Unit 4: Introduction to SQL

- What SQL is
- Data Definition Language
- Database creation, modification and removal
- Creating, updating and deleting tables
- Data Manipulation Language
- Integrity constraints

What SQL is

- Stands for: Structured Query Language
- Created by IBM in the 1970s based on the relational model
- It uses operations based on relational algebra and calculus
- Standard with multiple open-source, free and commercial implementations: MySQL, Oracle, MariaDB, PostgreSQL, SQLite, etc.
- Environment → interactive command line
- It can be easily integrated into programming languages

SQL

- SQL is a standard (check ISO SQL-92 or SQL:2016 versions)
- However, the implementations (either free or commercial) have taken the liberty of adding specific plug-ins

SQL elements

Basic data types:

- Numeric → number, integer, real
- Null values → NULL
- Dates → date ('yyyy-mm-dd')
- Characters → char, text, varchar
- Boolean → true, false
- Other types of data → composed by other types

SQL elements

- Delimiters: they are used to separate the language elements from each other:
  - Blank space
  - Comma (,)
  - Semicolon (;)
  - Parentheses → used to group expressions
  - Single and double quotes (‘, “)
  - Etc. (:, =, and others)
- Comparison operators: =, !=, <>, >, <, >=, <=
- Comments: written with -- (single line) or /* */ (multi-line)

SQL elements

- Expression: a valid combination of operators and operands (just like in a programming language)
- Examples: ‘Pepe’, 2 + 3, (age > 18)
- Predicate: a combination of several conditions using operators producing a result of boolean type (either true or false) when evaluated
- Examples: age > 18, address IS NOT NULL, name <> ‘Pepe’ AND last_name = ‘García Fernández’

Data Definition Language

- Known as DDL → Data Definition Language
- As its name points out, it is the module that allows us to define data structures
- Tip before starting with the data description: check out what there already is present
- That is done via operations that are not part of standard SQL but are usually implemented in DBMS:
  - show databases / tables / views
  - /list /dt

Data Definition Language

- The following operations are not part of the standard language
- Creating a new database:

  CREATE DATABASE <name> [options]
- Deletion of an existing database:

  DROP DATABASE <name>
- There is also the ALTER DATABASE statement to update aspects of an already existing table

Data Definition Language

- Table creation

  CREATE TABLE <identifier>(

  [constraint-1] <attribute1> <type1> [modifier1],

  ...

  [constraint-n] <attribute-n> <type-n> [modifier-n]

  );

Data Definition Language

- Types: char, number, integer, real, date, long, text
- Modifiers: null, not null, unique
- Constraints: primary key, foreign key references <table> [attribute]
- Example:

  CREATE TABLE customers(ID number(8) primary key, name char(30) not null, address text, telephone number(9));

Data Definition Language

- Modification:

  ALTER TABLE <table> ADD (<attribute> <type>);
- Example:

  ALTER TABLE customers ADD (registered_from date);
- Elimination:

  DROP TABLE <table name>;
- Example:

  DROP TABLE customers;

Data Manipulation Language

- Data entry:

  INSERT INTO <table name> [<list of attributes>] VALUES(<list of values>);
- List of attributes or values are comma separated
- Example:

  INSERT INTO customers VALUES(11111111, 'Juan García Pérez', 'c/ Gran vía, 27', 123456789, '2019-06-01');

Data Manipulation Language

- Data deletion:

  DEETE [*] FROM <table name> [WHERE <condition>]

- Example:

  DELETE FROM customers WHERE registered_from < '2021-01-01';

Data Manipulation Language

- Modification:

  UPDATE <table name> SET <attribute-1> = <value-1>, …, <attribute-n> = <value-n> [WHERE <condition>];

- Example:

  UPDATE customers SET name = 'Juan Antonio García Pérez' WHERE ID = 11111111;

Integrity constraints

- They are conditions imposed (directly or indirectly) on the database about the data sought to be represented
- They are managed either by the database designer or the administrator (DBA)
- Types:
  - Keys constraints → primary and candidate keys, non-null and non-repeated values
  - Referential constraints → foreign keys, check that a value must exist
  - User constraints → they are specific to each situation, and given the the database designer according to the needs

Integrity constraints

SQL completeness:

- Primary and candidate keys (PRIMARY KEY, UNIQUE)
- Non-null values → NOT NULL modifier
- Foreign keys (FOREIGN KEY … REFERENCES)
- Data types (domains) → through the CREATE DOMAIN instruction

Integrity constraints

- Domain creation example:

  CREATE DOMAIN province_extremadura AS text CHECK(value IN ('Badajoz', 'Cáceres'));

  → then…

  CREATE TABLE town(name char(50) primary_key, province province_extremadura);

Advanced constraints

- CHECK clause: this data modifier allows us to perform checks on attribute values to verify that they fulfil a condition.
- Example:

  CREATE TABLE employee(… age INTEGER CHECK (age > 18));

Advanced constraints

- Assertions: their purpose is to set conditions on a broader, more extensive part of the database
- They are used within triggers or procedures
- Example:

```
BEGIN
ASSERT (SELECT count(*) FROM employees) >= 2,
'There must be at least two employees');
END;
```