

## Lenguaje SQL – Material complementario

### 1. Fundamentos de SQL y Estructura

**SQL (Structured Query Language)** es el lenguaje estándar para interactuar con bases de datos relacionales. Su objetivo principal es permitir la gestión y manipulación de datos, definiendo la estructura de la base de datos y controlando el acceso a la información.

SQL se divide en varias categorías de comandos:

- **DML (Data Manipulation Language):** Se usa para manipular los datos dentro de las tablas. Incluye comandos como SELECT, INSERT, UPDATE y DELETE.
- **DDL (Data Definition Language):** Se utiliza para definir la estructura de la base de datos. Incluye comandos como CREATE, ALTER y DROP para tablas, vistas, índices, etc.
- **DCL (Data Control Language):** Controla el acceso a los datos. Los comandos principales son GRANT y REVOKE para otorgar o quitar permisos.
- **TCL (Transaction Control Language):** Gestiona las transacciones de datos. Los comandos principales son COMMIT y ROLLBACK.

### 2. Sentencias Básicas de Manipulación de Datos

Las sentencias DML son las más comunes y fundamentales para cualquier tarea con bases de datos.

#### SELECT

La sentencia SELECT se usa para consultar y recuperar datos de una o más tablas. Es la base de cualquier consulta.

- **Sintaxis básica:** SELECT columna1, columna2 FROM tabla;
- **Recuperar todas las columnas:** SELECT \* FROM tabla;
- **Filtros con WHERE:** Se usa para filtrar filas basándose en una condición. Ejemplo:  
SELECT \* FROM empleados WHERE salario > 50000;
- **Ordenar resultados con ORDER BY:** Ordena el conjunto de resultados. Ejemplo:  
SELECT nombre FROM productos ORDER BY precio DESC;

#### INSERT

La sentencia INSERT se utiliza para agregar nuevas filas de datos a una tabla.

- **Sintaxis básica:** INSERT INTO tabla (columna1, columna2) VALUES (valor1, valor2);
- **Insertar todas las columnas:** Si se insertan valores en todas las columnas de la tabla en el mismo orden en que fueron definidas, no es necesario especificar los nombres de las columnas. Ejemplo: INSERT INTO empleados VALUES ('Juan', 'Pérez', 60000);

## UPDATE

La sentencia UPDATE se usa para modificar los datos existentes en una o más filas de una tabla. Es crucial usar la cláusula WHERE para evitar actualizar todas las filas de la tabla.

- **Sintaxis básica:** UPDATE tabla SET columna1 = nuevo\_valor WHERE condicion;
- **Ejemplo:** UPDATE productos SET precio = 15.50 WHERE id\_producto = 101;

## DELETE

La sentencia DELETE se usa para eliminar filas existentes de una tabla. Al igual que con UPDATE, se debe usar WHERE para evitar borrar todas las filas.

- **Sintaxis básica:** DELETE FROM tabla WHERE condicion;
- **Ejemplo:** DELETE FROM empleados WHERE id\_empleado = 205;

## 3. Consultas Avanzadas en SQL

Para manejar tareas más complejas, SQL ofrece herramientas avanzadas que permiten combinar y analizar datos de formas sofisticadas.

### JOINS

Como se describió en el material anterior, los JOIN se usan para combinar filas de dos o más tablas basándose en una columna relacionada entre ellas. Los tipos principales son INNER JOIN, LEFT JOIN y RIGHT JOIN. Por ejemplo, para obtener el nombre de un empleado y el nombre de su departamento desde dos tablas distintas (empleados y departamentos), se usaría un JOIN.

### Subconsultas (Subqueries)

Las subconsultas son consultas anidadas dentro de otra consulta. Pueden ser utilizadas en cláusulas como WHERE, FROM o SELECT para filtrar o seleccionar datos de manera más dinámica.

- **Ejemplo:** SELECT nombre FROM empleados WHERE id\_departamento IN (SELECT id\_departamento FROM departamentos WHERE ubicacion = 'Ventas');

### Funciones de Agregación

Estas funciones realizan un cálculo en un conjunto de valores y devuelven un único valor. Son comunes para resumir datos.

- **COUNT():** Cuenta el número de filas.
- **SUM():** Calcula la suma total de una columna numérica.
- **AVG():** Calcula el promedio de una columna numérica.
- **MAX():** Encuentra el valor más alto en una columna.
- **MIN():** Encuentra el valor más bajo en una columna.

- **GROUP BY:** Se utiliza para agrupar filas que tienen los mismos valores en columnas específicas, lo que permite que las funciones de agregación operen en cada grupo.
  - **Ejemplo:** SELECT departamento, COUNT(\*) FROM empleados GROUP BY departamento;

## Vistas (Views)

Las vistas son tablas virtuales que no almacenan datos, sino que representan el resultado de una consulta SQL almacenada. Son útiles para simplificar consultas complejas, restringir el acceso a datos y mejorar la seguridad.

- **Creación de una vista:** CREATE VIEW vista\_empleados\_salarios AS SELECT nombre, salario FROM empleados WHERE salario > 60000;
- Una vez creada, puedes consultar la vista como si fuera una tabla normal: SELECT \* FROM vista\_empleados\_salarios;