Relational

Structured Query Language (SQL)

DataBase Foundations

Author: Eng. Carlos Andrés Sierra, M.Sc.
cavirguezs@udistrital.edu.co

Lecturer
Computer Engineer
School of Engineering
Universidad Distrital Francisco José de Caldas

2024-III

Outline

1. SQL Introduction
2. Data Definition Language (DDL)

something

subset

Outline

1 SQL Introduction

2 Data Definition Language (DDL)

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

SQL Introduction

Query ⇒ Based → Extraction

- **SQL** (*Structured Query Language*) is a standard language for accessing and manipulating databases.

CUD

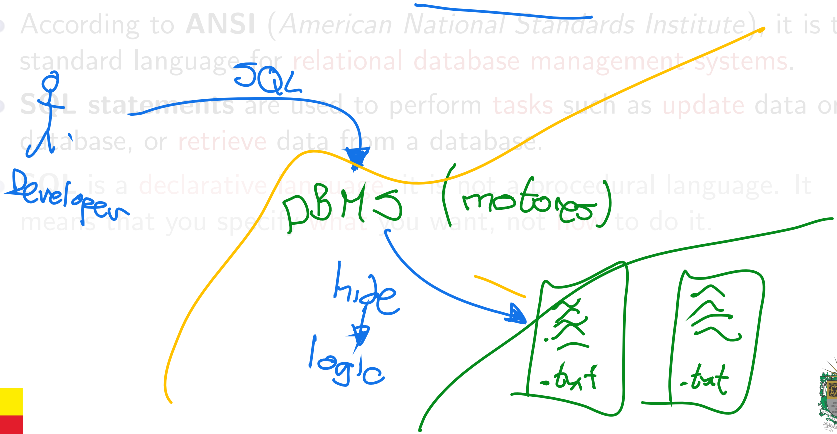
2

Δ Transactions → all operations

- SQL is used to communicate with a database.
- According to **ANSI** (*American National Standards Institute*), it is the standard language for relational database management systems.
- **SQL statements** are used to perform tasks such as update data on a database, or retrieve data from a database.
- **SQL** is a declarative language, it is not a procedural language. It means that you specify what you want, not how to do it.

SQL Introduction

- **SQL** (*Structured Query Language*) is a standard language for accessing and manipulating databases.
- **SQL** is used to communicate with a database.
- According to **ANSI** (*American National Standards Institute*), it is the standard language for relational database management systems.
- **SQL statements** are used to perform tasks such as update data on a database, or retrieve data from a database.
- **SQL** is a declarative language, it is not a procedural language. It means that you specify what you want, not how to do it.



SQL Introduction

- **SQL** (*Structured Query Language*) is a standard language for accessing and manipulating databases.
- **SQL** is used to communicate with a database.
- According to **ANSI** (*American National Standards Institute*), it is the standard language for relational database management systems.
- **SQL statements** are used to perform tasks such as update data on a database, or retrieve data from a database.
- SQL is a declarative language, it is not a procedural language. It means that you specify what you want, not how to do it.

USA Icontec

2 most Popular free DBMS

MySQL ≈ PostgreSQL ≈ SQLite ≈ SQLServer

≈ Oracle

reference

SQL Introduction

- **SQL** (*Structured Query Language*) is a standard language for accessing and manipulating databases.
- **SQL** is used to communicate with a database.
- According to **ANSI** (*American National Standards Institute*), it is the standard language for relational database management systems.
- **SQL statements** are used to perform tasks such as update data on a database, or retrieve data from a database.
- **SQL** is a declarative language, it is not a procedural language. It means that you specify what you want, not how to do it.

↳ operations over all database

SQL Introduction

- **SQL** (*Structured Query Language*) is a standard language for accessing and manipulating databases.
- **SQL** is used to communicate with a database.
- According to **ANSI** (*American National Standards Institute*), it is the standard language for relational database management systems.
- **SQL statements** are used to perform tasks such as update data on a database, or retrieve data from a database.
- **SQL** is a declarative language, it is not a procedural language. It means that you specify what you want, not how to do it.

C, C++, python, Java js

mandatory

not logic as traditional

SQL Syntax & Semantics

- **SQL** is a declarative language that allows you to query and manipulate data in a relational database.

Create (Add data)
Read / Retrieve
Update
Delete

- **SQL** is a standardized language that is used to create, update, delete, and query data in a relational database.
- **SQL** is a set-based language, which means that you can manipulate multiple rows of data at the same time.
- **SQL** is a case-insensitive language, which means that you can write keywords and identifiers in uppercase or lowercase.
- **SQL** is a structured language, which means that you can write statements in a logical order.
- **SQL** is a portable language, which means that you can write statements that will work on different database systems.

SQL Syntax & Semantics

- **SQL** is a declarative language that allows you to query and manipulate data in a relational database.
- **SQL** is a standardized language that is used to create, update, delete, and query data in a relational database.
- **SQL** is a set-based language, which means that you can manipulate multiple rows of data at the same time.

Only (?)

- **SQL** is a case-insensitive language, which means that you can write keywords and identifiers in uppercase or lowercase.
- **SQL** is a structured language, which means that you can write statements in a logical order.
- **SQL** is a portable language, which means that you can write statements that will work on different database systems.

NoSQL

- Function(...)
- + insert()
- select()
- update()
- delete

SQL Syntax & Semantics

- **SQL** is a declarative language that allows you to query and manipulate data in a relational database.
- **SQL** is a standardized language that is used to create, update, delete, and query data in a relational database.
- **SQL** is a set-based language, which means that you can manipulate multiple rows of data at the same time.
- **SQL** is a case-insensitive language, which means that you can write keywords and identifiers in uppercase or lowercase.
- **SQL** is a structured language, which means that you can write statements in a logical order.
- **SQL** is a portable language, which means that you can write statements that will work on different database systems.

theory set

relational algebra

SQL Syntax & Semantics

- **SQL** is a declarative language that allows you to query and manipulate data in a relational database.
- **SQL** is a standardized language that is used to create, update, delete, and query data in a relational database.
- **SQL** is a set-based language, which means that you can manipulate multiple rows of data at the same time.
- **SQL** is a case-insensitive language, which means that you can write keywords and identifiers in uppercase or lowercase.
- **SQL** is a structured language, which means that you can write statements in a logical order.
- **SQL** is a portable language, which means that you can write statements that will work on different database systems.

Select ✓

SELECT ✓

Select ✓

SeLeCt ✓

KEYWORD => UpperCase

name => lowercase

SQL Syntax & Semantics

- **SQL** is a declarative language that allows you to query and manipulate data in a relational database.
- **SQL** is a standardized language that is used to create, update, delete, and query data in a relational database.
- **SQL** is a set-based language, which means that you can manipulate multiple rows of data at the same time.
- **SQL** is a case-insensitive language, which means that you can write keywords and identifiers in uppercase or lowercase.
- **SQL** is a structured language ✓ Syntax, which means that you can write statements in a logical order.
- **SQL** is a portable language, which means that you can write statements that will work on different database systems.

SQL Syntax & Semantics

- **SQL** is a declarative language that allows you to query and manipulate data in a relational database.
- **SQL** is a standardized language that is used to create, update, delete, and query data in a relational database.
- **SQL** is a set-based language, which means that you can manipulate multiple rows of data at the same time.
- **SQL** is a case-insensitive language, which means that you can write keywords and identifiers in uppercase or lowercase.
- **SQL** is a structured language, which means that you can write statements in a logical order.
- **SQL** is a portable language, which means that you can write statements that will work on different database systems.

theory → in practice ~

CRUD Operations

- **CRUD** stands for Create, Read, Update, and Delete.
- **CRUD** operations are the basic operations that you can perform on a database.

all applications

- **CRUD** operations are the building blocks of database management systems.

tasks

- **CRUD** operations are used to query and manipulate data in a relational database.

Outline

1 SQL Introduction

2 Data Definition Language (DDL)

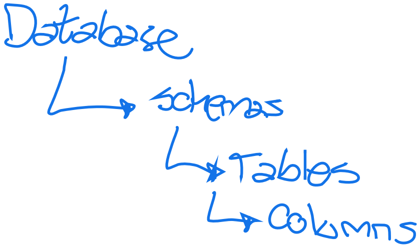
Data Definition Language (DDL)

DDL (*Data Definition Language*) is a subset of **SQL** that is used to define and modify the structure of a database.

PostgreSQL Example — MySQL Example

```
CREATE DATABASE mydatabase;
```

add



Database
- schemas
  - Tables
    - Columns

DDL for Schemas

DDL statements are used to define the schema of a database.

PostgreSQL Example — MySQL Example

```
CREATE SCHEMA mySchema;

DROP SCHEMA mySchema;
```

DDL for Table Creation I

DDL statements are used to define the data types of the columns in a table.

PostgreSQL Example

```
CREATE TABLE myTable (
   id SERIAL PRIMARY KEY,
   name VARCHAR(30)
);

DROP TABLE myTable;
```

DDL for Table Creation II

DDL statements are used to define the data types of the columns in a table.

MySQL Example

```
CREATE TABLE myTable (
   id INT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(30)
);

DROP TABLE myTable;
```

DDL for Table Constraints I

DDL statements are used to define the constraints that enforce the integrity of the data in a table.

PostgreSQL Example

```
CREATE TABLE IF NOT EXISTS myTable (
  id SERIAL PRIMARY KEY,
  name VARCHAR(30) UNIQUE NOT NULL,
  country VARCHAR(20) DEFAULT 'Colombia'
);
```

DDL for Table Constraints II

DDL statements are used to define the constraints that enforce the integrity of the data in a table.

MySQL Example

```
CREATE TABLE myTable (
   id INT AUTO_INCREMENT PRIMARY KEY,
   name VARCHAR(30) UNIQUE NOT NULL,
   country VARCHAR(20) DEFAULT 'Colombia'
);
```

UNIVERSIDAD DISTRITAL
FRANCISCO JOSÉ DE CALDAS

DDL for Table Modifications

DDL statements are used to alter database objects such as tables, indexes, and views.

PostgreSQL Example

```
ALTER TABLE myTable ADD COLUMN email VARCHAR(50);
ALTER TABLE myTable ALTER COLUMN name
    TYPE VARCHAR(100);
```

MySQL Example

```
ALTER TABLE myTable ADD COLUMN email VARCHAR(50);
ALTER TABLE myTable MODIFY COLUMN name VARCHAR(100);
```

Outline

1. SQL Introduction

2. Data Definition Language (DDL)

Thanks!

Questions?

Repo: *https://github.com/EngAndres/ud-public/tree/main/courses/databases-foundations*