Semana 1 – Fundamentos y primeros comandos SQL

Objetivos de la semana

- Comprender qué es SQL y por qué es fundamental en la gestión de bases de datos.
- Diferenciar el paradigma declarativo de los lenguajes imperativos.
- Conocer los sublenguajes básicos de SQL (DDL, DML, DCL).
- Crear tablas simples e insertar registros.
- Realizar consultas básicas con SELECT.

1. ¿Qué es SQL?

El Structured Query Language (SQL) es el lenguaje estándar para trabajar con bases de datos relacionales. Fue creado en los años 70 y, desde entonces, se convirtió en la forma más extendida de almacenar y consultar información.

A diferencia de otros lenguajes de programación, SQL no nos obliga a describir paso a paso lo que debe hacer la computadora. En lugar de eso, expresamos directamente qué información necesitamos, y el motor de la base de datos se encarga de planificar la mejor forma de obtenerla.

Por ejemplo, en un lenguaje imperativo (como Python) deberíamos recorrer una lista de registros con un bucle, verificar condiciones y armar una nueva lista con los resultados. En SQL, en cambio, con una sola instrucción SELECT podemos expresar la misma necesidad sin preocuparnos por el procedimiento exacto.

2. Sublenguajes de SQL

SQL no es un único bloque monolítico, sino que se organiza en "sub-lenguajes" que agrupan distintos tipos de operaciones:

- **DDL (Data Definition Language)**: se ocupa de definir la estructura de la base de datos. Incluye comandos como CREATE, ALTER, DROP. Con DDL diseñamos las tablas, especificamos sus columnas, claves primarias y foráneas.
- **DML (Data Manipulation Language)**: se centra en los datos que viven dentro de esas estructuras. Sus comandos principales son SELECT, INSERT,

UPDATE, DELETE. Gracias al DML consultamos y modificamos la información.

- **DCL (Data Control Language)**: menos usado en los primeros pasos, regula permisos y accesos a los objetos de la base de datos (GRANT, REVOKE).

En esta primera semana trabajaremos fundamentalmente con DDL y DML, ya que son la base de todo.

3. Creación de tablas (DDL)

El primer paso para armar una base de datos es definir las tablas. Una tabla es como una hoja de cálculo: tiene columnas con un nombre y un tipo de dato, y filas que representan registros.

Ejemplo: definimos una tabla de autores y otra de libros, donde cada libro hace referencia a un autor mediante una clave foránea (FOREIGN KEY).

```
CREATE TABLE Autores (
    AutorID INT PRIMARY KEY,
    Nombre VARCHAR(100) NOT NULL
);

CREATE TABLE Libros (
    LibroID INT PRIMARY KEY,
    Titulo VARCHAR(150) NOT NULL,
    AutorID INT,
    FOREIGN KEY (AutorID) REFERENCES Autores(AutorID)
);
```

Con esto ya tenemos la base para registrar autores y sus libros.

4. Primeras consultas con SELECT

El comando SELECT es el corazón de SQL. Con él pedimos información de una o varias tablas.

- Ver todos los registros de una tabla:

```
SELECT * FROM Autores;
```

El asterisco indica “todas las columnas”.

- Seleccionar columnas específicas:

```
SELECT Nombre FROM Autores;
```

De esta forma limitamos la salida a los datos que realmente nos interesan.

- Filtrar resultados con WHERE:

```
SELECT Titulo FROM Libros
WHERE AutorID = 1;
```

En este caso pedimos solo los libros del autor con identificador 1.

5. Insertar registros (INSERT)

Para que las consultas tengan sentido necesitamos cargar datos. El comando INSERT agrega filas nuevas en una tabla.

```
INSERT INTO Autores (AutorID, Nombre)
VALUES (1, 'Isabel Allende');

INSERT INTO Libros (LibroID, Titulo, AutorID)
VALUES (10, 'La Casa de los Espíritus', 1);
```

Al ejecutar estas sentencias, la tabla de autores tendrá un registro, y la de libros un libro asociado a ese autor.

6. Actividad práctica

1. Crear una tabla llamada Clientes con los campos: ClienteID (clave primaria), Nombre, Email.
2. Insertar al menos tres registros.
3. Hacer un SELECT para mostrar todos los clientes.
4. Hacer un SELECT que muestre solo los clientes con un email que termine en @gmail.com.