

# Learning Objectives

## Learners will be able to...

- Explain the key tasks performed by a database management system (DBMS)
  - including data definition
  - data updates and retrieval
  - user administration
- Differentiate between different types of databases and their structure :
  - hierarchical, network, relational (SQL)
  - non-relational (NoSQL)
  - object-oriented databases
- Identify various SQL commands
- Use SELECT, FROM, or LIMIT operators to retrieve specific data from a PostgreSQL database

info

## Make Sure You Know

Learners should have a basic understanding of databases and SQL concepts. They should also be familiar with relational databases and their structure, including but not limited to tables, fields and relationships between tables.

# What is a database?

It is highly likely that you have encountered the utilization of tables in real-world situations, such as in the applications Excel or Google Spreadsheets. The employment of tables allows for precise structuring of data, ensuring organization and coherence. Nevertheless, as the volume of data expands, operating and managing tables becomes progressively intricate.

A table with information about films and actors in it may look like this.

index	film_id	title	release_year	rating	length	first_name
1	1	ACADEMY DINOSAUR	2006	PG	86	ROCK
2	1	ACADEMY DINOSAUR	2006	PG	86	MARY
3	1	ACADEMY DINOSAUR	2006	PG	86	JOHNNY
4	1	ACADEMY DINOSAUR	2006	PG	86	PENELOPE
5	1	ACADEMY DINOSAUR	2006	PG	86	SANDRA
6	1	ACADEMY DINOSAUR	2006	PG	86	CHRISTIAN
7	1	ACADEMY DINOSAUR	2006	PG	86	OPRAH
8	1	ACADEMY DINOSAUR	2006	PG	86	WARREN
9	1	ACADEMY DINOSAUR	2006	PG	86	LUCILLE
10	1	ACADEMY DINOSAUR	2006	PG	86	MENA

But working with a large array of data is inconvenient, besides, the data in the table is duplicated.

It is much more convenient to break this data into smaller tables and create relationships between them. This is how databases came about.

info

## Database

A database (DB) is a well-organized collection of electronic information, arranged in a structured and systematic manner.

A database is usually controlled by a database management system (DBMS). Together, the data and the DBMS, along with the applications that are associated with them, are referred to as a database system, often shortened to just database.

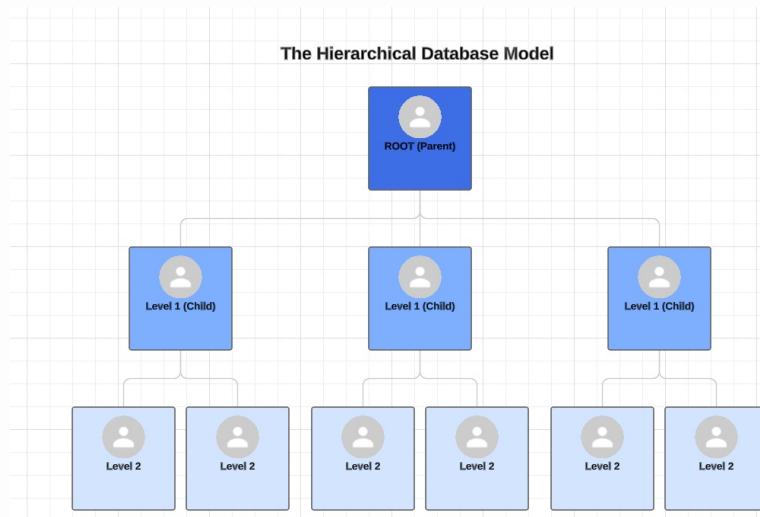
Database Management System	Description
Data Definition	Creation, modification, and removal of definition that defines the organization of data in the database
Data Updation	Insertion, modification, and deletion of the actual data in the database
Data Retrieval	Retrieves the data from the database which can be used by applications for various purposes
User Administration	Registering and monitoring users, maintaining data integrity, enforcing data security, dealing with concurrency control, monitoring performance, and recovering information corrupted by unexpected failure

# Database types

Usually databases are classified depending on how information is structured in them and how to interact with it.

## Hierarchical Data Model

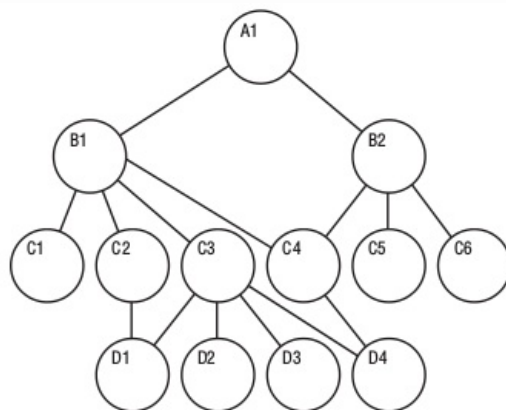
The simplest structure, where records, like branches, move away from the parent. Information is linked by analogy with folders on the desktop. Each entry has a physical relationship with only one previous entry, and a many-to-many relationship is not possible.



HierarchicalDatabase Model

## Network data model

Unlike a hierarchical structure, each entry can have more than one parent. Network databases are not a tree, but a general graph.

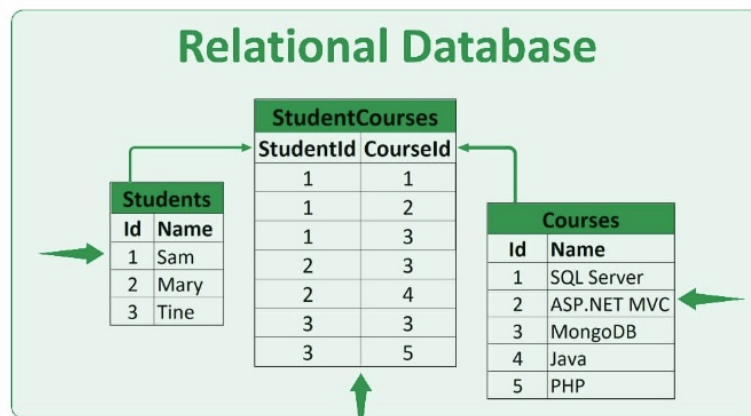


## Network Data Model

[Picture Source](#)

## Relational(SQL) Database

They are also called SQL - as a programming language with which to create, transform and manage data in relational databases. Records and relationships between them are organized using tables. The tables have a foreign key field with links to other tables. Due to the high organization and flexibility of the structure, relational databases are used for many types of data.



Relational Database

[Picture Source](#)

## Non-relational(NoSQL) Database

This group is also called NoSQL, because such databases require other than SQL queries.

- **Key-value databases**

In such databases, data is stored under keys. If you want to get an object, such as an image or text, you need to enter a key. Thus, information about the state of objects, represented by various types of data, is often stored.

- **Graph databases**

Relationships in graph databases are denoted by nodes, edges, and properties. Records in these databases can have any number of properties associated with them.

- **Column databases**

Similar to relational databases, these databases store data in the form of tables. But the structure of the columns is not strictly regulated - they can be combined into column families with a specific format.

## Object-Oriented Databases

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

# Popular DBMS

Let's explore some of the most popular database management systems (DBMS) tailored to address specific needs:

| Database Management Systems | Description |

|—————|—————|

| **PostgreSQL** | An object-relational SQL DBMS that is used for websites, services, and platforms. It offers free access and support for many programming languages, making it one of the most popular databases. |

| **Oracle** | Oracle is an object-relational DBMS created by the market leader of the same name. It is known for its fast installation, configuration, expandability, practicality, and reliability. |

| **MySQL (or MariaDB)** | An open-source relational SQL DBMS that is available for viewing, fixing errors, and creating new versions of programs. MySQL is free, fast, and flexible, suitable for different types of tables. MariaDB is a fork from MySQL and is 100% compatible with prior versions of MySQL. |

| **Microsoft SQL Server** | The optimal SQL DBMS for Windows operating systems, but also compatible with Linux. It easily integrates with other Microsoft products. |

| **MongoDB** | A document-oriented database management system that does not require a description of the table schema. It is considered one of the classic examples of NoSQL systems, using JSON-like documents and a database schema. |

| **Redis** | Redis is an open-source NoSQL resident database management system that works with key-value data structures. It is used for databases, caches, and message brokers, focusing on achieving maximum performance on atomic operations. |

| **Apache Cassandra** | A distributed database management system that belongs to the class of NoSQL systems. It is designed to create highly scalable and reliable storages of huge data arrays represented as a hash. |

info

## Databases

Please note that while SQL and NoSQL databases are different in their data models and storage approaches, they can still support object-oriented programming concepts. However, object-oriented databases are specifically designed to store and retrieve objects directly, offering more comprehensive support for object-oriented programming paradigms.





# SQL Intro



SQL is a structured query language designed to retrieve information from a database.

The syntax of the SQL language is different from the syntax of many other programming languages. An important feature: SQL is a declarative language. With its help, they describe not the algorithm of actions, but the data that they want to receive. How to execute such a command, the DBMS will decide for you.

## What are SQL commands?

SQL commands are keywords or statements used to manipulate data in relational databases. Operators are symbols or words used to perform specific operations, such as selecting data based on specific criteria.

SQL commands can be categorized as follows:

- **Data definition language**

- Data definition language (DDL) refers to SQL commands that design the database structure. Database engineers use DDL to create and modify database objects based on the business requirements. For example, the database engineer uses the CREATE command to create database objects such as tables, views, and indexes.

- **Data query language**

- Data query language (DQL) consists of instructions for retrieving data stored in relational databases. Software applications use the SELECT command to filter and return specific results from a SQL table.

- **Data manipulation language**

- Data manipulation language (DML) statements write new

information or modify existing records in a relational database. For example, an application uses the INSERT command to store a new record in the database.

- **Data control language**

- Database administrators use data control language (DCL) to manage or authorize database access for other users. For example, they can use the GRANT command to permit certain applications to manipulate one or more tables.

- **Transaction control language**

- The relational engine uses transaction control language (TCL) to automatically make database changes. For example, the database uses the ROLLBACK command to undo an erroneous transaction.

info

SQL is the standard used by almost all relational databases. But the implementation features, functionality, and possible data types may differ in different DBMS implementations.

# Simple SELECT

On the left side, you can see a console SQL client that is connected to a pre-created PostgreSQL database.

In it, we can execute queries and see what data is in the database.

To create a query, we need the following operators:

- **SELECT** is an operator that describes what to unload from the database.
- **FROM** is an operator with which they describe exactly where the data should be unloaded from.
- **LIMIT** is an operator that can be used to limit the amount of data returned (this operator is not required, but allows you to return a more readable piece of data).

The SELECT and FROM statements are used in pairs.

Our first request will look like this:

```
SELECT * FROM actor LIMIT 10;
```

After the select statement, we have specified the keyword \*, which allows you to return all the fields in the table.

After the FROM operator, we specified the table from which we want to load data, in our example this is the table actor.

And after the LIMIT statement, we put the number 10, which will limit the number of rows to 10.

And our request ends with the ; operator, which shows the end of the request. If it is not set, the client will wait for the end of the request and will not return any result.

info

## Syntax Conventions

For readability, all SQL commands should be written in uppercase letters. This allows the reader to identify the keywords in the SQL statement and easily determine what the query is executing.

However, the commands are case insensitive.

Try to execute example above. You will get data like this:

```
codio=# SELECT * FROM actor LIMIT 10;
```

actor_id	first_name	last_name	last_update
1	PENELOPE	GUINNESS	2006-02-15 04:34:33
2	NICK	WAHLBERG	2006-02-15 04:34:33
3	ED	CHASE	2006-02-15 04:34:33
4	JENNIFER	DAVIS	2006-02-15 04:34:33
5	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33
6	BETTE	NICHOLSON	2006-02-15 04:34:33
7	GRACE	MOSTEL	2006-02-15 04:34:33
8	MATTHEW	JOHANSSON	2006-02-15 04:34:33
9	JOE	SWANK	2006-02-15 04:34:33
10	CHRISTIAN	GABLE	2006-02-15 04:34:33

(10 rows)

We wrote “like” because relational databases do not guarantee order without the use of additional operators that are used for sorting.

You could change table to country and city for example and see the data in different tables.

# Simple SELECT from Python

On the previous page, we used the console client to the database, which comes with the DBMS we use.

Contemporary programming languages often offer dedicated client libraries that enable seamless integration with your preferred database management system (DBMS). Furthermore, you may find multiple client options available, allowing you to select the one that best aligns with your requirements and preferences.

We will use PyGreSQL to interact with the database.

To do this, simply import the DB wrapper class and create an instance of it, passing the necessary connection parameters, like this:

```
from pg import DB

db = DB(dbname='codio', port=5432, user='codio')
```

You can omit one or even all parameters if you want to use their default values. PostgreSQL will use the name of the current operating system user as the login and the database name, and will try to connect to the local host on port 5432 if nothing else is specified.

The db object has all methods of the lower-level Connection class plus some more convenience methods provided by the DB wrapper.

You can now execute database queries using the `DB.query()` method and print a result:

```
q = db.query("SELECT * FROM actor LIMIT 10")

print(q)
```

You could change table to country and city for example and see the data in different tables.