Works can empower business users to augment their managed enterprise BI solution with self-service BI.

Objectives

After completing this lab, you will be able to:

- Alter an Excel file to reduce its size, and then import the data into Power BI Desktop.
- View existing Excel Power View worksheets as reports in Power BI.

Note: Because of updates to Microsoft Power BI, the lab steps for this course change frequently. Microsoft Learning regularly updates the lab steps, so they are not available in this manual – but you can access them on GitHub.

Lab Setup

Estimated Time: 60 minutes

Virtual machine: **20778C-MIA-CLI**

User name: **ADVENTUREWORKS\Student**

Password: **Pa55w.rd**

All the lab steps are contained in 20778C_LAB_03.md.

Exercise 1: Importing Excel files into Power BI

Scenario

As a data analyst for Adventure Works, you will be using Power BI to create reports that the business analysts can use to create dashboards in the Power BI service. One of the business analysts has asked you to import an Excel file as the basis for a report. The file contains formatting that needs to be removed before you can import it. You will remove the formatting, and then import the data in the workbook to create a new dataset. As part of this exercise, you will alter the column names so that they are more suitable for Q&A to find answers within the dataset.

The main tasks for this exercise are as follows:

1. Prepare the lab environment
2. Reduce the size of Excel files
3. Import Excel files

▶ **Task 1: Prepare the lab environment**

▶ **Task 2: Reduce the size of Excel files**

▶ **Task 3: Import Excel files**

Results: After this exercise, the data in Excel will be available as a dataset in Power BI.

Exercise 2: Viewing reports from Excel files

Scenario

A business analyst has emailed to you an Excel workbook that contains a Power View report. The analyst wants you to upload the file to Power BI so that the sales department can reuse the work that has already been done on creating the interactive visuals. You will sign in to Power BI and upload the report.

The main tasks for this exercise are as follows:

1. View Excel Power View sheets as Power BI reports

▶ **Task 1: View Excel Power View sheets as Power BI reports**

Results: At the end of this exercise, the Power View report will be available as a Power BI report.

Question: Discuss the different data sources that your organization could use to create Power BI reports. Can you think of a scenario where users perhaps have Excel workbooks for one set of reports, and reports in SQL Server Reporting Services for another set of data? Could this be combined into a single dataset in Power BI?

Module Review and Takeaways

In this module, you have learned how to use Power BI to enable users to easily access data and create reports. You have seen how to publish data from Excel and from SQL Server and other types of database. In addition, you have seen how to use Q&A to ask questions in natural query language and how to share your reports with colleagues.

Review Question(s)

Question: Discuss the different ways in which Power BI could reduce your organization's dependency on shared Excel files. How would having a central location for data, reports, and dashboards benefit different departments? How could each department make use of features such as apps and the natural query language in Q&A?

Module 4

Shaping and Combining Data

Contents:

Module Overview

Power BI Desktop offers a self-service solution for creating visual, interactive reports and dashboards. Users can connect to a wide variety of data sources, combining data from on-premises databases, Software as a solution (SaaS) providers, cloud-based services, and local files such as Microsoft Excel®, into one report. The beauty of Power BI reports and dashboards is the ability to rapidly build reports to present this data so it is instantly readable—with clusters, outliers, and patterns in data visually brought to light. To achieve this, each report must have a dataset comprising tables and columns that are ready to add straight into visualizations. Data must be formatted for relevant currencies, numbers should have correct decimal places, additional columns and measures might be required, and data may have to be combined from multiple tables. With Power BI Desktop, you can do all of this, with powerful, built-in tools for shaping your data. This module introduces the tools that are available for preparing your data, and transforming it into a form ready for reporting.

Objectives

After completing this module, you will be able to:

- Perform a range of query editing tasks in Power BI.
- Shape data, using formatting and transformations.
- Combine data together from tables in your dataset.

Lesson 1
Power BI Desktop queries

In this lesson, you will learn about the tools in Power BI that shape and transform your data, so that it is ready to use in reports. You will also explore the main features of the Advanced Editor.

Lesson Objectives

After completing this lesson, you will be able to:

- Use the Power Query Editor to shape your data.
- Roll back your data shaping steps using the APPLIED STEPS feature.
- Change the code that Power BI uses to query the data sources.

The Power Query Editor

By using the Power BI Power Query Editor, you can load data from a wide number of data sources, and apply transformations, including adding new columns and measures. There are two ways of accessing the Power Query Editor: you can click **Edit** when loading data using **Get Data**, or, from Power BI Desktop, on the **Home** menu, in the **External data** group, click **Edit Queries**. In the Power Query Editor window, click a table or view in the Queries pane, to display the data. The column names are prefixed with letters (ABC), numbers (123), currency symbols ($), or a calendar for datetime columns to represent the data type of the column. The Power Query Editor ribbon comprises four tabs for shaping your data:

- Enables you to load data and apply transformations
- Ribbon comprises of four tabs:
  - **Home**: import data, hide or delete columns, reduce rows, merge and append queries
  - **Transform**: create aggregated columns, transpose, pivot, unpivot, split values
  - **Add Column**: add columns, add indexes, apply functions
  - **View**: show or hide the QUERY SETTINGS pane

Home

You can import data from the Power Query Editor using **New Source**, **Recent Sources**, or **Enter Data**, in the **New Query** group. These offer the same functionality as the **Get Data**, **Recent Sources**, or **Enter Data** options in the main Power BI design window.

In the **Data Sources** group, click **Data source settings** to change the properties of the data source. You cannot edit the query, but you can change the server and database, in addition to the login details. You can also choose to encrypt the connection and set the security level.

You use the **Parameters** group to manage and create parameters that can be used in a variety of ways within the report. Click **Manage Parameters** to edit the properties and data values of your parameters, or delete a parameter. Use **Edit Parameters** to select the current values for each of the parameters within the report. To add a new parameter, click **New Parameter**. You can explicitly specify the values for the parameters, or use a query, in addition to setting the data type and determining whether a value must be supplied.

The **Query** group includes a function to refresh the preview data for the current table, or all tables in the dataset. Click **Properties** to edit the name of the query, and the optional query description, and to configure whether to **Enable load to report** and **Include in report refresh**. Click **Advanced Editor** to view and edit the query code.

With the **Manage Columns** group, you can hide columns in your query using the **Choose Columns** function. You can also add them back in later if they are required. **Remove Columns** removes the currently hidden column(s) from the query.

Use the **Reduce Rows** group to keep or remove rows from the query. By using **Keep Rows**, you can retain the top or bottom specified number of rows, or use **Keep Range of Rows**, by entering the **First row** as the starting row, and then **Number of rows**. All other rows are removed. **Remove Rows** gives you the option to remove the top, or bottom specified number of rows, alternate rows, or blank rows. You can use **Remove Duplicates** to delete rows with identical values in one or more columns. You can also choose to **Remove Errors**, or **Keep Errors**.

The **Sort** group provides options for sorting data from **A-Z** or **Z-A**. You can apply a sort to multiple columns in a query, though you should always start with the column that has the less than unique values. For example, apply the sort in order of Country, Region, and then City.

For quick access, you will find the most common transformations on the **Transform** group. These are also included on the Transform tab.

The **Combine** group provides functions for merging and combining data between queries. You can use **Merge Queries** to combine columns, or use **Append Queries** to combine rows.

Transform

The **Table** group on the **Transform** tab includes a **Group By** function with which you can create a new column by applying an aggregation function to an existing column. This group also includes **Use First Row As Headers**, which is useful when importing data and Power BI has not detected that the first row contains the header. **Transpose** converts columns into rows, and rows into columns; **Reverse Rows** reorders the rows so that the last rows are at the top, inverting the order of the data. **Count Rows** returns a count of rows in the table.

The **Any Column** group functions can be applied to all columns in your table, regardless of data type. You can change the data type of columns, rename columns, replace values, and errors, and use the **Fill** function to fill empty cells with a neighboring value, going up or down. You can also move, pivot, and unpivot columns.

With the **Split Column** function in the **Text Column** group, you can split one column into one or more columns, based on a delimiter. This is very useful for extracting concatenated strings.

The **Structured Column** group gives you options for working with nested data such as tables or lists. The **Expand** function promotes nested data to become columns or rows in the top-level table. **Aggregate** summarizes data from a nested structure to reveal average, minimum, maximum, and count values.

You use the **Run R Script** function in the **Scripts** group to run R queries directly in the editor. You must have R installed to use this function.

The rest of the items included on this tab are also available on the **Add Column** tab.

Add Column

You can use the **Custom Column** button to create a new column using a formula. A list of available columns to work with is included, and the syntax checker ensures your formula is correct before applying to create the new column. You use the **Index Column** button to create a new index column on a table. The index might be an incremental value, starting at 1, and incrementing by 1—you can also set the starting value to be 1 or 0. With the custom index column, you set the starting value and increment to any value. You can also duplicate columns using existing values, applying built-in string functions such as uppercase, lowercase, or capitalizing each first letter.

You can use **Conditional Column** to add a column based on the values in another column. For example, if you have a Title column, and want to create a new Gender column, you could specify that if the value in the Title column was Mr, then the value in the new Gender column would be Male—for all other values, it would be Female. Instead of explicitly specifying the comparison value of Mr, you could choose a parameter to supply this value. If you create the new column based on a date or datetime column, you can use the date picker for the comparison value. Data can also be cleaned and trimmed, and you can add a suffix or prefix. You can extract a substring from a column value to create a new column, and parse JSON, and XML column data into columns.

You use the **From Number** group to apply statistical, standard, and scientific functions to numerical columns. Statistical functions include aggregations such as sum, minimum, maximum, and average; you can also count values and distinct values. Standard functions you can apply include add, subtract, multiply, and divide. The scientific functions include absolute value, square root, exponent, and factorial. You can also apply trigonometry, rounding, and information such as **Is Even**, or **Is Odd**.

The **From Date & Time** group offers useful functions for extracting dates, times, and durations from existing datetime columns. You create a new column by extracting the year, month, day, or quarter from a value, and compare two columns to extract the date or time difference.

View

With the **View** tab, you can show or hide the QUERY SETTINGS pane. Settings include the name of the query, or table, and the list of steps, which are the transformations performed on the query. From here, you can also open the Advanced Editor window to view and edit the query code. You can also show or hide the Formula Bar—and toggle to display the data as monospaced—and decide whether to show or hide white space.

Note: When you click a column in the Power Query Editor, Power BI determines the data type from the values, and enables the relevant features on the tabs, so you can only apply formatting to columns with applicable data. You use **Determine Data Type** to run automatic data type detection against select columns.

APPLIED STEPS

When you shape your data using Power Query Editor, Power BI saves a list of the transformations you applied to your data, such as rename a column, delete a column, or change a data type. This list is displayed in the APPLIED STEPS section of the QUERY SETTINGS pane. Each time you connect to the data source to run the query, Power Query Editor applies these steps to the data, so it is always presented exactly how you shaped it.

- The Query Editor records all transformations to a query in the APPLIED STEPS setting:
  - All transformation steps are listed in order of creation; Source is first, followed by Navigation if applicable
  - Source contains data source connection information, and Navigation includes select tables and views
  - Can reorder steps if no dependencies exist
  - Can delete steps, but be aware of dependencies
  - Can undo steps, rolling back a previous step
  - Can rename steps

Note: When you shape the data in Power BI using the Power Query Editor, you are only amending the query, and reflecting these changes in the data that has been imported. The data in the data source remains unaffected.

The steps order

The Power Query Editor saves each change you make sequentially in date order, so the first transformation you made will always be applied to the data first. If you share a query with a colleague, these steps are included as part of the query, so the shaping can be applied again and your colleague sees the data exactly as you have specified. The **Source** step is first in the list, followed by **Navigation** if you selected the data from a list of tables or views. The **Source** step is the data source you connected to, and **Navigation** is the table you selected from that source. You cannot delete the **Source** step, but you can delete the **Navigation** step. However, this breaks any following steps.

If you click the gear icon to the right of **Source**, this opens the connection dialog box. For example, if your connection is to an Azure SQL Server database, click **Source** to open the connection dialog box. This shows the server name and database, and the code if you entered a query. If you included a query at this stage, no **Navigation** step is included. However, if you did not include a query, and instead selected from a list of available tables and views, you can use **Navigation** to change the source table or view.

Reordering steps

You can change the order of the steps in the APPLIED STEPS list, but you must be careful that this does not break the query. For example, if you move the step **Renamed Columns** above **Navigation**, you will break the query, so be aware of the dependencies between the steps. To move a step, right-click the step in the APPLIED STEPS section, and choose **Move Up**, or **Move Down**, or drag the step.

Deleting steps

You can delete steps in the list, effectively rolling back and undoing an action, but only if there are no later dependencies on the steps. If the step is isolated and has no later transformations that are dependent on the previous step, you can probably delete it. This is a useful—and fast—method for undoing transformations, such as removing a column that you later realize you need to include. To remove a step, click the **X** to the left of it.

Undoing steps

In addition to undoing deleted columns, if you hide a column using the **Choose Column** option, you can click the gear icon to the right of a step, and it opens the list box that was used for selecting the column you wanted to hide. You then select or clear columns.

Renaming steps

Each step in the list is given a generic name, such as **Removed Column**, **Removed Other Columns1**, **Removed Duplicates**. This is not helpful if you have a long list of steps and want to go back and adjust the order, or roll back a step. However, you can rename a step. In the APPLIED STEPS pane, right-click a step, and click **Rename**. Type in the name of the step; for example, **CustomerID renamed Customer Code**. Providing sensible names for your steps helps you make future amendments, and assists colleagues with whom you share queries, as they see which transformations are applied.

Best Practice: Providing sensible names for the steps in your queries helps if you return to the data after a long time, and have forgotten exactly what transformations are applied. This is particularly helpful if you want to stop halfway through shaping your data, and return later. You can see the list of transformations, and pick up from where you finished. This will be helpful if you share the query with colleagues.

You can also add a description to each of the steps. Right-click a step and click **Properties** to open the **Step Properties** dialog. In the **Description** box, type the description of the transformation, and click **OK**. When you hover your mouse over the steps in the APPLIED STEPS pane, a tooltip displays the name of the step and the description.

The Advanced Editor

You can use the Advanced Editor to see the query that is run against the data source. The query is written in M, the Power Query Formula Language. To view the query code from Power BI Desktop, on the **Home** tab, click **Advanced Editor**.

- With the Advanced Editor, you can see the query that Power BI runs against the data source to import the data:
  - Query is written in M Power Query Formula Language
  - To view, on the Home tab, click Advanced Editor
  - The query includes the connection, and connection type; for example, Excel or SQL Database
  - All transformations you apply to your data using Advanced Editor are added to the query code
  - The list of steps are reflected in the query, and in the same order
  - You can edit the query, but use syntax checker

The following code connects to an Azure SQL Database, and returns all columns and rows in the SalesLT.SalesOrderDetail table, without any filtering applied:

Advanced Editor Query to Return Unfiltered Data from the SalesLT.SalesOrderDetail Table

```
let
    Source = Sql.Database("sqlazure.database.windows.net", "AdventureWorksLT"),
    SalesLT_SalesOrderDetail = Source{[Schema="SalesLT",Item="SalesOrderDetail"]}[Data]
in
    SalesLT_SalesOrderDetail
```

When you make transformations to your data in the Power Query Editor, the steps are saved to the APPLIED STEPS in the QUERY SETTINGS. These steps are also applied to the code in the Advanced Editor. For example, the following code shows the steps that have been applied to the SalesOrderDetail query. First, the SalesOrderDetailID column was removed, and then the OrderQty column was renamed Order Quantity. Finally, the rowguid and ModifiedDate columns were removed.

The following code shows the connection to the AdventureWorksLT database on Azure, with filtering applied using the Power Query Editor:

Advanced Editor Query to Return Filtered Data from the SalesLT.SalesOrderDetail Table

```
let
    Source = Sql.Database("sqlazure.database.windows.net", "AdventureWorksLT"),
    SalesLT_SalesOrderDetail = Source{[Schema="SalesLT",Item="SalesOrderDetail"]}[Data],
    #"Removed SalesOrderDetailID" =
Table.RemoveColumns(SalesLT_SalesOrderDetail,{"SalesOrderDetailID"}),
    #"Rename OrderQty" = Table.RenameColumns(#"Removed SalesOrderDetailID",{{"OrderQty",
"Order Quantity"}}),
    #"Removed rowguid and ModifiedDate" = Table.RemoveColumns(#"Rename
OrderQty",{"rowguid", "ModifiedDate"})
in
    #"Removed rowguid and ModifiedDate"
```

The transformations in the code reflect the order in the APPLIED STEPS—these must be in the correct order when run against the data source. You can alter the code in the Advanced Editor, but you should use the syntax checker to ensure you do not break the code.

Demonstration: Using APPLIED STEPS

In this demonstration, you will see how to:

- Add transformations to a query, and see the steps in APPLIED STEPS.
- Rename steps in the APPLIED STEPS list.
- See the steps reflected in Advanced Editor.
- Delete steps, and change the source table in the Navigation step.

Check Your Knowledge

Question	
Which of the following statements about APPLIED STEPS is false?	
Select the correct answer.	
	Steps are added in sequential order.
	You can rename the steps.
	The Source step is always the first step.
	The Navigation step only shows if you have selected tables or views from the data source, instead of using a query.
	You can move a step between the Source step, and the Navigation step.

Lesson 2
Shaping data

This lesson explores the powerful features in Power BI with which you can shape your data ready for use in reports. You will learn how to shape your data, by applying formatting to make your data better for Power BI to handle in visuals, and how to apply transformations.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe how data shaping makes your data easier for reporting.
- Format your data so that Power BI manages it correctly within charting visuals.
- Transform your data using techniques such as adding new columns and changing data types.

What is shaping data?

Shaping data is the process of transforming data, so the Power Query Editor loads and presents it in the best way for your reports. The original data source remains unchanged, because it is just the view in Power BI that you are adjusting. Each of the transformation steps is captured in the APPLIED STEPS under QUERY SETTINGS. You use the Power Query Editor to shape your data, using features such as adding or removing columns, renaming columns, combining data, changing data types, transposing columns, and applying functions. Ideally, you want to shape your data before working with it in visuals. The most common data shaping techniques are described here.

- Shaping data is the process of transforming and formatting data for best presentation in reports:
  - The original data in the source remains unchanged
  - Each shaping step is recorded in the APPLIED STEPS list
- When shaping data:
  - Remove columns and rows that are not needed
  - Rename columns using an obvious naming convention
  - Ensure columns have the correct data types
  - Use date and time functions to create new columns
  - Add columns, and indexes useful for appending data
  - Apply a sort order, or use an index to guarantee order

Removing columns and rows

You should always remove data that isn't required. The dataset should be as succinct as possible, so you do not have redundant data that is loaded unnecessarily. If you have a large dataset, remove everything that isn't required to make it as small as possible and improve the performance of handling the data in Power BI. This means less data is transferred from the source to Power BI; there is less data to be processed as the Power Query Editor applies the transformations; and you have less extraneous data when creating reports.

Renaming columns

Your columns should have names that make it easy to work with them when creating reports and viewing dashboards. Each column name should give the data in that column an adequate description. This is particularly relevant when working with datasets containing several tables and columns—it makes it easier to find the right fields to add to report visuals. Power BI Q&A, which uses the natural query language, also returns more accurate results if it can find the data needed to answer the question it's being asked.

Data types

The Power Query Editor makes a best guess at the data type of each column when loading in the data. It is a good idea to check the given column types are as you would expect, and then format any that are incorrect. This can be critically important for decimal columns, where changing the data types between a

decimal and a whole number could potentially give false results in calculations. Currency columns that don't contain a currency symbol in the source are not typed as currency, so checking these columns and formatting them as your local currency will give better results in aggregations—in addition to formatting the data so it presents better in data labels.

Datetime columns

You should also format datetime columns with the correct data type. You use the Date and Time transformations to add additional columns to extract the year, quarter, month, week, and day from a date column, and hours, minutes, and seconds from a time column. You calculate the difference between two data columns to create a new column. For example, you could subtract the **Delivery Date** from the **Order Date** to create a **Days to Fulfill** column, showing how long it took to deliver an order after it had been placed. If you have a **Date of Birth** column, you can use the **Age** function to create a new column for the person's current age.

Adding columns

There are many options and functions to help you create new columns in your queries. You can duplicate an existing column, split a column into multiple columns, use the data and time functions described previously, or concatenate values in multiple columns to create **Full Name** or **Full Address** fields. You can also use math functions to create calculated columns; for example, to subtract **Manufacturing Costs** from **Retail Price** to create **Profit**. You merge data from a date column with data from a time column to create a new datetime column.

Adding indexes

You can add indexes to your tables, with the seed value starting at 1 or 0, or you can create a custom index by defining the start number and the increment. If you combine data from different systems, you may find that there are overlaps in the index key columns, meaning you wouldn't have unique values when merged together. In the Power Query Editor, you can add an index as a surrogate key column to the two tables you are appending, so the index value is always unique.

Apply a sort order

By default, Power BI sorts data in alphabetical order in visualizations. While this may be desirable in some instances, you might want to order by a month column or by another categorization. The Home tab in the Power Query Editor includes a Sort group, with which you can sort A-Z, or Z-A. These may not fulfill your criteria, in which case you can add an index column and use this to sort by in the visualizations.

Formatting data

By formatting your data, you help Power BI categorize and identify the data, making it much easier to work with. In Power Query Editor, you apply string functions to your text columns to create consistency, ensuring that data is well presented.

- Power Query Editor provides many options for creating columns, formatting text, and numbers:
  - General Group:
    - Add custom columns using formulas or duplicate columns
    - Add an index column and move to the front of the table
  - From Text
    - String functions include lowercase, UPPERCASE, Capitalize Each Word, Trim, Clean, Add Prefix, and Add Suffix
    - Merge columns using optional character or space separator
  - From Numbers
    - Add, Multiply, Subtract, Divide columns, or calculate by value
- All formatting uses a query that you can view in the Formula Bar or in Advanced Editor

General formatting

The **Custom Column** button on the **Add Column** ribbon enables you to enter a custom formula to create a new column, including calculations using values from other columns. The syntax checker indicates when you have an

error and does not allow you to save a formula with errors. To create a new column, click **Custom Column**. In the **New column name** box, type the name of the column, and add your formula to the **Custom column formula** box; for example, **[ShipDate] - [OrderDate]**. To avoid errors in the column names, you select a column in the **Available columns** list and click **Insert** to add it to the Custom column formula text box. When your formula is complete, click **OK**. The new column is appended to the end of the table and the formula is visible in the Formula Bar. If you open Advanced Editor, you will see the formula appended to the query.

The following example shows the code in the Formula Bar, which subtracts the **OrderDate** from the **ShipDate** to return the number of days as the new column **DaysOrderToShip**.

The following code is the formula to create a custom column, which calculates the days from when the order was placed to when it shipped:

Custom Column Formula

```
= Table.AddColumn(#"Sorted Rows", "DaysOrderToShip", each [ShipDate] - [OrderDate])
```

The column is created with a data type of Any, and values are in the format of 7:00:00:00. To change the type, right-click the column, click **Change Type**, and then select the appropriate type. In this example, you could convert the data to a Whole Number type.

You can also add an index column to the end of your table. You can start the index at 1 or 0, or choose the starting value. By default, the index increments by 1, but you can change this using the custom index option. To add an index, click the down arrow on the **Index Column** button on the **Add Column** ribbon. Click **From 0** or **From 1** to immediately add an index using the chosen starting value. If you click **Custom**, this opens the **Add Index Column** dialog box. In the **Starting Index** box, type your starting number; for example, 100. In the **Increment** box, type the number you want the index to increase by with each row; for example, 10. The index in this case would be 100, 110, 120, 130, and so on. It is common practice to have your index as the first column in the table, so right-click the new index column, point to **Move**, and then click **To Beginning**. You can also select multiple columns to move them together.

The **Duplicate Column** function is useful when you have a string column that will be split, but you want to retain the original value. You click to select the column and choose **Duplicate Column** from the **General** group or right-click and select **Duplicate Column**. The new column is appended to the end of the table, and given a name such as **SalesOrderNumber - Copy**. You then work with this column to split the values, or perform other operations, such as replacing substrings or trimming repeated characters.

Formatting text

The **From Text** group functions provide options for formatting string values, merging columns, extracting values, and parsing to other formats. The **Format** function converts strings to **lowercase**, **UPPERCASE**, **Capitalize Each Word**, **Trim**, **Clean**, **Add Prefix**, and **Add Suffix**. You can use these to convert your string data into consistent formats, which is particularly helpful when importing raw data that has not been cleaned. If you import data from an e-commerce website, where customers have entered their names and addresses, and no formatting was applied before the data was saved to the database, it is likely to be inconsistent—with mixed casing across the various fields. You can use **Capitalize Each Word** so the columns all have the correct casing, and apply **UPPERCASE** to state codes, such as MA, NJ, WA, or country or area names depending on your reporting requirements.

You can create a new column by merging two or more columns together. To do this, click to select a column, and hold down the Ctrl key and click the other columns you want to merge. On the **From Text** group, click **Merge Columns**. In the **Merge Columns** dialog box, choose how you want the values to be separated, from **Colon**, **Comma**, **Equals Sign**, **Semicolon**, **Space**, **Tab**, or **Custom**. For Custom, enter the symbol or character, such as a dash. In the **New column name (optional)** box, give your column an appropriate name, and click **OK**.

Note: The column values are concatenated in the order in which you click the columns to select them. This gives you full control over the end result.

Best Practice: The Merge Columns function can be used on address fields to quickly create a full address column. Highlight your address columns in order, and click **Merge Columns**. For the separator, choose **Custom**, and enter ", " (comma and a space). This concatenates all the values together in a comma separated list.
However, you are likely to have null values or empty strings in some columns, perhaps Address2, which results in double commas. You then use the **Replace Values** function on the **Any Column** group of the **Transform** tab, to replace ", ,", with ", ".

You use the **Extract** function to copy a substring value from one column, to create a new column. You also use **Extract** to count the length of a string. Select a column in your table, and click **Extract** from the **From Text** group. Click **Length** to create a new column that counts the number of characters in the column. Spaces are included in the length. To extract a fixed number of characters from the beginning or the end of the column value, use **First Characters**, or **Last Characters**. Select the column and click **Extract**, and then either **First Characters**, or **Last Characters**. Enter a value for the **Count**, and click **OK**. This is useful if you want to split a **PostalCode** column to extract the first few characters to create a map based on area, rather than exact postal code. To extract a specific number of characters from the middle of a string, you use **Range**. Select the column, and click **Extract**, **Range**. Provide a **Starting Index**, and a **Number of Characters** value, and click **OK**. If you type 2 for the Starting Index, the extract starts on the third character.

The **Parse** function takes a column that is an XML or JSON format, and parses it into a table. Select the column with your XML or JSON data, and click **Parse** from the **From Text** group. Select **XML**, or **JSON**, and the Power Query Editor adds a table column to the current table. Click the double-arrow icon to expand the new table, and choose the attributes you want to include in the table. This is a very quick way to parse and extract data provided in XML or JSON format.

Formatting numbers

There is a wide range of formatting functions that you can apply to your numeric columns. The **From Number** group includes functions for **Statistics**, **Standard**, **Scientific**, **Trigonometry**, **Rounding**, and **Information**. This section focuses on the more common standard number functions. Choose from **Add**, **Multiply**, **Subtract**, **Divide**, **Divide (Integer)**, **Modulo**, or **Percentage**.

Add

To add two or more columns together, click the first column, hold the Ctrl key, and click the other columns. On the **From Number** group, click **Standard**, **Add**. This creates a new column with a default name of **Inserted Addition**. You can also add a whole or decimal number to a column. Select a single column, and then click **Standard**, **Add**. Enter the number that you want to be added to the existing column value.

Multiply

If you want to multiply two or more columns together, click the first column, hold the Ctrl key, and click the other columns. On the **From Number** group, click **Standard**, **Multiply**. This creates a new column with a default name, **Inserted Multiplication**. To multiply a column by a whole or decimal number, select a single column, and then click **Standard**, **Multiply**. Type the number that you want the existing column value to be multiplied by. For example, to calculate a net value to include 20 percent tax, click the **net** column, click **Standard**, **Multiply**, and type **1.2**. This creates a column with the additional tax amount.

Subtract

Subtract works in much the same way as the Add and Multiply functions; however, you can only use two columns in the calculation. Select the first column you want to use in the calculation, and then click the column to subtract from the first column, and click **Standard**, **Subtract**. This creates a new column named **Inserted Subtraction** by default. If you wanted to use more than two columns, you could use a custom column, with a formula such as **[RetailPrice] - [ManufacturingCost] - [StoreCommission]**. You can also select a single column, click **Standard**, **Subtract**, and then enter a whole or decimal number.

Note: The order in which you select your columns affects the calculation. For example, if you want to calculate **Profit**, click **RetailPrice**, click **ManufacturingCost**, and then click **Standard**, **Subtract**. The calculation is displayed in the Formula Bar, so if you have incorrectly ordered the columns, you can manually change the query. In this case, the query **Table.AddColumn(#"Changed Type", "Profit", each [ManufacturingCost] - [RetailPrice], type number)** is incorrect, because the ManufacturingCost should be subtracted from the RetailPrice.

Divide

The Divide function can also only operate on two columns. You should be aware of the order in which you select them, because this affects the calculation. Select the first column for the calculation, hold down the Ctrl key, and then click the second column for the number to divide by. Click **Standard**, **Divide**—a new column is created and by default is named **Inserted Division**. This returns a whole or decimal value. You can divide a single column by a specific value. Click the column, and then click **Standard**, **Divide**, and enter a whole or decimal number. Click **OK**.

Transforming data

While Power BI is flexible in the variety of data sources that you can import data from, visualizations work best with data that is in a columnar format. For example, data that is imported from Excel may be easy for the human eye to digest visually, but the data might not be structurally appropriate for Power BI to translate the values in a bar chart. The Power Query Editor offers plenty of functions for you to transform data into a structure that Power BI can use effectively in reports. This lesson explores the functions available on the **Transformations** tab.

- Table group:
  - Use Group By to apply aggregations on your table
  - Use First Rows As Headers and use Headers As First Row
  - Transpose to treat columns as rows, and rows as columns
  - Reverse Rows to reverse the order of the data
- Any Column group:
  - Change or detect data types
  - Replace Values and Replace Errors
  - Fill null values in a column
  - Pivot Column and Unpivot Columns
  - Move columns
- Text Column group:
  - Split single column in multiple columns

Table

The Table group offers some useful functions that you can quickly use to transform your data. Each of the functions is described next.

Group By

You can aggregate one or more columns in your table. Click **Group By** on the **Table** group, and select the columns you want to include. Ensure you include all columns that you want to include in the table, or they will be removed. In the **New column name** box, give the column a useful name, and choose the

Operation from **Sum**, **Average**, **Median**, **Min**, **Max**, **Count Rows**, **Count Distinct Rows**, or **All Rows**. If the operation such as Sum requires a column to aggregate, select this from the list, and then click **OK**.

Use First Row As Headers

This function is useful when data has been imported and it already has a header row, but Power BI has not detected this. If you import columns with numeric values that include a header, Power BI can detect that the first row is a string compared to the other values—and guess that it is the header. This is not so obvious when all of the columns contain string values. To apply this function, click **Use First Row As Headers**, and select **Use First Row As Headers**. The values of the first row are promoted to the column header. You can also perform this operation in reverse. Click **Use First Row As Headers**, and select **Use Headers As First Row**. The headers become the first row in your table, and you can then rename the columns.

Transpose

With Transpose, you can treat rows as columns, and columns as rows. This is useful if you import a table from a spreadsheet with columns and rows that are readable to the user in a matrix format, but don't translate into a format that Power BI can use easily. Select the table that you want to apply this function to, and then click **Transpose** from the **Table** group. You can then begin applying other functions, such as unpivot, to give your data a columnar format.

Reverse rows

This function reverses the order of the rows in the table, so that the bottom rows are at the top, and the top rows are at the bottom.

Count rows

Use this function to return the number of rows in the current table. The rows are replaced with the count of rows.

Any Column

The functions in the **Any Column** group can be applied to columns, regardless of the data type or format. Each of the functions is described next.

Data Type

You can use the **Data Type** function to select from a list of data types—this is useful for converting columns where Power BI has incorrectly guessed the type. Select one or more columns in the table, click **Data Type**, and then select the data type for your conversion. Types include **Decimal Number**, **Fixed Decimal Number**, **Whole Number**, **Date/Time**, **Date**, **Time**, **Date/Time/Timezone**, **Duration**, **Text**, **True/False**, and **Binary**.

Detect Data Type

You can select one or more columns and use the built-in data type detection function. Select a column, hold down the Ctrl key, and then click any additional columns you want to add. From the **Any Column** group, select **Detect Data Type**. Power BI automatically corrects any columns it guesses to be wrong.

Rename columns

To rename a column, select the column in the table, and then click **Rename** from the **Any Column** group, or you can right-click the column and then click **Rename**. Type in the new name of the column, and press Enter—the name is updated.

Replace Values and Replace Errors

With these two functions, you can very quickly replace a value or an error in a column with another value. Both functions work on one or more columns, so select a single column, or hold down the Ctrl key to click

and select multiple columns. Click **Replace Values** from the **Any Column** group, and in the dialog box, type a **Value To Find**, and a value to **Replace With**. If you are working with a text column, you can also click **Advanced Options** to **Match entire cell contents** and/or **Replace using special characters**. Special characters include **Tab**, **Carriage Return**, **Line Feed**, and **Carriage Return and Line Feed**. Click **OK**. The values in the column are replaced. To replace an error, click **Replace Errors** instead of Replace Values, and type a replacement value in the **Value** box of the **Replace Errors** dialog box.

Fill

You can use the fill function to fill in null values with the value from an adjacent cell. Click the cell you want to use to fill the adjacent cells, and click **Fill**, and then click **Up** or **Down**, depending on which direction you want to fill. This works at the column level.

Pivot Column and Unpivot Columns

The **Pivot Column** function takes the values in the selected column, and uses them to create new columns. This is particularly helpful if you import data that has a matrix format from Excel, and you want to convert it to a columnar format for reporting. **Unpivot** can also help with this, by converting selected columns into attribute-value pairs.

Move

The **Move** function moves one or more columns to another location in the table. Click the column you want to move, or hold down the Ctrl key to select multiple columns, and click **Move** from the **Any Column** group, or right-click. You can move **Left**, **Right**, **To Beginning**, or **To End**. The **To Beginning** option is useful if you add in an index column, because this is always appended to the right.

Text Column

The **Split Column** function splits a column based on a delimiter, or a specified number of characters. Much like the **Extract** function discussed in the previous topic, you can select from the list of delimiters, or use a custom delimiter. To split a column, select the column in the table, and from the **Text Column** group, click **Split Column**. Click **By Delimiter**, to open the **Split Column by Delimiter** dialog box. You can select a delimiter from **Colon**, **Comma**, **Equals Sign**, **Semicolon**, **Space**, **Tab**, or **Custom**. To use a custom character, click **Custom**, and enter the character, or symbol, such as a hyphen. You can split at the **Left-most delimiter**, **Right-most delimiter**, or **Each occurrence of the delimiter**. The number of new columns created will depend on which split option you choose.

To have further control over the split, click **Advanced options**. You specify the number of columns to split into, and the **Quote Style**, which is **CSV**, or **None**. You can also split using special characters, and choose from **Tab**, **Carriage Return**, **Line Feed**, and **Carriage Return and Line Feed**. Click **OK**. The column splits the values, and the original column is replaced. Use the **Duplicate Column** function on the **Add Column** tab if you want to retain this value.

Demonstration: Transforming data with the Power Query Editor

In this demonstration, you will see how to:

- Import data from Excel.
- Apply transformations to the table.

Check Your Knowledge

Question	
Which of the following is not good advice for shaping your data?	
Select the correct answer.	
	Remove all columns and rows that are not used in the reports.
	Rename columns to provide names that represent the column data, and can be used by Power BI Q&A.
	Let Power BI guess the data types of your columns because it will always be correct.
	Create an index column if you want to guarantee the sort order in a visual, or if you are appending data.
	Use the Age function on a Date of Birth column to calculate the current age.

Lesson 3
Combining data

In this lesson, you will learn how to import data using an internet address as a data source, how to shape that data, and how to merge it with existing data in your dataset.

Lesson Objectives

After completing this lesson, you will be able to:

- Import data into your dataset from the internet.
- Apply shaping to data you have imported from the internet.
- Merge data from different tables within your dataset.

Adding data from the internet

Power BI offers great flexibility for importing data. You use the web data source to pass a URL to Power BI so it can scrape the data into a new table. Data in the webpage you want to scrape should be in a tabular layout, so Power BI can determine the shape and import the data into a table structure. This is a useful way to import publicly available data, such as government statistics, or information gathered by organizations such as those monitoring climate change, or population socio-economics—you can combine this with your existing data to show trends or demographics.

- Import data from a website that provides data in a tabular structure:
  - Use publicly available datasets, and combine this with your existing data for reporting insights
  - Import using **Get Data**, **Web**, and enter the URL
  - Power BI establishes a connection, and imports the data
  - Use the data just as you would from any other source
  - Preview the table structures that Power BI has detected
  - Load data, or edit in Power Query Editor; data can be refreshed
  - Shape and transform the data as required
- Be aware that the source data could be removed

Importing data

To import data from a webpage, open Power BI Desktop. From the **Getting Started** dialog box, or from the **Home** tab, click **Get Data**, and select **Web**. In the **From Web** dialog box, type or paste the web address into the **URL** box, and click **OK**. Power BI establishes a connection to the webpage, and determines the data that can be imported. In the **Navigator** dialog box, you are presented with a list of tables for the data that can be imported. You select the tables and preview them as you would any other data source. Click **Edit** to load the data into Power Query Editor and begin shaping the data. Alternatively, you can click **Load** to import the data into Power BI designer, where you can use it immediately in visualizations, or later apply transformations and shaping.

Note: Public websites such as Wikipedia offer a wealth of information that you can freely use in your reporting. However, you should be aware that you have no control over when the data is updated, whether or not it is accurate, or even if the page or data is retained or removed.

Shaping the new data

- After importing data from the internet, use shaping and transforming to format and correct
  - All shaping is stored as steps, so will be reapplied each time the query is run, and data can be refreshed
  - Use the data as you would from any other data source
  - Remove columns that you won't use in reporting
  - Ensure the query and columns have names that reflect the content, and are obvious to users and Q&A
  - Make sure columns have the correct data type
  - Apply a sort order if required

When you import data from the internet, it is unlikely that you will know how the data will initially be shaped until Power BI has established a connection to the page, and determined the format and possible tables that it can scrape. If you regularly import or refresh from the same source, and this doesn't change, you can have some confidence in the end results. However, after first importing internet data, you are very likely to want to perform some shaping and transformations. The transformations that you apply to the data are stored in the QUERY SETTINGS under APPLIED STEPS, so each time you refresh the data, the query includes the code to shape and transform the data from the web—you should always see the results you expect.

Shaping the data

You can shape data from the internet exactly as you would with data from any other data source. As with any dataset, it is a good idea to remove the columns that will not be used in your reports and analysis, keeping the data succinct, and more efficient to work with. This reduces the size of the data, and does not present extraneous columns to colleagues who might share the query.

It is also important to ensure the name of the query (table) and the names of the columns are something obvious. Again, this keeps clarity within the dataset, and has the added advantage that you or other colleagues can understand the data just by looking at the names of the queries and columns. Furthermore, Power BI Q&A uses the natural query language and relies on being able to find answers to questions, based on relevant column names. The names should accurately describe the data.

When importing the data, the Power Query Editor makes a best guess at the data types for each of the columns. You will want to check the columns to ensure the type matches the data. Check that datetime columns have been detected correctly, especially if you want to use dates and times for drilldown. The Power Query Editor does not always recognize currencies unless there is a symbol included in the data—you should update any currency columns. Check that numerical columns have the correct data types, and include whole numbers and decimals, which you require for aggregations.

If the data needs a particular sort order, you can set this to be A-Z, or Z-A, or add the month number to a query that includes month in text format—so you can order on the numeric value in your visualizations.

Merging data

By using Power BI, you can gather data from different sources and of different types into a single dataset. You then combine the data in one report. You can import data from SaaS providers such as Bing and Salesforce, combine it with data from your Azure SQL Database in the cloud, an on-premises SQL Server Analysis Services (SSAS) data warehouse, and data from Excel. After importing into a single set of data, you can merge columns using tables from different sources and append rows.

- Merge columns:
  - Merge one table into another table, using a joining column
  - Choose from join types
  - All columns are initially merged, but use the selector to choose which columns you want to keep
  - Can retain original column names
- Append rows:
  - Adds rows from one or more tables to another table
  - Column data does not have to match
  - Mismatching can result in unclean data and/or nulls
  - Add index to combined table

Merging columns

To merge columns, the two tables must have a joining column, where the value will match the order to combine the values. From the Power BI Desktop designer, click **Edit Queries** to open the Power Query Editor window. Click the query (or table) into which you want to merge the other columns. From the **Home** tab, select **Merge Queries** from the **Combine** group. This opens the **Merge** dialog box. The top table is the one you elected as the destination table for merging the second table. Click to choose the column where you want to join. You select more than one column by holding the Ctrl key down while using your mouse to select. In the list, select the table from where you want to merge. In the second table, click to select the column, or columns, you are joining. The label at the bottom of the dialog box counts the matches, so you can usually determine if the match is correct. For example, if you are expecting all rows to match, and the label says, "The selection has matched 36 out of the first 48 rows", then something is wrong.

You choose the type of join used to connect the two tables, by selecting from the **Join Kind** list. Types of join include **Left Outer** (all from first, matching from second), **Right Outer** (all from second, matching from first), **Full Outer** (all rows from both), **Inner** (only matching rows), **Left Anti** (rows only in first), or **Right Anti** (rows only in second). Use the default **Left Outer**, or select another join, and click **OK**. The second query is merged as a single column, with a value of **Table**. Click the double-arrow icon in the column header, and select the columns you want to include from the second table. You might not want to include the joining column, if all your rows matched or partially matched as expected. Clear the **Use original column name as prefix** if you want the columns to keep their original names, otherwise the column is named **NewColumn.<original name>**. After making your selection, click **OK**. The second table columns now appear as columns in the first table—though you may need to rename them.

Appending rows

When you append rows, you take rows from one or more tables, and add them to the first table. In most situations, the columns and data types will match. However, you can append rows between two tables that have all different columns—but the result is unclean data and no values when the number of columns between the tables does not match. From Power BI Desktop designer, click **Edit Queries** to open the Power Query Editor window. Click the query (table) into which you want to append the rows, and click **Append Queries** from the **Combine** group on the **Home** tab. This opens the **Append** dialog box. From the **Select the table to append** list, choose the table you want to add in, and then toggle **Two tables**, or **Three or more tables**. If you are appending two tables, click **OK**. If you have clicked **Three or more tables**, in the **Available Table(s) list**, select each table you want to append, and click **Add**. You can append a table to itself if you need to. Click **OK**.

Best Practice: If you are appending rows from multiple sources, and the table contains index values that overlap when the data is combined, combine the data, and then create a new index column on the table into which the rows have been appended.

Demonstration: Adding and shaping data from the internet

In this demonstration, you will see how to:

- Import data from the internet.
- Shape the data that is imported.

Check Your Knowledge

Question	
Which of the following is not a true join type for merging columns?	
Select the correct answer.	
	Left Outer (all from first, matching from second).
	Right Outer (all from second, matching from first).
	Full Outer (all rows from both).
	Inner (matching rows only).
	Random (let Power BI decide).

Lab: Shaping and combining data

Scenario

Adventure Works employees are becoming increasingly frustrated by the time it takes to implement managed BI services. The existing managed BI infrastructure, including a data warehouse, enterprise data models, and reports and dashboards, are valued sources of decision-making information. However, users increasingly want to explore relationships with other, currently unmanaged data. It takes too long for the IT department to include these requirements in the corporate BI solution.

As a BI professional, you have been asked to explore ways in which Adventure Works can empower business users to augment their managed enterprise BI solution with self-service BI.

Objectives

After completing this lab, you will be able to:

- Connect to a SQL Server database and import data.
- Apply formatting to the data you have imported to shape it, ready for reporting.
- Combine related data to the shaped data.

Note: Because of updates to Microsoft Power BI, the lab steps for this course change frequently. Microsoft Learning regularly updates the lab steps, so they are not available in this manual – but you can access them on GitHub.

Lab Setup

Estimated Time: 60 minutes

Virtual machine: **20778C-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa55w.rd**

All the lab steps are contained in 20778C_LAB_04.md.

Exercise 1: Shape Power BI data

Scenario

You are exploring how Power BI can help shape and combine data that comes from multiple sources. Currently, much of the data is exported from SQL Server into Excel. You have two worksheets, one for sample sales data for the North America territory, and one for the European territory. After importing the data into Power BI, you will shape the data using transformations and formatting.

The main tasks for this exercise are as follows:

1. Preparing the environment

2. Import data from Excel

3. Apply formatting to the existing data

▶ **Task 1: Preparing the environment**

▶ **Task 2: Import data from Excel**

▶ **Task 3: Apply formatting to the existing data**

Results: At the end of this exercise, the data will be imported from Excel, and shaped ready to be combined.

Exercise 2: Combine Power BI data

Scenario

You have imported the two worksheets for sales in Europe and North America, and applied some shaping. You now want to combine the rows from the North America query, into the Europe query. You also want to include a Country Code column.

The main tasks for this exercise are as follows:

1. Add related data to the shaped data

▶ **Task 1: Add related data to the shaped data**

Results: At the end of this exercise, the Europe and North America data will be appended, and the Country Code column will be added to the query.

Question: Discuss the types of different data in your organization that could be combined using the Power Query Editor. Do you have data stored across locations that could be appended, or lookup data that could be merged into other tables to make it more useful for reporting?

Module Review and Takeaways

In this module, you have learned how to:

- Perform a range of query editing skills in Power BI.
- Shape data, using formatting and transformations.
- Combine data together from tables in your dataset.

Review Question(s)

Question: Discuss the benefits of using Power BI, rather than Excel, to shape and transform your data. Are there any disadvantages? What can Power BI do that Excel cannot, and vice versa? Which tool do you think is most straightforward to use?

Module 5

Modeling Data

Contents:

Module Overview

Microsoft® Power BI is making its mark in the self-service BI world—because it can quickly create visually stunning, interactive reports, and dashboards. Power BI provides a straightforward way to combine data from a wide range of sources into a single dataset, and work with that data to create cohesive reports. This module goes behind the scenes of the visualizations, and explores the techniques and features on offer to shape and enhance your data. With automatic relationship creation, a vast library of DAX functions, and the ability to add calculated columns, tables, and measures quickly, you will see how Power BI creates attractive reports, while helping you find hidden insights into data.

Objectives

At the end of this module, you will be able to:

- Describe relationships between data tables.
- Understand the DAX syntax and use DAX functions to enhance your dataset.
- Create calculated columns, calculated tables, and measures.

Lesson 1
Relationships

This lesson explores the relationships between the tables in your data, why they are important, and how to create them.

Lesson Objectives

At the end of this lesson, you will be able to:

- Describe the purpose of relationships between tables.
- View relationships in the Power BI Desktop Power Query Editor.
- Create relationships using the Power Query Editor.
- Understand cardinality.
- Choose the correct cross filter direction in your relationships.

What are relationships?

Relationships exist to join tables together so that you can work with them as if they were one. If you are familiar with relational databases, such as Microsoft SQL Server®, or data warehouse databases such as SQL Server Analysis Services (SSAS), you will understand the concept of relationships in Power BI, as this is much the same.

- Relationships join tables together so you can work with multiple tables as if they were one:
  - Usually created in an OLTP system as part of the normalization process, by adding keys to tables
  - Prevents repeated values, and each entity has only those attributes that belong to it
  - Data warehouse uses fact tables, with keys that join to dimension tables
  - Power BI Autodetect feature can recognize relationships, and creates them automatically

Relationships in an OLTP system

Relationships are usually created in an online transactional processing (OLTP), or relational database, as part of a normalization process. Normalization works at various levels, or forms, depending on how close to official normalization rules you want to adhere. Two of the main purposes of normalization are eliminating repeated data, and only including columns in a table, or entity, that are a direct attribute of that entity. For example, you would store your list of customers in a table with one row for each customer. Your Sales table would have a link back to the Customers table, using a key column such as CustomerID. This prevents you from repeating all the customer data, such as contact name, address, postcode, and so on, each time a customer places an order. Furthermore, when a customer updates their details, you only need to update one record, keeping your data consistent. The link from the Sales table to the Customer table using the CustomerID key is a relationship.

Relationships in a data warehouse

If you have worked with a data warehouse database, you know that a fact table connects to the dimension tables using keys. Although data stored in a star schema in a data warehouse is structured differently to data stored in an OLTP, or relational database system, the keys in both designs create relationships by joining tables together.

Table relationships in Power BI

The following table shows rows from the SalesOrderDetail table. Each row contains an order for a product. The SalesOrderID column values in this case are identical, so these rows are part of the same order. There is also a ProductID column in the table, linking the SalesOrderDetail table to the Product table.

SalesOrderID	ProductID	OrderQty	UnitPrice	LineTotal
43659	714	3	28.8404	86.521200
43659	716	1	8.8404	28.840400
43659	709	6	5.70	34.200000
43659	712	4	5.1865	10.373000
43659	711	2	20.1865	80.746000

The SalesOrderDetail table is related to the SalesOrderHeader table, shown next. There is one row in the SalesOrderHeader table for each order, though this order might comprise multiple rows in the SalesOrderDetail table. The CustomerID column links to the Customers table.

SalesOrderID	OrderDate	ShipDate	CustomerID
43659	2011-05-31 00:00:00.000	2011-06-07 00:00:00.000	29825

Traversing the tables using the relationships, the SalesOrderDetail table is related to the SalesOrderHeader table, and the Product table. In turn, the SalesOrderHeader table is related to the SalesOrderDetail table, in addition to the Customers table. You can use these relationships to view the four tables as one, so you see all the products ordered by a customer as if they are in one table. This is useful for aggregating data across tables in visualizations.

Autodetect feature

When you import data into Power BI, the Autodetect feature operates in the background, and works out the relationships in your dataset. It also automatically sets the cardinality and cross filter direction, both of which are covered as topics in a later lesson. For much of the time, Power BI makes a good guess, correctly identifying related tables and creating the relationships for you. In this scenario, you might not have to do any further work to establish relationships between the tables.

Viewing relationships

- Power BI Autodetect feature works out relationships in queries run against data source:
  - Relationships are created automatically after data load
  - Autodetect determines cardinality and cross filter direction in the relationship
  - View and edit relationships created by Power BI in the Model view, using a relationship diagram
  - When Power BI detects more than one relationship between two tables, only one can be active, and is set as the default; turn off incorrect active relationship
  - Delete relationships in the Model view

When you import data into Power BI, queries are run against the data source to copy the data required to fulfill your modeling requirements for the dataset. While these queries are running, Power BI observes them to determine if there are relationships between the tables. After the data has finished loading, you view and manage the relationships that Power BI has created for you.

Power BI Desktop comprises three main views: Report, Data, and Model. You view the tables and column names in Report view, and add fields to visualizations. In Data view, you apply extensive modeling and formatting to the data, and you view the values in the tables. The Model view shows the tables and columns linking together those tables that are related. In the views pane, click **Model** to open a diagrammatic view of the relationships in your model. The relationships appear the same in the Model view regardless of whether they have been created manually or by Power BI. All tables are included, even if they are not related to any others.

You can see information about the relationships, just by looking at the relationship diagram. Each relationship is represented by a line that joins the two tables. The arrow icon on the line indicates the cross filter direction of the relationship—either one arrow for **Single**, pointing in the direction of the filter, or two arrows when the cross filter direction is set to **Both**. At the end of each relationship line, where it joins to either table, another icon represents the cardinality. A star icon (*****) represents **Many**, and a **1** represents **One**, for either a Many to One (*:1), One to One (1:1), or One to Many (1:*) relationship. When you click a relationship line, the related columns in either table are highlighted with a black border, for quick identification.

Editing relationships

When a relationship line has focus, it is highlighted in yellow. Double-click the line to open the **Edit relationship** dialog box. You can also click **Manage Relationships** from the **Relationships** group on the **Home** tab, to view the **Manage relationships** dialog box. From the **Manage relationships** dialog box, you create new relationships, run the Autodetect feature, and then edit and delete existing relationships. In the **Manage relationships** dialog box, double-click a relationship to open the **Edit relationship** dialog box. This opens the same view as double-clicking a relationship line. In the **Edit relationship** dialog box, you can change the related table and column, switch cardinality between Many to One (*:1), One to One (1:1), or One to Many (1:*), and toggle the cross filter direction between **Single**, or **Both**. You can also turn on or off the **Make this relationship active** option. When the Power BI Autodetect feature runs, it sometimes finds more than one relationship between two tables. In this case, only one of the relationships is set to active, and this becomes the default relationship. You can use this setting when the active relationship is incorrect.

You can also delete relationships. Click the relationship line that joins two tables so it is highlighted yellow. Right-click the relationship line, and then click **Delete**.

Creating relationships

There are two ways to create relationships in Power BI. You use the Autodetect feature and Power BI works out the relationships for you, or you create them manually.

- Two ways to create relationships in Power BI:
  - Using Power BI Autodetect feature
  - Create relationship manually
- If you can't create a relationship, it's likely to be because of null, or empty values, or duplicate rows

Creating relationships using Autodetect

When data is imported into the model, Power BI automatically creates relationships. If you then create calculated tables, or use Enter Data to add new tables, relationships will not exist. You can run the Autodetect feature from the **Home** tab: in the **Relationships** group, click **Manage Relationships**, and then in the **Manage relationships** dialog box, click **Autodetect**. Power BI runs the Autodetect feature to look for new relationships and by default, also determines the cardinality, cross filter direction, and active relationships. However, bear in mind that the Autodetect feature is a best guess, and might need adjusting after it runs.

Creating relationships manually

The quickest way to create a relationship between two tables is to drag the column from the first table, to the related column in the second table where you want to join. If the data is valid for creating a relationship, the columns are connected. You can also click **Manage Relationships** from the **Relationships** group on the **Home** tab. This opens the **Manage relationships** dialog box. Click **New** to open the **Create relationship** dialog box. Select the first table in the upper table list—this displays a preview of the table. Click the column you want to use in the relationship, and then select the table to join to in the lower table list. This displays a preview of the second table. Again, click the column where you wish to join. Power BI automatically determines the cardinality and cross filter direction of the relationship. This is usually correct, so unless future data is likely to change this, click **OK** to create the relationship. Otherwise, change the cardinality and cross filter direction settings, and then click **OK**. Click **Close** to close the **Manage relationships** dialog box.

You might find that you cannot create a relationship between your tables. This might be due to columns with null, or empty, values or duplicate data. You remove rows with null or blank values by using the filter in the query tab, or replace them with valid data, including "NULL". Removing rows might affect calculations, yet using NULL can create artificial relationships. If you use the latter approach, make sure you include appropriate filters in your visualizations.

Cardinality

- Cardinality is the relationship between two tables
- Power BI supports three types:
  - Many to one (*:1):
  - One to one (1:1):
  - One to many (1:*):

In data modeling, cardinality refers to the relationship that one table has with another. In Power BI modeling, the cardinality can be one of the following three types:

1. **Many to one (*:1)**: Many to one is the default type in Power BI, and generally the most common. Many to one means that one table can have more than one instance of the value used in the column to join to the other table. The other table would have only one value. For example, your Sales table has many instances of the CustomerID because the customer has placed multiple orders, and joins to the Customers table using CustomerID. The Customers table has one instance of the CustomerID, or rather one row for each customer. This is also common for lookup tables, where you might have a list of states, or countries. Each state or country is listed only once, but the instance exists multiple times in the Customers (or other) table.
2. **One to one (1:1)**: In a one to one relationship, both tables in the relationship have one instance of a value. In relational database systems, one to one is not as common as many to one, but one of its uses can be to split up larger tables. For example, you might have an Employees table with an EmployeeID column and other columns for the employee name, address, date of birth, phone number, and salary. This data is used frequently by the Human Resources department. You have another table called EmployeeAdditionalDetails, with a row for each employee, and an EmployeeID column to join to, from Employees. The EmployeeAdditionalDetails table contains less used fields such as next of kin, number of dependents, training information, and qualifications. This would be a one to one relationship.
3. **One to many (1:*)**: This is the same as many to one, except the position of the tables is reversed in the relationship. In this case, you could have your Customers table with one row for each customer, related to many orders in the Sales table.

Having relationships between your tables prevents the need to flatten the tables, or combine them together into a single table, before importing the data into the model. Power BI uses an Autodetect feature to work out the cardinality of relationships, for both those you create manually, and those it has created automatically. You change the cardinality by clicking **Manage Relationships** from the **Home** tab. In the **Manage relationships** dialog box, double-click a relationship, or click **Edit**, to open the **Edit relationship** dialog box and select from the **Cardinality** list to change it. Click **OK**, and then click **Close**. In the Model view, double-click a relationship in the diagram to open the **Edit relationship** dialog box. Change the cardinality, and click **OK**.

Cross filter direction

The cross filter direction of the relationships in your dataset affects how Power BI treats the tables in visualizations in your reports. When you manually create a relationship, or the Autodetect feature generates the relationship for you, Power BI makes a best guess at the cross filter direction. The direction can be Both, or Single:

- The cross filter direction of relationships affects how Power BI treats the tables in visualizations:
  - The cross filter direction is automatically set when relationships are created manually or using Autodetect
  - Power BI makes best guess at the direction
- Two types of cross filter direction:
  - Both
  - Single

- **Both**: Both is the most common, and the default. When you apply filtering, the two tables are considered as one table for aggregating data in a visualization. The Both cross filter direction is ideal for a table that is related to numerous lookup tables, such as a fact table in a star schema. For example, a FactInternetSales table is surrounded by the related lookup tables, such as DimCustomer, DimCurrency, DimDate, DimProduct, DimPromotion, and DimSalesTerritory. The layout of the tables in the Model view might reflect this as a snowflake shape. In this example, there is a mix of cross filter direction types. However, if you have a lookup table that is related to more than one (non-lookup) table, you might want to set the cross filter direction to Single. For example, if you have two tables with values for aggregating, that are unrelated, but both reference a Country lookup table, then set the cross filter direction to Single. This prevents aggregations from including data that is not actually connected. The FactInternetSales table has a many to one relationship with DimCustomer, using a cross filter direction of Both. With this, you can use both tables as one in your visualizations. The DimCurrency table is also related to the FactInternetSales table with a many to one relationship, but this has a Single cross filter direction, preventing any other tables that use this lookup from inclusion in aggregations.
- **Single**: With a Single cross filter direction, the filters in related tables operate on the table where the values are aggregated. If you have imported data from Power Pivot for Excel® 2013 or earlier, all relationships have Single cross filter direction.

You manually change the cross filter direction by clicking **Manage Relationships** from the **Home** tab. In the **Manage relationships** dialog box, double-click a relationship, or click **Edit**, to open the **Edit relationship** dialog box, and select **Both**, or **Single** in the **Cross filter direction** list. Click **OK**, and then click **Close**. Alternatively, in the Model view, double-click a relationship in the diagram to open the **Edit relationship** dialog box.

Note: The direction of the cross filter is displayed as an arrow for Single, or a double arrow for Both on the relationship line. The Single arrow points in the direction of the filter.

Demonstration: Working with relationships in Power BI

In this demonstration, you will see how to:

- Import a data extract into Power BI.
- View and edit the relationships created automatically.
- Add new relationships.

Check Your Knowledge

Question	
Which of the following statements is false?	
Select the correct answer.	
	The Power BI Autodetect feature works out the cardinality of the relationship between two tables.
	When querying the data source, Power BI automatically determines the relationships, and creates them.
	The Sales table is related to the Customer table using the CustomerID column. There are many orders in the Sales table for each customer, and one row in the Customers table for each customer. This is a many to one relationship.
	The Employees table has one row for each employee, and is related to the EmployeeAdditionalDetails table using the EmployeeID column. There is one instance of each employee in the EmployeeAdditionalDetails table. This is a one to one relationship.
	After Power BI automatically creates a relationship, you cannot change the cardinality or cross filter direction options.

Lesson 2
DAX queries

In this lesson, you will learn about DAX, the syntax structure, and how to use functions.

Lesson Objectives

At the end of this lesson, you will be able to:

- Describe DAX and what it is used for.
- Understand the DAX syntax so you can create queries.
- Write DAX queries using functions.
- Understand the importance of context when using DAX.

What is DAX

Data Analysis Expressions (DAX) is a formula language that comprises a library of more than 200 functions, constants, and operators. You use DAX in a formula or expression, to calculate and return a single value, or multiple values. DAX is not new—you may have used it in Power Pivot for Excel or SQL Server Analysis Services (SSAS). If you have used Excel formulas, you will discover some similarity; however, DAX functions are designed specifically to work with relational data, which is what you work with in your Power BI datasets. DAX is commonly used in calculated columns and measures, both of which are covered in more detail in the next lesson.

- Data Analysis Expressions (DAX) is a formula language:
  - Comprises a library of more than 200 functions, constants, and operators
  - Use DAX in formula or expression to calculate and return single value, or multiple values
  - Not a new feature—it already exists for Power Pivot for Excel, and SQL Server Analysis Services (SSAS); in Power BI, it is designed to work with relational data
  - With DAX, you perform calculations such as year-on-year sales, running totals, like-for-like sales, and predict profit
  - Helps you gain insights into data that you would not necessarily see just by importing it

Why use DAX?

You import your data into Power BI Desktop and can begin creating reports straightaway. However, while this certainly presents your data visually, and facilitates interaction using the drill-through feature, what if you want to include year-on-year sales growth, or running totals based on monthly sales, or perhaps predict profit for next year? With DAX formulas, you can do this, and they can help you find the insights you want to extract from your data to make it more useful. For example, you might want to compare sales so far this year, like-for-like with last year. If the current month is May, you only want to compare that part of the previous year. DAX provides a function for this, as shown in the following code. This is not something that is easy to do without DAX:

The following DAX formula returns the sales from last year, using the sales dates for the current year, to provide a like-for-like comparison:

Calculate Sales for the Same Time Period Last Year

```
Last Year Sales = CALCULATE ([Total Sales], SAMEPERIODLASTYEAR('Date'[Date]))
```

The key to understanding and using DAX is learning the concepts of the syntax for structuring your formulas, the functions you use to make calculations, and context. These concepts are covered in detail in the remainder of this lesson.

Syntax

The DAX formulas that you write must be syntactically correct; otherwise, Power BI gives you a syntax error message. Therefore, it is important to understand how to structure your expressions. The following code shows an example of a typical formula you might use in Power BI to create a measure:

- DAX formulas must be syntactically correct before you can save them to the model:
  - Use DAX formulas to create measures and calculated columns
  - The first part of the formula is the name of the measure, or calculated column
  - This is followed by the equal operator (=)
  - The equal operator returns the result of the calculation to its right, back to the measure (much like a variable)
  - Functions must have at least one argument passed to it in parentheses ()
  - Measures created in context of current table—can move
  - Include table and column name:
    - Column name must be enclosed in square brackets []
    - Table names with spaces or reserved words must be enclosed with single quotation marks (')

The following DAX formula adds together the values in the LineTotal column of the InternetSales table, and returns a measure named Total Sales:

DAX Formula to Calculate Total Sales

```
Total Sales = SUM(InternetSales[LineTotal])
```

The first part of the formula is Total Sales. This example uses a measure, but it could be a calculated column, and you can rename both in the Report view. The name can contain spaces, in addition to symbols such as the percentage sign (%). The name of the measure is followed by the equal operator (=). The equal operator returns the value of the calculation to the right of it, to the measure, in much the same way as you assign values to a variable. This example uses the SUM function, and adds up all values in the argument you pass to it in the parenthesis (). An argument passes a value to the function, and all functions must have at least one argument. In this case, the argument is the LineTotal column in the InternetSales table.

When you write DAX formula, Power BI creates the new measure in the context of the current table. However, this is completely flexible, and you can move the measure to whichever table you want. Select the measure in the Fields pane, and then select the **Modeling** tab. In the Properties group, click **Home Table**, and then select the table where you want to move the measure. If you create the above example in the InternetSales table, you can move it elsewhere without affecting the formula. Because you passed the table and column name as the argument, this creates independence—the function knows exactly which values to operate on, regardless of its home table.

Note: When you refer to a column in a formula, and include the table name, this is known as a "fully qualified column name". You exclude the table name when the measure refers to a column in the same table in which it also resides; however, it's good practice to include it. While this can lengthen formulas that reference many columns, it provides clarity and the reassurance that you are referencing the correct columns—you can also create measures that span multiple tables, and move them as required.

If your table name contains spaces, reserved keywords, or disallowed characters, enclose the name using single quotation marks. Table names containing characters outside of the ANSI alphanumeric character range will also need enclosing with single quotation marks. The column name is always encased with square brackets; for example, [LineTotal].

You type DAX formula into the formula bar. There are two buttons to the left of the bar, a cross (X) icon, and a tick icon. The cross icon cancels the measure, and removes any work without saving. The tick icon validates your syntax, and enters your new measure into the model.

If you are already familiar with Power BI, you might know that numerical fields are automatically calculated, and wonder why you would want to create the above measure, because Power BI will sum this for you. By adding this measure, you use it as an argument for another formula, meaning you can create all the calculations you require within your dataset. For more information on DAX syntax, see *DAX syntax* in Microsoft Docs:

DAX syntax

http://aka.ms/tl7369

Functions

Functions are predefined formulas that perform calculations on one or more arguments. As you learned in the previous topic, you can pass a column as an argument, in addition to using other functions, expressions, formulas, constants, numbers, text, and TRUE or FALSE values. The DAX library of more than 200 functions, operators, and constructs, is segmented into the following 10 categories:

- DAX functions are predefined formulas that perform calculations on one or more arguments:
  - You pass a column, function, expression, formula, constant, number, text, TRUE or FALSE as arguments
  - DAX library of 200-plus functions, operators, and constructs, in the following categories: date and time, time intelligence, filter, information, logical, math and trig, other, parent and child, statistical, and text
  - DAX functions are similar to Excel, but reference an entire column or table; use filters to reference selected values
  - Functions that return a table do not display results
  - VLOOKUP is effectively replaced with relational data model

- **Date and time**: similar to the date and time functions used in Excel, but based on the datetime data types used by Microsoft SQL Server. Date and time functions include DATEDIFF, DAY, EOMONTH, NOW, WEEKDAY, WEEKNUM, and YEAR.
- **Time-intelligence**: with these functions, you create calculations using date and time ranges combined with aggregations. This is useful for building comparisons across time periods. Time intelligence functions include CLOSINGBALANCEMONTH, DATEADD, NEXTQUARTER, NEXTYEAR, PREVIOUSMONTH, SAMEPERIODLASTYEAR, and TOTALYTD.
- **Filter**: with filter functions, you return specific data types, look up values in related tables, or filter by related values. The functions work by using tables and the relationships between them. Filter functions include CALCULATE, FILTER, ISFILTERED, RELATED, RELATEDTABLE, and VALUES.
- **Information**: information functions evaluate a table or column provided as an argument to another function, and inform you if the value matches the expected type. Information functions include ISBLANK, ISERROR, ISEVEN, ISNUMBER, ISTEXT, LOOKUPVALUE, and USERNAME.
- **Logical**: these functions return information about the value in your expression. Logical functions include FALSE, IF, IFERROR, NOT, OR, and TRUE.
- **Math and trig**: similar to the mathematical and trigonometric functions in Excel, math and trig functions perform a wide variety of calculations. Functions include ABS, ASIN, CEILING, CURRENCY, DEGREES, EVEN, FLOOR, ODD, PI, ROUND, ROUNDDOWN, ROUNDUP, SQRT, SUM, and TRUNC.
- **Other**: these functions are unique and do not fall into any of the other categories. They include EXCEPT, GROUPBY, INTERSECT, NATURALINNERJOIN, UNION, and VAR.
- **Parent and child**: parent and child functions work on data that is presented in a parent/child hierarchy in the data model. Parent and child functions include PATH, PATHCONTAINS, PATHITEM, PATHINREVERSE, and PATHLENGTH.

- **Statistical**: statistical functions are used to perform aggregations, such as SUM, MIN, MAX, and AVERAGE. With DAX, you can filter a column prior to aggregating, and create aggregations based on related tables. Further functions include COUNT, COUNTBLANK, COUNTROWS, CROSSJOIN, MEDIAN, ROW, SIN, TAN, and TOPN.
- **Text**: text functions operate on string values. You can use them to search for text within a string; return a substring; format dates, times, and numbers; concatenate strings. Text functions include CONCATENATE, FIND, LEFT, LEN, LOWER, REPLACE, RIGHT, SEARCH, TRIM, and UPPER.

For a full list of DAX functions, and examples of how to use each function, see *DAX function reference* in Microsoft Docs:

DAX function reference

http://aka.ms/lrf8p9

If you have been using Excel functions, DAX functions might look familiar. However, DAX functions differ in the following ways:

- DAX functions reference an entire column or a table. To use selected values from a table or column, you include filters in your formula.
- If you want to customize a calculation to work on a row-by-row basis, use functions to utilize the current row value, or related value as an argument.
- If you use one of the DAX functions that returns a table, rather than a single value, the table is not displayed—it's used to provide input for another function. For example, return a table and count the values, count distinct values, or filter columns and aggregate the values.
- With the time intelligence functions, you define or select date ranges, and then perform calculations on them.
- Instead of using a VLOOKUP, as you would in Excel, DAX functions accept a column or table as a reference. In Power BI, you work on a relational data model, so finding values in another table is straightforward because you can create relationships, and might not actually need a formula.

Context

Context is an important concept to understand if you want to write expressions that return the results you expect. In DAX, there are two types of context: row context, and filter context:

- DAX expressions use two types of context:
  - Row context:
    - Row context is the current row
    - Often applied to measures to identify a single row
  - Filter context:
    - Exists in addition to row context
    - A filter context is one or more filters applied in a calculation to determine the single value or result
    - Filter contexts are used in visualizations; for example, a chart with Sales, Sales Person, and Month. The chart returns subsets of data based on a specific Sales Person, and Month
    - You can apply filter contexts using visualizations, and DAX

- **Row context**: you can think of row context as the current row. When a formula includes a function that uses filters to identify a single row in a table, this is considered row context. The function applies a row context to each row in the table to which the filter is applied. This type of context is often applied to measures.
- **Filter context**: filter context exists in addition to row context. A filter context is one or more filters applied in a calculation, which determine a single value or result. You use a filter context to reduce the values that are included in a calculation. The filter specifies the row context, and also a particular

value or filter, in that row context. Filter contexts select subsets of data. If you have a visualization in your report that includes Sales, Sales Person, and Month, the filter context works on subsets of data to return Sales by a specific Sales Person and Year. You can apply filter context by using filters this way in your reports, or by using DAX.

The following measure demonstrates how row context and filter context operate on the calculation in the formula. A new measure is created and named UK Sales. The CALCULATE function evaluates the expression in brackets, in a context set by the filters. The first argument in the expression is the measure [Total Sales], which has the formula, Total Sales = SUM(Sales[Revenue]). The comma separates the first argument from the filter argument. In this formula, the referenced column [Country], in the Customers table, sets the row context. Each row in the Country column specifies a country, such as France, Germany, UK, or US. This code filters on the UK, providing the filter context.

The following code is an example of a measure with the Country column in the Customers table as the row context, and the UK value as the filter context:

Using Row Context and Filter Context in a Measure

```
UK Sales = CALCULATE([Total Sales], Customers[Country] = "UK")
```

This formula uses Total Sales, and applies a filter of UK, so only the sum of UK sales is returned in the result. DAX is powerful in its ability to reference a selected value from a related table.

Demonstration: Using row and filter context in DAX formulas

In this demonstration, you will see how row and filter context works with measures.

Check Your Knowledge

Question	
You want to concatenate and manipulate columns containing string data. Which of the following functions will not be compatible for working with text?	
Select the correct answer.	
	CONCATENATE
	MEDIAN
	REPLACE
	TRIM
	UPPER

Lesson 3
Calculations and measures

In this lesson, you will see how to manipulate your data using calculated columns and calculated tables, and learn how measures provide additional insights into your data.

Lesson Objectives

At the end of this lesson, you will be able to:

- Add calculated columns to your tables.
- Create a new calculated table within your dataset.
- Add measures to your queries to deliver insights into your data.

Calculated columns

Calculated columns are added to your tables by applying DAX formulas to your existing data. The DAX formula defines the values in the new column, rather than querying the data source to create the column. Calculated columns are useful when the data source does not contain data presented in a format that you want. You can concatenate strings or multiple numbers together, combining data from anywhere in the model, to create a calculated column.

- Calculated columns are added to tables using DAX formulas to perform operations on existing data:
  - DAX formula defines the new column using data in the model, rather than querying the data source
  - Useful when the model does not have the data presented in a format you need
  - Concatenate strings, calculate numbers, or combine data from elsewhere in the model
  - Different from custom columns that query the data source
  - Similar to measures, as both use DAX formulas, but measures used in Values area of a visualization, calculated columns used in Axis, Legend, or Group
  - Use New Column on Modeling tab to create column
  - After creating, use in visualizations as you would any other column

Calculated columns differ from custom columns, because they use data that already resides in the model. They are similar to measures, as both measures and calculated columns use a DAX formula, but the difference is in how they are used. Generally, measures are used in the Values area of a visualization, to calculate the results based on other columns used in the Axis, Legend, or Group area of the visualization. Calculated columns are used for the fields you want to add to the Axis, Legend, or Group.

In Power BI Desktop, you use the **New Column** button on the **Modeling** tab to create a calculated column, or right-click the table name in the FIELDS pane and select **New column**. This opens the formula bar where you can type your DAX formula, and press Enter to create it. By default, Power BI names the new column as Column, but you can change this by typing in a new name. The following example creates a new column called Full Name, concatenating existing fields together.

The following code example concatenates the First Name and Last Name fields into a new calculated column called Full Name:

Create a Calculated Column Using Existing Data

```
Full Name = [First Name] & " " & [Last Name]
```

The above code does not include the table names, so these are classed as nonqualified column names. The columns exist within the Customers table, so they do not have to be qualified. In a small dataset, with no possibility of duplicate names in other tables, this is less of an issue, but it is considered good practice to include the table name for clarity. If you referred to a column in another table, then you must fully qualify the column. The following example uses the RELATED function to look up a value in another table.

The following code example returns the related Region value for the City column in the Customers table:

Create a Calculated Column Using the RELATED Function

```
Location = RELATED(Countries[Region]) & “, “ & [City]
```

After creating a calculated column, it appears in the FIELDS pane and behaves in the same way as other columns. However, you can identify calculated columns by the icon next to the name. Calculated columns can have any name you want, and added to visualizations in exactly the same way as other columns.

Calculated tables

Like calculated columns, calculated tables are created using data that already exists in the model—you use a DAX formula to define the values in the table. Tables are created in both the Report view, and the Data view in Power BI Desktop. Calculated tables work well for intermediate calculations, and data that you want to be stored in the model, rather than calculated when the data source is queried.

- Create calculated tables using data that exists in the model:
  - Create table in Report view, or Data view
  - Use data from the model to create the new table, rather than querying the data source
  - From the **Modeling** tab, in the **Calculations** group, click **New Table**, and then add DAX formula
  - Use functions such as UNION, NATURALINNERJOIN, NATURALLEFTOUTERJOIN, or DATATABLE
  - Calculated tables and columns are used in the same way as other tables. Rename table and columns, use in relationships with other tables, change data types, add columns, measures, and use in visualizations

To create a calculated table, open Power BI Desktop and the report with the dataset where you want to add the table. Click **Modeling**, and then in the **Calculations** group, click **New Table**. The formula bar opens and, by default, is populated with Table =. You can overwrite the word Table to give your table a better name. Write your DAX formula to the right of the equal sign, which creates your table. For example, you could use a union, inner, left, or cross join function in your DAX to create the table. The following example creates a calculated table using the UNION function:

The following code combines the existing NorthAmericanSales and EuropeanSales tables into one, to create a calculated table named Global Sales:

Combine Existing Tables with UNION to Create Calculated Table

```
Global Sales = UNION (NorthAmericanSales, EuropeanSales)
```

When using the UNION function to combine two tables into one new calculated table, the tables must have the same number of columns. The columns are combined on their position in the table, so make sure the column order matches between the two tables. UNION includes duplicate rows that exist in both tables. If you want to remove duplicate rows, open Power Query Editor, and from the **Reduce Rows** group on the **Home** tab, click **Remove Duplicates**. The new table has the same column names as the first table, so in the preceding example, the UNION would take the names of the columns in the NorthAmericanSales table. The order of the columns is also taken from the first table, and related tables are not included in the union.

While UNION appends rows from one table to another, you merge columns using one of the join functions. You can use NATURALINNERJOIN, or NATURALLEFTOUTERJOIN, to merge the columns of two tables that have a related column. The following example joins the Customers table to the Sales table on the CustomerID column, which is included in both tables. The columns from the Sales table are added to the right of the Customers table columns, to create the Customer Sales table.

The DAX function uses the NATURALINNERJOIN function to create a new table called Customer Sales, which adds the columns from the Sales table to the columns from the Customers table:

Create Calculated Table Using the NATURALINNERJOIN Function

```
Customer Sales = NATURALINNERJOIN (Customers, Sales)
```

In the preceding example, only rows with matching values in both tables are added to the new calculated table. To include all rows in the Customers table, regardless of a match in the Sales table, use NATURALLEFTOUTERJOIN instead. When using a join, the columns you are joining on must have the same data type.

Use the DATATABLE function in your DAX formula to create a new table, set the data types of the columns, and insert data. It is best to create your calculated tables in the Data view as you can view the new table immediately. The following code creates a Countries table, and adds values to the table:

The following code example creates a new table using the DATATABLE function. Use it to define the column names and data types, and enter values into the table:

Use the DATATABLE Function to Create a New Table

```
Countries = DATATABLE
      (
         "Country", STRING,
                 "Code", STRING,
        {
                {"United States", "US"},
                        {"United Kingdom", "UK"},
                        {"France", "FR"},
                        {"Germany", "DE"},
                        {"Spain", "ES"}
                }
           )
```

After you create a calculated table, you use it in exactly the same way as any other table that exists in the model, including for in relationships. You give the table and column names any name you like, and format them as you would with a standard table. You then use the columns in your visualizations alongside columns from other tables. You can also add calculated columns and measures to visualizations.

Measures

Power BI measures help you discover insights in your data that might otherwise be hidden. You use measures to answer questions about your data. Some common examples would be using aggregations such as average, minimum, maximum, count distinct, or more complex calculations that use a DAX function. The values in your measures will update and change alongside a data refresh, so your reports always display up-to-date figures.

- Measures help you discover insights into your data that might otherwise be hidden:
  - Include aggregations in your measures, such as average, minimum, maximum, count distinct DAX functions
  - Use other DAX functions to create complex calculations
  - Useful for highlighting running totals, comparing sales this year to date with sales for the same period last year, and sales forecasting
  - Create measures in Report view or Data view
  - Measures are used in visualizations as for any other column
  - Change the Home table where the measure resides

Measures are created using DAX formulas, and with an extensive library of functions, operators, and constructs, there is scope to create all the measures you require. Measures are useful for creating running totals, or comparing sales for a partial year to sales over the same time the previous year. You can

also predict sales by multiplying current year sales against a target percentage for growth, resulting in an expected sales target.

In Power BI Desktop, you create measures in Report view or Data view, and they appear in the FIELDS pane. To create a new measure in Report view or Data view, right-click the table in the FIELDS pane, and click **New measure**. Alternatively, from the **Modeling** tab, in the **Calculations** group, click **New Measure**. It is generally easier to work in the Data view, because you see the values of the data in the table to which you want to add the measure. The following example creates a measure named **YTD Sales**. Using the TOTALYTD function, the SalesAmount column in the FactInternetSales table is aggregated using SUM, and the dates for the current year.

The following code creates a measure called YTD Sales. It uses the TOTALYTD function to calculate the year to date sales:

Create a Measure Calculate Year to Date Sales

```
YTD Sales = TOTALYTD(SUM(FactInternetSales[SalesAmount]), DimDate[FullDateAlternateKey])
```

After creating measures, you add them to visualizations in your report, as you would any other column. If you have a visualization showing Last Year's Sales, you could create a new measure to calculate sales for the coming year, based on a predicted growth percentage. The following example creates a measure that multiplies sales for last year by 1.05, or 5 percent:

The following code creates a measure to predict sales for last year based on a 5 percent increase:

Create a Measure to Predict Sales for the Coming Year

```
Sales Forecast = SUM('Sales'[LY Sales]) * 1.05
```

You can change the table in which the measure resides. In the Fields pane, click the measure you want to move, and highlight it. From the **Modeling** tab, in the **Properties** group, click **Home Table**, and select the table.

Demonstration: Creating calculated columns and measures with DAX

In this demonstration, you will see how to:

- Create calculated columns.
- Add a new table.
- Create a new measure.

Check Your Knowledge

Question	
Which of the following DAX functions is not suitable for creating a calculated table?	
Select the correct answer.	
	UNION
	SUM
	CROSSJOIN
	NATURALINNERJOIN
	NATURALLEFTOUTERJOIN

Lab: Modeling data

Scenario

Adventure Works employees are increasingly frustrated by the time it takes to implement managed BI services. The existing managed BI infrastructure, including a data warehouse, enterprise data models, and reports and dashboards, are valued sources of decision-making information. However, users increasingly want to explore relationships with other, currently unmanaged data. It takes too long for the IT department to include these requirements into the corporate BI solution.

As a BI professional, you need to explore ways in which Adventure Works can empower business users to augment their managed enterprise BI solution with self-service BI.

Objectives

After completing this lab, you will be able to:

- View and create relationships in your dataset.
- Add a calculated column to a table.

Note: Because of updates to Microsoft Power BI, the lab steps for this course change frequently. Microsoft Learning regularly updates the lab steps, so they are not available in this manual – but you can access them on GitHub.

Lab Setup

Estimated Time: 60 minutes

Virtual machine: **20778C-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa55w.rd**

All the lab steps are contained in 20778C_LAB_05.md.

Exercise 1: Create relationships

Scenario

The data in your organization spreads across a number of sources. To begin with, you will import data extracts from Excel worksheets. The data should be related, so you will examine the relationships that Power BI detects automatically. Because the sales data is an extract, Power BI might not detect all of the relationships, or create them correctly, so you will have to configure them.

The main tasks for this exercise are as follows:

1. Preparing the environment
2. Automatic relationships
3. Manual relationships

▶ Task 1: Preparing the environment

▶ Task 2: Automatic relationships

▶ Task 3: Manual relationships

Results: At the end of this exercise, you will have a dataset combining data from two Excel worksheets, with relationships between the tables.

Exercise 2: Calculations

Scenario

Having created the required relationships in your dataset, you feel that you might benefit from some additional data that doesn't currently exist. You will add calculated columns to the tables in your dataset, to fill in the gaps.

The main tasks for this exercise are as follows:

1. Add a calculated column

▶ Task 1: Add a calculated column

Results: At the end of this exercise, you will have calculated columns added to the tables in your dataset.

Question: Discuss the functions covered in this topic, or use the link provided in the Functions topic of the DAX Queries lesson to look online at the DAX function reference. How many of these have you already used? Have you used the equivalent functions in Excel? Which functions can you use for creating columns and measures in your organizational datasets?

Question: Look at the dataset you used in the labs. How else can you use DAX formulas to add additional columns or create new measures? Do you think there are any gaps in the data that you could fill using DAX?

Module Review and Takeaways

Microsoft Power BI is making its mark in the self-service BI world—because it can quickly create visually stunning, interactive reports and dashboards. Power BI provides a straightforward way to combine data from a wide range of sources into a single dataset, and then work with that data to create cohesive reports. This module went behind the scenes of the visualizations, and explored the techniques and features on offer to shape and enhance your data. With automatic relationship creation, a vast library of DAX functions, and the ability to add calculated columns, tables, and measures quickly, you have seen how Power BI creates attractive reports, while helping you find hidden insights into your data.

Module 6

Interactive Data Visualizations

Contents:

Module Overview

Self-service business intelligence (BI) is becoming increasingly popular within organizations. This approach enables business users to access corporate data, and create and share reports and key performance indicators (KPIs), without dependency on a dedicated report developer. Business users can utilize the Microsoft Power BI suite of tools to connect to a wide variety of data sources. These include the main industry-standard databases, Microsoft cloud-based services—Microsoft Azure SQL Database, Azure Data Lake, and Azure Machine Learning—alongside Microsoft Excel® and other files, and software as a service (SaaS) providers such as Microsoft Bing®, Facebook, and MailChimp. The combination of flexibility and the ability to create visually stunning, interactive dashboards quickly makes Power BI an obvious choice for any organization that needs to provide its users with a self-service BI solution.

Objectives

After completing this module, you will be able to:

- Use Power BI Desktop to create interactive data visualizations.
- Manage a Power BI solution.

Lesson 1
Creating Power BI reports

This lesson concentrates on the visual report items that you add to Power BI charts. You will learn about the different types of charts, including custom visualizations.

Lesson Objectives

After completing this lesson, you will be able to:

- Set the page properties of your reports to customize display options.
- Work with multiple visualizations on a report, and change default categorization and summarization.
- Add charts to a Power BI report and customize chart settings.
- Work with geographic data and present it by using map visuals in a report.
- Use histograms to represent data.

Page layout and formatting

Each report can be customized with formatting options to enable you to design layouts suitable for your corporate look and feel. To view and edit the formatting of a report, click the report canvas without highlighting any visuals, shapes, or other elements on the report then, in the **VISUALIZATIONS** pane, click **Format**.

- Customize each report using formatting options:
  - **Page name**: give each report a name to describe the content, rather than the Power BI default of *Page 1*, *Page 2*
  - **Page size**: default aspect ratio is 16:9. Change to 4:3, Cortana, Letter, or set width and height in pixels using the Custom option
  - **Page background**: change the background color and transparency. Use theme color, or own color. Use image to create highly customized reports
  - **Page view**: alter the zoom on the page. Default is Page view—fits all visuals onto screen. Choose Actual Size for one-to-one pixel mapping

Page name

Under the **Page information** section, you can change the title of your report from the default naming convention of Page 1, Page 2, Page 3, and so on, automatically applied by Power BI. This is particularly useful if your report comprises multiple pages, because you can guide users through the report pages using applicable names. Type a new name into the **Name** box, and the name of the page is reflected immediately on the tab at the bottom of the Power BI Desktop screen.

Page size

By default, each new page you add to a report is created with an aspect ratio of **16:9**, which is the most suitable for modern monitors and laptop screens. You can change this to **4:3**, which creates a more square report size. If you already have visuals on your page, and change the aspect ratio, you might need to relocate or resize any that fall off the report canvas, or overlap. Other sizing options include **Cortana**, **Letter**, and **Custom**. If you select the **Cortana** layout, this might not appear useful for your current screen resolution because the page becomes tall and thin. However, you can use Power BI to produce a set of results displayed in Cortana when you ask a question of Cortana. The report can be formatted in a highly specified layout—for example, you could create a summary page and another page showing more detail. You use the **Letter** page size to add reports to Office documents without losing the aspect. The **Custom** page size option is useful if you want to pin visuals in dashboards in the Power BI service, or embed visuals within custom web pages or applications. You specify the exact width and height of the page in pixels.

Page background

Each report background can be altered using colors and transparency. You select from the default theme colors that match the default colors in visuals, or you specify a custom color. The color can then be altered using the transparency setting to lighten or darken the tone. You can also use an image for the background. The image displays increased transparency so that visuals remain readable, and it's an effective way of creating highly customized reports.

Page view

You change the zoom used to display the page by selecting **Page View** from the **View** group on the **View** ribbon. By default, this is set to **Fit to Page**. You select **Actual Size**, which zooms in to the report so you see visuals in a closer view, with one-to-one pixel mapping. Scroll bars appear to bring any hidden areas into sight. Select **Fit to Width** to make the page fit within the width of the screen.

Working with multiple visualizations

It's likely that you will create reports that incorporate several visuals. When you have more than one visual, you control how they interact with one another, and think about their alignment and positioning on the report page, to deliver the best possible experience to the end user.

- Use settings and formatting to ensure multiple visuals interact correctly on a report:
  - **Visual relationships**:
    - **Filter**: only show corresponding data
    - **None**: show all data, do not interact
    - **Highlight**: shows all data, corresponding values highlighted
  - **Show items with no data**: displays items with empty values, value of 0 included by default
  - **Default summarization**: change from the default sum to average, minimum, maximum, count, or count distinct
  - **Default categorization**: for example, ensures address fields are categorized as City rather than Country, or State
  - **Arrange report elements**: sending visuals forwards or backwards to create layers—known as z-order

Visual relationships

When you click a data point in one of the multiple visuals on your report, the other visuals respond by highlighting the corresponding data. For example, if you have a column chart showing sales by country, and a donut chart showing sales by product color, when you click the data point for Germany in the column chart, the following happens as a result: the column for Germany remains the same color, but the other country columns display with increased transparency. Furthermore, the corresponding data in the donut chart, represented as a ring, remains in high color for Germany, while the data representing all other countries shows increased transparency—you now see related sales. However, if you want to click the column chart and show data in the ring chart for Germany only, you achieve this by editing how the visuals interact:

1. Click the visual for which you want to edit the interactive properties.
2. The **Format** ribbon appears. Click **Edit interactions**.
3. The other visuals on the page now show **Filter** and **None** icons. Some visuals, such as donut and column charts, will also display a **Highlight** icon, depicted as a chart. Click an icon to choose how each visual responds when you click a data point in the highlighted visual:
   a. **Filter**: this displays only the data for the selected data point. Returning to the previous example, if the column chart displaying sales by country is selected, selecting the Highlight icon on the donut chart would display data only for the selected country.
   b. **None**: the visual will not change.
   c. **Highlight**: this is the default behavior, whereby the corresponding data remains in full color, and the remaining data is displayed with increased transparency.

You can set how each visual, including slicers, interacts when a data point is selected on another visual, giving you complete control. When you have a group of visuals with the same behavior, you use shapes, such as the rectangle, or lines, to show visually that they relate and operate together.

Items with no data

By default, Power BI only displays column headings in reports for items containing data. If you had a table showing sales by country, and did not yet have sales in Italy, then Italy would be excluded from the results. However, it might be that you want to see results for items with no data. You can then identify countries with no sales, or products that have not sold this week. To change this, right-click the field in the **Visuals** bucket in the **VISUALIZATIONS** pane, and then click **Show items with no data**. Any empty columns now appear with blank values. However, if a column has 0 as a value, it appears in visualizations.

Default summarization and categorization

The data model contains two properties that you use to set the default summarization and categorization of fields. When you import data into the data model in its raw form, it might not be obvious what this data represents, and Power BI defaults to using the sum aggregator. You might have fields for which you want to apply a count or average aggregation, and you change this using the **Default Summarization** property:

1. In the **Fields** pane, click the column in the **Fields** list that you want to change. The column will be highlighted with a yellow border.
2. On the **Modeling** tab, in the **Properties** group, click **Default Summarization: Sum** to show the full list of options. Choose from **Don't summarize**, **Sum**, **Average**, **Minimum**, **Maximum**, **Count**, or **Count (Distinct)**.
3. When a new visual is created using that field, the aggregator value is now changed. However, if you have an existing visual that uses the column for which you have changed the default summarization, this is not updated. In this case, right-click the column in the **Value** bucket, or select the down arrow, and choose another summarization option.

You use the Formatting group on the Modeling ribbon to add symbols to your data—which is particularly useful for indicating what the data represents. Choose from numerous currency symbols, apply a percentage format, and manage how you present numbers with commas to separate thousands.

The default categorizations of a field can also be customized. For example, when working with geographic data, if you have locations that could be considered either a country or a state, such as Georgia, or a city or a state, such as Washington, you can add a categorization so that map visuals plot the data with accuracy. To change the categorization, perform the following steps:

1. In the **Fields** pane, click the column in the **Fields** list that you want to change. The column will be highlighted with a yellow border.
2. On the **Modeling** tab, in the **Properties** group, click **Data Category: Uncategorized**. Select one of the following options: **Uncategorized**, **Address**, **City**, **Continent**, **Country/Region**, **County**, **Latitude**, **Longitude**, **Place**, **Postal Code**, **State or Province**, **Web URL**, **Image URL**, or **Barcode**.

Arranging report elements

When your reports include many elements, you can use settings within Power BI to control how they overlap each other, whether they are layered, or arranged one on top of the other. This design strategy is more commonly known as the z-order, and is particularly helpful for arranging visuals over shapes used as borders to group elements together. You control the z-order using the **Arrange** group on the **Format** ribbon. When you click a visual, the **Format** ribbon appears. Click **Bring forward** to move an element in front of another, or click **Send backward** to force it behind another. The **Arrange** group includes the **Align** menu, so you can align elements **left**, **center**, **right**, **top**, **middle**, or **bottom**. You can select

multiple elements and select **Distribute horizontally**, or **Distribute vertically**, to arrange the elements with equal spacing between them.

Creating charts

By using the chart visuals in Power BI Desktop, you can quickly create visually stunning interactive reports and dashboards.

- Power BI includes a wide range of chart types:
  - Bar and column
  - Line and area
  - Line and column
  - Ribbon
  - Scatter and bubble
  - Funnel
  - Gauge
  - Pie and donut
  - Slicers
  - Waterfall
  - Table and matrix
  - Tree map
  - R visual
- Formatting charts

Chart types

Power BI includes bar, column, area, line, pie, and scatter charts, along with maps, slicers, gauges, KPIs, R, and table visuals. You can select a chart from the **VISUALIZATIONS** pane to add to the report canvas, or you can drag a data field onto the report to create a table visual automatically—this can then be converted to another chart type. For example, you can drag a **Categories** field onto the report, which creates a table. You can then drag **TotalSales** onto the table, to add another column. You can then click one of the chart icons in the **VISUALIZATIONS** pane, and quickly switch between bar or pie charts. After adding charts to your report, you can optionally set the page filter property so that users can drill down, and chart items simultaneously reflect the page filter.

Bar and column charts

Stacked bar and column charts are identical, except that the bars on a stacked bar chart span horizontally, rather than vertically, as in a column chart. Each chart accepts an Axis field, such as **Sales Person**, and a Value, for example, **Sales YTD**. The data field in the Value will be a numeric value that can be summed. You include another data field for the Legend, such as **City**, to color-code the bars and show the city in which the salesperson operates.

Clustered bar and column charts are like stacked charts, but they include two data fields for the Value, which results in two bars or columns for each axis. To build on the previous example, you could add **Sales Quota** to the Value, to compare the amount of sales so far, with the target quota set for each salesperson.

Bar and column charts that are 100 percent stacked are like stacked and clustered charts, except that the bars and columns stretch the width or length of the chart area, and display the progress of each axis against a value. You add two data fields to the Value, such as **Sales YTD** and **Sales Quota**. Charts that are 100 percent stacked are useful for displaying progress in meeting a target figure. In this example, the **Sales YTD** figure can combine with the **Sales Quota** figure to show how far each salesperson is progressing toward meeting their annual target.

Line and area charts

The line and area charts are fundamentally the same, but the area chart is filled in, so the area below the line values appears as a solid block. Line and area charts are useful for displaying data over time, such as financial data. For example, you could chart sales over time, using year or month data for the Axis and **Gross Sales** for the Value. You use the stacked area chart to compare multiple values so, using the above example, you could add **Share Price** and **Net Sales** to show the profit that occurs over time, and how this affects the share price of the organization.

Line and column charts

The line and stacked column chart combines columns and lines. The columns and lines share the same data field for the axis, for example, **Year**. The column value could be **Gross Sales**, with a line value for **Share Price**. You can include multiple lines on a line and stacked column chart. The line and clustered column chart enables you to include multiple columns for each shared axis. To alter the previous example, the columns could represent **Gross Sales** and **Net Sales**, with a line for **Share Price**.

Ribbon charts

Ribbon charts initially look similar to stacked column charts, but they also enable you to visualize trends in your data. On a ribbon chart, the largest value is always at the top of the chart, so the values appear to flow like ribbons over time. You use a ribbon chart to display the sales figures for members of a team over time. The position of each member of the team will show you how they have performed relative to the other members of the team.

Scatter and bubble charts

A scatter chart shows the relationship between two numeric values by using circles that are plotted on the chart. Scatter charts are useful for displaying large sets of data and, in particular, highlighting nonlinear trends, outliers, and clusters. They also enable you to compare data without including time data. The more data you include, the better the results. Your scatter chart must include a point identifier, otherwise all of the data is aggregated into a single point—so add a non-numeric data field such as **Categories** to the **Details** property of the chart.

The bubble chart is based on the scatter chart and works with three numeric values. The bubbles are sized to represent the data proportionally. A bubble chart is created by using a scatter chart, and then adding a data field to the **Size** property.

Note: All the chart types listed previously enable you to add one or more reference lines. In the report view, click **Format**, and then toggle **Reference Line** to **On**. In the **Value** field, type a numeric value such as **100,000**. You can change the color and transparency, and choose a style from dotted, solid, or dashed. You use the **Arrange** property to decide whether you want the line behind or in front of the other elements on the chart.
Toggle **Data label** to **On** or **Off** to show or hide the number in the **Value** field. Power BI automatically displays the currency of the data, so if you add a reference line to a chart measuring sales, the reference line value appears as $100,000, for example. You can change the color of the data label, and choose the horizontal position, to display the label on the left or right, and above or below the line.
The scatter chart, which includes the bubble chart, enables you to set a reference line for the x-axis and y-axis. All formatting features are available, so you can fully customize both lines.

Funnel charts

Funnel charts help you visualize a process that flows between stages, such as a production process that has different steps, or a value that changes over a series. For example, you could show the relative time periods that a bicycle is in the different stages of production or the sales figures across different states. In the latter example, the data will generally be sorted on the sales figure and plotted horizontally, resulting in a funnel shaped chart.

Gauge charts

Gauge charts enable you to display a current value as progress towards a total goal. For example, you could show sales to date in a month as progress towards the monthly target.

Pie and donut charts

Pie and donut charts have similar functionality, except that the donut chart has a hollow center. For example, you could add **Salesperson** for the Legend value, and **Sales YTD** to Values. The pie or donut chart is divided into portions that are sized to represent the value. In this example, each **Salesperson** would have a portion of the pie or donut chart—the more sales they have achieved, the larger their portion.

Slicers

A slicer enables you to filter an entire report, applying the data selection to all visuals. You would add a slicer to filter on fields such as Territory, Region, Sales Person, Color, or Category. By default, visuals show values that include all data. Select a value in the slicer to filter all the visuals to show the data for the one selected value.

Note: The Power BI slicer includes the ability to search through the filter list, which is useful if the list is particularly long. On the slicer visual, click the ellipsis, then click Search, and start typing your search string. The list will filter the results as you type. Click to select the value to filter on.

Waterfall charts

The waterfall chart enables you to show changes in a value over time, such as annual revenue. Using a waterfall chart, you see how changes affect a value, and color-coded columns quickly highlight any increase or decrease in value. The chart includes two options: Category and Y Axis. For example, because waterfall charts are typically used to show changes in a value over time, you could add Year to the Category field, and Sales Variance to the Y Axis. This would display the data with the variance for each year, flowing left to right from the earliest to the latest year. By default, increases would show as green and decreases as red, though these are fully customizable. The chart also includes a total column on the far right.

Table and matrix charts

The table and matrix charts enable you to add data fields to create columns and build up a table. Each numeric column is automatically summed, with a total at the bottom of the column. Visually, the table and matrix charts look similar, with the matrix chart supporting the addition of rows and columns and providing drilldown functionality.

Tree map

The tree map might not physically represent a tree, but the principle behind its function is representative of a tree. On a tree map, larger data scales through to smaller data, as if the data were branches scaling down to twigs. The largest data value, represented as a rectangle, is in the lower-left corner, with the smallest in the upper-right corner. For example, add the **City** data field to Group, and **Total Sales** to Values. Each city is represented by a rectangle that is proportionate to the number of sales, so the cities that have the most sales have the largest rectangles.

R visual

Power BI Desktop supports statistical analysis through integration with R, and the hosting of R visualizations. When you select the R visual from the Visualizations pane, a placeholder is added to the page. You are then presented with an R script editor that you can use on the canvas. When you add fields to the R visual, they are automatically added to the R script editor pane. After you have created your script in the editor, click **Run**. The data added from the Fields pane is posted to the local installation of R. The script created in the R editor is then run on the local R installation. The R installation returns a visual to Power BI, which then displays on the canvas in the R visual.

Other charts

There are other types of chart in Power BI, including gauge, card, multi-row card, and KPI. For more information about using these charts, including how-to guides and tips, see *Visualizations in Power BI reports* in the Power BI documentation:

Visualizations in Power BI reports

http://aka.ms/Cfrub0

Formatting charts

Each chart includes options for formatting. The available options depend on the type of chart. If you use a data field—for example, **Salesperson**—in a column chart and a pie chart, the colors for each person are identical in the two charts. This retains consistency within the report, although you have the option to change the color for each data field. It also means that, when you click a **Salesperson**, all charts reflecting their data show as the same color. Use formatting to add data labels, change colors, and add titles, backgrounds, borders, and more.

When you add data to a visual, Power BI sorts values alphabetically. If you want to sort your data by another value, you can change the sort order by using the data model. The funnel chart is one example where you are likely to want to sort by a numeric value, rather than a string value; otherwise, the bars that form the funnel do not align to a funnel shape. To sort the data, view the dataset in **Data View**, on the **Sort** ribbon menu, select **Sort By Column**, and then choose the column from the list.

Using geographic data

In addition to the extensive list of chart types that are covered in the previous topic, Power BI Desktop includes the map chart, filled map chart, and ArcGIS map chart. Use these charts to map your data visually, by region and globally.

- Map and filled map charts:
  - Power BI integrates with Bing to determine location
  - Bing makes a best guess—known as geocoding
  - Always include location—longitude and latitude are aggregated
  - Add data categories to columns for better accuracy
  - Concatenate string address fields into one column
  - Display value data as:
    - Color-coded bubbles on map charts
    - Shaded regions on filled map charts
- ArcGIS map charts:
  - More customization
  - Provide reference layers, information pins, infographics

Map and filled map charts

Power BI integrates with Bing maps to find default coordinates for locations in map and filled map charts. It does this using a string value in a process known as geocoding. This integration means that you do not need to provide longitude and latitude coordinates in your data, with Bing making a best guess at the location.

Formatting your data for geocoding

The more information you provide for Bing to determine the location, the greater the chances of accuracy. Bing uses algorithms and hints to guess the location, so including additional location data helps Bing to make a better guess. Ensure that you name your columns usefully by using the geographic designation, such as **City**, **State**, **County**, **Province**, **Country**, and so on. When place names are ambiguous, such as Washington State or Washington DC, these column headings can help Bing to work out which you are referring to. You can also append additional information, so if your data refers to Washington in England, you can pass "Washington, England" to Bing. If you do have the longitude and latitude data for a location, you need to include a location field—otherwise the data is aggregated by default, and may not return the results that you expect.

Data categorization

When you import data, Power BI makes assumptions about that data based on the table and column names. Power BI assumes that you want to aggregate numeric columns—and always places them in the Values area when you drag them onto a chart. If you had a column named **Location Code**, with a value of "CA," this could refer to the state of California, or the country, Canada. Data categorization helps to solve this problem and can be applied in both the report view and the data view. In the **FIELDS** list, select the field that you want to categorize, and then on the **Modeling** ribbon, in the **Properties** group, select **Data Category**. You can choose from **Address**, **City**, **Continent**, **Country/Region**, **County**, **Latitude**, **Longitude**, **Place**, **Postal Code**, **State**, or **Province**. If a category is not appropriate for a data type, it is disabled in the list.

Creating specific location strings

In some instances, you might find that even using data categorization does not generate the desired locations in Bing. If this happens, you can create a new column and concatenate your address fields into a full address string. In Power BI Desktop, in either the report view or the data view, in the **FIELDS** list, select the dataset to which you want to add the new column. On the **Modeling** ribbon, in the **Calculations** group, click **New Column**. In the formula bar, concatenate your address fields, for example, by using the following code:

```
FullAddress = [AddressLine1] & " " & [AddressLine2] & " " & [City] & " " & [PostalCode]
```

The concatenation only works with string data types, so you might need to convert numeric values to string as part of your formula. You can then use this **FullAddress** field in your map chart.

Using map charts

The map chart accepts data for the **Location**, **Legend**, **Longitude**, **Latitude**, and **Size**properties. The **Location** property accepts fields such as **City**, **County**, and **Province**, whereas the **Size** property accepts numeric values such as **Total Sales** or **Number of Customers**. The numeric values are presented as colored bubbles on the applicable location. The bubbles are sized proportionally to the data that they represent within the field in the dataset—that is, the bigger the value, the bigger the bubble.

Note: If you drag a data field such as **City** or **Country** onto the report, Power BI detects that it is geographic data and automatically adds a map chart.

Using filled map charts

The filled map chart (also known as a choropleth), uses a slightly different visualization to represent the data. This chart uses colors to represent the data value across a geographic area.

You create the chart as for the map chart, but using the **Tooltips** property for the value data. This produces a chart with colored areas where data exists. You can then customize the colors using the Data colors section of the Format bucket. Click the vertical ellipsis next to **Default color**, and then click **Conditional formatting**. The default settings use red for the lowest value and green for the highest value, but you can change these to meet your needs. This shading is particularly useful for presenting socioeconomic data, because it provides a visual overview of data across a wide area, such as all of the states in the United States.

ArcGIS map charts

The ArcGIS map is provided by Esri (www.esri.com) for use in Power BI. When you first add an ArcGIS map to your report, you must agree to the Esri terms and privacy policy to continue using the chart.

You can add data to the ArcGIS map using **Location**, **Latitude**, **Longitude**, **Size**, and **Color** properties similar to the standard Power BI map charts. After populating the ArcGIS map, you can customize it using the ArcGIS menus. To open the menu, select the map on the report view, in the top-right of the map click the ellipsis, and then click **Edit**. An ArcGIS menu now appears across the top of the map to edit the visual in different ways:

- **Basemap** enables you to change the base map on which you display your data, choosing from different base shades or different street maps.
- **Location type** enables you to display the locations as boundaries or points on the map.
- **Map theme** enables you to modify how the value data displays, from heat maps to clustered points.
- **Symbol style** enables you to change the colors, transparency, and symbol types displaying the data.
- **Analytics** enables you to add information pins to your map, add reference layers such as household income or household size, and add infographics aligned to your data.

ArcGIS maps are supported in Power BI Desktop, Power BI Service, and Power BI mobile applications.

Histograms

Histograms might initially look very similar to bar charts, but there are two fundamental differences:

- Histograms differ from bar charts:
  - No spaces between the bars
  - Bars (or bins or buckets) represent a range of values
  - Ranges must be contiguous
  - Width of the bars represent a proportion of the total bin limit
- Download Histogram visual from the marketplace
- Specify a data field for the **Values** (bin) and a field for the **Frequency**

- **A histogram chart contains no spaces between the bars**. This is because each bar represents a range of data rather than a single value—for example, ages. The bars might be grouped into age ranges such as 0-17 years, 18-24 years, 25-34 years, 35-44 years, 45-59 years, and 60 years and above. These are known as bins, or buckets. The bin values are contiguous, so there are no gaps between the bars.
- **Each bar in a histogram chart is also proportionally representative in size**. Using the previous example, the 0-17 years bar is wider than the 18-24 years bar because it represents a range of 18 years inclusive, compared with the seven years (inclusive) of 18-24. Again, this requires a contiguous range of values in the buckets.

Power BI does not include a histogram chart by default; however, you can download a custom visual from the marketplace. To do this:

1. Open Power BI Desktop.
2. In the **VISUALIZATIONS** pane, click the ellipsis, and then click **Import from marketplace**.
3. In the **Caution** dialog box, click **Import**.
4. In the **Power BI Visuals** dialog box, in the search box, type **Microsoft Histogram**, and then click the search icon.
5. Click **Histogram Chart**, and then click **Add**.
6. In the **Import custom visual** dialog box, click **OK** and the new chart appears under **VISUALIZATIONS**.

Click the histogram icon, and the visual appears on your report as a watermark template. To use the histogram, provide a field for the **Values** (x-axis), and the field for aggregating in the **Frequency** (y-axis). The histogram automatically works out the bins, also known as buckets, and you can set the number of bins in the properties pane.

Alternatively, you can download other custom histogram visuals from the Power BI marketplace. Custom visuals are discussed later in this course.

Demonstration: Adding visualizations to a report

In this demonstration, you will see how to:

- Connect to a database in Azure SQL Database and import data.
- Add visualizations to a report in Power BI Desktop.

  Question: Discuss some of the charts that you could use to represent your organizational data. What types of chart would you use? Would different charts represent the data in different ways? Do you have data that would benefit from using a scatter chart, so that you can identify clusters, or outliers? Are there any missing chart types in Power BI that you might be able to download from the marketplace to fulfill your requirements?

Lesson 2
Managing a Power BI solution

This lesson discusses the management aspect of Power BI. It examines how to use the admin portal and how to configure settings in the Power BI interfaces.

Lesson Objectives

After completing this lesson, you will be able to:

- Access and use the Power BI admin portal.
- Use the Power BI service settings to manage the service environment.
- Use the Power BI Desktop settings to configure your working environment.
- Enhance reports and dashboards using additional settings in Power BI.

Power BI admin portal

The admin portal provides one location where you can view and manage administrative settings for Power BI.

- View and manage administrative settings
- Only available to:
  - Power BI service administrators
  - Office 365 Global Admins
- Includes:
  - Usage metrics: most used reports and dashboards
  - Users: link to Office 365
  - Audit logs: link to Office 365
  - Tenant settings: configure functionality for the tenant
  - Capacity settings: configure and monitor capacities

To access information in the admin portal, you must either be assigned the Power BI service administrator role or you must be a Global Admin in your organization's Office 365 team. To learn how to assign the Power BI service administrator role, see *Understanding the Power BI service administrator role* in the Power BI documentation:

Understanding the Power BI service administrator role

https://aka.ms/Miuvej

To access the admin portal, in the Power BI service, click the **Settings** gear, and then click **Admin portal**. The portal contains various sections that you can use to manage the service.

Usage metrics

The usage metrics is a dashboard displaying tiled sections:

- The first section displays usage analytics for individual users, with a count of user dashboards, reports, and datasets within the tenant. There are also a charts displaying the most consumed dashboards and packages by users and tables showing how many reports and dashboards individual users can access. This can be indicative of users who are most active within Power BI.
- The second section offers the same usage information, but is presented for groups. You see which groups are most active, in addition to the data that they are accessing. At the top of the section, there are counts for the number of group dashboards, reports, and datasets. As in the previous section, there are also charts displaying the most consumed dashboards and packages by groups and tables for the most accessible dashboards and reports.

You use this information to see how users access data and dashboards—it highlights those users and groups that are most active. Conversely, you can use this information to find out why other users are not very active in Power BI, and investigate why particular dashboards are not being used. There might be an underlying problem with the data: for example, perhaps the correct data has not been made available, or it does not cover a required period.

Users

Power BI user management is controlled by Office 365, so there are no actions available in the Power BI admin portal. The Users tab just contains a link to the Office 365 admin center. Being assigned the Power BI service administrator role does not provide any permissions to access the Office 365 admin center—therefore, you need further privileges to be assigned before you can manage the users of your Power BI service.

Audit logs

Similar to user management, audit logs are located and managed in Office 365, so the Audit Logs tab contains a link to the Security & Compliance center in Office 365. Again, you need Office 365 privileges to access this area—being a Power BI service administrator does not provide the relevant permissions.

Tenant settings

The Tenant settings enable you to configure and, in some cases, lock down, the functionality available in your tenant. For example, this is where you can disable sharing content with external users, creating app workspaces, publishing to the web, exporting information, and printing reports. For each of these actions, you can enable and disable for the whole organization. You can also enable for a subset of your organization by using security groups.

Capacity settings

This page displays the Power BI Premium capacity purchased and available for your apps. You can modify the capacities here to scale-up or scale-down resources, add and remove capacity admins, add and remove additional workloads, assign workspaces to capacities, and change regions. You can also use the Health tab to monitor workload metrics and system metrics.

Power BI service settings

Use the settings in Power BI to customize elements to behave appropriately for your needs. The options differ between the Power BI service, discussed here, and Power BI Desktop, which you'll see in the next topic. To customize the settings in the Power BI service, log in to your account at powerbi.com, click the **Settings** cog, and then click **Settings** from the menu. You can view and edit settings under the following tabs:

- General
  - Manage privacy, language, close account, enable custom visuals, and ArcGIS maps
- Dashboards
  - Show or hide the Q&A search box, and enable tile flow
- Datasets
  - View refresh history, manage connections, manage credentials, manage parameters, schedule refreshes, and configure Q&A
- Workbooks
  - Rename and delete workbooks
- Alerts
  - Turn off, rename, delete, and configure alerts

General

- **Privacy**: Microsoft automatically collects the search terms you use in Power BI, as part of their commitment to ongoing product improvement. If you do not wish to participate, clear the check box to stop sharing your search terms.

- **Language**: you can choose which language appears in the Power BI user interface and parts of the visuals. Select your relevant language from the list and click **Apply**. This resets the interface to show the chosen language, including menus, buttons, and messages. Certain features might only be available in English as the service undergoes continuous improvement and development.
- **Close account**: closing your account deletes any content you have created, and you no longer have access to the Power BI service. To close your account, optionally select the reason for closing from the list, and add any further information for Microsoft that you would like them to know about your reasons for closing the account. Click **Close account**. This option is not available if your account is managed by your organization.
- **Developer**: this setting enables developers to include visuals for testing. Turn this setting on if you want to create custom visuals for Power BI.
- **ArcGIS Maps for Power BI**: Power BI has integrated ArcGIS maps from Esri. Using the ArcGIS visual, you can create sophisticated maps and discover insights in your data that might otherwise be hidden in the standard mapping visuals—for example, by using the heat map feature.

Dashboards

You can change settings on an individual dashboard level, enabling you to control the behavior of dashboards more specifically. Under the Dashboards tab, you will find a list of all your dashboards. Click a dashboard to change the following settings:

- **Q&A**: you can toggle this setting to show or hide the Q&A search box on each dashboard. The search box is enabled by default.
- **Dashboard tile flow**: this setting ensures the dashboard content is automatically aligned to the canvas. If the setting is on, when you move a tile on the dashboard, the layout is adjusted automatically so the tile fits. This feature is turned off by default.

Datasets

Datasets are managed at the report level. On the Datasets tab, you will see a list of all the reports you have published to Power BI. You can alter the following settings for each report:

- **Refresh history**: click the **Refresh history** link to view scheduled and OneDrive data refresh history. The Start and End dates enable you to determine the length of time taken for each data refresh. The Status shows whether the refresh completed or failed.
- **Gateway connection**: use this setting to view and manage your gateway connections. This displays the status of your Power BI personal gateway—and if it is online and running. If you have other gateways connections, you can optionally switch using the toggle.
- **Data source credentials**: if you have connected to a data source that includes credentials such as a username and password, you can manage them by expanding the **Data source credentials** link.
- **Parameters**: if your query takes parameters, you can manage them here.
- **Scheduled refresh**: toggle the option **Keep your data up to date** to schedule the refresh of a dataset. You can set the refresh frequency, such as Daily, and set which time zone to use. Furthermore, you can have a notification emailed to you if the refresh fails.
- **Q&A and Cortana**: use the **Allow Cortana to access this dataset** option if you want Cortana to share the information with other Power BI users who have access to it. By default, this setting is turned off.
- **Featured Q&A questions**: You can add, edit, and delete featured questions that will be displayed as suggestions for the dataset in Q&A. This is helpful when sharing your dashboards with colleagues.

Workbooks

Workbooks are managed on an individual basis. You can rename and delete workbooks within your My Workspace area.

Alerts

Use the Alerts tab to turn off, edit, and delete your alerts that have been added from Power BI Mobile. You can rename alerts, and manage the conditions of the alert. Use this setting to change the frequency of the alert—you can also toggle between **At most every 24 hours**, or **At most once an hour**. Alerts are only sent when the data changes, and by default you receive notifications in the notification center. Select **Send me email, too** to receive alerts in your inbox.

For a list of countries and languages supported by Power BI, see *Supported languages and countries/regions for Power BI* in the Power BI documentation:

Supported languages and countries/regions for Power BI

https://aka.ms/adoke1

Power BI Desktop settings

The settings in the Power BI Desktop application differ to those in the Power BI service. You can change global settings that are applicable across the environment, reports and datasets, or adjust settings at a file level. To configure the global and file settings, in Power BI Desktop, on the **File** menu, point to **Options and settings**, and then click **Options**. The **Options** dialog box includes the following options:

- **Global**: set the data load cache; configure Power Query Editor options; configure R home directories and IDE; set global privacy; enable sending of usage data to Microsoft; turn on diagnostic tracing, and manage auto recovery options to prevent accidental loss of work
- **Current file**: manage data load options; change regional settings (locale); set privacy and auto recovery options
- **Data source**: change location of file sources; manage credentials and privacy levels: None, Private, Organizational, or Public

Global

- **Data load**: this setting enables you to manage the volume of data that is locally cached—for both query preview results and Q&A answers. The default limit for each type of cached data is 4096 MB and, although you can adjust this level up or down, Microsoft does not recommend that you reduce it below 32 MB. You can also clear the cached data and restore the cache sizes to the default setting.
- **Power Query Editor**: numerous settings are available for the Power Query Editor. You can toggle to show or hide the Query Settings pane and formula bar. You can configure the font and display setting for preview data. You can also turn on the setting to allow parameters in data sources and transformation dialogs, and enable M Intellisense in the UI.
- **R scripting**: you manage R script settings by configuring the location of your R home directories—and which R integrated development environment (IDE) to launch from within Power BI. This tab includes links to articles on installing R, and learning about the R IDE.
- **Security**: use this tab to set user approval for new native database queries. This is turned on by default. Furthermore, you can set the web preview warning level, choosing from **Strict**, **Moderate**, or **None**. Strict means the user sees a preview warning before a web preview is displayed; the Moderate option shows a warning only if the URL has not been explicitly entered, or is a trusted site; and None hides all warnings. You can also choose whether to show a security warning when adding a custom

visual to a report. This is on by default. Finally, you can view your approved ADFS authentication services, and delete unwanted entries.

- **Privacy**: use this tab to set the isolation level of your data connections to determine how they interact, if at all. By setting a higher privacy level, you can prevent data sources from exchanging data. However, this can have an impact on the functionality of your reports, and have an adverse effect on performance. You also set the privacy level at the data source and file levels, so the option you choose at a global level is affected by these lower level settings. For the highest level of security, choose **Always combine data according to your Privacy Level settings for each source**, which uses the privacy set at the data source level, and gives you the most control. The default setting is **Combine data according to each file's Privacy Level settings**, enabling you to manage security on a file basis. Alternatively, you can choose **Always ignore Privacy Level settings**, but be aware that you could potentially expose sensitive or confidential data. The hyperlink in the tab contains detailed information on setting privacy levels.
- **Usage data**: you can choose to send usage information to Microsoft to help improve the product, by sharing the features that you use. This does not disclose any personal information or data, and runs silently without affecting the performance of the application.
- **Diagnostics**: toggle **Enable tracing** to turn on the capture of diagnostic data. This is turned off by default. Use this tab to view the current version number and monthly release date of your installed application.
- **Auto recovery**: it's worth checking the auto recovery options before you begin using Power BI Desktop. This ensures that your work is saved as often as you need, in case of incidents that cause the application to close unexpectedly. The **Store auto recovery information every 10 minutes** option is selected by default. You can optionally turn this setting off, or adjust the frequency at which auto recovery information is stored—for example, to every five minutes. You can also toggle the **Keep the last auto recovery version if I close without saving** option. This useful feature is turned off by default but is certainly worth enabling to prevent any accidental loss of work. You can also change the location where Power BI auto saves your files.

Current file

- **Data load**: use this tab to configure how data is managed when connecting to, and importing from, a data source. The **Automatically detect column types and headers for unstructured sources** option is turned on by default, and helps you when importing loosely structured data. Power BI makes a best guess at the type of data in each column on import—this can be altered later in the Power Query Editor. You can configure several settings for the relationships within your data. You can turn off the default option to **Import relationships from data sources**, in which Power BI uses the foreign keys in the imported data to detect relationships between tables. The setting **Update relationships when refreshing queries** is turned off by default. This option looks for relationship changes that have occurred since the data was last imported; however, it can potentially remove any relationships you might have created manually. The **Autodetect new relationships after data is loaded** option is turned on by default and helps you find related data that might not have an existing foreign key relationship. The time intelligence setting, **Auto Date/Time**, is also turned on by default. This creates a hidden date table for each column in the dataset that has a date or datetime data type. The date table holds a set of contiguous dates, from the earliest to latest dates, enabling you to perform analysis over time. For example, you can compare sales on a date, or a date range from last year, to sales on those dates in this year. The **Allow data preview to download in the background** option is turned on by default. This is useful when connecting to a data source as you can see a preview of the data before you select it for importing. You can turn this off if you are connecting to a very large dataset and don't need to see the data values before you import the data. Another setting

that is automatically turned on is **Enable parallel loading of tables**. This enables Power BI to simultaneously load data from multiple sources, though be aware that it can affect performance.

- **Regional settings**: use this tab to set the **Locale** of the current file. The language setting you choose affects how numbers, dates, and time from imported data are interpreted. For example, you can select English from United States, United Kingdom, or Australia, or select French for France, Luxembourg, or Monaco.
- **Privacy**: you can edit the permissions for each connection in your files to work with or override the settings at the global level. Choose from the default **Combine data according to your Privacy Level settings for each source** (this is discussed in the section below), or **Ignore the Privacy Levels and potentially improve performance**. The last setting has the potential to expose your data to unauthorized users, so use caution if working with sensitive or confidential data.
- **Auto recovery**: the **Disable auto recovery for this file** option overrides the global setting and is turned off by default. Only turn this on if you are sure you can afford to lose work in the case of unexpected application behavior.

Data source settings

To manage settings for the data sources in the current file and globally, on the **File** menu, point to **Options and settings**, and then click **Data source settings**. If you have imported data from a document such as Excel, CSV, or Access, you can change the file location of the source, and add additional file parts. You can also manage how the file is opened—for example, if you import a text file, you can choose that this file is opened with Excel.

As previously discussed, you can configure privacy options at the file and data source level. Use **Edit Permissions** to set the privacy level for each data source, choosing from **None**, **Public**, **Organizational**, or **Private**. By default, data sources are set to None, so if your data contains sensitive or confidential data, ensure you change it to Private. The Private setting isolates the data source from other sources, and is useful if you want to restrict access to authorized users only. Data sources that only need to be visible within a trusted group of people can use the **Organizational** setting. This isolates the data source from all Public data sources, but enables visibility to other Organizational data sources. Only data that is freely available, such as that on a public website, or from a data marketplace, should be secured as **Public**, because the source becomes visible to everyone. If your data sources use credentials, you can also manage them on the **Edit Permissions** dialog.

Note: Changing options might require Power BI Desktop to be restarted before they can take effect.

Dashboard and report settings

Dashboards and reports include settings that help you to work with your data more efficiently, in addition to presenting options for printing and exporting data, and sharing it through a variety of media.

- Customize reports and dashboards using settings:
  - **Filter dashboard**: use Focus mode to open a tile, then apply filters to categories and values
  - **Featured questions**: add custom featured questions, and enable results in Cortana and Q&A
  - **Print dashboard and export questions**: print a hard copy of a report or dashboard; export dashboard data to CSV, or report data to XLSX or CSV
  - **Publish to web**: create URL to send in email or social media; use embed code in webpage or blog post
  - **Custom URL and title**: change the default destination when clicking on a tile to a URL; alter title and subtitle

Filtering a dashboard

When you view a dashboard in the Power BI service, unlike reports, filters are not immediately available to you. However, you can filter on individually pinned tiles. If you click **More options** in the top right corner of a tile, and then click **Open in focus mode—**the tile opens so it is the only one in view. You can then expand the **FILTERS** pane, to filter on categories in the chart, and set criteria on values. For example, you can filter on products, or show data where sales are above $150 million and less than $250 million.

Featured questions

Q&A enables you to ask natural query language questions of your data. There are a couple of features in your reports and dashboards that you can use to enhance the results returned to Q&A. When you click **Ask a question about your data**, the Q&A box expands to include suggestions from the fields, calculations, and measures in the data model, to help you get started. You can add suggestions to this list to enable other users to quickly find the answers they need, using the following steps:

1. Click **My Workspace**. In the **Dashboards** list, on the dashboard you want to configure, click **Settings**.
2. Under Q&A, use the toggle to enable Q&A.
3. Click **Save**.
4. In the top right corner of the screen, click **Settings**, and choose **Settings** from the menu.
5. Click **Datasets**, and select the data you want to configure from the list.
6. Click **Featured Q&A questions** to expand the list of available questions.
7. You can delete existing questions, or click **Add a question** to create a new question. This opens a new text box. Type your question, and click **Apply** after adding all the questions you want to include.
8. Click **My Workspace**. Select the dashboard to which you have added one or more Featured Questions. Click **Ask a question about your data**, and the Featured Questions you have just added now appear at the top of the list of suggestions. Click a question to see the results.

You can also add keywords and filtering to a report to guide users to find the exact data they need when asking a question:

1. From My Workspace, click the report you want to alter, and then on the menu bar, click **Edit report**.
2. Drag a column from the **FIELDS** pane to **Page level filters**—for example, **Manufacturer**.
3. Click **Require single selection** so the page is displayed in results in Cortana or Q&A.
4. Click **Format**, then click **Page information**, and toggle **Q&A** to **On**. This prompts Q&A to use the report if a user asks a question related to the data in the report.

5. In the text box, type in alternative names, or terms that users might type when asking questions, such as **manufacturer performance**. Separate each phrase with a comma.
6. Click **Save**.

When you, or another user, search using a combination of the phrase and the filter, more accurate results can be returned. For example, in the dashboard's Q&A box, type **manufacturer performance** and Power BI returns a list of all the manufacturers from the filter. You can select a filter and the results are displayed just for that manufacturer.

Printing a dashboard and exporting questions

Some occasions might demand a hard copy of a report or the underlying data, and Power BI offers print and export functionality to support this. Open the dashboard you want to print, on the **File** menu, click the ellipsis in the top right corner of the screen, and then click **Print dashboard**. This opens the standard print dialog where you can set the page layout, paper size, quality, and other printer options. After setting additional options, click **Print**. To print a report, open it and then on the **File** menu, click **Print**.

You can export the data for any visual within Power BI. Open the relevant report, click the ellipsis in the top right corner of the visual, and then click **Export data**. You can then choose to export **Summarized data** or **Underlying data**. In addition, you can choose to export to Excel format (.xlsx) or .csv. The data is exported with any filters that have been applied to generate the results in the visual. You can now ask a question of your data, view the results, and then export them.

Publish to web

You use the Power BI Publish to web feature to share a report by creating a URL you can send through email or social media, or embedded code that can be used in webpages, or blog posts. You can also edit, update, refresh, and unshare visuals you choose to publish. The Publish to web feature is available in reports that you can edit in your personal or group workspace. You cannot publish reports that were shared with you, or that use row level security to secure the data. To publish a report to the web, use the following steps:

1. From My Workspace, open the report you want to use, and then on the **File** menu, click **Publish to web**.
2. Review the information in the displayed dialog, and click **Create embed code**.
3. A warning is shown asking you to confirm that the data can be made public. If you agree, click **Publish**.
4. You are then shown a link to use in an email, and HTML iframe code that can be pasted directly into a webpage or blog. Optionally, you can use the sizing list to ensure your report displays in the best possible way to present the data. The default is 800 x 600 pixels. After copying and pasting the text, you can change the height and width values as required.

Each report has a single embed code. Click **Settings**, then click **Manage embed codes** to see a list of reports for which a code has been generated, and the time and date of generation. To copy the code, or delete the code, click the ellipsis next to a report. If you delete a code, any webpages that embed the report, will no longer be able to display it. This feature also supports custom visuals.

Note: Use the Publish to web feature with caution, because making your data publicly available allows anyone to view it. There is no inclusion of authentication, so always check that the data you publish is not sensitive.
This feature can be turned off by administrators who have access to the admin portal. Go to **Tenant Settings**, and set **Publish to web** to **Disabled**. This affects the current tenant.

For more detailed information, and the limitations of this feature, see *Publish to web from Power BI* in the Power BI documentation:

Publish to web from Power BI

https://aka.ms/eq7ft3

Custom URL and title

When you click a tile in a dashboard, by default you are taken to the original visual in the report from which it was pinned. You can change this behavior to point to another location. Open the dashboard and click the ellipsis in the top right corner of the tile you want to amend. Click **Edit details**. In the **Tile details** pane, select **Set custom link**, and type or paste a web address into the **URL** text box. Choose whether to open the custom link in the same tab using the radio buttons.

Additionally, you can change the title and subtitle for the tile by overwriting the text in the **Title** and **Subtitle** text boxes. It is also useful to show the date and time the item was last refreshed, using the check box to enable this addition to the tile. When this is done, click **Apply**. You can use this feature so a company logo directs the user to the main corporate home page, or perhaps other pages in the corporate site that contain more detailed information relevant to the visual.

Demonstration: Creating featured questions

In this demonstration, you will see how to:

- Add featured questions to a Power BI dashboard.
- Use featured questions to enhance the user's experience when using dashboards.

Check Your Knowledge

Question	
Which of the following statements about the Manage Data portal is false?	
Select the correct answer.	
	The portal enables you to manage your shared queries.
	You can edit and control access to your data sources in the portal.
	Using the portal enables you to delete data sources that you no longer need.
	The Usage Report shows how many times a dashboard was consumed in Power BI.
	The Usage Report displays the most active groups in Power BI.

Lab: Creating a Power BI report

Scenario

Adventure Works employees are increasingly frustrated by the time that it takes to implement managed BI services. The existing managed BI infrastructure, including a data warehouse, enterprise data models, and reports and dashboards, are valued sources of decision-making information. However, users increasingly want to explore relationships with other, currently unmanaged data, and it takes too long for the IT department to include these requirements in the corporate BI solution.

As a BI professional, you are asked to explore ways in which Adventure Works can empower business users to augment their managed enterprise BI solution with self-service BI.

Objectives

After completing this lab, you will be able to:

- Connect, shape, and combine data in Power BI.
- Create a report by using chart and map visuals.
- Publish reports and share dashboards.

Note: Because of updates to Microsoft Power BI, the lab steps for this course change frequently. Microsoft Learning regularly updates the lab steps, so they are not available in this manual – but you can access them on GitHub.

Lab Setup

Estimated Time: 60 minutes

Virtual machine: **20778C-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa55w.rd**

All the lab steps are contained in 20778C_LAB_06.md.

Exercise 1: Connecting to Power BI data

Scenario

You have decided to explore the features in Power BI because you believe that they offer the best solution to enable business users to create self-service BI solutions. To convince the business users that this is the best option, you will build a sample report to demonstrate the capabilities of the features in Power BI. You will create reports in Power BI Desktop by using corporate data that is stored in a database in Azure SQL Database. After importing the data, you will shape the data by using the Power BI transformation tools. You will then combine the data by merging columns and appending rows.

The main tasks for this exercise are as follows:

1. Prepare the environment
2. Connect to existing data in Azure
3. Shape data
4. Combine data

- ▶ **Task 1: Prepare the environment**
- ▶ **Task 2: Connect to existing data in Azure**
- ▶ **Task 3: Shape data**
- ▶ **Task 4: Combine data**

Results: After this exercise, you should have imported data from Azure, shaped it by using the Power BI transformation tools, and combined the data by merging columns and appending rows.

Exercise 2: Building Power BI reports

Scenario

You are happy that Power BI can import the data that you require and shape it, so you have decided to add visualizations to the report to display the data. After creating the report, you will show its capabilities to your senior managers to convince them that Power BI is a suitable platform for adopting self-service BI within your organization.

The main tasks for this exercise are as follows:

1. Create a chart
2. Create a map visualization

- ▶ **Task 1: Create a chart**
- ▶ **Task 2: Create a map visualization**

Results: After this exercise, you should have created a report that has chart visuals and is ready to publish to the Power BI service.

Exercise 3: Creating a Power BI dashboard

Scenario

The reports that you have created are ready to present to senior management. However, you have decided that you will first publish them to the Power BI service, to demonstrate its true potential by creating a dashboard.

The main tasks for this exercise are as follows:

1. Publish reports from Power BI Desktop
2. Create a Power BI dashboard

- ▶ **Task 1: Publish reports from Power BI Desktop**
- ▶ **Task 2: Create a Power BI dashboard**

Results: After this exercise, you should have published a report to the Power BI service and used the visuals to create a dashboard.

Question: Discuss the tools that you used to shape and combine data in the labs. How did this compare to using Excel, or coding Transact-SQL to deliver the same results? Do you think it is quicker to use Power BI rather than the applications that you currently use?

Question: Discuss some of the visualizations that you used in the optional exercise to create a report that was relevant for your organization. If you did not have time to do the optional exercise, which of the charts that you used in the lab will you reuse to create reports for your organization? Can you think of data that you can present by using the map charts?

Module Review and Takeaways

In this module, you have learned how to enhance your Power BI charts by using interactive data visualizations, and how to manage your Power BI solutions.

Review Question(s)

Question: Why do you think the Manage Data portal prevents you from deleting data sources? Do you agree with this, or should you be able to delete the data sources for the queries that you have shared?

Module 7

Direct Connectivity

Contents:

Module Overview

Power BI service supports live direct connections to Azure SQL database, Azure SQL Data Warehouse, big data sources such as Azure HDInsight® Spark, and SQL Server Analysis Services. DirectQuery means that whenever you slice data or add another field to a visualization, a new query is issued directly to the data source.

Power BI works with SQL Server Analysis Services models that are running in multidimensional mode, so that you can use OLAP cubes and models in reports and dashboards. It doesn't matter if you are using the Power BI service in the cloud, and an on-premises SQL Server Analysis Services implementation; the On-premises data gateway enables live connections between the cloud and on-premises data servers.

Objectives

After completing this module, you will be able to:

- Use Power BI direct connectivity to access data in Azure SQL Database and Azure SQL Data Warehouse, in addition to big data sources, such as Hadoop.
- Use Power BI with SQL Server Analysis Services data, including Analysis Services models running in multidimensional mode.

Lesson 1
Cloud data

In this lesson, you will learn how to use Power BI to directly connect to Azure SQL Database and Azure SQL Data Warehouse, and then use these datasets with visualizations, reports, and dashboards. You will then learn how Power BI works with big data sources, including Hadoop and Spark.

Lesson Objectives

After completing this lesson, you will be able to:

- Use direct connectivity in Power BI to access data in Azure SQL Database and in Azure SQL Data Warehouse.
- Connect Power BI to big data sources and use these sources with BI reports and visualizations.

Direct connectivity to SQL Services in Azure

You use Power BI to connect to your cloud-based instances of SQL Server as easily as connecting to your on-premises servers.

Before connecting to a database in Azure, ensure that you have configured the firewall settings in Azure to allow remote connections:

1. In Microsoft Azure, click **SQL databases**, and then click the name of the database to which you want to grant access.
2. Click **Set server firewall**, click **Add client IP** to add your current workstation, or add a range of IP addresses, and then click **Save**.

- Set firewall settings in Microsoft Azure to allow connections at server
- Importing data
- Using DirectQuery:
  - Good for very large datasets
  - Always current data
  - Single database

Note: Microsoft recommends that you allow access at the database level in Azure, rather than at the server level.

You can either import the data into Power BI Desktop or use DirectQuery to create a live connection to the data. DirectQuery restricts you to using a single database, but it is useful when you want to connect to very large datasets that could take a long time to load into Power BI. Loading data can also be problematic when making changes to report items that cause a refresh of the data—this can mean further delays and make it cumbersome to work with the data.

Importing data

To import data into Power BI Desktop:

1. Open Power BI Desktop, and then click **Get Data**.
2. In the **Get Data** dialog box, click **Azure**, click **Azure SQL database** or **Azure SQL Data Warehouse**, and then click **Connect**.

3. In the **Server** name box, type or paste the full name of the server—for example, ***<server name>*.database.windows.net**—then optionally in the **Database (optional)** box, type the name of the database. If you have previously created a parameter, Power BI gives you the option of using a parameter value for the server and database names.
4. Type or paste an optional query into the **SQL Statement (optional)** box, and then click **OK**.
5. If you did not specify a database name in the previous step, the Navigator screen displays a list of available databases; otherwise, it just shows the database that you specified. Expand a database to view the objects in it, and then click to select the tables and views that you want to import. You can select objects from multiple databases to combine the data into a single dataset.
6. Click **Load** to import the data into Power BI, or click **Edit** to open the Power Query Editor window and apply transformations. Transformations can also be performed at any point after loading the data.

Using DirectQuery

To connect using DirectQuery:

1. Open Power BI Desktop, and then click **Get Data**.
2. In the **Get Data** dialog box, click **Azure**, click **Azure SQL database** or **Azure SQL Data Warehouse**, and then click **Connect**.
3. In the **Server** name box, type or paste the full name of the server—for example, ***<server name>*.database.windows.net**—then optionally in the **Database (optional)** box, type the name of the database. If you have previously created a parameter, Power BI gives you the option of using a parameter value for the server and database names.
4. Under the **Data Connectivity** mode, click **DirectQuery**.
5. Type or paste an optional query into the **SQL Statement (optional)** box, and then click **OK**.
6. If you did not specify a database name in the previous step, the Navigator screen displays a list of available databases; otherwise, it just shows the database that you specified. Expand a database to view the objects in it, and then click to select the tables and views that you want to import. You select objects from multiple databases to combine the data into a single dataset.
7. Click **Load** to create the DirectQuery connection, or click **Edit** to open the Power Query Editor window and apply transformations. You can perform transformations at any point after loading the data.

Note: The Power BI Q&A natural language feature is not available when using DirectQuery. Q&A uses the data that is imported into datasets to build answers and cannot create this without the data being present.

After creating a report by using DirectQuery, you can publish to the Power BI service. You might need to provide credentials for the database in Azure SQL database to run the report. To provide credentials:

1. In Power BI, click the **Settings** gear icon, and then on the menu, click **Settings**.
2. Click the **Datasets** tab, and then click the dataset that connects to the database in Azure SQL Database by using DirectQuery.
3. Expand **Data source credentials**, click **Edit Credentials**, and then add your user name and password.

You cannot connect to Azure SQL database or Azure SQL Data Warehouse from the Power BI Service. You must connect from Power BI Desktop and then publish the report to Power BI Service.

Connecting to big data

- Connecting to HDFS:
  - Add IP address and host name of Hadoop cluster to host file
  - Connect using the fully qualified name of the Hadoop server or cluster, such as <*server name*>.cloudapp.net
- Connecting to Spark:
  - Connect using the fully qualified name of the server; for example, <*clustername*>.azurehdinsight.net

Big data describes data sets that are typically too large and complex to process using standard techniques such as data cubes, denormalized relational tables, and batch-based extract, transform and load (ETL) engines. Instead, you should use other approaches—Hadoop has become the standard for distributed data processing of big data. You can either run your own Hadoop servers and clusters, or use a hosted Hadoop service, such as HDInsight or Hortonworks Data Platform (HDP). HDInsight is an Apache Hadoop implementation based on HDP, which runs in globally distributed Microsoft datacenters. You use the HDInsight service to build Hadoop clusters easily and quickly when you need them.

There are several ways to use Power BI to connect to big data sources, and to use Power BI reports and visualizations with big data.

Connecting to HDFS

If you have an Azure virtual machine running Hadoop, or are using a Hortonworks Sandbox (if you don't have access to a Hadoop cluster), you can connect to the Hadoop Distributed File System (HDFS) for reporting with Power BI Desktop:

1. In Power BI Desktop, click **Get Data**.
2. In the **Get Data** dialog box, click **Hadoop File (HDFS)**, and then click **Connect**.
3. In the **Server** box, type the name of your server, and then click **OK**.

Note: To avoid potential name resolution problems, you should add the IP address and host name of the Hortonworks or Hadoop cluster details to the host file of the computer running the queries.

Connecting to Spark

Azure HDInsight provides a fully managed Spark service. Apache Spark is an open-source parallel processing framework that supports in-memory processing to boost the performance of big data analytic applications. This capability allows for scenarios such as iterative machine learning and interactive data analysis.

You use Power BI Desktop to connect directly to your Spark cluster then explore and monitor data without requiring a data model as an intermediate cache. It's a live connection, so any field selection or filter sends a query back to the source and the visual is updated with the new results. After saving your report, any of the visuals can be pinned to your customized dashboard. The data in the dashboard will be refreshed approximately every 15 minutes—no refresh schedule is required.

To connect to Spark in Power BI desktop:

1. In Power BI Desktop, click **Get Data**.
2. In the **Get Data** dialog box, click **Azure HDInsight Spark**., and then click **Connect**.
3. In the **Server** box, type or paste the fully qualified name of the server, such as**<*clustername*>.azurehdinsight.net**, and then click **OK**.

4. On the **Username/Password** page, enter your credentials, and then click **Connect**.
5. Select the newly created Spark dataset to begin exploring the data. Note that every field selection will generate a query back to the source data so, depending on the size of the query and any database optimizations, there might be some loading indicators while the visuals are created.

Demonstration: Using Azure SQL Database as a Power BI data source

In this demonstration, you will see how to:

- Import data from tables in a database in Azure SQL Database.
- View relationships between the tables.

Lesson 2
Connecting to Analysis Services

In this lesson, you will learn how to use Power BI Desktop to connect to a local SQL Server Analysis Services server, and then use the results in visualizations and reports. You will also learn how to access on-premises SQL Server Analysis Services data from the Power BI service in the cloud, through the On-premises data gateway. Finally, you will learn how to use Power BI with SQL Server Analysis Services models that are running in multidimensional mode, and how to use OLAP cubes and models in reports and dashboards.

Lesson Objectives

After completing this lesson, you will be able to:

- Use Power BI Desktop to access SQL Server Analysis Services data.
- Use the Power BI service, and the On-premises data gateway, to access on-premises SQL Server Analysis Services data.
- Use Power BI Desktop to connect to SQL Server Analysis Services models in multidimensional mode.

Direct connectivity to Analysis Services

SQL Server Analysis Services (SSAS) supports additional security options, including role-based security. For example, you may have Finance users who should only have access to a particular set of information in a dataset, and Sales users who need access to a slightly different set of data. These roles are managed in SSAS, and Power BI applies the SSAS security so that users only see the data that they are permitted to access. This delegation applies whether you are using the Power BI Desktop, or whether you are using a report that has been published to the Power BI service.

- Connect to SSAS from Power BI Desktop or service:
  - Connect to on-premises SSAS 2012 or later tabular models from Power BI Desktop and Power BI service
  - Use a live connection to connect to tabular models
- Power BI Desktop:
  - Can also connect to multidimensional models
  - Can import data from tabular or multidimensional models
  - Use a live connection for multidimensional models

You connect to an on-premises tabular model database in SQL Server Analysis Services (SSAS) from both Power BI Desktop and the Power BI service. You also have the option of connecting to SQL Server Analysis Services by using Excel®, and then uploading the workbook. By using Excel, you can explore and edit your tabular data in Power BI. In Power BI Desktop, you can also connect to multidimensional models in SQL Server Analysis Services.

Note: You must configure a Power BI gateway on your server before you connect to SSAS using a live connection from the Power BI service.

Connecting from Power BI Desktop

To connect to a database in SQL Server Analysis Services from Power BI Desktop:

1. Click **Get Data**, click **SQL Server Analysis Services database**, and then click **Connect**.
2. In the **Server** box, type the name of the server, and then optionally in the **Database (optional)** box, type the name of the database. If you have previously created a parameter, Power BI gives you the option of using a parameter value for the server and database names.
3. Use the **Connect live** or **Import** button to specify the type of connection you want to create. You also have the option to enter Multidimensional Expressions (MDX) code or a Data Analysis Expressions (DAX) query. Click **OK**.
4. In the next dialog box, enter your credentials, and then click **Connect**.
5. Click to add dimensions and measures from the list of available objects.
6. Click **Load** to create the dataset and import the data if you previously selected this option, or click **Edit** to open the Power Query Editor window and apply transformations. You can also edit the dataset later, after loading it.

Using the On-Premises data gateway

- Two types of gateway:
  - On-premises data gateway
  - On-premises data gateway (personal mode)
- Download the gateway from:
  - Link in Power BI service
  - Power BI website
- Using the gateway:
  - Add data sources and users to the gateway
  - Connect to SSAS to see the registered Analysis Services models

You can connect directly to SQL Server Analysis Services from Power BI Desktop, but if you want to upload a report file and start using it the Power BI service, you need to download and install the On-premises data gateway (previously called the Power BI Analysis Services connector). When the gateway is set up, it acts in a similar way to the personal gateway, providing a connection between your on-premises Analysis Services and the Power BI service. There is a single download and installer for both the On-premises data gateway and the Personal gateway. For information about the On-premises data gateway, see *On-premises data gateway* in the Power BI documentation:

On-premises data gateway

https://aka.ms/iddied

Two types of On-premises data gateway are available:

- **On-premises data gateway**. Multiple users can share and reuse a gateway in this mode. This gateway can be used by Power BI, PowerApps, Logic Apps, or Microsoft Flow. For Power BI, this includes support for both schedule refresh and DirectQuery.
- **On-premises data gateway (personal mode)**. This is for Power BI only and can be used by an individual without any administrator configuration. This can only be used for scheduled refresh.

Installing the gateway

To download and install the gateway:

1. From the Power BI service, on the **Downloads** menu, click **Data Gateway**. The gateway should be installed on a machine that can be constantly left running. The gateway is only supported on 64-bit Windows operating systems.
2. Click **DOWNLOAD GATEWAY**, and then run the installer.
3. Select the mode of the gateway to use, and then click **Next**.
4. On the reminder page, click **Next**.
5. On the getting ready page, select an installation folder. Review the terms of use and privacy statement, and then if you accept them, select the box, and then click **Install**.
6. On the **Installation was successful!** page, enter your Azure credentials and sign in.
7. On the register page, enter a name and recovery key for the new gateway, and then click **Configure**.
8. When the message confirming that the gateway is online appears, click **Close**.

Note: If you install the gateway in personal mode, you cannot install another gateway on the same machine.

After installing the data gateway, if it is for Power BI, log in to the Power BI service and add your data sources to the gateway to register the models in the Power BI service.

1. In Power BI service, on the **Settings** menu, click **Manage gateways**.
2. On the **Gateway Cluster Settings** page, click **Add data sources to use the gateway**.
3. On the **Data Source Settings** page, enter a name for the source and choose a type.
4. Enter the data source type specific information, and then click **Add**.

Installing and configuring a gateway is usually done by an administrator. It might require specialist knowledge of your on-premises servers and, in some cases, may require Server Administrator permissions. A Power BI Pro license is required to use the gateway. For more information about the gateway, see *On-premises data gateway in-depth* in the Power BI documentation:

On-premises data gateway in-depth

https://aka.ms/qkkwfu

Using the gateway

To use the On-premises data gateway to access SSAS from the Power BI service:

1. In the Power BI service, click **Get Data**, click **Databases**, and then click **SQL Server Analysis Services**.
2. Click **Connect** to see a list of all the Analysis Services models that were registered when the gateway was configured. If there are no servers listed here, it means either that the gateway and data source are not configured, or that your account is not listed in the **Users** tab of the data source, in the gateway.

When you select one of the tabular models that are available on the SSAS machine and click **Connect**, a dataset is added for you to use within the Power BI service; this is a pointer to the Analysis Services model. When you open this dataset in Power BI, the list of tables that are available in the model is shown in the right pane.

You can now build your visuals in the normal way, but by working on live data from the Analysis Services computer. You can save your work as an Analysis Services Report and pin the report to a dashboard. The gateway passes credentials to Analysis Services so that, if this dashboard is shared with other users in your organization, they will only see the data that they are permitted to access from within Analysis Services.

Note: If you pin visuals from a report to the dashboard, the pinned tiles are automatically refreshed every 10 minutes. So, if data in your on-premises Analysis Services server is updated, the tiles will get auto-updated within this 10-minute period.

SSAS multidimensional models

You can also use the Power BI Desktop to connect to SQL Server Analysis Services models that are running in multidimensional or OLAP mode. This feature of the Power BI Desktop is commonly referred to as SSAS MD.

- SQL Server Analysis Services multidimensional mode (SSAS MD)
  - Supported in Power BI service and Power BI Desktop
- For direct connection to an OLAP cube, select option to **Connect live**
- Browse databases, and select cubes, models, or perspectives
  - Perspectives show preview of available dimensions or measures
  - Cubes show KPIs that are defined in the cube

Note: SSAS multidimensional models in live connection mode are supported in both the Power BI service and in Power BI Desktop. You can also publish and upload reports that use SSAS multidimensional models in live mode to the Power BI service.

To connect to a multidimensional model from Power BI Desktop:

1. Click **Get Data**, and then click **Analysis Services**.
2. In the **Server** box, type the name of the server that is running a multidimensional model.
3. Ensure that the option to **Connect live** is selected, and click **OK**.
4. You can now browse the databases and the cubes, models, or perspectives that are available to you on that server. When you connect to a perspective, you get a preview of the dimensions or measures that are available—click **OK** and the fields list is populated. You are not importing any data into the Power BI Desktop; in the bottom right corner, you'll see that you are using a direct connection and that you are connected to an OLAP cube.

You can now build visualizations in the same way as you would do with any other data source; the only difference is that you are sending queries to the multidimensional cube every time you make a change to a visual. You can also use KPIs that are defined in the cube, and visuals display an indicator showing you where the KPI is in relation to a target. If you then break data down by category, you can get status indicators for each category. For more information about connecting to multidimensional models, see *Connect to SSAS multidimensional models in Power BI Desktop* in the Power BI documentation:

Connect to SSAS Multidimensional Models in Power BI Desktop

https://aka.ms/xfumh5

Lab: Direct connectivity

Scenario

Adventure Works employees wish to extend the scope of their business intelligence (BI) activities, and include cloud-based data sources that are hosted in Azure. These employees would like live connections to Azure SQL Database and Azure SQL Data Warehouse. They want to be able to make these connections whether they are using the Power BI Desktop or the Power BI service.

As a BI professional, you have been asked to create a report in the Power BI Desktop that uses DirectQuery pull data from the AdventureWorks data sources in Azure SQL Database. You have also been asked to ensure that this information is made available from the cloud, by publishing this desktop report to the Power BI service.

Objectives

After completing this lab, you will be able to:

- Configure a live connection from the Power BI Desktop to an Azure SQL Database, by using DirectQuery.
- Publish a desktop report that includes a DirectQuery to an Azure SQL Database, for use from the Power BI service.

Note: Because of updates to Microsoft Power BI, the lab steps for this course change frequently. Microsoft Learning regularly updates the lab steps, so they are not available in this manual – but you can access them on GitHub.

Lab Setup

Estimated Time: 60 minutes

Virtual machine: **20778C-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa55w.rd**

All the lab steps are contained in 20778C_LAB_07.md.

Exercise 1: Direct connections in Power BI

Scenario

As a data analyst for AdventureWorks, you are investigating the use of live connections to Azure SQL Database and Azure SQL Data Warehouse.

In this exercise, you will create a Power BI Desktop report and use DirectQuery to pull data from the AdventureWorks database hosted in Azure. You will then publish this report to the Power BI service, so that this information is also available for cloud use.

The main tasks for this exercise are as follows:

1. Prepare the lab environment
2. Direct connectivity from the Power BI Desktop
3. Direct connectivity from the Power BI service

- **Task 1: Prepare the lab environment**
- **Task 2: Direct connectivity from the Power BI Desktop**
- **Task 3: Direct connectivity from the Power BI service**

Results: At the end of this exercise, data from the AdventureWorks Azure SQL Database will be available for use in Power BI Desktop and in a desktop report that has been published to the Power BI service.

Question: Discuss the different online data sources that your organization could use to create Power BI reports. Can you think of a scenario where users perhaps have Azure SQL database for one set of reports, and data in another online database for another set of reports? Could this be combined into a single dataset in Power BI?

Question: Discuss the issues to consider as you decide whether to import data or use DirectQuery when building reports against large online databases. Ask students about their own organizations—ask how they would make such a decision.

Module Review and Takeaways

In this module, you learned how to:

- Use Power BI direct connectivity to access data in Azure SQL Database and Azure SQL Data Warehouse, in addition to big data sources, such as Hadoop.
- Use Power BI with SQL Server Analysis Services data, including Analysis Services models running in multidimensional mode.

Review Question(s)

Question: Discuss the different ways in which your organization could use Power BI to connect to online data sources. What would be some of the potential benefits of direct connectivity to services such as Azure SQL Database? Are there any scenarios in your organization that could use the On-premises data gateway?

Module 8

Development with Power BI

Contents:

Module Overview

The Power BI™ API is a REST-based API that developers use to access programmatically datasets, tables, and rows in Power BI. Using this API, you push data from an application into Power BI and integrate Power BI visualizations into an application. You can also add custom visuals to your applications and to Power BI dashboards and reports.

In this module, you will learn how to use the Power BI API to embed content in your applications and how to use custom visuals in your reports.

Objectives

After completing this module, you will be able to:

- Describe and use the Power BI API.
- List the steps for creating custom visualizations and import custom visuals into Power BI for use in Power BI reports.

Lesson 1
The Power BI API

At times, you may want to include Power BI reports, tiles, dashboards, and Q&A in other applications. For example, in your internal applications such as SharePoint Online, Microsoft Teams, and Dynamics 365, in custom applications within your organization, or in applications that you develop for external customers. Power BI supports all of these scenarios by using the Power BI API.

In this lesson, you will learn how to use the Power BI API in applications, to push data into Power BI, add data visualizations into applications, and customize visualizations. You will also see how to use the Power BI Embedded Playground to learn about the API and test your own applications.

Lesson Objectives

After completing this lesson, you will be able to:

- Describe the key tasks that can be completed using the Power BI API.
- Describe how to embed Power BI content in your applications.
- Embed content for your organization using the Power BI API.
- Embed content for your customers using the Power BI API with Power BI Embedded.
- Use the Power BI Embedded Playground to learn about the Power BI API.

What is the Power BI API?

The Power BI API is a set of REST APIs from the Power BI Service platform that enables you to interact with Power BI from your applications. Developers use the API to push data into Power BI, display visuals from Power BI, and retrieve metadata about the objects in the service. This enables in-house developers and independent software vendors (ISVs) to integrate Power BI visuals, such as reports, tiles, dashboards, and Q&A content, into their own applications.

- REST APIs to interact with Power BI
- Three forms:
  - Power BI REST API
  - Power BI .NET SDK
  - Power BI JavaScript API
- Typical tasks performed using Power BI APIs:
  - Pushing data to a Power BI dashboard
  - Embedding tiles into an app
  - Embedding reports into an app
  - Importing Power BI Desktop (PBIX) files
  - Authenticating Power BI web apps

You use the Power BI API in three different forms:

- Power BI REST API – a REST API that you use from any programming language that supports REST calls.
- Power BI .NET SDK – a wrapper to the Power BI API that you use from .NET languages to simplify your calls to the API.
- Power BI JavaScript API – a client-side library that you use to interact with your embedded content; for example, to filter a Power BI report displayed in your application.

In all of these scenarios, Power BI retains control of the authentication and authorization of the content, ensuring that unauthorized users can't access your data. You can even take advantage of the security features in Power BI, such as row level security.

Typical tasks performed using Power BI APIs include:

- Extending existing business workflows to push key data into a Power BI dashboard.
- Embedding tiles into an app.
- Embedding reports into an app.
- Importing Power BI Desktop (PBIX) files.
- Authenticating Power BI web apps.

Embedding Power BI content in your applications

You can embed Power BI content into websites and applications designed for internal use and for external customers.

- For your organization:
  - Configure embedding in SharePoint Online, Microsoft Teams, and Microsoft Dynamics
  - Use Power BI APIs to embed in custom applications, accessing content as the user
- For your customers
  - Use Power BI APIs with Power BI Embedded to embed in custom applications, accessing content with a master account

For your organization

You can easily embed Power BI content into various Microsoft systems, including SharePoint Online, Microsoft Teams, and Microsoft Dynamics 365. Embedding in these systems does not require any coding; you just create your Power BI content then configure the application to display it.

- **SharePoint Online**. Publish your report to your Power BI account, get the URL for that report, and then use the URL for a Power BI web part on your SharePoint Online page.
- **Microsoft Teams**. Add Power BI as a tab in your Microsoft Teams channel, and then select which reports to show in that channel.
- **Dynamics 365**. Enable Power BI visualization embedding in Dynamics 365, and then add a Power BI dashboard or tile to your Dynamics 365 dashboard.

You can also embed Power BI content into your own custom internal applications using the Power BI API. In this scenario, users utilize their existing Power BI accounts to access the content on the Power BI service they own or one that has been shared with them. They use the standard Power BI login procedure that returns an auth token that your application uses to access their data and reports as that user.

For your customers

When embedding content in applications for external customers, you're unlikely to know whether they have Power BI licenses. Even if they have licenses, they likely won't have access permissions on your content or data. In this scenario, you use the Power BI APIs in conjunction with Power BI Embedded to integrate reports, tiles, and dashboards seamlessly into your applications. Your users can work with these objects in your application without needing access to, or even knowing anything about, Power BI.

Your application accesses Power BI using a master account that has admin rights to the specific content that you want to embed. You can think of this account as a proxy account through which all users of your application will access Power BI content.

Power BI Embedded is a Microsoft Azure service that enables Power BI access using a capacity-based, hourly metered model. Using Azure capacity means that you can easily scale-up or scale-down resources and control the billing of your application.

Embedding for your organization

To embed Power BI content for your organization, you need to configure the application within Azure Active Directory to allow it to call the Power BI API. Then you set up the Power BI content to make it accessible to your application, and finally write code in your application to embed the content.

1. Register your application with Azure Active Directory at http://dev.powerbi.com/apps:
   - Server-side web application
   OR
   - Client-side native application
2. Make Power BI content accessible to your application
3. Code your application
4. If required, assign your app workspace to dedicated capacity

Register your application with Azure Active Directory

To register your application, perform the following steps:

1. In a web browser, go to **http://dev.powerbi.com/apps**.
2. Sign in using your Power BI credentials.
3. Specify the following, and then click **Register**:
   a. **Application Name** – to identify your application in Azure.
   b. **Application Type** – server-side web application or client-side native application.
   c. **Home Page URL** and **Redirect URL** – homepage and landing page after sign-in (for server-side web applications).
   d. **API access** – which API calls your application will use.

The **Success** dialog box returns an **Application ID** and, if you specified a server-side web application, an **Application secret** which you use in your application code.

Make Power BI content accessible to your application

To enable embedding of content, place your reports, dashboards, and tiles in an app workspace, and then publish that workspace to Power BI.

Code your application

Use the Power BI REST API, the Power BI .NET SDK, or the Power BI JavaScript API to write code to embed content in your application:

1. Use the **Application ID** and, if writing a server-side web application, the **Application secret** from the registration process to authenticate your user and create an auth token for Power BI.
2. Use Power BI API calls to perform the tasks your application requires. For example, you could use the auth token to obtain an embed URL for a Power BI report, and then use that URL to embed the report in your application.

For code examples showing how to embed Power BI content for your organization, see the *User Owns Data sample app* on GitHub.

User Owns Data sample app

https://aka.ms/AA55hb5

If required, assign your app workspace to dedicated capacity

If all of your internal users have Power BI Pro licenses, they will be able to use your application as is. However, if you want free users to be able to access it, you need to purchase capacity for the app workspace that contains your Power BI content and grant those free users permission to access the app workspace.

For more information about embedding Power BI content for your organization, see *Tutorial: Embed Power BI content into an application for your organization* in the Power BI documentation.

Tutorial: Embed Power BI content into an application for your organization

https://aka.ms/AA553vu

Embedding for your customers

Embedding Power BI content for your customers is a similar process to embedding it for your organization. However, it does contain additional steps and requirements to ensure that the application works seamlessly with your content, without the user interacting with the Power BI user interface.

1. Register your application with Azure Active Directory at http://dev.powerbi.com/apps:
   - Client-side native application
2. Grant admin consent to your content
3. Make Power BI content accessible to your application
4. Code your application
5. Assign your app workspace to a dedicated capacity

Register your application with Azure Active Directory

To register your application, perform the following steps:

1. In a web browser, go to **http://dev.powerbi.com/apps**.
2. Sign in using your Power BI credentials.
3. Specify the following, and then click **Register**:
   a. **Application name** – to identify your application in Azure
   b. **Application type** – choose client-side native application for Power BI Embedded applications that work with non-interactive logins
   c. **API access** – which API calls your application will use

The **Success** dialog box returns an **Application ID** that you use in your application code.

Grant admin consent to your content

To enable your application to make Power BI API calls without any user interaction, you need to grant consent for the master account user to make the calls in your Azure portal.

1. Sign in to your Azure portal.
2. On the **App registrations** blade, click your application name.
3. On your application blade, click **API permissions**, click **Grant admin consent for <your user name>**, and then in the message box, click **Yes**.
4. Wait for the grant process to complete.

Make Power BI content accessible to your application

To enable embedding of content, place your reports, dashboards, and tiles in an app workspace, and then publish that workspace to Power BI.

Code your application

Use the Power BI REST API, the Power BI .NET SDK, or the Power BI JavaScript API to write code to embed your content in your application:

1. Use the **Application ID** from the registration process to authenticate your master account and create an auth token for Power BI.
2. Use Power BI API calls to perform the tasks your application requires. For example, you could use the auth token to obtain an embed URL for a Power BI report, use that URL to create an embed token for the report, and then use the embed token to embed the report in your application.

For code examples showing how to embed Power BI content for your customers, see the *App Owns Data sample app* on GitHub.

App Owns Data sample app

https://aka.ms/AA55hb5

Assign your app workspace to a dedicated capacity

When using Power BI Embedded for your customers, your app workspace must be assigned to a dedicated capacity.

For more information about embedding Power BI content for your customers, see *Tutorial: Embed Power BI content into an application for your customers* in the Power BI documentation.

Tutorial: Embed Power BI content into an application for your customers

https://aka.ms/AA553vs

The Power BI Embedded Playground

The Power BI Embedded Playground is a good starting place for learning about the Power BI API, for trying out Power BI REST API operations without needing to write any code, and for using API calls to perform specific tasks against your data.

- Purpose of the Power BI Embedded Playground:
  - As a learning resource
  - For trying out Power BI REST API operations without writing code
  - To use API calls to perform specific tasks
- Access the playground at https://microsoft.github.io/PowerBI-JavaScript/demo/v2-demo/index.html

Browse to the site at https://microsoft.github.io/PowerBI-JavaScript/demo/v2-demo/index.html. On the **Samples** page, you can access code samples that show how to embed the following items in your applications:

- Report
- Report Visual
- Q&A
- Dashboard

- Tile

By default, you can review and run the sample code against the sample data and copy the sample code for your own use. You can also change the settings to run the code against your own Power BI content directly in the playground.

On the **Showcase** page, you can review and learn about new features in Power BI Embedded and the **Documentation** page contains links to useful content, in addition to videos about Power BI Embedded.

Demonstration: Using the Power BI Embedded Playground

In this demonstration, you will see how to use the Power BI Embedded Playground to learn about the Power BI API.

Lesson 2
Custom visuals

In this lesson, you will learn how to add custom visuals to reports, dashboards, and workspaces and how to use the Microsoft Power BI visuals project to create your own custom visualizations. You will also learn how to share custom visuals through the Power BI visuals gallery.

Lesson Objectives

After completing this lesson, you will be able to:

- Import custom visuals from the Power BI visuals gallery into Power BI and use them in Power BI reports.
- Describe the process used for creating custom visualizations for use with Power BI applications and dashboards.

Using custom visuals

Power BI Desktop and the Power BI service both provide a whole range of visualizations that you use to display information in your reports. There are, however, additional visuals that you can use called *custom visuals*.

- Custom visual files:
  - .pbiviz files containing code
  - May be a privacy or security risk
- Organizational visuals:
  - Custom visuals that a Power BI admin has approved for your organization
  - Cannot be displayed in emails or exported to PowerPoint
- Marketplace visuals:
  - Tested and approved for functionality and quality
  - Certified visuals—more rigorously tested subset that are supported in email and PowerPoint
- Importing organizational and marketplace visuals

There are three types of custom visual that you can use in your reports:

1. **Custom visual files**. These are .pbiviz files that you import into a Power BI report to render your data. Any developer can create a .pbiviz file and, as such, the code they contain might present a privacy or security risk. Only import custom visual files from authors and sources that you know you can trust.
2. **Organizational visuals**. These are custom visuals that a Power BI admin has approved for use in your organization. They cannot be rendered in emails if a user subscribes to the report or exports it to PowerPoint.
3. **Marketplace visuals**. These are custom visuals created by Microsoft and members of the Power BI community that Microsoft has tested and approved for functionality and quality. Power BI certified visuals are a subset of the marketplace visuals that have been tested more rigorously for quality and are supported in email and PowerPoint scenarios.

Importing organizational and marketplace visuals

The steps to import organizational or marketplace visuals are similar; you just need to ensure that you're viewing the relevant list of available visuals in the **Power BI Visuals** dialog box. You can only import custom visuals into one report at a time.

To import a custom visual:

1. In Power BI Desktop, in the **VISUALIZATIONS** pane, click the ellipsis, and then click **Import from marketplace**.

2. If the visual you want is an organizational visual, in the **Power BI Visuals dialog** box, click **MY ORGANIZATION**.
3. Either browse the categories or use the **Search** box to find the visual that you want to use.
4. Click the visual name to review a description of the visual.
5. Click **Add** to import the visual to Power BI.
6. In the **Import custom visual** message box, click **OK**.

You can then use the imported visual to present your data in the same way that you use other visualizations in Power BI.

You can also download and import custom visuals from Microsoft AppSource at https://appsource.microsoft.com/. This method enables you to download a sample report showing how to use the visual.

1. In Internet Explorer, go to **https://appsource.microsoft.com/**.
2. On the **Apps** menu, click **Apps**.
3. In the **Products** list, click **Power BI visuals**.
4. Either browse the categories or use the **Search Microsoft AppSource** box to find the visual that you want to use.
5. Click the visual name to see a description of the visual and reviews from other users.
6. Click **GET IT NOW**, and in the **Sign in to Microsoft AppSource** dialog box, enter your email address, and then click **Sign in**.
7. If prompted, review the terms of use and privacy policy, and if you agree, click **Continue**.
8. On the **Install** page, you can either download the visual as a .pbiviz file or download a sample report that uses the visual as a .pbix file.
9. In Power BI Desktop, in the **VISUALIZATIONS** pane, click the ellipsis, and then click **Import from file**.
10. In the **Caution: Import custom visual** dialog box click **Import**.
11. Browse to the location of your downloaded .pbiviz file, select the file, and then click **Open**.
12. In the **Import custom visual** message box, click **OK**.
13. Optionally, open the downloaded report file, .pbix file, to review how to use the custom visual in a report.

Creating custom visuals

- Microsoft Power BI visuals project from GitHub:
  - Open-source project
  - Compiles into JavaScript for browser compatibility
- Power BI Embedded Playground:
  - Review a custom visual
  - Test your own custom visuals

If you need to render data that is not facilitated using either the in-built visualizations or those from the organizational and marketplace lists, you can create your own custom visuals, You either extend the capabilities of the in-built visuals or develop your own from scratch.

The code for many Power BI visuals is available in the Microsoft Power BI visuals project from GitHub. This open-source project consists of visualization code, tooling, and a test suite. The project includes more than 20 types of visualization, the framework that's needed to run the visuals, and a testing infrastructure. The framework provides the required interface for integrating with the selecting controls, the filtering controls, and other user interface controls in Power BI. Because the code is written in TypeScript, it makes the visuals straightforward to build and debug. The visuals are built by using D3 (although you can use WebGL, Canvas, or SVG), and they compile into JavaScript and are compatible with modern browsers. This combination of technologies enables you to build your own custom visuals quickly.

You use the Power BI Embedded Playground at https://microsoft.github.io/PowerBI-JavaScript/demo/v2-demo/index.html to learn about coding custom visuals, in addition to testing how they look by using the Desktop or Phone view. You can also view and test code that interacts with the visual, such as filtering code and menu extensions.

For more information about creating custom visuals, see *Tutorial: Developing a Power BI custom visual* in the Power BI documentation:

Tutorial: Developing a Power BI custom visual

https://aka.ms/Y7zusi

Demonstration: Importing and using a custom visual

In this demonstration, you will see how to:

- Import a custom visualization into the Power BI Desktop.
- Use a custom visualization in a report.

Lab: Using marketplace visualizations

Scenario

Adventure Works employees are using Power BI to gain insights into sales patterns and trends, and for most purposes, the standard visualizations are all that is required. However, for some data there might be scope for using other types of visualization.

As a BI professional, you are asked to explore the use of custom visualizations, so that Adventure Works employees extract maximum value from sales datasets.

Objectives

After completing this lab, you will be able to:

- Use a custom visualization with Power BI data.

Note: Because of updates to Microsoft Power BI, the lab steps for this course change frequently. Microsoft Learning regularly updates the lab steps, so they are not available in this manual – but you can access them on GitHub.

Lab Setup

Estimated Time: 60 minutes

Virtual machine: **20778C-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa55w.rd**

All the lab steps are contained in 20778C_LAB_08.md.

Exercise 1: Use a custom visualization

Scenario

As a data analyst for AdventureWorks, you are investigating the types of visualizations that can be used with sales data. For some data, it's suggested that custom visualization might help AdventureWorks employees make better business decisions.

In this exercise, you will apply the Sunburst custom visualization to an existing report and compare this visualization with the standard visual that was previously in use.

The main tasks for this exercise are as follows:

1. Prepare the lab environment

2. Using custom visuals

▶ Task 1: Prepare the lab environment

▶ Task 2: Using custom visuals

Results: At the end of this exercise, the Sunburst custom visualization will be used in a Power BI report.

Question: Do you think that the Sunburst visualization provides additional insights into the Sales Order data, compared with the clustered column chart that was originally used?

Question: From your own experience, are there any other custom visuals from the Power BI visuals gallery that would add value to the Sales Order data?

Module Review and Takeaways

In this module, you learned about the Power BI API and how developers use it to create applications. You also learned about the Power BI Embedded Playground and how to use custom visuals in Power BI reports.

Review Question(s)

Question: Discuss the potential of the Power BI Developer API for your own organization. Are there any particular Power BI-based applications that you already use, or would like to see developed?

Module 9

Power BI Mobile

Contents:

Module Overview

Power BI™ mobile apps enable you to access and use Power BI information on a mobile device, including iOS (iPad, iPhone, iPod Touch, Apple Watch), Android phone or tablet, and Windows® 10 devices. This means that, potentially, Power BI reports and Power BI dashboards created in Power BI Desktop and the Power BI service can be used anywhere and at any time.

Power BI reports and dashboards are designed to work on a mobile device without modification. However, you can also create specific optimized reports and report layouts for display on mobile devices. The Power BI mobile apps support the sharing and annotation of dashboards, and you can use Power BI data on mobile devices even when you are not connected to a network. Power BI alerts and notifications also work across the Power BI service, including on mobile devices.

Objectives

After completing this module, you will be able to:

- Create dashboards and reports for mobile devices.
- Use the Power BI mobile app.

Lesson 1
Power BI mobile apps

In this lesson, you will learn how to view Power BI reports and dashboards on a mobile device, and understand the features included in the mobile apps. You will also learn how to optimize report layouts for display on mobile devices.

Lesson Objectives

After completing this lesson, you will be able to:

- Create dashboards for mobile devices.
- Understand the features of the Power BI app for iOS devices, including iPhone and iPad.
- Describe the available features included in the Power BI app for Android devices.
- Optimize reports for display on mobile devices.

Creating dashboards for mobile devices

The Power BI mobile app is available for iOS, Android, and Windows 10 mobile devices, enabling Power BI users to view reports and dashboards, and interact with data, from any location. You use the Power BI service to create dashboards, which you then view on a device running the app. Dashboards automatically adjust and resize to fit the target screen size, and the data refreshes in real time, giving up-to-the minute results. This means there is no need for any additional formatting, and no need to create resized visuals for mobile reports.

- The Power BI app is available for phones and tablets running iOS, Android, or Windows 10
- View and interact with dashboards from any location
- Dashboards automatically fit to the screen size
- Power BI Pro users can create app workspaces with dashboards designed for mobile devices
- Background refresh supports use of offline data
- Microsoft Intune for managing apps and devices

Note: At the time of writing, Windows 10 Mobile is deprecated and support will end in December 2019.

Design considerations

Although dashboard items scale to size, you might wish to pay attention to the visuals you include on a dashboard and the level of detail. If you know the target device is a tablet or phone, you make allowance for the screen size. A bar chart with 30 columns might display perfectly on a tablet, but may be more difficult to view on a mobile, even in landscape mode.

Note: If your organization uses Power BI Pro, and users connect using mobile phones, you can create app workspaces with reports and dashboards designed for the smaller screen. You can scale down the size of visuals, and ensure that the most important data is placed at the top of the dashboard. These items are then shown first when you vertically scroll dashboard items.

Offline data

By default, Power BI runs a background data refresh, so if you go offline, data remains reasonably up to date. While offline, you continue to have access to the dashboards and reports you have recently accessed from the app.

Manage apps and devices

Organizations can manage and control apps and devices with Microsoft Intune®. The Power BI apps for iOS and Android integrate with Intune, so you can manage the apps on the devices, in addition to controlling security. Intune works alongside Mobile Device Manager (MDM) within Office 365®.

For more information on managing your devices with Intune, see *Configure mobile apps with Microsoft Intune* in the Power BI documentation:

Configure mobile apps with Microsoft Intune

http://aka.ms/E4v70j

Power BI for iOS

The Microsoft Power BI for iOS app is compatible with the iPhone, iPad, and Apple Watch, and is part of the family of mobile BI experiences for Power BI. You use the app to view and interact with your organization's dashboards and reports from anywhere in the world. In addition to accessing live on-premises and cloud data, you can share dashboards with colleagues using email or text messages. You can also view SQL Server mobile reports and KPIs for your on-premises data by using the Power BI app.

- Power BI for iOS works with iPhone, iPad, and Apple Watch
- View and interact with Power BI reports and dashboards, and SQL Server mobile reports and KPIs
- Download from Apple App Store
- View dashboards in landscape mode for the same experience as viewing on the Power BI service portal
- Annotate and share tiles with colleagues
- Scan QR codes to open tiles directly
- Set up data alerts on single figure tiles: set above and below values to be alerted when value goes outside the boundaries

You can download the app by searching for Microsoft Power BI on the Apple App Store or from the *View dashboards on your iPhone* section of *View dashboards and reports in the Power BI mobile apps* in the Power BI documentation:

View dashboards on your iPhone

http://aka.ms/AA55urd

You don't need to sign in to Power BI to start using it on your iOS mobile device. The app includes sample dashboards, so you can see how the app works before you sign in and view your organization's content.

Viewing modes

When viewing dashboards on your iPhone in portrait mode, the tiles stack vertically, in the left to right order of the tile placement, on the web version of the dashboard in Power BI service. If you turn your phone sideways to view the dashboard in landscape mode, the dashboard tiles display exactly as they are on the portal, which is useful for tiles that are grouped together contextually.

Interacting with tiles

You interact with dashboard tiles on your iPhone and iPad in the same way as you do on the portal. You tap a tile to open it in **Focus mode**. You then tap to view items in more detail in pie, bar, and line charts. Tap a pie chart to put it in **Focus mode**, and the slicer automatically appears. Spin the chart to show each of the pie slices in detail.

Annotate and share tiles

You can add notes and emoticons directly to tiles. You tap a tile to bring it into **Focus mode**, and then use the tools to annotate the tile.

You can share dashboards with colleagues who then receive an email inviting them to access the dashboard. To view the dashboard, recipients must have a Power BI Pro license or you need to host the content in a Premium capacity. You can also send a snapshot of a tile from the iPhone app to anyone in or outside of your domain—they will receive an image of the tile, report, or visual alongside a link to the original content.

Power BI QR codes

The Power BI mobile app for the iPhone includes a QR scanner, which means users can scan a QR code that links directly to a dashboard tile, and opens in Power BI Mobile. Consider the following scenario: you create a dashboard in the Power BI service for presenting to the senior managers in your organization; you display the dashboard on a large TV in Full Screen Mode during the presentation—but you want the managers to view the data in more detail during the meeting. By creating a QR code for those tiles that need viewing in more detail, you can give the code to the managers, either on paper, in an email message, or from your iPhone. The code opens the tiles directly in the Power BI app.

To generate a QR code, open the relevant dashboard in Power BI service. Click the ellipsis (**...**) on the tile you want to create a code for, and then click **Open in focus mode** to open the tile. Click the ellipsis (**...**), and then click **Generate QR code**. After it has been generated, you download the code as a .jpg file. You use this file in email messages and PowerPoint slides, save it to your phone, or print it.

To scan a QR code, in the Power BI Mobile main menu, tap **Scanner**, or use a QR scanner app that is already installed on your mobile. Both methods require access to the camera on your phone, which you must allow.

Data alerts

Data alerts can be added to tiles that display a single number. You set thresholds to alert you when the number goes above or below the value you set, or you can set both. For example, your organization's sales for the year currently show $27.31 million. You can add an alert so you are notified when this figure reaches $30 million. For example, if you wanted to monitor your organization's share price, you could set an alert for when the value drops below $15, and goes above $25.

Power BI for Android

The Power BI app for Android devices has been created with much the same abilities as the app for iOS, with the emphasis on enabling data insights on the move. You can download and install the app, either by searching for Microsoft Power BI on Google Play or from *View dashboards and reports in the Power BI mobile apps* in the Power BI documentation:

View dashboards and reports on your Android phone

http://aka.ms/AA55urg

- Power BI for Android app is designed for Android phone users
- View and interact with Power BI reports and dashboards
- Download from Google Play
- View dashboards in landscape mode for the same experience as viewing on the Power BI service
- Annotate and share tiles with colleagues
- Scan QR codes to open tiles directly

After installing the app and signing in to Power BI, swipe right on the **Home** screen to see your dashboards, then tap any dashboard to view it.

Viewing modes

On an Android phone, you can view dashboards in portrait mode, which arranges the tiles one on top of another. For a uniform view, they all resize to the same width, filling the available screen space. Landscape mode is also supported, meaning you can view a dashboard in the same layout as it was designed on the Power BI service portal.

Interacting with tiles

While viewing a dashboard, you can tap the ellipsis (**...**) to invite a colleague to share the dashboard, **Refresh** the data, or find out **More about this dashboard**. Swipe up and down to see all the tiles in the dashboard. Tap a tile to put it in **Focus mode**, then tap the points in a chart to see specific details and values.

Annotate and share tiles

You can annotate and add stickers (emoticons) to your dashboard tiles. Tap a tile to open it in **Focus mode**, and then use the tools to annotate the tile.

You can share dashboards with colleagues and, if you are the dashboard owner, you can view the colleagues you have invited and see whether they have accepted your invitation. You can also allow colleagues to share the dashboard with others and dashboard owners can unshare a dashboard.

Power BI QR codes

The Power BI for Android app includes a QR scanner, or you can use any of your other QR code scanner apps. The scanner included with Power BI needs access to your phone's camera, so you must allow this before scanning. When you scan a QR code for a tile, it opens immediately after successfully interpreting the code, either when you use the Power BI scanner, or an alternative scanner app.

Optimizing reports for mobile apps

If you view a regular dashboard on a mobile device—especially a dashboard that has a lot of data and descriptive fields—by default, you will get a long list of tiles in a top to bottom, left to right, order. If you turn your mobile device to landscape mode, it will render in the same layout as on the web. However, if you want a dashboard to be optimized for viewing on a mobile device, you might need to change the layout of the tiles just for those devices; for example, you might have a lot of descriptive tiles that do not need to be displayed on the smaller screen.

- Creating dashboard views for mobile devices:
  - In the Power BI service, toggle Web view to Phone view
  - Changes propagate immediately
- Creating report views for mobile devices:
  - In Power BI Desktop, open Report View, and then click Phone Layout
  - Publish main report from Power BI Desktop to publish mobile version
- Planning for mobile-specific report layouts:
  - Optimize only the pages that require a mobile layout
  - Cannot modify formatting settings for just mobile devices

Creating dashboard views for mobile devices

You can optimize a dashboard in the Power BI service for mobile device use: open the dashboard in the Power BI service, click **Web view** in the top right corner, and then click **Phone view**.

You now get a view where the tiles are shown in the same layout as you would see on a mobile screen; you can remove tiles, reorder tiles, and resize tiles. If you want to remove everything and start from

scratch, there is an option to unpin all the tiles; you can also reset the phone view to match the default layout.

Note: Any changes you make to the phone view of a dashboard in Power BI service are automatically propagated to anyone who views that dashboard in a Power BI mobile app.

Creating report views for mobile devices

You can also optimize Power BI Desktop reports for mobile device consumption:

1. In Power BI Desktop, click **Report View** in the left navigation bar.
2. On the **View** tab, click **Phone Layout**. You now get a blank phone canvas. All of the visuals on the original report page are listed in the **VISUALIZATIONS** pane on the right.
3. To add a visual to the phone layout, drag it from the **VISUALIZATIONS** pane to the phone canvas.
4. To remove a visual, click the **X** in the top-right of the visual on the phone canvas, or select the tile and press **Delete**. Removing a visual in this view only removes it from the Phone canvas; the visual and the original report will not be affected.

Phone reports use a grid layout. As you drag visuals to the mobile canvas, they snap to that grid. You can add some or all of the master report page visuals to the phone report page. You can add each visual only once. You can resize your visuals on the grid, as you would for tiles on dashboards and mobile dashboards.

Note: The phone report grid scales across phones of different sizes, so your report will look equally good on small- and large-screen phones.

When planning for mobile-specific report layouts, note the following:

- For reports with multiple pages, you can optimize all the pages or only a few.
- On a phone, you move between pages by swiping from the side or tapping the page menu.
- You cannot modify formatting settings for just the phone. Formatting is consistent between master and mobile layouts. For example, font sizes will be the same. So to change a visual, such as changing its formatting, dataset, filters, or any other attribute, you must return to the regular report authoring mode.

Publishing a phone report

To publish the phone version of a report, you publish the main report from Power BI Desktop to the Power BI service, and the phone version publishes at the same time.

Viewing optimized and non-optimized reports on a phone

In the mobile apps on phones, Power BI automatically detects optimized and non-optimized phone reports. If a phone-optimized report exists, the Power BI phone app automatically opens the report in phone report mode. If a phone-optimized report does not exist, the report will open in the non-optimized, landscape view.

When viewing a phone report, changing the phone's orientation to landscape will open the report in the non-optimized view with the original report layout, whether you optimize the report or not. If you only optimize some pages, when users access a non-optimized page in portrait view, they will be prompted to rotate their device to view the page in landscape mode. For more information about creating reports for phone apps, see *Optimize reports for the Power BI mobile apps* in the Power BI documentation:

Optimize reports for the Power BI mobile apps

https://aka.ms/b1tebj

Demonstration: Creating reports and dashboards for mobile apps

In this demonstration, you will see how to:

- Create a phone layout for a report in Power BI Desktop.
- Create a phone view for a dashboard in Power BI service.

Question: If you have an iOS or Android phone or tablet, download the Power BI app if you don't already have it. You do not need to sign in with a Power BI account, because you can use the sample data.

Explore the features of the Power BI app, looking at the Power BI samples. Which useful features could be added to improve the app? In addition to features you like and don't like, discuss how they could be useful in your organization.

Lesson 2
Using the Power BI mobile app

In this lesson, you will learn how to create and publish reports specifically for mobile devices. You'll also learn about the features in the mobile apps that enable dashboards to be shared and annotated—and how the apps enable Power BI data to be displayed offline. Finally, you will learn about Power BI alerts and notifications, and how alerts work for mobile devices.

Lesson Objectives

After completing this lesson, you will be able to:

- Create and view mobile device-specific on-premises reports.
- Share and annotate a snapshot of a tile, report, or visualization from the Power BI mobile app.
- Use the Power BI mobile app for offline content consumption.
- Use the Power BI Notification Center, and set alerts for the mobile apps and the online Power BI service.

Creating and viewing on-premises reports

In addition to using the Power BI mobile apps to view Power BI dashboards and reports, you can also use them to view Reporting Services mobile reports and KPIs. You create mobile device-specific reports using SQL Server Mobile Report Publisher, and then publish these reports to a SQL Server Reporting Services web portal, where you can also create KPIs. You then view these mobile reports and KPIs in the Power BI mobile app, organized in folders or collected as favorites.

- Use Power BI mobile apps to view on-premises reports and KPIs
- Create as SQL Server Reporting Services mobile reports using SQL Server Mobile Report Publisher, optional phone layout
- Publish reports to a SQL Server Reporting Services web portal
- View reports in Power BI mobile app:
  - Select **Connect to server** using format: http://<servername>/reports or https://<servername>/reports

Note: If you do not have access to a Reporting Services web portal, you can still explore the features of Reporting Services mobile reports in the Power BI mobile app. Tap the options icon, and then tap **Report Server samples**. Browse the samples to interact with KPIs and mobile reports.

You can connect mobile reports to a range of data sources, including on-premises SQL Server and Analysis Services data. You design the layout of your mobile reports on a design surface with adjusting grid rows and columns, and flexible mobile report elements that scale well to any screen size. You then save these mobile reports to a Reporting Service portal, and view and interact with them in a browser or in the Power BI mobile app on iPads, iPhones, and Android phones.

Note: To use the Power BI mobile app to view your reports and KPIs, you need to enable Basic Authentication on your reporting server.

Prepare SQL Server Reporting Services

The key steps for preparing SQL Server Reporting Services ready for creating mobile reports are as follows:

1. Configure SQL Server Reporting Services (SSRS) if it is not already running.
2. In SSRS, create a shared data source.
3. In Report Builder, create a new dataset using your shared data source and save it to the SSRS server.

You use this dataset for KPIs and mobile reports; multiple datasets can use the same data source.

Create a new KPI

To create a KPI:

1. On the Reporting Services web portal, on the **New** menu, click **KPI**.
2. Select your dataset then select the field and aggregation you want to use.
3. Choose the trend set to show the past KPI values in the chart.
4. Choose the visualization type for the chart.

Create a new mobile report

To create a mobile report:

1. Download and install the SQL Server Mobile Report Publisher.
2. Start Mobile Report Publisher—you can begin with visuals or with data. If you start with visuals, sample data is automatically generated.
3. To add your own data, click **Add Data**, and select where your data is located. You can add local Excel data or a shared dataset from your SSRS instance.
4. To create a phone layout, select **Phone** in the layout dropdown menu in the top-right of the window, and design the report.
5. Save the report, typically to an instance of Reporting Services.

View KPIs and mobile reports in the mobile app

Use the Power BI app to connect to a Reporting Services server to view mobile reports and KPIs. To get started:

1. Ensure you have the latest version of the Power BI app downloaded to your device.
2. Open the app, tap **Sign In**, and sign in to the app using your credentials.
3. Tap the options icon, and then tap **Connect to server**.
4. Enter the server address and your user name and password. Use this format for the server address:
5. **http://<*servername*>/reports** or **https://<*servername*>/reports** (https is recommended for all production scenarios).
6. Optionally specify a display name for the server.
7. Tap **Connect**.
8. Tap the options icon anytime to go between your Reporting Services mobile reports and your dashboards in the Power BI service.
9. When you enter the app in future, you will see the server in the menu that you have set up. You can only have a connection to one instance of Reporting Services at a time in the app. If you want to connect to a different server, you need to disconnect from the current one.

In future, the Power BI mobile app will default to the KPI page as the first page you see, unless you are also connected to a Power BI instance. It will default to Power BI when both services are connected.

KPIs and mobile reports can be marked as favorites in the Reporting Services web portal; you then view them in a single folder on your device, together with Power BI favorite dashboards and reports. You can also mark KPIs and mobile reports as favorites within the Power BI mobile app.

For more information about this, see *Create mobile reports with SQL Server Mobile Report Publisher* in Microsoft Docs:

Create mobile reports with SQL Server Mobile Report Publisher

https://aka.ms/ac9os9

Annotating and sharing content

You can share a snapshot of a tile, report, or visualization from the Power BI mobile app for iOS, Android, and Windows 10 devices as a mail message. The snapshot shows the information as it was at the time when the mail was sent, and includes a link to the source tile, report, or visualization. Snapshots can be sent to anyone but, to access the current information through the link, the recipient must have the appropriate permissions—and you must have already shared the dashboard or report with them.

- Share a snapshot of a tile, report, or visualization from the Power BI mobile app as a mail message:
  - For iOS, Android, and Windows 10 devices
  - Snapshot shows the information when mail was sent
  - Email includes a link to the source tile, report, or visualization
  - To access source, recipient must have permissions and dashboard or report must be shared
- In the iOS and Android apps, you can also add annotations, including lines, text, and stamps, to the snapshot before it is shared

In the iOS and Android apps, you can also add annotations, including lines, text, and stamps, to the snapshot before it is shared.

To annotate a tile, report, or visual:

1. Open a report, or tap a tile or visualization to open it in focus mode.
2. Tap the annotate icon.
   - To draw lines, tap the squiggly line or pen icon, choose a width and color, and draw.
   - To type comments, tap the text icon, choose the text size and color, and type.
   - To paste stamps (such as emoticons), tap the smiley face, choose a color, and tap where you want them.

To share a tile, report, or visual:

1. In focus mode, tap **Share** in the upper-right corner.
2. Tap the Mail icon.
3. Type the recipients' names, and (optionally) edit the standard message text.
4. Tap **Send**.

The mail message includes a link to the live version of the tile, report, or visualization. The message recipients can click this link and go straight to that tile, report, or visualization, provided that:

- The recipients have been assigned the appropriate permissions to the dashboard or report.
- The dashboard or report has been shared with the recipients.

Taking mobile content offline

The Power BI mobile app supports offline content consumption so, unlike when you're using a mobile browser, data is still available when you are not connected to a network. The mobile app automatically caches the dashboards in your My Workspace, together with any other dashboards you have viewed in the previous two weeks. The Power BI app provides indicators to show that you are using offline data, so it's clear whether the dashboards, reports, and tiles are showing current live or cached information.

- Power BI mobile app supports offline content:
  - Automatically caches dashboards in your My Workspace and dashboards viewed in previous two weeks
  - Indicators show that you are using offline data
- Background data refresh:
  - Cached data is automatically refreshed with data on the Power BI service (not the data source), whenever your device is connected to a network:
    - Wi-Fi network: background refresh every two hours
    - Mobile data network: background refresh every 24 hours
- Limitations:
  - Access to Power BI reports is read-only
  - You cannot filter, cross filter, sort, or use slicers

Background data refresh

Cached data is automatically refreshed with data on the Power BI service (not the data source), whenever your device is connected to a network:

- **Wi-Fi network**. Background refresh updates the content every two hours.
- **Mobile data network**. Background refresh updates every 24 hours.

If you do not want your device to use background refresh, you can turn it off, to prevent excessive network usage.

Note: For iOS devices that are managed by Microsoft Intune Mobile Application Management (MAM) policies, background data refresh for the Power BI mobile app is turned off. To refresh the data from the Power BI service, you must go into the app.

Offline limitations

While offline, you can interact with cached dashboards and reports, with the following limitations:

- Your access to Power BI reports is read-only.
- You can see full reports, but you cannot filter, cross-filter, sort, or use slicers.

When working offline with the Power BI mobile app, you might also encounter the following additional limitations:

- The Power BI app can only cache a maximum of 250 MB of data.
- Some tile types are not available offline, because they require an always-on server connection, such as Bing map tiles.
- Reporting Services mobile reports and KPIs can be viewed offline, providing you have viewed them while connected. However, these reports and KPIs do not refresh in the background; instead, they refresh when you open them.

Using data alerts

If you work in an organization where many coworkers are using and sharing information, your Power BI experience could potentially become overwhelming; for example, as others share dashboards with you, alerts are generated, and new reports generated. The Power BI Notification Center addresses this challenge.

- Notification information feed:
  - Messages about new dashboards that have been shared with you
  - Information about Power BI events and meetings
  - Alerts that you have set
- Alerts:
  - Set in Power BI service or Power BI mobile apps
  - Used to notify you when data in a dashboard changes beyond particular limits
  - Alerts are personal to you, and are not shared with other users

The Notification Center is a constantly updated information feed, including messages about new dashboards that have been shared with you, information about Power BI events and meetings, and alerts that you have set. You set alerts in either the Power BI service or the Power BI mobile apps; because the alerts are shared across your login, the same alerts display wherever you are connected, and whether the alert was set in the service or in an app.

Alerts are used to notify you when data in a dashboard changes beyond particular limits, and are used for tiles that feature a single number, such as cards and gauges. Alerts are personal to you, and are not shared with other users, even when you share a dashboard that includes a tile for which you have set an alert.

Note: If your device gets lost or stolen, you should connect to the Power BI service to turn off all data-driven alert rules, to prevent any alert notifications on that device from providing information about your data to an unauthorized person.

Setting an alert

The following steps describe how to set an alert in the Power BI app for iOS (the steps for Android devices are similar):

1. Tap a number or gauge tile in a dashboard to open it in focus mode.
2. Tap the bell icon to add an alert.
3. Tap **Add alert rule**.
4. Select to receive alerts above or below a value, then set the value.
5. Select whether to receive hourly or daily alerts, and whether to also receive an email when you get the alert.
6. (Optionally) change the alert title.
7. Tap **Save**.

Receiving alerts

Alerts are received into the Power BI Notification Center on your mobile device or in the Power BI service; this is also where you get any notifications about new dashboards that have been shared with you. Alerts are only generated when data is refreshed; after a refresh, if data reaches an alert threshold, the following occurs:

1. The Power BI service checks when the last alert was sent.
2. Depending on the alert interval option you configured for the alert (every hour or every 24 hours, for example) a new alert will be generated.
3. If the alert is configured to send an email message, the email will be sent.
4. Power BI adds a message in the Notification center, and adds a new alert icon to the applicable tile.
5. On a mobile device, tap the global navigation button to open the mobile notification center and see the alert details.

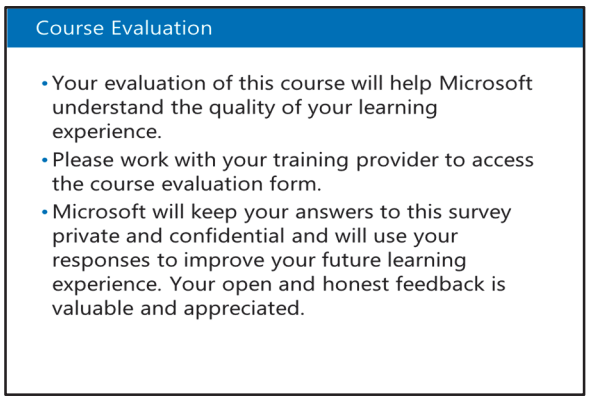
Module Review and Takeaways

In this module, you have learned how to create dashboards and reports for use on mobile devices, including specific features supported by iOS and Android operating systems. You learned how to create and publish reports for mobile devices, and how reports and dashboards can be optimized for consumption on mobile devices. You also learned how to share and annotate dashboards, how Power BI data can be used when mobile devices are offline, and how to set and use alerts.

Review Question(s)

Question: Which types of information are likely to work best on a mobile device? Ask students how they think Power BI mobile apps could be used in their own organizations, and which types of visualizations and data formats they would choose for mobile reports and dashboards.

Course Evaluation



Your evaluation of this course will help Microsoft understand the quality of your learning experience.

Please work with your training provider to access the course evaluation form.

Microsoft will keep your answers to this survey private and confidential and will use your responses to improve your future learning experience. Your open and honest feedback is valuable and appreciated.