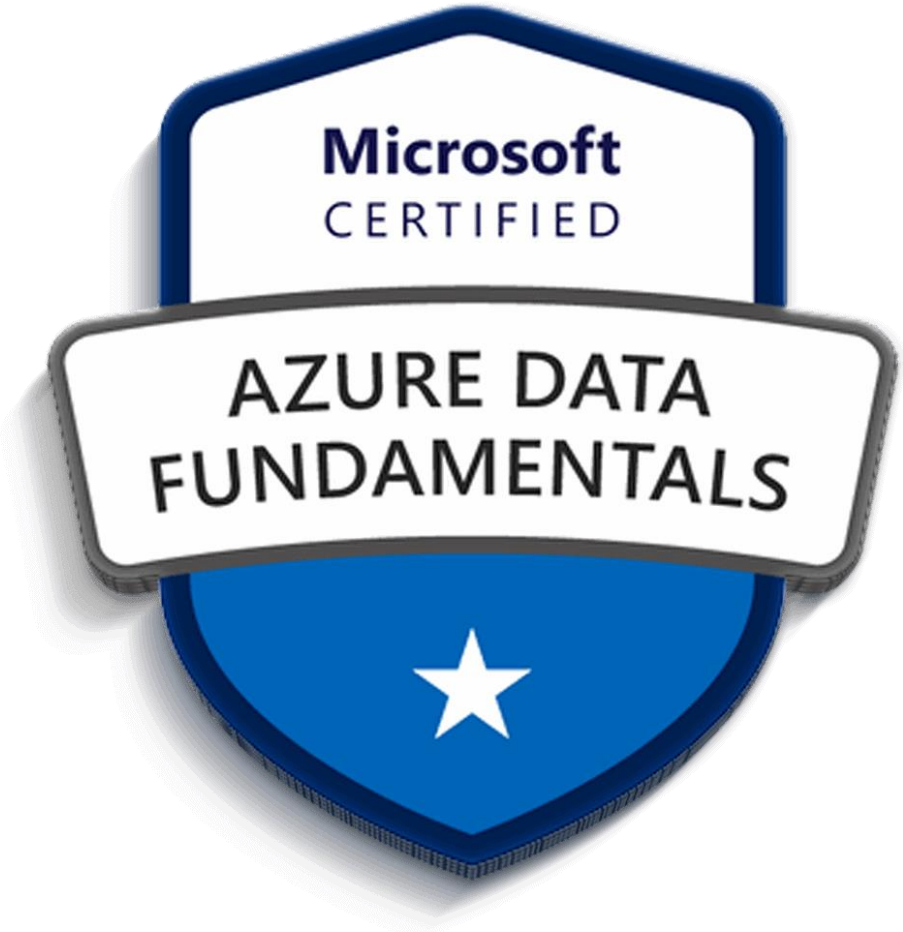
# Azure notebook

DP-900 Exam Preparation

By Carlos Pereira Coto



# Describe core data concepts

## Explore core data concepts

Data solutions include software technologies and platforms that can help facilitate the collection, analysis, and storage of valuable information. In this competitive market, data is a valuable asset. When analyzed properly, data provides a wealth of useful information and inform critical business decisions.

### What is data?

Data is a collection of facts such as numbers, descriptions, and observations used in decision making. You can classify data as structured, semi-structured, or unstructured.

#### Structured

Typically, tabular data that is represented by rows and columns in a database. Databases that hold tables in this form are called relational databases (the mathematical term relation refers to an organized set of data held as a table). Each row in a table has the same set of columns.

#### Semi-structured

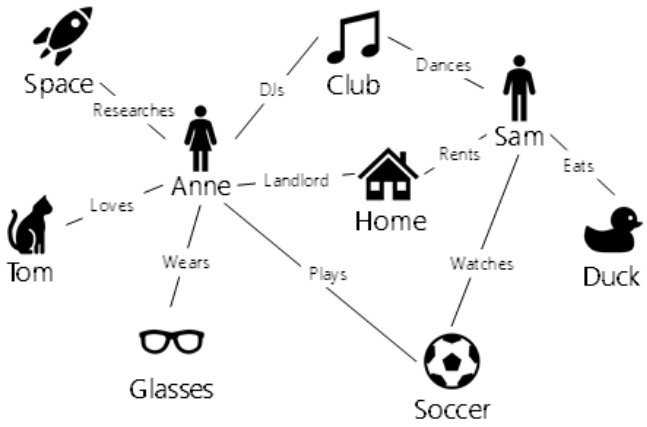
Information that doesn't reside in a relational database but still has some structure to it. Examples include documents held in JavaScript Object Notation (JSON) format.

There are other types of semi-structured data as well. Examples include key-value stores and graph databases.

A key-value database stores Associative arrays. In those arrays, a Key serves as a unique identifier to retrieve a specific value. Those values can be anything from a number or a string to a complex object, like a JSON file.

| *Person ID* | *Type* | *Attribute* | *Attribute* | *Attribute* | *Attribute* |
|---|---|---|---|---|---|
| 1 | President ID | Washington | Adams | Jefferson | Madison |
| 2 | Monarch ID | Henry VIII | Richard III | Elizabeth I | |

You can use a graph database to store and query information about complex relationships. A graph contains nodes (information about objects), and edges (information about the relationships between objects).



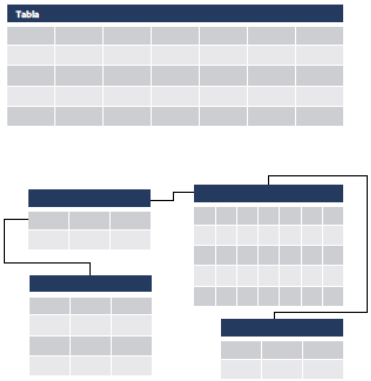

## Unstructured

Not all data is structured or even semi-structured. For example, audio and video files, and binary data files might not have a specific structure. They're referred to as unstructured data.

# ¿Qué son los datos?

Colección de hechos, números, descripciones, objetos, almacenados de forma estructurada, semiestructurada, no estructurada.

## Estructurados




## Semiestructurados

```
## Document 1 ## {
"customerID": "103248",
"name": { "first": "AAA",
"last": "BBB" }, "address": {
"street": "Main Street",
"number": "101", "city":
"Acity", "state": "NY" },
"ccOnFile": "yes",
"firstOrder": "02/28/2003" }
## Document 2 ## {
"customerID": "103249",
"name": { "title": "Mr",
"forename": "AAA",
"lastname": "BBB" },
"address": { "street":
"Another Street", "number":
"202", "city": "Bcity",
"county": "Gloucestershire",
"country-region": "UK" },
"ccOnFile": "yes" }
```

## No estructurados

## How is data defined, stored, and accessed in cloud computing?

Depending on the type of data such as structured, semi-structured, or unstructured, data will be stored differently.

Structured data is typically stored in a relational database such as SQL Server or Azure SQL Database. Azure SQL Database is a service that runs in the cloud. You can use it to create and access relational tables. The act of setting up the database server is called provisioning.

Store unstructured data such as video or audio files, you can use Azure Blob storage (Blob is an acronym for Binary Large Object).

Store semi-structured data such as documents, you can use a service such as Azure Cosmos DB.

After your service is provisioned, the service needs to be configured so that users can be given access to the data. You can typically define several levels of access.

- Read-only access means the users can read data but can't modify any existing data or create new data.
- Read/write access gives users the ability to view and modify existing data.
- Owner privilege gives full access to the data including managing the security like adding new users and removing access to existing users.

You can also define which users should be allowed to access the data in the first place. If the data is sensitive (or secret), you may want to restrict access to a few select users.

# Describe data processing solutions

Data processing solutions often fall into one of two broad categories: analytical systems, and transaction processing systems.

## What is a transactional system?

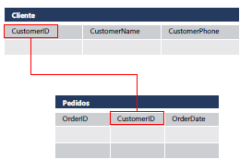
A transactional system is often what most people consider the primary function of business computing. A transactional system records transaction. Think of a transaction as a small, discrete, unit of work.

Transactional systems are often high-volume, sometimes handling many millions of transactions in a single day. The data being processed has to be accessible very quickly. The work performed by transactional systems is often referred to as Online Transactional Processing (OLTP).

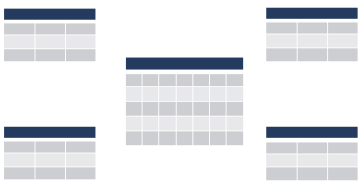
To support fast processing, the data in a transactional system is often divided into small pieces. Splitting tables out into separate groups of columns like this is called normalization. Normalization can enable a transactional system to cache much of the information required to perform transactions in memory, and speed throughput.

While normalization enables fast throughput for transactions, it can make querying more complex.

Almacenes de datos transaccionales frente a analíticos




Procesamiento transaccional en línea (OLTP)



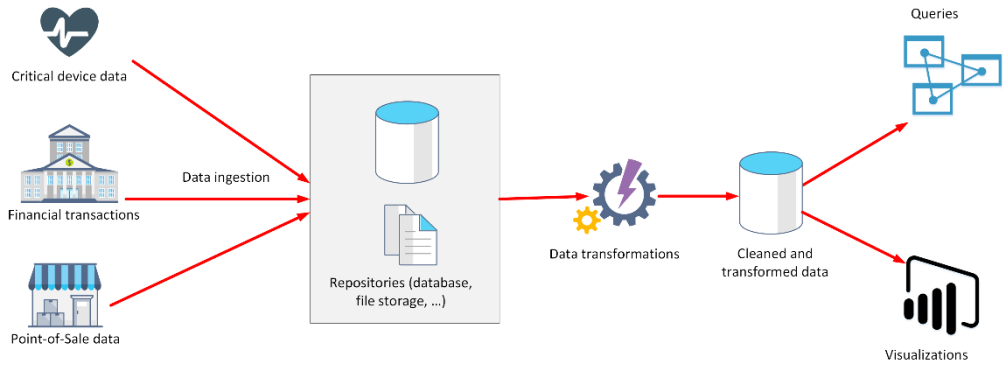
Procesamiento analítico en línea (OLAP)

What is an analytical system?

An analytical system is designed to support business users who need to query data and gain a big picture view of the information held in a database.

Analytical systems are concerned with capturing raw data and using it to generate insights. An organization can use these insights to make business decisions.

Most analytical data processing systems need to perform similar tasks: data ingestion, data transformation, data querying, and data visualization.




Data Ingestion

Data ingestion is the process of capturing the raw data. To process and analyze this data, you must first store the data in a repository of some sort. The repository could be a file store, a document database, or even a relational database.

Data Transformation/Data Processing

The raw data might not be in a format that is suitable for querying. The data might contain anomalies that should be filtered out, or it may require transforming in some way. After data is ingested into a data repository, you may want to do some cleaning operations and remove any questionable or invalid data, or perform some aggregations such as calculating profit, margin, and other Key Performance Indicators (KPIs).

### Data Querying

After data is ingested and transformed, you can query the data to analyze it. You may be looking for trends or attempting to determine the cause of problems in your systems.

### Data Visualization

Visualizing the data can often be useful as a tool for examining data. You can generate charts such as bar charts, line charts, plot results on geographical maps, pie charts, or illustrate how data changes over time. Microsoft offers visualization tools like Power BI to provide rich graphical representation of your data.

## Identify types of data and data storage

You can categorize data in many different ways, depending not only on how it's structured, but also on how the data is used.

### Describe the characteristics of relational and non-relational data

Relational databases provide probably the most well-understood model for holding data. The simple structure of tables and columns makes them easy to use initially, but the rigid structure can cause some problems.

You can solve these problems by using a process called normalization. Typically, the end result of the normalization process is that your data is split into a large number of narrow, well-defined tables (a narrow table is a table with few columns), with references from one table to another.

| CustomerID | FirstName | LastName |
|---|---|---|
| 1 | Jay | Adams |
| 2 | Donna | Carreras |
| 3 | Linda | Burnett |

| CustomerID | AddressID |
|---|---|
| 1 | A |
| 2 | B |
| 3 | C |

| AddressID | LineNumber | Text |
|---|---|---|
| A | 1 | 12 |
| A | 2 | Park Street |
| A | 3 | Some City |
| B | 1 | The Big House |
| B | 2 | High Road |
| B | 3 | Another City |
| B | 4 | 90210 |
| C | 1 | Freepost |
| C | 2 | AAA 123 |

Non-relational databases enable you to store data in a format that more closely matches the original structure. There are some disadvantages to using a document database though. If two customers cohabit and have the same address, in a relational database you would only need to store the address information once.

This duplication not only increases the storage required but can also make maintenance more complex (if the address changes, you must modify it in two documents).

## Describe transactional workloads

A transaction is a sequence of operations that are atomic. This means that either all operations in the sequence must be completed successfully, or if something goes wrong, all operations run so far in the sequence must be undone.

Each database transaction has a defined beginning point, followed by steps to modify the data within the database. At the end, the database either commits the changes to make them permanent, or rolls back the changes to the starting point, when the transaction can be tried again.

A transactional database must adhere to the ACID (Atomicity, Consistency, Isolation, Durability) properties to ensure that the database remains consistent while processing transactions.

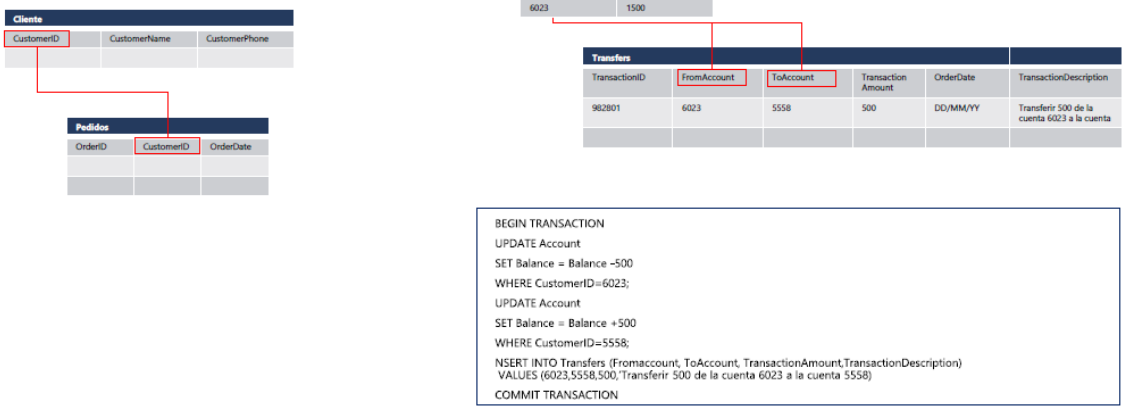
- **Atomicity** guarantees that each transaction is treated as a single unit, which either succeeds completely, or fails completely. An atomic system must guarantee atomicity in each and every situation, including power failures, errors, and crashes.
- **Consistency** ensures that a transaction can only take the data in the database from one valid state to another. A consistent database

should never lose or create data in a manner that can't be accounted for.

- **Isolation** ensures that concurrent execution of transactions leaves the database in the same state that would have been obtained if the transactions were executed sequentially.
- **Durability** guarantees that once a transaction has been committed, it will remain committed even if there's a system failure such as a power outage or crash.

Distributed databases are widely used in many organizations. A distributed database is a database in which data is stored across different physical locations. While this strategy helps to minimize latency, it can lead to temporary inconsistencies in the data.

## Cargas de trabajo transaccionales



# Describe analytical workloads

Analytical workloads are typically read-only systems that store vast volumes of historical data or business metrics, such as sales performance and inventory levels. Analytical workloads are used for data analysis and decision making.

Analytics can be based on a snapshot of the data at a given point in time, or a series of snapshots. Decision makers usually don't require all the details of every transaction. They want the bigger picture. An example of analytical information is a report on monthly sales.

Transactional information, however, is an integral part of analytical information. If you don't have good records of daily sales, you can't compile a useful report to identify trends.

## Sistema analítico

# Describe the difference between batch and streaming data

Data processing is simply the conversion of raw data to meaningful information through a process. Processing data as it arrives is called streaming. Buffering and processing the data in groups is called batch processing.

## Understand batch processing

In batch processing, newly arriving data elements are collected into a group. The whole group is then processed at a future time as a batch. Exactly when each group is processed can be determined in a number of ways.

An example of batch processing is the way that votes are typically counted in elections. The votes are not entered when they are cast but are all entered together at one time in a batch.

Advantages of batch processing include:

- Large volumes of data can be processed at a convenient time.
- It can be scheduled to run at a time when computers or systems might otherwise be idle, such as overnight, or during off-peak hours.

Disadvantages of batch processing include:

- The time delay between ingesting the data and getting the results.
- All of a batch job's input data must be ready before a batch can be processed. This means data must be carefully checked.

# Understand streaming and real-time data

In stream processing, each new piece of data is processed when it arrives. For example, data ingestion is inherently a streaming process.

Streaming handles data in real time. Unlike batch processing, there's no waiting until the next batch processing interval, and data is processed as individual pieces rather than being processed a batch at a time.

Examples of streaming data include:

- A financial institution tracks changes in the stock market in real time, computes value-at-risk, and automatically rebalances portfolios based on stock price movements.
- An online gaming company collects real-time data about player-game interactions and feeds the data into its gaming platform. It then analyzes the data in real time, offers incentives and dynamic experiences to engage its players.
- A real-estate website that tracks a subset of data from consumers' mobile devices and makes real-time property recommendations of properties to visit based on their geo-location.

Stream processing is ideal for time-critical operations that require an instant real-time response.

Understand differences between batch and streaming data

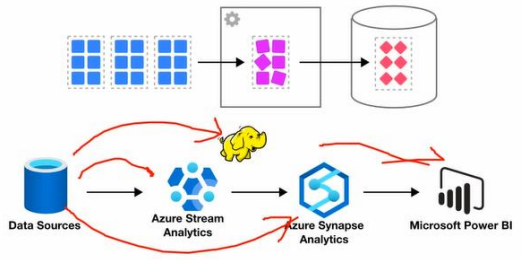
**Data Scope:** Batch processing can process all the data in the dataset. Stream processing typically only has access to the most recent data received.

**Data Size:** Batch processing is suitable for handling large datasets efficiently. Stream processing is intended for individual records or micro batches consisting of few records.

**Performance:** The latency for batch processing is typically a few hours. Stream processing typically occurs immediately, with latency in the order of seconds or milliseconds. Latency is the time taken for the data to be received and processed.

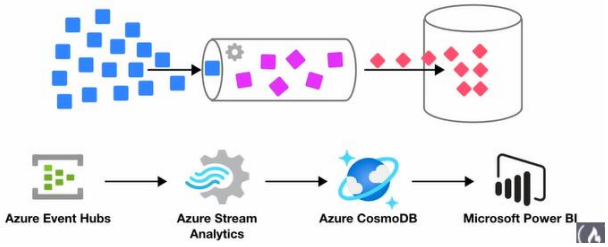
**Analysis:** You typically use batch processing for performing complex analytics. Stream processing is used for simple response functions, aggregates, or calculations such as rolling averages.

**Batch Processing**
When you send batches (a collection) of data to be processed.
Batches are generally scheduled: eg. Every day at 1PM
Batches are not real-time
Batches processing is ideal for very large processing workloads
Batch processing is more cost-effective




**Stream Processing**
When you process data as soon as it arrives:
- **Produces** will send data to a stream and
- **Consumers** will pull from the stream

Stream processing is good for real-time analytics or real-time processing (streaming video)
Much more expensive than batch processing



# Explore roles and responsibilities in the world of data

## Explore job roles in the world of data

There's a wide variety of roles involved in managing, controlling, and using data. Some roles are business-oriented, some involve more engineering, some focus on research, and some are hybrid roles that combine different aspects of data management.

### What are the roles in the world of data?

- **Database Administrators** manage databases, assigning permissions to users, storing backup copies of data and restore data in case of any failures.
- **Data Engineers** are vital in working with data, applying data cleaning routines, identifying business rules, and turning data into useful information.
- **Data Analysts** explore and analyze data to create visualizations and charts to enable organizations to make informed decisions.

### Azure Database Administrator role

An Azure database administrator is responsible for the design, implementation, maintenance, and operational aspects of on-premises and cloud-based database solutions built on Azure data services and SQL Server. They work with stakeholders to implement policies, tools, and processes for backup and recovery plans to recover following a natural disaster or human-made error.

## Data Engineer role

A data engineer collaborates with stakeholders to design and implement data-related assets that include data ingestion pipelines, cleansing and transformation activities, and data stores for analytical workloads. They're also responsible for ensuring that the privacy of data is maintained within the cloud and spanning from on-premises to the cloud data stores.

## Data Analyst role

A data analyst enables businesses to maximize the value of their data assets. They're responsible for designing and building scalable models, cleaning, and transforming data, and enabling advanced analytics capabilities through reports and visualizations.

**Database Administrator**: **configures and maintains a databases** eg. Azure Data services or SQL server.

| Responsibilities | Common Tools |
|---|---|
| • Database management | Azure Data Studio |
| • Manage security, granting user access | SQL Server Management Studio |
| • Backups | Azure Portal |
| • Monitors Performance | Azure CLI |

**Data Engineer**: **Design and implement data tasks** related to the transfer and storage of **big data**

| Responsibilities | Common Tools |
|---|---|
| • Database pipelines and process | • Azure Synapse Studio |
| • Data ingestion storage | • SQL |
| • Prepare data for analytics. | • Azure CLI |
| • Prepare data for analytical processing | |

**Data Analyst: Analyzes business data** to reveal important information

| Responsibilities | Common Tools |
|---|---|
| • Provides insights into the data | • Power BI Desktop |
| • Visual reporting | • Power BI Portal |
| • Modeling data for analysis | • Power BI services |
| • Combines data for visualization and analysis | • Power BI report bulder |

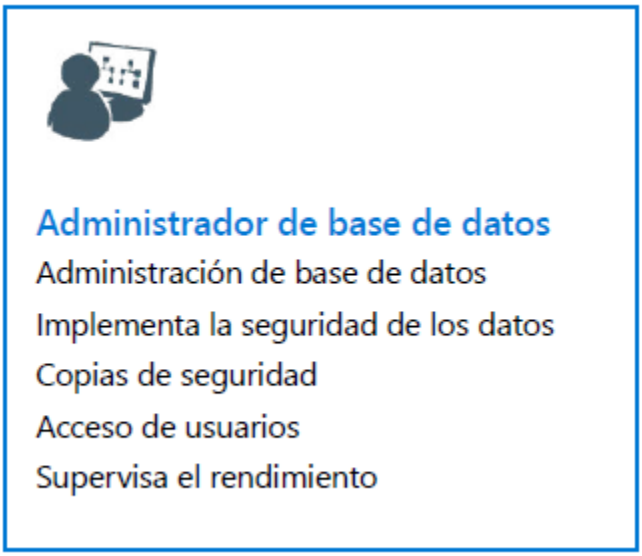
## Tasks and tools for database administration

A database administrator's primary job is to ensure that data is available, protected from loss, corruption, or theft, and is easily accessible as needed.

### Database Administrator tasks and responsibilities

Some of the most common roles and responsibilities of a database administrator include:

- Installing and upgrading the database server and application tools.
- Modifying the database structure, as necessary, from information given by application developers.
- Enrolling users and maintaining system security.
- Controlling and monitoring user access to the database.
- Monitoring and optimizing the performance of the database.
- Planning for backup and recovery of database information.
- Backing up and restoring databases.
- Managing and monitoring data replication.



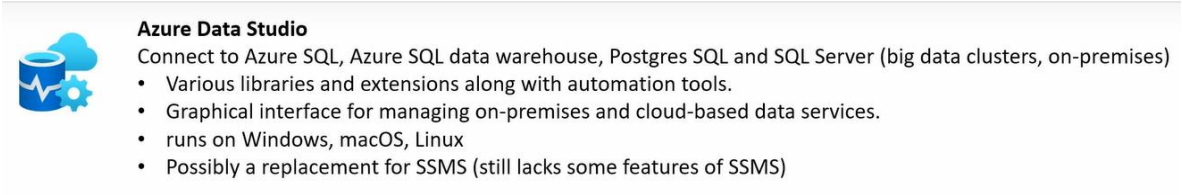
Common database administrator tools

Most database management systems provide their own set of tools to assist with database administration. For example, SQL Server Database Administrators use SQL Server Management Studio for most of their day-to-day database maintenance activities.

**What is Azure Data Studio?**

Azure Data Studio provides a graphical user interface for managing many different database systems. It currently provides connections to on-premises SQL Server databases, Azure SQL Database, PostgreSQL, Azure SQL Data Warehouse, and SQL Server Big Data Clusters, amongst others.

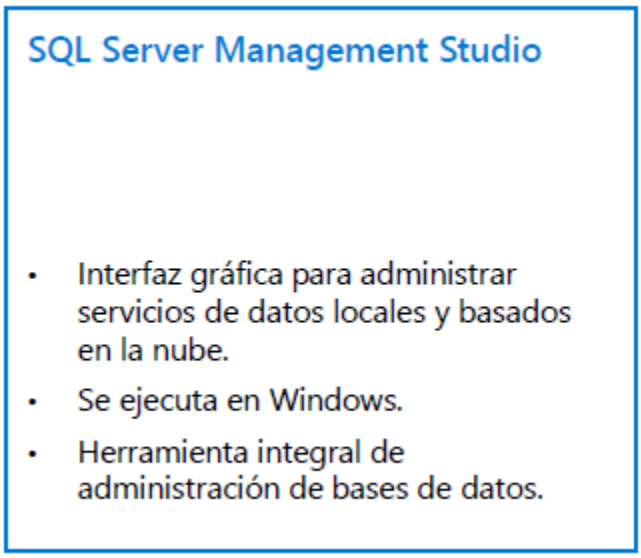
It's an extensible tool, and you can download and install extensions from third-party developers that connect to other systems or provide wizards that help to automate many administrative tasks.




## What is SQL Server Management Studio?

SQL Server Management Studio provides a graphical interface, enabling you to query data, perform general database administration tasks, and generate scripts for automating database maintenance and support operations.

A useful feature of SQL Server Management Studio is the ability to generate Transact-SQL scripts for almost all of the functionality that SQL Server Management Studio provides. This gives the DBA the ability to schedule and automate many common tasks.

## Use the Azure portal to manage Azure SQL Database

Azure SQL database provides database services in Azure. It's similar to SQL Server, except that it runs in the cloud. You can manage Azure SQL database using Azure portal.

Typical configuration tasks such as increasing the database size, creating a new database, and deleting an existing database are done using the Azure portal.

You can use the Azure portal to dynamically manage and adjust resources such as the data storage size and the number of cores available for the database processing.

**Azure Portal and CLI**

- Manage SQL database configurations. eg create, deleting, resizing, number of cores
- Manage and provision other Azure Data Services
- Automate the creating, updating or modifying resources via Azure Resource Manager templates (IaC)

**Azure Portal/CLI**

- Herramientas para la administración y el aprovisionamiento de Azure Data Services.
- Ejecución manual y automatizada de scripts usando Azure Resource Manager o interfaz de scripting de la línea de comandos.

# Tasks and tools for data engineering

Data engineers are tasked with managing and organizing data, while also monitoring for trends or inconsistencies that will impact business goals. Data engineers also need soft skills to communicate data trends to others in the organization and to help the business make use of the data it collects.

## Data Engineer tasks and responsibilities

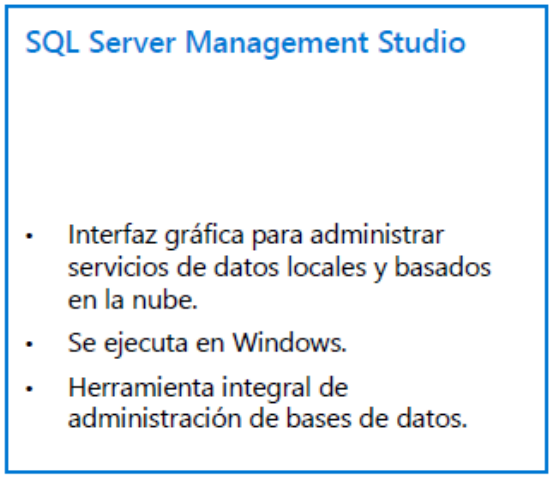
Some of the most common roles and responsibilities of a data engineer include:

- Developing, constructing, testing, and maintaining databases and data structures.
- Developing processes for creating and retrieving information from data sets.
- Using programming languages and tools to examine the data.
- Identifying ways to improve data reliability, efficiency, and quality.
- Deploying sophisticated analytics programs, machine learning, and statistical methods.
- Preparing data for predictive and prescriptive modeling.
- Using data to discover tasks that can be automated.

**Common data engineering tools**

You must have a thorough understanding of the architecture of the database management system, the platform on which the system runs, and the business requirements for the data being stored in the database.

If you're using a relational database management system, you need to be fluent in SQL. You must be able to use SQL to create databases, tables, indexes, views, and the other objects required by the database. Many database management systems provide tools that enable you to create and run SQL scripts. For example, SQL Server Management Studio.

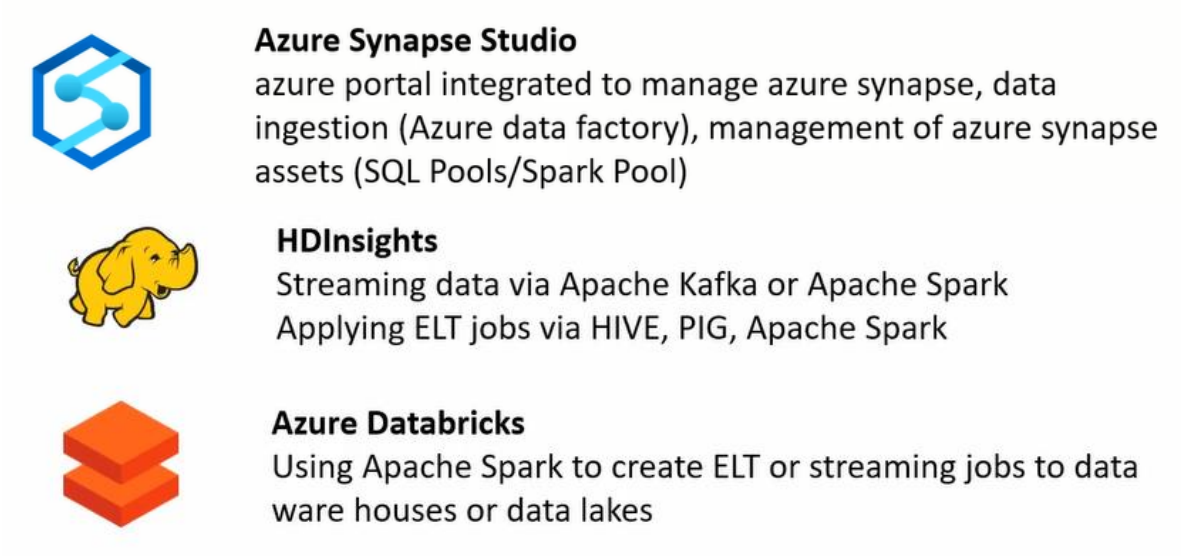
Many database management systems provide a command-line interface that supports these operations. For example, you can use the **sqlcmd** utility to connect to Microsoft SQL Server and Azure SQL Database and run ad-hoc queries and commands.

**Azure CLI**
Support operations SQL cmd to connect to Microsoft server Azure SQL data and run a talk queries and commands

**Azure Portal/CLI**

- Herramientas para la administración y el aprovisionamiento de recursos de Azure.
- Ejecución manual y automatizada de scripts usando Azure Resource Manager o interfaz de scripting de la línea de comandos.

As a SQL Server professional, your primary data manipulation tool might be Transact-SQL. As a data engineer you might use additional technologies, such as Azure Databricks, and Azure HDInsight to generate and test predictive models. If you're working in the non-relational field, you might use Azure Cosmos DB as your primary data store.



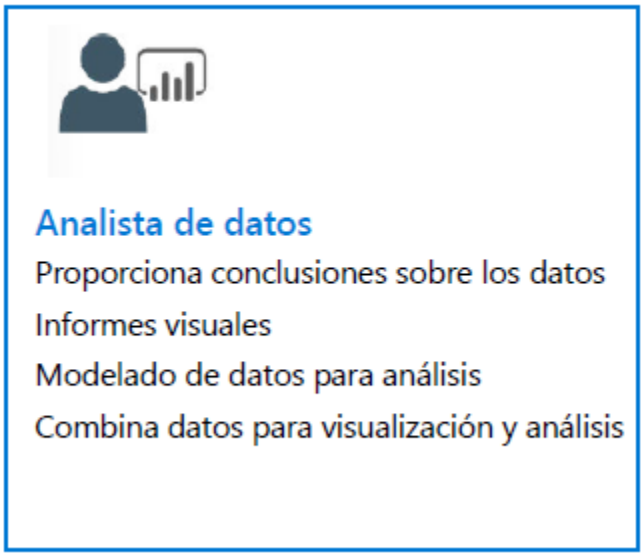

# Tasks and tools for data visualization and reporting

Data analysts are responsible for understanding what data actually means. A skilled data analyst will explore the data and use it to determine trends, issues, and gain other insights that might be of benefit to the company.

Data visualization is key to presenting large amounts of information in ways that are universally understandable or easy to interpret and spot patterns, trends, and correlations. These representations include charts, graphs, infographics, and other pictorial diagrams.

## Data Analyst tasks and responsibilities

- Making large or complex data more accessible, understandable, and usable.
- Creating charts and graphs, histograms, geographical maps, and other visual models that help to explain the meaning of large volumes of data, and isolate areas of interest.
- Combining the data result sets across multiple sources.
- Finding hidden patterns using data.



Common data visualization tools

Many analysts now use Microsoft Power BI, a powerful visualization platform, to create rich, graphical dashboards and reports over data that can vary dynamically.

Power BI is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive, and interactive insights.

Power BI lets you easily connect to your data sources, discover what's important in that data, and share your findings with others in the organization.

**Power BI Desktop**
A stand alone application for data visualization
You can do data modelling
Connect to many data sources
Create interactive reports

**Power BI Portal/Power BI Service**
A web ui for creating interactive dashboards

**Power BI Report builder**
Create paginated reports (printable reports)

| Power BI Desktop | Portal de Power BI/Servicio Power BI | Power BI Report Builder |
|---|---|---|
| • Herramienta de visualización de datos.<br>• Modelar y visualizar datos.<br>• Administración de activos de Azure Synapse (grupos de SQL/grupo de Spark). | • Crear y administrar informes de Power BI.<br>• Crear paneles de Power BI.<br>• Compartir informes/conjuntos de datos. | • Herramienta de visualización de datos para informes paginados.<br>• Modelar y visualizar informes paginados. |

# Describe concepts of relational data

## The characteristics of relational data

One of the main benefits of computer databases is that they make it easy to store information so it's quick and easy to find. A relational database provides a model for storing the data, and a query capability that enables you to retrieve data quickly.

### Understand the characteristics of relational data

In a relational database, you model collections of entities from the real world as tables. An entity is described as a thing about which information needs to be known or held. A table contains rows, and each row represents a single instance of an entity.

In the ecommerce scenario:

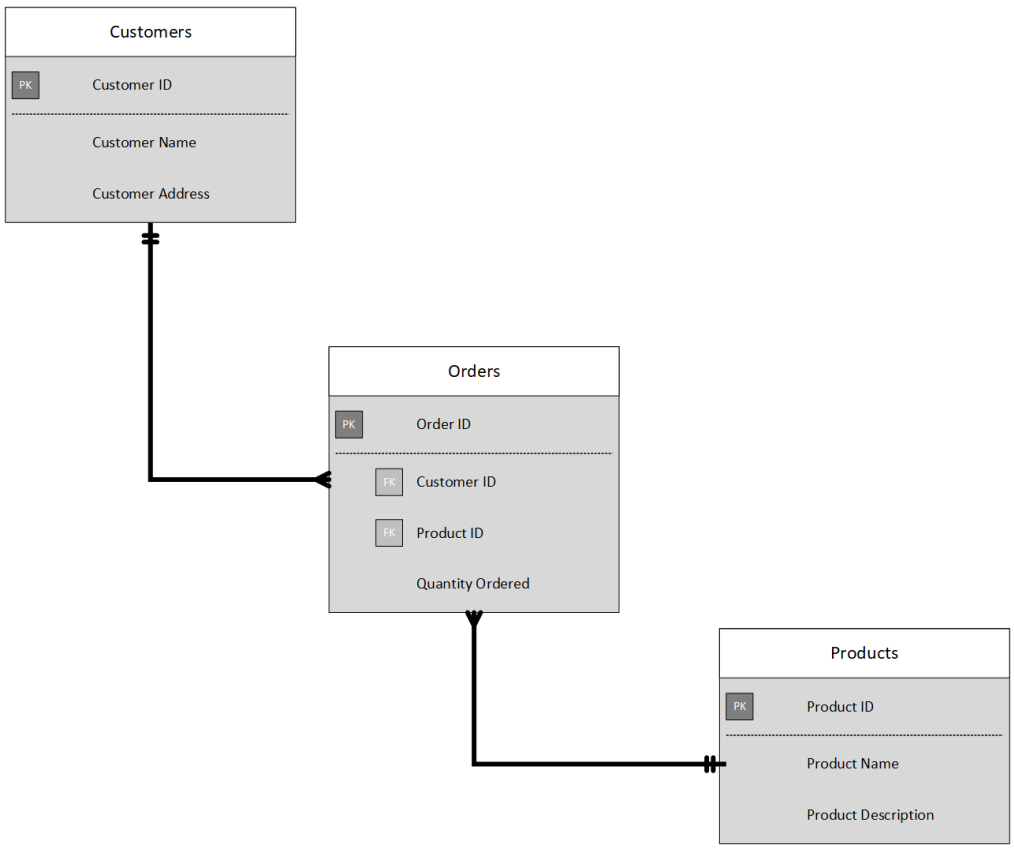
- Each row in the customers table contains the data for a single customer.
- Each row in the products table defines a single product.
- each row in the orders table represents an order made by a customer.

The rows in a table have one or more columns that define the properties of the entity, such as the customer's name, or product ID. All rows in the same table have the same columns. Some columns are used to maintain relationships between tables.

You design a relational database by creating a data model. The model below shows the structure of the entities from the previous example. The primary key indicates the column (or combination of columns) that uniquely identify each row. Every table should have a primary key.

The diagram also shows the relationships between the tables. The lines connecting the tables indicate the type of relationship. In this case, the relationship from customers to orders is 1-to-many (one customer can place many orders, but each order is for a single customer). Similarly, the relationship between orders and products is many-to-1 (several orders might be for the same product).

The columns marked FK are Foreign Key columns. They reference, or link to, the primary key of another table, and are used to maintain the relationships between tables. A foreign key also helps to identify and prevent anomalies.



The main characteristics of a relational database are:

- All data is tabular. Entities are modeled as tables, each instance of an entity is a row in the table, and each property is defined as a column.
- All rows in the same table have the same set of columns.
- A table can contain any number of rows.
- A primary key uniquely identifies each row in a table. No two rows can share the same primary key.
- A foreign key references rows in another, related table. For each value in the foreign key column, there should be a row with the same value in the corresponding primary key column in the other table.

Most relational databases support Structured Query Language (SQL). You use SQL to create tables, insert, update, and delete rows in tables, and to query data.

- You use the CREATE TABLE command to create a table
- The INSERT statement to store data in a table
- The UPDATE statement to modify data in a table
- The DELETE statement to remove rows from a table.
- The SELECT statement retrieves data from a table.

Rather than retrieve every row, you can filter data by using a WHERE clause. You can combine the data from multiple tables in a query using a join operation. A join operation spans the relationships between tables, enabling you to retrieve the data from more than one table at a time.

Explore relational database use cases

You can use a relational database any time you can easily model your data as a collection of tables with a fixed set of columns.

Relational databases are commonly used in ecommerce systems, but one of the major use cases for using relational databases is Online Transaction Processing (OLTP). OLTP applications are focused on transaction-oriented tasks that process a very large number of transactions per minute.

Relational databases are well suited for OLTP applications because they naturally support insert, update, and delete operations.

Examples of OLTP applications that use relational databases are:

- Banking solutions
- Online retail applications
- Flight reservation systems
- Many online purchasing applications.

## Tablas

| Clientes | | |
|---|---|---|
| CustomerID | CustomerName | CustomerPhone |
| 100 | Mateo Lara | *XXX-XXX-XXXX* |
| 101 | Noam Maoz | *XXX-XXX-XXXX* |
| 102 | Vanja Matkovic | *XXX-XXX-XXXX* |
| 103 | Qamar Mounir | *XXX-XXX-XXXX* |
| 104 | Oscar Zamora | *XXX-XXX-XXXX* |
| 105 | Alexander Romero | *XXX-XXX-XXXX* |
| 106 | Eduardo Ponce | *XXX-XXX-XXXX* |
| 107 | Francisco Robles | *XXX-XXX-XXXX* |

Los datos se almacenan en una tabla

La tabla consta de filas y columnas

Todas las filas tienen el mismo número de columnas

Cada columna está definida por un tipo de dato

# Explore relational data structures

A relational database comprises a set of tables. Apart from tables, a typical relational database contains other structures that help to optimize data organization and improve the speed of access.

## What is an index?

An index helps you search for data in a table. Think of an index over a table like an index at the back of a book. A book index contains a sorted set of references, with the pages on which each reference occurs. When you want to find a reference to an item in the book, you look it up through the index.

When you create an index in a database, you specify a column from the table, and the index contains a copy of this data in a sorted order, with pointers to the corresponding rows in the table. When the user runs a query that specifies this column in the WHERE clause, the database management system can use this index to fetch the data more quickly than if it had to scan through the entire table row by row.

You can create many indexes on a table. However, indexes aren't free. An index might consume additional storage space, and each time you insert, update, or delete data in a table, the indexes for that table must be maintained.

Some relational database management systems also support clustered indexes. A clustered index physically reorganizes a table by the index key. In database management systems that support them, a table can only have a single clustered index.

# Índices

**Clientes**

| CustomerID | CustomerName | CustomerPhone |
|---|---|---|
| 100 | Mateo Lara | *XXX-XXX-XXXX* |
| 101 | Nicolás Escobedo | *XXX-XXX-XXXX* |
| 102 | Vicente Morales | *XXX-XXX-XXXX* |
| 103 | Mónica Madera | *XXX-XXX-XXXX* |
| 104 | Oscar Zamora | *XXX-XXX-XXXX* |
| 105 | Alexander Romero | *XXX-XXX-XXXX* |
| 106 | Eduardo Ponce | *XXX-XXX-XXXX* |

**IDX-CustomerRegion**

| CustomerID | Región |
|---|---|
| 100 | Francia |
| 101 | Brasil |
| 102 | Croacia |
| 103 | Jordán |
| 104 | España |
| 105 | Francia |
| 106 | EE. UU. |

**Un índice**

- Optimiza las consultas de búsqueda para una recuperación de datos más rápida.
- Reduce la cantidad de páginas de datos que deben leerse para recuperar los datos en una instrucción SQL.
- Los datos se recuperan uniendo tablas en una consulta.

## What is a view?

A view is a virtual table based on the result set of a query. In the simplest case, you can think of a view as a window on specified rows in an underlying table.

You can query the view and filter the data in much the same way as a table. A view can also join tables together.

## Vista

**Clientes**

| CustomerID | CustomerName | CustomerPhone |
|---|---|---|
| 100 | Mateo Lara | *XXX-XXX-XXXX* |
| 101 | Nicolás Escobedo | *XXX-XXX-XXXX* |
| 102 | Vicente Morales | *XXX-XXX-XXXX* |
| 103 | Mónica Madera | *XXX-XXX-XXXX* |
| 104 | Oscar Zamora | *XXX-XXX-XXXX* |
| 105 | Alexander Romero | *XXX-XXX-XXXX* |
| 106 | Eduardo Ponce | *XXX-XXX-XXXX* |

**Una vista es una tabla virtual basada en el conjunto de resultados de la consulta**

- Las vistas se crean para simplificar la consulta.
- Combinan datos relacionales en una vista única de panel.

**Pedidos**

| OrderID | CustomerID | SalesPersonID |
|---|---|---|
| AD100 | 101 | *200* |
| AD101 | 101 | 200 |
| AD102 | 101 | 200 |
| AX103 | 103 | 201 |
| AS104 | 102 | 201 |
| AR105 | | |
| MK106 | | |
| DB205 | | |

Crear la definición de una vista:

```
CREATE VIEW vw_customerorders AS
SELECT Customers.CustomerID,
Customers.CustomerName, Orders.OrderID FROM
Customers JOIN Orders on Customers.CustomerID
= Orders.CustomerID
```

Recupere los pedidos realizados por el cliente 102 utilizando la vista:

```
SELECT CustomerName, OrderID from
vw_customerorders WHERE CustomerID=102
```

## Choose the right platform for a relational workload

Relational database management systems are one example of where the cloud has enabled organizations to take advantage of improved scalability. However, this scalability has to be balanced against the need for close control over the data.

### Compare on-premises hosting to the cloud

Hosting a relational database on-premises requires that an enterprise not only purchases the database software, but also maintains the necessary hardware on which to run the database. The organization is responsible for maintaining the hardware and software, applying patches, backing up databases, restoring them when necessary, and generally performing all day-to-day management required to keep the platform operational.

A cloud-based approach uses virtual technology to host a company's applications offsite. There are no capital expenses, data can be backed up regularly, and companies only have to pay for the resources they use.

| | On-premises | Cloud |
|---|---|---|
| Personal control of data security | X | |
| Scalable | | X |
| Hardware maintained | | X |
| Software maintained | | X |
| Low capital expenditure | | X |
| Low operational expenditure | X | |

# Understand IaaS and PaaS

You generally have two options when moving your operations and databases to the cloud. You can select an IaaS approach, or PaaS.

**IaaS** is an acronym for Infrastructure-as-a-Service. You can think of IaaS as a transition to fully managed operations in the cloud; you don't have to worry about the hardware but running and managing the software is still very much your responsibility.

You can run any software for which you have the appropriate licenses using this approach. You're not restricted to any specific database management system.

The IaaS approach is best for migrations and applications requiring operating system-level access. SQL virtual machines are lift-and-shift. That is, you can copy your on-premises solution directly to a virtual machine in the cloud.

**PaaS** stands for Platform-as-a-service. Rather than creating a virtual infrastructure, and installing and managing the database software yourself, a PaaS solution does this for you. You specify the resources that you require, and Azure automatically creates the necessary virtual machines, networks, and other devices for you.

Azure offers several PaaS solutions for relational databases, include Azure SQL Database, Azure Database for PostgreSQL, Azure Database for MySQL, and Azure Database for MariaDB. You just connect to them, create your databases, and upload your data.

# Explore concepts of non-relational data

## Explore characteristics of non-relational data

This is a common scenario in systems that consume data from a wide variety of sources, such as data ingestion pipelines. In these situations, a non-relational database can prove extremely useful.

### What are the characteristics of non-relational data?

You use a database to model some aspect of the real-world. Entities in the real-world often have highly variable structures. For example, in an ecommerce database that stores information about customers, how many telephone numbers does a customer have? In another scenario, if you are ingesting data rapidly, you want to capture the data and save it very quickly.

A key aspect of non-relational databases is that they enable you to store data in a very flexible manner. Non-relational databases don't impose a schema on data.

In a non-relational system, you store the information for entities in collections or containers rather than relational tables. Two entities in the same collection can have a different set of fields rather than a regular set of columns found in a relational table.

The data retrieval capabilities of a non-relational database can vary. Each entity should have a unique key value. The entities in a collection are usually stored in key-value order.

More advanced non-relational systems support indexing, in a similar manner to an index in a relational database. Queries can then use the index to identify and fetch data based on non-key fields.

# Explorar las características de los datos no relacionales

## Entidades

```
## Cliente 1 ID: 1
Nombre: Arturo Martinez
Teléfono: [ Particular: 1-999-9999999, trabajo: 1-888-8888888, móvil: 1-777- 7777777 ]
Dirección: [ Particular: 121 Main Street, Alguna ciudad, NY, 10110,
           Trabajo: 87 Big Building, Alguna ciudad, NY, 10111 ]
## Cliente 2 ID: 2
Título: Señor
Nombre: Jorge Salgado
Teléfono: [ Particular: 0044-1999-333333, móvil: 0044-17545-444444 ]
Dirección: [ Reino Unido: 86 High Street, Ciudad, Condado, GL8888, Reino Unido,
           EE. UU.: 777 7th Street, Otra ciudad, CA, 90111 ]
```

**Las colecciones no relacionales:**

- pueden tener varias entidades en la misma colección o contenedor con campos diferentes,
- pueden tener un esquema diferente no tabular,
- se suelen definir etiquetando cada campo con el nombre que representan.

Identify non-relational database use cases

**IoT and telematics:** These systems typically ingest large amounts of data in frequent bursts of activity. Non-relational databases can store this information very quickly.

**Retail and marketing:** It's also used in the retail industry for storing catalog data and for event sourcing in order processing pipelines.

**Gaming:** The database tier is a crucial component of gaming applications. A game database needs to be fast and be able to handle massive spikes in request rates during new game launches and feature updates.

**Web and mobile applications:** A non-relational database such as Azure Cosmos DB are commonly used within web and mobile applications, and is well suited for modeling social interactions, integrating with third-party services, and for building rich personalized experiences.

## Identificar casos de uso de bases de datos no relacionales

**IoT y telemática**

A menudo requieren ingerir grandes cantidades de datos en ráfagas frecuentes de actividad, los datos son semiestructurados o estructurados, a menudo requieren procesamiento en tiempo real

**Comercio minorista y marketing**

Escenarios comunes para datos distribuidos globalmente, almacenamiento de documentos

**Juegos**

Estadísticas del juego, integración en redes sociales, marcadores, aplicaciones de baja latencia

**Web y móvil**

Se suelen usar con análisis de clics en web, aplicaciones modernas que incluyen bots

# Describe types of non-relational data

Non-relational data generally falls into two categories: semi-structured and non-structured.

## What is semi-structured data?

Semi-structured data is data that contains fields. The fields don't have to be the same in every entity. You only define the fields that you need on a per-entity basis.

The data must be formatted in such a way that an application can parse and process it. One common way of doing this is to store the data for each entity as a JSON document.

- A **JSON** document is enclosed in curly brackets ({ and }). Each field has a name (a label), followed by a colon, and then the value of the field. Fields can contain simple values, or subdocuments (each starting and ending with curly brackets).
- **Avro** is a row-based format. Each record contains a header that describes the structure of the data in the record. This header is stored as JSON. The data is stored as binary information. An application uses the information in the header to parse the binary data and extract the fields it contains.
- **ORC** (Optimized Row Columnar format) organizes data into columns rather than rows. An ORC file contains stripes of data. Each stripe holds the data for a column or set of columns. A stripe contains an index into the rows in the stripe, the data for each row, and a footer that holds statistical information (count, sum, max, min, and so on) for each column.
- **Parquet** is another columnar data format. A Parquet file contains row groups. Data for each column is stored together in the same row group.

## Tipos de datos no relacionales

¿Qué son los datos semiestructurados?

La estructura de datos se define en los mismos datos por medio de campos.
Los tipos de formato/archivo incluyen:

JSON | AVRO | ORC | Parquet

## What is unstructured data?

Unstructured data is data that doesn't naturally contain fields. Examples include video, audio, and other media streams. Each item is an amorphous blob of binary data. You can't search for specific elements in this data.

You might choose to store data such as this in storage that is specifically designed for the purpose. A block blob only supports basic read and write operations.

You could also consider files as a form of unstructured data, although in some cases a file might include metadata that indicates what type of file it is (photograph, Word document, Excel spreadsheet, and so on), owner, and other elements that could be stored as fields.

### ¿Qué son los datos no estructurados?

- No contiene campos de forma natural
  *Ejemplos: vídeo, audio, streaming de multimedia, documentos*
- A menudo se usa para extraer formularios de datos y categorizar o identificar "estructuras"
- Se usa con frecuencia en combinación con las capacidades de Machine Learning o Cognitive Services para "extraer datos" mediante:
  - Text Analytics
  - Análisis de sentimiento con API cognitivas
  - Vision API

# Describe types of non-relational and NoSQL databases

Non-relational data is an all-encompassing term that means anything not structured as a set of tables. There are many different types of non-structured data, and the information is used for a wide variety of purposes.

## What is NoSQL?

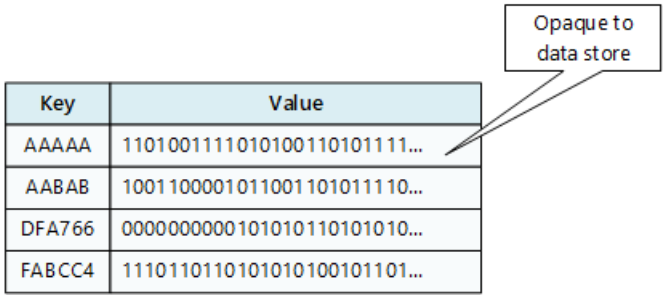
You might see the term NoSQL when reading about non-relational databases. NoSQL is a rather loose term that simply means non-relational. Some non-relational databases support a version of SQL adapted for documents rather than tables.

NoSQL (non-relational) databases generally fall into four categories: key-value stores, document databases, column family databases, and graph databases.

## What is a key-value store?

A key-value store is the simplest (and often quickest) type of NoSQL database for inserting and querying data. Each data item in a key-value store has two elements, a key, and a value.

The key uniquely identifies the item, and the value holds the data for the item. The value is opaque to the database management system. Items are stored in key order.



The focus of a key-value store is the ability to read and write data very quickly. Search capabilities are secondary. A key-value store is an excellent choice for data ingestion, when a large volume of data arrives as a continual stream and must be stored immediately.

## What is a document database?

In a document database, each document has a unique ID, but the fields in the documents are transparent to the database management system. Document databases typically store data in JSON format.

The fields in documents are exposed to the storage management system, enabling an application to query and filter data by using the values in these fields.

A document store does not require that all documents have the same structure. This free-form approach provides a great deal of flexibility.

| Key | Document |
|---|---|
| 1001 | { "CustomerID": 99, "OrderItems": [ { "ProductID": 2010, "Quantity": 2, "Cost": 520 }, { "ProductID": 4365, "Quantity": 1, "Cost": 18 }], "OrderDate": "04/01/2017" } |
| 1002 | { "CustomerID": 220, "OrderItems": [ { "ProductID": 1285, "Quantity": 1, "Cost": 120 }], "OrderDate": "05/08/2017" } |

An application can retrieve documents by using the document key. The key is a unique identifier for the document. The application can also query documents based on the value of one or more fields. Some document databases support indexing to facilitate fast lookup of documents based on one or more indexed fields.

Most document databases will ingest large volumes of data more rapidly than a relational database but aren't as optimal as a key-value store for this type of processing. The focus of a document database is its query capabilities.

## What is a column family database?

A column family database organizes data into rows and columns. Examples of this structure include ORC and Parquet files. The real power of a column family database lies in its denormalized approach to structuring sparse data.

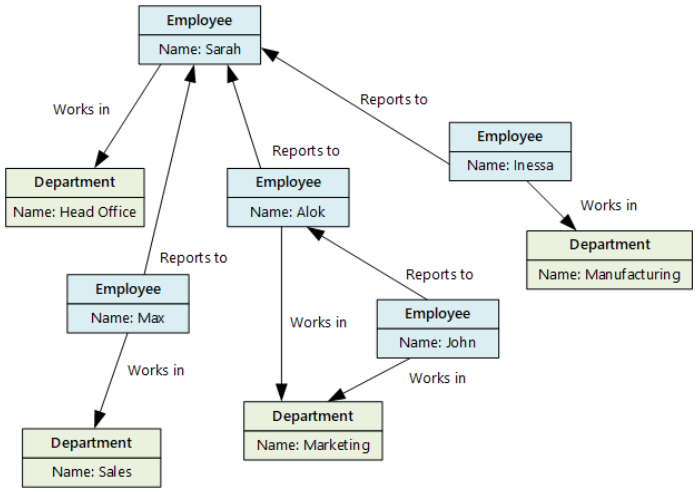
You can think of a column family database as holding tabular data comprising rows and columns, but you can divide the columns into groups known as column-families. Each column family holds a set of columns that are logically related together.

| Row Key | Column Families | |
|---|---|---|
| CustomerID | CustomerInfo | AddressInfo |
| 1 | CustomerInfo:Title Mr<br>CustomerInfo:FirstName Mark<br>CustomerInfo:LastName Hanson | AddressInfo:StreetAddress 999 500th Ave<br>AddressInfo:City Bellevue<br>AddressInfo:State WA<br>AddressInfo:ZipCode 12345 |
| 2 | CustomerInfo:Title Ms<br>CustomerInfo:FirstName Lisa<br>CustomerInfo:LastName Andrews | AddressInfo:StreetAddress 888 W. Front St<br>AddressInfo:City Boise<br>AddressInfo:State ID<br>AddressInfo:ZipCode 54321 |
| 3 | CustomerInfo:Title Mr<br>CustomerInfo:FirstName Walter<br>CustomerInfo:LastName Harp | AddressInfo:StreetAddress 999 500th Ave<br>AddressInfo:City Bellevue<br>AddressInfo:State WA<br>AddressInfo:ZipCode 12345 |

What is a graph database?

A graph database stores two types of information: nodes that you can think of as instances of entities, and edges, which specify the relationships between nodes. Nodes and edges can both have properties that provide information about that node or edge (like columns in a table).

The purpose of a graph database is to enable an application to efficiently perform queries that traverse the network of nodes and edges, and to analyze the relationships between entities.

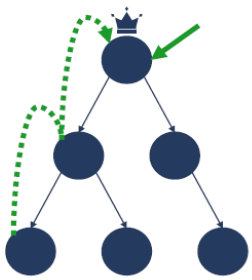



## ¿Qué es una base de datos de grafos?

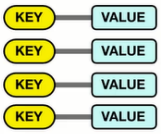
- Almacenes de entidades centradas en relaciones
- Permite que las aplicaciones realicen consultas atravesando una red de nodos y bordes

### ¿Qué aplicaciones requieren una base de datos de grafos?

Requisitos empresariales:
- Aplicaciones OLTP con alta correlación de datos.
- Fácil actualización de uno o varios objetos.
- Modelado de datos flexible.
- Requisitos de datos que evolucionan.
- Estructuras jerárquicas de datos.



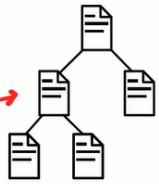
A non-relational database **stores data in a non-tabular form** and will be optimized for different kinds of data-structures.




**Types of non-relational databases:**

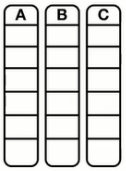
**Key/Value**

- Each value has a key
- Designed to scale
- Only simple lookups



**Document**

- Primary entity is a JSON-like data-structure called a document

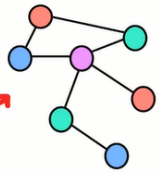



**Columnar**

- Has a table-like structure but data is stored around columns instead of rows

**Graph**

- Data is represented with nodes and structures. Where relationships matter



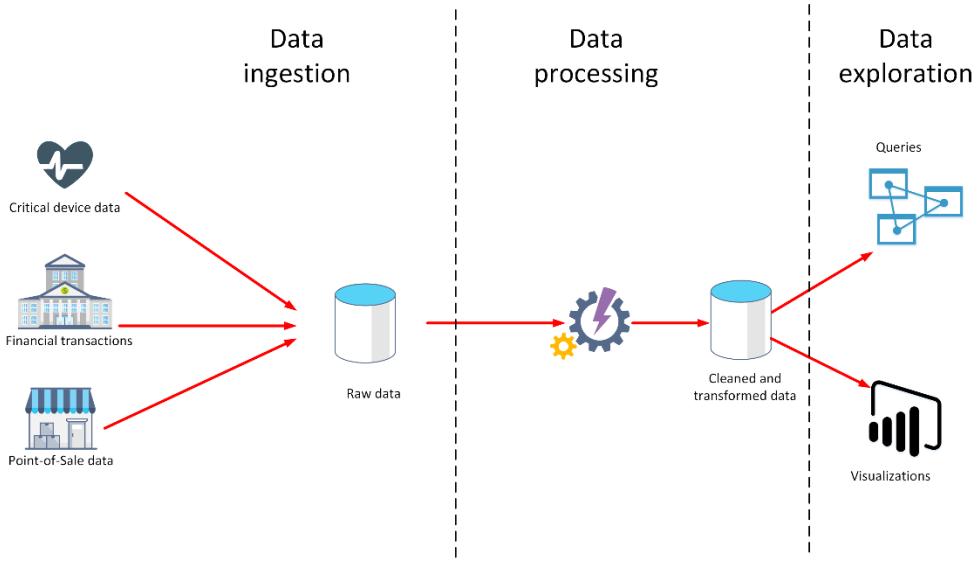
Sometimes non-relational database can be both Key/Value and Document
eg. Azure Cosmo DB or Amazon DynamoDB

# Explore concepts of data analytics

## Describe data ingestion and processing

Data analytics is concerned with taking data and finding meaningful information and inferences from it. Data analytics could help you to identify strengths and weaknesses in your organization and enable you to make appropriate business decisions.

The data a company uses can come from many sources. In a data analytics solution, you combine this data and construct a data warehouse that you can use to ask (and answer) questions about your business operations.




### What is data ingestion?

Data ingestion is the process of obtaining and importing data for immediate use or storage in a database. The data can arrive as a continuous stream, or it may come in batches, depending on the source. The purpose of the ingestion process is to capture this data and store it.

The ingestion process might also perform filtering. For example, ingestion might reject suspicious, corrupt, or duplicated data. Suspicious data might be data arriving from an unexpected source.

It may also be possible to perform some transformations at this stage, converting data into a standard form for later processing.

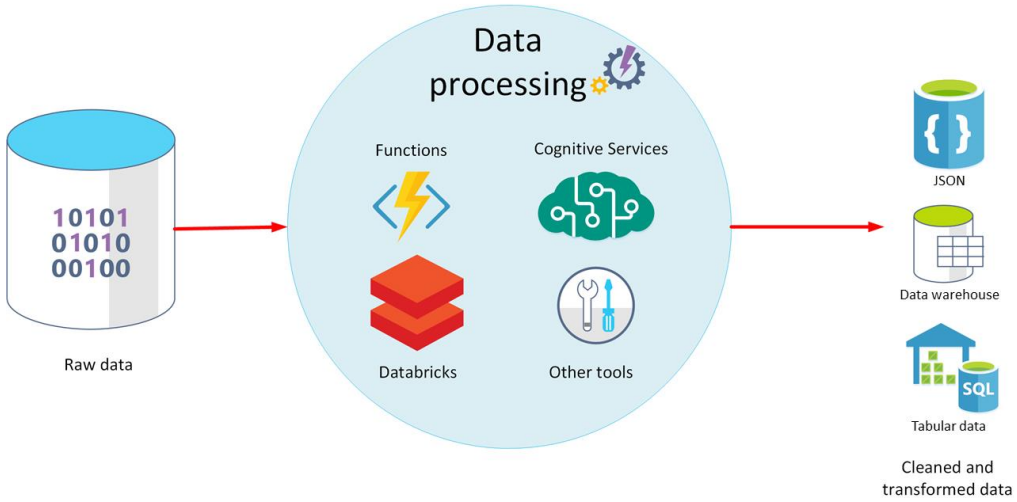
## What is data processing?

Data processing takes the data in its raw form, cleans it, and converts it into a more meaningful format (tables, graphs, documents, and so on).

The result is a database of data that you can use to perform queries and generate visualizations, giving it the form and context necessary to be interpreted by computers and used by employees throughout an organization.

The aim of data processing is to convert the raw data into one or more business models. A business model describes the data in terms of meaningful business entities and may aggregate items together and summarize information.

The data processing stage could also generate predictive or other analytical models from the data.



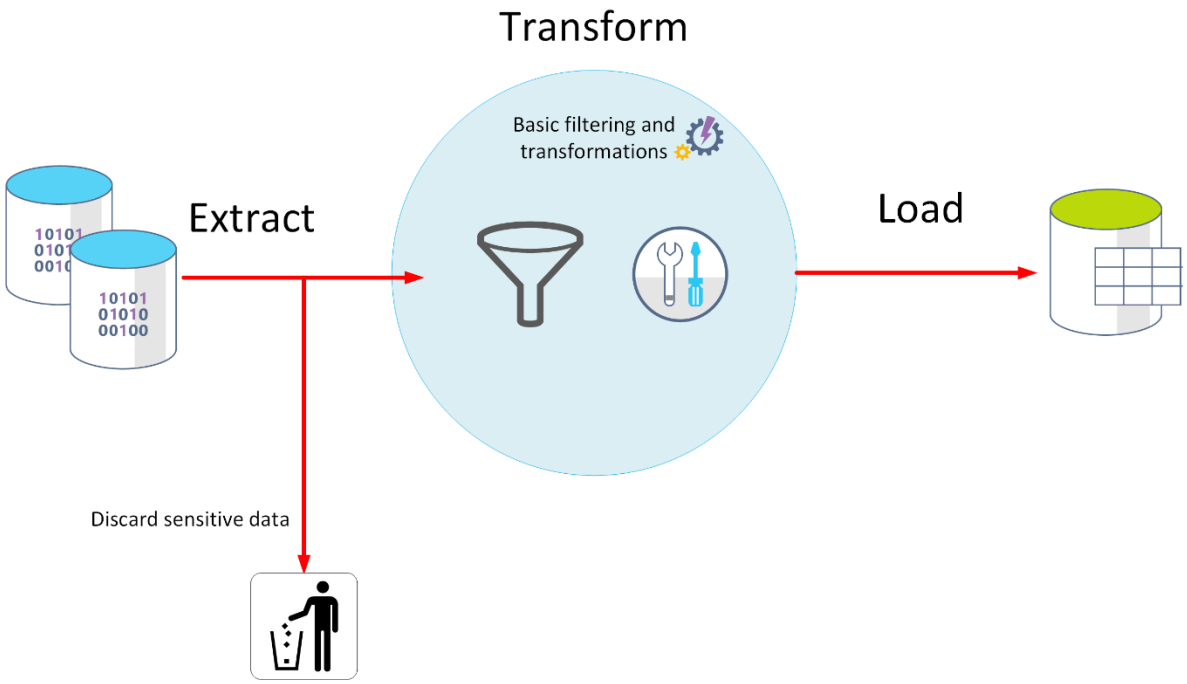
What is ELT and ETL?

The data processing mechanism can take two approaches to retrieving the ingested data, processing this data to transform it and generate models, and then saving the transformed data and models.

**ETL Extract, Transform, and Load.**

The raw data is retrieved and transformed before being saved. The extract, transform, and load steps can be performed as a continuous pipeline of operations.

It is suitable for systems that only require simple models, with little dependency between items. For example, this type of process is often used for basic data cleaning tasks, deduplicating data, and reformatting the contents of individual fields.

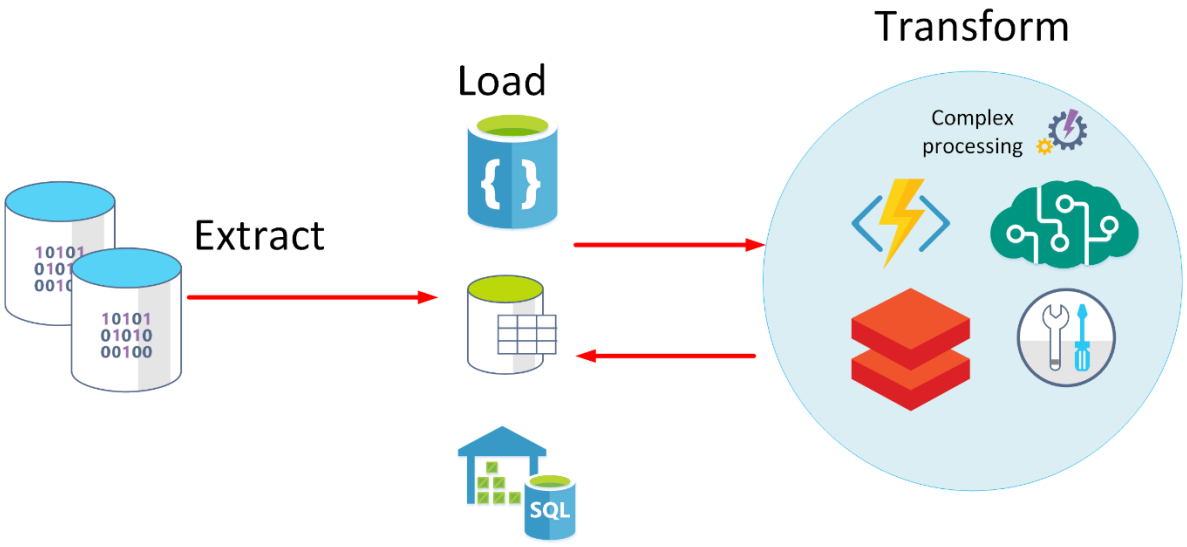


**ELT Extract, Load, and Transform.**

The process differs from ETL in that the data is stored before being transformed. The data processing engine can take an iterative approach, retrieving and processing the data from storage, before writing the transformed data and models back to storage.

ELT is more suitable for constructing complex models that depend on multiple items in the database, often using periodic batch processing.

ELT is a scalable approach that is suitable for the cloud because it can make use of the extensive processing power available. The more stream-oriented approach of ETL places more emphasis on throughput.




| | ETL | ELT |
|---|---|---|
| Improved data privacy and compliance | X | |
| Data lake support | | X |
| Does not require specialist skills | X | |
| Ideal for large volumes of data | | X |

# Explore data visualization

A business model can contain an enormous amount of information. The purpose of producing a model such as this is to help you reason over the information it contains, ask questions, and hopefully obtain answers that can help you drive your business forward.

## What is reporting?

Reporting is the process of organizing data into informational summaries to monitor how different areas of an organization are performing. Reporting helps companies monitor their online business and know when data falls outside of expected ranges.

## What is business intelligence?

The term Business Intelligence (BI) refers to technologies, applications, and practices for the collection, integration, analysis, and presentation of business information. The purpose of business intelligence is to support better decision making.

Business intelligence systems provide historical, current, and predictive views of business operations, most often using data that has been gathered into a data warehouse, and occasionally working from live operational data.

What is data visualization?

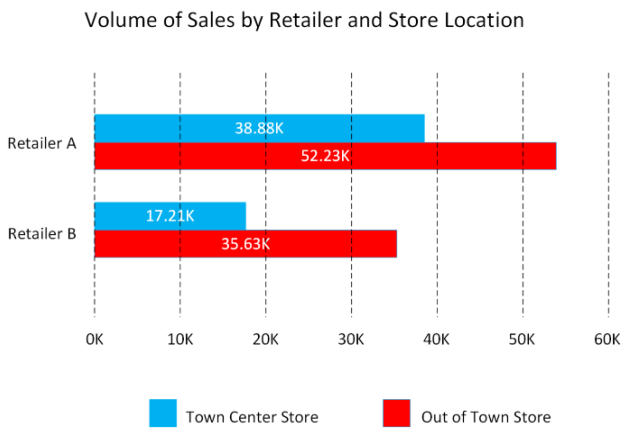
Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to spot and understand trends, outliers, and patterns in data.

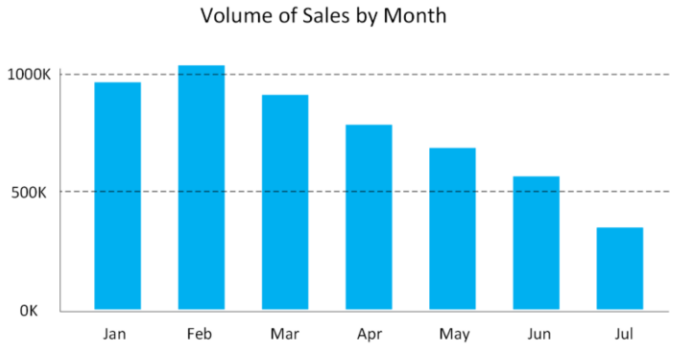
Using Power BI, you can connect to multiple different sources of data, and combine them into a data model. This data model lets you build visuals, and collections of visuals you can share as reports, with other people inside your organization.

A good data visualization enables you to quickly spot trends, anomalies, and potential issues. The most common forms of visualizations are:

**Bar and column charts:**

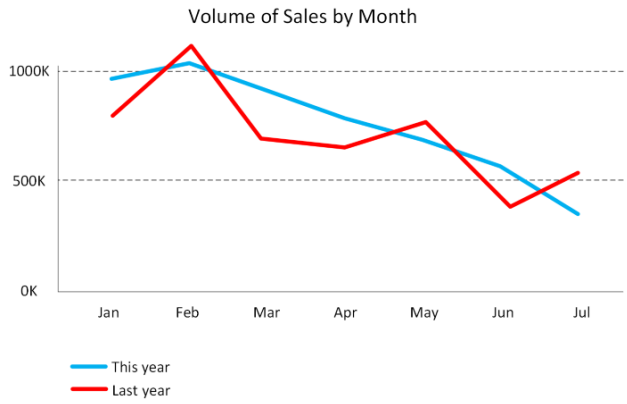
Bar and column charts enable you to see how a set of variables changes across different categories.



**Line charts:**

Line charts emphasize the overall shape of an entire series of values, usually over time.



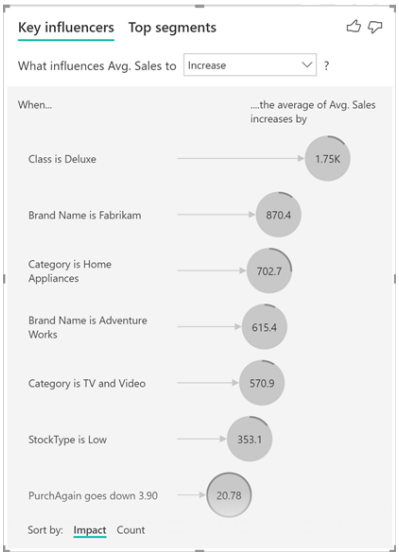

**Matrix:**

A matrix visual is a tabular structure that summarizes data.

| Quarter Year | Q1 Revenue | YTD Revenue | Q2 Revenue | YTD Revenue |
|---|---|---|---|---|
| 2015 | $45,186 | $45,186 | $70,609 | $115,795 |
| 2016 | $52,154 | $52154 | $73,542 | $125,696 |
| 2017 | $51,388 | $51,388 | $68,149 | $118,537 |
| 2018 | $48,281 | $48,281 | $66,853 | $115,134 |
| 2019 | $53,145 | $53,145 | $49,135 | $102,280 |

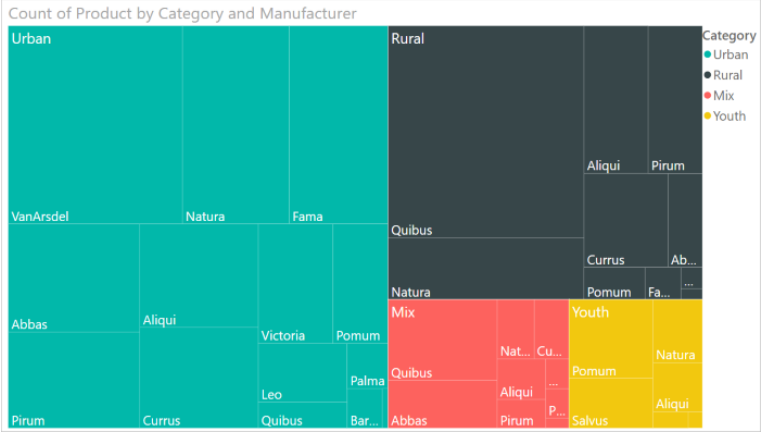
**Key influencers:**

A key influencer chart displays the major contributors to a selected result or value. Key influencers are a great choice to help you understand the factors that influence a key metric.



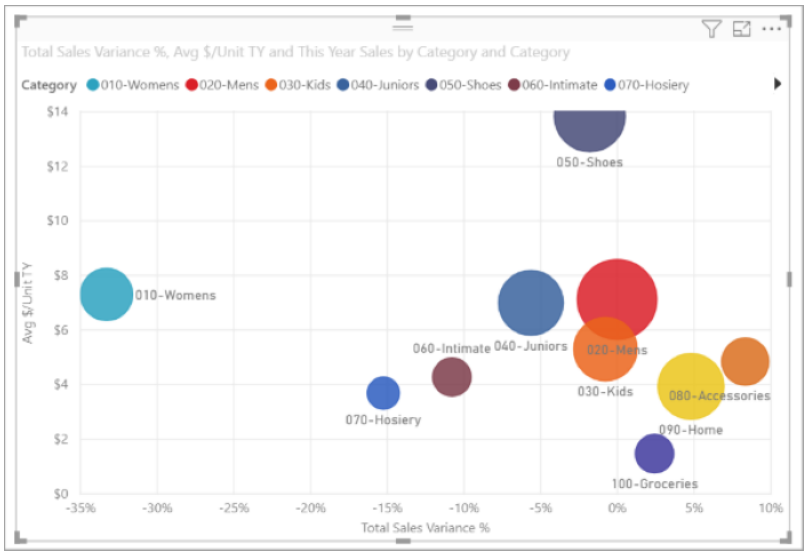
**Treemap:**

Treemaps are charts of colored rectangles, with size representing the relative value of each item. They can be hierarchical, with rectangles nested within the main rectangles.



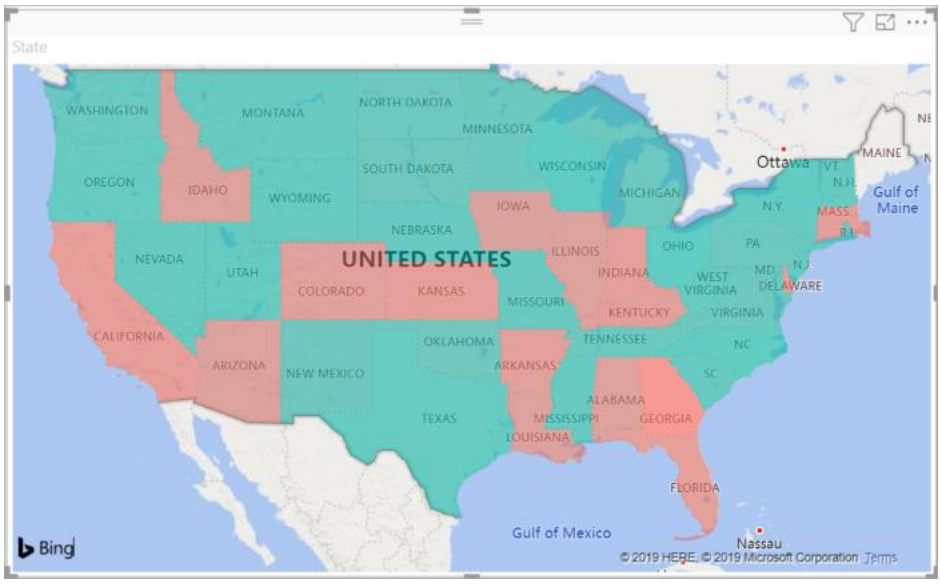

**Scatter:**

A scatter chart shows the relationship between two numerical values. A bubble chart is a scatter chart that replaces data points with bubbles, with the bubble size representing an additional third data dimension.



**Filled map:**

If you have geographical data, you can use a filled map to display how a value differs in proportion across a geography or region.




## Explore data analytics

Data analytics is concerned with examining, transforming, and arranging data so that you can study it and extract useful information. Data analytics is a discipline that covers the entire range of data management tasks.

These tasks not only include analysis, but also data collection, organization, storage, and all the tools and techniques used.

| Data analytics activity | Purpose |
|---|---|
| Descriptive analytics | Helps answer questions about what has happened, based on historical data. |
| Diagnostic analytics | Helps answer questions about why things happened. |
| Predictive analytics | Helps answer questions about what will happen in the future. |
| Prescriptive analytics | Helps answer questions about what actions should be taken to achieve a goal or target. |
| Cognitive analytics | Helps to draw inferences from existing data and patterns |

**Descriptive analytics**

Descriptive analytics helps answer questions about what has happened, based on historical data. Descriptive analytics techniques summarize large datasets to describe outcomes to stakeholders.

Examples of descriptive analytics include generating reports to provide a view of an organization's sales and financial data.

**Diagnostic analytics**

Diagnostic analytics helps answer questions about why things happened. The performance indicators are further investigated to discover why they got better or worse. This generally occurs in three steps:

1. Identify anomalies in the data. These may be unexpected changes in a metric or a particular market.
2. Collect data that's related to these anomalies.
3. Use statistical techniques to discover relationships and trends that explain these anomalies.

**Predictive analytics**

Predictive analytics helps answer questions about what will happen in the future. Predictive analytics techniques use historical data to identify trends and determine if they're likely to recur.

Techniques include a variety of statistical and machine learning techniques such as neural networks, decision trees, and regression.

**Prescriptive analytics**

Prescriptive analytics helps answer questions about what actions should be taken to achieve a goal or target. This technique allows businesses to make informed decisions in the face of uncertainty.

Prescriptive analytics techniques rely on machine learning strategies to find patterns in large datasets.

**Cognitive analytics**

Cognitive analytics attempts to draw inferences from existing data and patterns, derive conclusions based on existing knowledge bases, and then add these findings back into the knowledge base for future inferences.

Cognitive analytics helps you to learn what might happen if circumstances change, and how you might handle these situations.

Effective cognitive analytics depends on machine learning algorithms. It uses several NLP (Natural Language Processing) concepts to make sense of previously untapped data sources, such as call center conversation logs and product reviews.

# Describe how to work with relational data on Azure

## Explore relational data services in Azure

Azure offers a range of options for running a database management system in the cloud. For example, you can migrate your on-premises systems to a collection of Azure virtual machines. This approach still requires that you manage your DBMS carefully.

Alternatively, you can take advantage of the various Azure relational data services available. These data services manage the DBMS for you, leaving you free to concentrate on the data they contain and the applications that use them.

## Understand IaaS, PaaS, and SaaS

### IaaS Infrastructure-as-a-Service

You can create a set of virtual machines, connect them together using a virtual network, and add a range of virtual devices. You take responsibility for installing and configuring the software, such as the DBMS, on these virtual machines.

In many ways, this approach is similar to the way in which you run your systems inside an organization, except that you don't have to concern yourself with buying or maintaining the hardware.

## PaaS Platform-as-a-service

You specify the resources that you require (based on how large you think your databases will be, the number of users, and the performance you require), and Azure automatically creates the necessary virtual machines, networks, and other devices for you.

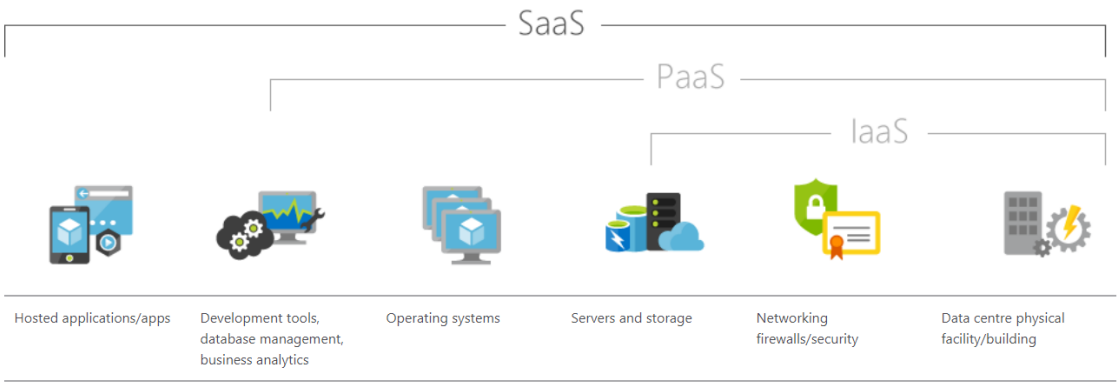
You can usually scale up or down (increase or decrease the size and number of resources) quickly, as the volume of data and the amount of work being done varies.

Azure handles this scaling for you, and you don't have to manually add or remove virtual machines or perform any other form of configuration.

## SaaS Software-as-a-Service

SaaS services are typically specific software packages that are installed and run-on virtual hardware in the cloud. SaaS packages are typically hosted applications rather than more generalized software such as a DBMS.

| | Example | Includes |
|---|---|---|
| IaaS | Azure virtual network | Servers, storage, networking, and physical data center. |
| PaaS | Azure SQL Databases | IaaS plus database management (or other server systems), and operating systems. |
| SaaS | Office 365 | PaaS plus apps. |



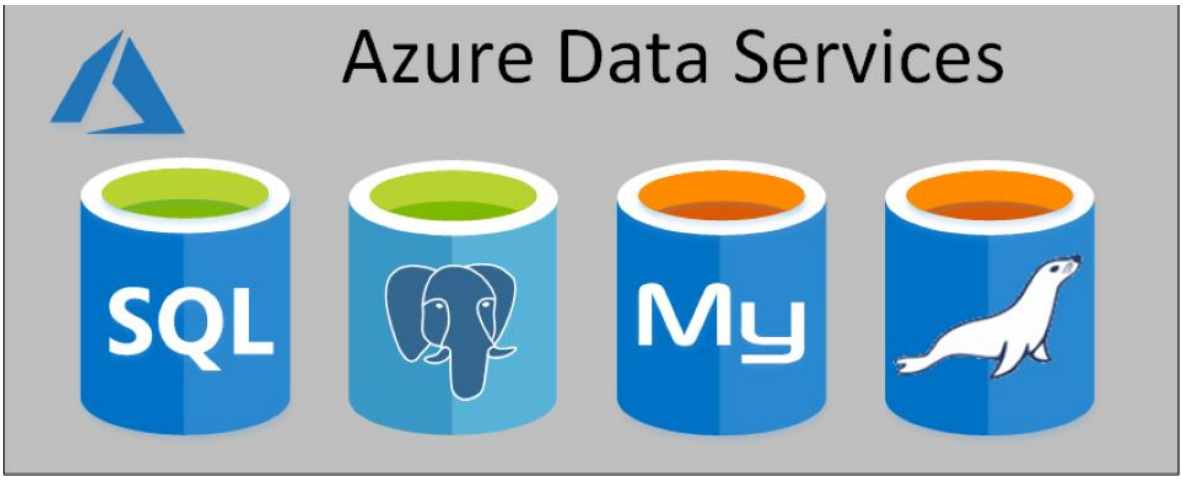
# What are Azure Data Services?

Azure Data Services fall into the PaaS category. These services are a series of DBMSs managed by Microsoft in the cloud.

Each data service takes care of the configuration, day-to-day management, software updates, and security of the databases that it hosts. All you do is create your databases under the control of the data service.

Azure Data Services are available for several common relational database management systems. The most well-known service is Azure SQL Database.

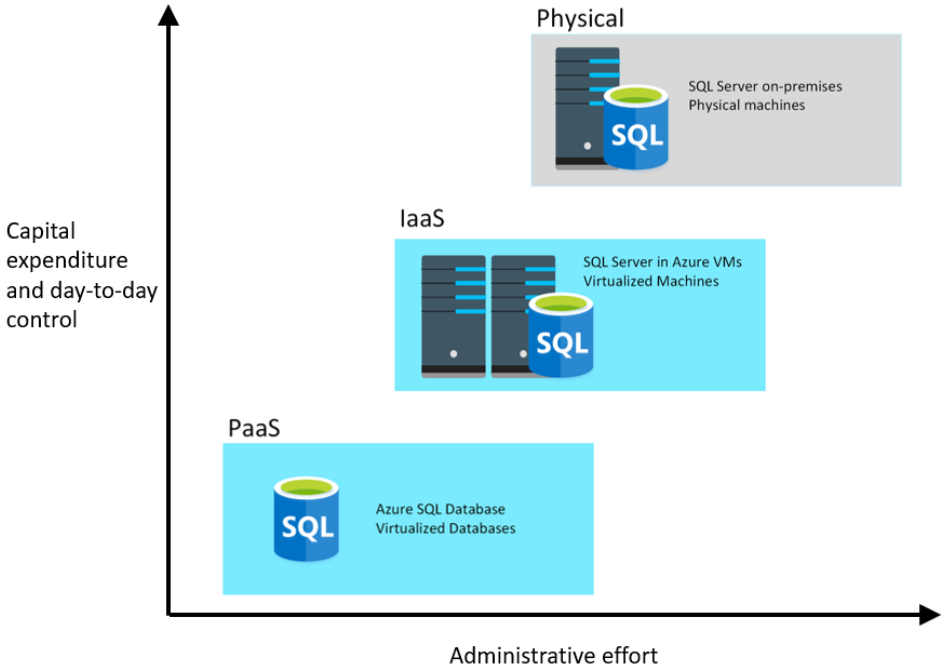
The others currently available are Azure Database for MySQL servers, Azure Database for MariaDB servers, and Azure Database for PostgreSQL servers.




Using Azure Data Services reduces the amount of time that you need to invest to administer a DBMS. However, these services can also limit the range of custom administration tasks that you can perform, because manually performing some tasks might risk compromising the way in which the service runs.

Apart from reducing the administrative workload, Azure Data Services ensure that your databases are available for at least 99.99% of the time.

Not all features of a database management system are available in Azure Data Services. This is because Azure Data Services takes on the task of managing the system and keeping it running using hardware situated in an Azure datacenter.




## ¿Qué es Azure Data Services?

# SQL Server on Azure virtual machines

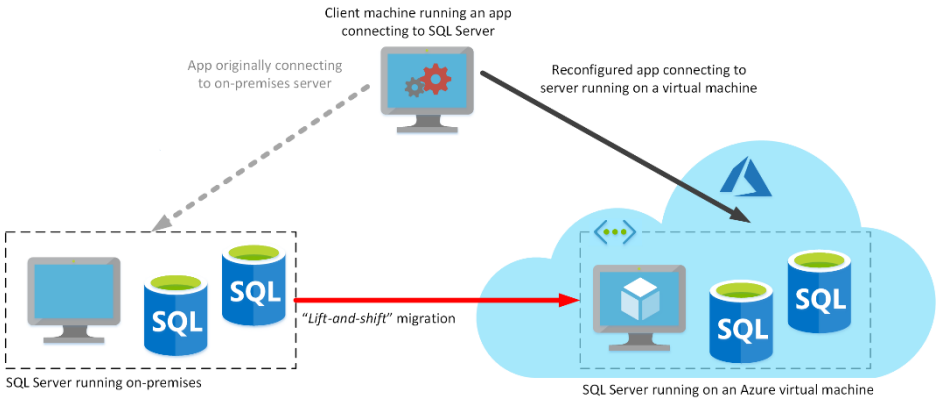
## What is SQL Server on Azure Virtual Machines?

SQL Server on Virtual Machines enables you to use full versions of SQL Server in the Cloud without having to manage any on-premises hardware.

SQL Server running on an Azure virtual machine effectively replicates the database running on real on-premises hardware.

You remain responsible for maintaining the SQL Server software and performing the various administrative tasks to keep the database running from day-to-day.

SQL virtual machines are lift-and-shift ready for existing applications that require fast migration to the cloud with minimal changes.




### Características clave

Acceso al servidor SQL Server y OS

Versiones expansivas de SQL y OS

Windows, Linux, Containers

Flujo de archivos, DTC y modelo de recuperación simple

SSAS, SSRS y SSIS

### Diferenciadores de Azure

Actualizaciones de seguridad extendidas gratuitas para SQL Server 2008/R2

Actualizaciones de seguridad y copias de seguridad automatizadas

Restauración a un momento dado con Azure Backup

Rendimiento de almacenamiento acelerado con Azure Blob Caching

435 % de retorno general de una inversión en IaaS de Azure durante cinco años[1]

## Use cases

This approach is optimized for migrating existing applications to Azure or extending existing on-premises applications to the cloud in hybrid deployments.

You can use SQL Server in a virtual machine to develop and test traditional SQL Server applications.

These capabilities enable you to:

- Create rapid development and test scenarios when you do not want to buy on-premises non-production SQL Server hardware.
- Become lift-and-shift ready for existing applications that require fast migration to the cloud with minimal changes or no changes.
- Scale up the platform on which SQL Server is running, by allocating more memory, CPU power, and disk space to the virtual machine. You can quickly resize an Azure virtual machine without the requirement that you reinstall the software that is running on it.

## Business benefits

Running SQL Server on virtual machines allows you to meet unique and diverse business needs through a combination of on-premises and cloud-hosted deployments, while using the same set of server products, development tools, and expertise across these environments.

Using virtual machines can offer a solution but using them does not eliminate the need to administer your DBMS as carefully as you would on-premises.

# Azure SQL Database

## What is Azure SQL Database?

Azure SQL Database is a PaaS offering from Microsoft. You create a managed database server in the cloud, and then deploy your databases on this server.

Azure SQL Database is available with several options: Single Database, Elastic Pool, and Managed Instance.

### Single Database

This option enables you to quickly set up and run a single SQL Server database. You create and run a database server in the cloud, and you access your database through this server.

Microsoft manages the server, so all you have to do is configure the database, create your tables, and populate them with your data. You can scale the database if you need additional storage space, memory, or processing power.

### Elastic Pool

Multiple databases can share the same resources, such as memory, data storage space, and processing power through multiple tenancy.

The resources are referred to as a pool. You create the pool, and only your databases can use the pool. This model is useful if you have databases with resource requirements that vary over time and can help you to reduce costs.

Elastic Pool enables you to use the resources available in the pool, and then release the resources once processing has completed.

## Use cases

It is not fully compatible with on-premises SQL Server installations. It is often used in new cloud projects where the application design can accommodate any required changes to your applications.

Azure SQL Database is often used for:

- Modern cloud applications that need to use the latest stable SQL Server features.
- Applications that require high availability.
- Systems with a variable load, that need the database server to scale up and down quickly.

## Business benefits

Azure SQL Database automatically updates and patches the SQL Server software to ensure that you are always running the latest and most secure version of the service.

The scalability features of Azure SQL Database ensure that you can increase the resources available to store and process data without having to perform a costly manual upgrade.

The service provides high availability guarantees, to ensure that your databases are available at least 99.99% of the time.

Advanced threat protection provides advanced security capabilities, such as vulnerability assessments, to help detect and remediate potential security problems with your databases.

Auditing tracks database events and writes them to an audit log in your Azure storage account.

SQL Database helps secure your data by providing encryption.

# Azure DB SQL

**Desafío para el cliente**
Quiero compilar aplicaciones modernas, potencialmente multiinquilino, con el mayor tiempo de actividad y un rendimiento predecible.

**Solución**
Azure SQL Database es un servicio de base de datos en la nube altamente escalable con alta disponibilidad y aprendizaje automático integrados

## Características clave

Base de datos única o grupo elástico

Almacenamiento de hiperescala (más de 100 TB)

Informática sin servidor

Servicio totalmente administrado

Soporte de Private Link

Alta disponibilidad con aislamiento de la zona de disponibilidad

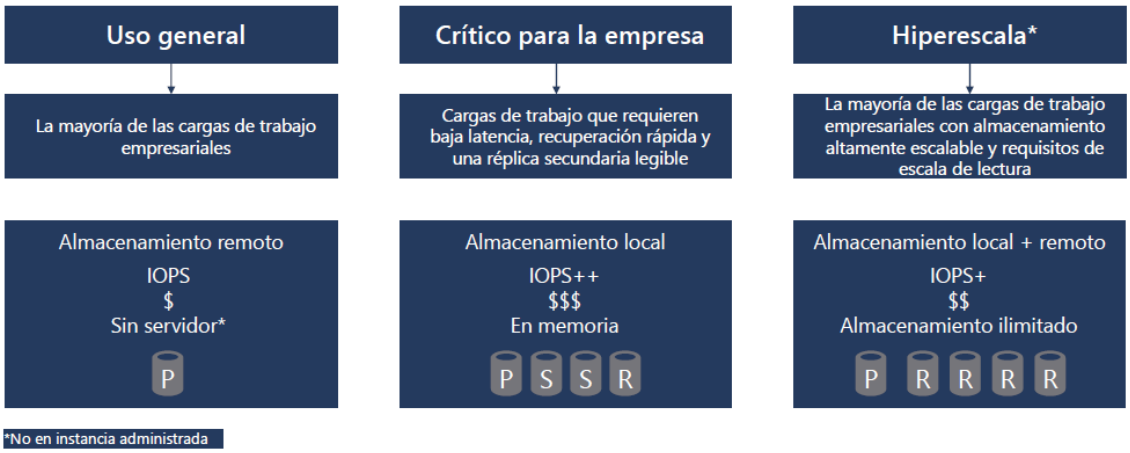
## Diferenciadores de Azure

Acuerdo de Nivel de Servicio de disponibilidad más alta de la industria del 99,995 %

Acuerdo de Nivel de Servicio de continuidad empresarial exclusivo de la industria con RPO de 5 segundos y RTO de 30 segundos

Líder en rendimiento de precios para cargas de trabajo de misión crítica con un coste de hasta un 86 % menos que AWS RDS (GigaOm)

# Niveles de servicio de Azure SQL DB




*No en instancia administrada

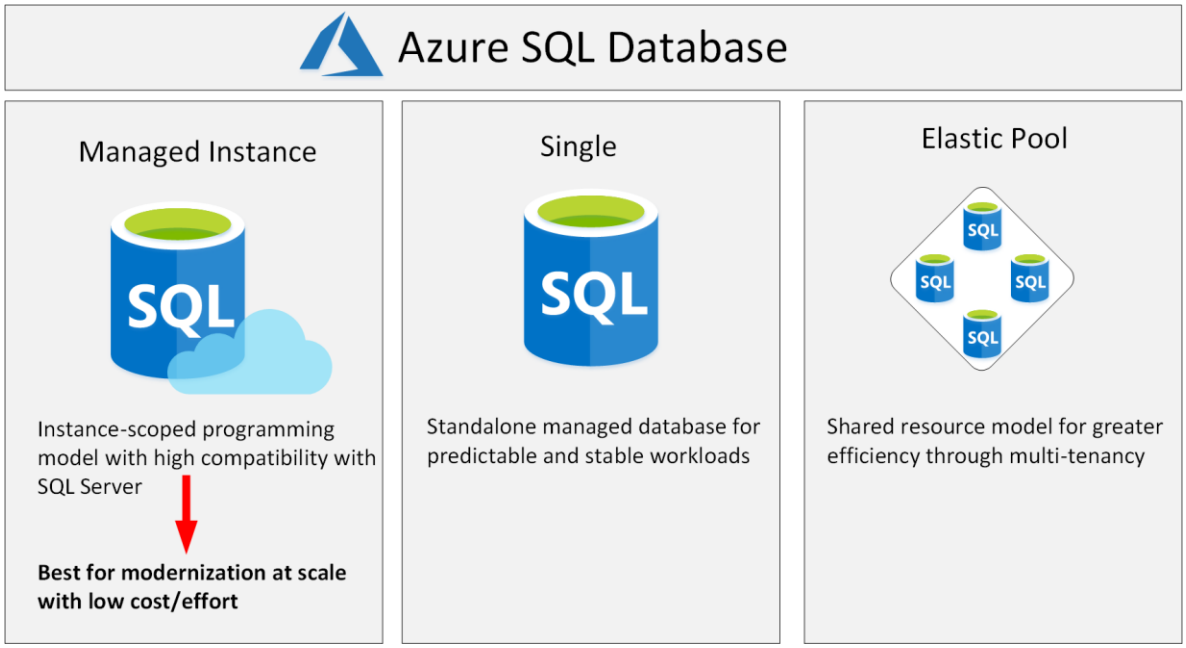
# Azure SQL Database Managed Instance

## What is Azure SQL Database managed instance?

Managed instance effectively runs a fully controllable instance of SQL Server in the cloud. You can install multiple databases on the same instance.

You have complete control over this instance, much as you would for an on-premises server. The Managed instance service automates backups, software patching, database monitoring, and other general tasks, but you have full control over security and resource allocation for your databases.

Managed instances depend on other Azure services such as Azure Storage for backups, Azure Event Hubs for telemetry, Azure Active Directory for authentication, Azure Key Vault for Transparent Data Encryption (TDE) and a couple of Azure platform services that provide security and supportability features. All communications are encrypted and signed using certificates.



## Use cases

Consider Azure SQL Database managed instance if you want to lift-and-shift an on-premises SQL Server instance and all its databases to the cloud, without incurring the management overhead of running SQL Server on a virtual machine.

SQL Database managed instance provides features not available with the Single Database or Elastic Pool options. To check compatibility with an existing on-premises system, you can install Data Migration Assistant (DMA). This tool analyzes your databases on SQL Server and reports any issues that could block migration to a managed instance.

## Business benefits

SQL Database managed instance provides all the management and security benefits available when using Single Database and Elastic Pool.

Managed instance has near 100% compatibility with SQL Server Enterprise Edition, running on-premises.

The SQL Database managed instance deployment option supports traditional SQL Server Database engine logins and logins integrated with Azure Active Directory (AD).

SQL Database managed instance supports linked servers, although some of the other advanced features required by the database might not be available.

# Instancia administrada de Azure SQL DB

**Desafío para el cliente**
Quiero migrar a la nube, eliminar la sobrecarga de administración, pero necesito funciones de ámbito de instancia (Service Broker, Agente SQL Server, CLR...)

**Solución**
La instancia administrada combina las principales características de seguridad con la compatibilidad de SQL Server y el modelo comercial diseñado para clientes locales

### Características clave

- Instancia única o grupo de instancias
- Área de superficie de SQL Server (gran mayoría)
- Soporte de red virtual nativa
- Servicio totalmente administrado
- Identidades locales habilitadas con Azure AD y AD Connect

### Diferenciadores de Azure

- Migración de tiempo de inactividad casi nulo mediante el trasvase de registros
- Continuidad empresarial totalmente administrada con grupos de conmutación por error
- Retorno de la inversión proyectado del 212 % durante tres años[1]
- Lo mejor de SQL Server con los beneficios de un servicio administrado

# Base de datos o instancia administrada de Azure SQL

## Instancia administrada de Azure SQL

### Instancia única

- Área de superficie de SQL Server (gran mayoría)
- Soporte de red virtual nativa
- Servicio totalmente administrado

### Grupo de instancias

- Aprovisionar previamente los recursos informáticos para la migración
- Permite una migración rentable.
- Capacidad para hospedar instancias más pequeñas (2 núcleos virtuales)
- Actualmente en versión preliminar pública

## Azure SQL Database

### Base de datos única

- Almacenamiento a hiperescala (hasta 100 TB)
- Informática sin servidor
- Servicio totalmente administrado

### Grupo elástico

- Compartir recursos entre múltiples bases de datos para optimizar el precio
- Gestión de rendimiento simplificada para múltiples bases de datos
- Servicio totalmente administrado

# PostgreSQL, MariaDB, and MySQL

## What are MySQL, MariaDB, and PostgreSQL

PostgreSQL, MariaDB, and MySQL are relational database management systems that are tailored for different specializations.

MySQL started life as a simple-to-use open-source database management system. It is the leading open-source relational database for Linux, Apache, MySQL, and PHP (LAMP) stack apps. The Community edition is available free-of-charge and has historically been popular as a database management system for web applications, running under Linux. Enterprise edition provides a comprehensive set of tools and features, including enhanced security, availability, and scalability.


MySQL es una base de datos relacional de código abierto líder para aplicaciones de pila LAMP

MariaDB is a database engine has since been rewritten and optimized to improve performance. MariaDB offers compatibility with Oracle Database. One notable feature of MariaDB is its built-in support for temporal data. A table can hold several versions of data, enabling an application to query the data as it appeared at some point in the past.

MariaDB es una bifurcación de MySQL desarrollada por la comunidad con un fuerte enfoque en la comunidad de usuarios

PostgreSQL is a hybrid relational-object database. You can store data in relational tables, but a PostgreSQL database also enables you to store custom data types, with their own non-relational properties. The database management system is extensible; you can add code modules to the database, which can be run by queries. PostgreSQL has its own query language called pgsql.

PostgreSQL es la base de datos más popular y demandada para aplicaciones modernas

| MySQL | MariaDB | PostgreSQL |
|---|---|---|
| • Very popular<br>• Available as free Community edition or paid-for, and more functional, Standard and Enterprise editions<br>• Azure Database for MySQL is based on the free Community edition, but adds high availability and scalability | • Compatible with Oracle Database<br>• Built-in support for temporal data | • Can store both relational and non-relational data<br>• Can store geometric data<br>• Extensible |

## What is Azure Database for MySQL?

Azure Database for MySQL is a PaaS implementation of MySQL in the Azure cloud, based on the MySQL Community Edition.

The Azure Database for MySQL service includes high availability at no additional cost and scalability as required. You only pay for what you use. Automatic backups are provided, with point-in-time restore.

The server provides connection security to enforce firewall rules and, optionally, require SSL connections.

**Benefits of Azure Database for MySQL**

You get the following features with Azure Database for MySQL:

- High availability features built in.
- Predictable performance.
- Easy scaling that responds quickly to demand.
- Secure data, both at rest and in motion.
- Automatic backups and point-in-time restore for the last 35 days.
- Enterprise-level security and compliance with legislation.

Azure Database for MySQL servers provides monitoring functionality to add alerts, and to view metrics and logs.

What is Azure Database for MariaDB?

Azure Database for MariaDB is an implementation of the MariaDB database management system adapted to run in Azure. It's based on the MariaDB Community Edition.

The database is fully managed and controlled by Azure. Once you've provisioned the service and transferred your data, the system requires almost no additional administration.

**Benefits of Azure Database for MariaDB**

Azure Database for MariaDB delivers:

- Built-in high availability with no additional cost.
- Predictable performance, using inclusive pay-as-you-go pricing.
- Scaling as needed within seconds.
- Secured protection of sensitive data at rest and in motion.
- Automatic backups and point-in-time-restore for up to 35 days.
- Enterprise-grade security and compliance.

What is Azure Database for PostgreSQL?

Azure Database for PostgreSQL to run a PaaS implementation of PostgreSQL in the Azure Cloud. This service provides the same availability, performance, scaling, security, and administrative benefits as the MySQL service.

Some features of on-premises PostgreSQL databases are not available in Azure Database for PostgreSQL. These features are mainly concerned with the extensions that users can add to a database to perform specialized tasks.

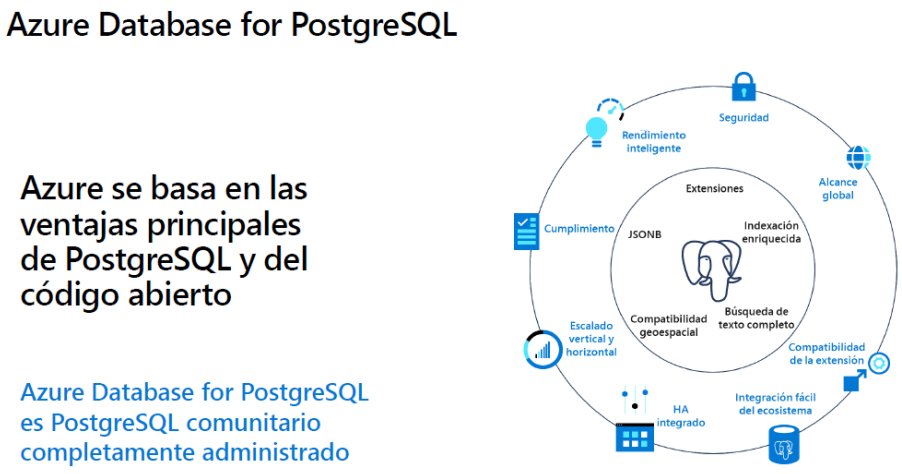
Azure Database for PostgreSQL has two deployment options: Single-server and Hyperscale.

**Azure Database for PostgreSQL single-server**

You choose from three pricing tiers: Basic, General Purpose, and Memory Optimized. Each tier supports different numbers of CPUs, memory, and storage sizes—you select one based on the load you expect to support.

**Azure Database for PostgreSQL Hyperscale (Citus)**

Hyperscale (Citus) is a deployment option that scales queries across multiple server nodes to support large database loads. Your database is split across nodes. Consider using this deployment option for the largest database PostgreSQL deployments in the Azure Cloud.

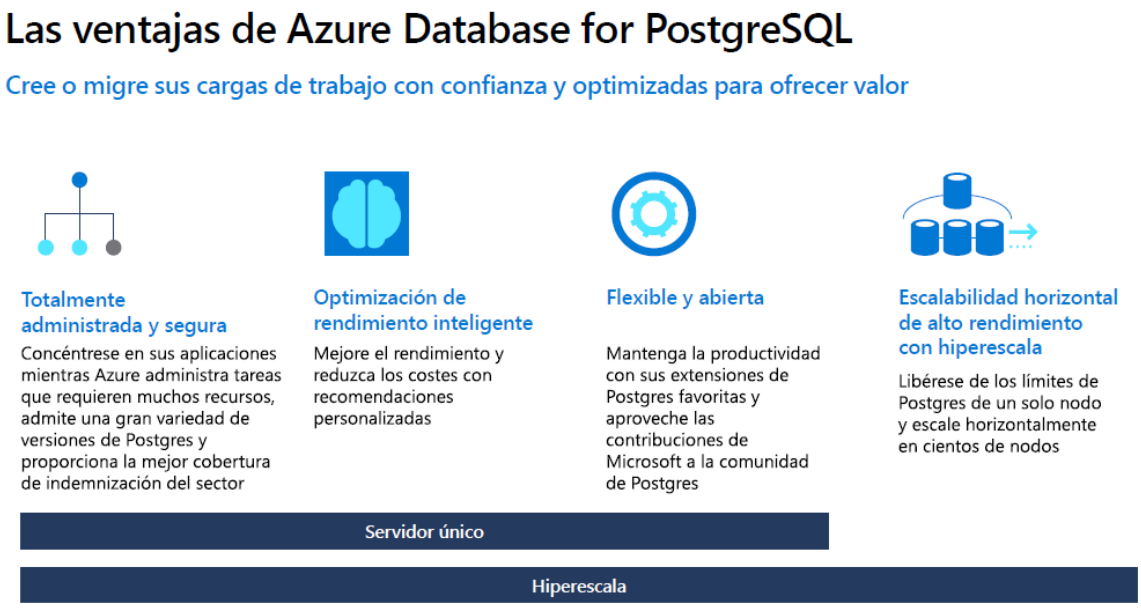


Benefits of Azure Database for PostgreSQL

Azure Database for PostgreSQL is a highly available service. It contains built-in failure detection and failover mechanisms.

Users of PostgreSQL will be familiar with the pgAdmin tool, which you can use to manage and monitor a PostgreSQL database. You can continue to use this tool to connect to Azure Database for PostgreSQL.

Azure Database for PostgreSQL servers' records information about the queries run against databases on the server and saves them in a database named azure_sys. You query the query_store.qs_view view to see this information, and use it to monitor the queries that users are running.



# Migrate data to Azure

The Database Migration Service enables you to restore a backup of your on-premises databases directly to databases running in Azure Data Services.

You can also configure replication from an on-premises database, so that any changes made to data in that database are copied to the database running in Azure Data Services.

# Explore provisioning and deploying relational database services in Azure

## Describe provisioning relational data services

### What is provisioning?

Provisioning is the act of running series of tasks that a service provider, such as Azure SQL Database, performs to create and configure a service. You'll be assigned these resources, and they remain allocated to you (and charged to you), until you delete the service.

All you do is specify parameters that determine the size of the resources required (how much disk space, memory, computing power, and network bandwidth). These parameters are determined by estimating the size of the workload that you intend to run using the service. The act of increasing (or decreasing) the resources used by a service is called scaling.

Azure provides several tools you can use to provision services:

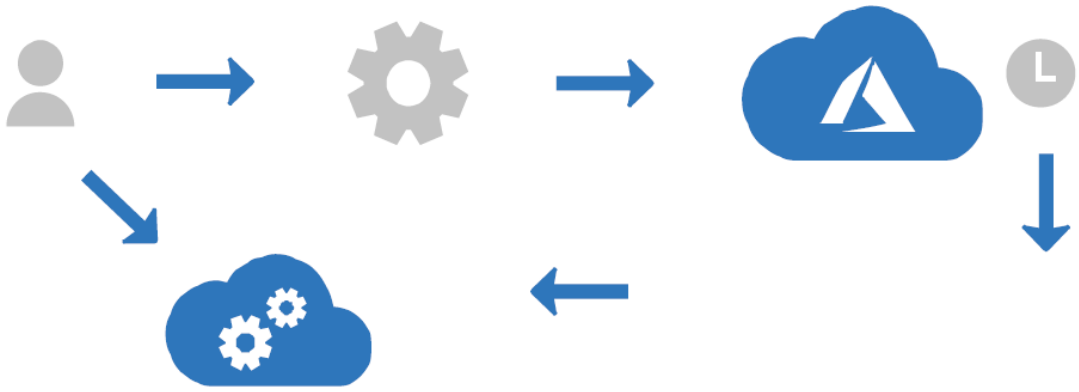
**The Azure portal:** The Azure portal displays a series of service-specific pages that prompt you for the settings required, and validates these settings, before actually provisioning the service.

**The Azure command-line interface (CLI):** The CLI provides a set of commands that you can run from the operating system command prompt or the Cloud Shell in the Azure portal. You can use these commands to create and manage Azure resources.

**Azure PowerShell:** Many administrators are familiar with using PowerShell commands to script and automate administrative tasks. Azure provides a series of commandlets (Azure-specific commands) that you can use in PowerShell to create and manage Azure resources.

**Azure Resource Manager templates:** An Azure Resource Manager template describes the service (or services) that you want to deploy in a text file, in a format known as JSON (JavaScript Object Notation).

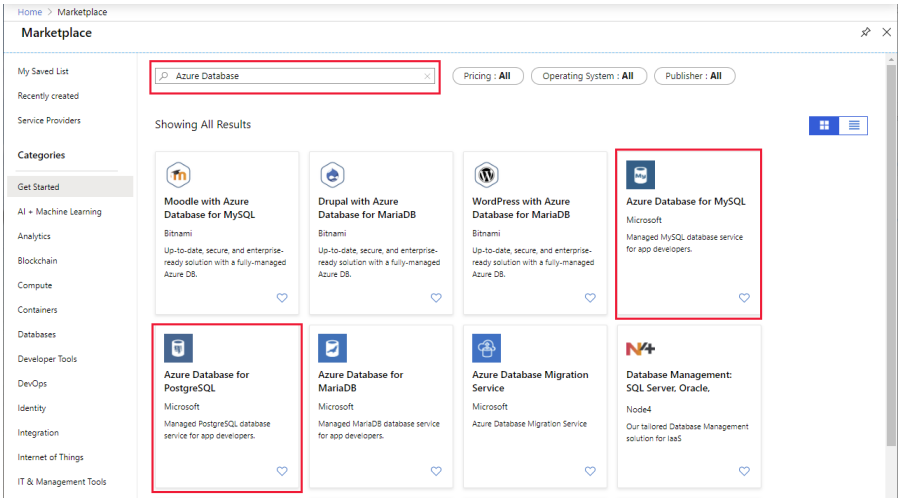
¿Qué es el aprovisionamiento?



# Describe provisioning PostgreSQL and MySQL

## How to provision Azure Database for PostgreSQL and Azure Database for MySQL

As with Azure SQL Database, you can provision a PostgreSQL or MySQL database interactively using the Azure portal. You can find both of these services in the Azure Marketplace.




The hyperscale deployment option supports:

- Horizontal scaling across multiple machines. This option enables the service to add and remove computers as workloads increase and diminish.
- Query parallelization across these servers. The service can split resource intensive queries into pieces which can be run in parallel on the different servers.
- Excellent support for multi-tenant applications, real time operational analytics, and high throughput transactional workloads.

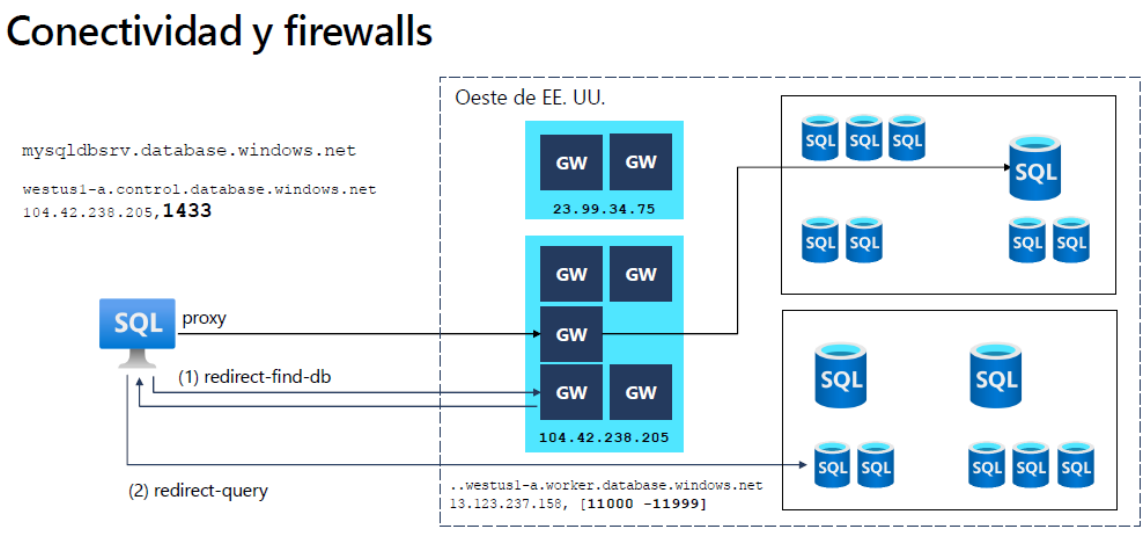
# Describe configuring relational data services

After you've provisioned a resource, you'll often need to configure it to meet the needs of your applications and environment.

## Configure connectivity and firewalls

The default connectivity for Azure relational data services is to disable access to the world.

In the Virtual networks section, you can specify which virtual networks are allowed to route traffic to the service. When you create items such as web applications and virtual machines, you can add them to a virtual network.

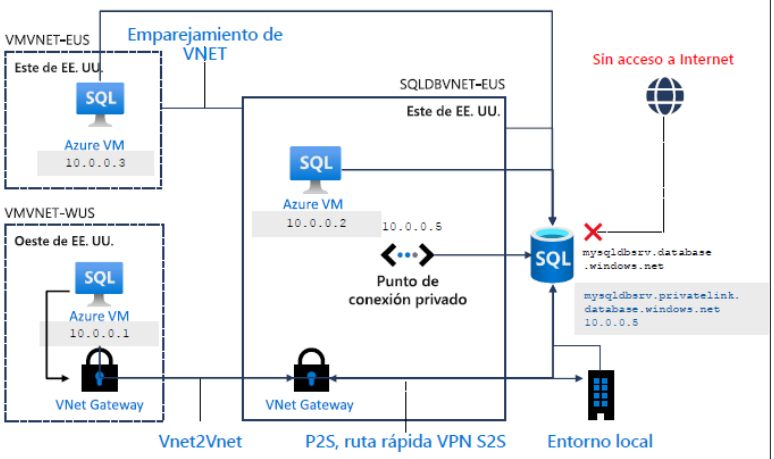


Configure connectivity from private endpoints.

Azure Private Endpoint is a network interface that connects you privately and securely to a service powered by Azure Private Link. Private Endpoint uses a private IP address from your virtual network, effectively bringing the service into your virtual network.

## Seguridad de la red: base de datos SQL

- Permitir el acceso a los servicios de Azure
- Reglas de firewall
- Reglas de red virtual
- Private Link



## Configure authentication

With Azure Active Directory (AD) authentication, you can centrally manage the identities of database users and other Microsoft services in one central location.

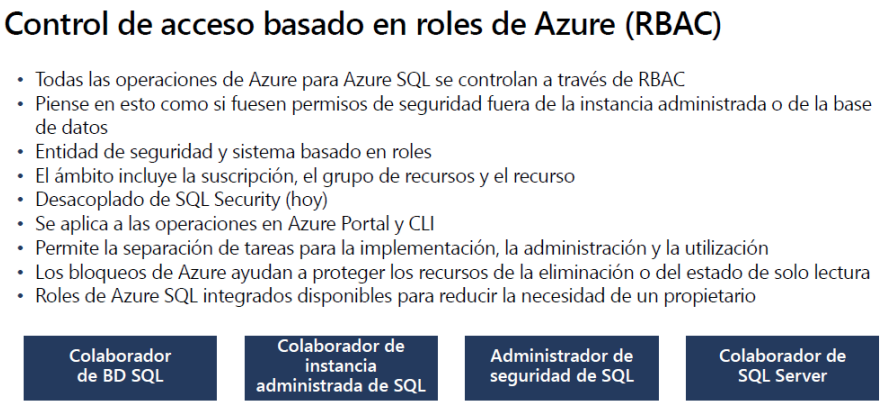
You can use these identities and configure access to your relational data services.

## Configure access control

Azure AD enables you to specify who, or what, can access your resources. Access control defines what a user or application can do with your resources once they've been authenticated.

Azure role-based access control (Azure RBAC) helps you manage who has access to Azure resources, and what they can do with those resources.

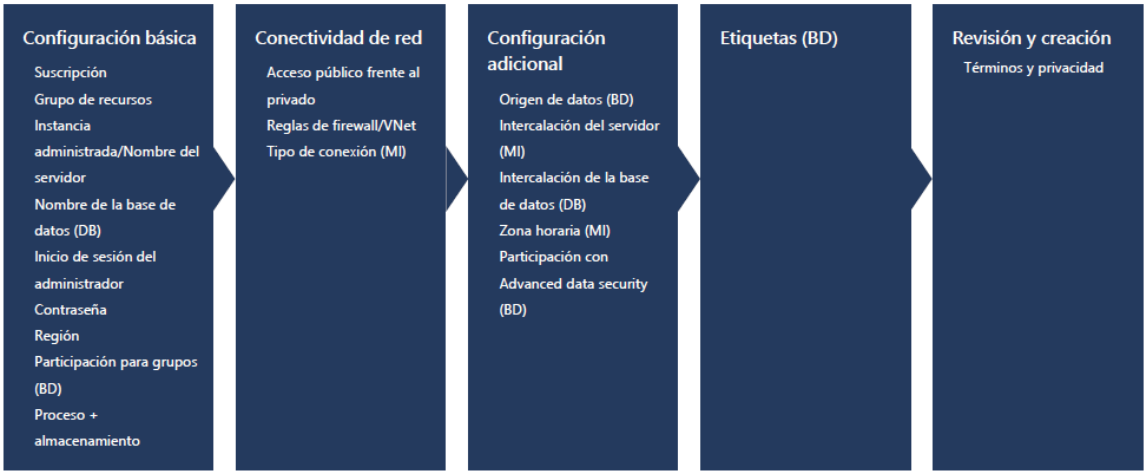
- Allow one user to manage virtual machines in a subscription and another user to manage virtual networks.
- Allow a database administrator group to manage SQL databases in a subscription.
- Allow a user to manage all resources in a resource group, such as virtual machines, websites, and subnets.
- Allow an application to access all resources in a resource group.



Configure advanced data security

Advanced data security implements threat protection and assessment. Threat protection adds security intelligence to your service. This intelligence monitors the service and detects unusual patterns of activity that could be harmful, or compromise the data managed by the service.

## Configurar los servicios de datos relacionales



# Query relational data in Azure

## Introduction to SQL

SQL stands for Structured Query Language. SQL is used to communicate with a relational database. It's the standard language for relational database management systems. SQL statements are used to perform tasks such as update data in a database or retrieve data from a database.

## Understand SQL dialects

You can use SQL statements such as SELECT, INSERT, UPDATE, DELETE, CREATE, and DROP to accomplish almost everything that one needs to do with a database.

Although these SQL statements are part of the SQL standard, many database management systems also have their own additional proprietary extensions to handle the specifics of that database management system. These extensions provide functionality not covered by the SQL standard and include areas such as security management and programmability.

Some popular dialects of SQL include:

- Transact-SQL (T-SQL). This version of SQL is used by Microsoft SQL Server and Azure SQL Database.
- pgSQL. This is the dialect, with extensions implemented in PostgreSQL.
- PL/SQL. This is the dialect used by Oracle. PL/SQL stands for Procedural Language/SQL.

## Understand SQL statement types

SQL statements are grouped into two main logical groups, and they are:

- Data Manipulation Language (DML)
- Data Definition Language (DDL)

## Use DML statements

You use DML statements to manipulate the rows in a relational table. These statements enable you to retrieve (query) data, insert new rows, or edit existing rows. You can also delete rows if you don't need them anymore.

The four main DML statements are:

| Statement | Description |
|---|---|
| SELECT | Select/Read rows from a table |
| INSERT | Insert new rows into a table |
| UPDATE | Edit/Update existing rows |
| DELETE | Delete existing rows in a table |

If a query returns many rows, they don't necessarily appear in any specific sequence. If you want to sort the data, you can add an ORDER BY clause.

You can also run SELECT statements that retrieve data from multiple tables using a JOIN clause. A join condition defines the way two tables are related in a query by:

- Specifying the column from each table to be used for the join. A typical join condition specifies a foreign key from one table and its associated primary key in the other table.
- Specifying a logical operator (for example, = or <>,) to be used in comparing values from the columns.

## Use DDL statements

You use DDL statements to create, modify, and remove tables and other objects in a database (table, stored procedures, views, and so on).

The most common DDL statements are:

| Statement | Description |
|---|---|
| CREATE | Create a new object in the database, such as a table or a view. |
| ALTER | Modify the structure of an object. For instance, altering a table to add a new column. |
| DROP | Remove an object from the database. |
| RENAME | Rename an existing object. |

The datatypes available for columns in a table will vary between database management systems. However, most database management systems support numeric types such as INT (an integer, or whole number), and string types such as VARCHAR (VARCHAR stands for variable length character data).

## Query relational data in Azure SQL Database

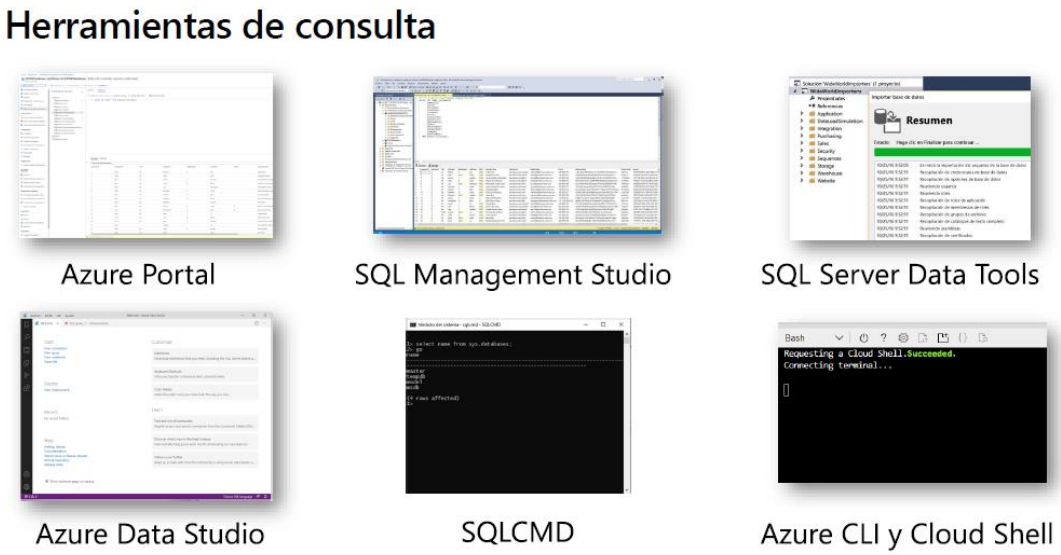
You run SQL commands from tools and utilities that connect to the appropriate database. The tooling available depends on the database management system you're using.

## Retrieve connection information for Azure SQL Database

You can use any of these tools to query data held in Azure SQL Database:

- The query editor in the Azure portal
- The sqlcmd utility from the command line or the Azure Cloud Shell
- SQL Server Management Studio
- Azure Data Studio
- SQL Server Data Tools

To use these tools, you first need to establish a connection to the database. You'll require the details of the server to connect to, an Azure SQL Database account (a username and password) that has access to this server, and the name of the database to use on this server.

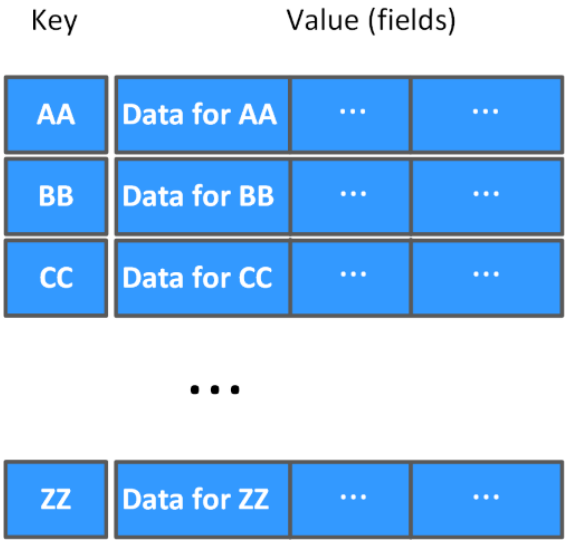


# Describe how to work with non-relational data on Azure

## Explore non-relational data offerings in Azure

### Explore Azure Table storage

Azure Table Storage implements the NoSQL key-value model. In this model, the data for an item is stored as a set of fields, and the item is identified by a unique key.




### What is Azure Table Storage?

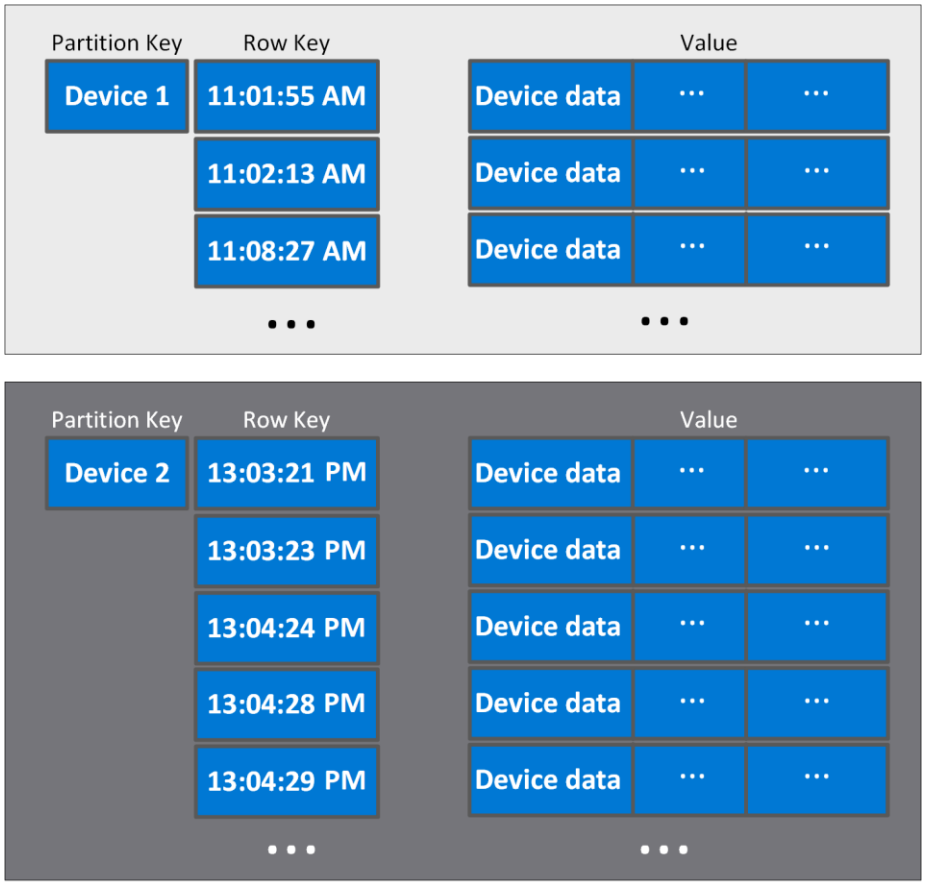
Azure Table Storage is a scalable key-value store held in the cloud. You create a table using an Azure storage account.

In an Azure Table Storage table, items are referred to as rows, and fields are known as columns. An Azure table enables you to store semi-structured data.

All rows in a table must have a key, but apart from that the columns in each row can vary. Unlike traditional relational databases, Azure Table Storage tables have no concept of relationships, stored procedures, secondary indexes, or foreign keys. Data will usually be denormalized.

To help ensure fast access, Azure Table Storage splits a table into partitions. Partitioning is a mechanism for grouping related rows, based on a common property or partition key.

The key in an Azure Table Storage table comprises two elements; the partition key that identifies the partition containing the row (as described above), and a row key that is unique to each row in the same partition.



# Use cases and management benefits of using Azure Table Storage

Azure Table Storage tables are schemeless. It's easy to adapt your data as the needs of your application evolve.

The primary advantages of using Azure Table Storage tables over other ways of storing data include:

- It's simpler to scale.
- A table can hold semi-structured data
- There's no need to map and maintain the complex relationships typically required by a normalized relational database.
- Row insertion is fast
- Data retrieval is fast if you specify the partition and row keys as query criteria.

There are disadvantages to storing data this way though, including:

- Consistency needs to be given consideration as transactional updates across multiple entities aren't guaranteed
- There's no referential integrity; any relationships between rows need to be maintained externally to the table
- It's difficult to filter and sort on non-key data. Queries that search based on non-key fields could result in full table scans

Azure Table Storage is an excellent mechanism for:

- Storing TBs of structured data capable of serving web scale applications.
- Storing datasets that don't require complex joins, foreign keys, or stored procedures, and that can be denormalized for fast access.
- Capturing event logging and performance monitoring data.

Azure Table Storage is intended to support very large volumes of data, up to several hundred TBs in size. As you add rows to a table, Azure Table Storage automatically manages the partitions in a table and allocates storage as necessary.

Azure Table Storage provides high availability guarantees in a single region. Azure Table Storage helps to protect your data. You can configure security and role-based access control to ensure that only the people or applications that need to see your data can actually retrieve it.

## Explore Azure Table Storage

# Explore Azure Blob storage

Many applications need to store large, binary data objects, such as images and video streams. Microsoft Azure virtual machines use blob storage for holding virtual machine disk images. These objects can be several hundreds of GB in size.

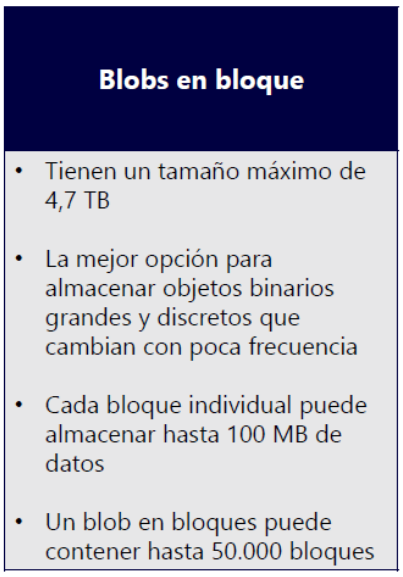
## What is Azure Blob storage?

Azure Blob storage is a service that enables you to store massive amounts of unstructured data, or blobs, in the cloud. Like Azure Table storage, you create blobs using an Azure storage account.

Azure currently supports three different types of blob:

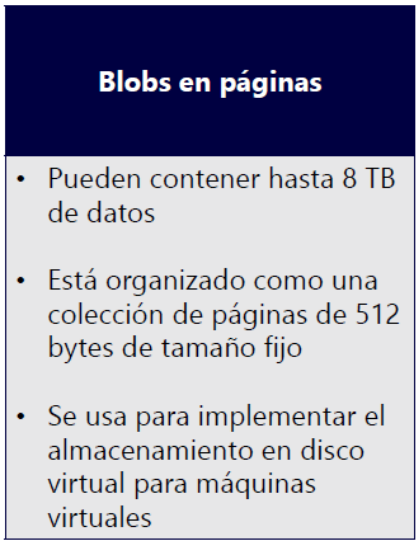
## Block blobs

A block blob is handled as a set of blocks. A block blob can contain up to 50,000 blocks, giving a maximum size of over 4.7 TB. The block is the smallest amount of data that can be read or written as an individual unit. Block blobs are best used to store discrete, large, binary objects that change infrequently.



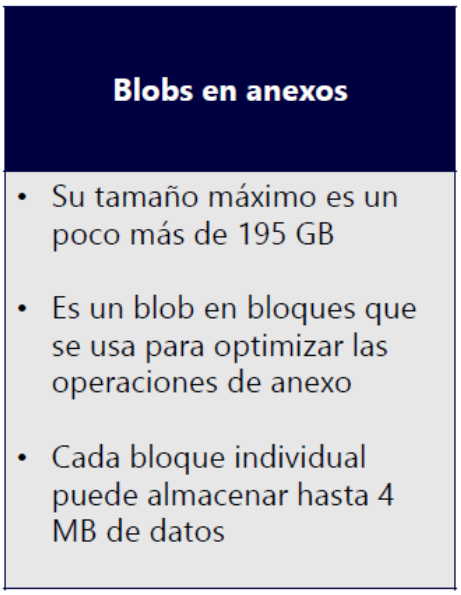
## Page blobs

A page blob is organized as a collection of fixed size 512-byte pages. A page blob is optimized to support random read and write operations; you can fetch and store data for a single page if necessary. A page blob can hold up to 8 TB of data.




## Append blobs

An append blob is a block blob optimized to support append operations. You can only add blocks to the end of an append blob; updating or deleting existing blocks isn't supported. Each block can vary in size, up to 4 MB.



Blob storage provides three access tiers, which help to balance access latency and storage cost:

- **The Hot tier** is the default. You use this tier for blobs that are accessed frequently. The blob data is stored on high-performance media.

- **The Cool tier:** This tier has lower performance and incurs reduced storage charges compared to the Hot tier. Use the Cool tier for data that is accessed infrequently.

- **The Archive tier:** This tier provides the lowest storage cost, but with increased latency. The Archive tier is intended for historical data that mustn't be lost but is required only rarely. Blobs in the Archive tier are effectively stored in an offline state.

Use cases and management benefits of using Azure Blob Storage

Common uses of Azure Blob Storage include:

- Serving images or documents directly to a browser, in the form of a static website.
- Storing files for distributed access
- Streaming video and audio
- Storing data for backup and restore, disaster recovery, and archiving
- Storing data for analysis by an on-premises or Azure-hosted service.

To ensure availability, Azure Blob storage provides redundancy. Blobs are always replicated three times in the region in which you created your account, but you can also select geo-redundancy, which replicates your data in a second region.

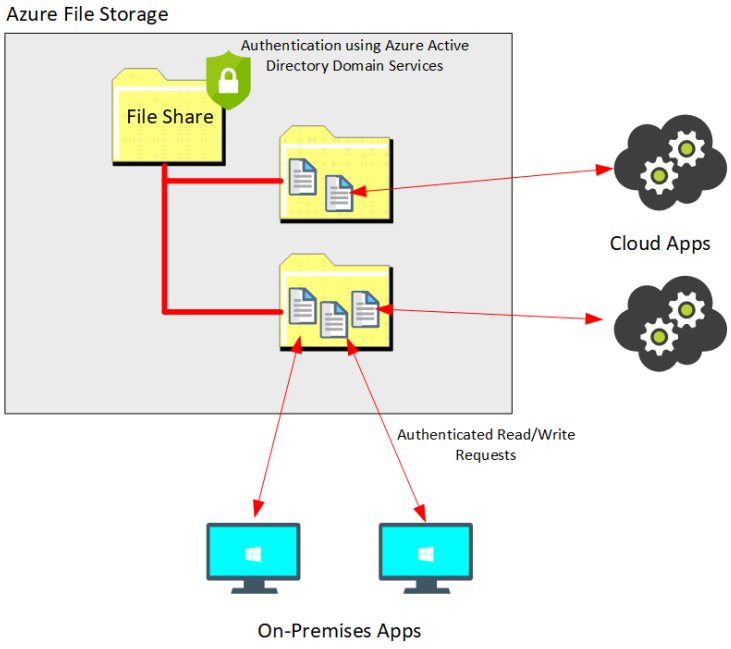
# Explore Azure File storage

A file share enables you to store a file on one computer, and grant access to that file to users and applications running on other computers.

## What is Azure File Storage?

Azure File Storage enables you to create files shares in the cloud and access these file shares from anywhere with an internet connection. Azure File Storage exposes file shares using the Server Message Block 3.0 (SMB) protocol.

You can control access to shares in Azure File Storage using authentication and authorization services available through Azure Active Directory Domain Services.



You create Azure File storage in a storage account. Azure File Storage enables you to share up to 100 TB of data in a single storage account. Azure File Storage supports up to 2000 concurrent connections per shared file.

Once you've created a storage account, you can upload files to Azure File Storage using the Azure portal, or tools such as the AzCopy utility.

Azure File Storage offers two performance tiers. The Standard tier uses hard disk-based hardware in a datacenter, and the Premium tier uses solid-state disks. The Premium tier offers greater throughput but is charged at a higher rate.

## Use cases and management benefits of using Azure File Storage

Azure File Storage is designed to support many scenarios, including the following:

- Migrate existing applications to the cloud: Azure File Storage enables you to migrate your on-premises file or file share-based applications to Azure without having to provision or manage highly available file server virtual machines.

- Share server data across on-premises and cloud: Applications running in the cloud can share data with on-premises applications using the same consistency guarantees implemented by on-premises SMB servers.

- Integrate modern applications with Azure File Storage: By leveraging the modern REST API that Azure File Storage implements in addition to SMB 3.0, you can integrate legacy applications with modern cloud applications, or develop new file or file share-based applications.

- Simplify hosting High Availability (HA) workload data.

# Explore Azure Cosmos DB

Sometimes you require a more generalized solution, that enables you to store and query data more easily, without having to worry about the exact mechanism for performing these operations. This is where a database management system proves useful.

Other models, collectively known as NoSQL databases exist. These models store data in other structures, such as documents, graphs, key-value stores, and column family stores.

## What is Azure Cosmos DB?

Azure Cosmos DB is a multi-model NoSQL database management system. Cosmos DB manages data as a partitioned set of documents. A document is a collection of fields, identified by a key.

The fields in each document can vary, and a field can contain child documents. Many documents' databases use JSON (JavaScript Object Notation) to represent the document structure.

A document can hold up to 2 MB of data, including small binary objects. Cosmos DB provides APIs that enable you to access these documents using a set of well-known interfaces.

The APIs that Cosmos DB currently supports include:

**SQL API:** This interface provides a SQL-like query language over documents, enable to identify, and retrieve documents using SELECT statements.

**Table API:** This interface enables you to use the Azure Table Storage API to store and retrieve documents. The purpose of this interface is to enable you to switch from Table Storage to Cosmos DB without requiring that you modify your existing applications.

**MongoDB API:** You can use the MongoDB API for Cosmos DB to enable a MongoDB application to run unchanged against a Cosmos DB database. You can migrate the data in the MongoDB database to Cosmos DB running in the cloud but continue to run your existing applications to access this data.

**Cassandra API:** Cassandra is a column family database management system. The Cassandra API for Cosmos DB provides a Cassandra-like programmatic interface for Cosmos DB. Cassandra API requests are mapped to Cosmos DB document requests.

**Gremlin API:** The Gremlin API implements a graph database interface to Cosmos DB. A graph is a collection of data objects and directed relationships. Using the Gremlin API, you can walk through the objects and relationships in the graph to discover all manner of complex relationships.

Documents in a Cosmos DB database are organized into containers. The documents in a container are grouped together into partitions. A partition holds a set of documents that share a common partition key. You designate one of the fields in your documents as the partition key.

Use cases and management benefits of using Azure Cosmos DB

Cosmos DB is a highly scalable database management system. Cosmos DB automatically allocates space in a container for your partitions, and each partition can grow up to 10 GB in size.

Indexes are created and maintained automatically. There's virtually no administrative overhead. To ensure availability, all databases are replicated within a single region. This replication is transparent, and failover from a failed replica is automatic. Cosmos DB guarantees 99.99% high availability.

Cosmos DB is certified for a wide array of compliance standards. Additionally, all data in Cosmos DB is encrypted at rest and in motion. Cosmos DB provides row level authorization and adheres to strict security standards.

Cosmos DB is highly suitable for the following scenarios:

- **IoT and telematics.** These systems typically ingest large amounts of data in frequent bursts of activity.
- **Retail and marketing.**
- **Gaming:** A game database needs to be fast and be able to handle massive spikes in request rates during new game launches and feature updates.
- **Web and mobile applications.** Azure Cosmos DB is commonly used within web and mobile applications.

# Casos de uso de Azure Cosmos DB

| | |
|---|---|
| **Web y comercio minorista** | Con el modelo de replicación multimaestro de Azure Cosmos DB y los compromisos de rendimiento de Microsoft, los ingenieros de datos pueden implementar una arquitectura de datos para admitir aplicaciones web y móviles que logren un tiempo de respuesta inferior a 10 ms en cualquier parte del mundo. |
| **Juegos** | El nivel de base de datos es un componente crucial de las aplicaciones de juegos. Los juegos modernos realizan procesamiento gráfico en los clientes de consolas/dispositivos móviles, pero dependen de la nube para ofrecer contenido personalizado, como estadísticas del juego, integración con redes sociales y tablas de clasificación con puntuaciones. |
| **Escenarios de IoT** | Se han diseñado y vendido cientos de miles de dispositivos conocidos como dispositivos de Internet de las cosas (IoT) para generar datos de sensores. Con tecnologías como Azure IoT Hub, los ingenieros de datos pueden diseñar fácilmente una arquitectura de solución de datos que capture datos en tiempo real. Cosmos DB puede aceptar y almacenar esta información muy rápidamente |

# Explore provisioning and deploying non-relational data services in Azure

## What is provisioning?

Provisioning is the act of running a series of tasks that a service provider, such as Azure Cosmos DB, performs to create and configure a service. Behind the scenes, the service provider will set up the various resources (disks, memory, CPUs, networks, and so on) required to run the service.

All you do is specify parameters that determine the size of the resources required (how much disk space, memory, computing power, and network bandwidth). These parameters are determined by estimating the size of the workload that you intend to run using the service.

Azure provides several tools you can use to provision services:

**The Azure portal:** This is the most convenient way to provision a service for most users.

**The Azure command-line interface (CLI):** The CLI provides a set of commands that you can run from the operating system command prompt or the Cloud Shell in the Azure portal.

**Azure PowerShell:** Many administrators are familiar with using PowerShell commands to script and automate administrative tasks.

**Azure Resource Manager templates:** An Azure Resource Manager template describes the service (or services) that you want to deploy in a text file, in a format known as JSON (JavaScript Object Notation).

# Provision Azure Cosmos DB

Azure Cosmos DB is a document database, suitable for a range of applications.

In Cosmos DB, you organize your data as a collection of documents stored in containers. Containers are held in a database. A database runs in the context of a Cosmos DB account. You must create the account before you can set up any databases.

## How to provision a Cosmos DB account

You can provision a Cosmos DB account interactively using the Azure portal, or you can perform this task programmatically through the Azure CLI, Azure PowerShell, or an Azure Resource Manager template.

If you prefer to use the Azure CLI or Azure PowerShell, you can run the following commands to create a Cosmos DB account. The parameters to these commands correspond to many of the options you can select using the Azure portal.

The other deployment option is to use an Azure Resource Manager template. The template for Cosmos DB can be rather lengthy, because of the number of parameters.

## How to create a database and a container

Databases and containers are the primary resource consumers. Resources are allocated in terms of the storage space required to hold your databases and containers, and the processing power required to store and retrieve data.

## Provision other non-relational data services

Besides Cosmos DB, Azure supports other non-relational data services. These services are optimized for more specific cases than a generalized document database store.

Data Lake storage, Blob storage, and File Storage, all require that you first create an Azure storage account.

## How to create a storage account

On the Basics tab, provide for the following details:

- Subscription.
- Resource Group.
- Storage account name.
- Location
- Performance:
    - Standard
    - Premium
- Account kind
    - General-purpose v2.
    - General-purpose v1.
    - BlockBlobStorage.
    - FileStorage
    - BlobStorage
- Replication
    - Locally redundant storage (LRS)
    - Geo-redundant storage (GRS)
    - Read-access geo-redundant storage (RA-GRS)
    - Zone redundant storage (ZRS)
- Access tier.

## Describe configuring non-relational data services

After you've provisioned a resource, you'll often need to configure it to meet the needs of your applications and environment. For example, you might need to set up network access, or open a firewall port to enable your applications to connect to the resource.

## Configure connectivity and firewalls

The default connectivity for Azure Cosmos DB and Azure Storage is to enable access to the world at large. Although this level of access sounds risky, most Azure services mitigates this risk by requiring authentication before granting access.

## Configure connectivity to virtual networks and on-premises computers

In the Virtual networks section, you can specify which virtual networks are allowed to route traffic to the service. If these applications and virtual machines require access to your resource, add the virtual network containing these items to the list of allowed networks.

If you need to connect to the service from an on-premises computer, in the Firewall section, add the IP address of the computer. This setting creates a firewall rule that allows traffic from that address to reach the service.

## Configure connectivity from private endpoints

Azure Private Endpoint is a network interface that connects you privately and securely to a service powered by Azure Private Link. Private Endpoint uses a private IP address from your VNet, effectively bringing the service into your VNet.

The Private endpoint connections page for a service allows you to specify which private endpoints, if any, are permitted access to your service.

## Configure authentication

Many services include an access key that you can specify when you attempt to connect to the service. If you provide an incorrect key, you'll be denied access.

Any user or application that knows the access key for a resource can connect to that resource. However, access keys provide a rather coarse-grained level of authentication.

Azure Active Directory (Azure AD) provides superior security and ease of use over access key authorization.

## Configure access control

Azure AD enables you to specify who, or what, can access your resources. Access control defines what a user or application can do with your resources after they've been authenticated.

You control access to resources using Azure RBAC to create role assignments. A role assignment consists of three elements: a security principal, a role definition, and a scope.

## Configure security

Security implements threat protection and assessment. Threat protection tracks security incidents and alerts across your service.

This intelligence monitors the service and detects unusual patterns of activity that could be harmful, or compromise the data managed by the service.

Recommendations identifies potential security vulnerabilities and recommends actions to mitigate them.

## Configure Cosmos DB

### Configure replication

Azure Cosmos DB enables you to replicate the databases and containers in your account across multiple regions. When you initially provision an account, you can specify that you want to copy data to another region. You don't have control over which region is used as the next nearest region is automatically selected.

The Replicate data globally page enables you to configure replication in more detail. You can replicate to multiple regions, and you select the regions to use. You can also use this page to configure automatic failover to help ensure high availability.

### Configure consistency

Within a single region, Cosmos DB uses a cluster of servers. This approach helps to improve scalability and availability. A copy of all data is held in each server in the cluster.

Eventual consistency provides the lowest latency and least consistency.

Configure Storage accounts

**General configuration**

The Configuration page for a storage account enables you to modify some general settings of the account. You can:

- Enable or disable secure communications with the service.
- Switch the default access tier between Cool and Hot.
- Change the way in which the account is replicated.
- Enable or disable integration with Azure Active Directory Domain Services (Azure AD DS) for requests that access file shares.

Configure encryption

All data held in an Azure Storage account is automatically encrypted. By default, encryption is performed using keys managed and owned by Microsoft. If you prefer, you can provide your own encryption keys.

**Configure shared access signatures**

You can use shared access signatures (SAS) to grant limited rights to resources in an Azure storage account for a specified time period. This feature enables applications to access resources such as blobs and files, without requiring that they're authenticated first.

You can create a token that grants temporary access to the entire service, containers in the service, or individual objects such as blobs and files.

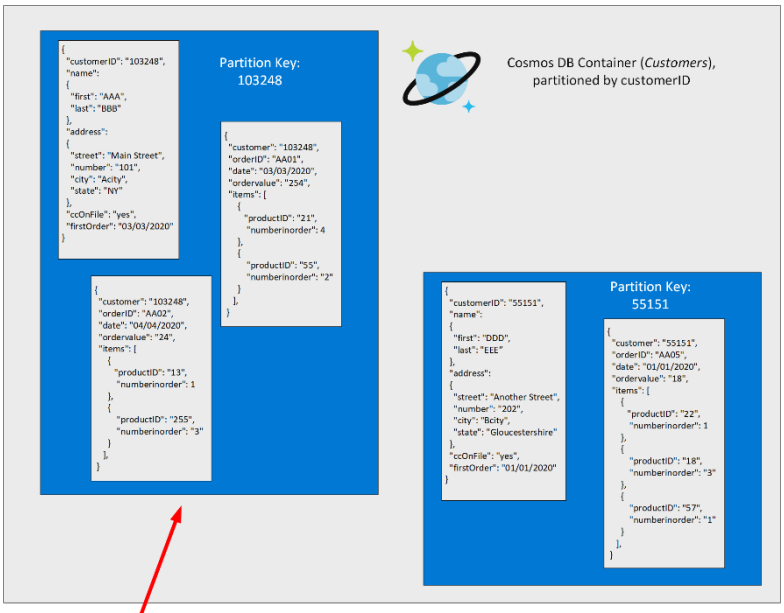
# Manage non-relational data stores in Azure

## Manage Azure Cosmos DB

Azure Cosmos DB is a NoSQL database management system. It's compatible with some existing NoSQL systems, including MongoDB and Cassandra.

Cosmos DB manages data as set of documents. A document is a collection of fields, identified by a key. Cosmos DB uses JSON (JavaScript Object Notation) to represent the document structure.

Documents in a Cosmos DB database are organized into containers. The documents in a container are grouped together into partitions. A partition holds a set of documents that share a common partition key. You designate one of the fields in your documents as the partition key.




```
SELECT *
FROM Customers a
WHERE a.customerID = "103248"
```

## What are Cosmos DB APIs?

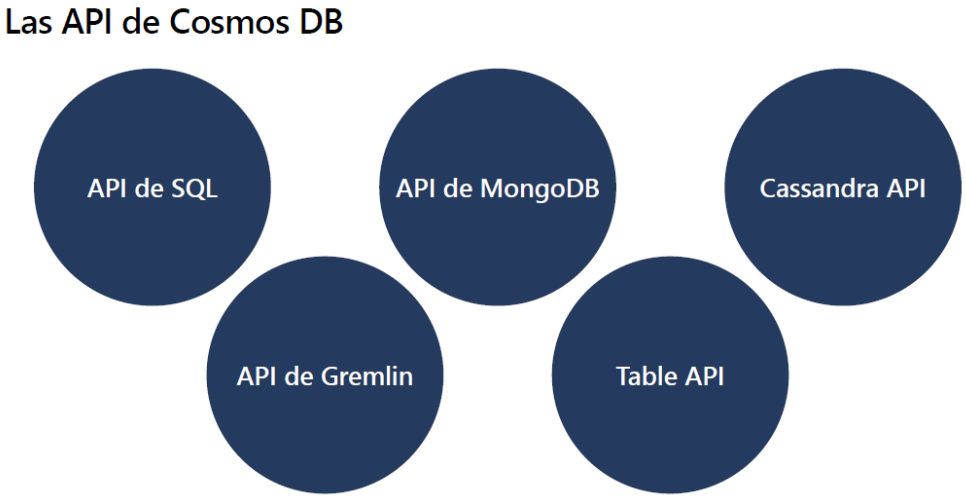
You access the data in a Cosmos DB database through a set of commands and operations, collectively known as an API, or Application Programming Interface.

Cosmos DB provides its own native API, called the SQL API. This API provides a SQL-like query language over documents, that enables you to retrieve documents using SELECT statements.

```sql
SELECT c.address
FROM customers c
WHERE c.customerID = "103248"
```

Cosmos DB also provides other APIs that enable you to access these documents using the command sets of other NoSQL database management systems.

- Table API
- MongoDB API
- Cassandra API
- Gremlin API



Perform data operations in Cosmos DB

- Cosmos DB provides several options for uploading data to a Cosmos DB database and querying that data.
- Use Data Explorer in the Azure portal to run ad-hoc queries.
- Use the Cosmos DB Data Migration tool to perform a bulk-load or transfer of data from another data source.
- Use Azure Data Factory to import data from another source.
- Write a custom application that imports data using the Cosmos DB BulkExecutor library.
- Create your own application that uses the functions available through the Cosmos DB SQL API client library to store data.

Load data using the Cosmos DB Data Migration tool

You can use the Data Migration tool to import data to Azure Cosmos DB from a variety of sources, including:

- JSON files
- MongoDB
- SQL Server
- CSV files
- Azure Table storage
- Amazon DynamoDB
- HBase
- Azure Cosmos containers

**Cargar datos con la herramienta de migración de Cosmos DB**

Puede usar la herramienta de migración de datos para importar datos a Azure Cosmos DB desde diferentes fuentes, que incluyen:

- Archivos JSON
- MongoDB
- SQL Server
- Archivos CSV
- Azure Table Storage
- Amazon DynamoDB
- HBase
- Azure Cosmos Containers

# Query Azure Cosmos DB

Although Azure Cosmos DB is described as a NoSQL database management system, the SQL API enables you to run SQL-like queries against Cosmos DB databases. These queries use a syntax similar to that of SQL, but there are some differences. This is because the data in a Cosmos DB is structured as documents rather than tables.

## Use the SQL API to query documents

The Cosmos DB SQL API supports a dialect of SQL for querying documents using SELECT statements. The SQL API returns results in the form of JSON documents. All queries are executed in the context of a single container.

### Understand a SQL API query

A SQL API SELECT query includes the following clauses:

- SELECT clause.
- FROM clause.
- WHERE clause.
- ORDER BY clause.

In the SQL API, you use JOIN clauses to connect fields in a document with fields in a subdocument that is part of the same document.

```
// Retrieve customers living in California in Name order
SELECT c.Name, c.Address.City
FROM customers c
WHERE c.Address.State = "CA"
ORDER BY c.Name
```

## Understand supported operators

The SQL API includes many common mathematical and string operations, in addition to functions for working with arrays and for checking data types.

| Type | Operator |
| --- | --- |
| Unary | +,-,~, NOT |
| Arithmetic | +,-,*,/,% |
| Bitwise | \|, &, ^, <<, >>, >>> |
| Logical | AND, OR |
| Comparison | =, !=, <, >, <=, >=, <> |
| String (concatenate) | \|\| |
| Ternary (if) | ? |

The SQL API also supports:

- The DISTINCT
- The TOP
- The BETWEEN
- The IS_DEFINED

```
// List all customer cities (remove duplicates) for customers living in states with
SELECT DISTINCT c.Address.City
FROM c
WHERE c.Address.State BETWEEN "AK" AND "MD"

// Find the 3 most common customer names
SELECT TOP 3 *
FROM c
ORDER BY c.Name

// Display the details of every customer for which the data of birth is recorded
SELECT * FROM p
WHERE IS_DEFINED(p.DateOfBirth)
```

**Understand aggregate functions**

You can use aggregate functions to summarize data in SELECT queries; you place aggregate functions in the SELECT clause.

- COUNT(p).
- SUM(p).
- AVG(p).
- MAX(p).
- MIN(p).

```
SELECT AVG(c.age) AS avg,
    MAX(c.age) AS max,
    SUM(c.age) AS sum,
    COUNT(1) AS count
FROM c
```

The SQL API also supports a large number of mathematical, trigonometric, string, array, and spatial functions.

You can use Data Explorer in the Azure portal to create and run queries against a Cosmos DB container.

## Consulte Azure Cosmos DB

**Conceptos básicos de la consulta SELECT**

```
SELECT <select_list>
[FROM <optional_from_specification>]
[WHERE <optional_filter_condition>]
[ORDER BY <optional_sort_specification>]
[JOIN <optional_join_specification>]
```

**Ejemplos**

```
SELECT *
FROM Products p WHERE p.id ="1"

SELECT p.id, p.manufacturer, p.description
FROM Products p WHERE p.id ="1"

SELECT p.price, p.description, p.productId
FROM Products p ORDER BY p.price ASC

SELECT p.productId
FROM Products p JOIN p.shipping
```

# Describe an analytics workload on Azure

## Examine components of a modern data warehouse

By combining all local data with useful external information, it's often possible to gain insights into the data that weren't previously possible. The process of combining all of the local data sources is known as data warehousing. The process of analyzing streaming data and data from the Internet is known as Big Data analytics. Azure Synapse Analytics combines data warehousing with Big Data analytics.

## Describe modern data warehousing

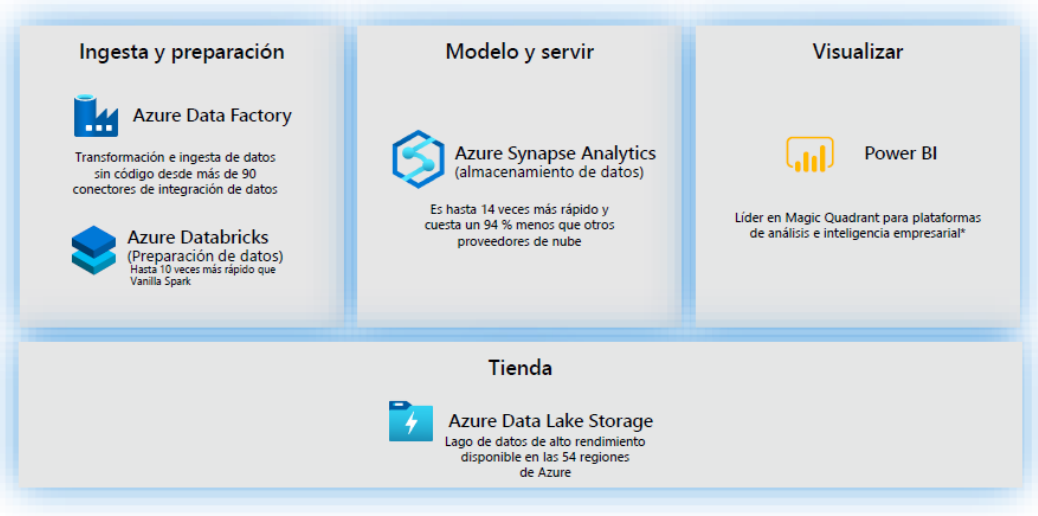
A data warehouse gathers data from many different sources within an organization. This data is then used as the source for analysis, reporting, and online analytical processing (OLAP).

Data warehouses have to handle big data. Big data is the term used for large quantities of data collected in escalating volumes, at higher velocities, and in a greater variety of formats than ever before. Businesses typically depend on their big data to help make critical business decisions.

What is modern data warehousing?

A modern data warehouse might contain a mixture of relational and non-relational data, including files, social media streams, and Internet of Things (IoT) sensor data. You can use tools such as Power BI to analyze and visualize the data, generating reports, charts, and dashboards.

## ¿Qué es el almacenamiento de datos moderno?



Combine batch and stream processing

A typical large-scale business requires a combination of up-to-the-second data, and historical information.

The up-to-the-second data might be used to help monitor real-time, critical manufacturing processes, where an instant decision is required.

Historical data is equally important, to give a business a more stabilized view of trends in performance. A manufacturing organization will require information such as the volumes of sales by products across a month, a quarter, or a year, to determine whether to continue producing various items.

Any modern data warehouse solution must be able to provide access to the streams of raw data, and the cooked business information derived from this data.

# Explore Azure data services for modern data warehousing

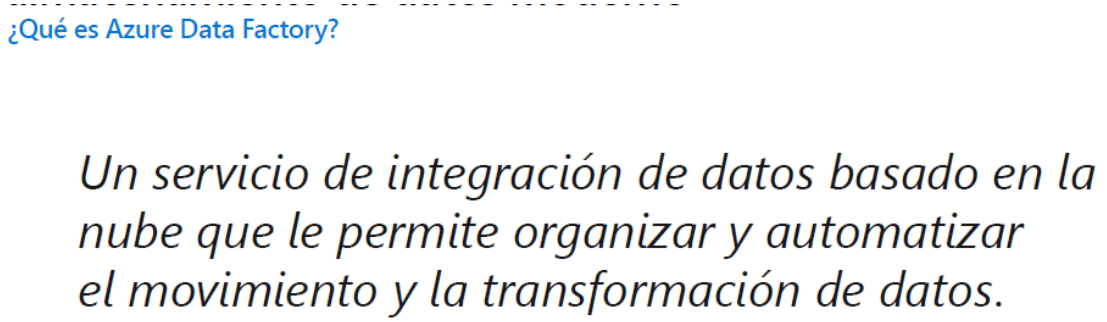
As a data engineer working at an organization with a large manufacturing operation, you want to understand more about the components that form a modern data warehouse. This information will help you determine which elements most closely meet your organization's requirements.

## What is Azure Data Factory?

Azure Data Factory is described as a data integration service. The purpose of Azure Data Factory is to retrieve data from one or more data sources and convert it into a format that you process.

Azure Data Factory enables you to extract the interesting data and discard the rest. The interesting data might not be in a suitable format for processing by the other services in your warehouse solution, so you can transform it.

You define the work performed by Azure Data Factory as a pipeline of operations. A pipeline can run continuously, as data is received from the various data sources.



# What is Azure Data Lake Storage?

A data lake is a repository for large quantities of raw data. Because the data is raw and unprocessed, it's very fast to load and update, but the data hasn't been put into a structure suitable for efficient analysis.

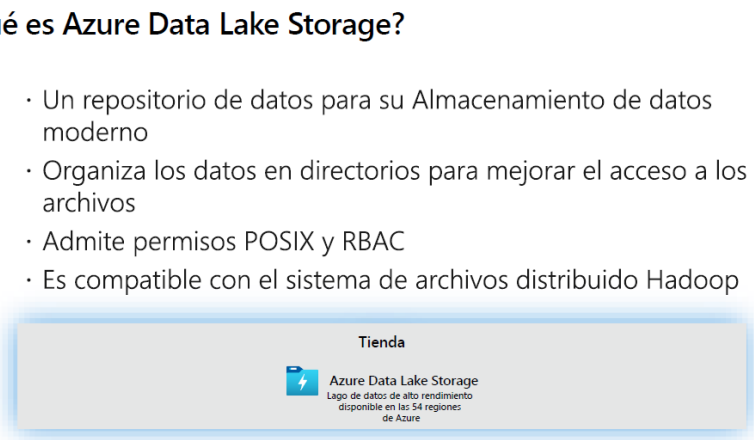
You can think of a data lake as a staging point for your ingested data before it's massaged and converted into a format suitable for performing analytics.

Azure Data Lake Storage combines the hierarchical directory structure and file system semantics of a traditional file system with security and scalability provided by Azure.

Azure Data Lake Storage is essentially an extension of Azure Blob storage, organized as a near-infinite file system.

- Data Lake Storage organizes your files into directories and subdirectories for improved file organization.
- Data Lake Storage supports the Portable Operating System Interface (POSIX) file and directory permissions to enable granular Role-Based Access Control (RBAC) on your data.
- Azure Data Lake Storage is compatible with the Hadoop Distributed File System (HDFS). Hadoop is highly flexible and programmable analysis service.

In an Azure Data Services data warehouse solution, data is typically loaded into Azure Data Lake Storage before being processed into a structure that enables efficient analysis in Azure Synapse Analytics.



What is Azure Databricks?

Azure Databricks is an Apache Spark environment running on Azure to provide big data processing, streaming, and machine learning.

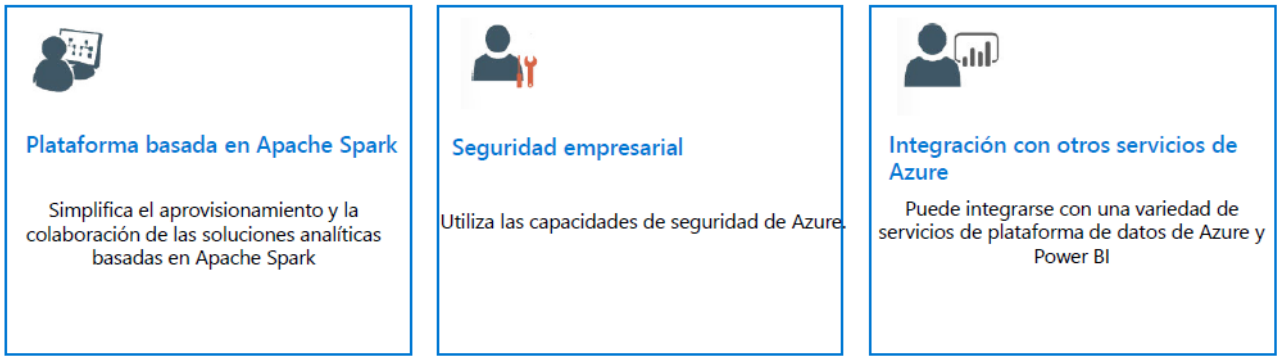
Apache Spark is a highly efficient data processing engine that can consume and process large amounts of data very quickly.

Azure Databricks provides a graphical user interface where you can define and test your processing step by step, before submitting it as a set of batch tasks.

You can create Databricks scripts and query data using languages such as R, Python, and Scala. You write your Spark code using notebooks.

A notebook contains cells, each of which contains a separate block of code. When you run a notebook, the code in each cell is passed to Spark in turn for execution.

## ¿Qué es Azure Databricks?



# What is Azure Synapse Analytics?

Azure Synapse Analytics is an analytics engine. It's designed to process large amounts of data very quickly.

Using Synapse Analytics, you can ingest data from external sources, such as flat files, Azure Data Lake, or other database management systems, and then transform and aggregate this data into a format suitable for analytics processing. You can perform complex queries over this data and generate reports, graphs, and charts.

The Control node is the brain of the architecture. It's the front end that interacts with all applications. When you submit a processing request, the Control node transforms it into smaller requests that run against distinct subsets of the data in parallel.

The Compute nodes provide the computational power. The control node sends the queries to compute nodes, which run the queries over the portion of the data that they each hold. When each node has finished its processing, the results are sent back to the control node where they're combined into an overall result.

Azure Synapse Analytics supports two computational models: SQL pools and Spark pools.

In a SQL pool, each compute node uses an Azure SQL Database and Azure Storage to handle a portion of the data.

In a Spark pool, the nodes are replaced with a Spark cluster. You run Spark jobs comprising code written in Notebooks, in the same way as Azure Databricks. You can write the code for notebook in C#, Python, Scala, or Spark SQL.

# ¿Qué es Azure Synapse Analytics?

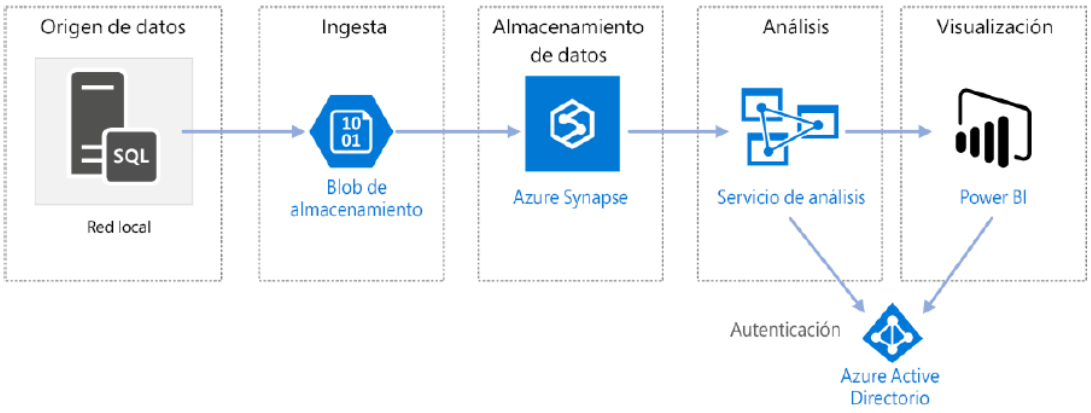

What is Azure Analysis Services?

Azure Analysis Services enables you to build tabular models to support online analytical processing (OLAP) queries. You use these data sources to build models that incorporate your business knowledge. A model is essentially a set of queries and expressions that retrieve data from the various data sources and generate results.

Analysis Services includes a graphical designer to help you connect data sources together and define queries that combine, filter, and aggregate data.

You can explore this data from within Analysis Services, or you can use a tool such as Microsoft Power BI to visualize the data presented by these models.

# ¿Qué es Azure Analysis Services?




Compare Analysis Services with Synapse Analytics

Use Azure Synapse Analytics for:

- Very high volumes of data (multi-terabyte to petabyte sized datasets).
- Very complex queries and aggregations.
- Data mining, and data exploration.
- Complex ETL operations.
- Low to mid concurrency (128 users or fewer).

Use Azure Analysis Services for:

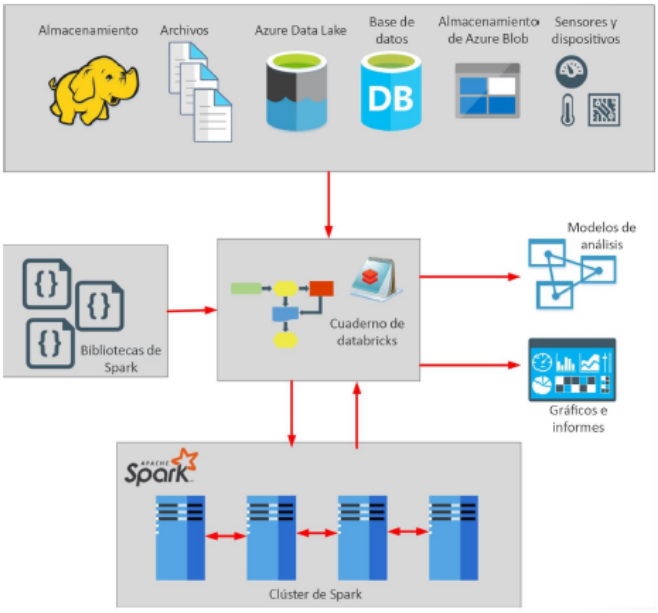
- Smaller volumes of data (a few terabytes).
- Multiple sources that can be correlated.
- High read concurrency (thousands of users).
- Detailed analysis, and drilling into data, using functions in Power BI.
- Rapid dashboard development from tabular data.

What is Azure HDInsight?

Azure HDInsight is a big data processing service, that provides the platform for technologies such as Spark in an Azure environment. HDInsight implements a clustered model that distributes processing across a set of computers.

You can use Azure HDInsight in conjunction with, or instead of, Azure Synapse Analytics. As well as Spark, HDInsight supports streaming technologies such as Apache Kafka, and the Apache Hadoop processing model.

## ¿Qué es Azure HDInsight?



# Explore large-scale data analytics

Large-scale data warehousing and analytics involves two key elements, data ingestion and data processing.

Data ingestion is the process used to load data from various sources into a central data store. Once ingested, the data becomes available for use.

Data processing involves operations on the data to clean, filter, restructure, and prepare the data for analysis.

## Describe common practices for data loading

Data ingestion is the first part of any data warehousing solution. It is arguably the most important part.

If you lose any data at this point, then any resulting information can be inaccurate, failing to represent the facts on which you might base your business decisions.

## Ingest data using Azure Data Factory

Azure Data Factory is a data ingestion and transformation service that allows you to load raw data from many different sources, both on-premises and in the cloud.

Data Factory can clean, transform, and restructure the data, before loading it into a repository such as a data warehouse.

Data Factory contains a series of interconnected systems that provide a complete end-to-end platform for data engineers.

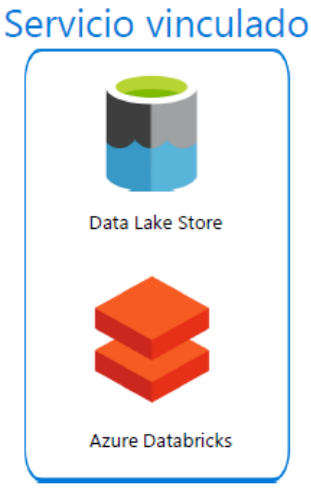
Data Factory provides an orchestration engine. Orchestration is the process of directing and controlling other services, and connecting them together, to allow data to flow between them.

Azure Data Factory uses a number of different resources

**Linked services**

Data Factory moves data from a data source to a destination. A linked service provides the information needed for Data Factory to connect to a source or destination.

The information a linked service contains varies according to the resource.




**Datasets**

A dataset in Azure Data Factory represents the data that you want to ingest (input) or store (output). If your data has a structure, a dataset specifies how the data is structured. Not all datasets are structured.

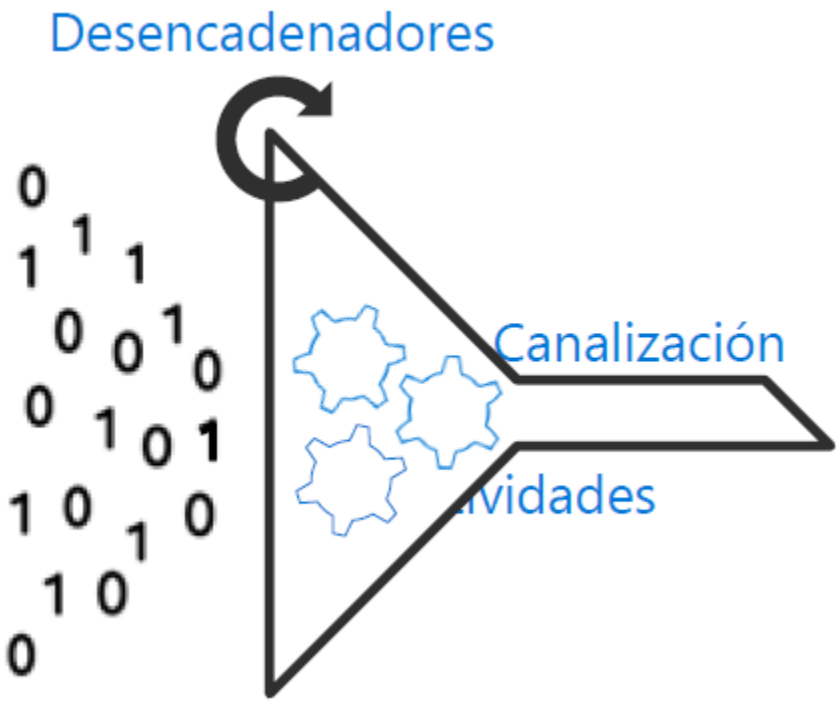
A dataset connects to an input or an output using a linked service.

**Pipelines**

A pipeline is a logical grouping of activities that together perform a task. The activities in a pipeline define actions to perform on your data.

You could include activities that transform the data as it is transferred, or you might combine data from multiple sources together. Other activities enable you to incorporate processing elements from other services.

Pipelines don't have to be linear. You can run a pipeline manually, or you can arrange for it to be run later using a trigger.



## Ingest data using PolyBase

PolyBase is a feature of SQL Server and Azure Synapse Analytics that enables you to run Transact-SQL queries that read data from external data sources.

PolyBase enables you to transfer data from an external data source into a table, as well as copy data from an external data source in Azure Synapse Analytics or SQL Server.

You can also run queries that join tables in a SQL database with external data, enabling you to perform analytics that span multiple data stores.

Azure Data Factory provides PolyBase support for loading data.

## Ingest data using SQL Server Integration Services

SQL Server Integration Services (SSIS) is a platform for building enterprise-level data integration and data transformations solutions.

You can use SSIS to solve complex business problems by copying or downloading files, loading data warehouses, cleaning, and mining data, and managing SQL database objects and data. SSIS is part of Microsoft SQL Server.

SSIS includes a rich set of built-in tasks and transformations, graphical tools for building packages, and the Integration Services Catalog database, where you store, run, and manage packages.

The SSIS Feature Pack for Azure is an extension that provides components that connect to Azure services, transfer data between Azure and on-premises data sources, and process data stored in Azure.

Ingest data using Azure Databricks

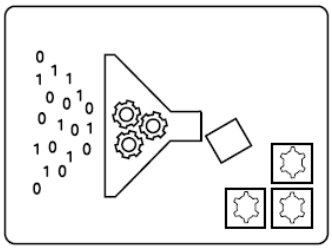
Azure Databricks is an analytics platform optimized for the Microsoft Azure cloud services platform. Databricks is based on Spark and is integrated with Azure to streamline workflows.

You write and run Spark code using notebooks. A notebook is like a program that contains a series of steps (called cells).

Azure Data Factory can incorporate Azure Databricks notebooks into a pipeline. A pipeline can pass parameters to a notebook.

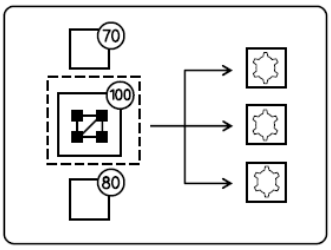
## Describir la ingesta de datos en Azure.

ADF



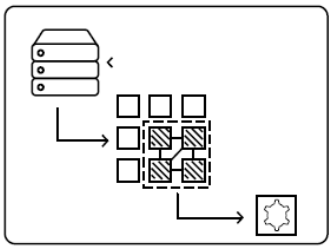
Heterogéneo

PolyBase




Basado en archivos

SSIS



Heterogéneo

# Describe data storage and processing

Organizations generate data throughout their business. For analysis purposes, this data can be left in its raw, ingested format, or the data can be processed and saved to a specially designed data store or data warehouse.

## Process data using Azure Synapse Analytics

Azure Synapse Analytics is a generalized analytics service. You can use it to read data from many sources, process this data, generate various analyses and models, and save the results.

You can select between two technologies to process data:

- Transact-SQL. This is the same dialect of SQL used by Azure SQL Database.
- Spark. This is the same open-source technology used to power Azure Databricks.

Azure Synapse Analytics uses a clustered architecture. Each cluster has a control node that is used as the entry point to the system.

## Process data using Azure Databricks

Databricks is integrated with Azure to provide one-click setup, streamlined workflows, and an interactive workspace that enables collaboration between data scientists, data engineers, and business analysts.

Databricks can also process streaming data. Databricks uses an extensible architecture based on drivers. The processing engine is provided by Apache Spark.

## Process data using Azure HDInsight

HDInsight stores data using Azure Data Lake storage. You can use HDInsight to analyze data using frameworks such as Hadoop Map/Reduce, Apache Spark, Apache Hive, Apache Kafka, Apache Storm, R, and more.

Hadoop Map/Reduce uses a simple framework to split a task over a large dataset into a series of smaller tasks over subsets of the data that can be run in parallel, and the results then combined.

Like Map/Reduce jobs, Spark jobs are parallelized into a series of subtasks tasks that run on the cluster. You can write Spark jobs as part of an application, or you can use interactive notebooks.

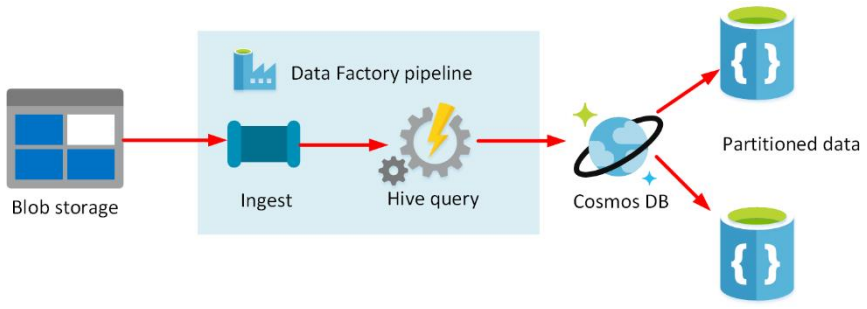
Apache Hive provides interactive SQL-like facilities for querying, aggregating, and summarizing data.

## Process data using Azure Data Factory

Using Azure Data Factory, you can create and schedule data-driven workflows (called pipelines) that can ingest data from the disparate data stores used.

You can build complex ETL processes that transform data visually with data flows or by using compute services such as Azure HDInsight, Azure Databricks, and Azure SQL Database.

You can then publish the transformed data to Azure Synapse Analytics for business intelligence applications to consume.



Process data using Azure Data Lake

**What is Data Lake Store?**

Data Lake Store provides a file system that can store near limitless quantities of data. It uses a hierarchical organization (like the Windows and Linux file systems), but you can hold massive amounts of raw data (blobs) and structured data.

Azure Data Lake Store provides granular security over data, using Access Control Lists.

**What is Data Lake Analytics?**

Azure Data Lake Analytics is an on-demand analytics job service that you can use to process big data. It provides a framework and set of tools that you use to analyze data held in Microsoft Azure Data Lake Store, and other repositories.

You define a job using a language called U-SQL. This is a hybrid language that takes features from both SQL and C# and provides declarative and procedural capabilities that you can use to process data.

Explore Azure Synapse Analytics

Azure Synapse Analytics provides a suite of tools to analyze and process an organization's data. It incorporates SQL technologies, Transact-SQL query capabilities, and open-source Spark tools to enable you to quickly process very large amounts of data.

What are the components of Azure Synapse Analytics?

Azure Synapse Analytics is an integrated analytics service that allows organizations to gain insights quickly from all their data at any hyperscale.

Azure Synapse is composed of the following elements:

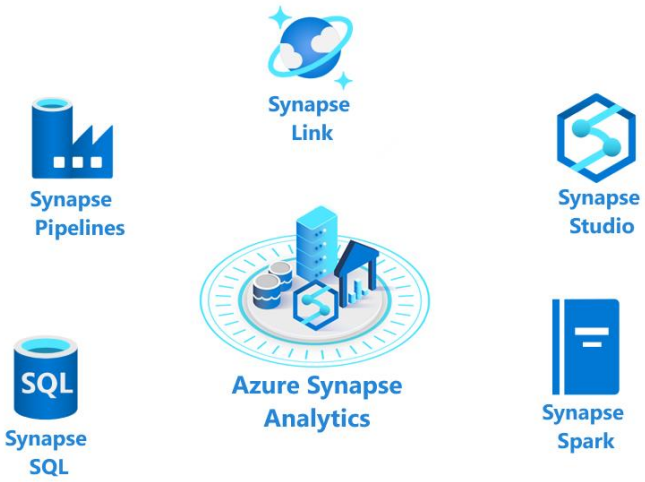
**Synapse SQL pool:** This is a collection of servers running Transact-SQL.

**Synapse Spark pool:** This is a cluster of servers running Apache Spark to process data.

**Synapse Pipelines:** A Synapse pipeline is a logical grouping of activities that together perform a task.

**Synapse Link:** This component allows you to connect to Cosmos DB.

**Synapse Studio:** This is a web user interface that enables data engineers to access all the Synapse Analytics tools.



## What are SQL pools?

In a SQL pool, the Control and Compute nodes in the cluster run a version of Azure SQL Database that supports distributed queries. You define your logic using Transact-SQL statements. You send your Transact-SQL statements to the control node.

The data is split into chunks called distributions. A distribution is the basic unit of storage and processing for parallel queries that run on distributed data.

## What are Spark pools?

You write your analytics jobs as notebooks, using code written in Python, Scala, C#, or Spark SQL (this is a different dialect from Transact-SQL). You can combine code written in multiple languages in the same notebook.

Spark pools enable you to process data held in many formats, such as csv, json, xml, parquet, orc, and avro. Spark can be extended to support many more formats with external data sources.

## What are Synapse pipelines?

A pipeline is a logical grouping of activities that together perform a task. The pipeline allows you to manage the activities as a set instead of each one individually. You deploy and schedule the pipeline instead of the activities independently.

The activities in a pipeline define actions to perform on your data. Synapse pipelines use the same Data Integration engine used by Azure Data Factory.

## What is Synapse Link?

Azure Synapse Link for Azure Cosmos DB is a cloud-native hybrid transactional and analytical processing (HTAP) capability that enables you to run near real-time analytics over operational data stored in Azure Cosmos DB.

Azure Synapse Link enables you to run workloads that retrieve data directly from Cosmos DB and run analytics workloads using Azure Synapse Analytics.

## What is Synapse Studio?

Synapse Studio is a web interface that enables you to create pools and pipelines interactively. With Synapse Studio you can develop, test, and debug Spark notebooks and Transact-SQL jobs.

You can monitor the performance of operations that are currently running, and you can manage the serverless or provisioned resources.

# Get started building with Power BI

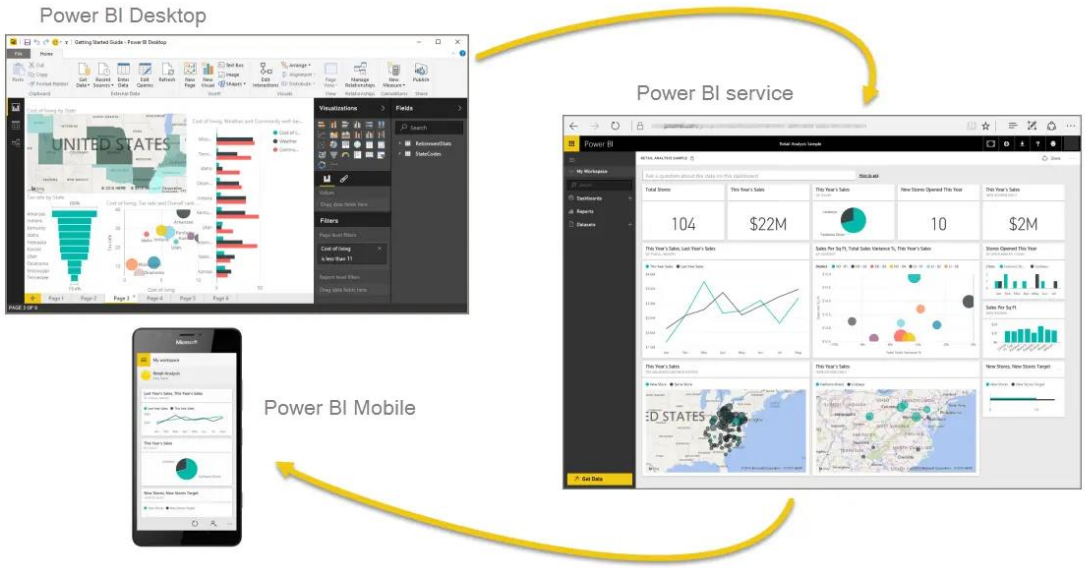
Microsoft Power BI is a collection of software services, apps, and connectors that work together to turn your unrelated sources of data into coherent, visually immersive, and interactive insights. Power BI lets you easily connect to your data sources, visualize (or discover) what's important, and share that with anyone or everyone you want.

Power BI is also robust and enterprise-grade, ready not only for extensive modeling and real-time analytics, but also for custom development.

## The parts of Power BI

Power BI consists of:

- Microsoft Windows desktop application called Power BI Desktop.
- Online SaaS (Software as a Service) service called the Power BI service.
- Mobile Power BI apps that are available on any device, with native mobile BI apps for Windows, iOS, and Android.



## How Power BI matches your role

How you use Power BI might depend on your role on a project or a team. And other people, in other roles, might use Power BI differently, which is just fine.

You also might use each element of Power BI at different times, depending on what you're trying to achieve, or what your role is for a given project or effort.

## The flow of work in Power BI

A common flow of work in Power BI begins in Power BI Desktop, where a report is created. That report is then published to the Power BI service and finally shared, so that users of Power BI Mobile apps can consume the information.

## Use Power BI

The common flow of activity looks like this:

- Bring data into Power BI Desktop and create a report.
- Publish to the Power BI service, where you can create new visualizations or build dashboards.
- Share dashboards with others, especially people who are on the go.
- View and interact with shared dashboards and reports in Power BI Mobile apps.

## Building blocks of Power BI

Everything you do in Microsoft Power BI can be broken down into a few basic building blocks. After you understand these building blocks, you can expand on each of them and begin creating elaborate and complex reports.

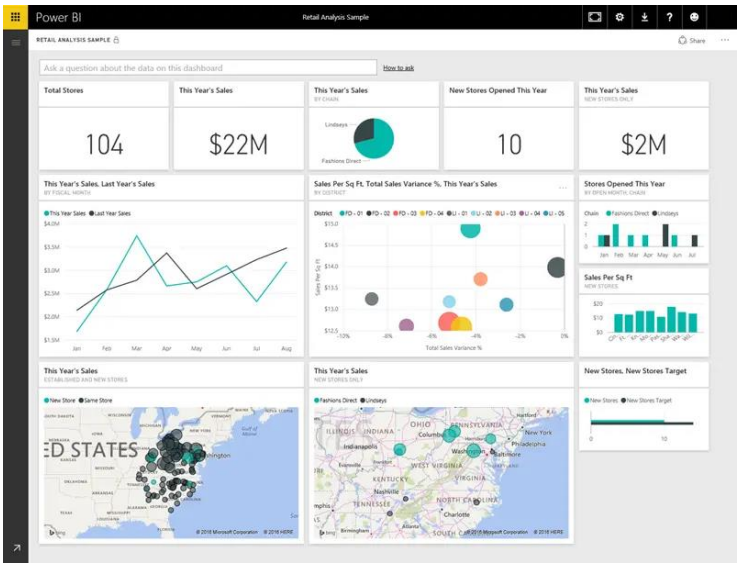
Here are the basic building blocks in Power BI:

- Visualizations
- Datasets
- Reports
- Dashboards
- Tiles

## Visualizations

A visualization (sometimes also referred to as a visual) is a visual representation of data, like a chart, a color-coded map, or other interesting things you can create to represent your data visually.

The goal of a visual is to present data in a way that provides context and insights, both of which would probably be difficult to discern from a raw table of numbers or text.




## Datasets

A dataset is a collection of data that Power BI uses to create its visualizations.

Datasets can also be a combination of many different sources, which you can filter and combine to provide a unique collection of data (a dataset) for use in Power BI. Filtering data before bringing it into Power BI lets you focus on the data that matters to you.

An important and enabling part of Power BI is the multitude of data connectors that are included.

C2132 | 2

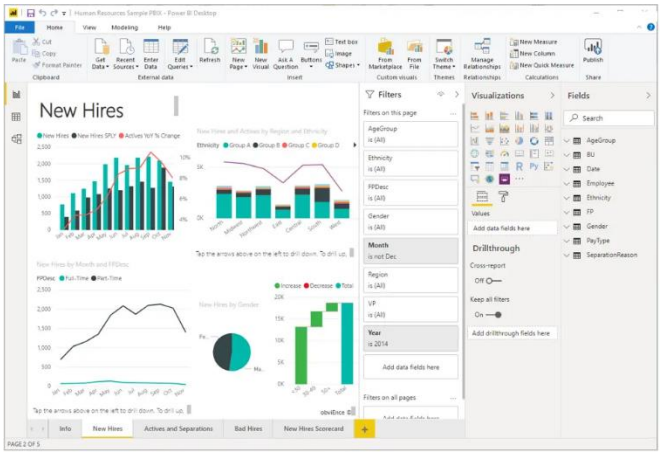
| | B | C | D | E | F | G | H |
|---|---|---|---|---|---|---|---|
| 1 | Year | Month | Month Name | Calendar Month | Births | Births Per Day | Births (Normalized) |
| 2119 | 2004 | 1 | January | 1/1/2004 | 2,937 | 94.7 | 2842 |
| 2120 | 2004 | 2 | February | 2/1/2004 | 2,824 | 97.4 | 2921 |
| 2121 | 2004 | 3 | March | 3/1/2004 | 3,128 | 100.9 | 3027 |
| 2122 | 2004 | 4 | April | 4/1/2004 | 2,896 | 96.5 | 2896 |
| 2123 | 2004 | 5 | May | 5/1/2004 | 3,008 | 97.0 | 2911 |
| 2124 | 2004 | 6 | June | 6/1/2004 | 3,047 | 101.6 | 3047 |
| 2125 | 2004 | 7 | July | 7/1/2004 | 2,981 | 96.2 | 2885 |
| 2126 | 2004 | 8 | August | 8/1/2004 | 3,079 | 99.3 | 2980 |
| 2127 | 2004 | 9 | September | 9/1/2004 | 3,219 | 107.3 | 3219 |
| 2128 | 2004 | 10 | October | 10/1/2004 | 3,547 | 114.4 | 3433 |
| 2129 | 2004 | 11 | November | 11/1/2004 | 3,365 | 112.2 | 3365 |
| 2130 | 2004 | 12 | December | 12/1/2004 | 3,143 | 101.4 | 3042 |
| 2131 | 2005 | 1 | January | 1/1/2005 | 2,921 | 94.2 | 2827 |
| 2132 | 2005 | 2 | February | 2/1/2005 | 2,699 | 96.4 | 2892 |
| 2133 | 2005 | 3 | March | 3/1/2005 | 3,024 | 97.5 | 2926 |
| 2134 | 2005 | 4 | April | 4/1/2005 | 3,037 | 101.2 | 3037 |
| 2135 | 2005 | 5 | May | 5/1/2005 | 3,231 | 104.2 | 3127 |
| 2136 | 2005 | 6 | June | 6/1/2005 | 3,163 | 105.4 | 3163 |
| 2137 | 2005 | 7 | July | 7/1/2005 | 3,119 | 100.6 | 3018 |
| 2138 | 2005 | 8 | August | 8/1/2005 | 3,156 | 101.8 | 3054 |
| 2139 | 2005 | 9 | September | 9/1/2005 | 3,439 | 114.6 | 3439 |

## Reports

In Power BI, a report is a collection of visualizations that appear together on one or more pages.

Just like any other report you might create for a sales presentation or write for a school assignment, a report in Power BI is a collection of items that are related to each other.

Reports let you create many visualizations, on multiple pages if necessary, and let you arrange those visualizations in whatever way best tells your story.




## Dashboards

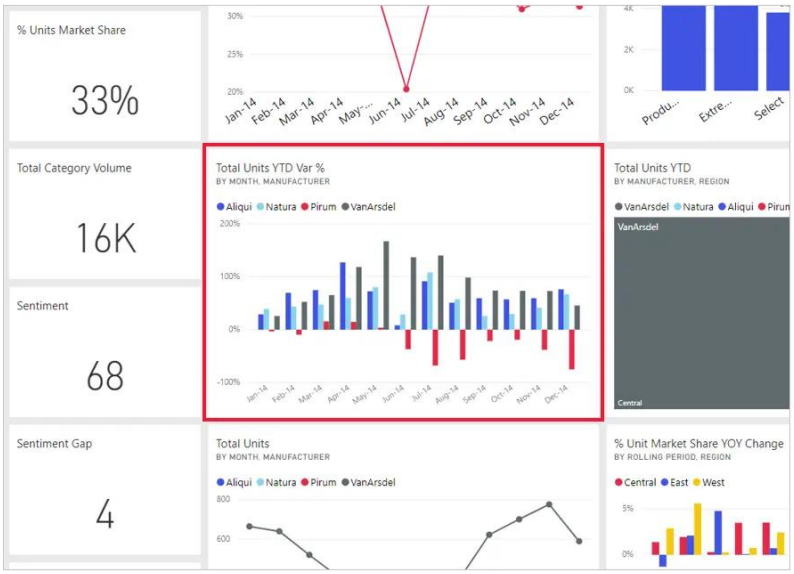
When you're ready to share a report, or a collection of visualizations, you create a dashboard. Power BI dashboard is a collection of visuals from a single page that you can share with others.

A dashboard must fit on a single page, often called a canvas. Think of it like the canvas that an artist or painter uses—a workspace where you create, combine, and rework interesting and compelling visuals.

Tiles

In Power BI, a tile is a single visualization on a dashboard. It's the rectangular box that holds an individual visual.

When you're creating a dashboard in Power BI, you can move or arrange tiles however you want. You can make them bigger, change their height or width, and snuggle them up to other tiles.



# References


Microsoft learn:

Microsoft Certified: Azure Data Fundamentals - Learn | Microsoft Docs

Microsoft Virtual Training Days:

https://mvtd.events.microsoft.com/?azureevent=allazure

Video of freeCodeCamp.org:

https://www.youtube.com/watch?v=P3qmqUZJ7l0&t=3279s

Optional resources:

Post "200 Practice Questions For Azure Data DP-900 Fundamentals Exam":

200 Practice Questions For Azure Data DP-900 Fundamentals Exam | by Bhargav Bachina | Bachina Labs | Medium

Exam tests:

DP-900 Microsoft Exam Info and Free Practice Test | ExamTopics

Microsoft DP-900 Free Practice Exam & Test Training - ITExams.com


# Good luck in your exams!


Carlos Pereira Coto

February 26, 2022
Cartago, Costa Rica