**Study Material**

Bachelor in

Data Science

Subject

BI & Data Analytics

**Faculty**

**Nitish Patil**

School of Data Science

Asian School of Media Studies

1. **COURSE DESCRIPTION**

This course is a semester long, project based curriculum based on Tableau & Power BI in Data Visualization that develops proficient skills in the field of Data Analytics for viasualising the Data sets. Each Data Visualization project has a research and development process from project planning to final outcome as ready for client delivery. Students will gain real world project experience throughout their learning cycle that will help them to better understand the roles and processes in wide range of Data Science careers.

1. **LEARNING OBJECTIVES**

Students will be able to –

* + Identify the application of Data Visualization
  + Recognize the tableau charts & maps
  + Develop Dashboards in Tableau
  + Demonstrate his/her skills with Data Visualizations

1. **LEARNING OUTCOME**

At the end of this course participant will be able to –

1. Identify the Tableau components & solve Tableau based problems.

2. Construct Different charts and graphs to solve data analytics problems.

3. Creation of dashboards for better visualizations.

**Projects**

* 1. Supersotre Sales Data
  2. Netflix Movie Review
  3. Stanford University Data

1. **LEARNING RESOURCE MATERIAL**

**Online References: Tableau**

* 1. [**https://www.tableau.com/**](https://www.tableau.com/)
  2. [**https://public.tableau.com/app/discover**](https://public.tableau.com/app/discover)
  3. [**https://powerbi.microsoft.com/en-us/what-is-business-intelligence/**](https://powerbi.microsoft.com/en-us/what-is-business-intelligence/)
  4. **https://learn.microsoft.com/en-us/power-bi/fundamentals/service-get-started**

**Unit-1: Business Analytics, Data Analytics, and Data Science:**

# **What is Data Visualization?**

Data visualization is a graphical representation of quantitative information and data by using visual elements like graphs, charts, and maps.

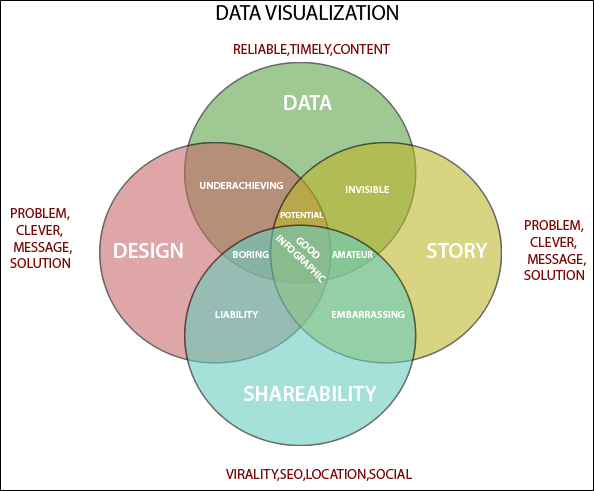
Data visualization convert large and small data sets into visuals, which is easy to understand and process for humans.

Data visualization tools provide accessible ways to understand outliers, patterns, and trends in the data. In the world of Big Data, the data visualization tools and technologies are required to analyze vast amounts of information.

What makes Data Visualization Effective?

Effective data visualization are created by communication, data science, and design collide. Data visualizations did right key insights into complicated data sets into meaningful and natural.

American statistician and Yale professor **Edward Tufte** believe useful data visualizations consist of ?complex ideas communicated with clarity, precision, and efficiency.



Why Use Data Visualization?

1. To make easier in understand and remember.
2. To discover unknown facts, outliers, and trends.
3. To visualize relationships and patterns quickly.
4. To ask a better question and make better decisions.
5. To competitive analyze.
6. To improve insights.

**Principles for data visualization**

The best [data](https://datafloq.com/read/cloud-data-influence-on-transportation/) in the world won’t be worth anything if no one can [understand](https://datafloq.com/read/entity/understanding/) it. The [job](https://datafloq.com/read/big-data-job-growth-trends-watch-2015/) of a [data](https://datafloq.com/read/entity/data-2/) analyst is not only to collect and analyze data, but also to present it to the end users and other interested parties who will then act on that data. Here’s where data [visualization](https://datafloq.com/read/3-fundamentals-using-data-visualizations-tools/) comes in.

Many data analysts are not necessarily experts in data communication or graphic designThis means a lot of them can be lost in the translation of data from the collection to the presentation in the boardroom. I often find myself teaching data [visualization](https://datafloq.com/read/entity/visualization/) classes to more and more data science teams, who recognize this as an area of weakness.

If your [job](https://datafloq.com/read/entity/employees/) entails presenting findings from a set of data or analysis to a [group](https://datafloq.com/read/entity/group-2/) of laymen, then it’s part of your job to present it to them in such a way that it’s easy to understand and therefore take appropriate action.

## ****Keep your****[Audience](https://datafloq.com/read/help-nontechnical-audience-understand-data-science/)****in Mind****

Any [data visualization](https://www.forbes.com/sites/bernardmarr/2017/07/20/the-7-best-data-visualization-tools-in-2017/) should be designed in such a way that it meets the [needs](https://datafloq.com/read/entity/needs/) of the [audience](https://datafloq.com/read/entity/audience/) and their [information](https://datafloq.com/read/why-information-governance-diet-exercise/) needs. As such, you need to determine exactly who is in that audience, and the kind of questions they may need answers to.

## ****Choose the Chart Wisely****

Not all charts are equal, and some will do a better job at displaying certain kinds of [information](https://datafloq.com/read/entity/information/) than others. Check out the following flowchart to help you choose the best type of chart to display your information.

## ****Think Beyond the PowerPoint Templates****

PowerPoint is by far the most popular visualization [tool](https://datafloq.com/read/entity/tools/), but the built-in templates in the program might not be doing your data any favors. Rather than trying to get fancy (yeah, this is directed at 3D pie charts), try to keep your visualizations as simple and uncluttered as possible. If you really want to go for it [Design Bundles](https://designbundles.net/)has a great selection of [tools](https://datafloq.com/read/big-data-startup-review-datasift/) for infographics. These can look spectacular and really make data sing.

## ****Form follows Function****

How will your audience use the data? Consider this and let it determine how you will present the data. Think of your audience as the dashboard of a cockpit, and be sure to only present the most useful, relevant information and in the clearest way possible.

## ****Direct Attention to the Important Details****

As you design your visualizations, be sure to leverage the sensory details like size, color, graphics, and fonts to direct the attention of your audience to the most important pieces of the information.

**Tableau** is an amazing platform for seeing, understanding, and making key decisions based on your data! With it, you will be able to achieve incredible data discovery, data analysis, and data storytelling. You'll accomplish these tasks and goals visually using an interface that is designed for a natural and seamless flow of thought and work.

To leverage the power of Tableau, you don't need to write complex scripts or queries. Instead, you will be interacting with your data in a visual environment where everything that you drag and drop will be translated into the necessary queries for you and then displayed visually. You'll be working in real time, so you will see results immediately, get answers as quickly as you can ask questions, and be able to iterate through potentially dozens of ways to visualize the data to find a key insight or tell a piece of the story.

This chapter introduces the foundational principles of Tableau. We'll go through a series of examples that will introduce the basics of connecting to data, exploring and analyzing the data visually, and finally putting it all together in a fully interactive dashboard. These concepts will be developed far more extensively in subsequent chapters. But don't skip this chapter, as it introduces key terminology and key concepts, including the following:

The cycle of analytics Connecting to data

Foundations for building visualizations Creating bar charts

Creating line charts

Creating geographic visualizations Using Show Me

Bringing everything together via a dashboard

# The cycle of analytics

As someone who works with and seeks to understand data, you will find yourself working within the cycle of analytics. This cycle might be illustrated as follows:

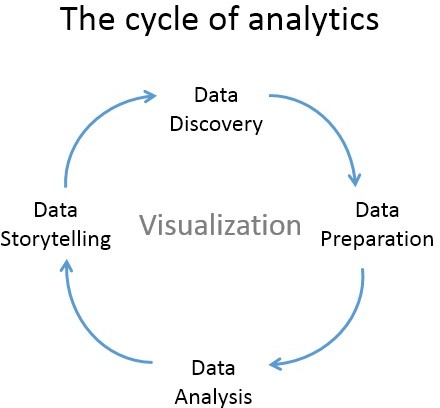


Tableau allows you to jump to any step of the cycle, move freely between steps, and iterate through the cycle very rapidly. With Tableau, you have the ability to do the following:

**Data discovery**: You can very easily explore a dataset using Tableau and begin to understand what data you have visually.

**Data preparation**: Tableau allows you to connect to data from many different sources and, if necessary, create a structure that works best for your analysis. Most of the time, this is as easy as pointing Tableau to a database or opening a file, but Tableau gives you the tools to bring together even complex and messy data from

## Database

A database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a [database management system (DBMS)](https://www.oracle.com/in/database/what-is-database/#WhatIsDBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

Data within the most common types of databases in operation today is typically modeled in rows and columns in a series of tables to make processing and data querying efficient. The data can then be easily accessed, managed, modified, updated, controlled, and organized. Most databases use structured query language (SQL) for writing and querying data.

**What’s the difference between a database and a spreadsheet?**

Databases and spreadsheets (such as Microsoft Excel) are both convenient ways to store information. The primary differences between the two are:

How the data is stored and manipulated

Who can access the data

How much data can be stored

Spreadsheets were originally designed for one user, and their characteristics reflect that. They’re great for a single user or small number of users who don’t need to do a lot of incredibly complicated data manipulation. Databases, on the other hand, are designed to hold much larger collections of organized information—massive amounts, sometimes. Databases allow multiple users at the same time to quickly and securely access and query the data using highly complex logic and language.

## Types of databases

There are many different types of databases. The best database for a specific organization depends on how the organization intends to use the data.

#### Relational databases

* [Relational databases](https://www.oracle.com/in/database/what-is-a-relational-database/) became dominant in the 1980s. Items in a relational database are organized as a set of tables with columns and rows. Relational database technology provides the most efficient and flexible way to access structured information.

#### Object-oriented databases

* Information in an object-oriented database is represented in the form of objects, as in object-oriented programming.

#### Distributed databases

* A distributed database consists of two or more files located in different sites. The database may be stored on multiple computers, located in the same physical location, or scattered over different networks.

#### Data warehouses

* A central repository for data, a data warehouse is a type of database specifically designed for fast query and analysis.

#### NoSQL databases

* A [NoSQL](https://www.oracle.com/in/database/nosql/), or nonrelational database, allows unstructured and semistructured data to be stored and manipulated (in contrast to a relational database, which defines how all data inserted into the database must be composed). NoSQL databases grew popular as web applications became more common and more complex.

#### Graph databases

* A graph database stores data in terms of entities and the relationships between entities.
* **OLTP databases.** An OLTP database is a speedy, analytic database designed for large numbers of transactions performed by multiple users.

These are only a few of the several dozen types of databases in use today. Other, less common databases are tailored to very specific scientific, financial, or other functions. In addition to the different database types, changes in technology development approaches and dramatic advances such as the cloud and automation are propelling databases in entirely new directions. Some of the latest databases include

#### Open source databases

* An open source database system is one whose source code is open source; such databases could be SQL or NoSQL databases.

#### Cloud databases

* A [cloud database](https://www.oracle.com/in/database/what-is-a-cloud-database/) is a collection of data, either structured or unstructured, that resides on a private, public, or hybrid cloud computing platform. There are two types of cloud database models: traditional and database as a service (DBaaS). With DBaaS, administrative tasks and maintenance are performed by a service provider.

#### Multimodel database

* Multimodel databases combine different types of database models into a single, integrated back end. This means they can accommodate various data types.

#### Document/JSON database

* Designed for storing, retrieving, and managing document-oriented information, [document databases](https://www.oracle.com/in/autonomous-database/autonomous-json-database/) are a modern way to store data in JSON format rather than rows and columns.

#### Self-driving databases

* The newest and most groundbreaking type of database, self-driving databases (also known as autonomous databases) are cloud-based and use machine learning to automate database tuning, security, backups, updates, and other routine management tasks traditionally performed by database administrators.

**Unit 2: Structured Query Language**

## What is Structured Query Language (SQL)?

SQL is a programming language used by nearly all [relational databases](https://www.oracle.com/in/database/what-is-database/#relational) to query, manipulate, and define data, and to provide access control. SQL was first developed at IBM in the 1970s with Oracle as a major contributor, which led to implementation of the SQL ANSI standard, SQL has spurred many extensions from companies such as IBM, Oracle, and Microsoft. Although SQL is still widely used today, new programming languages are beginning to appear.

**What are clauses in database?**



Clauses are **in-built functions available to us in SQL**. With the help of clauses, we can deal with data easily stored in the table. Clauses help us filter and analyze data quickly. When we have large amounts of data stored in the database, we use Clauses to query and get data required by the user.

## What is a MySQL database?

[MySQL](https://www.oracle.com/in/mysql/) is an open source relational database management system based on SQL. It was designed and optimized for web applications and can run on any platform. As new and different requirements emerged with the internet, MySQL became the platform of choice for web developers and web-based applications. Because it’s designed to process millions of queries and thousands of transactions, MySQL is a popular choice for ecommerce businesses that need to manage multiple money transfers. On-demand flexibility is the primary feature of MySQL.

MySQL is the DBMS behind some of the top websites and web-based applications in the world, including Airbnb, Uber, LinkedIn, Facebook, Twitter, and YouTube.

**Normalization**

**Normalization** is a database design technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies. Normalization rules divides larger tables into smaller tables and links them using relationships. The purpose of Normalisation in SQL is to eliminate redundant (repetitive) data and ensure data is stored logically.

The inventor of the [relational model](https://www.guru99.com/relational-data-model-dbms.html) Edgar Codd proposed the theory of normalization of data with the introduction of the First Normal Form, and he continued to extend theory with Second and Third Normal Form. Later he joined Raymond F. Boyce to develop the theory of Boyce-Codd Normal Form.

**Create Table**

The CREATE TABLE statement is used to create a new table in a database.

### Syntax

CREATE TABLE table\_name (  
    column1 datatype,  
    column2 datatype,  
    column3 datatype,  
   ....  
);

The column parameters specify the names of the columns of the table.

The datatype parameter specifies the type of data the column can hold (e.g. varchar, integer, date, etc.).

## CREATE TABLE Example

The following example creates a table called "Persons" that contains five columns: PersonID, LastName, FirstName, Address, and City:

### Example

CREATE TABLE Persons (  
    PersonID int,  
    LastName varchar(255),  
    FirstName varchar(255),  
    Address varchar(255),  
    City varchar(255)  
);

## MySQL INSERT INTO SELECT Statement

The INSERT INTO SELECT statement copies data from one table and inserts it into another table.

The INSERT INTO SELECT statement requires that the data types in source and target tables matches.

**Note:** The existing records in the target table are unaffected.

### INSERT INTO SELECT Syntax

Copy all columns from one table to another table:

INSERT INTO table2  
SELECT \* FROM table1WHERE condition;

Copy only some columns from one table into another table:

INSERT INTO table2 (column1, column2, column3, ...)  
SELECT column1, column2, column3, ...  
FROM table1  
WHERE condition;

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CustomerID** | **CustomerName** | **ContactName** | **Address** | **City** | **PostalCode** | **Country** |
| 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 05021 | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 05023 | Mexico |

"Suppliers" table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **SupplierID** | **SupplierName** | **ContactName** | **Address** | **City** | **Postal Code** | **Country** |
| 1 | Exotic Liquid | Charlotte Cooper | 49 Gilbert St. | Londona | EC1 4SD | UK |
| 2 | New Orleans Cajun Delights | Shelley Burke | P.O. Box 78934 | New Orleans | 70117 | USA |
| 3 | Grandma Kelly's Homestead | Regina Murphy | 707 Oxford Rd. | Ann Arbor | 48104 | USA |

## INSERT INTO SELECT Examples

The following SQL statement copies "Suppliers" into "Customers" (the columns that are not filled with data, will contain NULL):

### Example

INSERT INTO Customers (CustomerName, City, Country)  
SELECT SupplierName, City, Country FROM Suppliers;

The following SQL statement copies "Suppliers" into "Customers" (fill all columns):

### Example

INSERT INTO Customers (CustomerName, ContactName, Address, City, PostalCode, Country)  
SELECT SupplierName, ContactName, Address, City, PostalCode, Country FROM Suppliers;

The following SQL statement copies only the German suppliers into "Customers":

### Example

INSERT INTO Customers (CustomerName, City, Country)  
SELECT SupplierName, City, Country FROM Suppliers  
WHERE Country='Germany';

## CREATE DATABASE Statement

The CREATE DATABASE statement is used to create a new SQL database.

### Syntax

CREATE DATABASE databasename;

## CREATE DATABASE Example

The following SQL statement creates a database called "testDB":

### Example

CREATE DATABASE testDB;

## WHERE Clause

The WHERE clause is used to filter records.

It is used to extract only those records that fulfill a specified condition.

### WHERE Syntax

SELECT column1, column2, ...  
FROM table\_name  
WHERE condition;

**Note:** The WHERE clause is not only used in SELECT statements, it is also used in UPDATE, DELETE, etc.!

## Demo Database

Below is a selection from the "Customers" table in the Northwind sample database:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **CustomerID** | **CustomerName** | **ContactName** | **Address** | **City** | **PostalCode** | **Country** |
| 1 | Alfreds Futterkiste | Maria Anders | Obere Str. 57 | Berlin | 12209 | Germany |
| 2 | Ana Trujillo Emparedados y helados | Ana Trujillo | Avda. de la Constitución 2222 | México D.F. | 05021 | Mexico |
| 3 | Antonio Moreno Taquería | Antonio Moreno | Mataderos 2312 | México D.F. | 05023 | Mexico |
| 4 | Around the Horn | Thomas Hardy | 120 Hanover Sq. | London | WA1 1DP | UK |
| 5 | Berglunds snabbköp | Christina Berglund | Berguvsvägen 8 | Luleå | S-958 22 | Sweden |

SQL statement selects all the customers from "Mexico":

### Example

SELECT \* FROM Customers  
WHERE Country = 'Mexico';

SQL requires single quotes around text values (most database systems will also allow double quotes).

However, numeric fields should not be enclosed in quotes:

### Example

SELECT \* FROM Customers  
WHERE CustomerID = 1;

**Logical Operators**

## AND, OR and NOT Operators

The WHERE clause can be combined with AND, OR, and NOT operators.

The AND and OR operators are used to filter records based on more than one condition:

* The AND operator displays a record if all the conditions separated by AND are TRUE.
* The OR operator displays a record if any of the conditions separated by OR is TRUE.

The NOT operator displays a record if the condition(s) is NOT TRUE.

### AND Syntax

SELECT column1, column2, ...  
FROM table\_name  
WHERE condition1 AND condition2 AND condition3 ...;

### OR Syntax

SELECT column1, column2, ...  
FROM table\_name  
WHERE condition1 OR condition2 OR condition3 ...;

### NOT Syntax

SELECT column1, column2, ...  
FROM table\_name  
WHERE NOT condition;

## ORDER BY Keyword

The ORDER BY keyword is used to sort the result-set in ascending or descending order.

The ORDER BY keyword sorts the records in ascending order by default. To sort the records in descending order, use the DESC keyword.

### ORDER BY Syntax

SELECT column1, column2, ...  
FROM table\_name  
ORDER BY column1, column2, ... ASC|DESC

## LIMIT Clause

The LIMIT clause is used to specify the number of records to return.

The LIMIT clause is useful on large tables with thousands of records. Returning a large number of records can impact performance.

### LIMIT Syntax

SELECT column\_name(s)  
FROM table\_nameWHERE condition  
LIMIT number;

**Character Functions**

Single-row character functions accept character data as input and can return both character and

numeric values. Character functions can be divided into the following:

• Case-manipulation functions

• Character-manipulation functions

LOWER(*column|expression*) Converts alpha character values to lowercase

UPPER(*column|expression*) Converts alpha character values to uppercase

CONCAT(*column1|expression1*

*,*

*column2|expression2*)

Concatenates the first character value to the second character

value; equivalent to concatenation operator (||)

SUBSTR(*column|expression,m*

*[,n]*)

Returns specified characters from character value starting at

character position *m, n* characters long (If *m* is negative, the

count starts from the end of the character value. If *n* is

omitted, all characters to the end of the string are returned.)

# **TRIM() Function**

TRIM is a String function of Oracle. This function is used to remove the specified character from head of the string or tail of the string.

## Syntax

1. TRIM( [ [ LEADING | TRAILING | BOTH ] trim\_character **FROM** ] string1 )

## Parameters

**LEADING :** it will trim from head of the string.

**TRAILING:** it will trim from tail of the string.

**BOTH :** it will trim from head as well as from tail of the string

# **LENGTH() Function**

LENGTH is a String function of Oracle. This function returns the size of the given string.

## Syntax

1. LENGTH( string1 )

## Parameters

**string1:** string for getting the length.

## Return

This function returns a numeric value.

## Example 1

## Select length('oracle') from dual;

# **Boolean Expressions in SQL**

Boolean expressions are that expression that returns boolean datatype as result. In SQL there are three values for boolean datatype, those are:

* TRUE
* FALSE
* UNKNOWN

The boolean data type can not be specified during table creation, unlike other data types. Boolean expressions are mainly used with WHERE clauses to filter the data from a table. It can include comparison operators and other operators like ‘AND’ operator, ‘OR’ operator, etc.

# **CONCAT Function**

The CONCAT function in SQL is a String function, which is used to merge two or more strings. The Concat service converts the Null values to an Empty string when we display the result. This function is used to concatenate two strings to make a single string. The**operator**is used to link **character strings**and **column string**.

We can use a **literal in**CONCAT Function. A literal is a **number, character**, or **date** that includes the SELECT statement.

### Syntax of CONCAT function:

1. **SELECT** CONCAT (String 1, String 2, String3.., String N)
2. **FROM** [Source]

### Example-

1. **SELECT** CONCAT (id , **name** , work\_date )
2. ->**FROM** employee\_ tbl;
3. CONCAT(id, **name**, work\_date)

**Stringfunctions:**  
 String Fucntions are used to perform an operation on input string and return an output string.  
Following are the string functions defined in SQL:

1. **ASCII():** This function is used to find the ASCII value of a character.
2. **Syntax:** SELECT ascii('t');

**Output:** 116

1. **CHAR\_LENGTH():** Doesn’t work for SQL Server. Use LEN() for SQL Server. This function is used to find the length of a word.
2. **Syntax:** SELECT char\_length('Hello!');

**Output:** 6

1. **CHARACTER\_LENGTH():** Doesn’t work for SQL Server. Use LEN() for SQL Server. This function is used to find the length of a line.
2. **Syntax:** SELECT CHARACTER\_LENGTH('geeks for geeks');

**Output:** 15

1. **CONCAT():** This function is used to add two words or strings.
2. **Syntax:** SELECT 'Geeks' || ' ' || 'forGeeks' FROM dual;

**Output:** ‘GeeksforGeeks’

1. **CONCAT\_WS():** This function is used to add two words or strings with a symbol as concatenating symbol.
2. **Syntax:** SELECT CONCAT\_WS('\_', 'geeks', 'for', 'geeks');

**Output:** geeks\_for\_geeks

1. **FIND\_IN\_SET():** This function is used to find a symbol from a set of symbols.
2. **Syntax:** SELECT FIND\_IN\_SET('b', 'a, b, c, d, e, f');

**Output:** 2

1. **FORMAT():** This function is used to display a number in the given format.
2. **Syntax:** Format("0.981", "Percent");

**Output:** ‘98.10%’

1. **INSERT():** This function is used to insert the data into a database.
2. **Syntax:** INSERT INTO database (geek\_id, geek\_name) VALUES (5000, 'abc');

**Output:** successfully updated

1. **INSTR():** This function is used to find the occurrence of an alphabet.
2. **Syntax:** INSTR('geeks for geeks', 'e');

**Output:** 2 (the first occurrence of ‘e’)

**Syntax:** INSTR('geeks for geeks', 'e', 1, 2 );

**Output:** 3 (the second occurrence of ‘e’)

1. **LCASE():** This function is used to convert the given string into lower case.
2. **Syntax:** LCASE ("GeeksFor Geeks To Learn");

**Output:** geeksforgeeks to learn

1. **LEFT():** This function is used to SELECT a sub string from the left of given size or characters.
2. **Syntax:** SELECT LEFT('geeksforgeeks.org', 5);

**Output:** geeks

1. **LENGTH():** This function is used to find the length of a word.
2. **Syntax:** LENGTH('GeeksForGeeks');

**Output:** 13

1. **LOCATE():** This function is used to find the nth position of the given word in a string.
2. **Syntax:** SELECT LOCATE('for', 'geeksforgeeks', 1);

**Output:** 6

1. **LOWER():** This function is used to convert the upper case string into lower case.
2. **Syntax:** SELECT LOWER('GEEKSFORGEEKS.ORG');

**Output:** geeksforgeeks.org

1. **LPAD():** This function is used to make the given string of the given size by adding the given symbol.
2. **Syntax:** LPAD('geeks', 8, '0');
3. **Output:**

000geeks

1. **LTRIM():** This function is used to cut the given sub string from the original string.
2. **Syntax:** LTRIM('123123geeks', '123');

**Output:** geeks

1. **MID():** This function is to find a word from the given position and of the given size.
2. **Syntax:** Mid ("geeksforgeeks", 6, 2);

**Output:** for

1. **POSITION():** This function is used to find position of the first occurrence of the given alphabet.
2. **Syntax:** SELECT POSITION('e' IN 'geeksforgeeks');

**Output:** 2

1. **REPEAT():** This function is used to write the given string again and again till the number of times mentioned.
2. **Syntax:** SELECT REPEAT('geeks', 2);

**Output:** geeksgeeks

1. **REPLACE():** This function is used to cut the given string by removing the given sub string.
2. **Syntax:** REPLACE('123geeks123', '123');

**Output:** geeks

1. **REVERSE():** This function is used to reverse a string.
2. **Syntax:** SELECT REVERSE('geeksforgeeks.org');

**Output:** ‘gro.skeegrofskeeg’

1. **RIGHT():** This function is used to SELECT a sub string from the right end of the given size.
2. **Syntax:** SELECT RIGHT('geeksforgeeks.org', 4);

**Output:** ‘.org’

1. **RPAD():** This function is used to make the given string as long as the given size by adding the given symbol on the right.
2. **Syntax:** RPAD('geeks', 8, '0');

**Output:** ‘geeks000’

1. **RTRIM():** This function is used to cut the given sub string from the original string.
2. **Syntax:** RTRIM('geeksxyxzyyy', 'xyz');

**Output:** ‘geeks’

1. **SPACE():** This function is used to write the given number of spaces.
2. **Syntax:** SELECT SPACE(7);

**Output:** ‘ ‘

1. **STRCMP():** This function is used to compare 2 strings.
   * If string1 and string2 are the same, the STRCMP function will return 0.
   * If string1 is smaller than string2, the STRCMP function will return -1.
   * If string1 is larger than string2, the STRCMP function will return 1.
2. **Syntax:** SELECT STRCMP('google.com', 'geeksforgeeks.com');

**Output:** -1

1. **SUBSTR():** This function is used to find a sub string from the a string from the given position.
2. **Syntax:**SUBSTR('geeksforgeeks', 1, 5);

**Output:** ‘geeks’

1. **SUBSTRING():** This function is used to find an alphabet from the mentioned size and the given string.
2. **Syntax:** SELECT SUBSTRING('GeeksForGeeks.org', 9, 1);

**Output:** ‘G’

1. **SUBSTRING\_INDEX():** This function is used to find a sub string before the given symbol.
2. **Syntax:** SELECT SUBSTRING\_INDEX('www.geeksforgeeks.org', '.', 1);

**Output:** ‘www’

1. **TRIM():** This function is used to cut the given symbol from the string.
2. **Syntax:** TRIM(LEADING '0' FROM '000123');

**Output:** 123

1. **UCASE():** This function is used to make the string in upper case.
2. **Syntax:** UCASE ("GeeksForGeeks");
3. **Output:**

GEEKSFORGEEKS

**Group Functions**

• DISTINCT makes the function consider only nonduplicate values; ALL makes it consider every

value including duplicates. The default is ALL and therefore does not need to be specified.

• The data types for the functions with an expr argument may be CHAR, VARCHAR2, NUMBER,

or DATE.

• All group functions ignore null values. To substitute a value for null values, use the NVL, NVL2,

or COALESCE functions.

• The Oracle server implicitly sorts the result set in ascending order when using a GROUP BY

clause. To override this default ordering, DESC can be used in an ORDER BY clause.

**SELECT AVG(salary), MAX(salary),**

**MIN(salary), SUM(salary)**

**FROM employees**

**WHERE job\_id LIKE ’%REP%’;**

AVG, SUM, MIN, and MAX functions against columns that can store numeric data. The

example on the slide displays the average, highest, lowest, and sum of monthly salaries for all sales

representatives.

**SELECT MIN(hire\_date), MAX(hire\_date)**

**FROM employees;**

**COUNT Function**

The COUNT function has three formats:

• COUNT(\*)

• COUNT(*expr*)

• COUNT(DISTINCT *expr*)

COUNT(\*) returns the number of rows in a table that satisfy the criteria of the SELECT statement,

including duplicate rows and rows containing null values in any of the columns. If a WHERE clause is

included in the SELECT statement, COUNT(\*) returns the number of rows that satisfies the condition

in the WHERE clause.

In contrast, COUNT(*expr*) returns the number of non-null values in the column identified by *expr*.

COUNT(DISTINCT *expr*) returns the number of unique, non-null values in the column identified

by *expr*.

**SELECT COUNT(\*)**

**FROM employees**

**WHERE department\_id = 50;**

**Conditional Expressions in SQL**

1. **The CASE Expression**: Let you use IF-THEN-ELSE statements without having to invoke procedures.  
   In a simple CASE expression, the SQL searches for the first WHEN……THEN pair for which expr is equal to comparison\_expr and returns return\_expr. If above condition is not satisfied, an ELSE clause exists, the SQL returns else\_expr. Otherwise, returns NULL.  
   We cannot specify literal null for the return\_expr and the else\_expr. All of the expressions(expr, comparison\_expr, return\_expr) must be of the same data type.  
   **Syntax:**
2. **CASE** expr **WHEN** comparison\_expr1 **THEN** return\_expr1
3. [**WHEN** comparison\_expr2 **THEN** return\_expr2
4. .
5. .
6. .
7. **WHEN** comparison\_exprn **THEN** return\_exprn
8. **ELSE** else\_expr]

**END**

**DECODE Function :** Facilitates conditional inquiries by doing the work of a CASE or IF-THEN-ELSE statement.  
The DECODE function decodes an expression in a way similar to the IF-THEN-ELSE logic used in various languages. The DECODE function decodes expression after comparing it to each search value. If the expression is the same as search, result is returned.  
If the default value is omitted, a null value is returned where a search value does not match any of the result values.

**Tableau** is an amazing platform for seeing, understanding, and making key decisions based on your data! With it, you will be able to achieve incredible data discovery, data analysis, and data storytelling. You'll accomplish these tasks and goals visually using an interface that is designed for a natural and seamless flow of thought and work.

To leverage the power of Tableau, you don't need to write complex scripts or queries. Instead, you will be interacting with your data in a visual environment where everything that you drag and drop will be translated into the necessary queries for you and then displayed visually. You'll be working in real time, so you will see results immediately, get answers as quickly as you can ask questions, and be able to iterate through potentially dozens of ways to visualize the data to find a key insight or tell a piece of the story.

This chapter introduces the foundational principles of Tableau. We'll go through a series of examples that will introduce the basics of connecting to data, exploring and analyzing the data visually, and finally putting it all together in a fully interactive dashboard. These concepts will be developed far more extensively in subsequent chapters. But don't skip this chapter, as it introduces key terminology and key concepts, including the following:

The cycle of analytics Connecting to data

Foundations for building visualizations Creating bar charts

Creating line charts

Creating geographic visualizations Using Show Me

Bringing everything together via a dashboard

# The cycle of analytics

As someone who works with and seeks to understand data, you will find yourself working within the cycle of analytics. This cycle might be illustrated as follows:

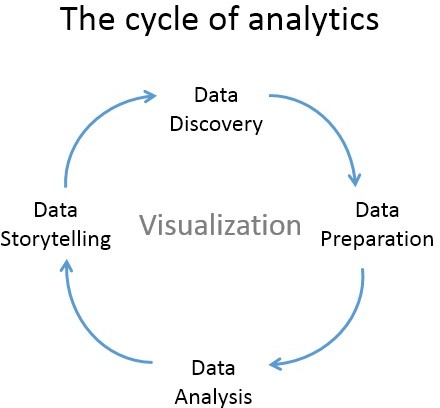


Tableau allows you to jump to any step of the cycle, move freely between steps, and iterate through the cycle very rapidly. With Tableau, you have the ability to do the following:

**Data discovery**: You can very easily explore a dataset using Tableau and begin to understand what data you have visually.

**Data preparation**: Tableau allows you to connect to data from many different sources and, if necessary, create a structure that works best for your analysis. Most of the time, this is as easy as pointing Tableau to a database or opening a file, but Tableau gives you the tools to bring together even complex and messy data from

multiple sources.

**Data analysis**: Tableau makes it easy to visualize the data, so you can see and understand trends, outliers, and relationships. In addition to this, Tableau has an ever-growing set of analytical functions that allow you dive deep into understanding complex relationships, patterns, and correlations in the data.

**Data storytelling**: Tableau allows you to build fully interactive dashboards and stories with your visualizations and insights so that you can share the data story with others.

All of this is done visually. **Data visualization** is the heart of Tableau. You can iterate through countless ways of visualizing the data to ask and answer questions, raise new questions, and gain new insights. And you'll accomplish this as a flow of thought.

# Connecting to data

Tableau connects to data stored in a wide variety of files and databases. This includes flat files, such as Excel documents, spatial files, and text files; relational databases, such as SQL Server and Oracle; cloud-based data sources, such as Google Analytics and Amazon Redshift; and OLAP data sources, such as Microsoft Analysis Services. With very few exceptions, the process of analysis and creating visualizations will be the same, no matter what data source you use.

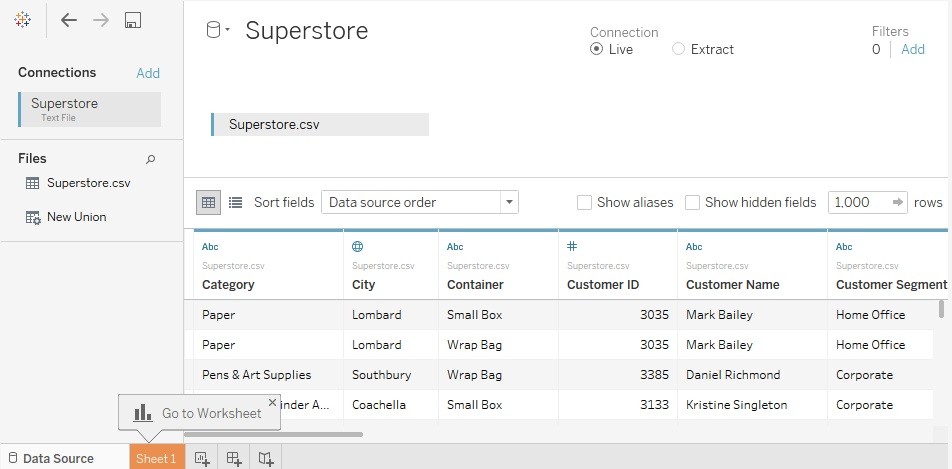
We'll cover details of connecting to different types of data sources in [Chapte r 2, *Working with Data in Tableau*. And we'll cover data spanning a wide](#_bookmark42) variety of industries in other chapters. For now, we'll connect to a text file, specifically, a **comma-separated values** file (.csv). The data is a variation of the sample that ships with Tableau: Superstore, a fictional retail chain that sells various products to customers across the United States. Please use the supplied data file instead of the Tableau sample data, as the variations will lead to differences in visualizations.

The Chapter 1 workbooks, included with the code files bundle, already have connections to the file, but for this example, we'll walk through the steps of creating a connection in a new workbook:

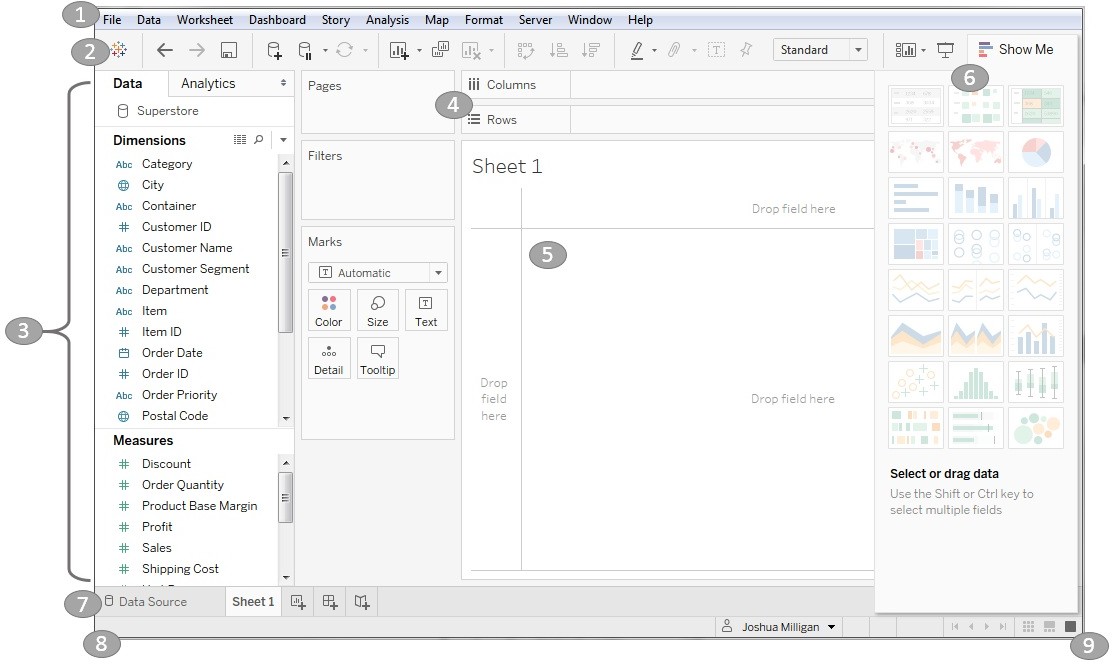
1. Open Tableau. You should see the home screen with a list of connection options on the left and, if applicable, thumbnail previews of recently edited workbooks in the center, along with sample workbooks at the bottom.
2. Under Connect and To a File, click Text File.
3. In the Open dialogue box, navigate to the \Learning Tableau\Chapter

01 directory and select the Superstore.csv file.

1. You will now see the data connection screen, which allows you to visually create connections to data sources. We'll examine the features of this screen in detail in the *Connecting to data* section of [Chapter 2](#_bookmark42), *Working with Data in Tableau*. For now, Tableau has already added and given a preview of the file for the connection:



For this connection, no other configuration is required, so simply click on the Sheet 1 tab at the bottom to start visualizing the data! You should now see the main work area within Tableau, which looks like this:



We'll refer to elements of the interface throughout the book using specific terminology, so take a moment to familiarize yourself with the terms used for various components numbered in the preceding screenshot:

1. The **Menu** contains various menu items for performing a wide range of functions.
2. The **Toolbar** allows for common functions such as undo, redo, save, add a data source, and so on.
3. The **Side Bar** contains tabs for Data and Analytics. When the Data tab is active, we'll refer to the side bar as the data pane. When the Analytics tab is active, we'll refer to the side bar as the analytics pane. We'll go into detail later in this chapter, but for now, note that the data pane shows the data source at the top and contains a list of fields from the data source below, divided into Dimensions and Measures.
4. Various shelves such as Columns, Rows, Pages, and Filters serve as areas to drag and drop fields from the data pane. The **Marks** card contains additional shelves such as Color, Size, Text, Detail, and Tooltip. Tableau will visualize data based on the fields you drop on to the shelves.

*Data fields in the data pane are available to add to a view. Fields that have been dropped on to a shelf are called* **in the view** *or* **active fields** *because they play an active role in the way Tableau draws the visualization.*

1. The **canvas** or **view** is where Tableau will draw the data visualization. You may also drop fields directly on to the view. You'll find the seamless title at the top of the canvas. By default, it will display the name of the sheet, but it can be edited or even

hidden.

1. Show Me is a feature that allows you to quickly iterate through various types of visualizations based on data fields of interest. We'll look at Show Me toward the end of the chapter.
2. The tabs at the bottom of the window give you options for editing the data source, as well as navigating between and adding any number of sheets, dashboards, or stories. Many times, any tab (whether it is a sheet, a dashboard, or a story) is referred to generically as a **sheet**.

*A Tableau workbook is a collection of data sources, sheets, dashboards, and stories. All of this is saved as a single Tableau workbook file (.twb or .twbx). We'll look at the difference in file types and explore details of what else is saved as part of a workbook in later chapters. A workbook is organized into a collection of tabs of various types:*

*A sheet is a single data visualization, such as a bar chart or a line graph. Since Sheet is also a generic term for any tab, we'll often refer to a sheet as a* **view** *because it is a single view of the data.*

*A* **dashboard** *is a presentation of any number of related views and other elements (such as text or images) arranged together as a cohesive whole to communicate a message to an audience. Dashboards are often designed to be interactive.*

*A* **story** *is a collection of dashboards or single views arranged to communicate a narrative from the data. Stories may also be interactive.*

1. As you work, the status bar will display important information and details about the view, selections, and the user.
2. Various controls allow you to navigate between sheets, dashboards, and stories, as well as view the tabs with **Show Filmstrip** or switch to a sheet sorter showing an interactive thumbnail of all sheets in the workbook. Now that you have connected to the data in the text file, we'll explore some examples that lay the foundation for data visualization and then move on to

building some foundational visualization types. To prepare for this, please do the following:

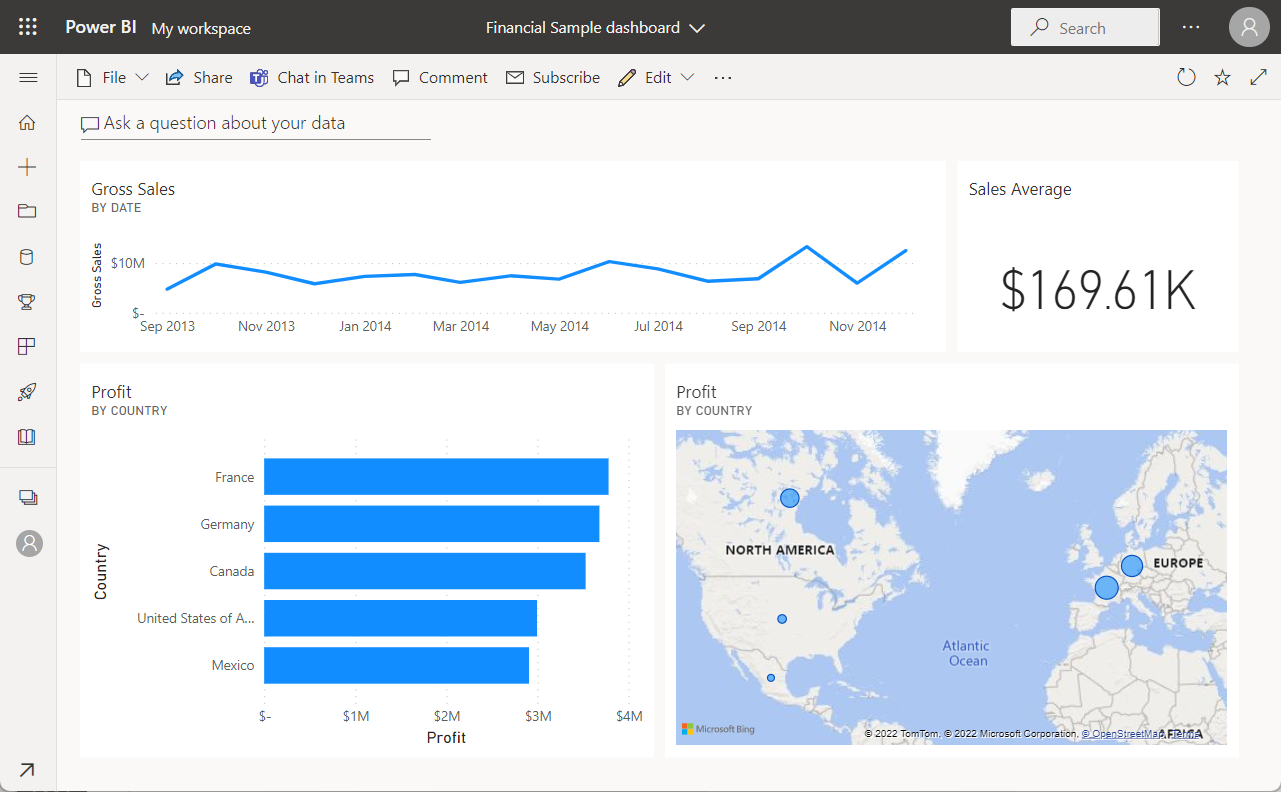
* 1. From the menu, select File | Exit.
  2. When prompted to save changes, select No.
  3. From the \learning Tableau\Chapter 01 directory, open the file Chapter 01 Starter.twbx. This file contains a connection to the Superstore data file and is designed to help you walk through the examples in this chapter.

*The files for each chapter include a Starter workbook that allows you to work through the examples given in this book. If at any time, you'd like to see the completed examples, open the Complete workbook for the chapter.*

With a connection to the data, you are ready to start visualizing and analyzing the data. As you begin to do so, you will take on the role of an analyst at the retail chain. You'll ask questions of the data, build visualizations to answer those questions, and ultimately design a dashboard to share the results. Let's start by laying some foundations for understanding how Tableau visualizes data.

**Unit 3: Data Mining and Business Intelligence**

This tutorial is an introduction to some of the features of the Power BI service. In it, you connect to data, create a report and a dashboard, and ask questions of your data. You can do much more in the Power BI service; this tutorial is just to whet your appetite. For an understanding of how the Power BI service fits in with the other Power BI offerings, we recommend reading [What is Power BI](https://learn.microsoft.com/en-us/power-bi/fundamentals/power-bi-overview).

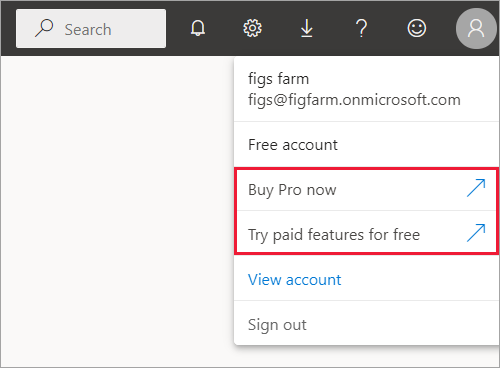


In this tutorial, you complete the following steps:

* Sign in to your Power BI online account, or sign up, if you don't have an account yet.
* Open the Power BI service.
* Get some data and open it in report view.
* Use that data to create visualizations and save it as a report.
* Create a dashboard by pinning tiles from the report.
* Add other visualizations to your dashboard by using the Q&A natural-language tool.
* Resize, rearrange, and edit details for the tiles on the dashboard.
* Clean up resources by deleting the dataset, report, and dashboard.

## Sign up for the Power BI service

You need a Power BI Pro or Premium Per User (PPU) license to create content in Power BI. If you don't have a Power BI account, and plan on creating content, [sign up for a free Power BI Premium Per User 60 day trial](https://app.powerbi.com/signupredirect?pbi_source=web) before you begin. Complete the wizard to get a free license, open the Power BI service (app.powerbi.com), select the **Me** icon and choose either **Buy Pro now** or **Try paid features for free**.



## Step 1: Get data

Often, when you want to create a Power BI report, you start in Power BI Desktop. Power BI Desktop offers more power. You can transform, shape, and model data, before you start designing reports. This time though, we're going to start from scratch creating a report in the Power BI service.

In this tutorial, we get data from a simple Microsoft Excel file. Want to follow along? [Download the Financial Sample file](https://go.microsoft.com/fwlink/?LinkID=521962).

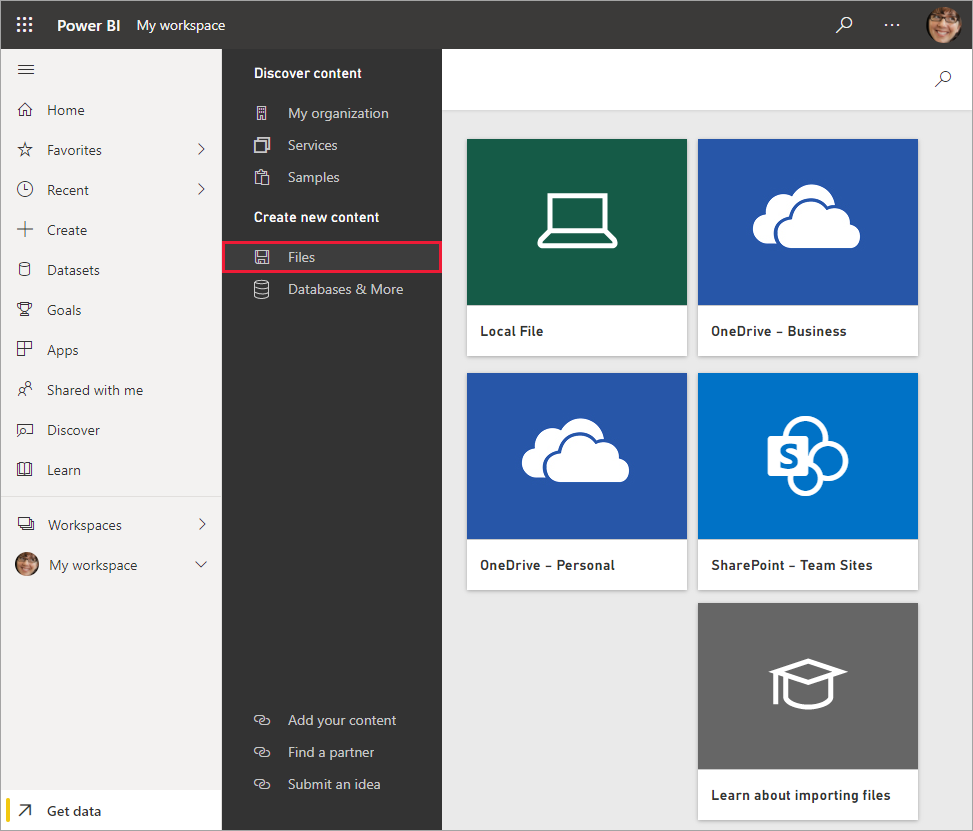
1. To begin, open the Power BI service (app.powerbi.com) in your browser.

Don’t have an account? No worries, you can [sign up for a free Power BI Premium Per User 60 day trial](https://app.powerbi.com/signupredirect?pbi_source=web)

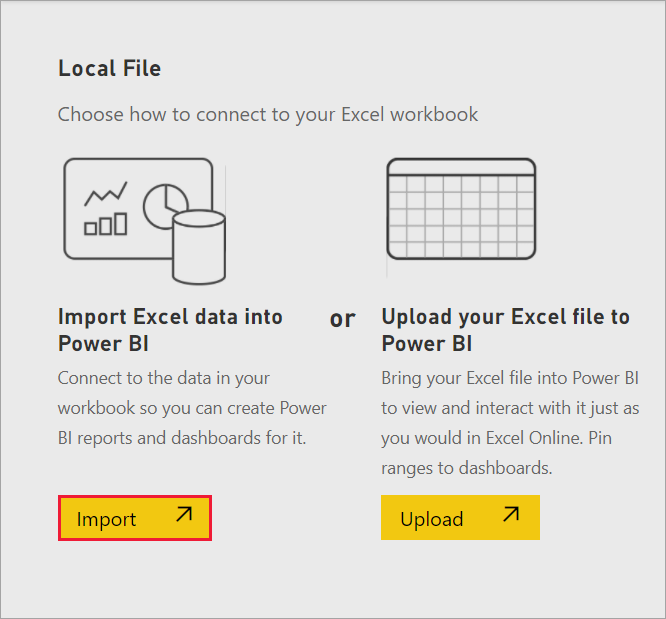
1. Select **My workspace** in the navigation pane.
2. In **My workspace**, select **New** > **Upload a file**.

The **Get Data** page opens.

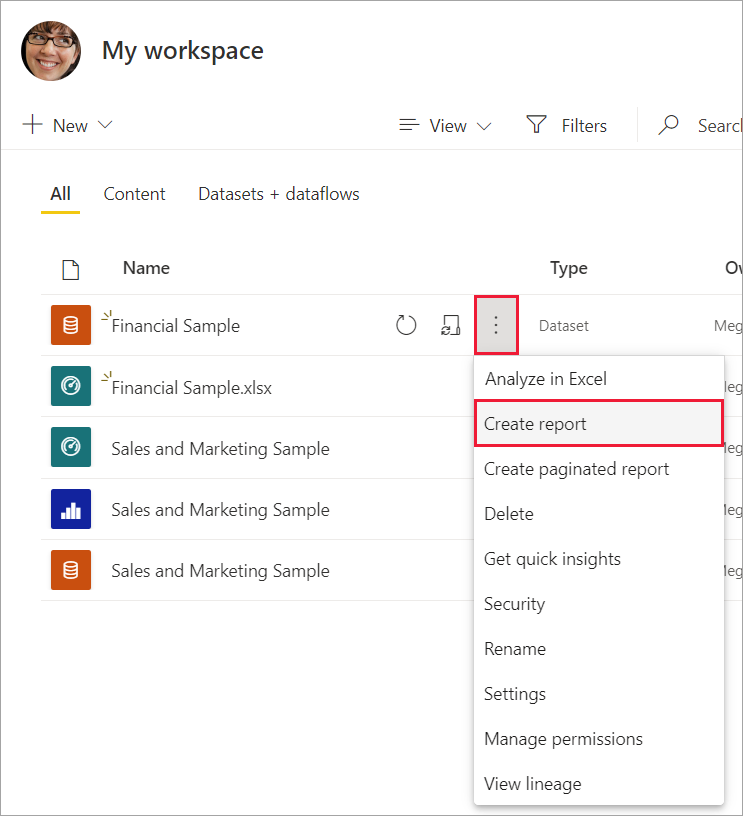
1. Under the **Create new content** section, select **Files** > **Local File**, then select the location where you saved the Excel file.



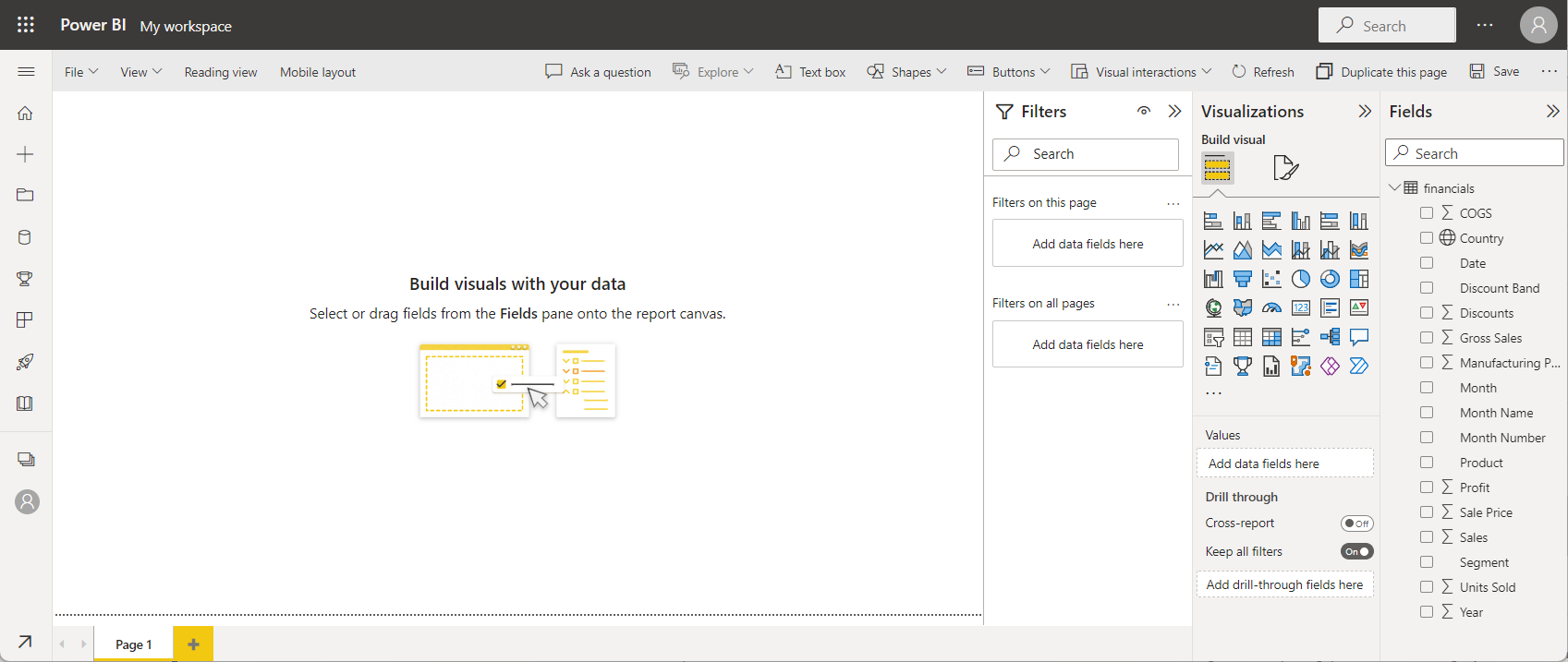
1. Browse to the file on your computer, and choose **Open**.
2. For this tutorial, we select **Import** to add the Excel file as a dataset, which we can then use to create reports and dashboards. If you select **Upload**, the entire Excel workbook is uploaded to Power BI, where you can open and edit it in Excel Online.



1. When your dataset is ready, select **More options (...)** next to your Financial Sample dataset, then select **Create report** to open the report editor.

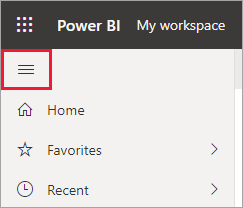


The report canvas is blank. We see the **Filters**, **Visualizations**, and **Fields** panes on the right.

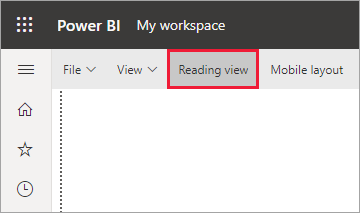


**Tip**

Select the global navigation button in the upper-left corner to collapse the navigation pane. That way your canvas has more room.



1. You're currently in Editing view. Notice the **Reading view** option in the menu bar.

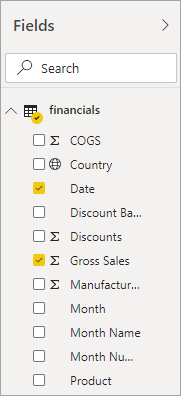


While in Editing view, you can modify reports, because you're the owner and creator of the report. When you share your report with colleagues, often they can only interact with the report in Reading view. They are consumers of reports in your **My workspace**.

## Step 2: Create a chart in a report

Now that you've connected to data, start exploring. When you've found something interesting, you can save it on the report canvas. Then you can pin it to a dashboard to monitor it and see how it changes over time. But first things first

1. In the report editor, start in the **Fields** pane on the right side of the page to build a visualization. Select the **Gross Sales** field, then the **Date** field.

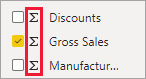


Power BI analyzes the data and creates a column chart visualization.

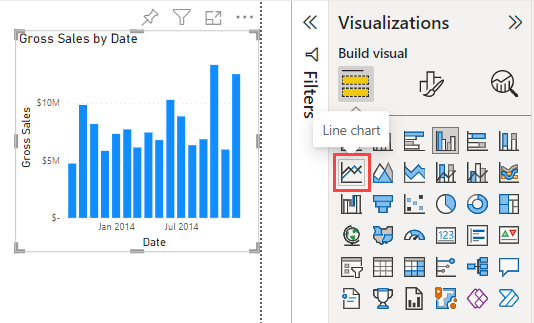
**Note**

If you selected the **Date** field first instead of **Gross Sales**, you see a table. No worries! We're going to change the visualization in the next step.

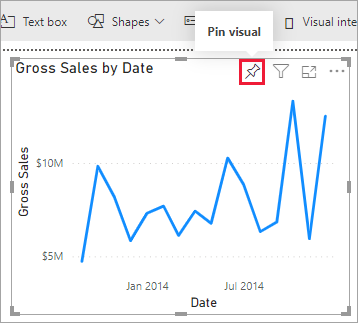
Some fields have sigma symbols next to them because Power BI detected that they contain numeric values.



1. Let's switch to a different way of displaying this data. Line charts are good visuals for displaying values over time. Select the **Line chart** icon from the **Visualizations** pane.



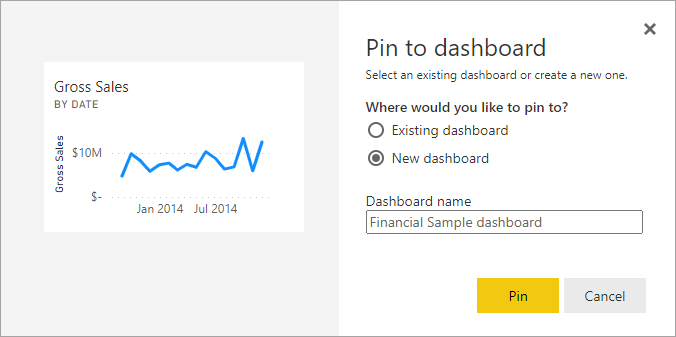
1. This chart looks interesting, so let's pin it to a dashboard. Hover over the visualization and select the pin icon that appears either above or below it.



1. Because this report is new, you're prompted to save it before you can pin a visualization to a dashboard. Give your report a name (for example, Financial Sample report), then **Save**.

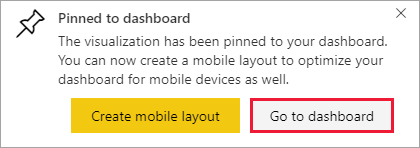
Now you're looking at the report in Reading view.

1. Select the **Pin** icon again.
2. Select **New dashboard** and name it Financial Sample dashboard, for example.

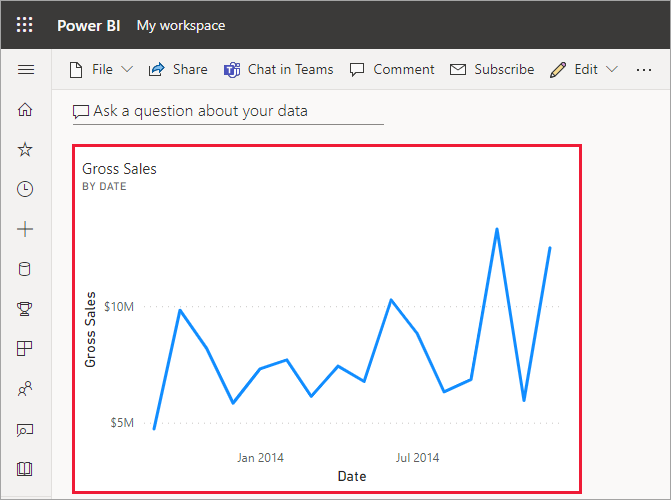


A success message (near the top-right corner) lets you know the visualization was added as a tile to your dashboard.

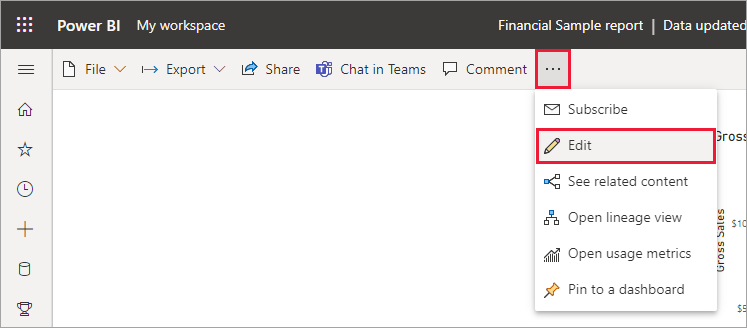
1. Select **Go to dashboard** to see your new dashboard with the line chart that you pinned to it as a tile.



Now that you've pinned this visualization, it's stored on your dashboard. The data stays up-to-date so you can track the latest value at a glance. However, if you change the visualization type in the report, the visualization on the dashboard doesn't change.



1. Select the new tile on your dashboard. Power BI returns you to the report in Reading view.
2. To switch back to Editing view, select **More options** (...) in the menu bar > **Edit**.

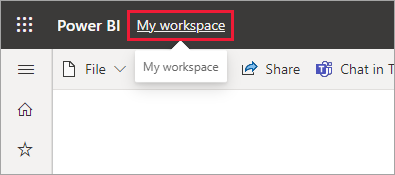


Back in Editing view, you can continue to explore and pin tiles.

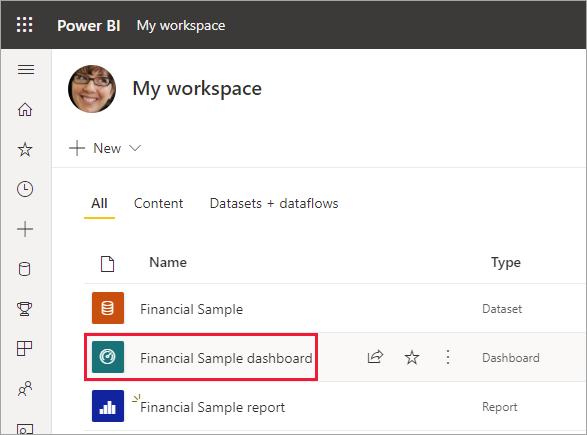
## Step 3: Explore with Q&A

For a quick exploration of your data, try asking a question in the Q&A question box. Q&A lets you ask natural-language queries about your data. In a dashboard, the Q&A box is at the top (**Ask a question about your data**) under the menu bar. In a report, it's in the top menu bar (**Ask a question**).

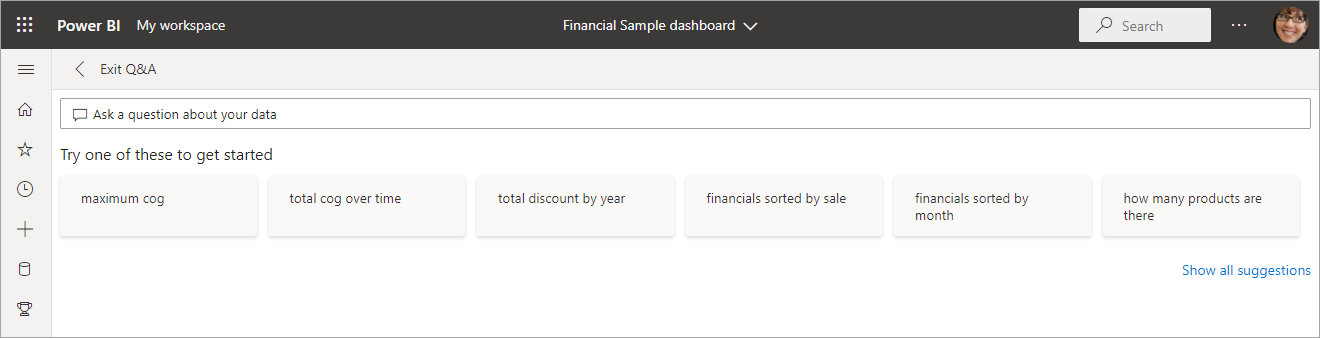
1. To go back to the dashboard, select **My workspace** in the black **Power BI** header bar.



1. In **My workspace**, select your dashboard.



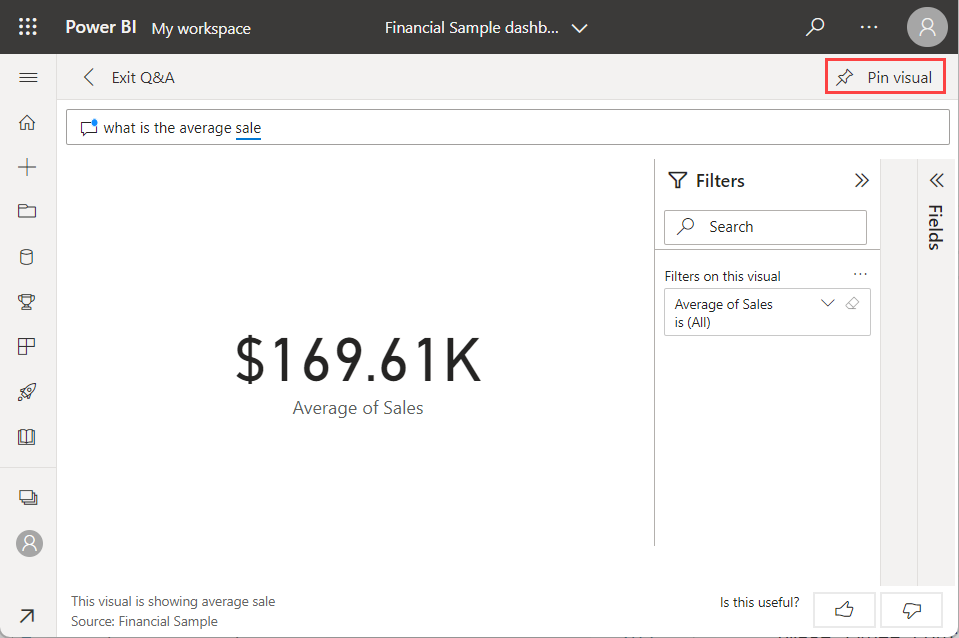
1. Select **Ask a question about your data**. Q&A automatically offers a number of suggestions.



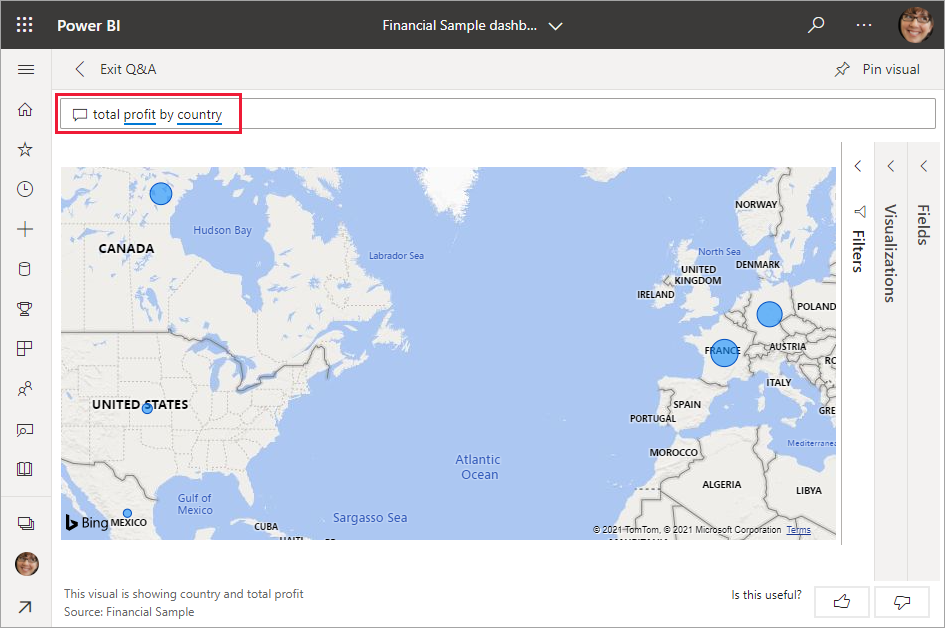
1. Some suggestions return a single value. For example, select **what is the average sale**.

Q&A searches for an answer and presents it in the form of a card visualization.

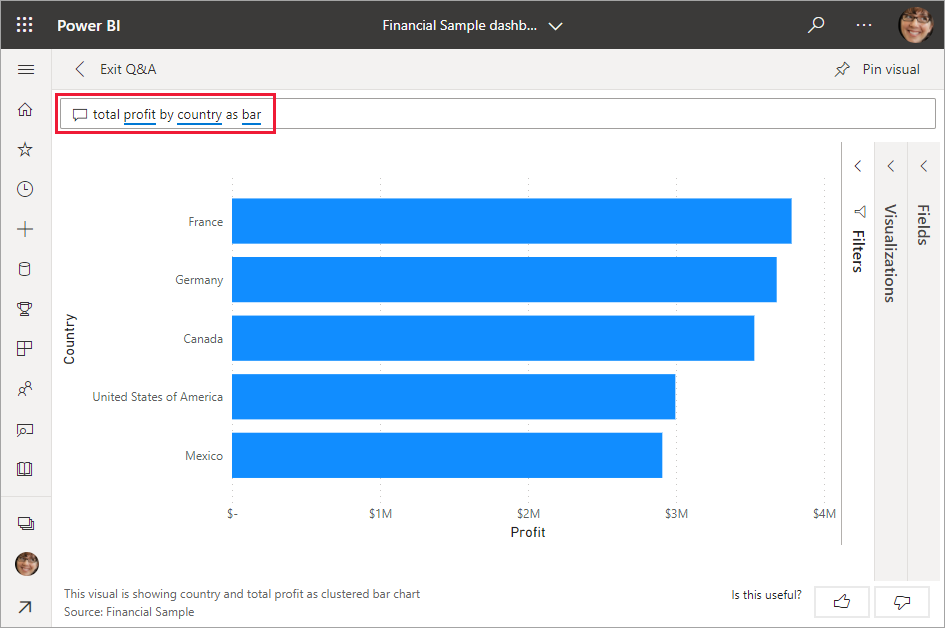
1. Select **Pin visual** and pin this visualization to the Financial Sample dashboard.



1. Go back to Q&A and type total profit by country.

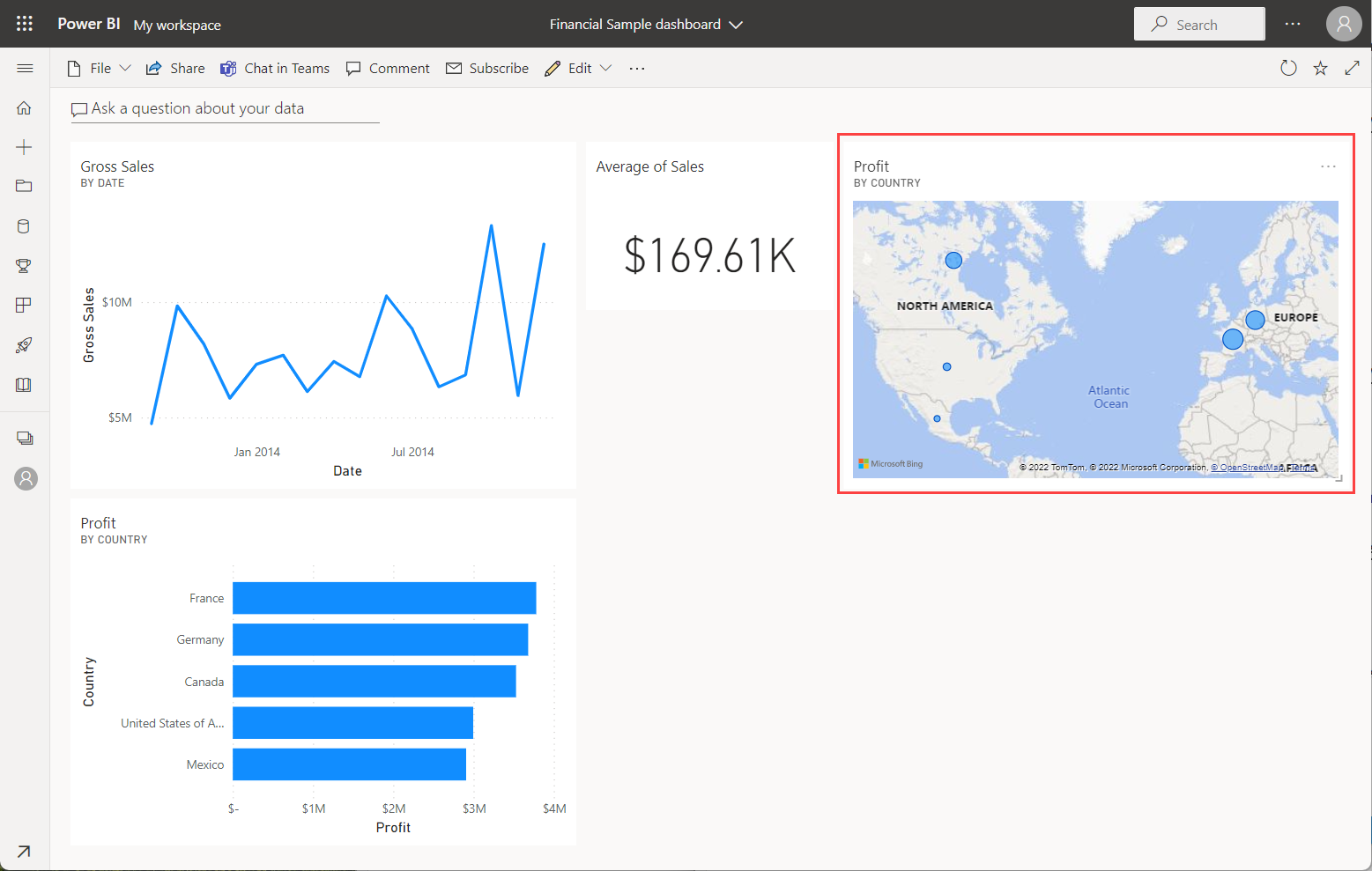


1. Pin the map to the Financial Sample dashboard, too.
2. On the dashboard, select the map you just pinned. See how it opens Q&A again?
3. Place the cursor after by country in the Q&A box and type as bar. Power BI creates a bar chart with the results.



1. Pin the bar chart to your Financial Sample dashboard, too.
2. Select **Exit Q&A** to return to your dashboard, where you see the new tiles you created.

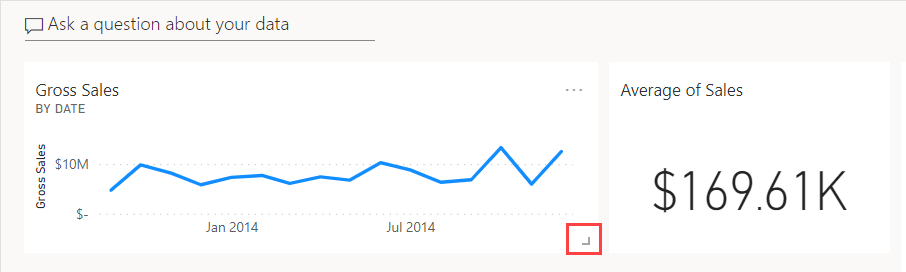
You see that even though you changed the map to a bar chart in Q&A, that tile remained a map because it was a map when you pinned it.



## Step 4: Reposition tiles

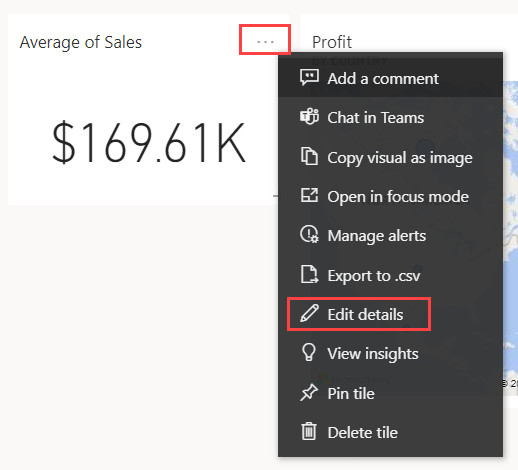
We can rearrange the tiles to make better use of the dashboard space.

1. Drag the lower-right corner of the Gross Sales line chart tile upward, until it snaps at the same height as the Average of Sales tile, then release it.

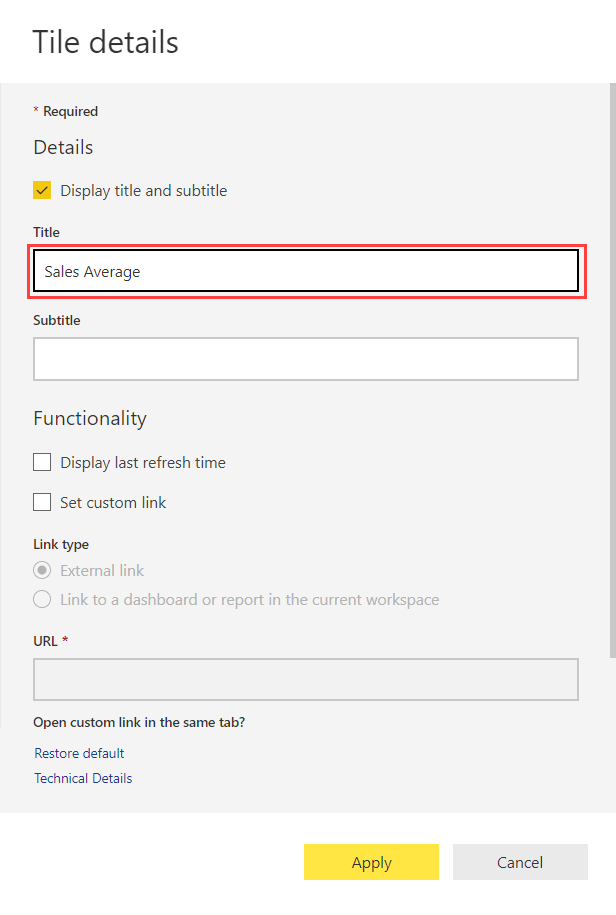


Now the two tiles are the same height.

1. Select **More options (...)** for the Average of Sales tile > **Edit details**.

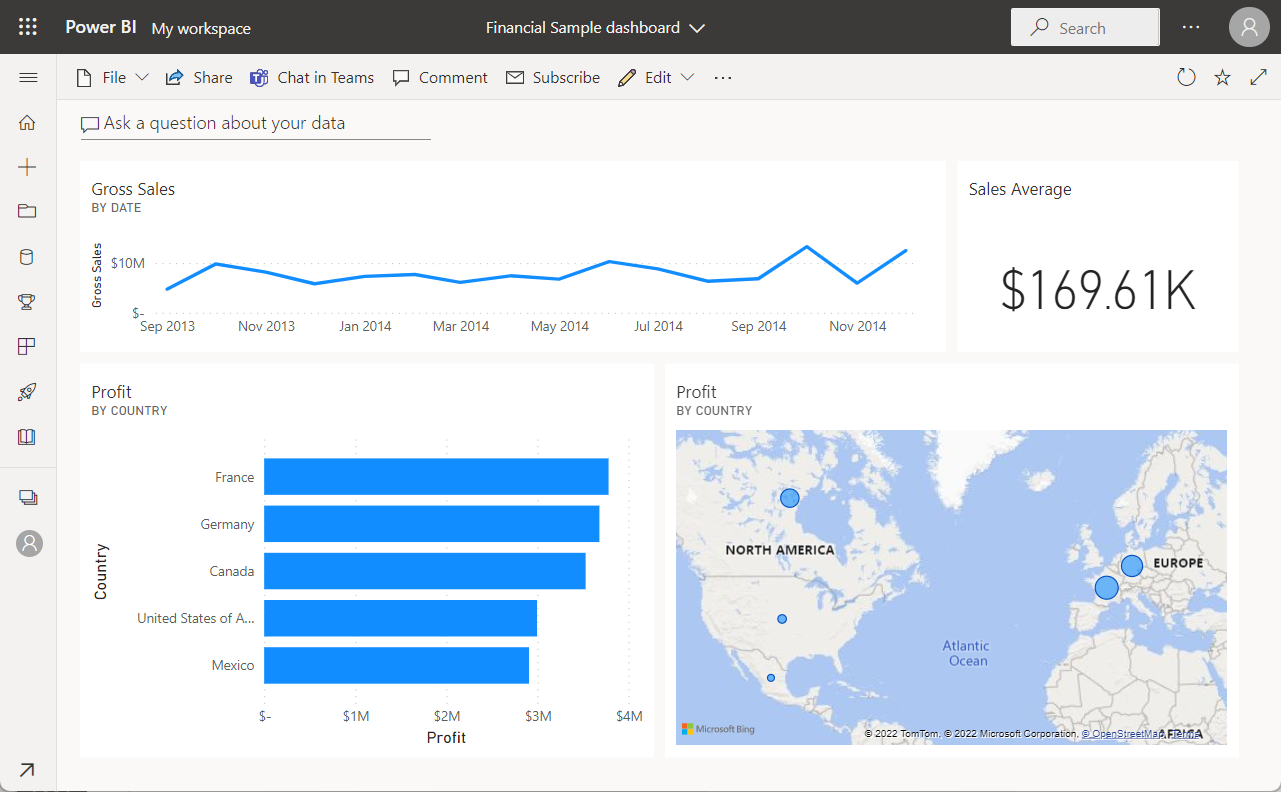


1. In the **Title** box, type Sales Average > **Apply**.



1. Rearrange the other visuals to fit together.

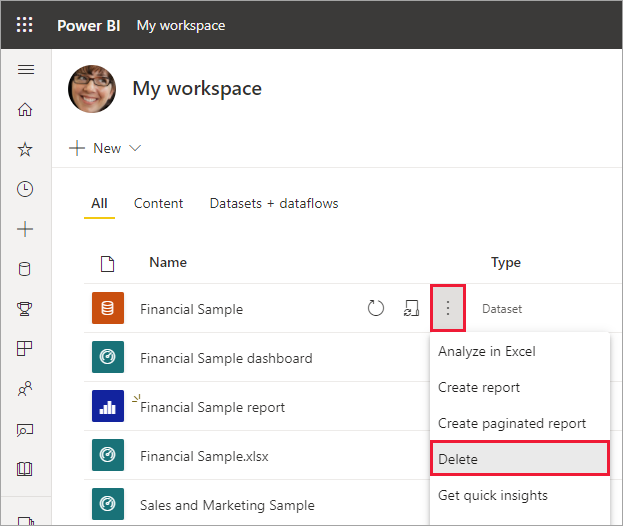
That looks better.



## Clean up resources

Now that you've finished the tutorial, you can delete the dataset, report, and dashboard.

1. Select **My workspace** in the black **Power BI** header bar.
2. Select **More options (...)** next to the Financial Sample dataset > **Delete**.



You see a warning that **All reports and dashboard tiles containing data from this dataset will also be deleted**.

1. Select **Delete**.

**Unit 4: SQL Server Analysis and Services (SSAS):**

Analysis Services is an analytical data engine (VertiPaq) used in decision support and business analytics. It provides enterprise-grade semantic data models for business reports and client applications such as Power BI, Excel, Reporting Services reports, and other data visualization tools.

Installed as an on-premises server instance, SQL Server Analysis Services supports tabular models at all compatibility levels (depending on version), multidimensional models, data mining, and Power Pivot for SharePoint.

**SQL Server Analysis Services workflow**

A typical implementation workflow includes installing a SQL Server Analysis Services instance, creating a tabular or multidimensional data model, deploying the model as a database to a server instance, processing the database to load it with data, and then assigning permissions to allow data access. When ready to go, the data model can be accessed by any client application supporting Analysis Services as a data source.

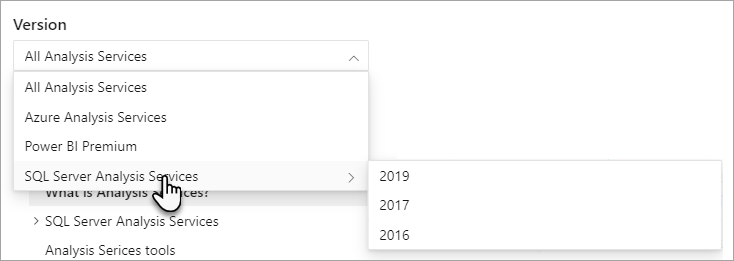
To create a model, use Visual Studio with Analysis Services projects extension, also known as SQL Server Data Tools or simply SSDT, choosing either a Tabular or Multidimensional project template. The project template contains folders for all of the objects needed in a model. You can use wizards to create all of the basic elements, such as data sources, data source views, dimensions, cubes, and roles. Visual Studio and DevOps supports efficient CI/CD pipelines.

Models are populated with data from external data systems, usually data warehouses hosted on a SQL Server or Oracle relational database engine (Tabular models support additional data source types). Models specify query objects, such as cubes, but also specify dimensions that can be used in multiple cubes, calculations and KPIs that encapsulate business logic, and interactions such as navigation and drill-through behaviors.

To use a model, it's deployed to a server instance that runs databases in a particular server mode, making the data available to authorized users who connect through Excel or other applications.

**Documentation**

Analysis Services documentation is located in different areas in Microsoft Learn, depending on the platform or version you are using. All SQL Server Analysis Services documentation is accessible by using the Table of Contents to the left. To see only those articles relevant to a particular version, use the **Version** selector above the ToC.



Documentation you see in the ToC to the left is known as the core Analysis Services documentation. Core documentation can apply to just one platform, like SQL Server Analysis Services, or to all Analysis Services platforms including Azure Analysis Services and Analysis Services in Power BI. This is because how you create and deploy a tabular model, or manage certain server properties or databases is much the same, regardless of platform.

In the core documentation, at the top of each article an **APPLIES TO** banner indicates the platforms and versions the article applies to. Keep in mind, feature and functionality changes are happening to each platform all the time. When they do, we make every effort to update the documentation.

## SQL Server 2022 Analysis Services

### Cumulative Update 1 (CU1)

#### Encryption upgrade

This update includes enhancement to the schema write operation encryption algorithm. This enhancement may require you to upgrade tabular and multidimensional model databases to ensure proper encryption. To learn more, see [Upgrade encryption](https://learn.microsoft.com/en-us/analysis-services/instances/encryption-upgrade?view=sql-analysis-services-2022&preserve-view=true).

### Generally Available (GA)

#### Horizontal fusion

This version introduces Horizontal Fusion, a query execution plan optimization aimed at reducing the number of data source queries required to generate and return results. Multiple smaller data source queries are fused together into a larger data source query. Fewer data source queries mean fewer round trips and fewer expensive scans over large data sources, which results in sizeable DAX performance gains and reduced processing demand at the data source. DAX queries run faster with Horizontal Fusion, especially in DirectQuery mode. In addition, scalability also increases.

#### Parallel Execution Plans for DirectQuery

This improvement enables the Analysis Services engine to analyze DAX queries against a DirectQuery data source and identify independent storage engine operations. The engine can then execute those operations against the data source in parallel. By executing operations in parallel, the Analysis Services engine can improve query performance by taking advantage of scalability large data sources may be able to provide. To ensure query processing does not overburden your data source, use the [MaxParallelism](https://learn.microsoft.com/en-us/analysis-services/tabular-models/partitions-ssas-tabular?view=asallproducts-allversions#maxparallelism) property setting to specify a fixed number of threads that can be used for parallel operations.

#### Support for Power BI DirectQuery datasets

This version introduces support for Power BI datasets with DirectQuery connections to SQL Server 2022 Analysis Services models. Data modelers and report authors using the May 2022 and later versions of Power BI Desktop can now combine other imported and DirectQuery data from Power BI datasets, Azure Analysis Services, and now SSAS 2022.

To learn more, see [Using DirectQuery for datasets and Analysis Services | Power BI Documentation](https://learn.microsoft.com/en-us/power-bi/connect-data/desktop-directquery-datasets-azure-analysis-services).

#### MDX query performance

First introduced in Power BI and now in SSAS 2022, MDX Fusion includes Formula Engine (FE) optimization reducing the number of Storage Engine (SE) queries per MDX query. Client applications that use Multidimensional Expressions (MDX) to query model/dataset data such as Microsoft Excel will see improved query performance. Common MDX query patterns now require fewer SE queries where previously numerous SE queries were necessary to support different granularity. Fewer SE queries mean fewer expensive scans over large models, which results in significant performance gains, especially when connecting to a tabular models in Direct Query mode.

To learn more, see [Announcing improved MDX query performance in Power BI | Microsoft Power BI Blog](https://powerbi.microsoft.com/blog/announcing-improved-mdx-query-performance-in-power-bi/).

#### Resource governance

This version includes improved accuracy for the QueryMemoryLimit server memory property and DbpropMsmdRequestMemoryLimit connection string property.

First introduced in SSAS 2019, the [QueryMemoryLimit](https://learn.microsoft.com/en-us/analysis-services/server-properties/memory-properties?view=asallproducts-allversions#querymemorylimit) server memory property applied only to memory spools where intermediate DAX query results are created during query processing. Now in SSAS 2022, it also applies to MDX queries, effectively covering all queries. You can better control process expensive queries that result in significant materialization. If the query hits the limit specified, the engine cancels the query and returns an error to the caller, reducing impact on other concurrent users.

Client applications can further reduce the memory allowed per query by specifying the [DbpropMsmdRequestMemoryLimit](https://learn.microsoft.com/en-us/analysis-services/instances/connection-string-properties-analysis-services?view=asallproducts-allversions#dbpropmsmdrequestmemorylimit) connection string property. Specified in Kilobytes, this property overrides the QueryMemoryLimit server memory property value for a connection.

#### Query interleaving - Short query bias with fast cancellation

This version introduces a new value that specifies Short query bias with fast cancellation for the Threadpool\SchedulingBehavior property setting. This property setting improves user query response times in high-concurrency scenarios. To learn more, see [Query interleaving - Configure](https://learn.microsoft.com/en-us/analysis-services/tabular-models/query-interleaving?view=asallproducts-allversions#configure).

#### Tabular model 1600 compatibility level

This version introduces the 1600 [compatibility level](https://learn.microsoft.com/en-us/analysis-services/tabular-models/compatibility-level-for-tabular-models-in-analysis-services?view=asallproducts-allversions) for tabular models. The 1600 compatibility level coincides with the latest functionality in Power BI and Azure Analysis Services.

### Deprecated features in SSAS 2022

There are no [deprecated](https://learn.microsoft.com/en-us/analysis-services/what-s-new-in-sql-server-analysis-services?view=asallproducts-allversions#definitions) features announced with this version.

### Discontinued features in SSAS 2022

The following features are [discontinued](https://learn.microsoft.com/en-us/analysis-services/what-s-new-in-sql-server-analysis-services?view=asallproducts-allversions#definitions) in this version:

| **Mode/Category** | **Feature** |
| --- | --- |
| Tabular | 1100 and 1103 [Compatibility levels](https://learn.microsoft.com/en-us/analysis-services/tabular-models/compatibility-level-for-tabular-models-in-analysis-services?view=asallproducts-allversions) |
| Multidimensional | [Data Mining](https://learn.microsoft.com/en-us/analysis-services/data-mining/data-mining-ssas?view=asallproducts-allversions) |
| Power Pivot mode | [Power Pivot for SharePoint](https://learn.microsoft.com/en-us/analysis-services/power-pivot-for-sharepoint-ssas?view=asallproducts-allversions) |

### Breaking changes in SSAS 2022

Tabular model 1100 and 1103 compatibility levels are discontinued in this version. To prevent a [breaking change](https://learn.microsoft.com/en-us/analysis-services/what-s-new-in-sql-server-analysis-services?view=asallproducts-allversions#definitions), upgrade models to the 1200 compatibility level prior to upgrading an earlier SSAS version to SSAS 2022.

### Behavior changes in SSAS 2022

There are no [behavior changes](https://learn.microsoft.com/en-us/analysis-services/what-s-new-in-sql-server-analysis-services?view=asallproducts-allversions#definitions) in this version.

## SQL Server 2019 Analysis Services

### SQL Server 2019 Analysis Services CU 5

SQL Server Analysis Services cumulative updates are included with SQL Server cumulative updates. To learn more about and download the latest cumulative update, see [SQL Server 2019 latest cumulative update](https://www.microsoft.com/download/details.aspx?id=100809). Cumulative update KB pages summarize known issues, improvements, and fixes for all SQL Server features, including SSAS. Additional details for major feature updates for SSAS are described here.

#### SuperDAX for multidimensional models (SuperDAXMD)

With CU5, DAX-based clients can now use SuperDAX functions and query patterns against multidimensional models, providing improved performance when querying model data. SuperDAX first introduced DAX query optimizations for tabular models with Power BI and SQL Server Analysis Services 2016. SuperDAXMD now brings these improvements to multidimensional models.

A separate announcement on the [Power BI blog](https://powerbi.microsoft.com/blog/improving-power-bi-performance-when-querying-multidimensional-models/) highlights how Power BI users can benefit from this multidimensional model performance improvement by downloading the latest version of Power BI Desktop. Existing interactive reports in the Power BI Service can benefit without any additional steps, as Power BI generates the optimized SuperDAX queries automatically. Power BI automatically detects connections to multidimensional models with SuperDAX support and uses the same optimized DAX functions and query patterns that it already uses against tabular models. While Power BI can automatically switch to SuperDAXMD, in your own business intelligence solutions, you might have to optimize DAX query patterns manually.

Optimized query patterns should use [SUMMARIZECOLUMNS](https://learn.microsoft.com/en-us/dax/summarizecolumns-function-dax) function to replace the less efficient standard [SUMMARIZE](https://learn.microsoft.com/en-us/dax/summarize-function-dax) function. Use DAX variables, [VAR](https://learn.microsoft.com/en-us/dax/var-dax), to calculate expressions only once at the place of definition, and then reuse the results in any other DAX expressions without having to perform the calculation again. Other, and perhaps less common SuperDAX functions are [SUBSTITUTEWITHINDEX](https://learn.microsoft.com/en-us/dax/substitutewithindex-function-dax), [ADDMISSINGITEMS](https://learn.microsoft.com/en-us/dax/addmissingitems-function-dax), as well as [NATURALLEFTOUTERJOIN](https://learn.microsoft.com/en-us/dax/naturalleftouterjoin-function-dax) and [NATURALINNERJOIN](https://learn.microsoft.com/en-us/dax/naturalinnerjoin-function-dax), [ISONORAFTER](https://learn.microsoft.com/en-us/dax/isonorafter-function-dax), and [GROUPBY](https://learn.microsoft.com/en-us/dax/groupby-function-dax). [SELECTCOLUMNS](https://learn.microsoft.com/en-us/dax/selectcolumns-function-dax) and [UNION](https://learn.microsoft.com/en-us/dax/union-function-dax) are also SuperDAX functions.

To learn more about how DAX works with multidimensional models, and important patterns and constraints to be aware of, be sure to see [DAX for multidimensional models](https://learn.microsoft.com/en-us/analysis-services/multidimensional-models/dax-for-multidimensional-models?view=asallproducts-allversions).

### SQL Server 2019 Analysis Services GA (Generally Available)

#### Tabular model compatibility level

This release introduces the 1500 [compatibility level](https://learn.microsoft.com/en-us/analysis-services/tabular-models/compatibility-level-for-tabular-models-in-analysis-services?view=asallproducts-allversions) for tabular models.

#### Query interleaving

Query interleaving is a tabular mode system configuration that can improve user query response times in high-concurrency scenarios. Query interleaving with short query bias allows concurrent queries to share CPU resources. To learn more, see [Query interleaving](https://learn.microsoft.com/en-us/analysis-services/tabular-models/query-interleaving?view=asallproducts-allversions).

#### Calculation groups in tabular models

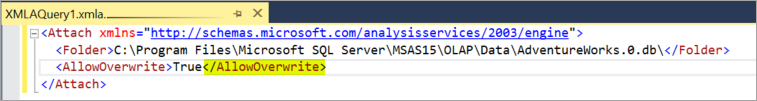
Calculation groups can significantly reduce the number of redundant measures by grouping common measure expressions as calculation items. Calculation groups are shown in reporting clients as a table with a single column. Each value in the column represents a reusable calculation, or calculation item, that can be applied to any of the measures. A calculation group can have any number of calculation items. Each calculation item is defined by a DAX expression. To learn more, see [Calculation groups](https://learn.microsoft.com/en-us/analysis-services/tabular-models/calculation-groups?view=asallproducts-allversions).

#### Governance setting for Power BI cache refreshes

The **ClientCacheRefreshPolicy** property setting is now supported in SSAS 2019 and later. This property setting is already available for Azure Analysis Services. The Power BI service caches dashboard tile data and report data for initial load of Live Connect report, causing an excessive number of cache queries being submitted to the engine, and in extreme cases overload the server. The **ClientCacheRefreshPolicy** property allows you to override this behavior at the server level. To learn more, see [General Properties](https://learn.microsoft.com/en-us/analysis-services/server-properties/general-properties?view=asallproducts-allversions).

#### Online attach

This feature provides the ability to attach a tabular model as an online operation. Online attach can be used for synchronization of read-only replicas in on-premises query scale-out environments. To perform an online-attach operation, use the **AllowOverwrite** option of the Attach XMLA command.



This operation may require double the model memory to keep the old version online while loading the new version.

A typical usage pattern could be as follows:

1. DB1 (version 1) is already attached on read-only server B.
2. DB1 (version 2) is processed on the write server A.
3. DB1 (version 2) is detached and placed on a location accessible to server B (either via a shared location, or using robocopy, etc.).
4. The <Attach> command with AllowOverwrite=True is executed on server B with the new location of DB1 (version 2).

Without this feature, admins are first required to detach the database and then attach the new version of the database. This leads to downtime when the database is unavailable to users, and queries against it will fail.

When this new flag is specified, version 1 of the database is deleted atomically within the same transaction with no downtime. However, it comes at the cost of having both databases loaded into memory simultaneously.

#### Many-to-many relationships in tabular models

This improvement allows many-to-many relationships between tables where both columns are non-unique. A relationship can be defined between a dimension and fact table at a granularity higher than the key column of the dimension. This avoids having to normalize dimension tables and can improve the user experience because the resulting model has a smaller number of tables with logically grouped columns.

Many-to-many relationships require models be at the 1500 and higher compatibility level. You can create many-to-many relationships by using Visual Studio 2019 with Analysis Services projects VSIX update 2.9.2 and higher, the Tabular Object Model (TOM) API, Tabular Model Scripting Language (TMSL), and the open-source Tabular Editor tool.

#### Memory settings for resource governance

The following property settings provide improved resource governance:

* **Memory\QueryMemoryLimit** - This memory property can be used to limit memory spools built by DAX queries submitted to the model.
* **DbpropMsmdRequestMemoryLimit** - This XMLA property can be used to override the Memory\QueryMemoryLimit server property value for a connection.
* **OLAP\Query\RowsetSerializationLimit** - This server property limits the number of rows returned in a rowset, protecting server resources from extensive data export usage. This property applies to both applies to both DAX and MDX queries.

These properties can be set by using the latest version of SQL Server Management Studio (SSMS). These settings are already available for Azure Analysis Services.

#### Deprecated features in SSAS 2019

There are no [deprecated](https://learn.microsoft.com/en-us/analysis-services/what-s-new-in-sql-server-analysis-services?view=asallproducts-allversions#definitions) features announced with this release.

#### Discontinued features in SSAS 2019

There are no [discontinued](https://learn.microsoft.com/en-us/analysis-services/what-s-new-in-sql-server-analysis-services?view=asallproducts-allversions#definitions) features announced with this release.

#### Breaking changes in SSAS 2019

There are no [breaking changes](https://learn.microsoft.com/en-us/analysis-services/what-s-new-in-sql-server-analysis-services?view=asallproducts-allversions#definitions) in this release.

#### Behavior changes in SSAS 2019

There are no [behavior changes](https://learn.microsoft.com/en-us/analysis-services/what-s-new-in-sql-server-analysis-services?view=asallproducts-allversions#definitions) in this release.

## SQL Server 2017 Analysis Services

SQL Server 2017 Analysis Services see some of the most important enhancements since SQL Server 2012. Building on the success of Tabular mode (first introduced in SQL Server 2012 Analysis Services), this release makes tabular models more powerful than ever.

Multidimensional mode and Power Pivot for SharePoint mode are a staple for many Analysis Services deployments. In the Analysis Services product lifecycle, these modes are mature. There are no new features for either of these modes in this release. However, bug fixes and performance improvements are included.

The features described here are included in SQL Server 2017 Analysis Services. But in order to take advantage of them, you must also use the latest versions of Visual Studio with Analysis Services projects and SQL Server Management Studio (SSMS). Analysis Services projects and SSMS are updated monthly with new and improved features that typically coincide with new functionality in SQL Server.

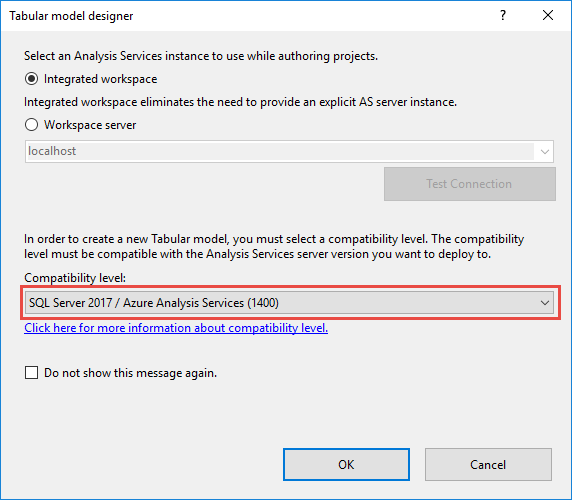
While it's important to learn about all the new features, it's also important to know what is being deprecated and discontinued in this release and future releases. To learn more, see [Deprecated features in SSAS 2017](https://learn.microsoft.com/en-us/analysis-services/what-s-new-in-sql-server-analysis-services?view=asallproducts-allversions#deprecated-features-in-ssas-2017).

Let's take a look at some of the key new features in this release.

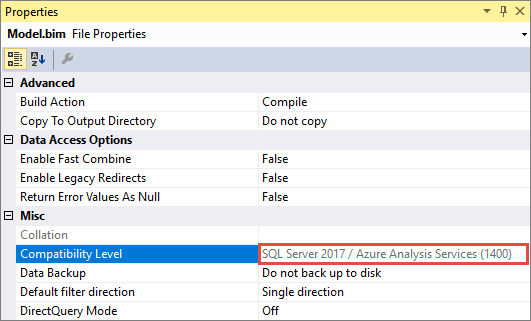
### 1400 Compatibility level for tabular models

To take advantage of many of the new features and functionality described here, new or existing tabular models must be set or upgraded to the 1400 compatibility level. Models at the 1400 compatibility level cannot be deployed to SQL Server 2016 SP1 or earlier, or downgraded to lower compatibility levels. To learn more, see [Compatibility level for Analysis Services tabular models](https://learn.microsoft.com/en-us/analysis-services/tabular-models/compatibility-level-for-tabular-models-in-analysis-services?view=asallproducts-allversions).

In Visual Studio, you can select the new 1400 compatibility level when creating new tabular model projects.



To upgrade an existing tabular model in Visual Studio, in Solution Explorer, right-click **Model.bim**, and then in **Properties**, set the **Compatibility Level** property to **SQL Server 2017 (1400)**.

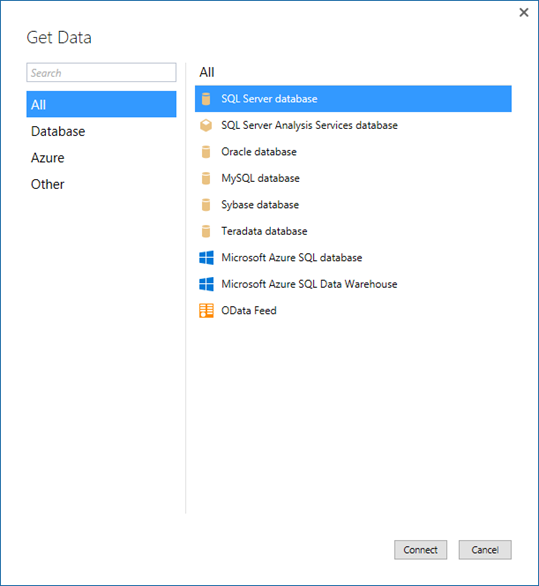


It's important to keep in mind, once you upgrade an existing model to 1400, you can't downgrade. Be sure to keep a backup of your 1200 model database.

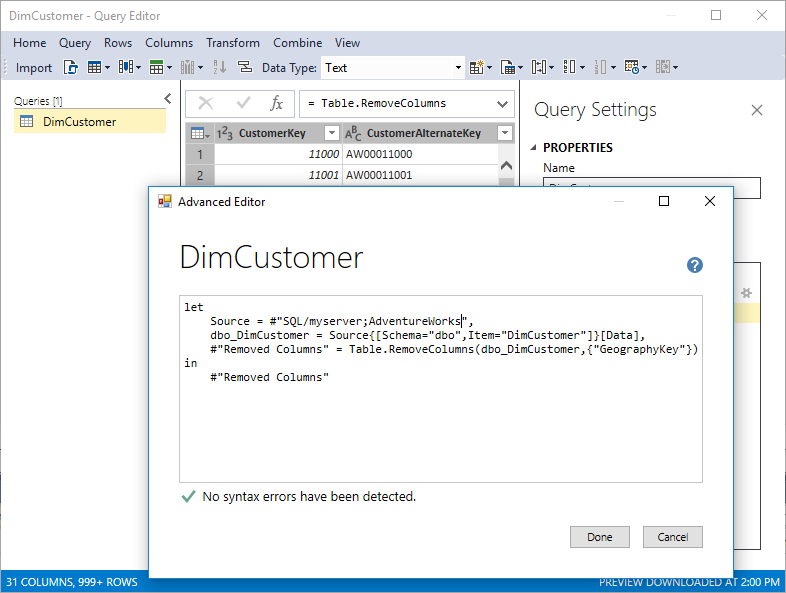
### Modern Get Data experience

When it comes to importing data from data sources into your tabular models, SSDT introduces the modern **Get Data** experience for models at the 1400 compatibility level. This new feature is based on similar functionality in Power BI Desktop and Microsoft Excel 2016. The modern Get Data experience provides immense data transformation and data mashup capabilities by using the Get Data query builder and M expressions.

The modern Get Data experience provides support for a wide range of data sources. Going forward, updates will include support for even more.



A powerful and intuitive user interface makes selecting your data and data transformation/mashup capabilities easier than ever.



The modern Get Data experience and M mashup capabilities do not apply to existing tabular models upgraded from the 1200 compatibility level to 1400. The new experience only applies to new models created at the 1400 compatibility level.

### Encoding hints

This release introduces encoding hints, an advanced feature used to optimize processing (data refresh) of large in-memory tabular models. To better understand encoding, see [Performance Tuning of Tabular Models in SQL Server 2012 Analysis Services](https://learn.microsoft.com/en-us/previous-versions/sql/sql-server-2012/dn393915(v=msdn.10)) whitepaper to better understand encoding.

* Value encoding provides better query performance for columns that are typically only used for aggregations.
* Hash encoding is preferred for group-by columns (often dimension-table values) and foreign keys. String columns are always hash encoded.

Numeric columns can use either of these encoding methods. When Analysis Services starts processing a table, if either the table is empty (with or without partitions) or a full-table processing operation is being performed, samples values are taken for each numeric column to determine whether to apply value or hash encoding. By default, value encoding is chosen when the sample of distinct values in the column is large enough - otherwise hash encoding usually provides better compression. It is possible for Analysis Services to change the encoding method after the column is partially processed based on further information about the data distribution, and restart the encoding process; however, this increases processing time and is inefficient. The performance-tuning whitepaper discusses re-encoding in more detail and describes how to detect it using SQL Server Profiler.

Encoding hints allow the modeler to specify a preference for the encoding method given prior knowledge from data profiling and/or in response to re-encoding trace events. Since aggregation over hash-encoded columns is slower than over value-encoded columns, value encoding may be specified as a hint for such columns. It is not guaranteed that the preference is applied. It is a hint as opposed to a setting. To specify an encoding hint, set the EncodingHint property on the column. Possible values are "Default", "Value" and "Hash". The following snippet of JSON-based metadata from the Model.bim file specifies value encoding for the Sales Amount column.

JSONCopy

{

"name": "Sales Amount",

"dataType": "decimal",

"sourceColumn": "SalesAmount",

"formatString": "\\$#,0.00;(\\$#,0.00);\\$#,0.00",

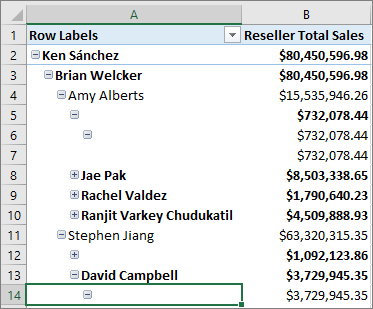
"sourceProviderType": "Currency",

"encodingHint": "Value"

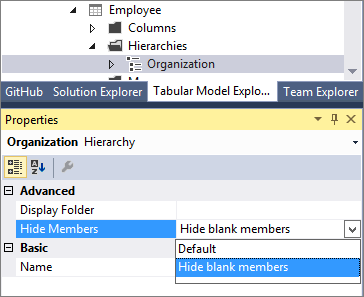
}

### Ragged hierarchies

In tabular models, you can model parent-child hierarchies. Hierarchies with a differing number of levels are often referred to as ragged hierarchies. By default, ragged hierarchies are displayed with blanks for levels below the lowest child. Here's an example of a ragged hierarchy in an organizational chart:



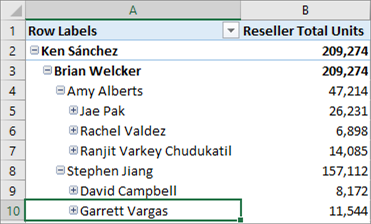
This release introduces the **Hide Members** property. You can set the **Hide Members** property for a hierarchy to **Hide blank members**.



**Note**

Blank members in the model are represented by a DAX blank value, not an empty string.

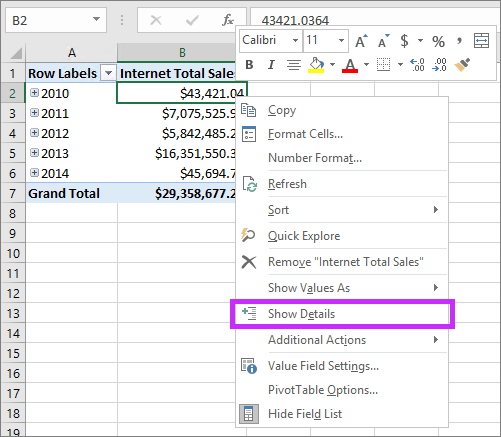
When set to **Hide blank members**, and the model deployed, an easier to read version of the hierarchy is shown in reporting clients like Excel.



### Detail Rows

You can now define a custom row set contributing to a measure value. Detail Rows is similar to the default drillthrough action in multidimensional models. This allows end-users to view information in more detail than the aggregated level.

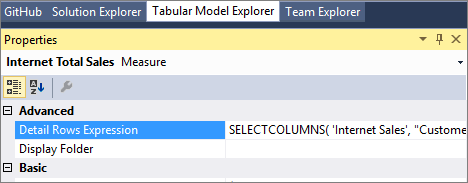
The following PivotTable shows Internet Total Sales by year from the Adventure Works sample tabular model. You can right-click a cell with an aggregated value from the measure and then click **Show Details** to view the detail rows.



By default, the associated data in the Internet Sales table is displayed. This limited behavior is often not meaningful for the user because the table may not have the necessary columns to show useful information such as customer name and order information. With Detail Rows, you can specify a **Detail Rows Expression** property for measures.

#### Detail Rows Expression property for measures

The **Detail Rows Expression** property for measures allows model authors to customize the columns and rows returned to the end-user.



The [SELECTCOLUMNS](https://learn.microsoft.com/en-us/dax/selectcolumns-function-dax) DAX function is commonly used in a Detail Rows Expression. The following example defines the columns to be returned for rows in the Internet Sales table in the sample Adventure Works tabular model:

DAXCopy

SELECTCOLUMNS(

'Internet Sales',

"Customer First Name", RELATED( Customer[Last Name]),

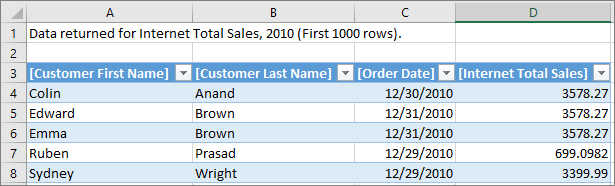
"Customer Last Name", RELATED( Customer[First Name]),

"Order Date", 'Internet Sales'[Order Date],

"Internet Total Sales", [Internet Total Sales]

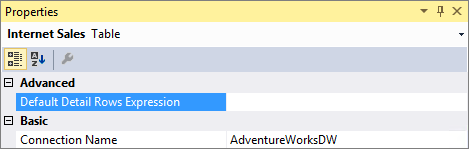
)

With the property defined and the model deployed, a custom row set is returned when the user selects **Show Details**. It automatically honors the filter context of the cell that was selected. In this example, only the rows for 2010 value are displayed:



#### Default Detail Rows Expression property for tables

In addition to measures, tables also have a property to define a detail rows expression. The **Default Detail Rows Expression** property acts as the default for all measures within the table. Measures that do not have their own expression defined inherits the expression from the table and show the row set defined for the table. This allows reuse of expressions, and new measures added to the table later automatically inherits the expression.



#### DETAILROWS DAX Function

Included in this release is a new DETAILROWS DAX function that returns the row set defined by the detail rows expression. It works similarly to the DRILLTHROUGH statement in MDX, which is also compatible with detail rows expressions defined in tabular models.

The following DAX query returns the row set defined by the detail rows expression for the measure or its table. If no expression is defined, the data for the Internet Sales table is returned because it's the table containing the measure.

DAXCopy

EVALUATE DETAILROWS([Internet Total Sales])

### Object-level security

This release introduces [object-level security](https://learn.microsoft.com/en-us/analysis-services/tabular-models/object-level-security?view=asallproducts-allversions) for tables and columns. In addition to restricting access to table and column data, sensitive table and column names can be secured. This helps prevent a malicious user from discovering such a table exists.

Object-level security must be set using the JSON-based metadata, Tabular Model Scripting Language (TMSL), or Tabular Object Model (TOM).

For example, the following code helps secure the Product table in the sample Adventure Works tabular model by setting the **MetadataPermission** property of the **TablePermission** class to **None**.

JSONCopy

//Find the Users role in Adventure Works and secure the Product table

ModelRole role = db.Model.Roles.Find("Users");

Table productTable = db.Model.Tables.Find("Product");

if (role != null && productTable != null)

{

TablePermission tablePermission;

if (role.TablePermissions.Contains(productTable.Name))

{

tablePermission = role.TablePermissions[productTable.Name];

}

else

{

tablePermission = new TablePermission();

role.TablePermissions.Add(tablePermission);

tablePermission.Table = productTable;

}

tablePermission.MetadataPermission = MetadataPermission.None;

}

db.Update(UpdateOptions.ExpandFull);

### Dynamic Management Views (DMVs)

[DMVs](https://learn.microsoft.com/en-us/analysis-services/instances/use-dynamic-management-views-dmvs-to-monitor-analysis-services?view=asallproducts-allversions) are queries in SQL Server Profiler that return information about local server operations and server health. This release includes improvements to [Dynamic Management Views](https://learn.microsoft.com/en-us/sql/analysis-services/instances/use-dynamic-management-views-dmvs-to-monitor-analysis-services) (DMV) for tabular models at the 1200 and 1400 compatibility levels.

DISCOVER\_CALC\_DEPENDENCY Now works with tabular 1200 and higher models. Tabular 1400 and higher models show dependencies between M partitions, M expressions and structured data sources. To learn more, see the [Analysis Services blog](https://learn.microsoft.com/en-us/archive/blogs/analysisservices/whats-new-in-sql-server-2017-rc1-for-analysis-services).

MDSCHEMA\_MEASUREGROUP\_DIMENSIONS Improvements are included for this DMV, which is used by various client tools to show measure dimensionality. For example, the Explore feature in Excel Pivot Tables allows the user to cross-drill to dimensions related to the selected measures. This release corrects the cardinality columns, which were previously showing incorrect values.

### DAX enhancements

One of the most important pieces of new DAX functionality is the new [IN Operator / CONTAINSROW Function](https://learn.microsoft.com/en-us/dax/in-operator-containsrow-function) for DAX expressions. This is similar to the TSQL IN operator commonly used to specify multiple values in a WHERE clause.

Previously, it was common to specify multi-value filters using the logical OR operator, like in the following measure expression:

DAXCopy

Filtered Sales:=CALCULATE (

[Internet Total Sales],

'Product'[Color] = "Red"

|| 'Product'[Color] = "Blue"

|| 'Product'[Color] = "Black"

)

This is simplified using the IN operator:

DAXCopy

Filtered Sales:=CALCULATE (

[Internet Total Sales], 'Product'[Color] IN { "Red", "Blue", "Black" }

)

In this case, the IN operator refers to a single-column table with 3 rows; one for each of the specified colors. Note the table constructor syntax uses curly braces.

The IN operator is functionally equivalent to the CONTAINSROW function:

DAXCopy

Filtered Sales:=CALCULATE (

[Internet Total Sales], CONTAINSROW({ "Red", "Blue", "Black" }, 'Product'[Color])

)

The IN operator can also be used effectively with table constructors. For example, the following measure filters by combinations of product color and category:

DAXCopy

Filtered Sales:=CALCULATE (

[Internet Total Sales],

FILTER( ALL('Product'),

( 'Product'[Color] = "Red" && Product[Product Category Name] = "Accessories" )

|| ( 'Product'[Color] = "Blue" && Product[Product Category Name] = "Bikes" )

|| ( 'Product'[Color] = "Black" && Product[Product Category Name] = "Clothing" )

)

)

By using the new IN operator, the measure expression above is now equivalent to the one below:

DAXCopy

Filtered Sales:=CALCULATE (

[Internet Total Sales],

FILTER( ALL('Product'),

('Product'[Color], Product[Product Category Name]) IN

{ ( "Red", "Accessories" ), ( "Blue", "Bikes" ), ( "Black", "Clothing" ) }

)

)

### Additional improvements

In addition to all the new features, Analysis Services, SSDT, and SSMS also include the following improvements:

* Hierarchy and column reuse surfaced in more helpful locations in the Power BI field list.
* Date relationships to easily create relationships to date dimensions based on date fields.
* Default installation option for Analysis Services is now for tabular mode.
* New Get Data (Power Query) data sources.
* DAX Editor for SSDT.
* Existing DirectQuery data sources support for M queries.
* SSMS improvements, such as viewing, editing, and scripting support for structured data sources.

### Deprecated features in SSAS 2017

The following features are [deprecated](https://learn.microsoft.com/en-us/analysis-services/what-s-new-in-sql-server-analysis-services?view=asallproducts-allversions#definitions) in this release:

| **Mode/Category** | **Feature** |
| --- | --- |
| Multidimensional | Data Mining |
| Multidimensional | Remote linked measure groups |
| Tabular | Models at the 1100 and 1103 compatibility level |
| Tabular | Tabular Object Model properties - Column.TableDetailPosition, Column.IsDefaultLabel, Column.IsDefaultImage |
| Tools | SQL Server Profiler for Trace Capture  The replacement is to use Extended Events Profiler embedded in SQL Server Management Studio. See [Monitor Analysis Services with SQL Server Extended Events](https://learn.microsoft.com/en-us/analysis-services/instances/monitor-analysis-services-with-sql-server-extended-events?view=asallproducts-allversions). |
| Tools | Server Profiler for Trace Replay Replacement. There is no replacement. |
| Trace Management Objects and Trace APIs | Microsoft.AnalysisServices.Trace objects (contains the APIs for Analysis Services Trace and Replay objects). The replacement is multi-part:  - Trace Configuration: Microsoft.SqlServer.Management.XEvent - Trace Reading: Microsoft.SqlServer.XEvent.Linq - Trace Replay: None |

### Discontinued features in SSAS 2017

The following features are [discontinued](https://learn.microsoft.com/en-us/analysis-services/what-s-new-in-sql-server-analysis-services?view=asallproducts-allversions#definitions) in this release:

| **Mode/Category** | **Feature** |
| --- | --- |
| Tabular | VertiPaqPagingPolicy memory property value (2), enable paging to disk using memory mapped files. |
| Multidimensional | Remote partitions |
| Multidimensional | Remote linked measure groups |
| Multidimensional | Dimensional writeback |
| Multidimensional | Linked dimensions |

### Breaking changes in SSAS 2017

There are no [breaking changes](https://learn.microsoft.com/en-us/analysis-services/what-s-new-in-sql-server-analysis-services?view=asallproducts-allversions#definitions) in this release.

### Behavior changes in SSAS 2017

Changes to MDSCHEMA\_MEASUREGROUP\_DIMENSIONS and DISCOVER\_CALC\_DEPENDENCY, detailed in the [What's new in SQL Server 2017 CTP 2.1 for Analysis Services](https://learn.microsoft.com/en-us/archive/blogs/analysisservices/whats-new-in-sql-server-2017-ctp-2-1-for-analysis-services) announcement.

## SQL Server 2016 Analysis Services

SQL Server 2016 Analysis Services includes many new enhancements providing improved performance, easier solution authoring, automated database management, enhanced relationships with bi-directional cross filtering, parallel partition processing, and much more. At the heart of most enhancements for this release is the new 1200 compatibility level for tabular model databases.

### SQL Server 2016 Service Pack 1 (SP1) Analysis Services

[Download SQL Server 2016 SP1](https://www.microsoft.com/download/details.aspx?id=54276)

SQL Server 2016 Service SP1 Analysis Services provides improved performance and scalability through Non-Uniform Memory Access (NUMA) awareness and optimized memory allocation based on **Intel Threading Building Blocks** (Intel TBB). This new functionality helps lower Total Cost of Ownership (TCO) by supporting more users on fewer, more powerful enterprise servers.

In particular, SQL Server 2016 SP1 Analysis Services features improvements in these key areas:

* **NUMA awareness** - For better NUMA support, the in-memory (VertiPaq) engine inside Analysis Services now maintains a separate job queue on each NUMA node. This guarantees the segment scan jobs run on the same node where the memory is allocated for the segment data. Note, NUMA awareness is only enabled by default on systems with at least four NUMA nodes. On two-node systems, the costs of accessing remote allocated memory generally doesn't warrant the overhead of managing NUMA specifics.
* **Memory allocation** - Analysis Services has been accelerated with Intel Threading Building Blocks, a scalable allocator that provides separate memory pools for every core. As the number of cores increases, the system can scale almost linearly.
* **Heap fragmentation** - The Intel TBB-based scalable allocator also helps to mitigate performance problems due to heap fragmentation that have been shown to occur with the Windows Heap.

Performance and scalability testing showed significant gains in query throughput when running SQL Server 2016 SP1 Analysis Services on large multi-node enterprise servers.

While most enhancements in this release are specific to tabular models, a number of enhancements have been made to multidimensional models; for example, distinct count ROLAP optimization for data sources like DB2 and Oracle, drill-through multi-selection support with Excel 2016, and Excel query optimizations.

### SQL Server 2016 General Availability (GA) Analysis Services

### Modeling

#### Improved modeling performance for tabular 1200 models

For tabular 1200 models, metadata operations in SSDT are much faster than tabular 1100 or 1103 models. By comparison, on the same hardware, creating a relationship on a model set to the SQL Server 2014 compatibility level (1103) with 23 tables takes 3 seconds, whereas the same relationship on a model created set to compatibility level 1200 takes just under a second.

#### Project templates added for tabular 1200 models in SSDT

With this release, you no longer need two versions of SSDT for building relational and BI projects. SQL Server Data Tools for Visual Studio 2015 adds project templates for Analysis Services solutions, including **Analysis Services Tabular Projects** used for building models at the 1200 compatibility level. Other Analysis Services project templates for multidimensional and data mining solutions are also included, but at the same functional level (1100 or 1103) as in previous releases.

#### Display folders

Display folders are now available for tabular 1200 models. Defined in SQL Server Data Tools and rendered in client applications like Excel or Power BI Desktop, display folders help you organize large numbers of measures into individual folders, adding a visual hierarchy for easier navigation in field lists.

#### Bi-directional cross filtering

New in this release is a built-in approach for enabling bi-directional cross filters in tabular models, eliminating the need for hand-crafted DAX workarounds for propagating filter context across table relationships. Filters are only auto-generated when the direction can be established with a high degree of certainty. If there is ambiguity in the form of multiple query paths across table relationships, a filter won't be created automatically. See [Bi-directional cross filters for tabular models in SQL Server 2016 Analysis Services](https://learn.microsoft.com/en-us/analysis-services/tabular-models/bi-directional-cross-filters-tabular-models-analysis-services?view=asallproducts-allversions) for details.

#### Translations

You can now store translated metadata in a tabular 1200 model. Metadata in the model includes fields for **Culture**, translated captions, and translated descriptions. To add translations, use the **Model** > **Translations** command in SQL Server Data Tools. See [Translations in tabular models (Analysis Services)](https://learn.microsoft.com/en-us/analysis-services/tabular-models/translations-in-tabular-models-analysis-services?view=asallproducts-allversions) for details.

#### Pasted tables

You can now upgrade an 1100 or 1103 tabular model to 1200 when the model contains pasted tables. We recommend using SQL Server Data Tools. In SSDT, set **CompatibilityLevel** to 1200 and then deploy to a SQL Server 2017 instance of SQL Server Analysis Services. See [Compatibility Level for Tabular models in Analysis Services](https://learn.microsoft.com/en-us/analysis-services/tabular-models/compatibility-level-for-tabular-models-in-analysis-services?view=asallproducts-allversions) for details.

#### Calculated tables in SSDT

A calculated table is a model-only construction based on a DAX expression or query in SSDT. When deployed in a database, a calculated table is indistinguishable from regular tables.

There are several uses for calculated tables, including the creation of new tables to expose an existing table in a specific role. The classic example is a Date table that operates in multiple contexts (order date, ship date, and so forth). By creating a calculated table for a given role, you can now activate a table relationship to facilitate queries or data interaction using the calculated table. Another use for calculated tables is to combine parts of existing tables into an entirely new table that exists only in the model. See [Create a Calculated Table](https://learn.microsoft.com/en-us/analysis-services/tabular-models/create-a-calculated-table-ssas-tabular?view=asallproducts-allversions) to learn more.

#### Formula fixup

With formula fixup on a tabular 1200 model, SSDT will automatically update any measures that is referencing a column or table that was renamed.

#### Support for Visual Studio configuration manager

To support multiple environments, like Test and Pre-production environments, Visual Studio allows developers to create multiple project configurations using the configuration manager. Multidimensional models already leverage this but tabular models did not. With this release, you can now use configuration manager to deploy to different servers.

### Instance management

#### Administer Tabular 1200 models in SSMS

In this release, an Analysis Services instance in Tabular server mode can run tabular models at any compatibility level (1100, 1103, 1200). The latest [SQL Server Management Studio](https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms) is updated to display properties and provide database model administration for tabular models at the 1200 compatibility level.

#### Parallel processing for multiple table partitions in tabular models

This release includes new parallel processing functionality for tables with two or more partitions, increasing processing performance. There are no configuration settings for this feature. For more information about configuring partitions and processing tables, see [Tabular model partitions](https://learn.microsoft.com/en-us/analysis-services/tabular-models/partitions-ssas-tabular?view=asallproducts-allversions).

#### Add computer accounts as Administrators in SSMS

SQL Server Analysis Services administrators can now use SQL Server Management Studio to configure computer accounts to be members of the SQL Server Analysis Services administrators group. In the **Select Users or Groups** dialog, set the **Locations** for the computers domain and then add the **Computers** object type. For more information, see [Grant server admin rights to an Analysis Services instance](https://learn.microsoft.com/en-us/analysis-services/instances/grant-server-admin-rights-to-an-analysis-services-instance?view=asallproducts-allversions).

#### DBCC for Analysis Services

Database Consistency Checker (DBCC) runs internally to detect potential data corruption issues on database load, but can also be run on demand if you suspect problems in your data or model. DBCC runs different checks depending on whether the model is tabular or multidimensional. See [Database Consistency Checker (DBCC) for Analysis Services tabular and multidimensional databases](https://learn.microsoft.com/en-us/analysis-services/instances/database-consistency-checker-dbcc-for-analysis-services?view=asallproducts-allversions) for details.

#### Extended Events updates

This release adds a graphical user interface to SQL Server Management Studio to configure and manage SQL Server Analysis Services Extended Events. You can set up live data streams to monitor server activity in real time, keep session data loaded in memory for faster analysis, or save data streams to a file for offline analysis. For more information, see [Monitor Analysis Services with SQL Server Extended Events](https://learn.microsoft.com/en-us/analysis-services/instances/monitor-analysis-services-with-sql-server-extended-events?view=asallproducts-allversions).

### Scripting

#### PowerShell for Tabular models

This release includes PowerShell enhancements for tabular models at compatibility level 1200. You can use all of the applicable cmdlets, plus cmdlets specific to Tabular mode: Invoke-ProcessASDatabase and Invoke-ProcessTable cmdlet.

#### SSMS scripting database operations

In the [latest SQL Server Management Studio (SSMS)](https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms), script is now enabled for database commands, including Create, Alter, Delete, Backup, Restore, Attach, Detach. Output is Tabular Model Scripting Language (TMSL) in JSON. See [Tabular Model Scripting Language (TMSL) Reference](https://learn.microsoft.com/en-us/analysis-services/tmsl/tabular-model-scripting-language-tmsl-reference?view=asallproducts-allversions) for more information.

#### Analysis Services Execute DDL Task

Analysis Services Execute DDL Task now also accepts Tabular Model Scripting Language (TMSL) commands.

#### SSAS PowerShell cmdlet

SSAS PowerShell cmdlet **Invoke-ASCmd** now accepts Tabular Model Scripting Language (TMSL) commands. Other SSAS PowerShell cmdlets may be updated in a future release to use the new tabular metadata (exceptions will be called out in the release notes). See Analysis Services PowerShell Reference for details.

#### Tabular Model Scripting Language (TMSL) supported in SSMS

Using the [latest version of SSMS](https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms), you can now create scripts to automate most administrative tasks for tabular 1200 models. Currently, the following tasks can be scripted: Process at any level, plus CREATE, ALTER, DELETE at the database level.

Functionally, TMSL is equivalent to the XMLA ASSL extension that provides multidimensional object definitions, except that TMSL uses native descriptors like **model**, **table**, and **relationship** to describe tabular metadata. See [Tabular Model Scripting Language (TMSL) Reference](https://learn.microsoft.com/en-us/analysis-services/tmsl/tabular-model-scripting-language-tmsl-reference?view=asallproducts-allversions) for details about the schema.

A generated JSON-based script for a tabular model might look like the following:

JSONCopy

{

"create": {

"database": {

"name": "AdventureWorksTabular1200",

"id": "AdventureWorksTabular1200",

"compatibilityLevel": 1200,

"readWriteMode": "readWrite",

"model": {}

}

}

}

The payload is a JSON document that can be as minimal as the example shown above, or highly embellished with the full set of object definitions. [Tabular Model Scripting Language (TMSL) Reference](https://learn.microsoft.com/en-us/analysis-services/tmsl/tabular-model-scripting-language-tmsl-reference?view=asallproducts-allversions) describes the syntax.

At the database level, CREATE, ALTER, and DELETE commands will output TMSL script in the familiar XMLA window. Other commands, such as Process, can also be scripted in this release. Script support for many other actions may be added in a future release.

| **Scriptable commands** | **Description** |
| --- | --- |
| create | Adds a database, connection, or partition. The ASSL equivalent is CREATE. |
| createOrReplace | Updates an existing object definition (database, connection, or partition) by overwriting a previous version. The ASSL equivalent is ALTER with AllowOverwrite set to true and ObjectDefinition to ExpandFull. |
| delete | Removes an object definition. ASSL equivalent is DELETE. |
| refresh | Processes the object. ASSL equivalent is PROCESS. |

### DAX

#### Improved DAX formula editing

Updates to the formula bar help you write formulas with more ease by differentiating functions, fields and measures using syntax coloring, it provides intelligent function and field suggestions and tells you if parts of your DAX expression are wrong using error squiggles. It also allows you to use multiple lines (Alt + Enter) and indentation (Tab). The formula bar now also allows you to write comments as part of your measures, just type "//" and everything after these characters on the same line will be considered a comment.

#### DAX variables

This release now includes support for variables in DAX. Variables can now store the result of an expression as a named variable, which can then be passed as an argument to other measure expressions. Once resultant values have been calculated for a variable expression, those values do not change, even if the variable is referenced in another expression. For more information, see [VAR Function](https://learn.microsoft.com/en-us/dax/var-dax).

#### New DAX functions

With this release, DAX introduces over fifty new functions to support faster calculations and enhanced visualizations in Power BI. To learn more, see [New DAX Functions](https://learn.microsoft.com/en-us/dax/new-dax-functions).

#### Save incomplete measures

You can now save incomplete DAX measures directly in a tabular 1200 model project and pick it up again when you are ready to continue.

#### Additional DAX enhancements

* Non empty calculation - Reduces the number of scans needed for non empty.
* Measure Fusion - Multiple measures from the same table will be combined into a single storage engine - query.
* Grouping sets - When a query asks for measures at multiple granularities (Total/Year/Month), a single - query is sent at the lowest level and the rest of the granularities are derived from the lowest level.
* Redundant join elimination - A single query to the storage engine returns both the dimension columns and the measure values.
* Strict evaluation of IF/SWITCH - A branch whose condition is false will no longer result in storage engine queries. Previously, branches were eagerly evaluated but results discarded later on.

### Developer

#### Microsoft.AnalysisServices.Tabular namespace for Tabular 1200 programmability in AMO

Analysis Services Management Objects (AMO) is updated to include a new tabular namespace for managing a Tabular Mode instance of SQL Server 2016 Analysis Services, as well as provide the data definition language for creating or modifying tabular 1200 models programmatically. Visit [Microsoft.AnalysisServices.Tabular](https://learn.microsoft.com/en-us/dotnet/api/microsoft.analysisservices.tabular) to read up on the API.

#### Analysis Services Management Objects (AMO) updates

[Analysis Services Management Objects (AMO)](https://learn.microsoft.com/en-us/dotnet/api/) has been re-factored to include a second assembly, Microsoft.AnalysisServices.Core.dll. The new assembly separates out common classes like Server, Database, and Role that have broad application in Analysis Services, irrespective of server mode. Previously, these classes were part of the original Microsoft.AnalysisServices assembly. Moving them to a new assembly paves the way for future extensions to AMO, with clear division between generic and context-specific APIs. Existing applications are unaffected by the new assemblies. However, should you choose to rebuild applications using the new AMO assembly for any reason, be sure to add a reference to Microsoft.AnalysisServices.Core. Similarly, PowerShell scripts that load and call into AMO must now load Microsoft.AnalysisServices.Core.dll. Be sure to update any scripts.

#### JSON editor for BIM files

Code View in Visual Studio 2015 now renders the BIM file in JSON format for tabular 1200 models. The version of Visual Studio determines whether the BIM file is rendered in JSON via the built-in JSON Editor, or as simple text.

To use the JSON editor, with the ability to expand and collapse sections of the model, you will need the latest version of SQL Server Data Tools plus Visual Studio 2015 (any edition, including the free Community edition). For all other versions of SSDT or Visual Studio, the BIM file is rendered in JSON as simple text. At a minimum, an empty model will contain the following JSON:

JSONCopy

{

"name": "SemanticModel",

"id": "SemanticModel",

"compatibilityLevel": 1200,

"readWriteMode": "readWrite",

"model": {}

}

**Warning**

Avoid editing the JSON directly. Doing so can corrupt the model.

### New elements in MS-CSDLBI 2.0 schema

The following elements have been added to the **TProperty** complex type defined in the [MS-CSDLBI] 2.0 schema:

| **Element** | **Definition** |
| --- | --- |
| DefaultValue | A property that specifies the value used when evaluating the query. The DefaultValue property is optional, but it is automatically selected if the values from the member cannot be aggregated. |
| Statistics | A set of statistics from the underlying data that is associated with the column. These statistics are defined by the TPropertyStatistics complex type and are provided only if they are not computationally expensive to generate, as described in section 2.1.13.5 of the Conceptual Schema Definition File Format with Business Intelligence Annotations document. |

### DirectQuery

#### New DirectQuery implementation

This release sees significant enhancements in DirectQuery for tabular 1200 models. Here's a summary:

* DirectQuery now generates simpler queries that provide better performance.
* Extra control over defining sample datasets used for model design and testing.
* Row level security (RLS) is now supported for tabular 1200 models in DirectQuery mode. Previously, the presence of RLS prevented deploying a tabular model in DirectQuery mode.
* Calculated columns are now supported for tabular 1200 models in DirectQuery mode. Previously, the presence of calculated columns prevented deploying a tabular model in DirectQuery mode.
* Performance optimizations include redundant join elimination for VertiPaq and DirectQuery.

#### New data sources for DirectQuery mode

Data sources supported for tabular 1200 models in DirectQuery mode now include Oracle, Teradata and Microsoft Analytics Platform (formerly known as Parallel Data Warehouse). To learn more, see [DirectQuery Mode](https://learn.microsoft.com/en-us/analysis-services/tabular-models/directquery-mode-ssas-tabular?view=asallproducts-allversions).

### Deprecated features in SSAS 2016

The following features are [deprecated](https://learn.microsoft.com/en-us/analysis-services/what-s-new-in-sql-server-analysis-services?view=asallproducts-allversions#definitions) in this release:

| **Mode/Category** | **Feature** |
| --- | --- |
| Multidimensional | Remote partitions |
| Multidimensional | Remote linked measure groups |
| Multidimensional | Dimensional writeback |
| Multidimensional | Linked dimensions |
| Multidimensional | SQL Server table notifications for proactive caching. The replacement is to use polling for proactive caching. See [Proactive Caching (Dimensions)](https://learn.microsoft.com/en-us/analysis-services/multidimensional-models-olap-logical-dimension-objects/proactive-caching-dimensions?view=asallproducts-allversions) and [Proactive Caching (Partitions)](https://learn.microsoft.com/en-us/analysis-services/multidimensional-models-olap-logical-cube-objects/partitions-proactive-caching?view=asallproducts-allversions). |
| Multidimensional | Session cubes. There is no replacement. |
| Multidimensional | Local cubes. There is no replacement. |
| Tabular | Tabular model 1100 and 1103 compatibility levels will not be supported in a future release. The replacement is to set models at compatibility level 1200 or higher, converting model definitions to tabular metadata. See [Compatibility Level for Tabular models in Analysis Services](https://learn.microsoft.com/en-us/analysis-services/tabular-models/compatibility-level-for-tabular-models-in-analysis-services?view=asallproducts-allversions). |
| Tools | SQL Server Profiler for Trace Capture  The replacement is to use Extended Events Profiler embedded in SQL Server Management Studio. See [Monitor Analysis Services with SQL Server Extended Events](https://learn.microsoft.com/en-us/analysis-services/instances/monitor-analysis-services-with-sql-server-extended-events?view=asallproducts-allversions). |
| Tools | Server Profiler for Trace Replay Replacement. There is no replacement. |
| Trace Management Objects and Trace APIs | Microsoft.AnalysisServices.Trace objects (contains the APIs for Analysis Services Trace and Replay objects). The replacement is multi-part:  - Trace Configuration: Microsoft.SqlServer.Management.XEvent - Trace Reading: Microsoft.SqlServer.XEvent.Linq - Trace Replay: None |

### Discontinued features in SSAS 2016

The following features are [discontinued](https://learn.microsoft.com/en-us/analysis-services/what-s-new-in-sql-server-analysis-services?view=asallproducts-allversions#definitions) in this release:

| **Feature** | **Replacement or workaround** |
| --- | --- |
| CalculationPassValue (MDX) | None. This feature was deprecated in SQL Server 2005. |
| CalculationCurrentPass (MDX) | None. This feature was deprecated in SQL Server 2005. |
| NON\_EMPTY\_BEHAVIOR query optimizer hint | None. This feature was deprecated in SQL Server 2008. |
| COM assemblies | None. This feature was deprecated in SQL Server 2008. |
| CELL\_EVALUATION\_LIST intrinsic cell property | None. This feature was deprecated in SQL Server 2005. |

### Breaking changes in SSAS 2016

#### .NET 4.0 version upgrade

Analysis Services Management Objects (AMO), ADOMD.NET, and Tabular Object Model (TOM) client libraries now target the .NET 4.0 runtime. This can be a breaking change for applications that target .NET 3.5. Applications using newer versions of these assemblies must now target .NET 4.0 or later.

#### AMO version upgrade

This release is a version upgrade for [Analysis Services Management Objects (AMO)](https://learn.microsoft.com/en-us/dotnet/api/) and is a breaking change under certain circumstances. Existing code and scripts that call into AMO will continue to run as before if you upgrade from a previous version. However, if you need to recompile your application and you are targeting a SQL Server 2016 Analysis Services instance, you must add the following namespace to make your code or script operational:

c#Copy

using Microsoft.AnalysisServices;

using Microsoft.AnalysisServices.Core;

The [Microsoft.AnalysisServices.Core](https://learn.microsoft.com/en-us/dotnet/api/microsoft.analysisservices.core) namespace is now required whenever you reference the Microsoft.AnalysisServices assembly in your code. Objects that were previously only in the **Microsoft.AnalysisServices** namespace are moved to the Core namespace in this release if the object is used the same way in both tabular and multidimensional scenarios. For example, server-related APIs are relocated to the Core namespace.

Although there are now multiple namespaces, both exist in the same assembly (Microsoft.AnalysisServices.dll).

#### XEvent DISCOVER changes

To better support XEvent DISCOVER streaming in SSMS for SQL Server 2016 Analysis Services, DISCOVER\_XEVENT\_TRACE\_DEFINITION is replaced with the following XEvent traces:

* DISCOVER\_XEVENT\_PACKAGES
* DISCOVER\_XEVENT\_OBJECT
* DISCOVER\_XEVENT\_OBJECT\_COLUMNS
* DISCOVER\_XEVENT\_SESSION\_TARGETS

### Behavior changes in SSAS 2016

#### Analysis Services in SharePoint mode

Running the Power Pivot Configuration wizard is no longer required as a post-installation task. This is true for all supported versions of SharePoint that load models from the current SQL Server 2016 Analysis Services.

#### DirectQuery mode for Tabular models

DirectQuery is a data access mode for tabular models, where query execution is performed on a backend relational database, retrieving a result set in real time. It's often used for very large datasets that cannot fit in memory or when data is volatile and you want the most recent data returned in queries against a tabular model.

DirectQuery has existed as a data access mode for the last several releases. In SQL Server 2016 Analysis Services, the implementation has been slightly revised, assuming the tabular model is at compatibility level 1200 or higher. DirectQuery has fewer restrictions than before. It also has different database properties.

If you are using DirectQuery in an existing tabular model, you can keep the model at its currently compatibility level of 1100 or 1103 and continue to use DirectQuery as its implemented for those levels. Alternatively, you can upgrade to 1200 or higher to benefit from enhancements made to DirectQuery.

There is no in-place upgrade of a DirectQuery model because the settings from older compatibility levels do not have exact counterparts in the newer 1200 and higher compatibility levels. If you have an existing tabular model that runs in DirectQuery mode, you should open the model in SQL Server Data Tools, turn DirectQuery off, set the **Compatibility Level** property to 1200 or higher, and then reconfigure the DirectQuery properties. See [DirectQuery Mode](https://learn.microsoft.com/en-us/analysis-services/tabular-models/directquery-mode-ssas-tabular?view=asallproducts-allversions) for details.

## Definitions

A deprecated feature will be discontinued from the product in a future release, but is still supported and included in the current release to maintain backward compatibility. It's recommended you discontinue using deprecated features in new and existing projects to maintain compatibility with future releases. Documentation is not updated for deprecated features.

A discontinued feature was deprecated in an earlier release. It may continue to be included in the current release, but is no longer supported. Discontinued features may be removed entirely in the stated or future release.

A breaking change causes a feature, data model, application code, or script to no longer function after upgrading to the current release.

A behavior change affects how the same feature works in the current release as compared to the previous release. Only significant behavior changes are described. Changes in user interface are not included. Changes to default values, manual configuration required to complete an upgrade or restore functionality, or a new implementation of an existing feature are all examples of a behavior change.

**Unit 5: Business Intelligence and Automated Reports using Power BI**

## How business intelligence works

There are four keys steps that business intelligence follows to transform raw data into easy-to-digest insights for everyone in the organization to use. The first three—data collection, analysis, and visualization—set the stage for the final decision-making step. Before using BI, businesses had to do much of their analysis manually, but BI tools automate many of the processes and save companies time and effort.

### Step 1: Collect and transform data from multiple sources

Business intelligence tools typically use the extract, transform, and load (ETL) method to aggregate structured and unstructured data from multiple sources. This data is then transformed and remodeled before being stored in a central location, so applications can easily analyze and query it as one comprehensive data set.

### Step 2: Uncover trends and inconsistencies

Data mining, or data discovery, typically uses automation to quickly analyze data to find patterns and outliers which provide insight into the current state of business. BI tools often feature several types of [data modeling](https://powerbi.microsoft.com/en-us/what-is-data-modeling/) and analytics—including exploratory, descriptive, statistical, and predictive—that further explore data, predict trends, and make recommendations.

### Step 3: Use data visualization to present findings

Business intelligence reporting uses [data visualizations](https://powerbi.microsoft.com/en-us/data-visualization/) to make findings easier to understand and share. Reporting methods include interactive [data dashboards](https://powerbi.microsoft.com/en-us/data-dashboards/), charts, graphs, and maps that help users see what’s going on in the business right now.

### Step 4: Take action on insights in real time

Viewing current and historical data in context with business activities gives companies the ability to quickly move from insights to action. Business intelligence enables real time adjustments and long-term strategic changes that eliminate inefficiencies, adapt to market shifts, correct supply problems, and solve customer issues.

## Why companies benefit from using business intelligence tools

Because [business intelligence tools](https://powerbi.microsoft.com/en-us/business-intelligence-tools/) speed up information analysis and performance evaluation, they’re valuable in helping companies reduce inefficiencies, flag potential problems, find new revenue streams, and identify areas of future growth.

Some of the specific benefits that businesses experience when using BI include:

* Increased efficiency of operational processes.
* Insight into customer behavior and shopping patterns.
* Accurate tracking of sales, marketing, and financial performance.
* Clear benchmarks based on historical and current data.
* Instant alerts about data anomalies and customer issues.
* Analyses that can be shared in real-time across departments.

In the past, business intelligence tools were primarily used by data analysts and IT users. Now, self-service BI platforms make business intelligence available to everyone from executives to operations teams.

## Here’s how business intelligence improves the way work is done in six key areas:

### Customer experience

Access all your customer information in one place, so you can direct resources to key areas that will positively impact customer engagement and support.

### Sales and marketing

Gain visibility into sales and marketing performance, consumer behavior, and buying trends which ensures future marketing initiatives are effective and drive revenue.

### Operations

Improve operations by automating routine analytics tasks, refining processes, reducing inefficiencies, and increasing productivity.

### Finance

Use custom dashboards to get a holistic view of the company’s financial health, study historical data, calculate risk, and predict trends.

### Inventory control

Automate data analysis and reporting to improve inventory management, accelerate fulfillment, and help anticipate buying trends.

### Security and compliance

Centralize data to improve accuracy and transparency, making it easier to uncover errors, security issues, and reduce compliance risks.

When evaluating [business intelligence tools](https://powerbi.microsoft.com/en-us/business-intelligence-tools/), look for a product that’s secure, compliant, globally available, and trusted. It should also have features that make BI insights accessible to your entire organization—such as data visualization, shared dashboards, artificial intelligence, and machine learning.

**“It has gotten our key decision makers out of the weeds by providing them with metrics they need to do their job, and not have to wrangle the data to get answers. Flexibility, interactivity … just jump in.”**

**Daksha R**

Manager, Clinical Research Analytics

## Help everyone turn data into immediate impact with Power BI

Empower people across your organization to uncover insights hidden in your data and find clarity with up-to-the-minute analytics and rich, visual business intelligence reports.

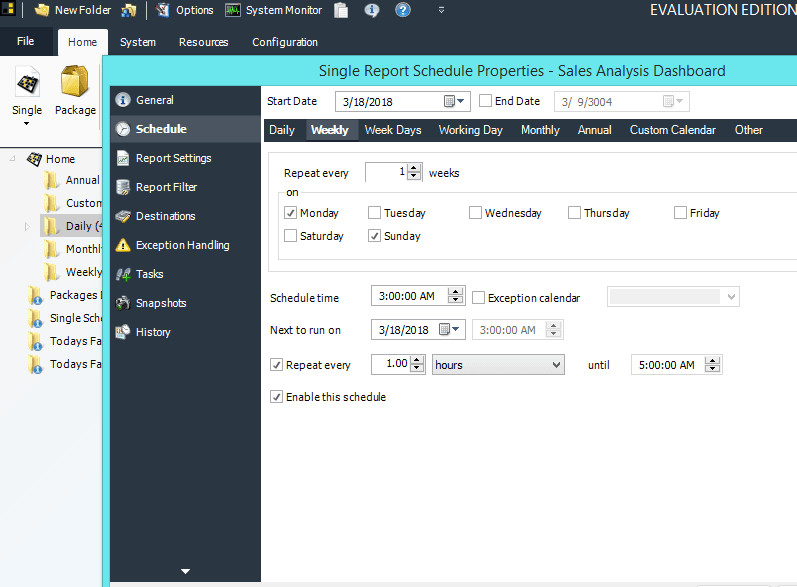
## Real-life business intelligence examples

Business intelligence is applied differently from business to business and across a range of sectors—finance, retail and consumer goods, energy, technology, government, education, healthcare, manufacturing, and professional services. Here’s how business intelligence is being used by different industries to achieve success.

## Example Power BI Reports



## Schedule Power BI Reports



**Here's how you can send UNLIMITED reports to UNLIMITED users with ONLY ONE Power BI license — in any format.**

If you are familiar with [SSRS scheduling](https://go.christiansteven.com/ssrs/sql-reporting-services-ssrs-design-generate-deploy-business-reports) then scheduling reports with [PBRS](https://go.christiansteven.com/power-bi-report-scheduler) will be familiar territory.  
  
With PBRS, you’re able to do data-driven subscriptions and scheduling for any number of Power BI reports and dashboards.

## On-Premise or In the Cloud

For businesses with high-security and data protection needs, PBRS is also an **on-premise** solution that allows you to be completely in control of the environment it runs on, which means you can have all the functionality you need, without the security concerns.

## Power BI Automated Reports

Save time and money by automating the export and delivery of your Power BI and SSRS reports using PBRS.