

TRABAJO TEORICO 1



Asignatura: Laboratorio IV

Trabajo: DML *Data manipulation language*

Año: 2024

Alumno: Gallac Leandro Gabriel

1. ¿Qué diferencias hay entre SQL, Base de Datos y SGBD? de ejemplos de cada uno.

SQL: “*Structured Query Language*”, es un lenguaje de programación utilizado para gestionar y manipular, bases de datos relacionales. Es el estándar para interactuar con bases de datos, permitiendo realizar operaciones como consultas, inserciones, actualizaciones, y eliminación de datos en una base de datos relacional.

Base de datos: Una base de datos es una colección organizada de datos que puede incluir tablas, índices, vistas, procedimientos almacenados, etc. Los datos están estructurados de manera que se pueda acceder, gestionar y actualizar fácilmente. Las bases de datos pueden ser relacionales o no relacionales, siendo las primeras las que organizan los datos en tablas que se relacionan entre sí.

Algunos ejemplos de Base de datos:

- Base de Datos de Empleados: Contiene una tabla con los datos de los empleados de una empresa con atributos como nombre, edad, departamento, etc.
- Base de Datos de un Banco: Contiene tablas para almacenar información de clientes, cuentas, transacciones, préstamos, etc.

SGBD: Son las siglas de “*Sistema de gestión de base de datos*”, es el software que permite crear, administrar y gestionar bases de datos relacionales. Es el responsable de todas las operaciones relacionadas con el almacenamiento, acceso, manipulación y protección de los datos. Además, se encarga de asegurar la integridad y consistencia de los datos, así como la seguridad.

Algunos ejemplos de SGBD:

- MySQL;
- PostgreSQL;
- Microsoft SQL Server;
- Oracle Database;

Relación entre SQL, Base de Datos y SGBD:

- **SQL:** Es el lenguaje que se usa para interactuar con los datos.
- **Base de Datos:** Es el lugar donde se almacenan los datos de manera estructurada, como tablas, relaciones, etc.
- **SGBD:** Es el software que te permite gestionar y operar sobre la base de datos usando SQL.

2. Describa los tipos de datos que soporta el ANSI SQL.

ANSI SQL “*American National Standards Institute SQL*” es un estándar utilizado para la creación y gestión de bases de datos relacionales. Este estándar define varios tipos de datos que pueden ser utilizados para definir las columnas de las tablas. Estos tipos de datos se dividen en varias categorías, cada una diseñada para almacenar diferentes tipos de información. Los tipos de datos principales que soporta ANSI SQL son:

- Numerico
- Datos de texto
- Fecha y hora
- Logico
- Binarios
- Otros

3. ¿Cuáles son los grupos de sentencia de SQL y que sentencias comprende cada uno? describa brevemente para qué sirve cada sentencia

En el lenguaje SQL podemos destacar tres grupos de sentencias:

- **DML, *Data manipulation language*;**
 - ➔ Insert: Inserta nuevos registros en una tabla;
 - ➔ Delete: Elimina registros específicos de una tabla;
 - ➔ Update: Actualiza registros existentes en una tabla;
 - ➔ Select: Recupera datos de una o más tablas.

- **DDL, *Data declaration language***;
 - Create: Crea una nueva tabla o base de datos;
 - Alter: Modifica una tabla o base de datos existente;
 - Drop: Elimina una tabla o base de datos;
 - Truncate: Vacía una tabla eliminando todos sus registros sin borrar la estructura.
- **DCL, *Data control language***;
 - Grant: Concede permisos a los usuarios para realizar ciertas acciones en la base de datos;
 - Revoke: Retira permisos previamente otorgados a los usuarios;
 - Commit: Confirma las transacciones, haciendo permanentes los cambios realizados;
 - Rollback: Deshace los cambios realizados por una transacción antes de que se confirme.

4. ¿Qué es una cláusula? de dos ejemplos indicando que función cumplen en cada una.

Una *cláusula* en SQL es una parte de una sentencia que especifica condiciones o modificadores adicionales para la ejecución de una consulta. Estas nos permiten refinar los resultados y definir condiciones específicas para la manipulación de los datos.

- **Ejemplo 1: WHERE**: Filtra los registros que cumplen una condición específica.

```
SELECT * FROM Padrón WHERE Edad > 30;
```

Salida: Selecciona solo los registros donde la edad es mayor a 30.

- **Ejemplo 2: ORDER BY**: Ordena los resultados de una consulta de acuerdo con uno o más criterios.

```
SELECT Nombre FROM Padrón ORDER BY Edad DESC;
```

Salida: Ordena los nombres del padrón de mayor a menor según la edad.

5. ¿Cuál es la jerarquía de contenedores con la que se organizan las Bases de Datos?

La jerarquía de contenedores en bases de datos SQL generalmente sigue el siguiente esquema:

- I. **Servidor:** El contenedor que administra las bases de datos.
- II. **Base de datos:** Conjunto organizado de datos que contiene tablas, vistas, índices, procedimientos almacenados, etc.
- III. **Esquema:** Organización lógica dentro de una base de datos que agrupa objetos como tablas y vistas.
- IV. **Tabla:** Conjunto de registros organizados en filas y columnas.
- V. **Filas (Registros):** Unidades de datos dentro de una tabla.
- VI. **Columnas (Campos):** Atributos de los registros en una tabla.

6. ¿Cuáles son las funciones de agregación?. Dé dos ejemplos de uso.

Las funciones de agregación en SQL se usan para realizar cálculos sobre un conjunto de valores y devolver un solo valor.

➤ **Ejemplos:**

- **AVG:** Calcula el promedio de una columna numérica.

```
SELECT AVG(Edad) FROM Padron;
```

Salida: Devuelve el promedio de las edades en la tabla Padrón.

- **SUM:** Calcula la suma de una columna numérica.

```
SELECT SUM(Edad) FROM Padron;
```

Salida: Devuelve la suma de las edades en la tabla Padrón.

7. ¿Qué es una restricción de clave primaria y que es la restricción de clave foránea? De un ejemplo de cada una.

Clave primaria: Una clave primaria es un conjunto de uno o más campos (columnas) que identifican de manera única cada fila de una tabla. No puede haber valores nulos en estos campos y no pueden repetirse.

➤ Ejemplo:

```
CREATE TABLE Estudiantes ( ID INT PRIMARY  
KEY, Nombre VARCHAR(100), Edad INT );
```

Aquí, **ID** es la clave primaria, lo que garantiza que cada estudiante tenga un identificador único.

Clave foránea: Una clave foránea es una columna o conjunto de columnas que establece una relación entre dos tablas. Su valor debe coincidir con una clave primaria en otra tabla, asegurando así la integridad referencial.

➤ Ejemplo:

```
CREATE TABLE Inscripciones ( IDInscripcion  
INT PRIMARY KEY, IDEstudiente INT, Curso  
VARCHAR(100), FOREIGN KEY (IDEstudiante)  
REFERENCES Estudiantes(ID) );
```

En este caso, **IDEstudiante** es una clave foránea que se refiere a la clave primaria **ID** de la tabla **Estudiantes**.

8. Escriba el formato completo de cada una de las sentencias DML y de un ejemplo de uso de cada una sobre la tabla Padrón.

Recordando: Las sentencias **DML (Data Manipulation Language)** se utilizan para manipular los datos dentro de las tablas.

→ **INSERT:** Inserta nuevos registros en la tabla.

```
INSERT INTO Padrón (ID, Nombre) VALUES  
(1, 'Juan');
```

→ **UPDATE:** Actualiza registros existentes en la tabla.

```
UPDATE Padrón SET Nombre = 'Juan Pérez'  
WHERE ID = 1;
```

→ **DELETE:** Elimina registros de la tabla.

```
DELETE FROM Padrón WHERE ID = 1;
```

→ **SELECT:** Recupera datos de la tabla.

```
SELECT * FROM Padrón;
```

9. ¿Qué es una base de datos de sistema y que es una base de datos de usuario? De un ejemplo de cada una.

- **Base de datos de sistema:** Son bases de datos que almacenan información del sistema de gestión de bases de datos. Contienen metadatos sobre las bases de datos y su estructura.

Ejemplo: La base de datos `mysql` en MySQL, que almacena información sobre usuarios y permisos.

- **Base de datos de usuario:** Son bases de datos creadas por usuarios para almacenar datos específicos de una aplicación o negocio.

Ejemplo: Una base de datos llamada `Escuela`, que contiene tablas como `Estudiantes`, `Cursos`, y `Inscripciones`.

10. Escriba la sentencia que me devuelva del padrón todos los nombres que tengan la letra "M" pero que no tengan la letra "O".

Para obtener todos los nombres en la tabla **Padrón** que contengan la letra "M" pero no la letra "O", podemos usar la siguiente consulta SQL:

```
SELECT Nombre FROM Padrón WHERE Nombre LIKE  
'%M%' AND Nombre NOT LIKE '%O%';
```