DEVELHOPE

Introduction to SQL

Teacher: Daniele Montesi

Data Scientist

Index

1. Background on SQL
2. SQL Commands
3. Practical Examples of SQL Queries
4. Best Practices
5. Conclusion

Background on SQL

Why Using SQL?

You cannot effectively replace query language with a GUI-based* application.

1. **Reusability**

2. **Scalability**

3. **Well Established**

*GUI: Graphical User Interface, applications like Google Sheets, PowerBI, etc.

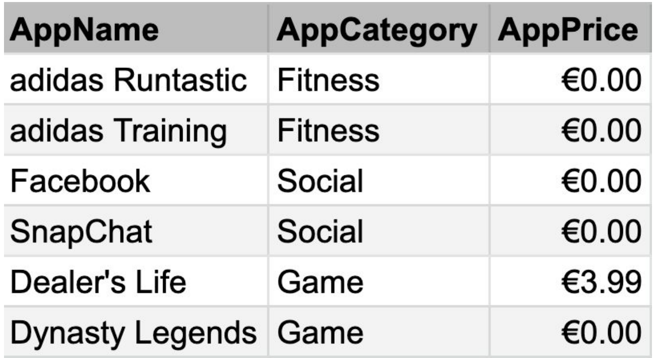
What is SQL?

Definition

SQL (Structured Query Language) is a **declarative query language** used to **perform operations** on **relational databases**

Example queries on a table:

1. **SELECT * FROM Apps**
2. **SELECT COUNT(*) FROM Apps**
3. **SELECT AppName, AppCategory FROM Apps**
4. **SELECT AppName FROM Apps WHERE AppPrice > 0**
5. **SELECT AppName FROM Apps ORDER BY Appname**



AppName	AppCategory	AppPrice
adidas Runtastic	Fitness	€0.00
adidas Training	Fitness	€0.00
Facebook	Social	€0.00
SnapChat	Social	€0.00
Dealer's Life	Game	€3.99
Dynasty Legends	Game	€0.00

Table name: Apps

Database

Definition

A Database is a container storing data, through which a user **can retrieve stored information in an efficient manner**.

People use databases every day, **without realizing it.**

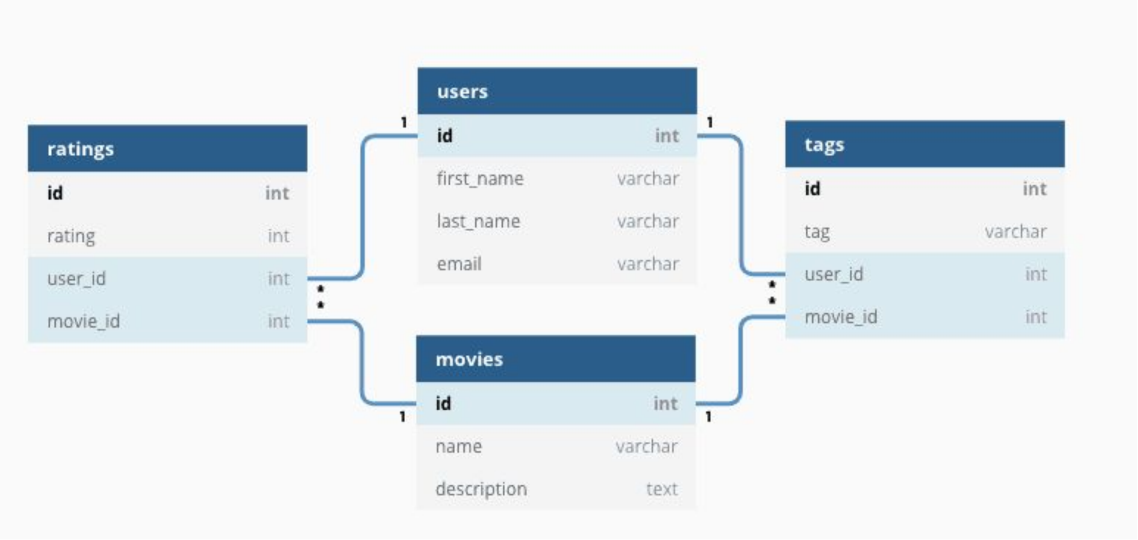
However, databases working with SQL must be **stored in computers** in a way that are easy to be **retrieved, create, updated and deleted.**

Relational Database

Definition

A **Relational Database** is a database organized as a **set of tables** with **columns and rows** with pre-defined **relationships**

Example:



ratings	
id	int
rating	int
user_id	int
movie_id	int

users	
id	int
first_name	varchar
last_name	varchar
email	varchar

movies	
id	int
name	varchar
description	text

tags	
id	int
tag	varchar
user_id	int
movie_id	int

Declarative Query Language

Definition

Language used to communicate with databases to specify **what** to obtain without taking care of the task **efficiency**

In **Declarative languages**, the result is specified by the user using predefined syntaxis defined by the language itself
Example of declarative languages:

1. SQL
2. Regex
3. VHDL

In **Imperative languages**, the result is specified by the user in **multiple steps** and at an **higher level of abstraction**
Example of imperative languages:

1. PHP
2. C
3. Java

SQL Commands

Operations in SQL

There are 5 types of Operations that can be done in SQL:

1. **DDL: Data Definition Language**

2. **DQL: Data Query Language**

3. **DML: Data Manipulation Language**

4. **DCL: Data Control Language**

5. **TCL: Transaction Control Language**

DDL: Data Definition Language

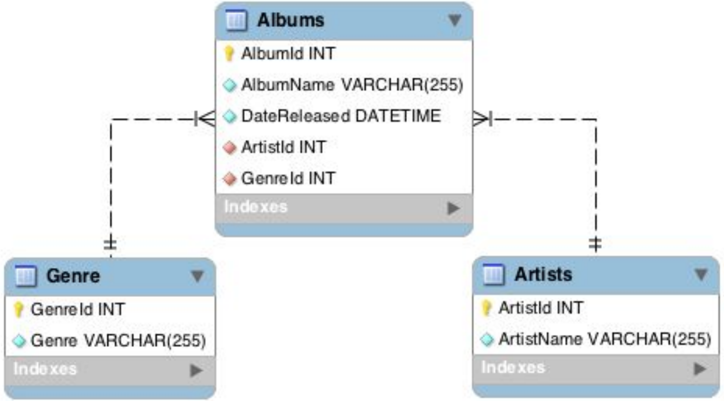
Usage

All the commands used to define the database **schema** when creating, modifying or deleting a table

Schema (of a database): is the **database architecture** defining how data is organized within a **relational database**

Schema properties:

- It can be represented as a **diagram**
- It defines a **consistent formatting** for all data entries
- It defines **unique keys** for all entries and database tables
- It defines column in a table with a **name** and **data type**



List of DDL Commands

Main Commands:

1. **CREATE:** used to **create the database** or its **objects** (like a table, index, function, views ...)
2. **DROP:** used to **delete** the objects from the database, or the database itself
3. **ALTER:** used to **modify** the **structure** of the database, including the column types
4. **TRUNCATE:** used to **remove** all **records** from a table, including the **physical space** allocated
5. **COMMENT:** used to add **comments** to the data dictionary.
6. **RENAME:** used to **rename** objects existing in the database.

DQL: Data Query Language

Usage

All the commands to get **schema relations** based on the specified query

DQL includes only a single command:

- **SELECT:** used to retrieve or fetch data from a database.

With SELECT we can fetch entire tables or only some parts of it.

The data returned is stored in a result table in the front-end application (Client) connected to the database (Server).

DML: Data Manipulation Language

Usage

All the commands dealing with the manipulation of **data entries** present in the database

Main Commands:

- **INSERT** : used to insert data entries into a table.
- **UPDATE**: used to update existing data entries within a table.
- **DELETE** : used to delete data entries from a database table.

Main Commands used with the statement SELECT:

- **FROM:** used to specify the table from which to query data entries from
- **WHERE**: used to define the filtering condition of the data entries to return

DCL: Data Control Language

Usage

All the commands dealing with the rights, permissions, and other controls of the database system.

Commands:

- **GRANT:** used to give users access privileges to the database.
- **REVOKE:** used to withdraw the user’s access privileges given by using the GRANT command.

TCL: Transaction Control Language

Usage

All the commands dealing with the **transactional operations** that can be done with the database

A **transaction** in SQL is a group of tasks executed **atomically** together.
Transactions have only 2 results: **success** or **failure.**

Commands:

- **BEGIN TRANSACTION:** used to indicate the start point of a transaction
- **SET TRANSACTION:** used to specify characteristics for the transaction.
- **COMMIT:** used to commit a transaction.
- **ROLLBACK:** used to undo a transaction in case of any error occurs.

Practical Examples

Basic Queries in SQL

Goal

To write basic SQL explorative queries to execute the tasks in the list below

Dataset: McDonald's Menu with products and some of their nutrition info

Schema: *mcdonald_menu* (category, item, serving_size, calories, calories_from_fat, total_fat)

List of tasks:

1. **Display only values with more than 1000 calories**
2. **Display only values with more than 1000 calories AND of category 'Breakfast'**
3. **Get all products with the smallest calories on all the McDonald's products (**
4. **For each category, display the element having the highest calories** and **sort it descending on calories**
5. **For each category, display the element with the smaller fats, excluding elements with zero fats**