



UNIVERSIDAD POLITECNICA DE PACHUCA

ING. REDES Y TELECOMUNICACIONES



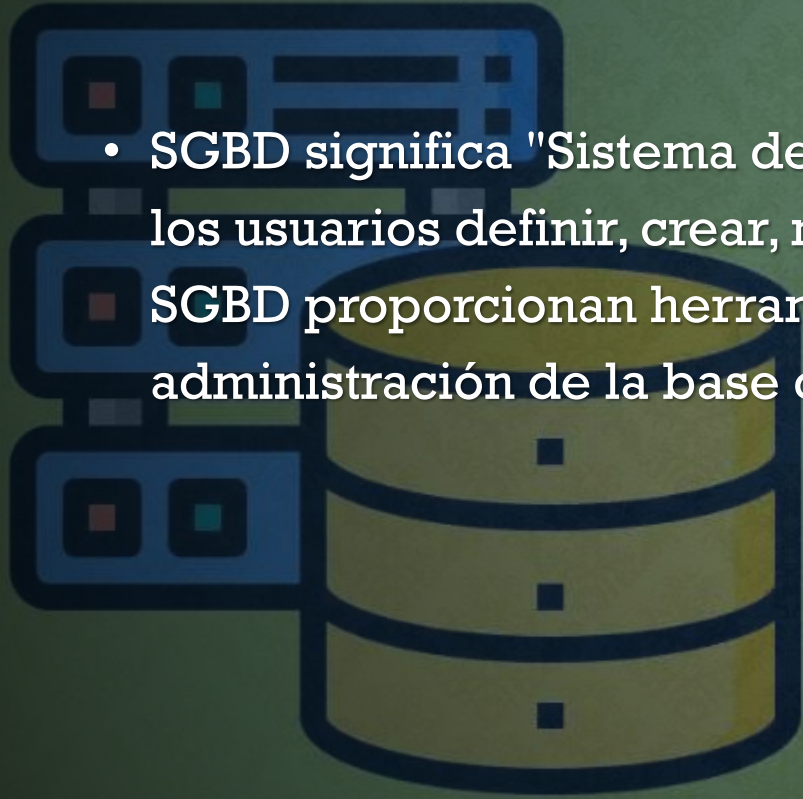
Implementación de Base de Datos en un SGBD

Tecnologías web

Silva Juárez Diego
Arenas Olivares Froylan
González Acosta Juan David
Rivas Vicuña Jesús Eduardo

¿QUE ES SGBD?

- SGBD significa "Sistema de Gestión de Bases de Datos". Es un software que permite a los usuarios definir, crear, mantener y controlar el acceso a las bases de datos. Los SGBD proporcionan herramientas y funciones para la manipulación de datos, la administración de la base de datos y la seguridad de la información almacenada.



SQL

Hablemos de

Bases de

Datos

FUNCIONES PRINCIPALES DE UN SGBD:

1. **Definición de datos:** Permite definir la estructura de la base de datos, especificando las tablas, campos, tipos de datos y relaciones entre los datos.
2. **Manipulación de datos:** Facilita la inserción, actualización, eliminación y consulta de datos a través de lenguajes de manipulación de datos como SQL.
3. **Control de acceso:** Gestiona la autenticación de usuarios y el control de permisos para garantizar la seguridad de los datos.
4. **Integridad de los datos:** Asegura que los datos sean precisos y consistentes mediante la implementación de restricciones de integridad.
5. **Recuperación ante fallos:** Proporciona mecanismos para recuperar datos en caso de fallos del sistema, como copias de seguridad y restauración.
6. **Optimización del rendimiento:** Utiliza técnicas de indexación, almacenamiento y consulta para mejorar la eficiencia en el acceso y gestión de datos.

EJEMPLOS DE SISTEMAS DE GESTIÓN DE BASE DE DATOS (SGBD):

- **1. SGBD Relacionales (RDBMS)**
 - **MySQL:**
 - **PostgreSQL:**
 - **Oracle Database:**
 - **Microsoft SQL Server:**
- **2. SGBD NoSQL**
 - **MongoDB:**
 - **Cassandra:**
 - **Redis:**
- **3. SGBD Orientados a Objetos**
 - **ObjectDB:**
- **4. SGBD Distribuidos**
 - **Apache HBase:**
 - **Amazon DynamoDB:**
- **5. SGBD en Memoria**
 - **SAP HANA:**
 - **Memcached:**

IMPLEMENTACIÓN DE BASE DE DATOS



- La implementación de una base de datos en un Sistema de Gestión de Bases de Datos (SGBD) implica varios aspectos clave, entre ellos la definición de la estructura de la base de datos, la ejecución de consultas mediante un lenguaje de consulta, y la gestión de transacciones para asegurar la integridad y consistencia de los datos.

CONSULTAS

- Son preguntas específicas que se hacen a la base de datos para recuperar información. En el contexto de los SGBD, las consultas se realizan utilizando un Lenguaje de Consulta Estructurado (SQL, por sus siglas en inglés).

Tipos de Consultas:

- **Consultas de Selección (SELECT):** Estas consultas se utilizan para recuperar datos de una o más tablas.
- **Consultas de Acción:** Incluyen comandos como INSERT, UPDATE, DELETE que permiten modificar datos en la base de datos.
- **Consultas de Definición de Datos (DDL):** Utilizadas para definir y modificar la estructura de la base de datos, como CREATE TABLE, ALTER TABLE y DROP TABLE.
- **Consultas de Control de Datos (DCL):** Incluyen comandos como GRANT y REVOKE para gestionar los permisos de acceso a los datos.

EJECUCIÓN DE CONSULTAS:



- Las consultas SQL son procesadas por el motor de base de datos del SGBD, que optimiza y ejecuta las instrucciones para acceder y manipular los datos almacenados.
- Las consultas pueden ser simples, involucrando una sola tabla, o complejas, incluyendo múltiples tablas, uniones (joins), subconsultas y agregaciones.

LENGUAJE DE CONSULTA

- El **Lenguaje de Consulta** más utilizado en los SGBD es SQL. SQL permite a los usuarios definir, manipular y controlar los datos en una base de datos relacional.

COMPONENTES PRINCIPALES DEL SQL:



Lenguaje de Definición de Datos (DDL):
Incluye comandos para definir y modificar la estructura de las bases de datos. **CREATE:** Para crear nuevas bases de datos y tablas.



ALTER: Para modificar la estructura de tablas existentes.



DROP: Para eliminar bases de datos y tablas.



Lenguaje de Manipulación de Datos (DML): Incluye comandos para la manipulación de datos dentro de las tablas. **INSERT:** Para insertar nuevos registros en las tablas.



UPDATE: Para actualizar datos existentes.



DELETE: Para eliminar registros.

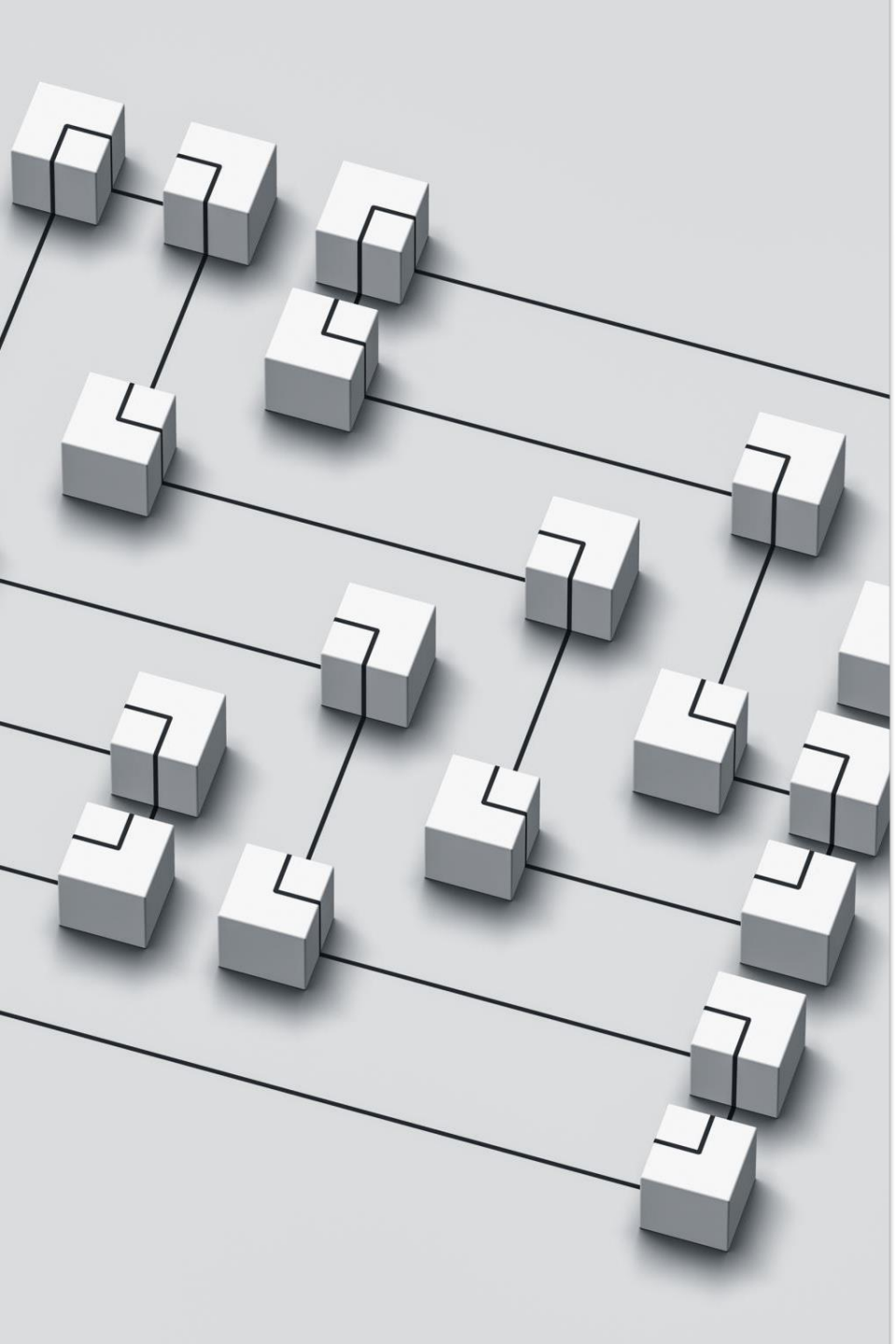


- **Lenguaje de Consulta de Datos (DQL):** Principalmente el comando SELECT para consultar datos.
- **Lenguaje de Control de Datos (DCL):** Utilizado para controlar el acceso a los datos en la base de datos.
- **GRANT:** Para otorgar permisos.
- **REVOKE:** Para revocar permisos.



FUNCIONES DE CONSULTAS

- **SELECT:** Recuperar datos de la base de datos.
- **INSERT:** Insertar nuevos datos en la base de datos.
- **UPDATE:** Modificar datos existentes en la base de datos.
- **DELETE:** Eliminar datos de la base de datos.
- **CREATE TABLE:** Crear una nueva tabla en la base de datos.
- **ALTER TABLE:** Modificar la estructura de una tabla existente.
- **DROP TABLE:** Eliminar una tabla de la base de datos.
- **GRANT:** Otorgar permisos de acceso a los datos.
- **REVOKE:** Revocar permisos de acceso a los datos.



FUNCIONES DEL LENGUAJE DE CONSULTA (SQL)

- **CREATE DATABASE:** Crear una nueva base de datos.
- **CREATE TABLE:** Definir una nueva tabla y sus columnas.
- **ALTER TABLE:** Agregar, eliminar o modificar columnas en una tabla existente.
- **DROP DATABASE:** Eliminar una base de datos completa.
- **INSERT INTO:** Agregar nuevos registros a una tabla.
- **UPDATE SET:** Actualizar datos específicos en una tabla.
- **DELETE FROM:** Eliminar registros de una tabla.
- **SELECT:** Consultar y recuperar datos de una o más tablas.
- **JOIN:** Combinar filas de dos o más tablas basadas en una columna relacionada.
- **WHERE:** Filtrar registros basados en una condición específica.
- **GROUP BY:** Agrupar registros basados en una o más columnas.
- **ORDER BY:** Ordenar los resultados de una consulta en un orden específico.
- **HAVING:** Filtrar registros después de agruparlos.

TRANSACCIONES

Las **transacciones** son una secuencia de operaciones realizadas como una única unidad lógica de trabajo. Una transacción debe cumplir las propiedades ACID para asegurar la integridad de los datos:

- **Atomicidad (Atomicity):** Todas las operaciones dentro de una transacción se completan exitosamente o ninguna se completa. Esto garantiza que la transacción se ejecute de manera atómica.
- **Consistencia (Consistency):** La transacción lleva la base de datos de un estado consistente a otro estado consistente.
- **Aislamiento (Isolation):** Las operaciones de una transacción son invisibles para otras transacciones hasta que la transacción es completada.
- **Durabilidad (Durability):** Una vez que una transacción ha sido confirmada, los cambios persisten en la base de datos incluso en caso de fallas del sistema.

GESTIÓN DE TRANSACCIONES



BEGIN TRANSACTION: Marca el inicio de una transacción.



COMMIT: Confirma una transacción, haciendo permanentes todos los cambios realizados.



ROLLBACK: Deshace una transacción, revirtiendo todos los cambios realizados durante la transacción.

FUNCIONES DE TRANSACCIONES



BEGIN TRANSACTION: Iniciar una nueva transacción.



COMMIT: Confirmar la transacción y hacer permanentes los cambios.



ROLLBACK: Revertir la transacción y deshacer los cambios.



SAVEPOINT: Crear un punto de restauración dentro de una transacción.



RELEASE SAVEPOINT: Eliminar un punto de restauración marcado.



ROLLBACK TO SAVEPOINT: Revertir la transacción hasta el punto de restauración especificado.

PASOS

Planificación de la Transacción

- **Definición de Objetivos:** Determina qué cambios deseas realizar en la base de datos y cómo estas modificaciones afectan a los datos.
- **Análisis de Impacto:** Evalúa el impacto de la transacción en otras partes de la base de datos y en otras transacciones concurrentes.

Preparación

- **Establecer el Entorno:** Asegúrate de que el sistema de base de datos esté en un estado adecuado para realizar transacciones (por ejemplo, no en modo de solo lectura).
- **Configurar el Manejo de Errores:** Prepara procedimientos para manejar errores y excepciones, como la implementación de BEGIN TRY...END TRY en SQL Server.

Inicio de la Transacción

- **Comando de Inicio:** Usa BEGIN TRANSACTION, START TRANSACTION, o el comando equivalente en tu SGBD.
- **Transparencia:** Asegúrate de que todos los cambios estén encapsulados dentro del contexto de la transacción.

Ejecutar Operaciones

- **Instrucciones SQL:** Ejecuta las operaciones necesarias, como INSERT, UPDATE, DELETE.
- **Verificación Intermedia:** Opcionalmente, verifica los cambios intermedios, especialmente en transacciones complejas, para asegurar que los resultados sean los esperados.

Confirmación de la Transacción

- **Comando de Confirmación:** Usa COMMIT para guardar los cambios permanentemente.
- **Confirmación Automática:** Algunos sistemas permiten la confirmación automática al finalizar la transacción si no se especifica un comando ROLLBACK.

Manejo de Errores

- **Comando de Reversión:** Si ocurre un error, usa ROLLBACK para revertir todos los cambios realizados durante la transacción.
- **Registro de Errores:** Implementa mecanismos para registrar y gestionar errores, como el uso de ERROR_LOG o TRY...CATCH.

Cierre y Limpieza

- **Liberar Recursos:** Asegúrate de liberar cualquier recurso asociado con la transacción, como bloqueos de registros.
- **Cerrar Conexiones:** Si la transacción finaliza, cierra las conexiones abiertas a la base de datos si es necesario.

Pruebas y Validación

- **Pruebas Unitarias:** Realiza pruebas para asegurar que la transacción funcione correctamente en diferentes escenarios.

Validación de Integridad: Verifica que la base de datos mantenga la integridad y consistencia después de la transacción


```
1 BEGIN TRANSACTION;
2
3 BEGIN TRY
4     -- Operaciones de base de datos
5     INSERT INTO empleados (nombre, salario) VALUES ('Juan Pérez', 3000);
6     UPDATE departamentos SET presupuesto = presupuesto - 3000 WHERE id = 1;
7
8     -- Confirmar los cambios
9     COMMIT;
10 END TRY
11 BEGIN CATCH
12     -- Revertir los cambios en caso de error
13     ROLLBACK;
14     -- Manejo del error
15     PRINT 'Error en la transacción: ' + ERROR_MESSAGE();
16 END CATCH;
17
18 START TRANSACTION;
19
20 -- Operaciones de base de datos
21 INSERT INTO empleados (nombre, salario) VALUES ('Juan Pérez', 3000);
22 UPDATE departamentos SET presupuesto = presupuesto - 3000 WHERE id = 1;
23
24 -- Confirmar los cambios
25 COMMIT;
26
27 -- Manejo de errores (MySQL no tiene bloques TRY...CATCH nativos, puedes manejar errores con aplicaciones externas)
28
```


CONCLUSIÓN

- La implementación de una base de datos en un SGBD es esencial para gestionar eficientemente la información. Involucra el uso de consultas (SELECT, INSERT, UPDATE, DELETE) para interactuar con los datos, y el lenguaje de consulta estructurado SQL para definir y manipular la estructura y contenido de la base de datos. Además, la gestión de transacciones asegura la integridad y consistencia de los datos mediante las propiedades ACID, garantizando que todas las operaciones dentro de una transacción se completen correctamente o se reviertan en caso de error. Estos componentes combinados permiten una administración robusta y confiable de los datos, fundamental para cualquier sistema de información.