3. Structured Query Language

www.LearnDB.com

Dr. Imed Bouchrika
Dept of Mathematics & Computer Science
University of Souk-Ahras
imed@imed.ws

SQL

- SQL = Structured Query Language
- The ANSI standard language for the definition and manipulation of ***relational database***.
- Includes data definition language (DDL), statements that specify and modify database schemas.
- Includes a data manipulation language (DML), statements that manipulate database content

SQL

- SQL History
  - 1970 : Creation of SQL by IBM
  - 1977 : IBM Sequel, first database using such system
  - 1979 : Launch of Oracle SQL RDBMS
  - 1986 : Normalisation of SQL1 (SQL-86)
  - 1989 : Extension of SQL1 (SQL-89)
  - 1992 : Normalisation of SQL2 (SQL-92)
  - 1999 : Normalisation of SQL3

Data Definition Language

- CREATE SCHEMA : used to create a new database
- CREATE TABLE: used to create a table.
- ALTER TABLE: modifies a table after it was created.
- DROP TABLE: removes a table from a database.
- DROP SCHEMA : to delete a database.

CREATE SCHEMA

- To create a database, we use the statement :

  CREATE SCHEMA abc ;

- In MySQL, we use the statement :

  CREATE DATABASE abc ;

- To select a database for usage →

  USE DATABASE abc ;

CREATE TABLE

- To create a database table, we use the syntax:

```
CREATE TABLE client (
        client_id INT NOT NULL AUTO_INCREMENT,
        NOM VARCHAR( 20 ) NOT NULL,
        PRENOM VARCHAR( 20 ) ,
        ADRESSE  TEXT,
        AGE INT DEFAULT 18,
        CITY VARCHAR(100) DEFAULT 'Boston',
        PRIMARY KEY(client_id)
    );
```

- Auto_increment : is used mostly for MySQL.

CREATE TABLE

- Possible SQL Data Types:
  - CHAR
  - VARCHAR
  - INT
  - INTEGER
  - NUMBER
  - DECIMAL
  - FLOAT
  - DOUBLE
  - DATE
  - TIME
  - TIMESTAMP
  - BIGINT
  - BLOB
  - TEXT

DESCRIBE

- To find the structure of an existing table:

  DESCRIBE client

- **Or**

  DESC client

INSERT

- To insert data into a database table:

INSERT INTO abc **(column1, column2) VALUES ('value1',value2');**

- Example:

insert into clients (name, email, city) values
('Mike', 'mike@gmail.com', 'LA')

UPDATE

- The SQL UPDATE Query is used to modify the existing records in a table.

- You can use WHERE clause with UPDATE query to update selected rows otherwise all the rows would be affected.

- **Syntax**

  UPDATE table_name

  SET column1 = value1, column2 = value2...., columnN = valueN

  WHERE [condition];

- **Example:**

UPDATE clients ***SET*** ADDRESS = 'Pune' ***WHERE*** ID = 6;

DELETE

- The SQL DELETE Query is used to delete existing records in a table.

- You can use WHERE clause with DELETE query to delete selected rows otherwise all the rows would be removed from the table.

- **Syntax**

  DELETE FROM table_name WHERE [condition];

- **Example:**

  DELETE FROM clients ***WHERE*** ID = 6;

DROP

- The SQL DROP DATABASE statement is used to drop an existing database in SQL schema.

DROP DATABASE abc;

- The SQL DROP TABLE statement is used to remove :
  - a table definition
  - all data
  - Indexes
  - Triggers
  - Constraints
  - permission specifications.

DROP TABLE clients;

ALTER

- The SQL ALTER TABLE command is used to add, delete or modify columns in an existing table.

- You can also use ALTER TABLE command to add and drop various constraints on a an existing table.

- Examples:
  - **ALTER TABLE** clients **ADD** email varchar(20);
  - **ALTER TABLE** clients **DROP COLUMN** age;
  - **ALTER TABLE** clients **CHANGE** age dob date NOT NULL
  - ….

SELECT

- SQL SELECT statement is used to search and retrieve the data from a database table

- These result tables are called result-sets.

- Simple Syntax :
  - SELECT * FROM clients;

Displays all results with all attributes (id, name, age, address, ...)

  - SELECT id, name FROM client;

Displays all results showing only id and email for every client.

SELECT : WHERE

- The SQL WHERE clause is used to specify a condition while fetching the data from single table or joining with multiple tables.
- If the given condition is satisfied then only it returns specific value from the table.
- The WHERE clause is not only used in SELECT statement, but it is also used in UPDATE, DELETE statement

SELECT : WHERE

- Syntax:

```
SELECT column1, column2, columnN
FROM table_name
WHERE [condition]
```

- Examples
  - Select * from clients where id=6
  - Select id, email from clients where id>6
  - Select id, name, email from clients where id<>6

SELECT : Logical Operators

- The SQL AND and OR operators are used to combine multiple conditions to narrow data in an SQL statement.

- Syntax:

  select * from table where
  (condition) and (condition) or (condition)

- Example:
  - Select * from clients where id>6 **AND** name='abc'
  - Select id, email from clients where
    (id>6 **AND** name='abc') **OR** email='b@c'

SELECT : LIKE

- The **SQL LIKE** clause is used to compare a value to similar values using ***wildcard*** operators.

- There are two wildcards used with the LIKE operator:
  - The percent sign (%):
    The percent sign represents zero, one, or multiple characters.

  - The underscore (_)
    The underscore represents a single number or character.

SELECT : LIKE

- Further Examples:
  - Select * from clients where name like ‘a%’
  - Select * from clients where name like ‘%a’
  - Select * from clients where name like ‘%a%’
  - Select * from clients where name like ‘_a’
  - Select * from clients where name like ‘_a_’
  - Select * from clients where name like ‘_%a’
  - Select * from clients where name like ‘_%’

SELECT : Order By

- The SQL **ORDER BY** clause is used to sort the data in ascending or descending order.
- The ordering is based on one or more columns.
- Some database sorts results in **ascending** order by default.
- Syntax:
  SELECT column-list FROM table_name
  [WHERE condition]
  [**ORDER BY** column1, column2, .. columnN] [**ASC** | **DESC**];

SELECT : Order By

- Examples:
  - Select * from clients order by id ASC
  - Select * from clients order by id DESC
  - Select * from clients where 1=1 order by name, email DESC

SELECT : LIMIT or TOP or ROWNUM

- The SQL **TOP** clause is used to fetch a TOP N number or X percent records from a table.
- All the databases do not support TOP clause.
- For example MySQL supports LIMIT clause to fetch limited number of records.
- Oracle uses ROWNUM to fetch limited number of records.

SELECT : LIMIT or TOP or ROWNUM

- Examples:
  - SELECT TOP 3 * FROM clients;
  - SELECT * FROM clients LIMIT 3;
  - SELECT * FROM clients LIMIT 3,10;
  - SELECT * FROM clients WHERE ROWNUM <= 3;

SELECT : Distinct

- The SQL **DISTINCT** keyword is used with SELECT statement to eliminate all the ***duplicate*** records and fetching only ***unique*** records.

- Example:
  - select **distinct name** from clients

SELECT : Between

- The BETWEEN operator selects values within a range.

- The values can be
  - Numbers
  - Text
  - dates.

- Example:
  - Select * from clients where id **between 7 and 10**

  - Select * from clients where dob **between '2010-06-1' and '2010-06-30'**

SELECT : in (...)

- The IN operator allows you to specify multiple values in a WHERE clause.
- To inverse, use **NOT in**
- Examples:
  - Select * from clients where id in (1 , 2 , 3 , 4)
  - Select * from clients where city in ('LA', 'Boston')
  - Select * from clients where id not in (1 , 2 , 3 , 4)

SELECT : AS

- SQL aliases are used to give a database table, or a column in a table, a temporary name.
- Basically aliases are created to make column names more readable.
- Examples:
  - Select first_name **as fn** , last_name **as ln** from clients
  - Select * from clients **as c**

For you to search !

- **Check .**
- **Blob**
- **Unique vs Primary key vs Index**
- **CREATE VIEW**
- **ANSI**