GUIA DE APRENDIZAJE BASES DE DATOS NUMERO 1

JUAN MANUEL QUIAZUA ORDUZ
1.000.271.734

SENA – 2996234

DEMETRIO MAURICIO ESTUPIÑAN

MAYO 5 DE 2025

1.Activdad De reflexión Inicial

¿Qué es un SGBD (Sistema Gestor de Bases de Datos)?

Un SGBD (Sistema Gestor de Bases de Datos), o DBMS por sus siglas en inglés (Database Management System), es un conjunto de programas (software) que permite a los usuarios crear, mantener, controlar y acceder a bases de datos de manera eficiente y segura. Actúa como una interfaz entre el usuario (o las aplicaciones) y la base de datos física almacenada en el disco.

Su objetivo principal es facilitar la gestión de grandes volúmenes de información, garantizando su integridad, seguridad y disponibilidad.

Características Principales:

Independencia de Datos: Separa la definición lógica y física de los datos. Los cambios en la forma en que se almacenan los datos físicamente no deberían afectar a las aplicaciones que los usan (independencia física), y los cambios en la estructura lógica (añadir un campo) pueden hacerse con mínimo impacto (independencia lógica).

Control de Redundancia: Intenta minimizar la duplicación innecesaria de datos, aunque permite redundancia controlada para mejorar el rendimiento (ej. índices).

Integridad de Datos: Permite definir reglas (restricciones) para asegurar que los datos almacenados sean correctos y consistentes (ej. un campo numérico no puede tener texto, una clave primaria no puede ser nula).

Seguridad: Proporciona mecanismos para controlar el acceso a los datos, definiendo quién puede ver o modificar qué información (autenticación y autorización).

Control de Concurrencia: Gestiona el acceso simultáneo de múltiples usuarios a la base de datos, evitando conflictos y asegurando que las transacciones se realicen de forma aislada y consistente.

Respaldo y Recuperación: Ofrece herramientas para crear copias de seguridad (backups) y restaurar la base de datos en caso de fallos (hardware, software, errores humanos).

Lenguajes de Consulta: Proporciona lenguajes (como SQL) para definir, manipular y consultar los datos de forma estructurada.

Ventajas:

1. Gestión centralizada de datos: Simplifica la administración.
2. Consistencia e Integridad: Asegura la calidad de los datos.
3. Seguridad mejorada: Protege la información sensible.
4. Acceso eficiente: Optimiza las consultas y la recuperación de datos.
5. Compartición de datos: Facilita que múltiples usuarios y aplicaciones accedan a la misma información actualizada.
6. Reducción del tiempo de desarrollo: Proporciona funcionalidades estándar para la gestión de datos.

Desventajas:

1. Costo: El software SGBD (especialmente los comerciales), el hardware necesario y el personal especializado pueden ser caros.
2. Complejidad: Diseñar y administrar una base de datos requiere conocimientos técnicos.
3. Rendimiento: Un SGBD mal configurado o una base de datos mal diseñada pueden convertirse en un cuello de botella.
4. Sobrecarga: Para bases de datos muy pequeñas y simples, la overhead (recursos consumidos por el propio SGBD) puede ser excesiva.
5. Vulnerabilidad centralizada: Si el SGBD falla, todo el acceso a los datos se detiene.

SGBD Conocidos

Se suelen clasificar principalmente en Relacionales y NoSQL:

SGBD Relacionales (usan SQL principalmente):

- MySQL: Muy popular, de código abierto.
- PostgreSQL: Avanzado, de código abierto, muy robusto y extensible.
- Microsoft SQL Server: Solución comercial de Microsoft, muy usada en entornos Windows.
- Oracle Database: Solución comercial muy potente y escalable, líder en entornos corporativos grandes.
- SQLite: Base de datos embebida, muy ligera, usada en móviles y aplicaciones de escritorio.
- MariaDB: Fork de MySQL, de código abierto.

SGBD NoSQL (usan diversos modelos y lenguajes):

- MongoDB: Base de datos orientada a documentos (JSON-like).
- Cassandra: Base de datos columnar distribuida, alta disponibilidad.
- Redis: Base de datos en memoria clave-valor, muy rápida.
- Couchbase: Base de datos documental y clave-valor distribuida.
- Neo4j: Base de datos orientada a grafos.

¿Qué es SQL?

SQL (Structured Query Language - Lenguaje de Consulta Estructurado) es el lenguaje estándar utilizado para comunicarse con bases de datos relacionales. Permite realizar diversas operaciones como:

- Consultar datos (recuperar información).
- Insertar nuevos datos.
- Actualizar datos existentes.
- Eliminar datos.
- Crear, modificar y eliminar estructuras de la base de datos (tablas, vistas, índices).
- Controlar el acceso a los datos.

Es un lenguaje declarativo, lo que significa que el usuario especifica qué quiere hacer, pero no cómo debe hacerlo el SGBD (el motor de la base de datos se encarga de optimizar la ejecución).

¿Qué es DDL – DML; sentencias y ejemplos?

DDL y DML son subconjuntos del lenguaje SQL, cada uno enfocado en un tipo diferente de tareas:

DDL (Data Definition Language – Lenguaje de Definición de Datos)

Se utiliza para definir, modificar y eliminar la estructura de los objetos de la base de datos. No trabaja con los datos en sí, sino con el "molde" o esquema donde se almacenarán.

Sentencias Comunes DDL:

1. CREATE: Se usa para crear nuevos objetos en la base de datos.

   Ejemplo (Crear una tabla):

   ```
   CREATE TABLE Clientes (
   ID_Cliente INT PRIMARY KEY,
   Nombre VARCHAR(100),
   Email VARCHAR(100) UNIQUE,
   FechaRegistro DATE
   );
   ```

2. ALTER: Se usa para modificar la estructura de un objeto existente.

   Ejemplo (Añadir una columna a la tabla Clientes):

ALTER TABLE Clientes

ADD COLUMN Telefono VARCHAR(20);

Ejemplo (Modificar el tipo de dato de una columna):

ALTER TABLE Clientes

ALTER COLUMN Nombre TYPE VARCHAR(150

3. DROP: Se usa para eliminar objetos de la base de datos.

Ejemplo (Eliminar la tabla Clientes):

DROP TABLE Clientes;

Ejemplo (Eliminar un índice):

DROP INDEX idx_email_clientes;

DML (Data Manipulation Language – Lenguaje de Manipulación de Datos)

Se utiliza para gestionar los datos almacenados dentro de los objetos (principalmente tablas) de la base de datos.

Sentencias Comunes DML:

1. SELECT: Se usa para consultar y recuperar datos de una o más tablas.

Ejemplo (Seleccionar todos los clientes):

*SELECT * FROM Clientes;*

Ejemplo (Seleccionar nombre y email de clientes registrados hoy):

```
SELECT Nombre, Email
FROM Clientes
WHERE FechaRegistro = CURRENT_DATE; -- CURRENT_DATE es una función
común, puede variar
```

2. INSERT: Se usa para añadir nuevas filas (registros) a una tabla.

   Ejemplo (Insertar un nuevo cliente):

```
INSERT INTO Clientes (ID_Cliente, Nombre, Email, FechaRegistro, Telefono) VALUES (101,
'Ana López', 'ana.lopez@email.com', '2025-05-03', '555-1234');
```

3. UPDATE: Se usa para modificar datos existentes en una o más filas de una tabla.

   Ejemplo (Actualizar el teléfono de un cliente específico):

```
UPDATE Clientes
SET Telefono = '555-9876'
WHERE ID_Cliente = 101;
```

4. DELETE: Se usa para eliminar filas de una tabla.

   Ejemplo (Eliminar un cliente específico):

```
DELETE FROM Clientes
WHERE ID_Cliente = 101;
```

¿Qué son las Bases de Datos Relacionales y No Relacionales?

Bases de Datos Relacionales

Modelo: Se basan en el modelo relacional, propuesto por Edgar F. Codd. Organizan los datos en tablas (llamadas relaciones) compuestas por filas (tuplas o registros) y columnas (atributos).

Estructura: Tienen un esquema fijo y predefinido. La estructura de cada tabla (columnas y tipos de datos) debe definirse antes de insertar datos.

Relaciones: Los datos en diferentes tablas se relacionan mediante claves foráneas (foreign keys) que apuntan a claves primarias (primary keys) de otras tablas.

Lenguaje: Utilizan SQL como lenguaje estándar para consultas y manipulación.

Consistencia: Generalmente garantizan las propiedades ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad), lo que las hace muy fiables para transacciones (ej. operaciones bancarias).

Escalabilidad: Tradicionalmente escalan verticalmente (aumentando la potencia del servidor: CPU, RAM, disco). Escalar horizontalmente (añadiendo más servidores) puede ser más complejo.

Ejemplos: MySQL, PostgreSQL, SQL Server, Oracle, SQLite.

Casos de uso: Aplicaciones empresariales (ERP, CRM), sistemas bancarios, aplicaciones donde la integridad y consistencia de los datos son cruciales, datos estructurados.

Bases de Datos No Relacionales (NoSQL - "Not Only SQL")

Modelo: No se basan en el modelo relacional tradicional. Engloban una variedad de modelos de datos:

Documentales: Almacenan datos en documentos flexibles (ej. JSON, BSON, XML). Ej: MongoDB, Couchbase.

Clave-Valor: Almacenan datos como pares de clave y valor simples. Ej: Redis, Memcached.

Columnares (o de Familia de Columnas): Optimizadas para leer y escribir columnas de datos en lugar de filas. Ej: Cassandra, HBase.

Orientadas a Grafos: Diseñadas para almacenar y navegar relaciones complejas entre entidades (nodos y aristas). Ej: Neo4j, ArangoDB.

Estructura: Suelen tener esquemas dinámicos o flexibles (schema-less o schema-flexible). No es necesario definir toda la estructura de antemano.

Relaciones: Las relaciones se manejan de diversas formas, a menudo mediante datos embebidos o referencias directas, o no son el foco principal.

Lenguaje: No tienen un lenguaje estándar único como SQL. Usan APIs específicas, lenguajes de consulta propios (MQL en MongoDB, CQL en Cassandra) o interfaces similares a SQL.

Consistencia: A menudo priorizan la disponibilidad y la tolerancia a particiones sobre la consistencia inmediata (modelo BASE: Basically Available, Soft state, Eventually consistent), aunque esto varía según el SGBD NoSQL.

Escalabilidad: Diseñadas para escalar horizontalmente (distribuyendo datos en múltiples servidores), lo que las hace ideales para grandes volúmenes de datos (Big Data) y alta carga de usuarios.

Ejemplos: MongoDB, Cassandra, Redis, Neo4j, Couchbase.

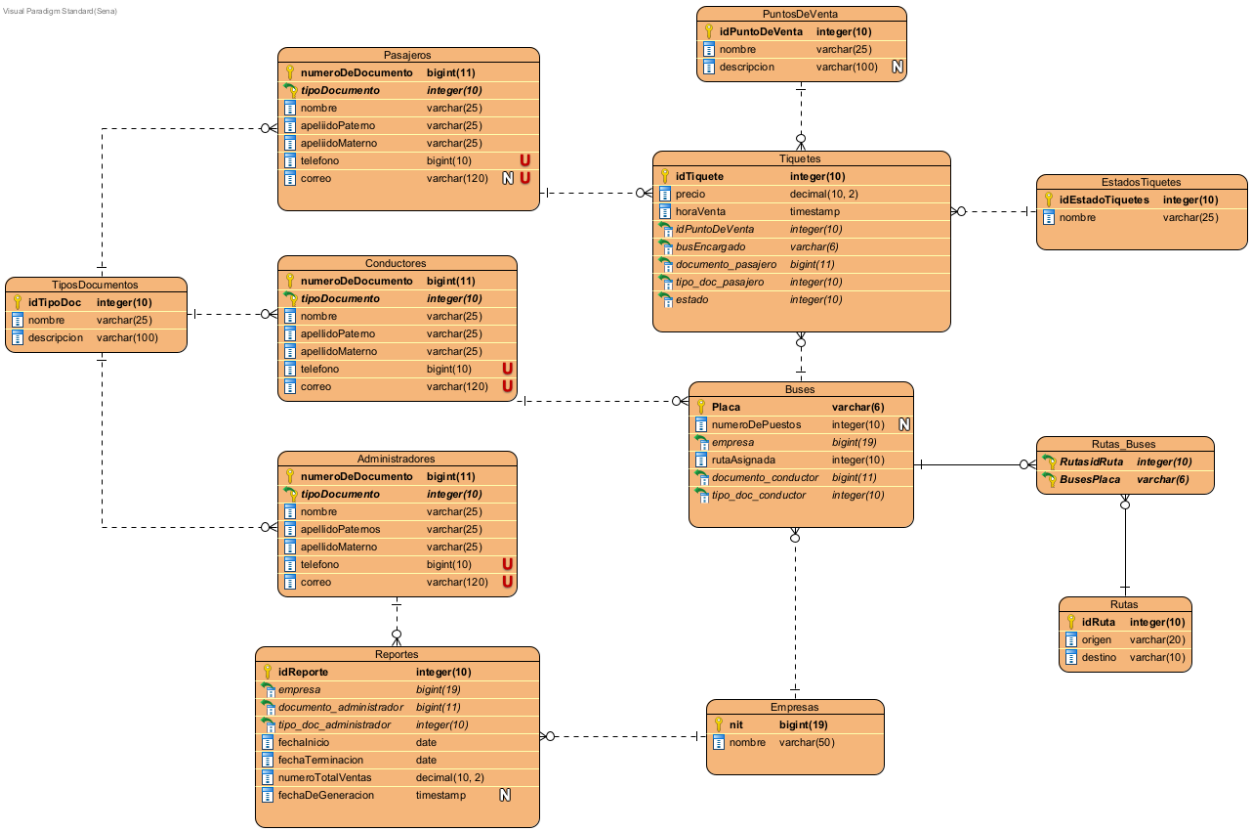
Casos de uso: Big Data, aplicaciones en tiempo real, redes sociales, IoT (Internet de las Cosas), gestión de contenidos, catálogos de productos, sistemas que requieren alta disponibilidad y escalabilidad masiva, datos semi-estructurados o no estructurados.

2. Actividades de contextualización

Palabra	conocido	Concepto propio	Conceptos en la web	fuente
Dato	x	Unidad mínima de informacion		
Llave principal	x	Primary Key, hace referencia al dato que identifica a una tabla de datos		
Llave foránea	X	Foreign Key, hace referencia al dato que funciona de referencia (puntero) a un campo en otra tabla		
SGBD			Es el software que permite crear, administrar y interactuar con las bases de datos.	https://gemini.google.com/app/59dd3fb63a2c4612?hl=es
Base de datos	X	Espacio que contiene datos de manera organizada		
SQL			Es el lenguaje estándar usado para consultar y manipular datos en bases de datos relacionales.	https://gemini.google.com/app/59dd3fb63a2c4612?hl=es
Base de dato Relacional	x	Base de datos que se caracteriza principalmente por usar tablas y manejar los datos de manera rígida por medio de relaciones		
Normalización	x	Proceso por el cual una tabla de datos optimiza su funcionamiento		
Desnormalización			Es la técnica de introducir redundancia controlada en las tablas para mejorar la	https://gemini.google.com/app/59dd3fb63a2c4612?hl=es

			velocidad de las consultas.	
NoSQL	X	Aquellas bases de datos que no son relacionales, es decir no funcionan con tablas y tampoco son estrictamente rígidas		
Workbench			Es una herramienta visual (software) para diseñar, administrar y consultar bases de datos (frecuentemente MySQL).	https://gemini.google.com/app/59dd3fb63a2c4612?hl=es
MongoDB			Es una base de datos NoSQL popular que guarda la información en documentos flexibles similares a JSON.	https://gemini.google.com/app/59dd3fb63a2c4612?hl=es
Objetos	x	Instancias nacidas de clases, por tanto, donde obtienen las características del molde padre llamado clase.		
Colleccion			En bases de datos NoSQL orientadas a documentos (como MongoDB), una colección es un agrupamiento de documentos, análogo a lo que sería una tabla en una base de datos relacional	https://gemini.google.com/app/59dd3fb63a2c4612?hl=es

3.Actividades de apropiación del conocimiento



4.Actividades De Transferencia De Conocimiento

Véase anexo1_script.sql que se encuentra junto a este documento