Microsoft

OFFICIAL MICROSOFT LEARNING PRODUCT

10989B

Analyzing Data with Power BI

Product Number: 10989B

Part Number (if applicable): X21-24686

Released: 11/2016

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.

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

Note: This version of the course is built on the RTM release of SQL Server 2016 and replaces the A revision built on CTP3.

This two-day instructor-led course provides students with the knowledge and skills 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 key features of a self-service BI solution
- Describe Power BI and its data sources
- Model, shape, and combine data
- Describe Power BI data visualizations

Course Outline

The course outline is as follows:

- Module 1: 'Introduction to Self-service BI Solutions' introduces the key concepts in business intelligence, data analysis, and data visualization. Additionally they will learn the rationale for self-service BI, considerations for using self-service BI, and how Microsoft products can be used to implement a self-service BI solution.
- Module 2: 'Introducing Power BI' introduces Power BI.
- Module 3: 'Power BI Data' describes Power BI data sources.
- Module 4: 'Shaping and Combining Data' describes how to shape and combine data.
- Module 5: 'Modelling Data' describes how to model fata in Power BI.

- Module 6: 'Interactive Data Visualizations' describes Power BI visualizations.

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
10989B-MIA-DC	MIA-DC1 is a domain controller.
10989B-MIA-SQL	MIA-SQL has SQL Server 2016 RTM
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 10989B-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 is now 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 is 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. Organizations now need BI to see not only what has been done in the past, but also more of what is to come. There is an overwhelming amount of data to gather and compose into reports. Add to this an increasing need for data to offer up-to-the-minute numbers, so business can react faster to changing trends in markets and industries. Those businesses that can react fast and predict near-term trends to provide products and services where there is consumer demand, have the best chance of survival in our modern and highly competitive world. With the rise of big data, there is 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

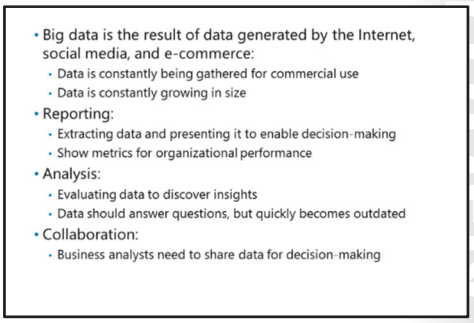
This lesson introduces you to the concepts that comprise BI. You will explore scenarios for using BI and how current trends affect the use of BI, along with project roles and data models.

Lesson Objectives

After completing this lesson, you will be able to:

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

Business Intelligence Scenarios



Big data has been big news for a while now. 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 want to track spending habits and use this data to sell you more. They gather the 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 sites, tempting you to buy something you might have searched for earlier.

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 their 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 and 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 can provide 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 predict the future.

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, and then cleaned, modeled, and transformed until it can be appropriately presented in a report. The report can be a simple table in a spreadsheet, or a visual and dynamic, colorful solution—but 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 can be asked of it. With an increase in publicly available datasets, including population changes, socioeconomic data, weather patterns, and climate change, corporate data can be analyzed against a backdrop of relevant statistics.

Collaboration

Data is ubiquitously generated and consumed. It is no longer retained and controlled by a handful of decision makers in an organization; instead, it is used at all levels, meaning colleagues can react to data, 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® 2016 offering in-memory analytics, data does not have to be moved outside of the database, and organizations can perform real-time operational analytics. The BI trend is moving away from analyzing past data only, to analyzing real-time data, and using historical data to predict the future.

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 Microsoft Excel has enabled this trend to continue. With the recent integration of the four power tools—Power Pivot, Power Query, Power View, and Power Map—Excel users can acquire data from a myriad of sources, and 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. If users can access the data they need, they can 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, and a wait for the report to be developed and published—only to find it does not deliver the right data. And 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 that they can see what is available for analysis, and lets them 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

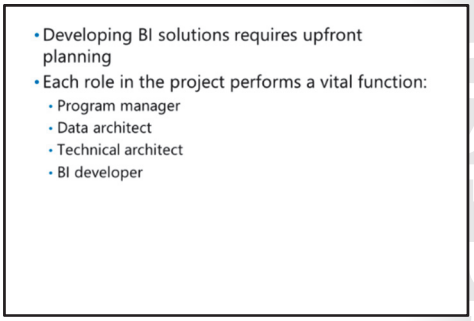
BI is no longer the reserve of large organizations with large budgets to throw at data warehousing projects. Any business operating on the web gathers information about their customers' spending habits, the products they viewed, and their buying decisions. It now seems that our online presence, enhanced 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 to some extent. However, gathering this data is no use unless it is 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 can 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, Pyramid Analytics, Microsoft, Oracle, IBM, SAP, SAS, and more. You can use these solutions to create highly visual reports. By connecting to a variety of data sources, you can then create reports and dashboards. However, depending on the vendor, many of the major solutions require expensive server and client licenses, in addition to trained users who can 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 is a new project, the program manager may 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.



Program Manager

The program manager is responsible for the organizational BI strategy and delivery, often coordinating multiple projects at any one 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 can understand. 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.

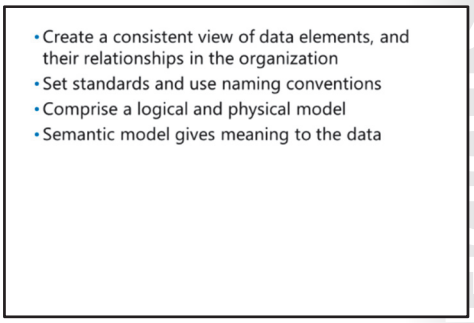
BI Developer

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, one developer may 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



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 is 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. Also, conventions can be applied to columns to denote keys, codes, and identifiers. The model can 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. If the database is an OLTP database, then 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, whereas 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 fully represent itself.

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

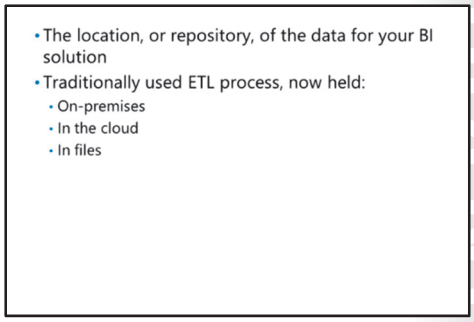
This lesson breaks down the components of data analysis. It looks at using queries to extract data from a variety of data sources, using transformations to make imported data easier to work with, and using visualizations to present data.

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 ready 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, because in-memory processing is designed for optimal performance and can better handle aggregations.

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 on-premises data. These may include your Microsoft SQL Servers, including SQL Server Analysis Services (SSAS), Active Directory® (AD), Exchange, and Access® databases. Your organization may also use other main industry databases including Teradata, Oracle, MySQL, Sybase, IBM DB2, SAP HANA, and PostgreSQL.

- **Cloud Databases**

  Cloud is becoming 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.

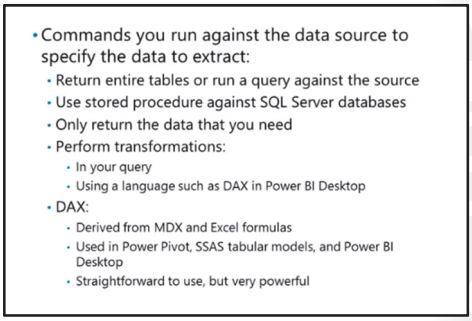
- **Software as a Service Providers**

  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 may use third-party solutions such as Facebook, Marketo, and MailChimp, alongside Bing®, Google Analytics, GitHub, and Zendesk. Being able 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. Also, business users may have data stored in text format, which requires importing into the BI solution.

Queries



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 may 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 a preferable option. A stored procedure is a query that is 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. They can also be used by other colleagues; sharing code prevents duplication of effort.

Depending on your role within the organization, you may be dependent on a database developer to write the queries or stored procedures for you and, for security reasons, you may not have access to all objects in the database. It is 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. You may be able to perform some transformations in your queries or stored procedures, but your BI solution might provide features to shape, format, and transform the data. If you are importing data from flat files, you will not be able to query the data to be selective about which columns to import.

Using DAX

If you are an advanced Excel user familiar with Excel formulas, you will find Data Analysis Expressions (DAX) to be very much the same. Whereas Excel formulas operate at a row level, DAX is used with relational datasets. You may have already used DAX in Power Pivot, or SQL Server Analysis Services tabular models—DAX is now available in Power BI Desktop. 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 are using Power BI Desktop for your self-service BI solution, then you can use DAX to enhance the data you import, without having to depend on developers to do this for you. If you are importing data that cannot be altered until after it has been imported, DAX again comes in useful. Writing Transact-SQL or MDX scripts can be complex and time-consuming, whereas DAX is straightforward to learn and apply to your datasets. For example, you can use DAX to concatenate columns in your dataset:

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

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]
```

DAX is useful for creating calculations. In the following example, DAX functions are used to multiply the current sales by 1.05, to give an estimated five percent increase, and then the figure is formatted as currency:

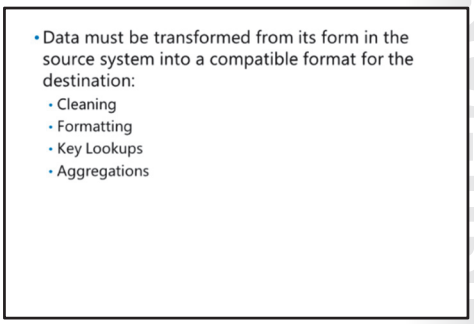
The following DAX formula multiplies the TotalSales figure by 1.05, to give a predicated target sales figure of five percent higher than the current year's sales. It also formats the result into the local currency.

The following DAX formula multiplies the TotalSales figure by 1.05, to give a predicated target sales figure of five percent higher than the current year's sales. It also formats the result into the local currency.

Calculate a Five Percent Sales Increase Using DAX

```
Target Sales = CURRENCY(CALCULATE([TotalSales] * 1.05))
```

Data Transformations



The transformations step of the ETL process is often the most time consuming. The data that is extracted from the source system must be transformed into the correct format for loading into the destination database. How much transformation is needed depends on how different the source and destinations are, and also if multiple source systems are extracted into the staging area. In even the most straightforward ETL processes, it is likely that some transformations are required. Metadata must exist before transformations can be applied. The metadata determines what transformations need to be applied to the source data held in the staging tables, so it can be loaded into the destination database.

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:

Cleaning

Before applying any transformations to your data, it is 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 into the staging area—you either apply a transformation to data that can be cleaned, or filter the dirty data into a separate table for further investigation into why it is incorrect in the source system.
- Removing duplicate rows.
- Eliminating incomplete rows.
- Performing logic tests to check date fields, such as checking to see 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 codes fields are correct.
- Performing character pattern testing to ensure phone numbers and email addresses are in the correct format.
- Logging missing values that are compulsory in the destination database.
- Checking data matches the business rules. For example, only one Sales Person manages a single customer.

Formatting

After the data is cleaned, you can apply formatting to ensure the source data is compatible with the destination data. Depending on how raw the data in the source system is, this often influences how much formatting needs to be done. 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 may want to ensure country or state codes are all uppercase, and names and addresses 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 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, then 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, then you have more freedom to apply some formatting.

Key Lookups

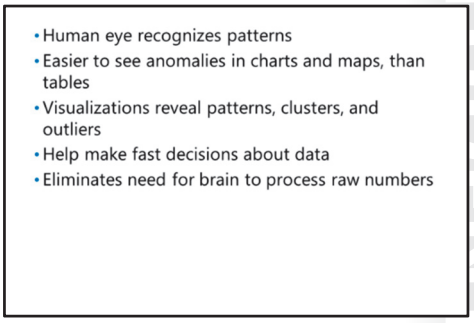
If you are extracting data from an OLTP system for loading into a data warehouse, then you need to convert the database design from a relational format into a star schema. A star schema comprises a fact table, with keys that relate to lookup values in dimension tables. To load data into the fact table, you need to look up values from the dimension tables and select the appropriate key. You also need to account for NULL values.

Aggregations

With very large datasets, it is common for aggregations to be performed in the staging area, and loaded into the data warehouse. Aggregations across millions of rows can take a considerable amount of time to run—this might mean that there would be an unacceptable amount of time for a user to wait while the numbers were aggregated for a report. However, this is based on the traditional data warehousing model. Modern database features, such as in-memory data, and columnstore indexes, enable faster performance alongside up-to-the-minute results. In the traditional data warehouse model, the data is usually loaded overnight so that, more often than not, reports are at least one working day behind the actual data.

Visualization

Evolution has given humans the ability to recognize patterns—this means we can instantly read and deal with dangerous situations, helping us to survive. We can very quickly identify irregularities, which means we can 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 mankind endured, we have retained the ability to visually assess and make judgements within incredibly small timeframes. In our modern world of information, this innate ability can be applied 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 are presented with a table of numbers in a spreadsheet, it is likely you would need to reorder the data and take some 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 can quickly take data that is in a table format and difficult to comprehend, and convert it into colorful charts and maps, which become instantly readable to the human eye. 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 do not 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.

Demonstration Steps

1. Ensure that the MSL-TMG1, 10989B-MIA-DC, and 10989B-MIA-SQL virtual machines are running, and then log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. In the **D:\Demofiles\Mod01** folder, run **Setup.cmd** as Administrator.
3. In the **User Account Control** dialog box, click **Yes**. When asked if you want to continue this operation, type **Y**, and then wait for the script to finish.
4. If you do not have a Power BI login, open Internet Explorer, go to **https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-signing-up-for-power-bi-with-a-new-office-365-trial**, and follow the steps to create an account.
5. In Internet Explorer, go to **https://www.microsoft.com/en-us/download/details.aspx?id=45331**, and then click **Download**.
6. On the **Choose the download you want** page, select the **PBIDesktop_x64.msi** check box, and then click **Next**.
7. In the message box, click **Run**.
8. In the **Microsoft Power BI Desktop (x64) Setup** dialog box, on the **Welcome to the Microsoft Power BI Desktop (x64) Setup Wizard** page, click **Next**.
9. On the **Microsoft Software License Terms** page, select the **I accept the terms in the License Agreement** check box, and then click **Next**.
10. On the **Destination Folder** page, click **Next**.
11. On the **Ready to install Microsoft Power BI Desktop (x64)** page, click **Install**.
12. In the **User Account Control** dialog box, click **Yes**.
13. On the **Completed the Microsoft Power BI Desktop (x64) Setup Wizard** page, clear the **Launch Microsoft Power BI Desktop** check box, and then click **Finish**.
14. Close Internet Explorer.
15. On the desktop, double-click **Power BI Desktop**.
16. When the Get Data screen shows, click **Get Data**.
17. In the **Get Data** dialog box, click **SQL Server Database**, and then click **Connect**.
18. In the **SQL Server Database** dialog box, in the **Server** box, type **MIA-SQL**.
19. In the **Database (optional)** box, type **AdventureWorksDW2016**, and then click **OK**.
20. In the **Access a SQL Server Database** dialog box, leave the default settings unchanged, click **Connect**.
21. In the **Encryption Support** dialog box, click **OK**.

22. In the **Navigator** dialog box, select the **FactInternetSales** check box.
23. Click **Select Related Tables**, and then click **Edit**.
24. If the **Connection Settings** dialog box appears, leave **Import** selected and then click **OK**.
25. In the **Untitled - Query Editor** window, in the **Queries** pane, click **FactInternetSales**.
26. Right-click the **CarrierTrackingNumber** column, and click **Remove**.
27. Right-click the **CustomerPONumber** column, and click **Remove**.
28. In the **Queries** pane, click **DimCustomer**.
29. Right-click the **Title** column, and select **Remove**.
30. Right-click the **NameStyle** column, and select **Remove**.
31. Right-click the **Suffix** column, and click **Remove**.
32. Right-click the **MaritalStatus** column, and click **Replace Values**.
33. In the **Replace Values** dialog box, in the **Value To Find** box, type **M**.
34. In the **Replace With** box, type **Married**, and then click **OK**.
35. Right-click the **MaritalStatus** column, and click **Replace Values**.
36. In the **Replace Values** dialog box, in the **Value To Find** box, type **S**.
37. In the **Replace With** box, type **Single**, and then click **OK**.
38. Right-click the **Gender** column, and click **Replace Values**.
39. In the **Replace Values** dialog box, in the **Value To Find** box, type **F**.
40. In the **Replace With** box, type **Female**, and then click **OK**.
41. Right-click the **Gender** column, and click **Replace Values**.
42. In the **Replace Values** dialog box, in the **Value To Find** box, type **M**.
43. In the **Replace With** box, type **Male**, and then click **OK**.
44. Click **Close & Apply**.
45. After the data has successfully loaded, click the **Modeling** tab.
46. In the **Fields** pane, expand **FactInternetSales**, and click **SalesAmount** to highlight the field.
47. In the **Formatting** group, click **Format: Currency General**, point to **Currency**, and then click **$ English (United States)**.
48. In the **Fields** pane, right-click **DimCustomer**, and click **New column**.
49. In the formula bar, type the following code, and then press Enter:

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

50. Click **Save**.
51. In the **Save As** dialog box, navigate to the **D:\Demofiles\Mod01** folder, in the **File name** box, type **Adventure Works Sales**, and then click **Save**.
52. Leave Power BI Desktop open for the next demonstration.

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.

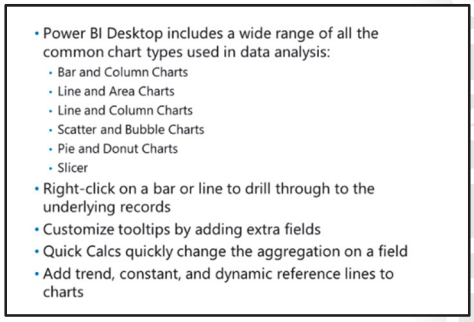
Lesson Objectives

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.

Charts

Using the chart visuals in Power BI Desktop, you can 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 you can drag a data field onto the report to automatically create a table visual—which 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.



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 value, 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.

The 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. These 100 percent stacked charts are useful for displaying progress in attempting to meet a target figure.

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 can include multiple lines on a line and stacked column chart. You can 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.

Scatter and Bubble Charts

A scatter chart shows the relationship between two numeric values using circles plotted on the chart. Scatter charts are useful for displaying large sets of data and, in particular, highlighting nonlinear trends, outliers, and clusters. You can also use a scatter chart 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 the data is aggregated into a single point. You should add a nonnumeric data field, such as Categories, to the chart Details property.

Based on the scatter chart, the bubble chart works with three numeric values, the bubbles being 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.

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 Sales Person 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.

Slicer

Use a slicer 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 you want to filter on.

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 including the data point's value and category. You can 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 can be applied to the field.

Quick Calcs

Use the Quick Calcs feature to quickly change the aggregation that is applied to the data in a visual. The default aggregation function is Sum, but you can change this by right-clicking on the Value field in the Visualizations pane, and choosing a different function, such as Average, Minimum, Maximum, or Count.

Reference Lines

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

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:

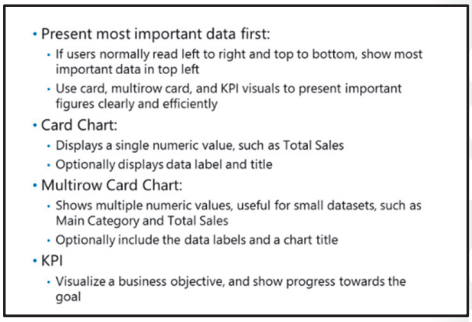
- Stacked Area
- Stacked Bar
- Stacked Column
- 100% Stacked Bar
- 100% Stacked Column

The following visuals can only include a trend line:

- Non-stacked Line
- Clustered Column chart

Cards

When presenting your data in a report or dashboard, 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 to the user. If you have important figures that need to be presented clearly, so that they can be easily read, then the Card and Multirow Card charts suit this purpose.



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 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 is entirely clear what the figure refers to, this is best included. You can rename the field by right-clicking on it in the Fields pane, and 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 can 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

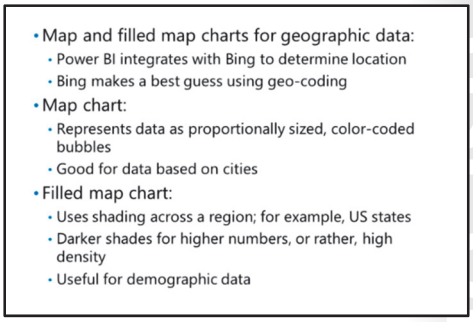
The multirow card chart is a useful way to clearly present numbers, without using the format of a table or matrix chart—which are difficult to digest. Like tables, the multirow card chart works best for smaller data sets; otherwise, there is too much data and text to read. For example, a multirow card 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.

KPI

Companies use Key Performance Indicators to measure their progress towards a business objective or goal. KPIs can be created at a high level to measure the overall performance of the company, in addition to being set at lower levels, such as by departments—for example, sales, call center, or warehouse. Adding a KPI visual to your report in Power BI enables you 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. These colors can be changed using the Format options.

Maps

Power BI Desktop includes a map chart, and a filled map chart. These charts enable you to visually map your data, 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.



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.

Shape Map

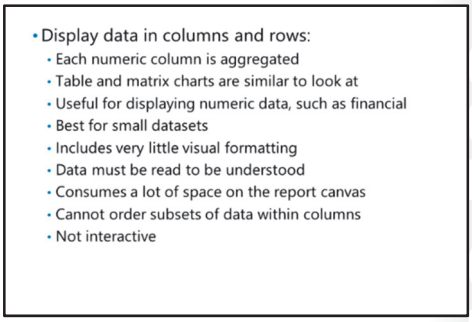
Shape Maps are similar to filled maps, in that they use color and saturation of color to represent the underlying data, and do not display numeric data values on the map itself. The shape map is ideal for comparing values across regions, such as demographic data. The visual includes predefined maps that you can use to map your data. These include French regions, Canadian provinces, Mexican states, and UK countries. Each region can be colored using the Data colors setting. Furthermore, you can use a custom map using your own data, as long as this is in the TopoJson format. To convert shapefiles or GeoJSON maps into the correct format, you can use an online tool such as MapShaper.

Map Shaper

http://www.mapshaper.org

After creating the TopoJson file, add a Shape map visual to your Power BI report, and select **Format**. Expand **Shape**, and click **Add Map** to import the data.

Tables



You can 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. Visually, the table and matrix charts look quite similar. Using a table or matrix is useful when you want to display the actual numbers, such as for financial data, and is best used for smaller sets of data. The table chart includes the option to apply predefined styles, which makes the data easier to read. You can 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 through 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 may 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
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

Category Name	Sales Territory Country	Total Sales
Clothing	Germany	$14,093.26
Clothing	United Kingdom	$18,219.16

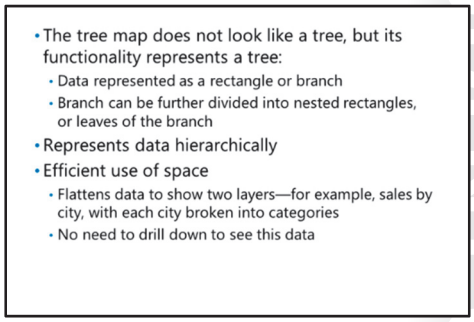
There is little difference in displaying data in a table in Power BI Desktop compared to Excel, or even a SQL Server Reporting Services report. From the table, you can see that it consumes space and takes time to read. It is not interactive and does not offer any drill-through capability.

Conditional Formatting

Depending on its value, you can customize the background color of a cell, including the ability to use gradient colors. After creating a table in Power BI, right-click the field in the Fields bucket of the Visualizations pane where you want to add colorize. From the menu, select **Conditional Formatting**. You can 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 may 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. The largest data value, represented as a rectangle, is located in the bottom left-hand corner, with the smallest in the upper right-hand corner. For example, in Power BI Desktop, 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 with the most sales have the largest rectangles. This style of representing data is classed as hierarchical.



You can also have a second value within each of the main rectangles in a tree map. Using the above example of Total Sales for each City, you could further break this down to include **Category**. Each Category would be represented by a nested rectangle within the parent **City** rectangle. This presents the data in a visual hierarchy that makes it quick to understand how the sales are spread across categories within each city.

Note: The tree map chart visualization has been added to SQL Server 2016 Reporting Services and is available for use in charts, in much the same way as you would use it in Power BI Desktop.

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 City and Category in the tree map, it has effectively flattened the data, and prevents the need for drilling down to see categories for each city.

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.

Demonstration Steps

1. In Power BI Desktop, in the **Fields** pane, expand **DimCustomer**, and then select the **Gender**, and **MaritalStatus** check boxes.
2. Expand **FactInternetSales**, and then select the **SalesAmount** check box.
3. In the **Visualizations** pane, click **Clustered column chart**.
4. Click **Format**, and then expand **Title**.
5. Change the **Title Text** to **Sales by Gender and Marital Status**.
6. Change **Alignment** to **Center**.
7. Expand **DimProduct**, and drag the **Color** field onto the report canvas to create a new table.
8. Under **FactInternetSales**, drag the **OrderQuantity** field onto the new table.
9. In the **Visualizations** pane, click **Donut chart**.
10. Click **Format**, expand **Title**, and then change the **Title Text** to **Orders by Color**.
11. Change **Alignment** to **Center**.
12. Under **FactInternetSales**, drag the **SalesAmount** field onto the report canvas to create a new column chart.
13. In the **Visualization** pane, click **Fields**.
14. In the **Fields** pane, expand **DimDate**, and drag the **EnglishMonthName** to the **Axis** property.
15. Grab the resizer on the column chart to widen the chart so that the month names display clearly.
16. In the **Visualization** pane, click **Format**, expand **Title**, and then change the **Title Text** to **Sales by Month**.
17. Change **Alignment** to **Center**.
18. On the **File** menu, click **Save**.
19. Leave Power BI Desktop open for the next demonstration.

Check Your Knowledge

Question	
Which of the following is not a real chart type?	
Select the correct answer.	
	100% 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.

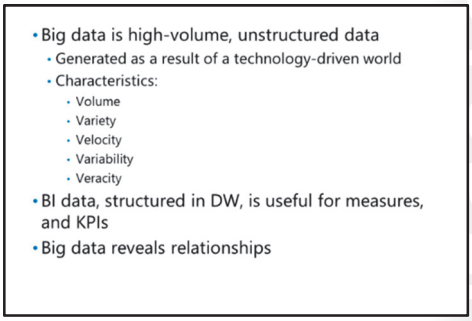
Lesson Objectives

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" has been 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 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, these days, big data is unavoidable. Big data is data that 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 described using the following characteristics:

- **Volume**: Volume is the quantity of data that is 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**: Variety refers to the type of data. For example, data gathered from a Facebook feed would gather text, photos and images, and video.
- **Velocity**: Velocity is the speed at which the data is generated and processed. Big data can be available in real time, using in-stream technology to view it as it is in motion.
- **Variability**: Variability refers to the consistency of the data; that is, how much does it vary? Inconsistency causes issues with data processing and management.
- **Veracity**: Veracity is the quality of the data that is captured. 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. Also, there is a cost associated with storing and managing the data.

Difference Between Big Data and BI Data

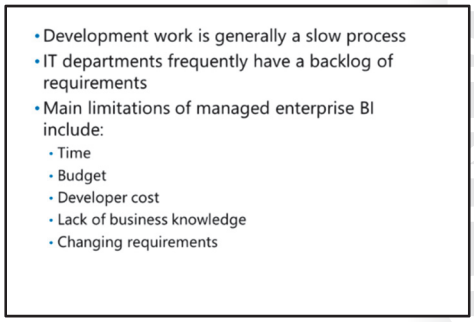
BI data is extracted from operational systems and processed using ETL. The staging area enables the data to be highly structured, consistent, and organized, ready for loading into the data warehouse. The data is highly dense, trends can be highlighted, and data can be measured. Big data, because of its size and unstructured format, 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 that have been 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, but 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.



The main limitations of managed enterprise BI include:

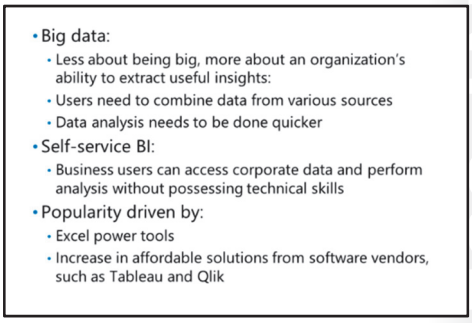
- **Time**: one of the biggest factors in managed enterprise BI is the time taken to develop the ETL system, build the data warehouse and cubes, write code to query the data, and design, develop, and publish reports. Even in a small organization, this is not a quick process—it requires planning, and a team of skilled developers. Much of the work is often centered around 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 very much linked to the time it takes to build the BI infrastructure, and associated code base. The amount of work required upfront before anything tangible can be delivered is often a concern for stakeholders. Developers may 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 to be 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 may 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 a while for a developer to be available to create the report, only to find it is not what they need. A request for change must then be 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. Being able 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.

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 suitable for analysis, that 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 recent 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 can be applied. With the ubiquity of big data in business, it is fast becoming a requirement that a BI tool can 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 may have when accessing data; the importance of data source reliability; how users require analysis skills; and how a data steward can help with these issues.

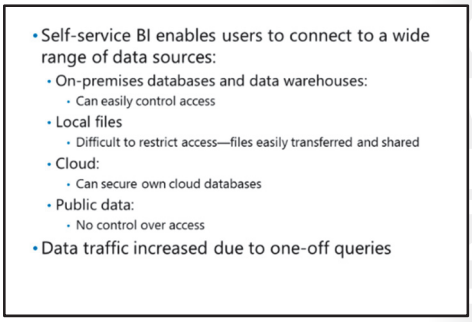
Lesson Objectives

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 can 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.



On-Premises Data

Self-service access to on-premises data can generally be 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 can be 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 also that business users have access to the data they need to do their job.

Data from files such as Excel, CSV, text, and XML, can be 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

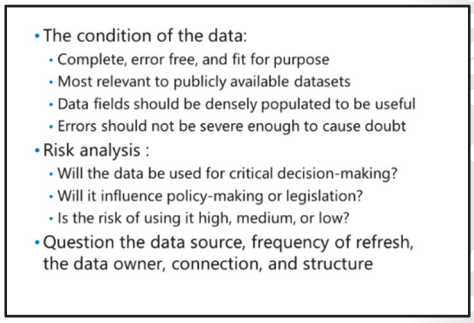
Self-service BI enables business users to take advantage of publicly available data. Data repositories such as Microsoft Azure Marketplace, Amazon Web Services, and Wikipedia, all provide datasets, some of which are free. These can easily be incorporated into a self-service BI solution, by downloading 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 can enhance 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 and the same security principles can be 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 can also be cached.

Data Reliability



Data reliability refers to the condition of data, and whether it is complete, and sufficiently free of errors, so that the data is 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 be of the correct data type. This ensures that the data can be analyzed with confidence.

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 as a result of 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 concerned of a sensitive nature?
- Will the results of the data be made publicly available?

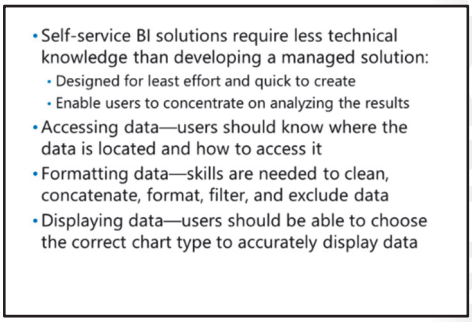
When performing risk analysis to determine the reliability of the data, the following questions should be answered as part of the assessment:

- **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 on it?
- **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.



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 may be required for accessing data held by third parties, including SaaS providers, and publicly available datasets.

Formatting Data

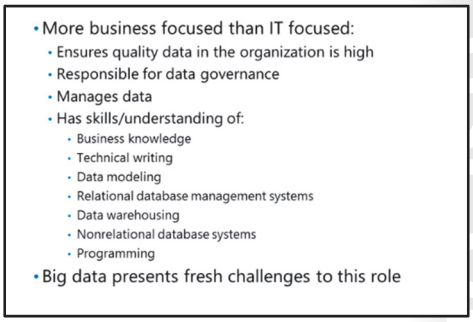
Transforming and formatting data is an important step in the process of building reports. If the data is not right, then the results will not be right. Users must understand the principles and structures of data that is sourced from a relational database, a data warehouse, or an unstructured big data 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 resources on the Internet to show examples of all the chart types and how they can be used, which 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 the quality of the data in an organization is high and is responsible for data governance. With the proliferation of data in organizations, a data steward is now 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:

- 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 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.

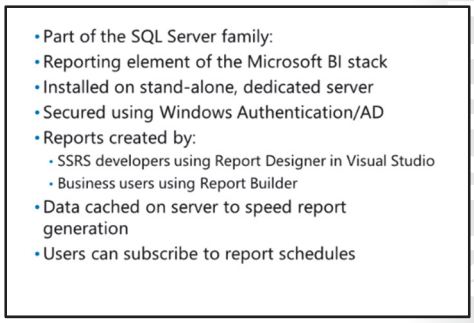
Lesson Objectives

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® Server can be used for sharing and collaborating.
- See the benefits of using Power BI Desktop as your self-service BI solution.

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 first 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. The Reporting Services service 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 can be managed using Windows authentication and Active Directory (AD).



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 SQL Server 2016 Reporting Services. 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 may be unknown at design time. 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 can receive an email message with a report attached. With the right permissions, users can generate reports using the Report Manager portal, and subscribe to report subscriptions. Reports can be delivered as soon as data is updated, or it 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.

More recently with the launch of SQL Server 2016, SSRS has been upgraded to support HTML5 rendering, and mobile reports. Also, it now includes a wider range of charts, including sunburst, and tree map charts, which were introduced in SQL Server 2016 Reporting Services.

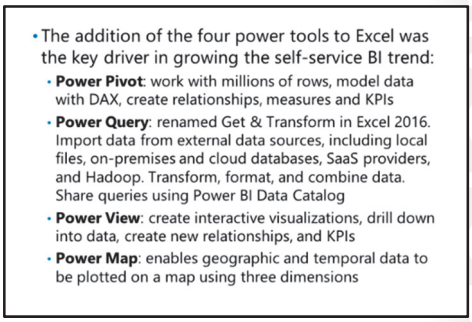
For more information on using Report Builder for SQL Server 2016, see:

Report Builder in SQL Server 2016

http://aka.ms/pjna7f

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 four power tools—Power Pivot, Power Query, Power View, and Power Map—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 four power tools have liberated business users, and reduced the workload on IT to develop and manage a time-consuming BI solution.



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 can 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, 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 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 can 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. Data types can be altered to ensure number and currency values are aggregated and displayed correctly. Text data can be cleaned and trimmed, and the case can be changed to upper, lower, or title case. You can also write your own transformations using the M Language.
- **Combine**: you can 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 can 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 are recorded 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. Also, 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**.

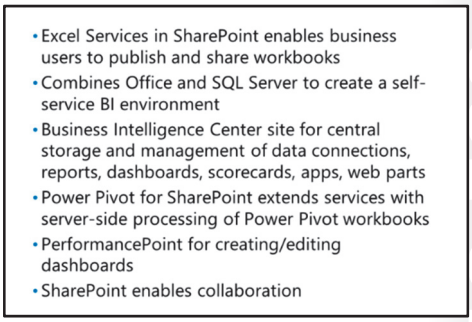
Power Map

With Power Map, you can plot and visualize your geographic data in three dimensions. The third dimension offers greater insight into geographic and temporal data, that may not be discovered using a two-dimensional map. Power map can take millions of rows from a table or data model, and plot these on a map. You can also create custom regions to highlight localized data models.

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

SharePoint Server

Excel Services in SharePoint enables business users to publish Excel workbooks for sharing with colleagues. SharePoint combines with Office, Excel, and SQL Server to create a self-service BI environment. SharePoint 2013 features a Business Intelligence Center site, so users can centrally store and manage their data connections, reports, dashboards, scorecards, apps, and web part pages.



Users can create a dashboard experience, using workbook data in the browser window. After importing your data into Excel, applying transformations and formatting, you can use the charts and maps to visualize the data. You can then publish this to an on-premises SharePoint server, and colleagues can view and interact with the data, in addition to opening it locally in Excel. Online data can be refreshed, and users can search for values within the data. The Excel Services application loads the data, runs calculations, and renders the report in the browser window. It can use live data connections, so analysis can be performed on the most up-to-date of data.

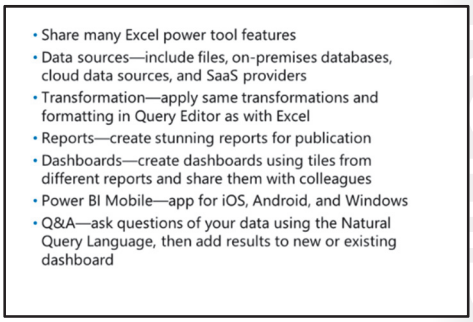
Power Pivot for SharePoint extends the services offered by Excel Services in SharePoint, by delivering server-side processing of Power Pivot workbooks. The Power Pivot document gallery enables users to browse published Power Pivot workbooks, and configure when data is refreshed. Using the Power Pivot Services, the embedded data model is deployed to an Analysis Services instance, where Excel Services can query the model.

PerformancePoint services enables the creation and sharing of centrally managed dashboards. The reports can be updated at any time, and can interact with KPIs and scorecards. A web part feature can filter the data to deliver a specific report, or enable drill-down into the data. PerformancePoint includes a Dashboard Designer that offers a friendly experience for creating and editing dashboards.

Sharing workbooks on a SharePoint Server removes the need for Excel files to be emailed and transferred around the organization. Users can collaborate on the same projects, facilitating the sharing of ideas, analysis, and data insights.

Power BI Desktop

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. Reports are uploaded to the Power BI service, where colleagues can 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.



Data Sources

From Power BI 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, and SAP HANA.
- **Azure**: Microsoft Azure SQL Database, Microsoft Azure SQL Data Warehouse, Microsoft Azure Marketplace, Microsoft Azure HDInsight, Microsoft Azure Blob Storage, Azure HDInsight Spark, Microsoft Azure DocumentDB, Microsoft Azure Data Lake Store.
- **Online Services**: the main SaaS providers are supported, including Dynamics CRM, Facebook, Google Analytics, Salesforce, GitHub, MailChimp, Marketo, QuickBooks Online, Webtrends, and Zendesk.
- **Others**: you can also import from a webpage, and 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 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 your 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 can be 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 can be published to the Power BI service, or Pyramid Analytics.

Power BI Mobile

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

Q&A

One very useful feature of Power BI is Q&A. This means you can ask questions of your data using the natural query language. You can type in a topic, such as **Total sales last year in Canada**. You can also specify which chart visual the data should be presented in. When Power BI returns the result, you can pin the visual to a new or existing dashboard.

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.

Demonstration Steps

1. In Power BI Desktop, on the **Home** tab, click **Publish**.
2. If you are prompted to save your changes, click **Save**.
3. In the **Sign in to Power BI** dialog box, click **Sign in**.
4. Enter the email address and password for your account, and click **Sign in**.
5. The report will then be published to the Power BI portal.
6. In the **Published to Power BI** dialog box, click **Open 'Adventure Works Sales.pbix' in Power BI** to view the report online.
7. When the browser opens, if you are prompted to enter your Power BI credentials, click **Sign in**.
8. Enter your email address and password, click **Sign in**, and then wait for the report to open.
9. On the **Sales by Gender and Marital Status** column chart, click **Pin visual**.
10. In the **Pin to dashboard** dialog box, select **New dashboard**, and type **Adventure Works Sales**, and then click **Pin**.
11. On the **Orders by Color** donut chart, click **Pin visual**.
12. In the **Pin to dashboard** dialog box, click **Existing dashboard**, in the list, click **Adventure Works Sales**, and then click **Pin**.
13. On the **Sales by Month** column chart, click **Pin visual**.
14. In the **Pin to dashboard** dialog box, click **Existing dashboard**, in the list, click **Adventure Works Sales**, and then click **Pin**.
15. In the upper-left corner of the window, below the PowerBI icon, click **Show the navigation pane** icon.
16. Under **My Workspace**, under **Dashboards**, point out the star icon to indicate a new dashboard, and click **Adventure Works Sales**.
17. Drag the lower-right corner of the **Sales by Month** column chart, and expand it so it is as wide as the two charts above it.
18. Close Internet Explorer.
19. In the **Published to Power BI** dialog box, click **Got it**.
20. Close Power BI Desktop.

Check Your Knowledge

Question	
Which of the following is not an Excel power tool?	
Select the correct answer.	
	Power Map
	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.

Estimated Time: 60 minutes

Virtual machine: **10989B-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa$$w0rd**

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

1. Ensure that the 10989B-MIA-DC, 10989B-MIA-SQL, and MSL-TMG1 virtual machines are running, and then log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. Run **Setup.cmd** in the **D:\Labfiles\Lab01\Starter** folder as Administrator.
3. If you do not already have a Power BI login, browse to **https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-signing-up-for-power-bi-with-a-new-office-365-trial**, and then follow the steps to create an account.
4. Download and install Microsoft Power BI Desktop from **https://www.microsoft.com/en-us/download/details.aspx?id=45331** using the default options.

▶ Task 2: View Reports in SharePoint Server

1. Open the **Adventure Works Sales.xlsx** file from the **D:\Labfiles\Lab01\Starter\Project** folder.
2. Click **Enable Content** if the security warning shows.

3. In the **SalesPerson** sheet, click **Summarize with PivotTable** to create a new tab.
4. Add the **FirstName** to the **Axis**, and the **SalesYTD** to the **Values**.
5. Add a new **PivotChart**, as a **clustered column chart**.
6. Move the chart to a new sheet called **Sales Person Chart**.
7. Save the file to the **http://mia-sql/sites/adventureworks/Shared Documents**.
8. Open Internet Explorer from the taskbar, and navigate to **http://mia-sql/sites/adventureworks/Shared Documents**.
9. Open the **Adventure Works Sales** workbook online, and then view the **Sales Person Chart**.
10. Close Internet Explorer, and then close Excel.

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

1. Open **Power BI Desktop**.
2. Import the **FactInternetSales** table, and related tables from the **AdventureWorksDW2016** database.
3. Name the file **Adventure Works Sales**, and save the file to the **D:\Labfiles\Lab01\Starter\Project** folder.
4. Leave Power BI Desktop open for the next exercise.

▶ Task 2: Add Visualizations to the Report

1. Drag the **SalesAmount** field from the **FactInternetSales** table onto the report to create a column chart.
2. Add the **EnglishDayNameOfWeek** field from **DimDate** to the **Axis**.
3. Move the chart to the top left-hand corner of the report, and expand it to show all the days of the week.
4. Change the title to **Sales by Day of Week**.
5. Center align the chart title.
6. Drag the **SalesAmount** field from **FactInternetSales** onto the report, and add **CalendarQuarter** from **DimDate**.
7. Move the **CalendarQuarter** to the **Axis** property.

8. Rename the title of the chart **Sales by Calendar Quarter**, and center align the text.
9. Change the data colors so quarter 1 is **red**, quarter 2 is **blue**, and quarter 3 is **yellow**.
10. Move the chart to the right of the Sales by Day of Week chart, and make them the same size.
11. Drag **SalesTerritoryCountry** onto the report to create a map visual, and add **SalesAmount** from **FactInternetSales**.
12. Rename the map title, **Sales by Country**, and center align the title.
13. Expand the map to show all values.
14. Drag the **CommuteDistance** field from **DimCustomer** onto the report under the Sales by Calendar Quarter chart. Add **SalesAmount** from **FactInternetSales**.
15. Change the chart to a donut.
16. Rename the chart **Sales by Commute Distance**, and center align the text.
17. Save the file.

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

1. Publish the report to the Power BI Service. Sign in using your email address and password.
2. Pin **Sales by Day of Week** to a new dashboard named **Adventure Works Sales**.
3. Pin **Sales by Calendar Quarter** to the **Adventure Works Sales** dashboard.
4. Pin **Sales by Country** to the **Adventure Works Sales** dashboard.
5. Pin **Sales by Commuter Distance** to the **Adventure Works Sales** dashboard.

▶ Task 2: Ask Questions of Your Data

1. In the Adventure Works Sales dashboard, click in the **Ask a question about your data** box.
2. View the data in **DimProducts**.
3. Ask Power BI how many customers there are.
4. Pin the visual to the **Adventure Works Sales** dashboard.
5. Ask Power BI who the oldest customer is.
6. Ask Power BI how many products are red.

7. Ask Power BI which country has the most male customers, and pin the results to the **Adventure Works Sales** dashboard.
8. Pin the visual to the **Adventure Works Sales** dashboard.
9. Under My Workspace, Dashboards, click **Adventure Works Sales** to refresh the dashboard.

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 has been driven by Microsoft's commitment to improving Excel® and Power BI, both products having seen many enhancements over recent years. However, despite Microsoft enabling deeper data analysis with the four power tools added to Excel—Power Pivot, Power View, Power Query, and Power Map—they are not fully integrated into the Excel interface, existing instead 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 rapidly publish to a cloud-based portal to easily share reports. 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.
- Create dashboards for mobile devices on the iOS, Android, and Windows 10 operating systems.

Lesson 1
Power BI

In this lesson, you will learn about the main features of Power BI that will help you create and publish reports to the Power BI portal, where you can create dashboards.

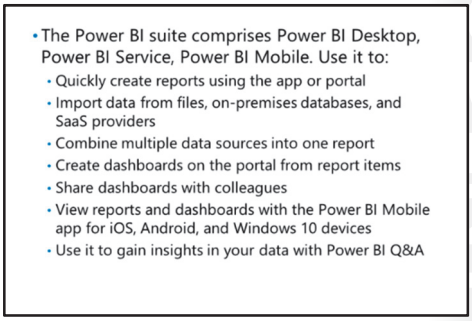
Lesson Objectives

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.
- 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?

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.

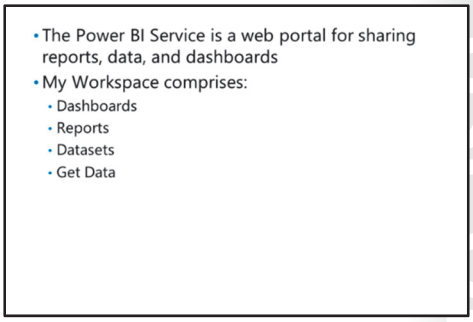


Power BI comprises the Power BI Desktop app, the Power BI Service, and Power BI Mobile. You import data and create reports using the desktop app, transforming your data into rich, interactive visualizations. Using Power BI Desktop, you can connect to a wide range of data sources, and combine data from multiple sources within one report. You can connect to, but are not limited to, Microsoft SQL Server, Microsoft Azure SQL Database, Excel, Oracle, and MySQL.

Furthermore, you can 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 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 can then be used in dashboards. You can then view and interact with reports and dashboards using the Power BI Mobile app for iOS, Android, and Windows 10 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

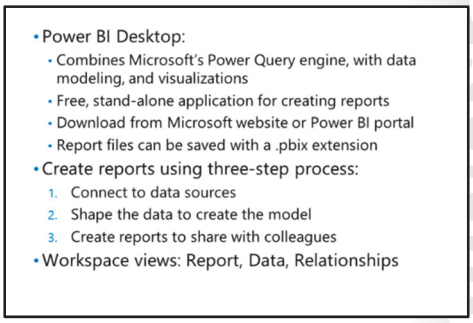


The PowerBI.com web portal, part of the Power BI Service, is where you publish your reports, create dashboards and content packs, and share data with others in your organization. Microsoft is applying weekly updates to the portal, continuously enhancing the service. Furthermore, if you require functionality that is not on the portal, you can 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.

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. Dashboards can be included in content packs and shared with others in your organization. When you add a content pack created by someone else in your organization, or from a service, any dashboards included in the content pack are included in the list.
- **Reports**. All the reports you have published from Power BI Desktop are listed alphabetically in this section. If you add content packs to your workspace that include reports, they are included here and integrated into the list.
- **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 can use these datasets to create new reports while signed in to the portal. When you add a content pack, it is very likely to include datasets. These are shown here, with an icon indicating that they have been shared with you.
- **Get Data**. You can import data into the portal from a number of sources. You can create shared content packs, or connect to content packs provided by SaaS companies. Data can be imported from files, including reports and workbooks in Excel, CSV, and Power BI Desktop. You can connect to your local file system, SharePoint® team sites, OneDrive® Personal, and OneDrive Business.

Power BI Desktop



Power BI Desktop combines Microsoft's 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 can be downloaded from the Microsoft website, or from the Power BI portal. The Power BI Desktop app is available as a free download. You can use this powerful tool to connect to a plethora of data sources, so you can create datasets and reports that could be shared. Report files can be saved 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:

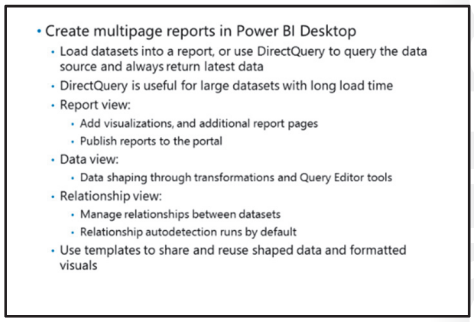
1. **Report View**. This is your main workspace for adding report items, such as bar charts, maps, and pie charts, and displaying data using these report items.
2. **Data View**. You can use the data view to view imported datasets, in addition to shaping the data using transformations and M expressions.
3. **Relationship View**. Power BI autodetects relationships from structured data sources, such as SQL Server, or Microsoft Access®. Autodetection may not work for flat files, but after you have imported your 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-hand corner of the screen. You can also use this link to switch accounts when signed in. To sign out, select **File**, **Sign Out**.

Reports

You can 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 being used in visualizations.



You have the choice to load the data into the report—and either refresh it manually or on a schedule—or you can use DirectQuery, which does not import any data. After you import data, the data 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 this 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, this is the view you are presented with. 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 can 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 can begin to design your report only after you have added at least one dataset.

You can add pages to your report from the **Insert** ribbon menu, which gives you the option of **New Page**, or **Duplicate Page**. Report pages can be added and deleted using the tab at the bottom of each page.

After you have added a dataset, the **Measures** menu is activated, and enables you to create measures and add columns. The **Publish** button on the **Share** menu prompts you to sign in to your Power BI account, so you can upload reports to the portal, from where you can create dashboards.

The Data View

You can use the data view to perform transformation operations on your imported datasets, so data can be shaped appropriately for the reports you are producing. Click on a dataset to view the imported rows and see the data you want to work with. You can 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 on any column and choose **Edit Query**, which will open the Query Editor window. From the Query Editor window, you can split columns, apply statistical functions, pivot and unpivot columns, and more. The Advanced Editor 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 Query Editor window where you can shape your data.

The Relationship 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 relationships view enables you to manage and create relationships. You can set the cardinality to **Many to One (*:1)**, **One to One (1:1)**, or **One to Many (1:*)**. The cross filter can be switched between **Both** or **Single**. You can also delete relationships.

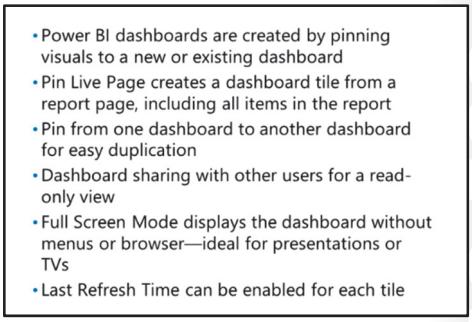
Creating Report Templates

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 a number of reports that share data, visuals, and formatting, templates are a useful feature for avoiding the duplication of work, while ensuring consistency across reports.

You can 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 choose **File**, **Save As**, and then select **Save as type: Power BI Template File (*.pbit)**. Alternatively click **File**, **Export**, then **Power BI Template**. You can open an existing template by clicking **File**, **Import**, and then **Power BI Template**, or **File**, **Open**, and then navigate to the location of the template file, selecting **Power BI Template File (*.pbit)** from the 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, and click **Publish**. You may 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 will be prompted to confirm the overwrite of any existing datasets that have changed.



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 can 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 report 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 dashboard. Click **Open menu** (the ellipsis) on a tile to open the **Select an Action** menu, then click **Pin visual**. 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. Click **Share** to open the Share dashboard window. You can then enter one or more email recipients, along with a message to describe the dashboard.

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 in Power BI. Click **Esc** or **Exit Full Screen Mode** to return to Power BI.

The **Fit to Screen** button improves 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 **Fit to Screen** button zooms in to enlarge the items and fill as much empty 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**, can be enabled at an individual tile level by using the **Tile Details** menu.

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.

Demonstration Steps

Create a Report with Power BI Desktop

1. Ensure the MSL-TMG1, 10989B-MIA-DC and 10989B-MIA-SQL virtual machines are running, log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. Run **D:\Demofiles\Mod02\Setup.cmd** as an Administrator, when prompted click **Yes**, type **Y**, and then press Enter.
3. When the script completes, press any key to close the window.
4. Start SQL Server Management Studio, and connect to the **MIA-SQL** database engine instance using Windows authentication.
5. Open the **Demo.ssmssln** solution in the **D:\Demofiles\Mod02\Demo** folder.
6. In Solution Explorer, expand **Queries**, and then open the **1 - Power BI.sql** script file.
7. On the taskbar, click **Power BI Desktop**.
8. In the **Power BI Desktop** window, click **Get Data**.
9. In the **Get Data** dialog box, click **Microsoft Azure SQL Database**, and then click **Connect**.
10. In the **SQL Server Database** window, in the **Server** box, type the URL of the Azure server <Server Name>.database.windows.net (where <Server Name> is the name of the server you created), and in the **Database** box, type **AdventureWorksLT**.
11. Expand **Advanced options**.
12. In SQL Server Management Studio, copy the **1 - Power BI.sql** query.
13. In Power BI Desktop, paste the query into the **SQL statement (optional)** box, and then click **OK**.
14. In the **Access a SQL Server Database** window, click **Database**.
15. In the **Username** box, type **Student**.
16. In the **Password** box, type **Pa$$w0rd**, and then click **Connect**.

17. The data preview window will appear, click **Load**.
18. If the **Connection Settings** window opens, leave **Import** selected, and then click **OK**.
19. In the **Visualizations** pane, click **Stacked column chart**.
20. In the **Fields** pane, under **Query1**, select **ProductName** and **TotalSales**. The chart will auto populate. Expand the chart control to horizontally show the full names of the products.
21. In the **Visualizations** pane, click **Format**.
22. Expand **Title**, and change the **Title Text** value to **Top 10 Selling Products**.
23. Next to **Alignment**, click the **center** icon.
24. Toggle **Data labels** to be **On**.
25. Expand the **Data colors** list, and choose another color to change the bars on the chart.
26. On the **File** menu, click **Save As**. Name the report **Adventure Works Sales**, and save to the **D:\Demofiles\Mod02\Demo** folder.
27. Leave Power BI Desktop and the report open for the next demonstration.
28. Close SQL Server Management Studio.

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 sign up to PowerBI.com using a Hotmail email account.
	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 content packs, learn how the natural query language can answer questions about your data, and understand tenant management.

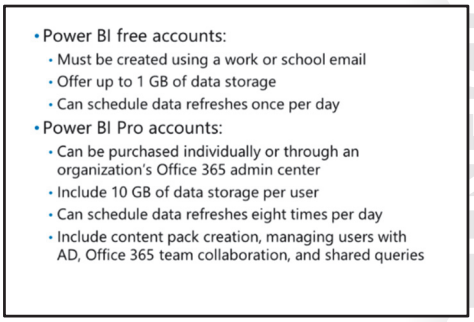
Lesson Objectives

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 content packs.
- Describe the benefits of the natural query language.

Licensing

Power BI offers a straightforward licensing model, with a choice of a free Power BI account, or a Power BI Pro subscription account. A free account requires a work or school email address, so personal domains such as Gmail, Hotmail, or Yahoo, are not permissible. Power BI Pro accounts can be purchased individually, or for an organization using the organization's Office 365 Admin Portal. You do not need to purchase an Office 365 subscription to use a free Power BI account.



Free Power BI account users can do the following:

- Store up to 1 GB of data.
- Schedule content to refresh once per day.
- Consume up to 10,000 rows of streaming data per hour, in dashboards and reports.
- Create reports and datasets with Power BI Desktop.
- Import data and reports from Microsoft Excel, CSV, and Power BI Desktop files.
- Create and share reports and dashboards with other Power BI users.
- View dashboards on a mobile device using any of the Power BI Mobile apps for iOS, Android, and Windows.
- Ask questions using natural language queries.
- Consume content packs from SaaS providers, including Bing®, Salesforce, Zendesk, and MailChimp.
- Use the **Publish to web** feature to share Power BI reports on public websites, including blogs.

A Power BI Pro subscription includes all of the above features, along with the following:

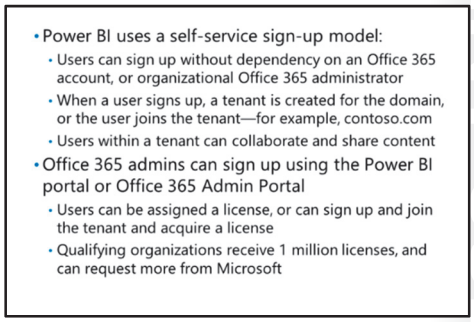
- Store up to 10 GB data.
- Schedule content to be refreshed up to eight times a day.
- Consume up to 1 million rows of streaming data per hour in dashboards and reports.
- Import on-premises data using the **Data Connectivity Gateways**.
- Use the Power BI REST API to push live data into a Power BI dataset.
- View live data by directly connecting to sources, rather than bringing the data into Power BI.
- Manage user access with Microsoft Active Directory®.
- Work with other team members using Office 365 Groups in Power BI.
- Create and publish organizational content packs.
- Share data queries using the **Data Catalog**.

Organizations can have a mix of free and Power BI Pro accounts. However, to consume Power BI Pro content, users must have a Power BI Pro license. For full service details and local pricing, see:

Power BI Pricing

http://aka.ms/Qz9yz8

Tenant Management



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 automatically created. 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 can 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 can sign up to Power BI via the PowerBI.com website, or through the **Purchase Services** section within the **Office 365 Admin Portal**. Administrators can 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 Tenant Management Guidance

http://aka.ms/Ug2h9n

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—which 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.

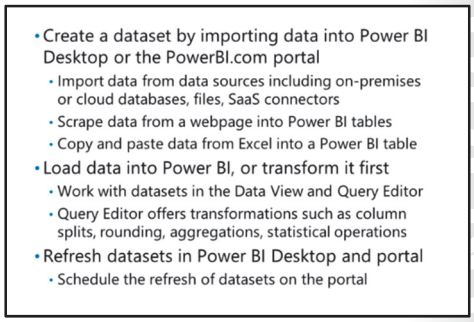
For more information on the Power BI architecture, and Power BI security, see:

Power BI Security

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. In Power BI Desktop, you have a wider 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. Recent additions include the R Script connector for querying a local R installation, and the SQL Server Analysis Services (SSAS) multidimensional model connector. Database connectors include:

- SQL Server
- SSAS tabular and multidimensional models
- R Script
- 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

Using 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:

- Bing
- Google Analytics
- Intuit QuickBooks
- MailChimp
- Facebook
- Microsoft Dynamics® CRM Online
- Salesforce
- GitHub

Users can connect to SaaS applications and import the data to create reports and dashboards. Due to its flexibility, Power BI can combine 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 may 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. From the **Home** ribbon, click **Enter Data** to open the Create Table window. **Right-click** and choose **Paste** to copy data from another file. You can work with 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 can preview and select the data you want to import. From here you can select **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 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 Query Editor. The Query Editor offers more complex transformations than the Data View—such as column splits, rounding, aggregations, statistical and scientific operations.

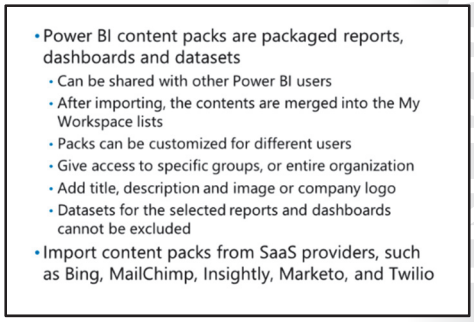
Refreshing Data

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. 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 Query Editor window, you may see a message such as "This preview may be up to 35 days old". You can click the **Refresh** button to update the data, though the data may not have actually been altered. This applies to each table within your dataset, so you have control over exactly which tables to update.

Content Packs

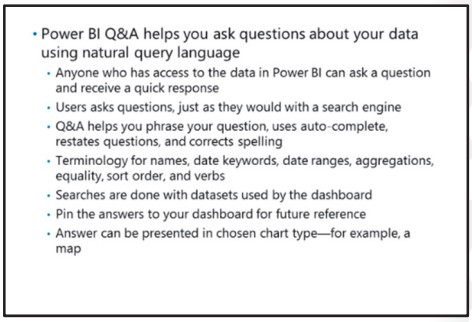
Content packs are packaged reports, dashboards and datasets, which can be shared with other Power BI users in your organization. When you connect to a content pack on the PowerBI.com portal, the report items are merged into your workspace lists. Users with a free Power BI account can view content packs, but they cannot create them. Content packs can be created 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, as each department is likely to want different data with which to measure performance.



When you publish a content pack, 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. The content pack can be customized with a title, and a description to help users determine if the content pack is applicable to their needs. You can also upload an image or company logo for the content pack. You can 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 content pack is then available in your organization's content gallery. Users who can the content pack can create new dashboards from the contents.

Furthermore, you can import content packs from SaaS providers such as Adobe Analytics, AlpineMetrics Sales Prediction, Insightly, Marketo, and Twilio. To add a content pack from a SaaS provider that you have an account with, click **Get Data**, then under **Services**, click **Get**. In the Microsoft AppSource window, under **Apps**, either click the SaaS logo to view provider details, or click **Get**. You will be prompted to enter your customer details for the service. After you have been authenticated, you can import a content pack with reports and dashboards that have been designed to visualize your data without you needing to do any work.

Natural Language Queries



Finding answers to questions can be difficult if your organization has a lot of 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 can 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 who has access to Power BI can quickly find answers, as no additional programming skills are needed.

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 can 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 by 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".

Question: What are the benefits of using Power BI in an organization looking to create reports to analyze their data?

Demonstration: Creating a Content Pack

In this demonstration, you will see how to:

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

Demonstration Steps

1. In Power BI Desktop, on the **Home** ribbon, click **Publish**.
2. If you are prompted to save your changes, click **Save**.
3. In the **Sign in to Power BI** window, click **Sign in**.
4. In the **Sign in to your account** dialog box, enter the email address and password for your Microsoft account, and then click **Sign in**.
5. The report will then be published to the Power BI portal.
6. In the **Publishing to Power BI** dialog box, click **Open 'Adventure Works Sales.pibx' in Power BI** to view the report online.
7. When the browser opens, if you are prompted to enter your Power BI credentials, enter your email address and password, and wait for the report to open in Internet Explorer.
8. When the report is visible, click **Pin Live Page**.
9. In the **Pin to dashboard** dialog box, click **New dashboard**, in the text box, type **Adventure Works Sales**, and then click **Pin live**.
10. In the upper left, click **Show the navigation pane**. The dashboard will appear under the **Dashboards** list.
11. Click **Settings**, then click **Create content pack**.
12. In the **Choose who will have access to this pack** window, click **My entire organization**.
13. In the **Title** box, type **Adventure Works Sales**.
14. In the **Description** box, type **Top 10 selling products**, and then click **Upload**.
15. In the **Choose File to Upload** dialog box, navigate to **D:\Demofiles\Mod02\Demo\Demo**, click **content_pack.png**, and then click **Open**.
16. Under **Dashboards**, select the **Adventure Works Sales** check box. The Report and Dataset are automatically added, and then click **Publish**.
17. Click **Get Data**, and then under **My organization**, click **Get**.

18. In the **AppSource** window, the content pack appears in the list under **My organization**.
19. Close Internet Explorer, and then close Power BI Desktop.

Lesson 3
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.

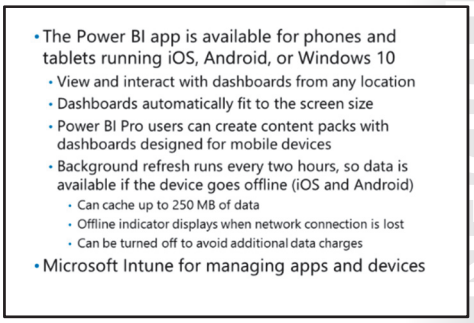
Lesson Objectives

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.
- Use the features in the Power BI app for Windows 10 phones.

Creating Dashboards for Mobile Devices

The Power BI 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 can then be viewed on a device running the app. Dashboards automatically adjust and resize to fit the target screen size, and the data refreshes in real time, for up-to-the minute results. This means you do not need to do any additional formatting, or create resized visuals for mobile reports.



Design Considerations

Although dashboard items scale to size, you may 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 can make allowance for the screen size. A bar chart with 30 columns may display perfectly on a tablet, but may be more difficult to view on a mobile, even in landscape mode.

Best Practice: If your organization uses Power BI Pro, and users connect using mobile phones, you can create content packs with reports and dashboards designed for the smaller screen. You can scale down the size of visuals, and ensure the most important data is placed at the top of the dashboard, so these items are shown first when vertically scrolling through dashboard items.

Offline Data

Power BI runs a background data refresh every two hours, so if you go offline, data remains reasonably up-to-date. While offline, you continue to have access to all items in your My Workspace, and can interact with dashboards. Power BI can cache up to 250 MB of data; however, data sources requiring an active connection cannot show data. Reports can also be viewed in read-only mode, and you cannot filter, sort, or use slicers. This scheduled refresh can be turned off to avoid data usage charges. An offline indicator

displays at the top of your dashboard when you have no signal. The offline data feature is currently only applicable to devices running iOS and Android.

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.

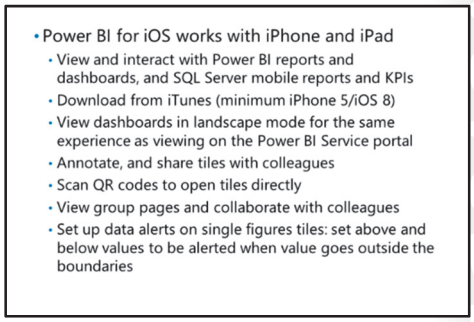
For more information on managing your devices with Intune, see:

Configure Power BI 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 and iPad, and is part of the family of mobile BI experiences for Power BI. You can use the app to view and interact with your organization's dashboards from anywhere. 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.



You can download the app by searching for Microsoft Power BI on iTunes, or by using the following link:

Dashboards in the iPhone app (Power BI for iOS)

http://aka.ms/Gug35u

You do not 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 if the app works before you sign in and view your organization's content. If you have an iPhone, it needs to be at least iPhone 5, running iOS 8.0.

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 can 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 can 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**, then tap the **Share** icon. You can tap the **Pencil** icon to add arrows, lines and symbols using the freehand tool, and highlight areas of the tile. You can also change the color of the lines you add. Tap the **Text** icon to add text onto the tile using the keypad, and tap the **Smiley Face** icon to paste in emoticons.

It is easy to share your dashboards with colleagues. Open the dashboard you want to share, and tap the ellipsis. Click **Invite**, then **Invite a Colleague**. Type the recipients' names or email addresses in the **Add names or emails** box. You can also include a message in the **Add text** box. You can also toggle the setting to **Allow recipients to share this dashboard**. Click **Send**. The recipients receive an email message inviting them to add the dashboard. This email expires after one month, and you can view whether invitation requests have been accepted or not. You can only send emails to colleagues in the same domain, and they need Power BI to view the dashboard. However, you can send a snapshot of a tile from the iPhone app to anyone in or outside of your domain, but they cannot interact with the tile or open the dashboard.

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, then click **Focus mode**. This opens the tile. Click the **ellipsis**, and then click **Generate QR code**. After it has been generated, you can download the code as a .jpg file. You can use this file in email messages and PowerPoint® slides, or save it to your phone, or print it.

To scan a QR code, select **QR Scanner** from the main menu in Power BI Mobile, 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.

Groups

Power BI groups are built upon Office 365 groups—you can use them to collaborate with other members, and interact with group reports and dashboards. You can view your groups in the Power BI for iOS app, by tapping the **Options** icon, and then selecting a group. The group page is displayed.

Data Alerts

Data alerts can be added to tiles that display a single number. You can 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.

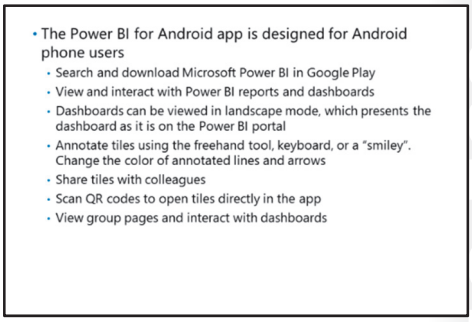
To create an alert, open the tile that you want to monitor, and tap the **Bell** icon. Use the **Above** and **Below** toggles to enable (or disable) the alerts, and add your values. Tap **Save**. The values are checked each time the data in the dashboard is refreshed. The alerts you create are only available to you, so if you share a tile, or dashboard, other users cannot see them—but they can create their own. You can also create alerts on tiles imported through content packs.

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 an 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 using the following link:

Dashboards in the Android app for Power BI

http://aka.ms/d0emi0



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**, then tap the **Share** icon. This opens the annotate bar, and offers the option of adding lines and shapes using the **Paintbrush**, tapping the **Smiley** to add stickers, and using **AA** to add comments using the keyboard. When you have finished your annotations, tap **Share** to send the tile to your colleagues.

You can share dashboards using the **Invite a colleague** function, located under the ellipsis menu on the dashboard view. If you are the dashboard owner, you can view the colleagues you have invited, and see whether they have accepted your invitation. Type an email address, and an optional message if you don't want the default Power BI message. Tap the **Airplane** icon to send the invite. You can also allow colleagues to share the dashboard with others. 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.

Groups

You can use the Power BI for Android app to interact with group dashboards. Groups are listed under **My Workspace**. Tap a group to show the group page, and display the dashboards belonging to the group, then tap a dashboard to view it.

Power BI for Windows 10

As with the Power BI mobile app for Android, the app for Windows 10 has not yet caught up with some of the available features on the app for iOS. No doubt this gap will be closed in the near future. However, the Windows 10 app does boast a handful of features not yet in the iOS or Android versions. You can download the Power BI mobile app for Windows 10 from the Windows Store on your phone, or see:

Microsoft Power BI - Windows Apps on Windows Store

http://aka.ms/Rxmqxc

To enable the app to perform optimally, ensure your device has at least 1 GB RAM, and 8 GB internal storage. After installing and opening the app, sign in to your Power BI account, and tap **Start exploring** to view your dashboards.

Viewing Mode

The Power BI mobile app for Windows 10 supports viewing dashboards in portrait and landscape mode. In portrait mode, tile items are vertically stacked and displayed with an identical width. Landscape mode displays the dashboard exactly as it is on the Power BI Service.

Interacting with Tiles

You can interact with dashboard tiles on your Windows phone in much the same way as you would on the Power BI Service. Tap a tile to open it in **Focus mode**, and tap to view items in more detail in pie, bar, and line charts. When you tap a pie chart to put it in **Focus mode**, the slicer automatically appears. Spin the chart to show each of the pie slices in detail.

Share Dashboards

You can share a dashboard with colleagues within your organizational domain. On the dashboards home page, press and hold, and tap **Invite**. Add email addresses, and optionally include a message; otherwise Power BI includes a default message. If you want the recipients to be able to share the dashboard, select **Allow recipients to share your dashboard**. Tap the **Send** icon. You can see whether your colleagues have accepted or rejected your invite, or if the invite is pending.

If you have colleagues in a different domain, you can send a read-only snapshot. Open a dashboard, and tap a tile to open it in Focus mode. Tap the **Share Snapshot** icon to share the tile.

Groups

View your groups by tapping the **Navigation** icon to show **My Workspace**. Tap a group to display the dashboards that have been published to the group.

Search Facility

You can search for reports, dashboards, and groups. Tap the **Search** icon in **My Workspace**. Power BI displays a list of recently viewed reports and dashboards, and recent groups. When you begin typing in the **Search** box, Power BI shows you the best results.

Pin a Dashboard to Your Start Screen

There are two ways you can pin a dashboard to your Windows Start screen. You can press and hold on a dashboard and tap **Pin to Start**, or tap the ellipsis while viewing a dashboard, and then tap **Pin to Start**. This is a fast and useful facility for accessing the dashboard and monitoring data.

Microsoft Power BI Community

The Microsoft Power BI Community is a useful website for finding information about all aspects of Power BI, including Power BI Service, Power BI Desktop, and Power BI Mobile. You can ask for help with your technical or design questions, gain support and find ideas, in addition to seeing upcoming events on data insights. The community includes support for the Power BI for mobile app for iOS, Android, and Windows 10. For more information, see:

Microsoft Power BI Community

http://aka.ms/Eemd41

Question: If you have an iOS, Android, or Windows 10 phone or tablet, download the Power BI app if you don't already have it. iOS users do not need a sign-up, as you can use the sample data. However, an account will be needed for the lab. Users who do not have an account can create one using the following steps:

Ensure that the 10989B-1-MIA-DC and 10989B-1-MIA-SQL virtual machines are both running, and then log on to 10989B-1-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.

Open Internet Explorer from the taskbar, and browse to the **Office 365 Trial sign-up** page, and follow the instructions: http://aka.ms/Y682m2.

Sign in to Power BI on your phone or tablet.

Explore the features of the Power BI app, and look at the samples and demo server, if these are available. Discuss useful features that could be added to improve the app, in addition to features you like and don't like, and how they could be useful in your organization.

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.

Estimated Time: 60 minutes

Virtual machine: **10989B-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa$$w0rd**

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 2016. You have been provided with a set of business requirements for data and will 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

1. Ensure that the MSL-TMG1, 10989B-MIA-DC, and 10989B-MIA-SQL virtual machines are running, and then log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. Run Setup.cmd in the **D:\Labfiles\Lab02\Starter** folder as Administrator. You may receive a prompt asking whether you want to close SQL SERVER LAUNCHPAD. If so, type **Y**.
3. If you do not already a Power BI login, go to **https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-signing-up-for-power-bi-with-a-new-office-365-trial**, and follow the steps to create an account.
4. Download and install the **Microsoft Power BI Desktop** from **https://www.microsoft.com/en-us/download/details.aspx?id=45331**.

▶ Task 2: Connect to SQL Server from the Power BI Desktop

1. Open the **Power BI Desktop** application.
2. Connect to **MIA-SQL** and the **AdventureWorks2016** database using the **Get Data** tool in Power BI Desktop.
3. Add the **Sales.vSalesPerson** data.
4. Add the **Sales.vStoreWithDemographics** data.
5. Use a query to import data for the top 10 selling products.

▶ Task 3: Add Charts to the Report

1. Add a bar chart to the report to display the year-to-date (YTD) sales for each sales person.
2. Resize the bar chart to display all sales persons.
3. Change the bar color for the three highest sales.
4. Add a pie chart to show the proportion of employees by specialty.
5. Add a bar chart to display the top 10 selling products.
6. Change the top 10 selling products bar to a donut chart.
7. Create a bar chart by dragging a field on to the report canvas.
8. Rename fields in the data sets.
9. Rename the report.
10. Save the report to the local machine.

▶ Task 4: Publish the Report to the Power BI Portal

1. Connect to the Microsoft Power BI portal using your Microsoft account.
2. Publish the report.
3. View the report online to check it has published correctly.

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

1. Create a dashboard in the Power BI portal.
2. Name the new dashboard.

▶ Task 2: Add Chart Items to the Dashboard

1. Pin the **SalesYTD by FirstName** chart to the **Adventure Works Sales** dashboard.
2. Pin the **LineTotal by Product** chart to the **Adventure Works Sales** dashboard.
3. Pin the **Annual Sales and Annual Revenue** chart to the **Adventure Works Sales** dashboard.

▶ Task 3: Customize the Dashboard

1. Reorder the sequence of the charts on the **Adventure Works Sales** dashboard.
2. Change the size of the **SalesYTD** chart to span the width of the above charts.
3. Change the titles and subtitles of the charts.

▶ Task 4: Display the Dashboard in Full Screen Mode

1. Use **Enter Full Screen Mode** to display the report for presentations or on a TV.
2. When in the full screen mode, use the **Fit to Screen** feature to remove excess space and fill more of the screen.
3. Click a chart and use the **Focus mode** to display and zoom in on a single tile.
4. Use **Pin visual** to create a new dashboard from the current dashboard.

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 can 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, consideration must be given to 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 rapidly publish to a cloud-based portal to easily share reports. 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®. Although Microsoft Excel® has long been the tool of choice for data analysts who work in a self-service style, 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.

Four power tools have added to Excel in recent years: Power View, Power Query (known as Get & Transform in Excel 2016), Power Pivot, and Power Map. Power BI brings much of this power into an integrated environment in the form of Power BI Desktop. Previously, Excel users have been inconvenienced by needing to transition between the four power tools, but Power BI Desktop brings these 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 foreseeable future. Power BI easily cooperates with Excel, and many other data sources. It is this ability to rapidly create reports by using data from a combination of sources that really puts the "power" into Power BI.

Objectives

After completing this module, you will be able to:

- Connect to Excel files and import data.
- 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 content packs and groups.

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, in addition to how to update and refresh data.

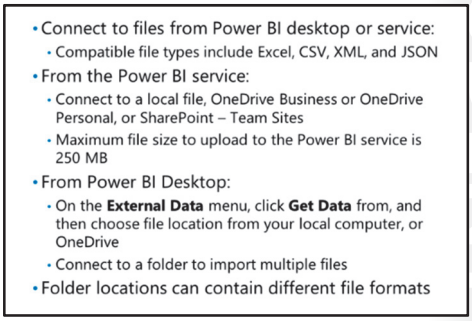
Lesson Objectives

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 can 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. Furthermore, you can 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).



You can 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 about an hour.

When importing CSV files, it is 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**. Under **File**, click **Get**. You can 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, and updates to the file are automatically reflected 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 just click **Connect** to view content that is available to you at the root level.

Connecting to Files from Power BI Desktop

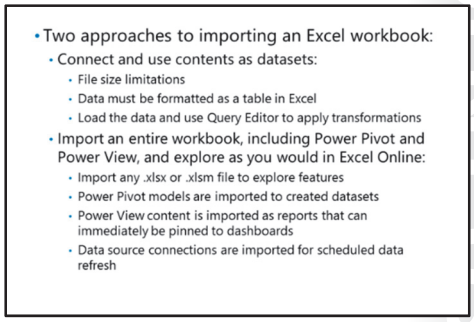
When you initially start Power BI Desktop, the splash screen gives you options to connect to data. If you have selected **show this page on startup**, you can click **Get Data** to open the Get Data window. Alternatively, on the **External Data** menu, click **Get Data**. This presents you with a list 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**, **Azure**, 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.



Importing Excel Content as a Dataset

The file that you choose to upload can be no larger than 250 MB. The workbook can consist of a data model and the core worksheet contents. Within the 250-MB limit, the core worksheet can be up to 10 MB, with the rest of the space used by the data model. If your workbook meets these criteria, you can 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 in which you can 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 further reduce the size. If the report contains a data model, you can move data from the worksheet to the data model. Ensure that you exclude columns that are not necessary to the analysis that you want to perform. If your data has originated from a data warehouse, it may 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 the following article.

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 can select the worksheets and tables that you want to import, and then click **Load** to import these immediately, or you can click **Edit** to open the 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

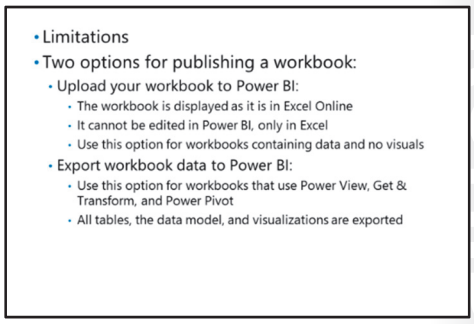
Power BI can import any Excel .xlsx or .xlsm file, enabling you to explore features as if you were using Excel Online. If you have created data models by using Power Pivot, Power BI imports your tables, calculated columns, measures, and hierarchies. Power View sheets are also imported and created as reports. When the reports have been 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 can use the connection information to make a direct connection to the data source. The data can then be 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 2016

Excel 2016 enables you to publish your workbooks straight to the Power BI service, where you can create reports and dashboards, and then share visuals with your colleagues.



Limitations

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

- The workbooks that you want to publish must be saved 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

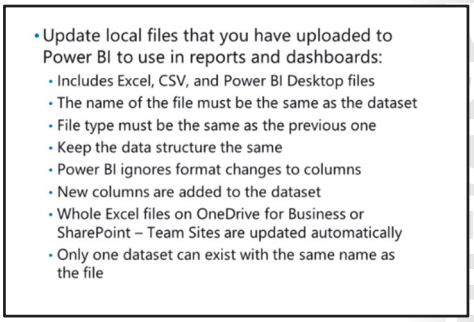
If necessary, save your Excel 2016 workbook to OneDrive for Business, open it from this location, click **File**, and then click **Publish**. This gives you two options for uploading your file to Power BI:

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 that are 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. You can also 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 either 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. As long as the file name is the same, Power BI can update the file. This applies to Excel, CSV, and Power BI Desktop files. There are several limitations that apply to this:



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

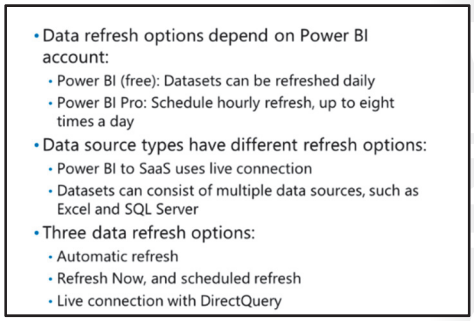
To update a file in the Power BI service:

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. Power BI displays a message to say that you already have a dataset with that name.
3. Click **Replace** to upload the updated file.

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



Subscription Types

The options that are available to you depend on whether you have a free Power BI subscription 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 do not 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 My Workspace pane. You can click the **Open Menu** icon either to **Refresh Now** or to schedule a refresh. After running Refresh Now, or a scheduled 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 can 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 can 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

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.

Demonstration Steps

Import Data from an Excel File

1. Ensure that the MSL-TMG1, 10989B-MIA-DC, and 10989B-MIA-SQL virtual machines are running, and then log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. In the **D:\Demofiles\Mod03** folder, run **Setup.cmd** as Administrator, and then click **Yes** when prompted.
3. When prompted to close the **SQL Server Launchpad** enter **Y**.
4. When the script completes, press any key to close the window.
5. On the taskbar, click **Power BI Desktop**.
6. In the **Power BI Desktop** window, click **Get Data**.
7. In the **Get Data** dialog box, click **Excel**, and then click **Connect**.

8. In the **Open** dialog box, navigate to **D:\Demofiles\Mod03\Demo\Files for Import**, click **Sales.xlsx**, and then click **Open**.
9. In the **Navigator** window, click **Sales** to show a preview of the data. Use the horizontal scrollbar to display the columns, select the **Sales** check box, and then click **Load**.
10. When the load completes, in the **Fields** pane, point out the **Sales** table. Mention that Power BI has detected columns that can be used in aggregations, as indicated by the **Sum** symbol next to the column names.

Import Data from a CSV File

1. On the **Home** ribbon, click **Get Data**.
2. In the **Get Data** dialog box, click **CSV**, and then click **Connect**.
3. In the **Open** dialog box, navigate to **D:\Demofiles\Mod03\Demo\Files for Import**, click **SalesPerson.csv**, and then click **Open**.
4. In the preview window, drag the lower-right corner to enlarge the window and display more of the data.
5. Click **Load**.
6. In the **Fields** pane, expand the **SalesPerson** table to show the columns. Mention that the two tables from different sources are now available to use together in a report. If the report is published, the tables will be part of the same dataset.
7. On the **File** menu, click **Save As**.
8. In the **Save As** dialog box, name the report **Adventure Works Sales**, and then save to the **D:\Demofiles\Mod03\Demo** folder.
9. Leave Power BI open for the next demonstration.

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
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.

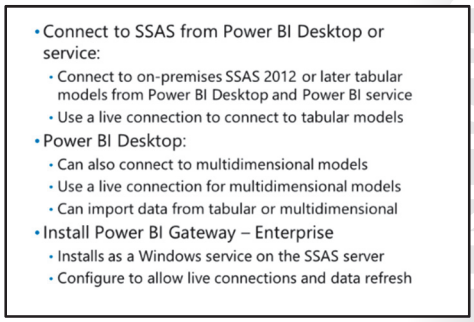
Lesson Objectives

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

- Connect to databases in SQL Server Analysis Services.
- Import data from a database in Azure SQL Database.
- Connect to Azure SQL Data Warehouse.
- Describe other data sources that are compatible with Power BI.
- Use the R script data connector to import predictive data.

SQL Server Analysis Services

You can connect to an on-premises tabular model database in SQL Server Analysis Services from both Power BI Desktop and the Power BI service. For tabular models, the installation of SQL Server Analysis Services must be SQL Server 2012 or later. 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.



To connect to a database in SQL Server Analysis Services from Power BI Desktop:

1. Click **Get Data**, and then click **Analysis Services**.
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. Toggle your connection between **Connect live** or **Import data**. In addition, you have the option to enter Multidimensional Expressions (MDX) code or a Data Analysis Expressions (DAX) query. Click **OK**.
4. In the next dialog box, type 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 Query Editor window and apply transformations. You can also edit the dataset later, after loading it.

Before you can connect to SQL Server Analysis Services by using a live connection from the Power BI service, you must configure a Power BI gateway on your server. This is applicable to both tabular and multidimensional models. You can also use the gateway for the scheduled refresh of data that is imported to Power BI. The gateway runs as a Windows® service on the server running SQL Server Analysis Services.

However, users need a Power BI Pro subscription to view content via the gateway. For more information about setting up and configuring a gateway, in addition to determining if you need one, see:

Power BI Gateway – Enterprise

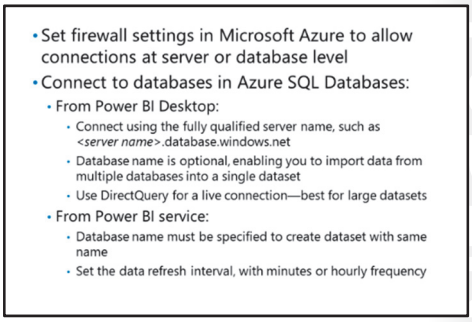
http://aka.ms/Y3jbnd

To connect to SQL Server Analysis Services from the Power BI service:

1. In the My Workspace pane, click **Get Data**.
2. In the **Databases** box, click **Get**.
3. Click **SQL Server Analysis Services**, and then click **Connect**.
4. You will see a list of servers that are available to you. Power BI Gateway – Enterprise must be installed to enable a connection to SQL Server Analysis Services from the Power BI service.

Azure SQL Database

Power BI enables you 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 SQL Database, ensure that you have configured the firewall settings 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 the ***server name***, such as *<server name>*.database.windows.net, and then click **Show firewall settings**.
3. Click **Add client IP** to add your current workstation, or add a range of IP addresses, and then click **Save**.

Note: Microsoft recommends that you allow access at the database level in Azure, rather than at the server level.

Connecting from Power BI Desktop

To connect from Power BI Desktop:

1. Open Power BI Desktop, and then click **Get Data**.
2. Select **More**, and then in the **Get Data** dialog box, click **Azure**.
3. Click **Microsoft Azure SQL Database**, and then click **Connect**.
4. In the **Server name** box, type or paste the full name of the server; for example, ***<server name>*.database.windows.net**, 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.
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 can select objects from multiple databases to combine the data into a single dataset.
7. Click **Load** to import the data into Power BI, or click **Edit** to open the Query Editor window and apply transformations. Transformations can also be performed at any point after loading the data.

Databases in Azure SQL Database and DirectQuery

In addition to loading the data into Power BI, you can use DirectQuery with Power BI Desktop. DirectQuery restricts you to using a single database, but it is very useful when you want to connect to very large datasets that could take a long time to load into Power BI. This can also be problematic when making changes to report items that cause a refresh of the data, which can cause further delays, and make it cumbersome to work with 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 may 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. Click **Edit Credentials**, and then add your user name and password.

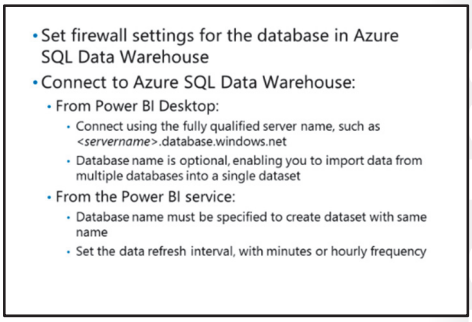
Connecting from the Power BI Service

The process for connecting to a database in Azure SQL Database from the Power BI service is just as straightforward as using Power BI Desktop:

1. In the My Workspace pane, click **Get Data** from the navigation pane.
2. In the **Databases** box, click **Get**, click **Azure SQL Database**, and then click **Connect**.
3. In the **Connect to Azure SQL Database** box, type or paste the fully qualified name of the server; for example, ***<server name>*.database.windows.net**.
4. In the **Database** box, type the name of the database.
5. If you want to set the data refresh interval, toggle **Enable Advanced Options**. You can set whether you want the data to be refreshed in minutes or hours, and the frequency.
6. The **Custom Filters** box enables you to optionally enter filtering code, such as a SELECT statement, to query the database.
7. Click **Next**. Complete the **Username** and **Password** boxes, and then click **Sign in**. The data loads into a dataset that has the same name as the database to which you connected. You can now begin creating reports and dashboards by using the dataset.

Azure SQL Data Warehouse

You can connect to a database in Azure SQL Data Warehouse in much the same way as you connect to a database in Azure SQL Database. After you have created the database in Microsoft Azure, ensure that you have configured the firewall settings to give access to your own IP address, or a range of IP addresses. Again, it is best to give access at the database level rather than at the server level.



Connecting from Power BI Desktop

To connect from Power BI Desktop:

1. Open Power BI Desktop, and then click **Get Data**.
2. Select **More**, and then in the **Get Data** dialog box, click **Azure**.
3. Click **Microsoft Azure SQL Data Warehouse**, and then click **Connect**.
4. In the **Server** name box, type or paste the full name of the server; for example, ***<server name>*.database.windows.net**, 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.
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 objects, 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.
7. Click **Load** to import the data into Power BI, or click **Edit** to open the Query Editor window and apply transformations. Transformations can also be performed at any point after loading the data.

Connecting from the Power BI Service

The process for connecting to a database in Azure SQL Data Warehouse from the Power BI service is almost identical to connecting to a database in Azure SQL Database:

1. In the My Workspace pane, click **Get Data** from the navigation pane.
2. In the **Databases** box, click **Get**, click **Azure SQL Data Warehouse**, and then click **Connect**.
3. In the **Connect to Azure SQL Data Warehouse** box, type or paste the fully qualified name of the server; for example, ***<server name>*.database.windows.net**.
4. In the **Database** box, type the name of the database.
5. If you want to set the data refresh interval, toggle **Enable Advanced Options**. You can set whether you want the data to be refreshed in minutes or hours, and the frequency.
6. The **Custom Filters** box enables you to optionally enter filtering code, such as a SELECT statement, to query the database.
7. Click **Next**. Complete the **Username** and **Password** boxes, and then click **Sign in**. The data loads into a dataset that has the same name as the database to which you connected. You can now begin creating reports and dashboards by using the dataset.

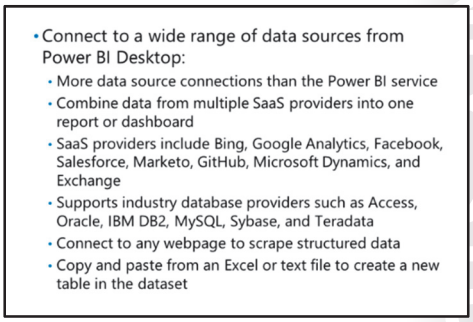
Opening in Power BI

After creating a database in Azure SQL Data Warehouse, you can use the **Open in Power BI** button to begin importing data into the Power BI service:

- In Microsoft Azure, navigate to the database in Azure SQL Data Warehouse, and then click **Open in Power BI**. This opens the connection screen in Power BI, with the **Server** and **Database** boxes prepopulated with the fully qualified server name, and the name of the database that you selected in Microsoft Azure.

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. However, 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 can then upload the dataset to work with it on the Power BI portal.



SaaS Connections

You can 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 can 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:

- Microsoft Bing®
- Google Analytics
- Intuit QuickBooks
- MailChimp
- Facebook
- Microsoft Dynamics® CRM Online
- 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

Webpage 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 may 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 you have imported the data into Power BI Desktop. To do this:

1. On the ribbon menu, click **Get Data**, and then click **Web**.
2. 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 of using a parameter value for the URL. Power BI imports the structured data, and ignores page titles and text.

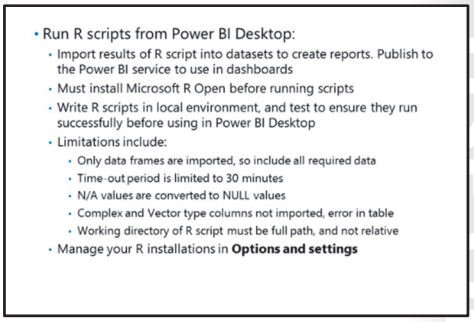
Copy and Paste

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

1. On the **Home** ribbon, click **Enter Data** to open the **Create Table** window.
2. Right-click, and then click **Paste** to copy in data from the file. The table is then 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.
3. 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 new service enables you to run R scripts in Power BI Desktop, and import the results into a Power BI Desktop data model. You can create reports by using this data, which can then be uploaded to the Power BI service.



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

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 running 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 on the **Options and settings** menu:

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: Using Databases in 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.

Demonstration Steps

Import Data from Tables in a Database in Azure SQL Database

1. On the **Home** ribbon, click **Get Data**.
2. In the **Get Data** dialog box, click **Azure**, click **Microsoft Azure SQL Database**, and then click **Connect**.
3. In the **SQL Server Database** dialog box, in the **Server** box, type the URL of the Azure server ***<Server Name>*.database.windows.net** (where <*Server Name*> is the name of the server that you created).
4. In the **Database (optional)** box, type **AdventureWorksLT**, and then click **OK**.
5. In the **Access a SQL Server Database** dialog box, click **Database**.
6. In the **Username** box, type **Student**.
7. In the **Password** box, type **Pa$$w0rd**, and then click **Connect**.
8. In the Navigator window, select the **SalesLT.Customer**, **SalesLT.SalesOrderDetail**, and **SalesLT.SalesOrderHeader** check boxes, and then click **Load**.
9. In the **Fields** pane, notice that the three tables have been added alongside the tables that were imported in the previous demonstration. When the report is published to the Power BI service, the tables are combined into a single dataset.

View Relationships Between the Tables

1. On the left of the page, click **Relationships**, and then expand the **SalesLT SalesOrderDetail**, **SalesLT SalesOrderHeader**, and **SalesLT Customer** tables to display all columns.
2. Position the cursor on the relationship arrow between **SalesLT SalesOrderDetail** and **SalesLT SalesOrderHeader**. Notice that the related columns are highlighted.
3. Position the cursor on the relationship arrow between **SalesLT SalesOrderHeader** and **SalesLT Customer**. Point out that the related columns are highlighted.
4. On the left of the page, click **Report** to return to the report canvas.
5. In the **Fields** pane, expand **SalesLT Customer**, drag the **CompanyName** field onto the canvas to create a table.
6. In the **Fields** pane, expand **SalesLT SalesOrderDetail**, drag the **LineTotal** field onto the **Customers** table on the report.
7. In the **Visualizations** pane, click **Stacked column chart**.
8. Drag the right edge of the chart to stretch it across the report and display the customers in full.
9. In the **Visualizations** pane, click **Format**, expand **Title**, and then rename the chart **Line Total by Company Name**.
10. In the **Fields** pane, under **SalesLT Customer**, drag the **CompanyName** field onto the canvas to create a table below the chart.

11. In the **Fields** pane, under **SalesLT SalesOrderDetail**, drag the **OrderQty** field onto the **CompanyName** table on the report.
12. In the **Visualizations** pane, click **Stacked column chart**.
13. Drag the right edge of the chart to stretch it across the report and display the customers in full.
14. In the **Visualizations** pane, click **Format**, expand **Title**, and then rename the chart **Order Quantity by Company Name**.
15. Expand **Data colors**, and then select a different color from the **Default color** selector, and then click on the canvas.
16. In the **Fields** pane, under **SalesLT Customer**, drag the **CompanyName** field onto **Page level filters**, and then click **Save**.
17. Leave Power BI open for the next demonstration.

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 3
The Power BI Service

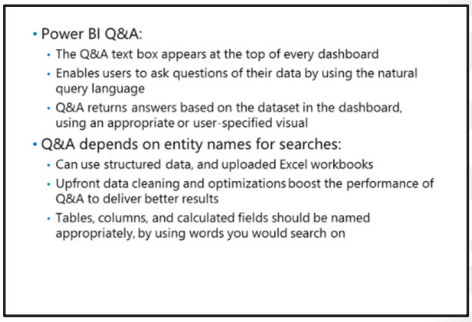
This lesson explores some of the advanced features in the Power BI service, including how to use natural query language with Power BI Q&A to ask questions of your data, create content packs to share collections of dashboards and reports, and how and why you would want to create a group.

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 and share content packs with colleagues.
- Set up groups to collaborate 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 can recognize the words in your questions, and works out where in your dataset it can find an answer. Q&A also 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.

For more information about asking questions by using Power BI Q&A, see:

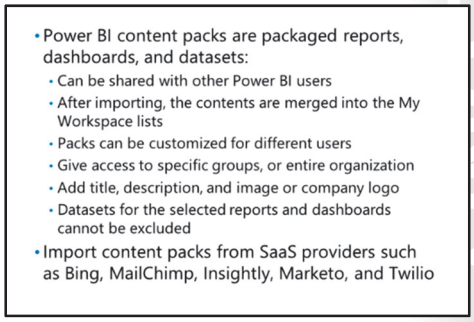
How to use Power BI Q&A

http://aka.ms/A6ziks

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 can see how easy it is to find answers to questions such as, "What were the sales last year by category and month?" Furthermore, Q&A 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 can 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 Content Packs



Content packs are packaged reports, dashboards, and datasets that can be shared with other Power BI users in your organization. When you connect to a content pack on the Power BI portal, the report items are merged into your workspace lists. Users who have a free Power BI account can view content packs, but they cannot create them. Content packs can be created 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 a content pack, 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. The content pack can be customized with a title and a description to help users to determine whether the content pack is applicable to their needs. You can also upload an image or company logo for the content pack. 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, and these cannot be excluded. The content pack is then available in your organization's content gallery. Users who have access to the content pack can create new dashboards from the contents.

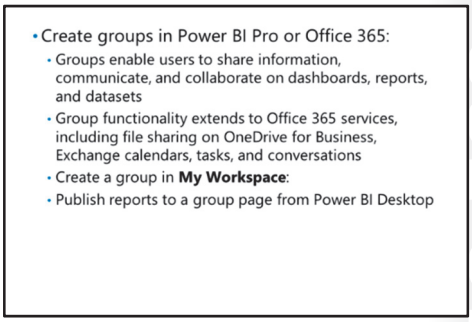
You can import content packs from SaaS providers such as Adobe Analytics, Alpine Metrics Sales Predictions, Insightly, Marketo, and Twilio. To add a content pack from an SaaS provider with whom you have an account:

1. Click **Get Data**, and then under **Services**, click **Get**.
2. Click the provider's SaaS logo, and then click **Connect**. You will be prompted to enter your login details for the service.
3. After you have been authenticated, you can import a content pack 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 content packs. You do not need a Power BI Pro account to view content packs from your organization, or from SaaS providers.

Creating a Group

You can create a group in Power BI Pro or Microsoft Office 365®. Both methods enable users to share information, communicate, and collaborate. After you create a group, you can invite members to join the group and share the group workspace. This workspace enables members to collaborate on the dashboards, reports, and datasets that are specific to the group. Group administrators can give permission to members to allow them to make modifications, or grant read-only access. The group functionality extends to Office 365 group services, including sharing files on OneDrive for Business, and Exchange calendars, tasks, and conversations.



Note: Dashboards, reports, and datasets can be shared with colleagues outside the group, which is the same process as sharing a dashboard from **My Workspace**.

Creating a Group

To create a group:

1. Sign in to the Power BI service, and then click the down arrow next to **My Workspace**.
2. In the **Name your group** box, type the name of the group. Power BI checks whether the name is available.
3. Under **Privacy**, select **Private - Only approved members can see what's inside**, or **Public - Anyone can see what's inside**.
4. Then, choose the level of access that users will have—**Members can edit Power BI content** or **Members can only view Power BI content**.
5. In the **Add group members** box, type the email address for the first colleague whom you want to invite, and then click **Add**. Repeat this step for each member, or paste a list of email addresses from Microsoft Outlook®. As each member is added, the email address is shown in a list. An invitation is emailed to each colleague and, after they have accepted, you can set their membership to **Admin** or **Member**. If a colleague has not yet created a Power BI account, the email invites them to join. You can also remove members from the group.
6. Click **Save** to finish. The group is listed under **My Workspace**, and members who have edit permission can add content to the workspace.

Editing a Group

Only group administrators can alter the group settings and change membership options. Administrators see an ellipsis next to the group name in **My Workspace**. Click the ellipsis to open the menu, and then click **Edit Group**. This opens the same form that was used to create the group, so from here, you can change the name of the group. Update the privacy settings, add or remove members, and switch membership levels between **Admin** and **Member**.

Publishing Reports to a Group

You can publish a report directly to a group page from Power BI Desktop. Create your report, and then click **Publish**. Power BI will ask if you want to publish to **My Workspace**, or any of the groups to which you belong. Select the relevant group, and then click **Publish**.

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.

Demonstration Steps

Ask a Question by Using Q&A

1. In Power BI Desktop, on the **Home** ribbon, click **Publish**.
2. In the **Sign in to Power BI** window, click **Sign in**.
3. In the **Sign in to your account** dialog box, enter the email address and password for your Microsoft account in the Sign in to your Microsoft account window, and then click **Sign in**.
4. The report will then be published to the Power BI portal.
5. In the **Publish to Power BI** dialog box, click **Open 'Adventure Works Sales.pbix' in Power BI** to view the report online.
6. When the browser opens, click **Sign in**, enter your Power BI credentials, click **Sign in**, and then wait for the report to open in Internet Explorer.
7. When the report is visible, click **Pin Live Page**.
8. In the **Pin to dashboard** dialog box, click **New dashboard**, in the box, type **Adventure Works Sales**, and then click **Pin live**.
9. If the **Introducing Featured dashboard** message appears, click **Got it**.
10. Click **Show the navigation pane**, and then under **My Workspace**, under **Dashboards**, click **Adventure Works Sales**.
11. Click in the **Ask a question about your data** box, and then point out the list of tables and fields that automatically appears.
12. Click **sales persons** from the list. Q&A returns a list of suggestions.
13. Click **by city** and the table is reordered.
14. In the **Q&A** box, type **show company name and unit price as pie chart**. When the chart is visible, in the top-right of the report, click **Pin visual**.
15. In the **Pin to dashboard** dialog box, leave **Existing dashboard** selected, and then click **Pin**.
16. In the **Q&A** box, click **unit price** to highlight it, and then scroll down the list of suggestions in the Q&A box, to show how Q&A is picking up the columns from the dataset to create suggestions.
17. In the list, click **unit price discount**. When the chart is visible, in the top-right of the report, click **Pin visual**.
18. In the **Pin to dashboard** dialog box, leave **Existing dashboard** selected, and then click **Pin**.
19. Under **My Workspace**, under **Dashboards**, click **Adventure Works Sales**. Scroll down the dashboard if necessary, to display the new charts on the dashboard.
20. Close Internet Explorer.
21. In the **Publishing to Power BI** window, click **Got it**, and then close Power BI Desktop.

Question: How could your organization use content packs and groups? What are the major advantages of content packs and groups?

Lab: Importing Data into Power BI

Scenario

Adventure Works employees are increasingly frustrated by the time that it takes to implement managed business intelligence (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.

Estimated Time: 60 minutes

Virtual machine: **10989B-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa$$w0rd**

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

1. Ensure that the 10989B-MIA-DC, 10989B-MIA-SQL, and MSL-TMG1 virtual machines are running, and then log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. Run **Setup.cmd** in the **D:\Labfiles\Lab03\Starter** folder as Administrator.
3. If you do not already have a Power BI login, browse to **https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-signing-up-for-power-bi-with-a-new-office-365-trial**, and then follow the steps to create an account.
4. Download and install Microsoft Power BI Desktop from **https://www.microsoft.com/en-us/download/details.aspx?id=45331** using the default options.

▶ Task 2: Reduce the Size of Excel Files

1. In the **D:\Labfiles\Lab03\Starter\Project** folder, double-click **Adventure Works Data.xlsx** to open the file.
2. On the **Product Category** tab, change the names of the columns to user friendly versions.
3. On the **Product Category** tab, change the cell style to normal.
4. On the **Product Category** tab, convert the cells into a table.
5. On the **Product Category** tab, change the name of the table to **ProductCategory**.
6. On the **Product Subcategory** tab, change the names of the columns to user friendly versions.
7. On the **Product Subcategory** tab, change the cell style to normal.
8. On the **Product Subcategory** tab, convert the cells into a table.
9. On the **Product Subcategory** tab, change the name of the table to **ProductSubcategory**.
10. On the **Products** tab, change the names of the columns to user friendly versions.
11. On the **Products** tab, change the cell style to normal.
12. On the **Products** tab, convert the cells into a table.
13. On the **Products** tab, change the name of the table to **Products**.
14. On the **Sales** tab, change the names of the columns.
15. On the **Sales** tab, change the cell style to normal.
16. On the **Sales** tab, convert the cells into a table.
17. On the **Sales** tab, convert the **Order Date** cells to a date type.
18. On the **Sales** tab, convert the **Unit Price**, **Unit Price Discount**, **Line Total**, and **Total Due** cells to the US dollar currency type.
19. On the **Sales** tab, change the name of the table to **Sales**.
20. Save the file.

▶ Task 3: Import Excel Files

1. In Internet Explorer, go to **https://powerbi.microsoft.com** and sign in to Power BI.
2. Upload the **Adventure Works Data.xlsx** file that you formatted in the previous task.
3. Notice that the table and column names match the names of the tabs and columns in Excel.

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

Exercise 2: Viewing Reports from Excel Files

Scenario

A business analyst has emailed an Excel workbook to you 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

1. Open the **Adventure Works Power View.xlsx** file to look at the Power View report.
2. Sign in to Power BI if you are logged out.
3. Import the Excel file into Power BI.
4. View the Power View report that has been converted to a Power BI report.
5. Test the **Sales Person** filter.

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 upon 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 content packs 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 highly visual, interactive reports and dashboards. Users can connect to a wide variety of data sources, combing data from on-premises databases, software as a service (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 life. 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 may be required, and data might 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 skills in Power BI.
- Shape data, using formatting and transformations.
- Combine data together from tables in your dataset.

Lesson 1
Power BI Desktop Queries

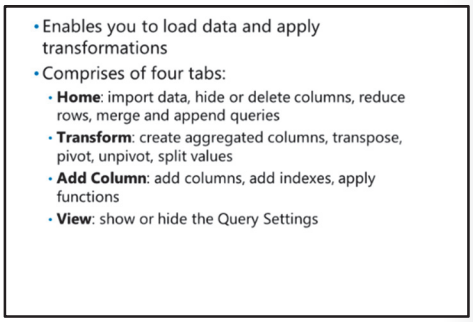
In this lesson, you will learn about the tools in Power BI that will shape and transform your data, so that it is ready to use in reports. You will also explore the main features of the Advanced Editor.

Lesson Objectives

After completing this lesson, you will be able to:

- Use the 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 Query Editor



By using the Power BI 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 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 Query Editor window, click on 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 Query Editor comprises four tabs for shaping your data:

Home

You can import data from the 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 on **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.

The **Parameters** group enables you 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. **Enter Parameters** enables you 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, as well as setting the data type and whether or not 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 **Enable refresh of this query**. 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** hides the currently selected column(s).

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 also use **Remove Duplicates** to delete rows with identical values in one or more columns. You can choose to **Remove Errors**, or **Keep Errors**.

The **Sort** group offers 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 least unique values. For example, apply the sort in order of Country, Region, 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 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 on the **Scripts** group to run R queries directly in the editor. To use this function, you must have R installed.

The rest of the items included on this tab are covered next.

Add Column

You can use the **Add 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 can use the **Add Index Column** button to create a new index column on a table. The index can be an incremental value, starting at 1, and incrementing by 1, and you can also set the starting value to be 1 or 0. With the custom index column, you can 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 wanted 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.

Using the **From Number** group you can 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, multiple, 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 can 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 Editor settings. These settings include the name of the query, or table, and the list of Applied 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 displaying the data as monospaced, and whether to show or hide white space.

Note: When you click a column in the 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 can use **Determine Data Type** to run automatic data type detection against select columns.

Applied Steps

When you shape your data using Query Editor, Power BI saves a list of the transformations you applied to your data, such as rename a column, delete a column, change a data type. This list is retained in the Applied Steps of the Query Settings. Each time you connect to the data source to run the query, 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
  - Steps can be reordered if no dependencies exist
  - You can delete steps, but be aware of dependencies
  - Steps can be undone, rolling back a previous step
  - You can rename steps

Note: When you shape the data in Power BI using the 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 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 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 pane, and choose **Move Up**, or **Move Down**, or drag the step to a new area.

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, fast method for undoing transformations such as removing a column that you later realize you have 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. It opens the list box that was used for selecting the column you wanted to hide. You can then select or unselect columns.

Inserting Steps

You can add a transformation to an existing step. If you already have a list of steps, one of which was renaming a column, followed by a number of other transformations, you can click the **Renamed Columns** step to highlight it, and then choose another column to rename. Query Editor will ask you to confirm that you want the new step to be inserted into the existing step.

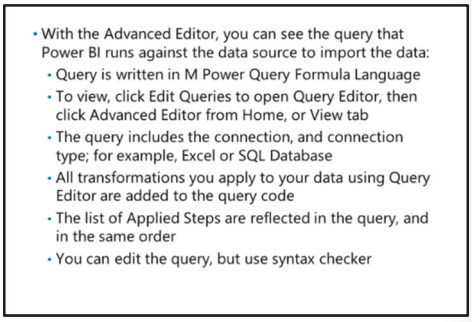
Renaming Steps

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 make adjustments to the order, or roll back a step. However, you can rename a step. In the Applied Steps pane, right-click a step, and select **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 also assists colleagues with whom you share queries, as they see which transformations have been applied.

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

You can also add a description to each of the steps. Right-click on a step and choose **Properties** to open the **Step Properties** dialog. In the **Description** box, type in 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, click **Edit Queries** from the **Home** tab. This opens the Query Editor window. From either the **Home** or **View** tabs, click **Advanced Editor**. The Advanced Editor window opens, displaying the code for the currently selected query.

The following code shows a straightforward connection to an Azure SQL Database, to query the SalesLT.SalesOrderDetail table in the AdventureWorksLT database without applying any filtering.

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 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 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.

Demonstration Steps

1. Start the MSL-TMG1, 10989B-MIA-DC and 10989B-MIA-SQL virtual machines, log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. On the taskbar, click **Power BI Desktop**.
3. In the **Power BI Desktop** window, click **Get Data**.
4. In the **Get Data** dialog box, click **Microsoft Azure SQL Database**, and then click **Connect**.
5. In the **SQL Server Database** dialog box, in the **Server** box, type the URL of the Azure server <*Server Name*>.database.windows.net (where <*Server Name*> is the name of the server you created), and in the **Database** box, type **AdventureWorksLT**, and then click **OK**.
6. In the **Access a SQL Server Database** dialog box, click **Database**.
7. In the **Username** box, type **Student**, in the **Password** box type **Pa$$w0rd**, and then click **Connect**.
8. In the **Navigator** window, select **SalesLT.SalesOrderDetail**, and then click **Edit**.
9. On the ribbon, in the **Query** group, click **Advanced Editor**. The window opens to display the query code. Note that no transformations have been applied yet, and then click **Cancel**.
10. Right-click the **SalesOrderDetailID** column, and click **Remove**.
11. In the **Applied Steps** list, right-click **Removed Columns**, click **Rename**, type **Removed SalesOrderDetailID**, and then press Enter.

12. In the center pane, right-click the **OrderQty** column, click **Rename**, type **OrderQuantity**, and then press Enter.
13. In the **Applied Steps** list, right-click **Renamed Columns**, click **Rename**, type **Renamed OrderQty**, and then press Enter.
14. In the center pane, click the **rowguid** column, and with the Ctrl key held down, click **ModifiedDate**. Right-click either column heading, and then click **Remove Columns**.
15. In the **Applied Steps** list, right-click **Removed Columns**, click **Rename**, type **Removed rowguid and ModifiedDate**, and then press Enter.
16. On the ribbon, in the **Query** group, click **Advanced Editor**. The window opens to display the query code. Note that the transformations have been added, and they are in the same order as the list of Applied Steps. Click **Cancel**.
17. In the **Applied Steps** list, right-click **Removed rowguid and ModifiedDate**, and click **Move Up**.
18. In the **Applied Steps** list, next to **Removed SalesOrderDetailID**, click the **delete** icon.
19. In the **Delete Step** dialog box, click **Delete**. The **SalesOrderDetailID** column reappears in the table.
20. In the **Applied Steps** list, next to **Navigation**, click the **gear** icon.
21. In the **Navigation** window, click **SalesLT.SalesOrderHeader**, and click **OK**. Note that the data preview has been updated with the SalesLT.SalesOrderHeader data. Also Note the warning icon under Queries[1].
22. In the **Applied Steps** list, next to **Removed rowguid and ModifiedDate**, click the **delete** icon.
23. In the **Delete Step** dialog box, click **Delete**.
24. In the **Applied Steps** list, next to **Renamed OrderQty**, click the **delete** icon.
25. Note that the warning is no longer displayed.
26. On the ribbon, in the **Query** group, click **Advanced Editor**. The window opens to display the query code. Note that the transformations have been removed, and the source table has been changed. Click **Cancel**.
27. On the ribbon, click **Close & Apply** to return to Power BI Desktop.
28. Close Power BI Desktop without saving any changes.

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.

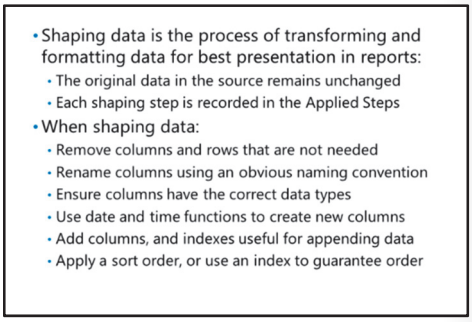
Lesson Objectives

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 Query Editor loads and presents it in the best way for your reports. The original data source remains unchanged, as 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 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 next.



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 to 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 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 a lot of 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 being asked of it.

Data Types

The 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

Datetime columns should also be formatted with the correct data type. You can 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 can 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 above, 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. Data from a date column can be merged with data from a time column to create a new datetime column.

Adding Indexes

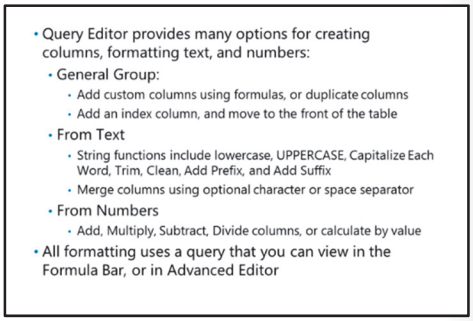
You can add indexes to your tables, with the seed value starting at 1 or 0, or you can create a custom index by defining the start number, and the increment. If you combine data from different systems, you might find that there are overlaps in the index key columns, meaning you don't have unique values when merged together. In the 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 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. Applying string functions to your text columns to create consistency ensures that data is presented well.



General Formatting

The **General Formatting** group includes the **Add Custom Column** function. You enter a custom formula to create the 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 **Add 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]**. Select a column from the **Available columns** list, and click **Insert**, or double-click to add it to the Custom column formula text box. When you are finished, click **OK**. The new column is appended to the table, and the formula is visible in the Formula Bar. If you open Advanced Editor, you see that this formula is 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 and choose **Change Type**, and then select the appropriate type. In this example, the data was converted to a Whole Number type.

The **Add Index Column** function appends the index 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, select **Add Index Column** from the **General** Group. Select **From 0**, **From 1**, or **Custom**. If you choose From 0, or From 1, the index is added immediately. If you choose Custom, this opens the **Add Index Column** dialog box. In the **Starting Index** box, type your starting number; for example, 100. Then 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, and choose **Move**, **To Beginning**. You can also select multiple rows 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 can 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 can 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, and customers have entered their names and addresses, and no formatting has been applied prior to the data being saved to the database, it is likely to be inconsistent, with mixed casing across the various fields. You can use **Capitalize Each Word** so that 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)**, give your column an appropriate name, and then 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 can 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 can 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 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 **Sum**. 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, **Multiply**. 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 **Subtract** 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.

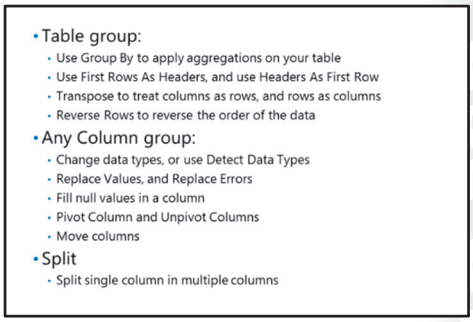
Note: The order in which you select your columns affects the calculation. For example, if you want to calculate **Profit**, click **RetailPrice**, and then click **ManufacturingCost**, and then click **Standard**, **Subtract**. The calculation is displayed in the Formula Bar, so if you have incorrectly 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, and you should be aware of the order in which you select them, as 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. Then click **Standard**, **Divide**. A new column is created and by default is named Divide. 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**. By default, this creates a new column named **Inserted Division**.

Transforming Data

While Power BI is flexible in the variety of data sources 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 may not be structurally appropriate for Power BI to translate the values in a bar chart. The 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

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

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 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. 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 are described below.

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 Column** 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 in a column, or replace 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**. Click **Advanced Options** if you want 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 select **Fill Up**, or **Fill Down**, depending on which direction you want to fill. This works at the column level.

Pivot Column and Unpivot Columns

The **Pivot Column** 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**, **To End**. The **To Beginning** option is useful if you add an index column, as this is always appended to the right.

Split 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 lesson, 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, select **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, select **Custom**, and enter the character, or symbol, such as a hyphen. You can split **At the left-most delimiter**, **At the right-most delimiter**, or **At each occurrence of the delimiter**. The number of new columns that are created will depend on which split option you choose.

To have further control over the split, click **Advanced options**. You can 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 Query Editor

In this demonstration, you will see how to:

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

Demonstration Steps

1. On the taskbar, click **Power BI Desktop**.
2. In the **Power BI Desktop** window, click **Get Data**.
3. In the **Get Data** dialog box, click **Excel**, and then click **Connect**.
4. In the **Open** dialog box, browse to the **D:\Demofiles\Mod04\Demo** folder, click **Sales Matrix.xlsx**, and then click **Open**.
5. In the **Navigator** dialog box, select the **Sales** check box, and then click **Load**.
6. When the data has finished loading, on the ribbon, click the **Edit Queries** drop-down list, and then click **Edit Queries**.
7. In the **Untitled - Query Editor** dialog box, in the **Queries** pane, click **Sales**.
8. On the **Transform** ribbon, click **Transpose**.
9. Note that the columns are now rows.
10. Click the **table** icon in the top left-hand corner of the table, and click **Use First Row As Headers**.
11. Right-click **Column1**, click **Rename**, type **Country**, and then press Enter.
12. Right-click **Column2**, click **Rename**, type **Category**, and then press Enter.
13. Click the **Country** column, and on the ribbon, in the **Any Column** group, click **Fill**, and then click **Down**. The null values are replaced.
14. Select the **2005** column, hold down the Ctrl key and select the **2006**, **2007**, and **2008** columns.
15. Right-click any of the selected column headers, and then click **Unpivot Columns**.
16. Note that the names of the columns are **Attribute** and **Value** for the attribute-value pairing.
17. Right-click the **Attribute** column, click **Rename**, type **Year**, and then press Enter.
18. Right-click the **Value** column, click **Rename**, type **Sales**, and then press Enter.
19. On the **File** menu, click **Close & Apply**.

20. In the **Fields** pane, under **Sales**, click **Country** to select the field.
21. On the **Modeling** ribbon, click **Data Category: Uncategorized**, and then click **Country/Region**.
22. In the **Fields** pane, next to **Country**, note the map icon.
23. In the **Fields** pane, under **Sales**, click **Sales** to select the field.
24. In the **Formatting** group, click **Data Type: Text**, and then click **Fixed Decimal Number**.
25. In a **Data type change** dialog box, click **Yes**.
26. Click **Format: Currency General**, point to **Currency**, and then click **$ English (United States)**.
27. In the **Fields** pane, next to **Sales**, note the sum symbol.
28. Drag the **Country** field onto the report. Note that Power BI automatically chooses the map chart.
29. Drag the **Sales** field onto the map, and note that the bubble sizes now represent the Sales figure.
30. Click the report canvas, and then in the **Visualizations** pane, click **Clustered column chart**.
31. Drag **Category** onto the **Axis** property.
32. Drag **Year** onto the **Axis** property.
33. Drag **Country** onto the **Legend** property.
34. Drag **Sales** onto the **Value** property.
35. Grab the corner edge of the chart to expand the width and height.
36. In the top right-hand corner of the chart, click **Click to Turn on Drill Down**.
37. Click the tallest column in the **Bikes** group. This now breaks down the sales by year.
38. Save the file report as **AdventureWorks Sales.pbix** to **D:\Demofiles\Mod04\Demo**.
39. Close Power BI Desktop.

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 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, and you can 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.

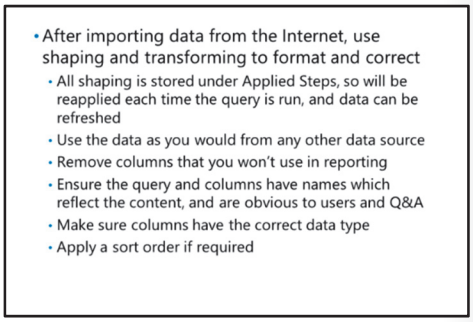
- Import data from a website that provides data in a tabular structure:
  - Use publicly available datasets, and combine this with your existing data for reporting insights
  - Import using **Get Data**, **Web**, and enter the URL
  - Power BI establishes a connection, and imports the data
  - Use the data just as you would from any other source
  - Preview the table structures that Power BI has detected
  - Load data, or edit in 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, and 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 in 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 can select the tables and preview them as you would any other data source. Click **Edit** to load the data into 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



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. 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 may share the query.

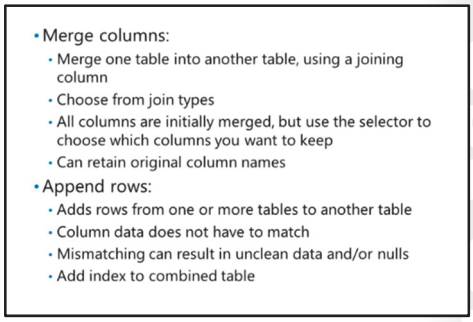
It is also important to ensure the name of the query (table) is something obvious, and the same applies to the column's name. Again, this keeps clarity within the dataset, and has the added advantage that you or other colleagues can understand the data just by looking at the names of the queries and 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 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 drill-down. The Query Editor does not always recognize currencies unless there is a symbol included in the data, so you should update any currency columns. Check that numerical columns have the correct data types, and also 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 the 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. The data can then be combined 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 with data from Excel. After importing into a single set of data, you can merge columns using tables from different sources, and append rows.



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 Query Editor window. Click the query (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 into. Click to choose the column you want to join on. You can select more than one column by holding down the Ctrl key while using your mouse to select. In the list, select the table you want to merge from. 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 can 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 may 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 there are no values when the number of columns between the tables does not match. From Power BI Desktop designer, click **Edit Queries** to open the 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 have 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.

Demonstration Steps

1. On the taskbar, click **Power BI Desktop**.
2. In the **Power BI Desktop** window, click **Get Data**.
3. In the **Get Data** dialog box, click **Web**, and then click **Connect**.
4. In the **From Web** dialog box, in the **URL** box, type **http://www.imdb.com/chart/top**, and then click **OK**.
5. In the **Navigator** window, select the **Table 0** check box, and then click **Edit**.
6. In the **Untitled - Query Editor** window, right-click the left-most empty column, and click **Remove**.
7. Right-click the right-most empty column, and click **Remove**.
8. Right-click the **Your Rating** column, and click **Remove**.
9. Note that these steps have been grouped together in the **Applied Steps** list as **Removed Columns**.
10. Click the **Rank & Title** column, and then on the **Home** tab, in the **Transform** group, click **Split Column**, and then click **By Delimiter**.
11. In the **Split Column by Delimiter** dialog box, in the **Select or enter delimiter** list, click **--Custom--**, and then type a period (**.**) in the box.
12. Under **Split**, click **At the left-most delimiter**, and then click **OK**.
13. The Rank data now shows in its own column. Right-click the **Rank & Title.1** column, click **Rename**, type **Rank**, and then press Enter.
14. Click the **Rank & Title.2** column, and on the **Transform** ribbon, in the **Any Column** group, click **Replace Values**.
15. In the **Replace Values** dialog box, in the **Value to Find** box, type **(**, and then click **OK**.
16. With focus on the **Rank & Title.2** column, from the **Any Column** group, click **Replace Values**.
17. In the **Replace Values** dialog box, in the **Value to Find** box, type **)**, and then click **OK**.
18. With focus on the **Rank & Title.2** column, in the **Text Column** group, click **Split Column**, and then click **By Number of Characters**.
19. In the **Split Column by Number of Characters** dialog box, in the **Number of characters** box, type **4**.
20. Under **Split**, click **Once, as far right as possible**, and then click **OK**.
21. The Year data has been moved to a separate column.

22. Right-click the **Rank & Title.2.1** column, click **Rename**, type **Title**, and then press Enter.
23. In the **Text Column** group, click **Format**, and then click **Trim**. The white space around the titles is removed.
24. Right-click the **Rank & Title.2.2** column, click **Rename**, type **Year**, and then press Enter.
25. In the **Query Settings** pane, under **Properties**, in the **Name** box, type **IMDB Top 250 Movies**, and then press Enter.
26. On the **File** menu, click **Close & Apply**.
27. In Power BI Desktop, on the **File** menu, click **Exit**. If prompted to save your changes as **IMDB Top 250 Movies**, and then click **Save**.

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; and 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.

Estimated Time: 60 minutes

Virtual machine: **10989B-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa$$w0rd**

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 been given 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

1. Ensure that the MSL-TMG1, 10989B-MIA-DC, and 10989B-MIA-SQL virtual machines are running, and then log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. Run Setup.cmd in the **D:\Labfiles\Lab04\Starter** folder as Administrator.
3. If a message asks **Do you want to continue with this operation?**, type **Y** and press Enter.
4. If you do not have a Power BI login, go to **https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-signing-up-for-power-bi-with-a-new-office-365-trial**, and follow the steps to create an account.
5. Download and install the **Microsoft Power BI Desktop** from https://www.microsoft.com/en-us/download/details.aspx?id=45331.

▶ Task 2: Import Data from Excel

1. In **Power BI Desktop**, open the **Sales - Europe.xlsx** from the **D:\Labfiles\Lab04\Starter** folder, and load the **Europe** data.
2. Open **Sales - North America.xlsx**, from the **D:\Labfiles\Lab04\Starter** folder, and edit the **North America** data.
3. Leave the Query Editor window open for the next exercise.

▶ Task 3: Apply Formatting to the Existing Data

1. Select **Europe** in the Queries pane, and remove the following columns:
   - **ProductKey**
   - **SalesOrderNumber**
2. Rename the following columns:
   - **SalesTerritoryCountry** to **Country**
   - **SalesTerritoryGroup** to **Sales Territory**
   - **EnglishProductCategoryName** to **Main Category**
   - **EnglishProductSubCategoryName** to **Sub Category**
   - **EnglishProductName** to **Product**
3. Move the **Color** column to the left.
4. Select **North America** in the Queries pane, and remove the following columns:
   - **ProductKey**
   - **SalesOrderNumber**
5. Rename the following columns:
   - **SalesTerritoryCountry** to **Country**
   - **SalesTerritoryGroup** to **Sales Territory**
   - **EnglishProductCategoryName** to **Main Category**
   - **EnglishProductSubCategoryName** to **Sub Category**
   - **EnglishProductName** to **Product**
6. Move the **Color** column to the left.
7. View the query that has been updated with the applied steps.
8. Leave Query Editor open for the next exercise.

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

1. Select **Europe** in the Queries pane.
2. Click **Append Queries**, and combine the North America data with the Europe data.
3. Use the selection menu on the Country column header to check that the data has loaded in.
4. Open the **Country Codes.xlsx** file in the D:\Labfiles\Lab04\Starter folder, and copy the data.
5. In Power BI Desktop, click **Enter Data**, and paste in the copied data.
6. Name the table **Country Codes**.
7. Select **Europe** in the Queries pane.
8. Merge the **Country Codes** table with the **Europe** table.
9. Exclude the **Territory**, and **Country** columns, and clear the **Use original column name as prefix** setting.
10. Move the **Code** column to the beginning of the table.
11. Rename the **Code** column as **Country Code**.
12. Apply the changes.

Results: At the end of this lab, 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 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

With its ability to quickly create visually stunning, interactive reports and dashboards, Microsoft® Power BI is making its mark in the self-service BI world. Much of the appeal of Power BI is how straightforward it is 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 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 quickly add calculated columns, tables, and measures, you will see how Power BI creates attractive reports, while helping you find hidden insights into your 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.

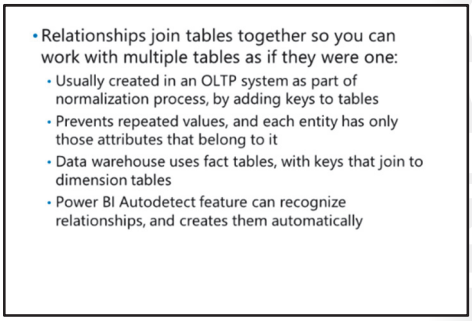
Lesson Objectives

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 Query Editor.
- Create relationships using the 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), then you will understand the concept of relationships in Power BI, as this is much the same.



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, post code and so on, each time a customer places an order. 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 is connected 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. In this case, the SalesOrderID column values 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 below. There is one row in the SalesOrderHeader table for each order, though this order may comprise multiple rows in the SalesOrderDetail table. Also, 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

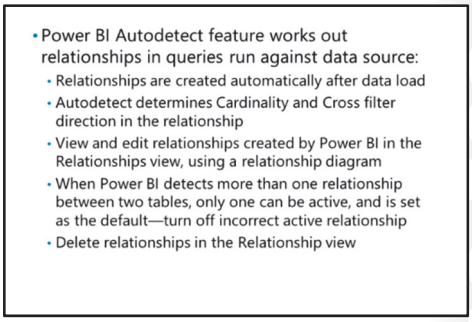
Traversing the tables using the relationships, the SalesOrderDetail table is related to the SalesOrderHeader table, and the Product table. The SalesOrderHeader table is related to the SalesOrderDetail table, and also the Customers table. These relationships mean you can view the four tables as one, so you can see all the products ordered by a customer as if it were 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 topics are covered 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 case, you may not have to do any further work to establish relationships between the tables.

Viewing Relationships

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. As these queries are running, Power BI observes them to determine if there are relationships between the tables. After the data has finished loading, you can view and manage the relationships that Power BI has created for you.



Viewing Relationships

Power BI Desktop comprises three main views: Report, Data, and Relationship. 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 also view the values in the tables. The Relationship View shows the tables and columns, linking together those tables that are related. Click **Relationships** in the Views pane to open a diagrammatic view of the relationships in your model. The relationships appear the same in the Relationship 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, which joins the two tables together. 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, is another icon representing 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 on a relationship line, the related columns in either table are highlighted with a black border, for quick identification.

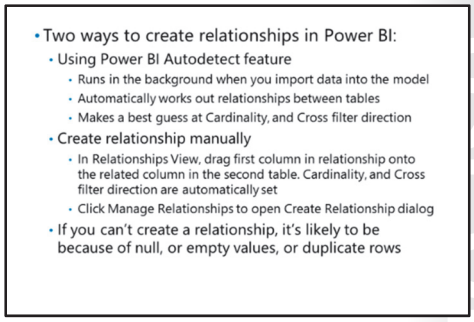
Editing Relationships

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 can create new relationships, run the Autodetect feature, edit, and also 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 on 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 in yellow. Right-click the relationship line, and select **Delete**.

Creating Relationships

There are two ways to create relationships in Power BI. You can use the Autodetect feature and Power BI works out the relationships for you, or you can create them manually.



Creating Relationships Using Autodetect

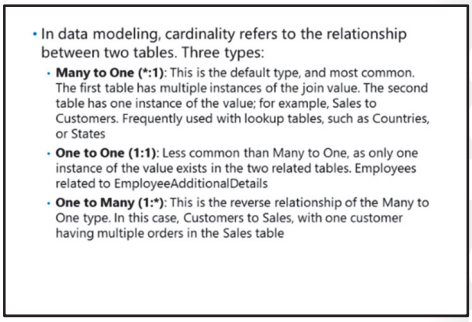
When data is imported into the model, Power BI automatically creates relationships. If you then create calculated tables, or use Enter Data to add new tables, relationships will not exist. You can run the Autodetect feature from the **Home** tab. In the **Relationships** group, click **Manage Relationships**, then click **Autodetect** in the **Manage Relationships** dialog box. 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 may 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 you want to join to. 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 from the 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 you wish to join to. 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, then click **OK**. Click **Close** to hide the **Manage Relationships** dialog box.

You may find that you cannot create a relationship between your tables. This can be due to columns with null, or empty values, or duplicate data. You can 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 can 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

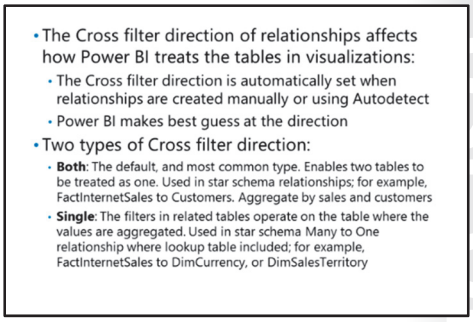


In data modeling, Cardinality refers to the relationship that one table has with another. In Power BI modeling, the Cardinality can be one of the following three types:

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, and one of its uses can be to split up larger tables. For example, if you 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 frequently used 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 can 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 Relationships View, double-click on 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 affect 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:

- **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, in the relationship diagram in the Relationship View, the FactInternetSales table is surrounded by the related lookup tables, such as DimCustomer, DimCurrency, DimDate, DimProduct, DimPromotion, and DimSalesTerritory. Indeed, the layout of the tables in the Relationships View may reflect 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 (nonlookup) table, you may want to set the Cross filter direction to Single. For example, if you have two unrelated tables with values for aggregating, 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 can 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**. Or, in the Relationships View, double-click on 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 double arrow for Both on the relationship line. The Single arrow points in the direction of the filter.

Demonstration: Viewing 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.

Demonstration Steps

1. Run **D:\Demofiles\Mod05\Setup.cmd** as an Administrator, and, in the **User Account Control** dialog box, click **Yes**.
2. When prompted press **Y**, and when the script completes, press any key to close the window.
3. On the desktop, double-click **Power BI Desktop**.
4. On the **Power BI Desktop** page, click **Get Data**.
5. In the **Get Data** dialog box, click **Get Data**, click **Excel**, and click **Connect**.
6. In the **Open** dialog box, navigate to **D:\Demofiles\Mod05\Demo**, click **Adventure Works Sales Data.xlsx**, and then click **Open**.
7. In the **Navigator** dialog box, select the following check boxes, and then click **Load**:
    a. **DimCurrency**
    b. **DimCustomer**
    c. **DimDate**
    d. **DimProduct**
    e. **DimPromotion**
    f. **DimSalesTerritory**
    g. **FactInternetSales**.
8. In the Views pane on the left-hand side, click **Relationships**.
9. Point out that Power BI has created the relationships automatically. The layout represents a star schema.
10. Maximize the tables in the relationship diagram to display all columns.
11. Point out that Power BI has not created a relationship to **DimDate** from **FactInternetSales**.
12. On the **Home** tab, click **Manage Relationships**.
13. In the **Manage Relationships** dialog box, click **New**.
14. In the **Create Relationship** dialog box, in the top table list, click **FactInternetSales**. When the table preview appears below, click the **OrderDateKey** column.
15. In the bottom table list, click **DimDate**. When the table preview appears below, click the **DateKey** column.
16. Check that the **Cardinality** is selected to **Many to One (*:1)**, the **Cross filter direction** is **Single**, and **Make this relationship active** is selected, and then click **OK**.
17. In the **Manage Relationships** dialog box, click **Close**.

18. In the diagram, in the **FactInternetSales** table, click the **DueDateKey** column. Drag the **DueDateKey** column to the **DateKey** column in the **DimDate** table. Point out the dotted line to show that the relationship is inactive. This is because there is more than one related column in the two tables.
19. In the diagram, in the **FactInternetSales** table, click the **ShipDateKey** column. Drag the **ShipDateKey** column to the **DateKey** column of the **DimDate** table. Point out the dotted line to show that the relationship is inactive.
20. Point out that the relationships from FactInternetSales to DimCurrency, DimProduct, DimPromotion, and DimSalesTerritory, have a Cross filter direction of Both, indicated by the double arrow icon. These are lookup tables, so should be Single.
21. On the **Home** tab, click **Manage Relationships**.
22. In the **Manage Relationships** dialog box, double-click the **FactInternetSales (CurrencyKey)** relationship.
23. In the **Edit Relationship** dialog box, in the **Cross filter direction** list, click **Single**, and then click **OK**.
24. In the **Manage Relationships** dialog box, double-click the **FactInternetSales (ProductKey)** relationship.
25. In the **Edit Relationship** dialog box, in the **Cross filter direction** list, click **Single**, and then click **OK**.
26. In the **Manage Relationships** dialog box, double-click the **FactInternetSales (PromotionKey)** relationship.
27. In the **Edit Relationship** dialog box, in the **Cross filter direction** list, click **Single**, and then click **OK**.
28. In the **Manage Relationships** dialog box, double-click the **FactInternetSales (SalesTerritoryKey)** relationship.
29. In the **Edit Relationship** dialog box, in the **Cross filter direction** list, click **Single**, and then click **OK**.
30. In the **Manage Relationships** dialog box, click **Close**.
31. Click the relationship line between **FactInternetSales** and **DimCustomer**. Point out that this is a One to One relationship because the FactInternetSales table only contains an extract. Normally this would be Many to One. This must be changed so it is ready for the rest of the data to be loaded at a later date.
32. Click the relationship line between **FactInternetSales** and **DimCustomer**, and then press Delete.
33. In the **Delete Relationship** dialog box, click **Delete**.
34. On the **Home** tab, click **Manage Relationships**.
35. In the **Manage Relationships** dialog box, click **New**.
36. In the **Create Relationship** dialog box, in the top table list, click **FactInternetSales**, and in the data preview, click the **CustomerKey** column.
37. In the bottom table list, click **DimCustomer**, and in the data preview, click **CustomerKey**.
38. In the **Cardinality** list, click **Many to One (*:1)**, and then click **OK**.
39. In the **Manage Relationships** dialog box, click **Close**. In the diagram, point out that the 1 icon next to FactInternetSales is now a star icon.
40. Click **Save**, and save the file to the **D:\Demofiles\Mod05\Demo** folder as **Adventure Works Sales.pbix**.
41. Leave Power BI Desktop open for the next demonstration.

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.

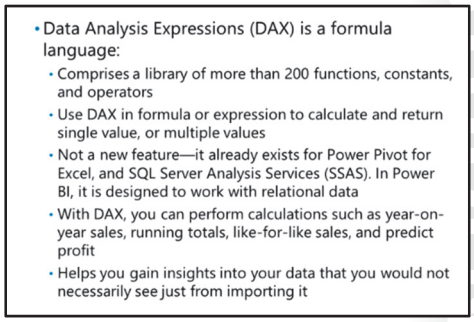
Lesson Objectives

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

- Describe DAX, and what it is used for.
- Understand the DAX syntax so that 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 specifically designed to work with relational data, which is what you use 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.



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 may 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:

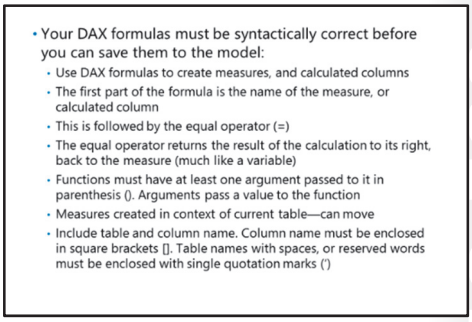
Calculate Sales for the Same Time Period Last Year

```
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 can 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:



The following DAX formula adds together the values in the LineTotal column of the InternetSales table, and returns a measure named Total Sales:

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 then 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 your 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 to move the measure to. If you create the above example in the InternetSales table, you can move it elsewhere without the formula being affected. Because you passed the table and column name as the argument, this creates independence, as 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 can exclude the table name when the measure refers to a column in the same table in which it also resides; however, it is 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 your 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.

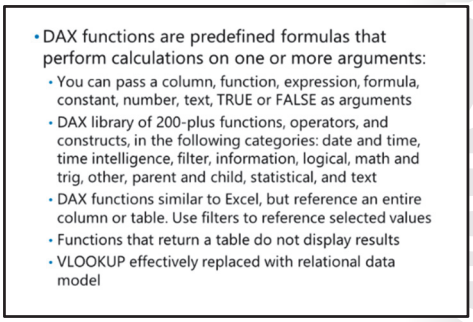
If you are already familiar with Power BI, you may know that numerical fields are automatically calculated, and wonder why you would want to create the above measure, as Power BI will sum this for you. By adding this measure, you can use it as an argument for another formula, meaning you can create all the calculations you require within your dataset. For more information on DAX syntax, see:

DAX Syntax Reference

http://aka.ms/tl7369

Functions

Functions are predefined formulas that perform calculations on one or more arguments. As you learned in the previous lesson, you can pass a column as an argument, and you can also use 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:



- **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 can 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 can 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 then. 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

http://aka.ms/lrf8p9

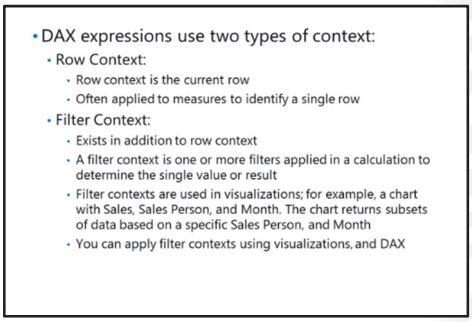
If you have been using Excel functions, DAX functions may look familiar. However, DAX functions differ in the following ways:

- DAX functions reference an entire column, or a table. To use selected values from a table or column, you can include filters in your formula.
- If you want to customize a calculation to work on a row-by-row basis, use functions to use 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, but rather it is 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 can 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 as you can create relationships, and may 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:

- **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 to be 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 can use a filter context to reduce the values that are included in a calculation. The filter can specify 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 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: Row and Filter Context in DAX Formulas

In this demonstration, you will see how row and filter context works with measures.

Demonstration Steps

1. In Power BI Desktop, on the left side of the window, click **Report**.
2. In the **Fields** pane, click **FactInternetSales**.
3. On the **Modeling** tab, in the **Calculations** group, click **New Measure**.
4. In the formula bar, highlight **Measure =**, type the following script, and then press Enter:

```
TotalSales = SUM(FactInternetSales[SalesAmount])
```

5. In the **Fields** pane, click **FactInternetSales**.
6. Click **New Column**.
7. In the formula bar, highlight **Column =**, type the following script, and then press Enter:

```
European Sales = CALCULATE(FactInternetSales[TotalSales],
DimSalesTerritory[SalesTerritoryGroup] = "Europe")
```

8. Point out that the TotalSales measure has been used in the formula.
9. In the **Fields** pane, select the **European Sales** check box to add it to the report.
10. In the **Visualizations** pane, click **Gauge**.
11. Click **Format**, and expand **Gauge axis**.

12. In the **Max** box, type **1000000**.
13. In the **Target** box, type **1000000**.
14. Leave Power BI open for the next demonstration.

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.

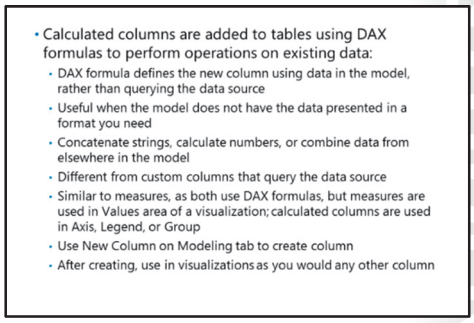
Lesson Objectives

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 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. Measures are generally 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 feature on the **Modeling** tab to create a calculated column, or right-click on 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.

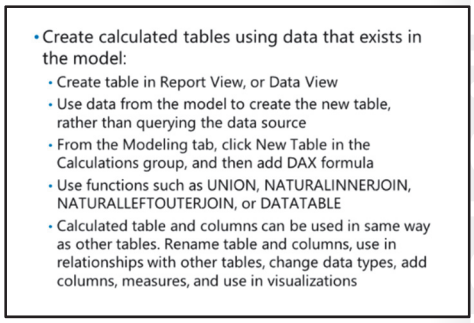
The code example below returns the related Region value for the City column in the Customers table.

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 can be 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, and you use a DAX formula to define the values in the table. Tables can be 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.

To create a calculated table, open Power BI Desktop and the report with the dataset you want to add the table to. Click **Modeling**, then click **New Table** from the **Calculations** group. 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 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 example above, 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 can 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 Sales table, to the columns from the Customers table:

Create Calculated Table Using the NATURALINNERJOIN function

```
Customer Sales = NATURALINNERJOIN (Customers, Sales)
```

In the above example, only rows with matching values in both tables will be 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.

Furthermore, you can 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. From the **Modeling** tab, click **New Table** in the **Calculations** group. In the formula bar, type your DAX formula using DATATABLE to define the columns, types, and values. The following code creates a Countries table, and adds values to the table.

The code example below creates a new table using the DATATABLE function. Use it to define the column names and data types, and enter values into the table:

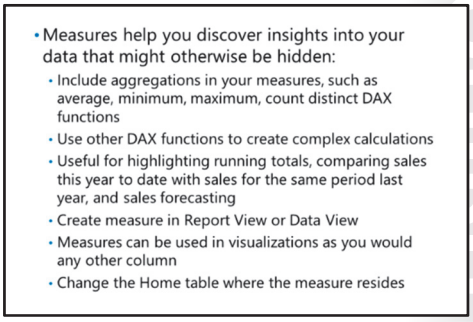
Use the DATATABLE Function to Create a New Table

```
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 can use it in exactly the same way as any other table that exists in the model, including using it in relationships. You can 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 using 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 are created using DAX formulas, and with an extensive library of functions, operators, and constructs, there is much 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 on the table in the Fields pane, and select **New Measure**. Or from the **Modeling** tab, select **New Measure** from the **Calculations** group. It is generally easier to work in the Data View, as you can 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 can 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 on the last year:

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, select **Home Table** from the **Properties** group, 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.

Demonstration Steps

1. In Power BI Desktop, on the left side of the window, click **Data**.
2. In the **Fields** pane, click **DimCustomer** to select the table, and preview the data.
3. Right-click **DimCustomer**, and click **New column**.
4. In the formula bar, highlight **Column =**, type the following script, and then press Enter:

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

5. If the new column is not visible, scroll to the right of the table. Note the new **FullName** column in the table.
6. In the **Fields** pane, point out the icon next to the new column, which indicates that this has been created using a DAX formula.
7. In the **Fields** pane, right-click **DimCustomer**, and then click **New column**.
8. In the formula bar, highlight **Column =**, and type the following script, and then press Enter:

```
MaleFemale = IF([Gender] = "M", "Male", "Female")
```

9. Note the new column at the end of the table.
10. On the **Modeling** tab, in the **Calculations** group, click **New Column**.
11. In the formula bar, highlight **Column =**, and type the following script, and then press Enter:

```
Relationship = IF([MaritalStatus] = "M", "Married", "Single")
```

12. Note the new column at the end of the table.
13. On the **Modeling** tab, in the **Calculations** group, click **New Table**.
14. In the formula bar, highlight **Table =**, and type the following script, and then press Enter:

```
DimCountry = DATATABLE (
"Country", STRING, "Code", STRING,
{
{"United States", "US"},
{"United Kingdom", "UK"},
{"France", "FR"},
{"Germany", "DE"},
{"Spain", "ES"}
}
)
```

15. In the **Fields** pane, note the new table.
16. On the **Modeling** tab, in the **Calculations** group, click **New Measure**.
17. In the formula bar, highlight **Measure =**, and type the following script, and then press Enter:

```
MostRecentOrder = MAX(FactInternetSales[OrderDateKey])
```

18. In the **Fields** pane, note the icon next to the measure, to indicate this is a calculated field.
19. In the **Fields** pane, click the **MostRecentOrder** field.
20. On the **Modeling** tab, in the **Properties** group, click **Home Table**, and click **FactInternetSales**. This moves the measure so that it resides in the FactInternetSales table. Note that the **MostRecentOrder** measure now appears under FactInternetSales in the Fields pane.
21. Close Power BI Desktop, saving your work.

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; 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 the relationships that have been created automatically in your data.
- Create relationships between the tables in your dataset.
- Add a calculated column to a table.

Estimated Time: 60 minutes

Virtual machine: **10989B-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa$$w0rd**

Exercise 1: Create Relationships

Scenario

The data in your organization is spread 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 may not detect all 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

1. Ensure that the 10989B-MIA-DC and 10989B-MIA-SQL virtual machines are both running, and then log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. Run **Setup.cmd** in the **D:\Labfiles\Lab05\Starter** folder as Administrator.
3. If you do not already have a Power BI login, go to **https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-signing-up-for-power-bi-with-a-new-office-365-trial**, and follow the steps to create an account.
4. Download and install the Microsoft Power BI Desktop from **https://www.microsoft.com/en-us/download/details.aspx?id=45331**.

▶ Task 2: Automatic Relationships

1. Open Power BI Desktop from the taskbar.
2. From Get Data, connect to **Adventure Works Sales Data.xlsx** in the D:\Labfiles\Lab05\Starter\Project folder.
3. Import the **DimCurrency, DimCustomer, DimDate, DimProduct, DimPromotion, DimSalesTerritory**, and **FactInternetSales** worksheets.
4. Click the **Relationships** view.
5. Create a new relationship between the **FactInternetSales** table **OrderDateKey** column, and the **DateKey** column in **DimDate**. Set the Cardinality to **Many to One (*:1)**, the Cross filter direction to **Single**, and make this relationship **active**.
6. Create a new relationship between the **FactInternetSales** table **DueDateKey** column, and the **DateKey** column in **DimDate**. Set the Cardinality to **Many to One (*:1)**, and the Cross filter direction to **Single**.
7. In the Relationships view, drag the **ShipDateKey** column in the **FactInternetSales** table to the **DateKey** column of the **DimDate** column to create a new relationship.
8. Use Manage Relationships to change the Cross filter direction of the relationship between **FactInternetSales** and **DimCurrency** to **Single**.
9. Use Manage Relationships to change the Cross filter direction of the relationship between **FactInternetSales** and **DimProduct** to **Single**.
10. Use Manage Relationships to change the Cross filter direction of the relationship between **FactInternetSales** and **DimPromotion** to **Single**.
11. Use Manage Relationships to change the Cross filter direction of the relationship between **FactInternetSales** and **DimSalesTerritory** to **Single**.
12. Change the relationship between **FactInternetSales** and **DimCustomer** so this is **Many to One (*:1)** from **FactInternetSales**. Set the **Cross filter direction** to **Both**.
13. Save the file to the **D:\Labfiles\Lab05\Starter** folder and name it **Adventure Works Sales.pbix**.
14. Leave Power BI Desktop open for the next exercise.

▶ Task 3: Manual Relationships

1. Open the **Adventure Works Product Categories.xlsx** file, located in the **D:\Labfiles\Lab05\Starter\Project** folder.
2. Add **DimProductCategory**, and **DimProductSubcategory** to the dataset.
3. Delete the relationship between **DimProductCategory**, and **DimProductSubcategory**.
4. Create a new **One to Many (1:*)** relationship between **DimProductCategory**, and **DimProductSubcategory**, by dragging the **CategoryKey** from **DimProductCategory**, to **CategoryKey** on **DimProductCategory**. The Cross filter direction should be **Both**.
5. Drag the **ProductSubcategoryKey** column in the **DimProduct** table, to the **SubcategoryKey** column in the **DimProductSubcategory** table, to create a **Many to One (*:1)** relationship, and a Cross filter direction of **Both**.
6. Save the file.
7. Leave Power BI Desktop open for the next exercise.

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

You have created the required relationships in your dataset, but feel that you could 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

1. Click **Data** in the Views pane in Power BI Desktop.
2. Add a calculated column named IncomeStatus to the **DimCustomer** table, based on the **YearlyIncome** column. Put the income into income brackets.
3. Add a calculated column named **DaysSinceFirstPurchase** to the **DimCustomer** table, to show the number of days since the customer made their first purchase.
4. Add a calculated column to **DimCustomer**, which concatenates the **FirstName** and **LastName** columns into a column named **FullName**.
5. Add a calculated column to **DimCustomer**, called **MaleFemale**, which converts the value of the **Gender** column to Male, or Female.
6. Add a calculated column to **DimCustomer**, called **Relationship**, which converts the value of the **MaritalStatus** column to Married, or Single.
7. Add a calculated column called **MainCategory** to the **DimProductSubcategory** table, which uses the RELATED function to return the name of the category from **DimProductCategory**.
8. Add a calculated column called **PromotionLengthDays** to the **DimPromotion** table to show how many days the promotion lasted. This is the difference between **StartDate** and **EndDate**.
9. Add a calculated column called **Profit** to **FactInternetSales**. Show the difference between **UnitPrice** and **ProductStandardCost**, formatted as currency.

Results: At the end of this lesson, 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 lesson of the Dax Queries topic 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

With its ability to quickly create visually stunning, interactive reports and dashboards, Microsoft® Power BI is making its mark in the self-service BI world. Much of the appeal of Power BI is how straightforward it is 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 quickly add calculated columns, tables, and measures, 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. The Microsoft Power BI suite of tools enables business users to connect to a wide variety of data sources, including the main industry-standard databases, Microsoft cloud-based services—including 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 quickly create visually stunning, interactive dashboards makes Power BI an obvious choice for any organization that needs to give its users 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 your 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:

- 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.
- Add custom visualizations to your reports.

Creating Charts

By using the chart visuals in Power BI Desktop, you can quickly create visually stunning, interactive reports and dashboards.

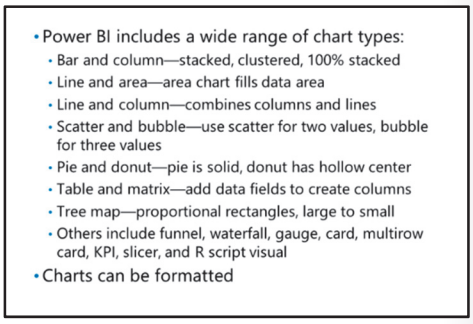


Chart Types

Included in Power BI are 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 automatically create a table visual, which 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. Then, you can 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 value, 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 can also 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 similar to 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 again similar to 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 be combined 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 a period of 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. The stacked area chart enables you to compare multiple values, so using the above example, you could add **Share Price** and **Net Sales** to show the profit that is made 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**.

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 nonnumeric 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 proportionally represent the data. A bubble chart is created by using a scatter chart, and then adding a data field to the **Size** property.

 **Note:** All of the chart types that are listed 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. The **Arrange** property enables you 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.

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.

Table and Matrix Charts

You use the table and matrix charts 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.

Tree Map

The Tree map may 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 located 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.

Other Charts

There are other types of chart in Power BI, including waterfall, funnel, gauge, card, multirow card, KPI, slicer, and R script visual. For more information about using these charts, including how-to guides and tips, see the following article:

Visualizations in Power BI

http://aka.ms/Cfrub0

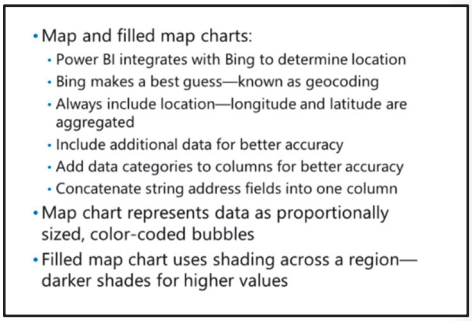
Formatting Charts

Each chart includes options for formatting. The options that are available 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 on a **Salesperson**, all charts reflecting their data show as the same color. Formatting enables you 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 a map chart and a filled map chart. These charts enable you to visually map your data 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 geocoding. This integration means that you do not need to provide longitude and latitude coordinates in your data, although this is optional because Bing makes 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. This helps Bing to work out whether you are referring to Washington State, or Washington DC. You can also append additional information, so if your data refers to Washington in England, you can pass "Washington, England" to Bing. If you 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, on the **Properties** menu, 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 may find that even using data categorization does not generate the desired locations in Bing. When this problem arises, 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 menu, click **Add 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 may 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

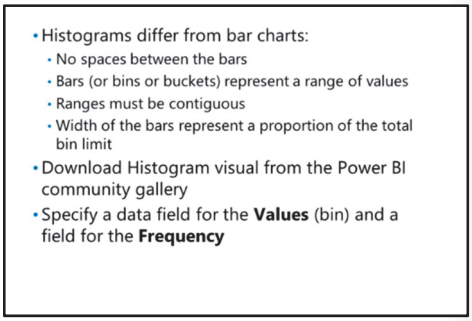
The map chart accepts data for the **Legend**, **Longitude**, **Latitude**, **Values**, and **Color Saturation** properties. The **Legend** property accepts fields such as **City**, **County**, and **Province**, whereas the **Values** property accepts numeric values such as **Total Sales** or **Number of Customers**. The numeric values are presented as colored bubbles on the applicable location that is specified in the **Legend** property. 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.

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, with smaller values represented by lighter shades. This 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. To use this example, you would add **States** to the **Location** property of the filled map chart, and your numerical data to the **Values** property.

 **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.

Histograms

Histograms may initially look very similar to bar charts, but there are two fundamental differences:



- **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.

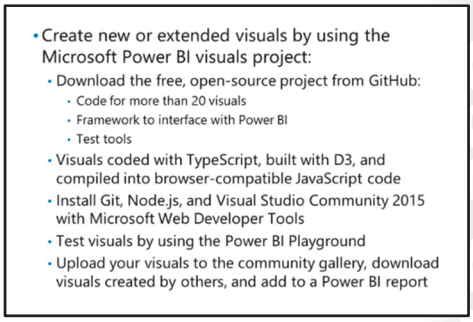
Power BI does not include a histogram chart by default; however, you can download a custom visual from the community gallery. To do this:

1. Visit the Power BI community gallery, and then click **Histogram**.
2. Check that this is the visual that Microsoft has developed, and then in the **Description** dialog box, click **Download Visual**.
3. Read the terms and, if you agree with them, click **I agree** in the **Licensing** dialog box. The visual is saved to the default download folder that is specified in your browser's settings.
4. Open Power BI Desktop.
5. In the Visualizations pane, click the **Import from a file** icon (...).
6. In the Caution window, click **Import**.
7. Browse to the download folder, click the visualization (histogram.pbiviz) file, and then click **Open**.
8. When the **Success confirmation** dialog box appears, click **OK**. The histogram icon now appears in the Visualizations pane and is ready to use.

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 community gallery. Custom visuals are discussed in the next topic.

Adding a Custom Visualization



You can add custom visuals to reports, dashboards, and content packs. You can extend the capabilities of the visuals in Power BI, or create new ones, by downloading the Microsoft Power BI visuals project from GitHub. Microsoft has provided an open-source project that consists of visualization code, tooling, and a test suite, which is available on GitHub. The project has more than 20 types of visualization, the framework that is 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. Furthermore, 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 quickly build your own custom visuals.

Creating Custom Visuals

To build the library in the Microsoft Power BI visuals project, and run the sample application, you need to install Git and Node.js. Microsoft recommends using Visual Studio® Community 2015 for your IDE, making sure that you install the optional Microsoft Web Developer Tools. This also enables you to install the VSIX Package add-in, and use the Visual Studio Template to create new visuals.

You can use the Power BI Playground on GitHub to test how visuals look by using the Web view or Mobile view, with the Mobile view offering Dashboard and In-focus views. The playground offers a list of visuals to browse through, and see how they appear. You can view animations and interactions, in addition to resizing the tile. A playground app is also included in the Microsoft Power BI visuals project.

For more information about creating custom visuals, ideas, and support, see the Developers category of the Power BI blog.

Microsoft Power BI Blog

http://aka.ms/Yta0x4

Adding Custom Visuals to Reports

You can only import custom visuals into one report at a time. Either create your own visual and save it to a local drive, or download a visual from the community gallery. To download a custom visual:

1. Visit the community gallery, and then click the visual that you want.
2. In the **Description** dialog box, click **Download Visual**.
3. Read the terms, and then if you agree, click I **agree** in the **Licensing** dialog box. The visual is saved to the default download folder that is specified in your browser's settings.
4. Open Power BI Desktop.
5. In the Visualizations pane, click the **Import from a file** icon (...).
6. In the Caution window, click **Import**.
7. Browse to the download folder, select the visualization (.pbiviz) file, and then click **Open**.

8. When the **Success confirmation** dialog box appears, click **OK**. A new icon appears in the Visualizations pane, and is ready to use.

Click the icon, and the visual appears on your report as a watermark template. Drag fields onto the pane to add data.

To upload a custom visual that you have created, or download a custom visual from the Power BI community gallery, see the following article:

Welcome to Power BI custom visuals

http://aka.ms/Y527x6

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.

Demonstration Steps

1. If you do not have a Power BI login, open **Internet Explorer**, go to **https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-signing-up-for-power-bi-with-a-new-office-365-trial**, and follow the steps to create an account.
2. In Internet Explorer®, go to **https://www.microsoft.com/en-us/download/details.aspx?id=45331**, and then click **Download**.
3. On the **Choose the download you want** page, select the **PBIDesktop_x64.msi** check box, and then click **Next**.
4. In the message box, click **Run**.
5. In the **Microsoft Power BI Desktop (x64) Setup** dialog box, on the **Welcome to the Microsoft Power BI Desktop (x64) Setup Wizard** page, click **Next**.
6. On the **Microsoft Software License Terms** page, select the **I accept the terms in the License Agreement** check box, and then click **Next**.
7. On the **Destination Folder** page, click **Next**.
8. On the **Ready to install Microsoft Power BI Desktop (x64)** page, click **Install**.
9. In the **User Account Control** dialog box, click **Yes**.
10. On the **Completed the Microsoft Power BI Desktop (x64) Setup Wizard** page, clear the **Launch Microsoft Power BI Desktop** check box, and then click **Finish**.
11. Close Internet Explorer.

Connect to a Database in Azure SQL Database and Import Data

1. In the **D:\Demofiles\Mod06** folder, run **Setup.cmd** as Administrator, and then click **Yes** when prompted. If asked: Do you want to continue with this operation? type **Y**, and then press Enter.
2. When the script completes, press any key to close the window.
3. Start Microsoft SQL Server Management Studio from the taskbar, and then connect to the **MIA-SQL** database engine instance by using Windows® authentication.

4. In the **D:\Demofiles\Mod06\Demo** folder, open the **Demo.ssmssln** solution.
5. In Solution Explorer, open the **1 - Charts.sql** script file.
6. On the desktop, double-click the **Power BI Desktop** icon.
7. In the **Power BI Desktop** window, click **Get Data**.
8. In the **Get Data** dialog box, click **Microsoft Azure SQL Database**, and then click **Connect**.
9. In the **SQL Server Database** window, in the **Server** box, type the URL of the Azure server ***<Server Name>*.database.windows.net** (where <*Server Name*> is the name of the server that you created).
10. In the **Database (optional)** box, type **AdventureWorksLT**.
11. Expand the **Advanced options** box.
12. In SQL Server Management Studio, copy the query under **Customer Address** in the **1 - Charts.sql** query.
13. In Power BI Desktop, paste the query into the **SQL Statement (optional)** box, and then click **OK**.
14. In the **Access a SQL Server Database** window, click **Database**, in the **Username** box, type **Student**, and in the **Password** box, type **Pa$$w0rd**, and then click **Connect**.
15. In the data preview window, click **Load**. The window will close and a blank report canvas will open.
16. In the **Untitled - Power BI Desktop** window, click the **Get Data** drop-down menu, and then click **More**.
17. In the **Get Data** dialog box, click **Microsoft Azure SQL Database**, and then click **Connect**.
18. In the **SQL Server Database** window, in the **Server** box, type the URL of the Azure server ***<Server Name>*.database.windows.net** (where <*Server Name*> is the name of the server that you created).
19. In the **Database (optional)** box, type **AdventureWorksLT**.
20. Expand the **Advanced options** box.
21. In SQL Server Management Studio, copy the query under **Sales** in the **1 - Charts.sql** query.
22. In Power BI Desktop, paste the query into the **SQL Statement (optional)** box, and then click **OK**.
23. In the data preview window, click **Load**. The window will close and return to the report.

Add Visualizations to a Report in Power BI Desktop

1. In the **Fields** pane, right-click **Query1**, click **Rename**, type **Customers**, and then press Enter.
2. Right-click **Query2**, click **Rename**, type **Sales**, and then press Enter.
3. Expand the two tables to display all of the fields.
4. In the **Sales** table, select the **SubCategory**, and **OrderQty** check boxes. Power BI creates a table.
5. In the **Visualizations** pane, click **Stacked column chart**.
6. Grab the expander on the right edge of the chart, and then widen the chart so that all category labels are visible.
7. In the **Visualizations** pane, click **Analytics**.
8. Expand **Constant Line**, and click **Add**.
9. In the **Value** box, type **100**.
10. Change the color to **red**.

11. Toggle **Data label** to **On**.
12. Change the color to **red** to match the reference line.
13. Click **Format**, and expand **Title**, in the **Title Text** box, type **Orders by Sub Category**, and then click **Center** to align to the center.
14. In the **Fields** pane, click **Sales**.
15. On the **Modeling** tab, click **New Column**.
16. In the formula bar, type the following code, and then press Enter:

```
LineTotal = Sales[OrderQty] * Sales[ListPrice]
```

17. Click **Format: General**, point to **Currency**, and then click **$ English (United States)**.
18. Click a blank area of the page.
19. In the **Sales** table, select the **Product** check box, which adds a table, and then select the **LineTotal** check box.
20. In the **Visualizations** pane, under **Filters**, expand **LineTotal(All)**, click **is greater than**, and then type **25000** in the box.
21. Click **Apply filter**, and then note that the number of products in the table is reduced.
22. In the **Visualizations** pane, click **Format**, click **Title,** and change the **Title** slider to **On**.
23. In the **Title text** box, type **Product Sales Over $25k**, and then click **Center**.
24. In the **Visualizations** pane, click **Stacked bar chart**.
25. Use the expander to widen the chart to the same width as the column chart.
26. On the chart, click **More Options**, and then click **Sort By LineTotal**.
27. Click the **+** icon at the bottom of the window to add a new report.
28. On the **Home** tab, click **Manage Relationships**, and then point out that Power BI has auto-detected the relationship on the **CustomerID** columns, and then click **Close**.
29. In the **Customers** table, select the **City** check box. Power BI automatically adds a map chart. Expand the map to show all countries.
30. In the **Sales** table, select the **LineTotal** check box to add it to the map. Grab the right corner of the map, and then drag it to fill the whole of the report page.
31. Zoom in on the map to focus on the **UK**. Point out that the bubbles now represent the sales for each customer, and are proportionately sized. Hold your mouse over some of the bubbles to display the data labels.
32. Close Power BI Desktop and SQL Server Management Studio without saving your changes.

    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 community gallery to fulfill your requirements?

Lesson 2
Managing a Power BI Solution

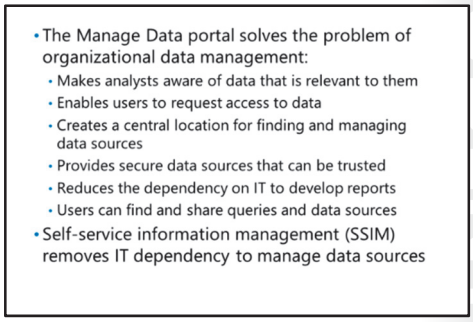
This lesson discusses the management aspect of Power BI. It examines how to use the Manage Data portal, what data is included in the Usage Report to show user activity within your Power BI tenant, and how to manage shared queries and data sources.

Lesson Objectives

After completing this lesson, you will be able to:

- Understand the benefits of the Manage Data portal.
- View usage analytics by using the Usage Report.
- Manage your shared queries.
- Manage your data sources.

The Manage Data Portal



A common problem in organizations is the ability for business users to find the relevant data to use in their analysis and reporting. More often than not, data is hard to find, secured, and even hidden. This means that key personnel are forced to make choices without all of the data that could be available to them, resulting in misguided decisions. Many organizations do not put enough resources into their information management infrastructure, making it harder to manage and explore the data that does exist. Many data sources provide data in different formats, from on-premises and cloud databases, publicly available datasets, SaaS vendors, internally shared workbooks, and other reporting solutions, all of which further compound the problem. The ability to manage data in a straightforward way provides a welcome solution.

The Manage Data portal that is delivered through Power BI for Microsoft Office 365® offers a cloud-based solution to these problems by helping to:

- Make analysts aware of business data that is relevant to them.
- Enable users to request access to data sources that they require for their analysis and reporting.
- Create a central location for finding data sources.
- Provide secure data sources that analysts can trust.
- Reduce the dependency on IT to develop reports that can take months to deliver.

Business users can find and share queries that are derived from a wide range of data sources by using the searching and sharing features in the Get & Transform functionality of Excel 2016. Business users can then manage these by using the following tasks in the Manage Data portal:

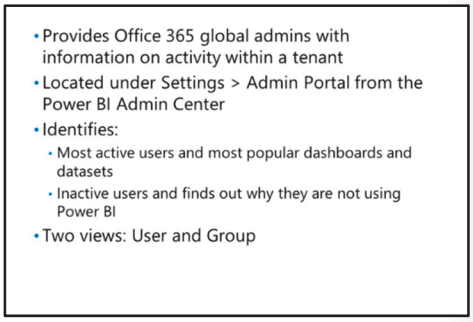
- View Usage Analytics
- View Shared Queries

- View and Manage Data Sources

Self-service information management (SSIM) enhances the experience of self-service BI, empowering users to manage their data without depending on IT.

Usage Analytics

Usage analytics provide administrators with detailed information regarding how Power BI users within the tenant are querying and accessing the data, reports, and dashboards. These analytics can help to identify the most active users, and the most frequently consumed dashboards and datasets, in addition to unused data and reports.

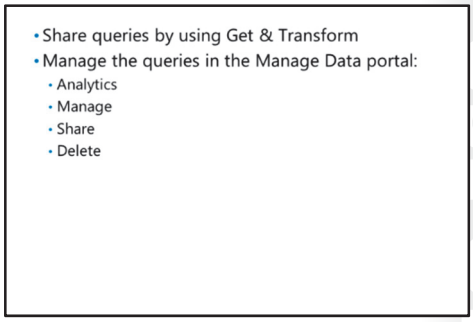


Office 365 global admins can view the usage analytics from the Power BI Admin Center. Click the **Settings** gear icon, and then click **Admin Portal**. This loads the Usage Report, which may take a while to load all of the data the first time you view it. After loading, a dashboard displays two tiled sections:

- The first section displays usage analytics for individual users, with a count of user dashboards, reports, and datasets within the tenant. Further to this is a bar chart that displays **Most Consumed Dashboards by Users**, and a tree map that shows **Most Consumed Packages by Users**. The term "packages" refers to content packs that have been created within the organization, and from SaaS providers such as Bing, Salesforce, and Facebook. The **Top Users with Most Dashboards** and **Top Users with Most Reports** tables show how many reports and dashboards individual users have access to. This can be indicative of users who are most active within Power BI.
- The second section offers the same usage information, but it is presented for groups. You can see which groups are most active and the data that they are accessing. At the top of the section are counts for the number of group dashboards, reports, and datasets. As per the previous section, a bar chart displays the **Most Consumed Dashboards by Groups**, and a tree map shows **Most Consumed Packages by Groups**. There is also a table each for **Top Groups with Most Dashboards** and **Top Groups with Most Reports**.

This information enables you to see how users access data and dashboards, and 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 also investigate why particular dashboards are not being used. There may be an underlying problem with the data: for example, perhaps the right data has not been made available, or it does not cover a required time period.

Shared Queries



In Excel 2016, queries can be shared by business users, or data stewards who are using Power BI for Office 365. After queries have been shared, they are available to users who have sufficient permissions. Using the online search option on the **Get & Transform** menu in Excel, users can search for and reuse shared queries in their data analysis and reports. Users can view their shared queries in the Manage Data portal. Log on to the portal, and then click **my queries** in the left pane. You are presented with a list of queries that are sorted in alphabetical order by the name and description that you gave them in Get & Transform. The portal provides several features for managing your shared queries by clicking the **open menu** ellipsis next to the query, or by right-clicking.

Analytics

Click **ANALYTICS** to view the usage of a shared query by using different views:

- **Queries shown in search over time**. See how many times the query was searched over a selected period of time. The analytics display how often the query appeared in a search result, so if the count is very low, users may not be finding it easily when searching. In this case, you can alter the metadata to make it more discoverable.
- **Clicks from search**. Find out how many times the shared query was imported into Excel by users over a selected period of time. See the **Top Query Users** who imported the query, and the number of times the user clicked the shared query. This information enables you to find out whether there are issues with a query, such as the data not meeting requirements and therefore not being clicked and imported. The Microsoft Lync® information for each user is included in the analysis display area, including online presence, location, status, and contact details. You contact users if you want to follow up on their usage.
- **Filters**. Use the filters to select the shared query that you want to analyze, and then set the timeframe from **last day**, or **last 30 days**. You can compare multiple shared queries under the **Queries** section to the compare usage statistics. After logging on to the Manage Data portal, you automatically see the analytics for all of your shared queries.

Manage

You can edit your query by using the **MANAGE** option in the Manage Data portal. Furthermore, you can change the name of the query and add or update the description, and the documentation URL.

Share

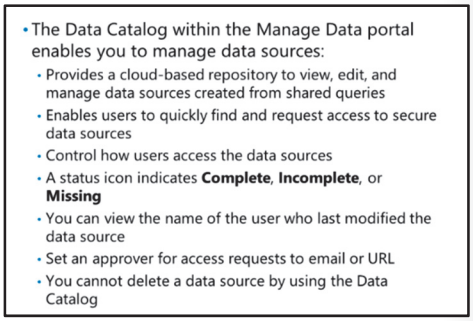
The **SHARE** option enables you to view the users who already share the query, and send an invitation to other users. Click **Invite people**, and then type the users' email addresses, or type "everyone" for sharing the query with all users within your organization. Click **Share**, and users are invited to share your query.

Delete

Click **DELETE** if you no longer need the query. Confirm that you want to delete the query by clicking **Delete** in the **Confirmation** dialog box.

Data Sources

In the Manage Data portal in Power BI for Office 365, there is a cloud-based repository called the Data Catalog. When you share a query, the metadata for the data source connection is held in this storage area, enabling you to manage your data sources from a single location. The Data Catalog enables users in your organization to quickly find and request access to data sources. Users then feel reassured that they are using data sources that are secure.



The Manage Data portal enables you to view data sources that are shared by others in your organization, and edit basic information about the data sources from the queries that you have shared. You can edit the display name and description, and control how colleagues in your organization can request access to the data source. You can allow them to create and share data from the data source, or use the shared queries that are included with the data source. You can also provide an email address so that, when a user wants to connect to data by using the data source from Get & Transform, an already composed email message is opened via the user's default mail program, enabling him or her to request access.

Use the following steps to manage your data sources:

1. Sign in to the Manage Data portal.
2. In the left pane, click **data sources** to display a list of all of your data sources. You can view, sort, and filter the list of data sources. The list contains information about each of the data sources, including **Name**, **Status**, **Location**, **Description**, **Modified Date**, and **Modified By**. The **Status** column in the data source includes a color icon so that you can quickly see an overview of all sources. A green icon indicates that the data source is **Complete**, yellow is **Incomplete**, and red highlights that the data source is **Missing**. The **Modified By** column derives information from Lync, so if the user who last modified the data source is present, this is indicated by a colored icon.
3. In the list, select the data source that you want to edit, and then right-click the list item, or click **open menu** that is located next to the data source name. In the dialog window, click **EDIT**.
4. In the next screen, you can edit the name of the data source, and the description that users see when they search for the data source in the Power BI Data Catalog. Try to be specific and informative in naming and describing data sources.
5. In the **Approver for access requests** field, you can enter an email address or URL:
    - **Email address**. Type the email address of the person who will respond to requests for access. When a user requests access to the data source, an already composed email message opens on his or her local computer by using their default mail client. The user can optionally include details of why they need access, and then send the email message.
    - **URL**. Enter a URL string. When the user requests access, the URL opens in the user's default browser.

Note: The Power BI Data Catalog enables you to view and manage data sources. You cannot delete a data source from here because of the implications elsewhere.

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 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, shape, and combine data in Power BI.
- Create a report by using chart and map visuals.
- Publish reports and share dashboards.
- Optionally add visualizations that are relevant to your organization.

Estimated Time: 60 minutes

Virtual machine: **10989B-MIA-SQL**

User name: **ADVENTUREWORKS\Student**

Password: **Pa$$w0rd**

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 help 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 it 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

1. Ensure that the MSL-TMG1, 10989B-MIA-DC and 10989B-MIA-SQL virtual machines are running, and then log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. Run **Setup.cmd** in the **D:\Labfiles\Lab06\Starter** folder as Administrator.
3. Sign up for an Office 365 login if you do not already have one.

▶ Task 2: Connect to Existing Data in Azure

1. Open the **Lab Exercise 1.sql** file in the **D:\Labfiles\Lab06\Starter\Project** folder.
2. Open Power BI Desktop.
3. Connect to the **AdventureWorksLT** database in Azure SQL Database.
4. Run the query under **Task 1** in the **Lab Exercise 1.sql** file to import the customer data.
5. Connect to the **AdventureWorksLT** database in Azure SQL Database.
6. Run the query under **Task 2** in the **Lab Exercise 1.sql** file to import the sales data.
7. Leave Power BI Desktop open for the next task.

▶ Task 3: Shape Data

1. Rename **Query1** as **Customers**.
2. Rename **Query2** as **Sales**.
3. Delete the **NameStyle** column from the **Customers** table.
4. Delete the **SalesPerson** column.
5. Hide the **CustomerID** column in the report view.
6. Change the data category of the **AddressLine1** column to **Address**.
7. Change the data category of the **City** column to **City**.
8. Change the data category of the **StateProvince** column to **State or Province**.
9. Change the data category of the **CountryRegion** column to **Country/Region**.
10. Change the data category of the **PostalCode** column to **Postal Code**.
11. Add a new column called **FullAddress**, and then for the value of each row, concatenate **AddressLine1**, **City**, **StateProvince**, **CountryRegion**, and **PostalCode**.
12. In the **Sales** table, delete the **RevisionNumber** column.
13. Delete the **SalesOrderNumber** column.
14. Hide the **CustomerID** column in the report view.
15. Hide the **SalesOrderID** column in the report view.
16. Hide the **SalesOrderDetailID** column in the report view.
17. Add a new column called **LineTotal**, which multiplies the **OrderQty** column by the **ListPrice** column.
18. Change the format of the **LineTotal** column to the currency **$ English (United States)**.
19. Add a new measure called **TargetSales**, which multiplies the **LineTotal** column by **1.2**.
20. Save the report as **AdventureWorksLT Sales.pbix** in the **D:\Labfiles\Lab06\Starter** folder.
21. Leave Power BI Desktop open for the next task.

▶ Task 4: Combine Data

1. Open the **States.xlsx** file in the D:\Labfiles\Lab06\Starter\Project folder.
2. In the **States** worksheet, select the data, and then copy it.
3. Create a new table named **Sales by States** by pasting in the data from the workbook.

4. Connect to Wikipedia to import a list of states in America.
5. Remove the last 26 rows of the imported data.
6. Remove columns that will not be used.
7. Change the name of the table to **States with Codes**.
8. Set the first row to be the header row.
9. Rename the **United States of America** column to **State Name**.
10. Rename the **US USA 840** column to **State Code Long**.
11. Rename the **US** column to **State Code Short**.
12. Merge the data into the **Sales by State** table, and then exclude the **State Name** column.
13. Name the new column **State Code**.
14. Click **Close & Apply**.
15. Hide the **States with Codes** table in the report view.
16. Save the file, and then leave Power BI Desktop open for the next exercise.

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

1. Add a gauge chart to the report. Use the **LineTotal** field for the **Value** property, and the **TargetSales** measure for the **Target value** property.
2. Set the **Max** value of the gauge to **146000**.
3. Change the title of the gauge to **Target Sales**, and then center-align the text.
4. Drag the **CompanyName** and **LineTotal** fields onto the report to create a table.
5. Change the table to a pie chart, and then expand the chart to show all company names.
6. Change the title of the pie chart to **Top Selling Companies**, and then center-align the text.
7. Drag the **MainCategory** field onto the report canvas to create a table.
8. Add the **OrderQty** column.
9. Convert the table to a stacked bar chart.

10. Use the **OrderQty** field values for the **Color saturation** property of the chart.
11. Click **Analytics**, expand **Constant Line**, and click **Add**. Set **Value** to **500**, and then change **Color** to **red**.
12. Toggle the data label for **Constant Line**, and then set **Color** to **red**.
13. Change the title of the chart to **Orders by Main Category**, and then center-align the text.
14. Drag a donut chart onto the report.
15. Tick **MainCategory** and **LineTotal** in the **Sales** table.
16. Change the title to **Sales by Main Category**, and center-align the text.
17. Create a table by dragging the **Product** field onto the report canvas.
18. Add the **LineTotal** field to the table chart.
19. Tick **MainCategory** to include this column.
20. Filter the products to display only those that have sales greater than **$32,000**.
21. Filter the list to display only bikes.
22. Change the table to a stacked column chart.
23. Change the title to **Top 10 Selling Bikes**, and center-align the text.
24. Add a **Constant Line**, with the **Value** set to **35000**, and then set **Color** to **red**.
25. Expand the chart to fill the remaining space on the report. If necessary, move your visuals around to make them fit.
26. Save the report.

▶ Task 2: Create a Map Visualization

1. Add a new page to the report.
2. Create a map by adding the **City** field from the **Customers** table.
3. Add the **LineTotal** field from the **Customers** table.
4. Expand the map so that all bubbles are visible.
5. Rename the map **World Sales by City**.
6. Click the report canvas, and then create a new map by adding the **State Code** field from the **Sales by State** table.
7. Add the **SalesYTD** field from the **Sales by State** table.
8. Change the type of map to **Filled Map**.
9. Resize the map to show all states, and then see how the sales are clustered in one area.
10. View the data for **California(CA)**, and then change the format of the **SalesYTD** column to **$ English (United States)**.
11. Rename the map **Sales by State**.
12. Save the report, and then leave it open for the next exercise.

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

1. On the **Home** ribbon, select **Publish**.
2. Sign in to Power BI.
3. Publish the report to **My Workspace**.
4. When the report has published, click **Open 'AdventureWorksLT Sales.pbix' in Power BI**.
5. In Internet Explorer, sign in to Power BI if you are prompted to do so.
6. View the report in Power BI.
7. Remain signed in to Power BI for the next task.

▶ Task 2: Create a Power BI Dashboard

1. In the My Workspace pane, under **Reports**, select **AdventureWorksLT Sales**.
2. On the **Target Sales** visual, click **Pin visual**, and then create a new dashboard called **AdventureWorksLT Sales**.
3. Pin the **Top Selling Customers** visual to the **AdventureWorksLT Sales** dashboard.
4. Pin the **Orders by Main Category** visual to the **AdventureWorksLT Sales** dashboard.
5. Pin the **Top 10 Selling Bikes** visual to the **AdventureWorksLT Sales** dashboard.
6. Pin the **Sales by Main Category** visual to the **AdventureWorksLT Sales** dashboard.
7. Open the dashboard from the link under the My Workspace pane.
8. On the **Target Sales** tile, open the menu, and then in **Tile details**, give the chart a subtitle of **Sales target for 2016**.
9. On the **Top Selling Customers** tile, open the menu, and then in **Tile details**, give the chart a subtitle of **Customers selling the most products**.
10. Open the **Top 10 Selling Bikes** tile in Focus mode. Filter **LineTotal** from **32000** to **40000**.
11. Click **Back to AdventureWorksLT Sales**.
12. Click **Enter Full Screen Mode**. Notice that the browser is hidden.
13. Press Esc to exit full-screen mode and close Internet Explorer.

Results: After this exercise, you should have published a report to the Power BI service and used the visuals to create a dashboard.

Exercise 4: (Optional): Adding Visualizations

Scenario

You presented the concept dashboard to your senior managers, and they agreed that Power BI is the best strategy for delivering self-service BI. Now that you have their agreement, you will start creating a report by using visualizations that will be useful to department heads.

The main tasks for this exercise are as follows:

1. Add Additional Visualizations That Are Relevant to Your Organization

▶ Task 1: Add Additional Visualizations That Are Relevant to Your Organization

1. Add a new report page.
2. In the **Visualizations** pane, add visuals that are relevant to your organization.
3. Try downloading a custom visual from the community gallery.
4. Use existing data to test the visuals.

Results: After this exercise, you should have created a report containing visuals that will serve as the basis for a departmental report.

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 use at the moment?

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?

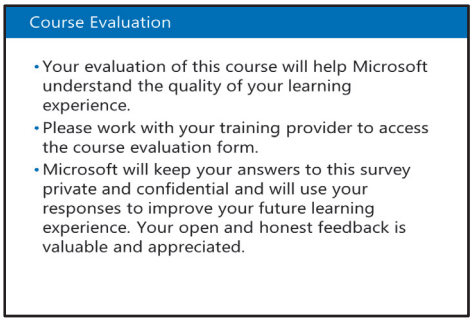
Module Review and Takeaways

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?

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.

Module 1: Introduction to Self-Service BI Solutions

Lab: Exploring an Enterprise BI Solution

Exercise 1: Viewing Reports

▶ Task 1: Prepare the Lab Environment

1. Ensure that the 10989B-MIA-DC, 10989B-MIA-SQL, and MSL-TMG1 virtual machines are running, and then log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. In the **D:\Labfiles\Lab01\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**.
4. At the command prompt, type **Y**, and then press Enter.
5. Wait until the script completes, and then press any key.
6. If you do not have a Power BI login, open Internet Explorer, browse to **https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-signing-up-for-power-bi-with-a-new-office-365-trial**, and then follow the steps to create an account.
7. In Internet Explorer, browse to **https://www.microsoft.com/en-us/download/details.aspx?id=45331**, and then click **Download**.
8. On the **Choose the download you want** page, select the **PBIDesktop_x64.msi** check box, and then click **Next**.
9. In the message box, click **Run**.
10. In the **Microsoft Power BI Desktop (x64) Setup** dialog box, on the **Welcome to the Microsoft Power BI Desktop (x64) Setup Wizard** page, click **Next**.
11. On the **Microsoft Software License Terms** page, select the **I accept the terms in the License Agreement** check box, and then click **Next**.
12. On the **Destination Folder** page, click **Next**.
13. On the **Ready to install Microsoft Power BI Desktop (x64)** page, click **Install**.
14. In the **User Account Control** dialog box, click **Yes**.
15. On the **Completed the Microsoft Power BI Desktop (x64) Setup Wizard** page, clear the **Launch Microsoft Power BI Desktop** check box, and then click **Finish**.
16. Close Internet Explorer.

▶ Task 2: View Reports in SharePoint Server

1. In **File Explorer**, navigate to the **D:\Labfiles\Lab01\Starter\Project** folder, and double-click **Adventure Works Sales.xlsx**.
2. If the security warning appears to say "External Data connections have been disabled", click **Enable Content**.
3. On the **SalesPerson** sheet, click any cell in the table to select the table so the **Design** tab appears.
4. On the **Design** tab, click **Summarize with PivotTable**.
5. In the **Create PivotTable** dialog box, click **OK**.

6. In the new sheet, in **PivotTable Fields**, select the **FirstName** check box to add this to the Axis.
7. Select the **SalesYTD** check box, to add this to Values.
8. In the **Tools** group, click **PivotChart**.
9. In the **Insert Chart** dialog box, leave **Clustered Column** selected, and click **OK**.
10. On the **Design** tab, in the Location group, click **Move Chart**.
11. In the **Move Chart** dialog box, click **New sheet**, type **Sales Person Chart**, and then click **OK**.
12. On the **File** menu, click **Save As**, click **Browse**.
13. In the **Save As** dialog box, in the file location bar, type **http://mia-sql/sites/adventureworks**, and then press Enter.
14. Click **Documents**, click **Open**, and then click **Save**.
15. Open **Internet Explorer**, and navigate to **http://mia-sql/sites/adventureworks/Shared Documents**.
16. Click **Adventure Works Sales** to open the workbook in Excel Services.
17. Click **Sales Person Chart** to view the clustered column chart.
18. Close Internet Explorer, and then close Excel.

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

Exercise 2: Creating a Power BI Report

▶ Task 1: Import Data into Power BI Desktop

1. On the desktop, double-click **Power BI Desktop**.
2. When the Get Data screen shows, click **Get Data**.
3. In the **Get Data** dialog box, click **SQL Server Database**, and then click **Connect**.
4. In the **SQL Server Database** dialog box, in the **Server** box, type **MIA-SQL**.
5. In the **Database (optional)** box, type **AdventureWorksDW2016**, and then click **OK**.
6. If the **Access a SQL Server Database** dialog box appears, leave the default settings unchanged, and then click **Connect**.
7. If the **Encryption Support** dialog box appears, click **OK**.
8. In the **Navigator** dialog box, select the **FactInternetSales** check box, click **Select Related Tables**, and then click **Load**.
9. On the **File** menu, click **Save**.
10. In the **Save As** dialog box, navigate to **D:\Labfiles\Lab01\Starter\Project** folder, in the **File name** box, type **Adventure Works Sales**, and the click **Save**.
11. Leave Power BI Desktop open for the next exercise.

▶ Task 2: Add Visualizations to the Report

1. In the **Fields** pane, expand **FactInternetSales**, and drag the **SalesAmount** field onto the report canvas to create a column chart.
2. Expand **DimDate**, and drag the **EnglishDayNameOfWeek** field to the **Axis** property.
3. Move the chart to the top left-hand corner of the canvas, and expand the chart width so the days of the week display in full.
4. In the **Visualizations** pane, click **Format**.
5. Expand **Title**, and in the **Title Text** box, type **Sales by Day of Week**.
6. Next to **Alignment**, click the **Center** icon.
7. In the **Fields** pane, under **FactInternetSales**, drag the **SalesAmount** field onto the report canvas to create a column chart.
8. Under **DimDate**, drag the **CalendarQuarter** field onto the chart. Notice that there is only one column.
9. In the Visualizations pane, click **Fields**. Drag the **CalendarQuarter** field from **Value** to **Axis**.
10. In the **Visualizations** pane, click **Format**.
11. Expand **Title**, and in the **Title Text** box, type **Sales by Calendar Quarter**.
12. Next to **Alignment**, click the **Center** icon.
13. Expand **Data colors**, change **Show all** to **On**, and for **1**, select **red**, for **2**, select **blue**, and for **3**, select **yellow**.
14. Move the chart to the right of the **Sales by Day of Week** chart, and expand it so both charts are the same height.
15. In the **Fields** pane, expand **DimSalesTerritory**, and drag the **SalesTerritoryCountry** column onto the report canvas under the **Sales by Day of Week** chart.
16. Under **FactInternetSales**, drag the **SalesAmount** field onto the map.
17. Expand the map to show all the values.
18. In the **Visualization** pane, click **Format**.
19. Expand **Title**, and in the **Title Text** box, type **Sales by Country**.
20. Next to **Alignment**, click the **Center** icon.
21. In the **Fields** pane, expand **DimCustomer**, and drag the **CommuteDistance** field onto the report canvas under the **Sales by Calendar Quarter** chart.
22. Under **FactInternetSales**, drag the **SalesAmount** field onto the chart.
23. In the **Visualizations** pane, click **Donut chart**.
24. In the **Visualization** pane, click **Format**.
25. Expand **Title**, and in the **Title Text** box, type **Sales by Commute Distance**.
26. Next to **Alignment**, click the **Center** icon.
27. On the **File** menu, click **Save**.

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

Exercise 3: Creating a Power BI Dashboard

▶ Task 1: Create a Power BI Dashboard

1. In Power BI Desktop, on the **Home** tab, click **Publish**.
2. If you are prompted to save your changes, click **Save**.
3. In the **Sign in to Power BI** dialog box, click **Sign in**.
4. Enter the email address and password for your account, and click **Sign in**.
5. The report will then be published to the Power BI portal.
6. In the **Publishing to Power BI** dialog box, click **Open Adventure Works Sales.pbix in Power BI** to view the report online.
7. When the browser opens, if you are prompted to enter your Power BI credentials, enter your email address and password, and wait for the report to open.
8. On the **Sales by Day of Week** column chart, click **Pin visual**.
9. In the **Pin to dashboard** dialog box, click **New dashboard**, and type **Adventure Works Sales**, and then click **Pin**.
10. On the **Sales by Calendar Quarter** column chart, click **Pin visual**.
11. In the **Pin to dashboard** dialog box, click **Existing dashboard**, in the list click **Adventure Works Sales**, and then click **Pin**.
12. On the **Sales by Country** map chart, click **Pin visual**.
13. In the **Pin to dashboard** dialog box, click **Existing dashboard**, in the list click **Adventure Works Sales**, and then click **Pin**.
14. On the **Sales by Commuter Distance** donut chart, click **Pin visual**.
15. In the **Pin to dashboard** dialog box, click **Existing dashboard**, in the list click **Adventure Works Sales**, and then click **Pin**.
16. In the upper-left corner of the window, below the PowerBI icon, click **Show the navigation pane**.
17. Under **My Workspace**, expand **Dashboards**, and then click **Adventure Works Sales**.

▶ Task 2: Ask Questions of Your Data

1. In the **Adventure Works Sales** dashboard, click in the **Ask a question about your data** box.
2. In the list, click **dim products**.
3. Power BI shows you a table of the data in the **DimProduct** table.
4. Type **how many customers**, the count of 18484 shows in the results, and then click **Pin visual**.
5. In the **Pin to dashboard** dialog box, click **Existing dashboard**, in the list click **Adventure Works Sales**, and then click **Pin**.
6. Type **who is the oldest customer**, and the results show the customers ordered by BirthDate.
7. Type **how many products are red**, and the result is displayed.
8. Type **which country has the most male customers**, and a bar chart shows the results, and then click **Pin visual**.
9. In the **Pin to dashboard** dialog box, click **Existing dashboard**, in the list click **Adventure Works Sales**, and then click **Pin**.

10. In the Navigation pane on the left-hand side, under **My Workspace**, under **Dashboards**, click **Adventure Works Sales**. The two additional tiles now appear.

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

Module 2: Introducing Power BI

Lab: Creating a Power BI Dashboard

Exercise 1: Connecting to Power BI Data

▶ Task 1: Prepare the Lab Environment

1. Ensure that the MSL-TMG1, 10989B-MIA-DC, and 10989B-MIA-SQL virtual machines are running, and then log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. In the **D:\Labfiles\Lab02\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**, and then wait for the script to finish.
4. If prompted whether you want to continue the operation, type **Y**, and then press Enter.
5. When the script completes, press any key to close the window.
6. If you do not have a Power BI login, open **Internet Explorer**, go to **https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-signing-up-for-power-bi-with-a-new-office-365-trial**, and follow the steps to create an account.
7. In Internet Explorer, go to **https://www.microsoft.com/en-us/download/details.aspx?id=45331**, and then click **Download**.
8. On the **Choose the download you want** page, select the **PBIDesktop_x64.msi** check box, and then click **Next**.
9. In the message box, click **Run**.
10. In the **Microsoft Power BI Desktop (x64) Setup** dialog box, on the **Welcome to the Microsoft Power BI Desktop (x64) Setup Wizard** page, click **Next**.
11. On the **Microsoft Software License Terms** page, select the **I accept the terms in the License Agreement** check box, and then click **Next**.
12. On the **Destination Folder** page, click **Next**.
13. On the **Ready to install Microsoft Power BI Desktop (x64)** page, click **Install**.
14. In the **User Account Control** dialog box, click **Yes**.
15. On the **Completed the Microsoft Power BI Desktop (x64) Setup Wizard** page, clear the **Launch Microsoft Power BI Desktop** check box, and then click **Finish**.
16. Close Internet Explorer.
17. On the Desktop, right-click the **Power BI Desktop** shortcut, and then click **Pin to Taskbar**.

▶ Task 2: Connect to SQL Server from the Power BI Desktop

1. On the taskbar, click **Power BI Desktop**.
2. In the **Power BI Desktop** window, click **Get Data**.
3. In the **Get Data** dialog box, click **SQL Server Database**, and then click **Connect**.
4. In the **SQL Server Database** dialog box, in the **Server** box type **MIA-SQL**, in the **Database (optional)** box, type **AdventureWorks2016**, and then click **OK**.
5. In the **Access a SQL Server Database** dialog box, accept the default values, and then click **Connect**.

6. If an **Encryption Support** message is displayed, click **OK**.
7. In the **Navigator** dialog box, select the **Sales.vSalesPerson** check box, and then click **Load**.
8. In the **Fields** pane, expand **Sales vSalesPerson** to view all the columns.
9. On the **Home** ribbon, click **Recent Sources**, and then click **MIA-SQL: AdventureWorks2016**.
10. In the **Navigator** dialog box, select the **Sales.vStoreWithDemographics** check box, and then click **Load**.
11. In the **Fields** pane, expand **Sales.vStoreWithDemographics** to view all the columns.
12. On the **Home** ribbon, click **Get Data**, and then click **SQL Server**.
13. In the **SQL Server Database** dialog box, in the **Server** box, type **MIA-SQL**, and then in the **Database (optional)** box, type **AdventureWorks2016**.
14. Expand **Advanced options**, in the **SQL statement (optional)** box, type the following code, and then click **OK**:

```
SELECT TOP 10 P.ProductID, P.Name AS Product, SUM(CAST(LineTotal AS decimal(18,2)))
AS LineTotal
FROM Purchasing.PurchaseOrderDetail AS POD
INNER JOIN Production.Product AS P
ON POD.ProductID = P.ProductID
GROUP BY P.ProductID, P.Name
ORDER BY LineTotal DESC
```

15. If the **Connection Settings** window appears, leave **Import** selected, and then click **OK**.
16. In the **MIA-SQL: AdventureWorks2016** dialog box, click **Load**.
17. In the **Fields** pane, expand **Query1** to view all columns.
18. Next to **Query1**, click the ellipsis, click **Rename**, type **Top 10 Selling Products**, and then press Enter.

▶ Task 3: Add Charts to the Report

1. In the **Visualizations** pane, click **Stacked column chart**.
2. In the **Fields** pane, under **Sales vSalesPerson**, drag the **FirstName** field to the **Axis** box in the Visualizations pane.
3. Drag the **SalesYTD** field to the **Value** box. The chart will populate with the data.
4. On the chart in the report, click and drag the sizer on the right-hand side of the chart to widen the chart and display all the salespeople.
5. Ensure the chart has focus, and then in the **Visualizations** pane, click **Format**.
6. Expand **Data colors**, then toggle **Show all** to **On**. Change the color for **Jae**, **Linda** , and **Michael** to red.
7. Click on the report canvas, then in the **Visualizations** pane, click **Pie chart**.
8. Drag the pie chart to the right of the bar chart, or below if there is not enough space.
9. In the **Fields** pane, expand **Sales vStoreWithDemographics**, drag the **Specialty** field to the **Legend** box in the Visualizations pane.
10. Drag the **NumberEmployees** field to the **Values** box. The chart will populate with the data and should display three pie sections.

11. Click on the report canvas, then in the **Visualizations** pane, click **Stacked column chart**. Position the chart so it is under the previous charts.
12. In the **Fields** pane, expand **Top 10 Selling Products**, drag the **Product** field to the **Axis** box in the Visualizations pane.
13. Drag the **LineTotal** field to the **Value** box. The chart will populate with the data.
14. Click the **Top 10 Selling Products** chart to give it focus, then in the **Visualizations** pane, click **Donut chart**. Note how easy it is to switch to a different chart type.
15. On the chart, grab the sizer on the right-hand side of the donut chart to widen the chart to display all the product names in full.
16. In the **Fields** pane, under **Sales vStoreWithDemographics**, click and drag the **AnnualSales** field directly onto the report canvas. See how this automatically creates a bar chart.
17. In the **Fields** pane, select the **AnnualRevenue** field check box, and note that this adds the field to the bar chart.
18. In the **Fields** pane, click the ellipsis next to the **AnnualRevenue** field, click **Rename**, type **Annual Revenue**, and then press Enter.
19. Repeat step 18 to rename the **AnnualSales** field to **Annual Sales**. Note that the names in the title and legend of the bar chart update accordingly.
20. Click on the report canvas, and then in the **Visualizations** pane, click **Format**.
21. Expand **Page Information**, in the **Name** box, type **Sales**.
22. Click the report canvas and note the name has changed in the tab at the bottom of the report.
23. On the **File** menu, click **Save**.
24. In the **Save As** dialog box, navigate to **D:\Demofiles\Mod02**, create a folder named **Power BI**. Name the report **Adventure Works Sales**, and save in the **Power BI** folder.
25. Leave Power BI Desktop open on the report for the next task.

▶ Task 4: Publish the Report to the Power BI Portal

1. In Power BI Desktop, on the **Home** ribbon, click **Publish**.
2. If you are prompted to save your changes, click **Save**.
3. In the **Sign in to Power BI** window, click **Sign In**.
4. Enter the email address and password for your Microsoft account in the Sign in to your Microsoft account window, then click **Sign in**.
5. The report will then be published to the Power BI portal.
6. In the **Publishing to Power BI** dialog box, click **Open Adventure Works Sales.pibx in Power BI** to view the report online.
7. When the browser opens, if you are prompted to enter your Power BI credentials, enter your email address and password, and wait for the report to open.
8. Leave the browser window open for the next lab exercise.

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

Exercise 2: Create a Power BI Dashboard

▶ Task 1: Create a New Dashboard

1. In the Power BI portal, click **Show the navigation pane**.
2. In the navigation pane, next to **Dashboards**, click the **Create dashboard** icon.
3. In the box, type **Adventure Works Sales**, and then press Enter.
4. The blank dashboard canvas will appear in the main window.
5. Leave the window open ready for the next lab task.

▶ Task 2: Add Chart Items to the Dashboard

1. In the navigation pane, under **Reports**, click **Adventure Works Sales**. This will open the report in the main window.
2. Position the cursor on the **SalesYTD by FirstName** bar chart, and then click **Pin visual**.
3. In the **Pin to dashboard** dialog box, click **Existing dashboard**, and in the drop-down list, click **Adventure Works Sales**, and then click **Pin**. You will see a notification to say the visual has been pinned.
4. Position the cursor on the **LineTotal by Product** bar chart, and then click **Pin visual**.
5. In the **Pin to dashboard** dialog box, click **Existing dashboard**, and in the drop-down list, click **Adventure Works Sales**, and then click **Pin**.
6. Position the cursor on the **Annual Sales and Annual Revenue** bar chart, and then click **Pin visual**.
7. In the **Pin to dashboard** dialog box, click **Existing dashboard**, and in the drop-down list, click **Adventure Works Sales**, and then click **Pin**.
8. In the navigation pane, under **Dashboards**, click **Adventure Works Sales** to open the dashboard. Leave the dashboard open for the next task.

▶ Task 3: Customize the Dashboard

1. In the **Adventure Works Sales** dashboard, position the cursor on the **Annual Sales, Annual Revenue** chart to change the icon to a hand, then drag the chart to the top left-hand corner of the canvas. It will change places with the **SalesYTD** chart. Notice that the names of the charts have been changed.
2. Position the cursor on the **SalesYTD** chart so that the arrow appears in the bottom right-hand corner of the chart. Grab the arrow and widen the chart so it spans the width of the charts above.
3. Position the cursor on the **SalesYTD** chart, in the top right-hand corner, click the ellipsis, and then click **Tile details**.
4. In the **Tile details** pane, in the **Title** box, change the name to **Year to Date Sales**.
5. In the **Subtitle** box, type **By Sales Person**, and then click **Apply**.
6. Repeat steps 3 to 5 for the **LineTotal** chart. Rename the **Title** to **Top 10 Selling Products**, and the **Subtitle** to **By Sales**.
7. Repeat steps 3 to 5 for the **Annual Sales, Annual Revenue** chart. Rename the **Title** to **Annual Sales v Annual Revenue**, and the **Subtitle** to **By Stores**.
8. In the **Ask a question about your data** box, type **best selling sorted by product**. When the table of products appears beneath the question box, click **Pin visual**.

9. In the **Pin to dashboard** dialog box, click **Existing dashboard**, and in the drop-down list, click **Adventure Works Sales**, and then click **Pin**.
10. In the navigation pane, under **Dashboards**, click **Adventure Works Sales**. The table of products will be at the bottom. Drag the table to the top row.
11. Repeat steps 3 to 5 for the table of products. Rename the **Title** to **Top 10 Selling Products**, and the **Subtitle** to **By Sales**.
12. Leave the dashboard open for the next lab task.

▶ Task 4: Display the Dashboard in Full Screen Mode

1. In the **Adventure Works Sales** dashboard, click **Enter Full Screen Mode**. Notice that the browser disappears, and a floating menu appears in the bottom right-hand corner of the screen.
2. Click **Fit to Screen** and notice how the screen zooms in to remove as much of the excess background space as possible.
3. Position the cursor on the **Year to Date Sales** tile, in the top right-hand corner of the chart, click the ellipsis, and then click **Focus mode**. The single tile is displayed and fits the screen.
4. In the bottom right-hand corner of the screen, on the floating menu, click **Exit Full Screen Mode**.
5. In the top right-hand corner of the screen, click **Pin visual**.
6. In the **Pin to dashboard** dialog box, click **New dashboard**, type **Year to Date Sales**, then click **Pin**.
7. In the top left-hand corner of the screen, click **Back to Adventure Works Sales** to return to the dashboard.

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

Module 3: Power BI Data

Lab: Importing Data into Power BI

Exercise 1: Importing Excel Files into Power BI

▶ Task 1: Prepare the Lab Environment

1. Ensure that the 10989B-MIA-DC, 10989B-MIA-SQL, and MSL-TMG1 virtual machines are running, and then log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. In the **D:\Labfiles\Lab03\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**.
4. At the command prompt, type **Y**, and press Enter.
5. When the script has completed, press any key to close the window.
6. If you do not have a Power BI login, open Internet Explorer, browse to **https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-signing-up-for-power-bi-with-a-new-office-365-trial**, and then follow the steps to create an account.
7. In Internet Explorer, browse to **https://www.microsoft.com/en-us/download/details.aspx?id=45331**, and then click **Download**.
8. On the **Choose the download you want** page, select the **PBIDesktop_x64.msi** check box, and then click **Next**.
9. In the message box, click **Run**.
10. In the **Microsoft Power BI Desktop (x64) Setup** dialog box, on the **Welcome to the Microsoft Power BI Desktop (x64) Setup Wizard** page, click **Next**.
11. On the **Microsoft Software License Terms** page, select the **I accept the terms in the License Agreement** check box, and then click **Next**.
12. On the **Destination Folder** page, click **Next**.
13. On the **Ready to install Microsoft Power BI Desktop (x64)** page, click **Install**.
14. In the **User Account Control** dialog box, click **Yes**.
15. On the **Completed the Microsoft Power BI Desktop (x64) Setup Wizard** page, clear the **Launch Microsoft Power BI Desktop** check box, and then click **Finish**.
16. Close Internet Explorer.
17. On the desktop, right-click the **Power BI Desktop** shortcut, and then click **Pin to Taskbar**.

▶ Task 2: Reduce the Size of Excel Files

1. In the **D:\Labfiles\Lab03\Starter\Project** folder, double-click **Adventure Works Data.xlsx** to open the file.
2. On the **Product Category** tab, click the **ProductCategoryID** column header, and then change the name to **Product Category ID** in the formula bar.
3. Click the **Name** column header, and then change the name to **Product Category** in the formula bar.

4. Select all rows and columns to highlight cells **A1** to **B5**, click the **Cell Styles** drop-down list, and then click **Normal**. The cells lose their color.
5. Select all rows and columns to highlight cells **A1** to **B5**, and then press CTRL+T.
6. In the **Create Table** dialog box, ensure that **My table has headers** is selected, and then click **OK**. The cells are transformed into a table.
7. On the **Design** ribbon, in the **Properties** group, change the name of the table to **ProductCategory**.
8. Click the **Product Subcategory** tab.
9. Click the **ProductSubcategoryID** column header, and then change the name to **Product Subcategory ID** in the formula bar.
10. Click the **ProductCategoryID** column header, and then change the name to **Product Category ID** in the formula bar.
11. Click the **Name** column header, and then change the name to **Product Subcategory** in the formula bar.
12. Select all rows and columns to highlight cells **A1** to **C38**, click the **Cell Styles** drop-down list, and then click **Normal**. The cells lose their color.
13. Select all rows and columns to highlight cells **A1** to **C38**, and then press CTRL+T.
14. In the **Create Table** dialog box, ensure that **My table has headers** is selected, and then click **OK**. The cells are transformed into a table.
15. On the **Design** ribbon, in the **Properties** group, change the name of the table to **ProductSubcategory**.
16. Click the **Products** tab.
17. Click the **ProductID** column header, and then change the name to **Product ID** in the formula bar.
18. Click the **Name** column header, and then change the name to **Product Name** in the formula bar.
19. Click the **ProductNumber** column header, and then change the name to **Product Number** in the formula bar.
20. Click the **StandardCost** column header, and then change the name to **Standard Cost** in the formula bar.
21. Click the **ListPrice** column header, and then change the name to **List Price** in the formula bar.
22. Click the **ProductSubcategoryID** column header, and then change the name to **Product Subcategory ID** in the formula bar.
23. Click the **ProductModelID** column header, and then change the name to **Product Model ID** in the formula bar.
24. Select all rows and columns to highlight cells **A1** to **H505**, click the **Cell Styles** drop-down list, and then click **Normal**. The cells lose their color.
25. Select all rows and columns to highlight cells **A1** to **H505**, and then press CTRL+T.
26. In the **Create Table** dialog box, ensure that **My table has headers** is selected, and then click **OK**. The cells are transformed into a table.
27. On the **Design** ribbon, in the **Properties** group, change the name of the table to **Products**.
28. Click the **Sales** tab.

29. Click the **SalesOrderID** column header, and then change the name to **Sales Order ID** in the formula bar.
30. Click the **SalesPerson** column header, and then change the name to **Sales Person** in the formula bar.
31. Click the **ProductCategory** column header, and then change the name to **Product Category** in the formula bar.
32. Click the **ProductSubcategory** column header, and then change the name to **Product Subcategory** in the formula bar.
33. Click the **ProductName** column header, and then change the name to **Product Name** in the formula bar.
34. Click the **OrderQty** column header, and then change the name to **Order Qty** in the formula bar.
35. Click the **OrderDate** column header, and then change the name to **Order Date** in the formula bar.
36. Click the **UnitPrice** column header, and then change the name to **Unit Price** in the formula bar.
37. Click the **UnitPriceDiscount** column header, and then change the name to **Unit Price Discount** in the formula bar.
38. Click the **LineTotal** column header, and then change the name to **Line Total** in the formula bar.
39. Click the **TotalDue** column header, and then change the name to **Total Due** in the formula bar.
40. Click the **Order Date** (I) column to select all of the cells, right-click, and then click **Format Cells**.
41. In the **Format Cells** dialog box, click **Date**, in the **Type** list, click ***Wednesday, March 14, 2012**, and then click **OK**.
42. Click to highlight and select all cells in columns **J** through **M**, right-click, and then click **Format Cells**.
43. In the **Format Cells** dialog box, click **Currency**, in the **Symbol** list, click **$ English (United States)**, and then click **OK**.
44. Select all rows and columns to highlight cells **A1** to **M60920**, click the **Cell Styles** drop-down list, and then click **Normal**. The cells lose their color.
45. Select all rows and columns to highlight cells **A1** to **M60920**, and then press CTRL+T.
46. In the **Create Table** dialog box, ensure that **My table has headers** is selected, and then click **OK**. The cells are transformed into a table.
47. On the **Design** ribbon, in the **Properties** group, change the name of the table to **Sales**.
48. Click **Save**.

▶ Task 3: Import Excel Files

1. In Internet Explorer, go to **https://powerbi.microsoft.com**, and sign in to your Power BI account.
2. If the Welcome to Power BI page displays, under **Files**, click **Get**, and then click **Local File**.
3. If the main Power BI page displays, in the navigation pane, click **Get Data**., and then under **Files**, click **Get**, and then click **Local File**.
4. In the **Choose File to Upload** dialog box, navigate to the **D:\Labfiles\Lab03\Starter\Project** folder, click **Adventure Works Data.xlsx**, and then click **Open**.
5. On the Local File page, click **Import** to import the Excel data into Power BI. This may take a minute or so to load.

6. After loading has completed, click Show the navigation pane, under **My Workspace**, under **Datasets**, click **Adventure Works Data**.
7. In the **Fields** pane, notice that each of the tabs from Excel have been imported and converted into a table. Expand each of the tables to view the list of columns. These match the names of the columns in Excel.
8. Leave Internet Explorer open and remain signed in to Power BI for the next exercise.

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

Exercise 2: Viewing Reports from Excel Files

▶ Task 1: View Excel Power View Sheets As Power BI Reports

1. In File Explorer, in the **D:\Labfiles\Lab03\Starter\Project** folder, double-click the **Adventure Works Power View.xlsx** file.
2. On the **Power View Sales** tab, click **Enable** to display the Power View report.
3. If the **Install Silverlight** message appears, complete the following steps:
   a. Click the **Install Silverlight** link.
   b. When the file has downloaded, click **Run**.
   c. In the **User Account Control** dialog box, click **Yes**.
   d. In the **Install Silverlight** window, clear the **Make Bing my search engine**, and **Make MSN my homepage** check boxes, and then click **Install now**.
   e. On the **Installation successful** page, click **Close**.
   f. In Excel, in the **Adventure Works Power View.xlsx** file, click **Reload**.
4. Notice the visuals on the report and the **Sales Person** filter, and then close the file.
5. In Internet Explorer, sign in to your Power BI account if you are logged out.
6. In the **My Workspace** pane, click **Get Data**.
7. On the **Get Data** page, under **File**, click **Get**, and then click **Local File**.
8. In the **Choose File to Upload** dialog box, navigate to the **D:\Labfiles\Lab03\Starter\Project** folder, click **Adventure Works Power View.xlsx**, and then click **Open**.
9. On the **Local File** page, click **Import**. This will take a minute or so to load.
10. After loading has completed, under **My Workspace**, in the **Reports** list, click **Adventure Works Power View**.
11. At the bottom of the screen, click **Power View Sales** to open the report. The **Adventure Works Sales** report loads.
12. Click **Filters**.
13. Position the cursor next to **Sales Person(All)** so that the drop-down arrow appears, and then click the arrow icon. The list of salespeople appears.
14. Test the report by clicking some of the salespeople, and then check the data changes in the report.

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

Module 4: Shaping and Combining Data

Lab: Shaping and Combining Data

Exercise 1: Shape Power BI Data

▶ Task 1: Preparing the Environment

1. Ensure that the MSL-TMG1, 10989B-MIA-DC, and 10989B-MIA-SQL virtual machines are running, and then log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. In the **D:\Labfiles\Lab04\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**.
4. If a message asks **Do you want to continue with this operation?** type **Y** and press Enter.
5. Wait for the script to finish, and then press any key when prompted.
6. If you do not have a Power BI login, open **Internet Explorer**, go to **https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-signing-up-for-power-bi-with-a-new-office-365-trial**, and follow the steps to create an account.
7. In Internet Explorer, go to **https://www.microsoft.com/en-us/download/details.aspx?id=45331**, and then click **Download**.
8. On the **Choose the download you want** page, select the **PBIDesktop_x64.msi** check box, and then click **Next**.
9. In the message box, click **Run**.
10. In the **Microsoft Power BI Desktop (x64) Setup** dialog box, on the **Welcome to the Microsoft Power BI Desktop (x64) Setup Wizard** page, click **Next**.
11. On the **Microsoft Software License Terms** page, select the **I accept the terms in the License Agreement** check box, and then click **Next**.
12. On the **Destination Folder** page, click **Next**.
13. On the **Ready to install Microsoft Power BI Desktop (x64)** page, click **Install**.
14. In the **User Account Control** dialog box, click **Yes**.
15. On the **Completed the Microsoft Power BI Desktop (x64) Setup Wizard** page, clear the **Launch Microsoft Power BI Desktop** check box, and then click **Finish**.
16. Close Internet Explorer.
17. On the desktop, right-click the **Power BI Desktop** shortcut, and then click **Pin to Taskbar**.

▶ Task 2: Import Data from Excel

1. On the taskbar, click **Power BI Desktop**.
2. In the **Power BI Desktop** window, click **Get Data**.
3. In the **Get Data** dialog box, click **Excel**, and then click **Connect**.
4. In the **Open** dialog box, navigate to the **D:\Labfiles\Lab04\Starter** folder, click **Sales - Europe.xlsx**, and then click **Open**.
5. In the **Navigator** window, select the **Europe** check box, and then click **Load**.

6. In the **Untitled - Power BI Desktop** window, click **Get Data**, and select **Excel**.
7. In the **Open** dialog box, navigate the **D:\Labfiles\Lab04\Starter** folder, click **Sales - North America.xlsx**, and then click **Open**.
8. In the **Navigator** window, select the **North America** check box, and then click **Edit**.
9. Leave the **Query Editor** window open for the next exercise.

▶ Task 3: Apply Formatting to the Existing Data

1. In the **Queries** pane, click **Europe** to show the data preview.
2. Right-click the **ProductKey** column, and click **Remove**.
3. Right-click the **SalesOrderNumber** column, and click **Remove**.
4. Right-click the **SalesTerritoryCountry** column, click **Rename**, type **Country**, and then press Enter.
5. Right-click the **SalesTerritoryGroup** column, click **Rename**, type **Sales Territory**, and then press Enter.
6. Right-click the **EnglishProductCategoryName** column, click **Rename**, type **Main Category**, and then press Enter.
7. Right-click the **EnglishProductSubCategoryName** column, click **Rename**, type **Sub Category**, and then press Enter.
8. Right-click the **EnglishProductName** column, click **Rename**, type **Product**, and then press Enter.
9. Right-click the **Color** column, point to **Move**, and then click **Left**.
10. In the **Queries** pane, click **North America**.
11. Right-click the **ProductKey** column, and click **Remove**.
12. Right-click the **SalesOrderNumber** column, and click **Remove**.
13. Right-click the **SalesTerritoryCountry** column, click **Rename**, type **Country**, and then press Enter.
14. Right-click the **SalesTerritoryGroup** column, click **Rename**, type **Sales Territory**, and then press Enter.
15. Right-click the **EnglishProductCategoryName** column, click **Rename**, type **Main Category**, and then press Enter.
16. Right-click the **EnglishProductSubCategoryName** column, click **Rename**, type **Sub Category**, and then press Enter.
17. Right-click the **EnglishProductName** column, and click **Rename**, type **Product**, and then press Enter.
18. Right-click the **Color** column, point to **Move**, and then click **Left**.
19. On the **Home** tab, in the **Query** group, click **Advanced Editor**. Notice that the query includes the changes you have made, then click **Cancel**.
20. Leave the Query Editor window open for the next exercise.

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

▶ Task 1: Add Related Data to the Shaped Data

1. In the **Queries** pane, click **Europe**.
2. On the **Home** tab, click **Combine**, and then click **Append Queries**.
3. In the **Append** dialog box, in the **Table to append** list, click **North America**, and then click **OK**. The rows are combined.
4. On the **Country** column header, click the **Arrow**, and then click **Load more**. You should now see that United States, and Canada are included. Click **Cancel**.
5. In File Explorer, navigate to the **D:\Labfiles\Lab04\Starter** folder, and then double-click **Country Codes.xlsx**.
6. In Excel, select the rows and columns with data, right-click on the highlighted cells, and then click **Copy**.
7. In Power BI Desktop, on the **Home** tab, in the **External Data** group, click **Enter Data**.
8. Right-click in the top-left cell, and select **Paste**.
9. In the **Name** box, type **Country Codes**, and then click **Load**.
10. In the **Queries** pane, click **Europe**.
11. On the **Home** tab, in the **Combine** group, click **Merge Queries**.
12. In the **Merge** dialog box, click the **Country** column to select it.
13. In the list below the table, click **Country Codes**.
14. Click the **Country** column, and then click **OK**.
15. The **NewColumn** is added to the Europe query.
16. In the **NewColumn** header, click the **double-arrow** icon, clear **Territory**, **Country**, and **Use original column name as prefix**, and then click **OK**.
17. Right-click the **Code** column, point to **Move**, and click **To Beginning**.
18. Right-click the **Code** column, click **Rename**, type **Country Code**, and then press Enter.
19. In the **Close** group, click **Close & Apply**, and then click **Apply**.
20. Close Power BI Desktop.

Results: At the end of this lab, the Europe and North America data will be appended, and the Country Code column will be added to the query.

Module 5: Modeling Data

Lab: Modeling Data

Exercise 1: Create Relationships

▶ Task 1: Preparing the Environment

1. Ensure that the 10989B-MIA-DC and 10989B-MIA-SQL virtual machines are both running, and then log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. In the **D:\Labfiles\Lab05\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**. Type **Y** and press Enter and then wait for the script to finish.
4. Press any key.
5. If you do not have a Power BI login, open Internet Explorer, go to **https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-signing-up-for-power-bi-with-a-new-office-365-trial**, and follow the steps to create an account.
6. In Internet Explorer, go to **https://www.microsoft.com/en-us/download/details.aspx?id=45331**, and then click **Download**.
7. On the **Choose the download you want** page, select the **PBIDesktop_x64.msi** check box, and then click **Next**.
8. In the message box, click **Run**.
9. In the **Microsoft Power BI Desktop (x64) Setup** dialog box, on the **Welcome to the Microsoft Power BI Desktop (x64) Setup Wizard** page, click **Next**.
10. On the **Microsoft Software License Terms** page, select the **I accept the terms in the License Agreement** check box, and then click **Next**.
11. On the **Destination Folder** page, click **Next**.
12. On the **Ready to install Microsoft Power BI Desktop (x64)** page, click **Install**.
13. In the **User Account Control** dialog box, click **Yes**.
14. On the **Completed the Microsoft Power BI Desktop (x64) Setup Wizard** page, clear the **Launch Microsoft Power BI Desktop** check box, and then click **Finish**.
15. Close Internet Explorer.
16. On the desktop, right-click the **Power BI Desktop** shortcut, and then click **Pin to Taskbar**.

▶ Task 2: Automatic Relationships

1. On the taskbar, click **Power BI Desktop**.
2. On the **Get Data** page, click **Get Data**.
3. In the **Get Data** dialog box, click **Excel**, and click **Connect**.
4. In the **Open** dialog box, navigate to **D:\Labfiles\Lab05\Starter\Project**, click **Adventure Works Sales Data.xlsx**, and click **Open**.

5. In the **Navigator** dialog box, select the **DimCurrency**, **DimCustomer**, **DimDate**, **DimProduct**, **DimPromotion**, **DimSalesTerritory**, and **FactInternetSales** check boxes, and then click **Load**.
6. In the **Views** pane, click **Relationships**.
7. On the **Home** tab, click **Manage Relationships**.
8. In the **Manage Relationships** dialog box, click **New**.
9. In the **Create Relationship** dialog box, in the first list, click **FactInternetSales**. When the table preview appears below, click the **OrderDateKey** column.
10. In the second list, click **DimDate**. When the table preview appears below, click the **DateKey** column.
11. Ensure the **Cardinality** is set to **Many to One (*:1)**, the **Cross filter direction** is **Single**, and **Make this relationship active** is selected, and then click **OK**.
12. In the **Manage Relationships** dialog box, click **Close**.
13. On the left hand side, click **Relationships**.
14. In the diagram, in the **FactInternetSales** table, click the **DueDateKey** column. Drag the **DueDateKey** column to the **DateKey** column of the **DimDate** table.
15. In the diagram, in the **FactInternetSales** table, click the **ShipDateKey** column. Drag the **ShipDateKey** column to the **DateKey** column of the **DimDate** table.
16. On the **Home** tab, click **Manage Relationships**.
17. In the **Manage Relationships** dialog box, double-click the **FactInternetSales (CurrencyKey)** relationship.
18. In the **Edit Relationship** dialog box, in the **Cross filter direction** list, click **Single**, and then click **OK**.
19. In the **Manage Relationships** dialog box, double-click the **FactInternetSales (ProductKey)** relationship.
20. In the **Edit Relationship** dialog box, in the **Cross filter direction** list, click **Single**, and then click **OK**.
21. In the **Manage Relationships** dialog box, double-click the **FactInternetSales (PromotionKey)** relationship.
22. In the **Edit Relationship** dialog box, in the **Cross filter direction** list, click **Single**, and then click **OK**.
23. In the **Manage Relationships** dialog box, double-click the **FactInternetSales (SalesTerritoryKey)** relationship.
24. In the **Edit Relationship** dialog box, in the **Cross filter direction** list, click **Single**, and then click **OK**.
25. In the **Manage Relationships** dialog box, click **Close**.
26. Click the relationship line between **FactInternetSales** and **DimCustomer**, and press Delete.
27. On the **Delete Relationship** dialog box, click **Delete**.
28. Click **Manage Relationships**.
29. In the **Manage Relationships** dialog box, click **New**.
30. In the **Create Relationship** dialog box, in the first list, click **FactInternetSales**. When the table preview appears below, click the **CustomerKey** column.
31. In the second list, click **DimCustomer**. When the table preview appears below, click the **CustomerKey** column.
32. In the **Cardinality** list, click **Many to One (*:1)**, and then click **OK**.

33. In the **Manage Relationships** dialog box, click **Close**.
34. Click **Save**, and save the file to the **D:\Labfiles\Lab05\Starter** folder as **Adventure Works Sales.pbix**.
35. Leave Power BI Desktop open for the next exercise.

▶ Task 3: Manual Relationships

1. In Power BI Desktop, on the **Home** tab, click **Get Data**, and then click **Excel**.
2. In the **Open** dialog box, navigate to **D:\Labfiles\Lab05\Starter\Project**, click **Adventure Works Product Categories.xlsx**, and then click **Open**.
3. In the **Navigator** dialog box, select the **DimProductCategory**, and **DimProductSubcategory** check boxes, and then click **Load**.
4. In the **Relationships** pane, look at the relationship that Power BI has created between the two tables.
5. Right-click the relationship line between **DimProductCategory**, and **DimProductSubcategory**, and click **Delete**.
6. In the **Delete Relationship** dialog box, click **Delete**.
7. In the **DimProductSubcategory** table, drag the **CategoryKey** column, to the **CategoryKey** column in the **DimProductCategory** table, to create a **Many to One (*:1)** relationship, and a **Cross filter direction** of **Both**.
8. In the **DimProduct** table, drag the **ProductSubcategoryKey** column, to the **SubcategoryKey** column in the **DimProductSubcategory** table, to create a **Many to One (*:1)** relationship, and a **Cross filter direction** of **Both**.
9. Click **Save**.
10. Leave Power BI Desktop open for the next exercise.

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

▶ Task 1: Add a Calculated Column

1. In Power BI Desktop, on the left hand side, click **Data**.
2. In the **Fields** pane, click **DimCustomer**.
3. On the **Modeling** tab, in the **Calculations** group, click **New Column**.
4. In the formula bar, highlight **Column =**, type the following code, and then press Enter:

```
IncomeStatus = IF (DimCustomer[YearlyIncome] < 25000, "Lower Income",
IF (AND(DimCustomer[YearlyIncome] >= 25000, DimCustomer[YearlyIncome] < 60000),
"Middle Income",
IF (AND(DimCustomer[YearlyIncome] >= 60000, DimCustomer[YearlyIncome] < 100000),
"Higher Income",
IF (DimCustomer[YearlyIncome] >= 100000, "Very High Income", "Other"))))
```

5. In the **Calculations** group, click **New Column**.

6. In the formula bar, highlight **Column =**, type the following code, and then press Enter:

```
DaysSinceFirstPurchase = DATEDIFF(DimCustomer[DateFirstPurchase], TODAY(), DAY)
```

7. In the **Calculations** group, click **New Column**.
8. In the formula bar, highlight **Column =**, type the following code, and then press Enter:

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

9. In the **Calculations** group, click **New Column**.
10. In the formula bar, highlight **Column =**, type the following code, and then press Enter:

```
MaleFemale = IF([Gender] = "M", "Male", "Female")
```

11. In the **Calculations** group, click **New Column**.
12. In the formula bar, highlight **Column =**, type the following code, and then press Enter:

```
Relationship = IF([MaritalStatus] = "M", "Married", "Single")
```

13. In the **Fields** pane, click **DimProductSubcategory**.
14. In the **Calculations** group, click **New Column**.
15. In the formula bar, highlight **Column =**, type the following code, and then press Enter:

```
MainCategory = RELATED(DimProductCategory[CategoryName])
```

16. In the **Fields** pane, click **DimPromotion**.
17. In the **Calculations** group, click **New Column**.
18. In the formula bar, highlight **Column =**, type the following code, and then press Enter:

```
PromotionLengthDays = DATEDIFF(DimPromotion[StartDate], DimPromotion[EndDate], DAY)
```

19. In the **Fields** pane, click **FactInternetSales**.
20. In the **Calculations** group, click **New Column**.
21. In the formula bar, highlight **Column =**, type the following code, and then press Enter:

```
Profit = CURRENCY(FactInternetSales[UnitPrice] -
FactInternetSales[ProductStandardCost])
```

Results: At the end of this lesson, you will have calculated columns added to the tables in your dataset.

Module 6: Interactive Data Visualizations

Lab: Creating a Power BI Report

Exercise 1: Connecting to Power BI Data

▶ Task 1: Prepare the Environment

1. Ensure that the MSL-TMG1, 10989B-MIA-DC and 10989B-MIA-SQL virtual machines are running, and then log on to 10989B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa$$w0rd**.
2. In the **D:\Labfiles\Lab06\Starter** folder, right-click **Setup.cmd**, and then click **Run as administrator**.
3. In the **User Account Control** dialog box, click **Yes**, press **Y** when prompted, wait for the script to finish, and then press Enter to close the command window.
4. If you do not have a Power BI login, open **Internet Explorer**, go to **https://powerbi.microsoft.com/en-us/documentation/powerbi-admin-signing-up-for-power-bi-with-a-new-office-365-trial**, and follow the steps to create an account.
5. In **Internet Explorer**, go to **https://www.microsoft.com/en-us/download/details.aspx?id=45331**, and then click **Download**.
6. On the **Choose the download you want** page, select the **PBIDesktop_x64.msi** check box, and then click **Next**.
7. In the message box, click **Run**.
8. In the **Microsoft Power BI Desktop (x64) Setup** dialog box, on the **Welcome to the Microsoft Power BI Desktop (x64) Setup Wizard** page, click **Next**.
9. On the **Microsoft Software License Terms** page, select the **I accept the terms in the License Agreement** check box, and then click **Next**.
10. On the **Destination Folder** page, click **Next**.
11. On the **Ready to install Microsoft Power BI Desktop (x64)** page, click **Install**.
12. In the **User Account Control** dialog box, click **Yes**.
13. On the **Completed the Microsoft Power BI Desktop (x64) Setup Wizard** page, clear the **Launch Microsoft Power BI Desktop** check box, and then click **Finish**.
14. Close Internet Explorer.

▶ Task 2: Connect to Existing Data in Azure

1. On the taskbar, click **Microsoft SQL Server Management Studio**, and then connect to the **MIA-SQL** database engine instance by using Windows authentication.
2. On the **File** menu, point to **Open**, click **Project/solution**.
3. In the **Open Project** dialog box, browse to the **D:\Labfiles\Lab06\Starter\Project** folder, and then double-click **Project.ssmssln**.
4. In Solution Explorer, expand **Queries**, and then double-click **Lab Exercise 1.sql**.
5. On the Desktop, double-click **Power BI Desktop**.
6. In the **Power BI Desktop** window, click **Get Data**.
7. In the **Get Data** dialog box, click **Microsoft Azure SQL Database**, and then click **Connect**.

8. In the **SQL Server Database** window, in the **Server** box, type the URL of the Azure server ***<Server Name>*.database.windows.net**.
9. In the **Database** box, type **AdventureWorksLT**.
10. Expand the **Advanced options** box.
11. In SQL Server Management Studio, in the **Lab Exercise 1.sql** query, copy the query under **Task 1**.
12. In Power BI Desktop, paste the query into the **SQL Statement (optional)** box, and then click **OK**.
13. If the **Access a SQL Server Database** window appears, click **Database**, and then in the **Username** box, type **Student**, and in the **Password** box, type **Pa$$w0rd**, and then click **Connect**.
14. In the data preview window, click **Load**. The window will close and a blank report canvas will open.
15. In the **Untitled - Power BI Desktop** window, click the **Get Data** drop-down list, and then click **More**.
16. In the **Get Data** dialog box, click **Microsoft Azure SQL Database**, and then click **Connect**.
17. In the **SQL Server Database** window, in the **Server** box, type the URL of the Azure server ***<Server Name>*.database.windows.net**.
18. In the **Database** box, type **AdventureWorksLT**.
19. Expand the **Advanced options** box.
20. In SQL Server Management Studio, in the **Lab Exercise 1.sql** query, copy the query under **Task 2**.
21. In Power BI Desktop, paste the query into the **SQL Statement (optional)** box, and then click **OK**.
22. In the data preview window, click **Load**. The window will close and return to the report.
23. Leave Power BI Desktop open for the next task.

▶ Task 3: Shape Data

1. In the **Fields** pane, right-click **Query1**, click **Rename**, type **Customers**, and then press Enter.
2. Right-click **Query2**, click **Rename**, type **Sales**, and then press Enter.
3. Expand the two tables to display all of the fields.
4. In the left navigation bar, click **Data**, and then select the **Customers** table if it is not already selected.
5. Right-click the **NameStyle** column, and then click **Delete**.
6. In the **Delete Column** dialog box, click **Delete**.
7. Right-click the **SalesPerson** column, and then click **Delete**.
8. In the **Delete Column** dialog box, click **Delete**.
9. Right-click the **CustomerID** column, and then click **Hide in Report View**.
10. Click the **AddressLine1** column to select it.
11. On the **Modeling** tab, in the **Properties** section, click **Data Category: Uncategorized**, and then click **Address**.
12. Click the **City** column to select it.
13. In the **Properties** section, click **Data Category: Uncategorized**, and then click **City**.
14. Click the **StateProvince** column to select it.
15. In the **Properties** section, click **Data Category: Uncategorized**, and then click **State or Province**.
16. Click the **CountryRegion** column to select it.

17. In the **Properties** section, click **Data Category: Uncategorized**, and then click **Country/Region**.
18. Click the **PostalCode** column to select it.
19. In the **Properties** section, click **Data Category: Uncategorized**, and then click **Postal Code**.
20. Click **New Column**, and then in the formula bar, type the following code, and press Enter:

```
FullAddress = Customers[AddressLine1] & ", " & Customers[City] & ", " &
Customers[StateProvince] & ", " & Customers[CountryRegion] & ", " &
Customers[PostalCode]
```

21. In the **Fields** pane, in the **Sales** table, right-click the **RevisionNumber** column, and then click **Delete**.
22. In the **Delete Column** dialog box, click **Delete**.
23. Right-click the **SalesOrderNumber** column, and click **Delete**.
24. In the **Delete Column** dialog box, click **Delete**.
25. Right-click the **CustomerID** column, and then click **Hide in Report View**.
26. Right-click the **SalesOrderID** column, and then click **Hide in Report View**.
27. Right-click the **SalesOrderDetailID** column, and then click **Hide in Report View**.
28. Click **New Column**, and then in the formula bar, type the following code, and press Enter:

```
LineTotal = Sales[OrderQty] * Sales[ListPrice]
```

29. Select the **LineTotal** column, click **Format: General**, point to **Currency**, and then select **$ English (United States)**.
30. Click **New Measure**, and then in the formula bar, type the following code, and press Enter:

```
TargetSales = SUM('Sales'[LineTotal]) * 1.2
```

31. On the **File** menu click **Save**.
32. In the **Save As** dialog box, navigate to **D:\Labfiles\Lab06\Starter**, in the **File name** box, type **AdventureWorksLT Sales**, and then click **Save**.
33. Leave Power BI Desktop open for the next task.

▶ Task 4: Combine Data

1. In File Explorer, browse to the **D:\Labfiles\Lab06\Starter\Project** folder, and then double-click **States.xlsx**.
2. In the **States** worksheet, select all of the values in the two columns, and then press Ctrl+C.
3. Return to Power BI Desktop, and then on the **Home** menu, click **Enter Data**.
4. In the **Create Table** dialog box, click in the table, and then press Ctrl+V. Power BI detects that the first row is a column header.
5. In the **Name** box, type **Sales by State**, and then click **Load**.
6. Click the **Get Data** drop-down menu, click **Web**.
7. In the **From Web** dialog box, in the **URL** box, type the following address, and then click **OK**:

```
http://en.wikipedia.org/wiki/List_of_U.S._state_abbreviations
```

8. In the **Navigator** dialog box, select the **Code and Abbreviations for U.S. states, territories and other regions** check box, and then click **Load**.

9. On the left, click **Data**, if this view is not already showing.
10. In the **Fields** pane, click **Code and Abbreviations for U.S. states, territories and other regions** to display the data. The table has 26 rows at the bottom that are not needed.
11. Click the **Edit Queries** drop-down list, and then click **Edit Queries**.
12. In the Query Editor, click **Code and Abbreviations for U.S. states, territories and other regions** to select the table.
13. Click **Reduce Rows**, click **Remove Rows**, and then click **Remove Bottom Rows**.
14. In the **Remove Bottom Rows** dialog box, in the **Number of rows** box, type **26**, and then click **OK**.
15. Click the **ANSI2** column, and then hold down the Ctrl key while selecting all of the columns to the right. This selects multiple rows.
16. Still holding down Ctrl, click the **Name and status of region2** and **Header** columns to include this in the selection.
17. Click **Manage Columns**, click **Remove Columns**, and then click **Remove Columns**.
18. In the **Query Settings** pane, under **Properties**, in the **Name** box, type **States with Codes**, and then press Enter.
19. Click **Use First Row as Headers**.
20. Right-click the **United States of America** column, click **Rename**, type **State Name**, and then press Enter.
21. Right-click the **US USA 840** column, click **Rename**, type **State Code Long**, and then press Enter.
22. Right-click the **US** column, click **Rename**, type **State Code Short**, and then press Enter.
23. Select the **Sales by State** table, on the **Home** tab, click **Combine**, and then click **Merge Queries**.
24. In the **Merge** dialog box, in the **Sales by State** table, click the **States** column.
25. Select **States with Codes** from the list, select the **State Name** column, and then click **OK**. The new column is added to the table and contains the merged **States with Codes** table.
26. In the column header, click the **Expand** icon, clear **(Select All Columns)**, select **State Code Short**, and then click **OK**. The column now shows just the state codes.
27. Right-click the column, click **Rename**, type **State Code**, and then press Enter.
28. On the **File** menu, click **Close & Apply**.
29. In the **Fields** pane, right-click **States with Codes**, and then click **Hide in Report View**.

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

▶ Task 1: Create a Chart

1. In Power BI Desktop, in the left hand menu, click **Report** to display the file in the report view.
2. In the **Visualizations** pane, click **Gauge**.
3. In the **Sales** table, drag the **LineTotal** field to the **Value** property of the gauge.
4. In the **Sales** table, drag the **TargetSales** measure to the **Target value** property of the gauge.
5. Click **Format**, expand **Gauge axis**, and then in the **Max** box, type **146000**.
6. Expand **Title**, in the **Title Text** box, type **Target Sales**, and then click **Center**.
7. Click the report canvas, and then in the **Customers** table, drag the **CompanyName** field onto the report. Power BI automatically creates a table.
8. In the **Sales** table, drag the **LineTotal** field onto the **CompanyName** table.
9. Make sure that the **CompanyName** table has focus, and then in the **Visualizations** pane, click **Pie chart**.
10. Expand the chart to make all of the company names visible by using the resizer handles on the edge of the chart.
11. With the focus still on the pie chart, in the **Visualizations** pane, click **Format**, and then expand **Title**.
12. In the **Title Text** box, type **Top Selling Customers**, and then click **Center**.
13. In the **Sales** table, drag the **MainCategory** field onto the report canvas. Power BI creates a table.
14. Drag the **OrderQty** field onto the table.
15. In the **Visualizations** pane, click **Stacked bar chart**.
16. In the **Visualizations** pane, click **Fields**.
17. Drag the **OrderQty** field onto the **Color saturation** property. Notice that the colors change.
18. In the **Visualizations** pane, click **Analytics**.
19. Expand **Constant Line**, and click **Add**.
20. In the **Value** box, type **500**.
21. Change **Color** to red, and then toggle **Data label** to **On**.
22. In the **Visualizations** pane, click **Format**.
23. Expand **Title**, in the **Title Text** box, type **Orders by Main Category**, and then click **Center**.
24. Click the report canvas to give it focus, and then in the **Visualizations** pane, click **Donut chart**.
25. In the **Sales** table, select the **MainCategory** and **LineTotal** check boxes.
26. In the **Visualizations** pane, click **Format**.
27. Expand **Title**, in the **Title Text** box, type **Sales by Main Category**, and then click **Center**.
28. In the **Sales** table, drag the **Product** field onto the report canvas. Power BI creates a table.
29. In the **Sales** table, drag the **LineTotal** field onto the products table chart.
30. In the **Sales** table, select the **MainCategory** check box.
31. In the **Visualizations** pane, click **Fields**.
32. In the **Filters** pane, expand **LineTotal(All)**.

33. In the **Show items when the value** list, select **is greater than**, and then in the box below, type **32000**, and then click **Apply filter**.
34. Expand **MainCategory(All)**, and then select the **Bikes** check box.
35. In the **Visualizations** pane, click **Stacked column chart**.
36. In the **Visualizations** pane, click **Format**.
37. Expand **Title**, in the **Title Text** box, type **Top 10 Selling Bikes**, and then click **Center**.
38. In the **Visualizations** pane, click **Analytics**.
39. Expand **Constant Line**, and then click **Add** to create another constant line.
40. In the **Value** box, type **35000**, and then set **Color** to **red**.
41. Toggle **Data label** to **On**, and then set **Color** to **red**.
42. Expand the chart to fill the remaining space on the report canvas. If necessary, move your visuals around to make them fit.
43. Click **Save**.

▶ Task 2: Create a Map Visualization

1. Click the **+** icon at the bottom of the report to add a new page.
2. In the **Fields** pane, in the **Customers** table, select the **City** check box. Power BI adds a map to the report.
3. In the **Fields** pane, in the **Sales** table, select the **LineTotal** check box.
4. Using the grabber tool on the right side of the chart, resize the map to show all of the bubbles.
5. Notice that the bubbles are proportionally sized to represent the data.
6. In the **Visualizations** pane, click **Format**.
7. Expand **Title**, in the **Title Text** box, type **World Sales by City**, and then click **Center**.
8. Click the report canvas, and then in the **Sales by State** table, select the **State Code** check box. Power BI automatically adds a map.
9. In the **Sales by State** table, select the **SalesYTD** check box.
10. In the **Visualizations** pane, click **Filled Map**. Using the grabber tool on the right side and at the bottom of the chart, resize the map to show all the states.
11. Notice that the sales cluster in one area.
12. Position the cursor on **California(CA)** to see the sales figure. The value has not been formatted as currency.
13. In the **Sales by State** table, click **SalesYTD**.
14. On the **Modeling** ribbon, click **Format: General**, point to **Currency**, and then click **$ English (United Stated)**.
15. Position the cursor on **California(CA)** on the map, and notice that the value has been formatted.
16. In the **Visualizations** pane, click **Format**.
17. Expand **Title**, in the **Title Text** box, type **Sales by State**, and then click **Center**.
18. Click **Save**, and then leave the report open for the next exercise.

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

▶ Task 1: Publish Reports from Power BI Desktop

1. On the **Home** ribbon, click **Publish**.
2. In the **Sign in to Power BI** dialog box, click **Sign in**.
3. In the **Sign in to your account** dialog box, enter your username (email address) and password, and then click **Sign in**.
4. If **Select a destination** window appears, check that **My Workspace** is selected, and then click **Publish**.
5. In the **Publishing to Power BI** dialog box, click **Open 'AdventureWorksLT Sales.pbix' in Power BI**.
6. Internet Explorer will open. If you are prompted to sign in, click the account name and enter your password if necessary. If the **Already have an account** page appears, click **Sign in**, type your email address and password in new page, and then click **Sign in**.
7. The report is loaded into the Power BI service.
8. Remain signed in for the next task.

▶ Task 2: Create a Power BI Dashboard

1. In the **AdventureWorksLT Sales** report, click the **Page 1** tab on the bottom of page.
2. Click the **Target Sales** visual, and then click **Pin visual**.
3. In the **Pin to dashboard** dialog box, click **New dashboard**, type **AdventureWorksLT Sales**, and then click **Pin**.
4. On the **Top Selling Customers** visual, click **Pin visual**.
5. In the **Pin to dashboard** dialog box, click **Existing dashboard**, in the list, click **AdventureWorksLT Sales**, and then click **Pin**.
6. On the **Orders by Main Category** visual, click **Pin visual**.
7. In the **Pin to dashboard** dialog box, click **Existing dashboard**, in the list, click **AdventureWorksLT Sales**, and then click **Pin**.
8. On the **Top 10 Selling Bikes** visual, click **Pin visual**.
9. In the **Pin to dashboard** dialog box, click **Existing dashboard**, in the list, click **AdventureWorksLT Sales**, and then click **Pin**.
10. On the **Sales by Main Category** visual, click **Pin visual**.
11. In the **Pin to dashboard** dialog box, click **Existing dashboard**, in the list, click **AdventureWorksLT Sales**, and then click **Pin**.
12. The dashboard is listed in the **My Workspace** pane, under **Dashboards**. Notice the yellow star icon to denote that this is a new dashboard.
13. Click the **AdventureWorksLT Sales** dashboard to open it.
14. Notice that the tiles are all the same size.

15. On the **Target Sales** tile, click **Open menu** (…), and then click **Tile details**.
16. In the **Subtitle** box, type **Sales target for 2016**, and then click **Apply**.
17. On the **Top Selling Customers** tile, click **Open menu** (…), and then click **Tile details**.
18. In the **Subtitle** box, type **Customers selling the most products**, and then click **Apply**.
19. On the **Top 10 Selling Bikes** tile, click **Focus mode**. The tile opens into its own space.
20. Expand the **Filters** pane, expand **LineTotal**, change the value of **32000** to **40000**, and then click **Apply filter**.
21. Next to the report title, **Top 10 Selling Bikes**, click the **Back** button.
22. Click **Enter Full Screen Mode**. Notice that the dashboard displays without any of the browser interface. This is ideal for presentations.
23. Press Esc to exit full-screen mode, and return to the dashboard.
24. Close Internet Explorer.
25. In the Publishing to Power BI window, click **Got it**.

Results: After this exercise, you should have published a report to the Power BI service and used the visuals to create a dashboard.

Exercise 4: (Optional): Adding Visualizations

▶ Task 1: Add Additional Visualizations That Are Relevant to Your Organization

1. Click the **+** icon at the bottom of the report to add a new page.
2. In the **Visualizations** pane, add visuals that you think are relevant to your organization.
3. You can visit **http://aka.ms/Y527x6** to download and add custom visuals.
4. Where possible, test the visuals by using the existing data.

Results: After this exercise, you should have created a report containing visuals that will serve as the basis for a departmental report.