Base de Datos I

I. CONCEPTOS SOBRE BASE DE DATOS

1.1. Conceptos básicos - Definiciones

Perspectiva Relacional.

"Una base de datos relacional es una colección de datos organizados en tablas que están interrelacionadas mediante claves primarias y foráneas, permitiendo la gestión y recuperación eficiente de la información a través del uso de operaciones basadas en la lógica de predicados.“

Codd, E. F. (1970). "A Relational Model of Data for Large Shared Data Banks

"Una base de datos relacional es una base de datos que se organiza en términos de relaciones formales, donde una relación se representa como una tabla compuesta por filas (tuplas) y columnas (atributos). Las operaciones sobre estas tablas se basan en la teoría de conjuntos y el álgebra relacional.".

Date, C. J. (2003). "An Introduction to Database Systems.

"Las bases de datos relacionales son sistemas de gestión de bases de datos que utilizan el modelo relacional, el cual organiza los datos en tablas relacionadas entre sí mediante claves primarias y foráneas. Estas tablas permiten realizar consultas complejas y manipulación de datos utilizando el lenguaje SQL.

Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). "Database Systems: The Complete Book.

1.1. Conceptos básicos - Introducción

Perspectiva Relacional.

"Una base de datos relacional es una colección de datos organizados en tablas que están interrelacionadas mediante claves primarias y foráneas, permitiendo la gestión y recuperación eficiente de la información a través del uso de operaciones basadas en la lógica de predicados.“

Codd, E. F. (1970). "A Relational Model of Data for Large Shared Data Banks

"Una base de datos relacional es una base de datos que se organiza en términos de relaciones formales, donde una relación se representa como una tabla compuesta por filas (tuplas) y columnas (atributos). Las operaciones sobre estas tablas se basan en la teoría de conjuntos y el álgebra relacional.".

Date, C. J. (2003). "An Introduction to Database Systems.

"Las bases de datos relacionales son sistemas de gestión de bases de datos que utilizan el modelo relacional, el cual organiza los datos en tablas relacionadas entre sí mediante claves primarias y foráneas. Estas tablas permiten realizar consultas complejas y manipulación de datos utilizando el lenguaje SQL.

Garcia-Molina, H., Ullman, J. D., & Widom, J. (2008). "Database Systems: The Complete Book.

1.1. Conceptos básicos - Introducción

Perspectiva Objeto.

"Una base de datos orientada a objetos es una colección de objetos persistentes cuyos estados (datos y variables) se gestionan y mantienen utilizando las capacidades de un sistema de gestión de bases de datos orientado a objetos.“

William J. Premerlani y Michael R. Blaha, *"An Object-Oriented Modeling Methodology for Database Applications* (1994)

Las bases de datos orientadas a objetos son sistemas de gestión de bases de datos que representan información en forma de objetos, como se hace en la programación orientada a objetos.

Rick Cattell, *"Object Data Management Object-Oriented and Extended Relational Database Systems* (1991)

Un sistema de gestión de bases de datos orientado a objetos (OODBMS) es un sistema que extiende el concepto de bases de datos al dominio de los objetos, manteniendo la integridad y coherencia de los datos a través de las técnicas y métodos de la programación orientada a objetos.

Ramez Elmasri y Shamkant B. Navathe, *"Fundamentals of Database Systems* (2010)

1.1. Conceptos básicos - Objetivos

Confiabilidad:
Asegurar que los datos sean accesibles y precisos en todo momento, incluso en caso de fallos. Las BD relacionales implementan mecanismos de respaldo, recuperación y transacciones ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad) para garantizar que los datos se mantengan íntegros y disponibles, incluso en situaciones de fallos del sistema o errores humanos.

Integridad de los Datos:
Garantizar que los datos almacenados sean precisos, coherentes y estén completos. Utilizando restricciones y reglas de integridad (como claves primarias, claves foráneas, y restricciones de unicidad), las BD relacionales aseguran que las relaciones entre los datos sean correctas y que no haya datos redundantes o inconsistentes, manteniendo la fiabilidad de la información.

Disponibilidad: Garantizar que los datos y los servicios estén disponibles para los usuarios y aplicaciones en todo momento. Mediante técnicas de replicación, balanceo de carga y estrategias de alta disponibilidad, las BD relacionales aseguran que los datos sean accesibles de manera continua, minimizando el tiempo de inactividad y asegurando el acceso ininterrumpido a la información.

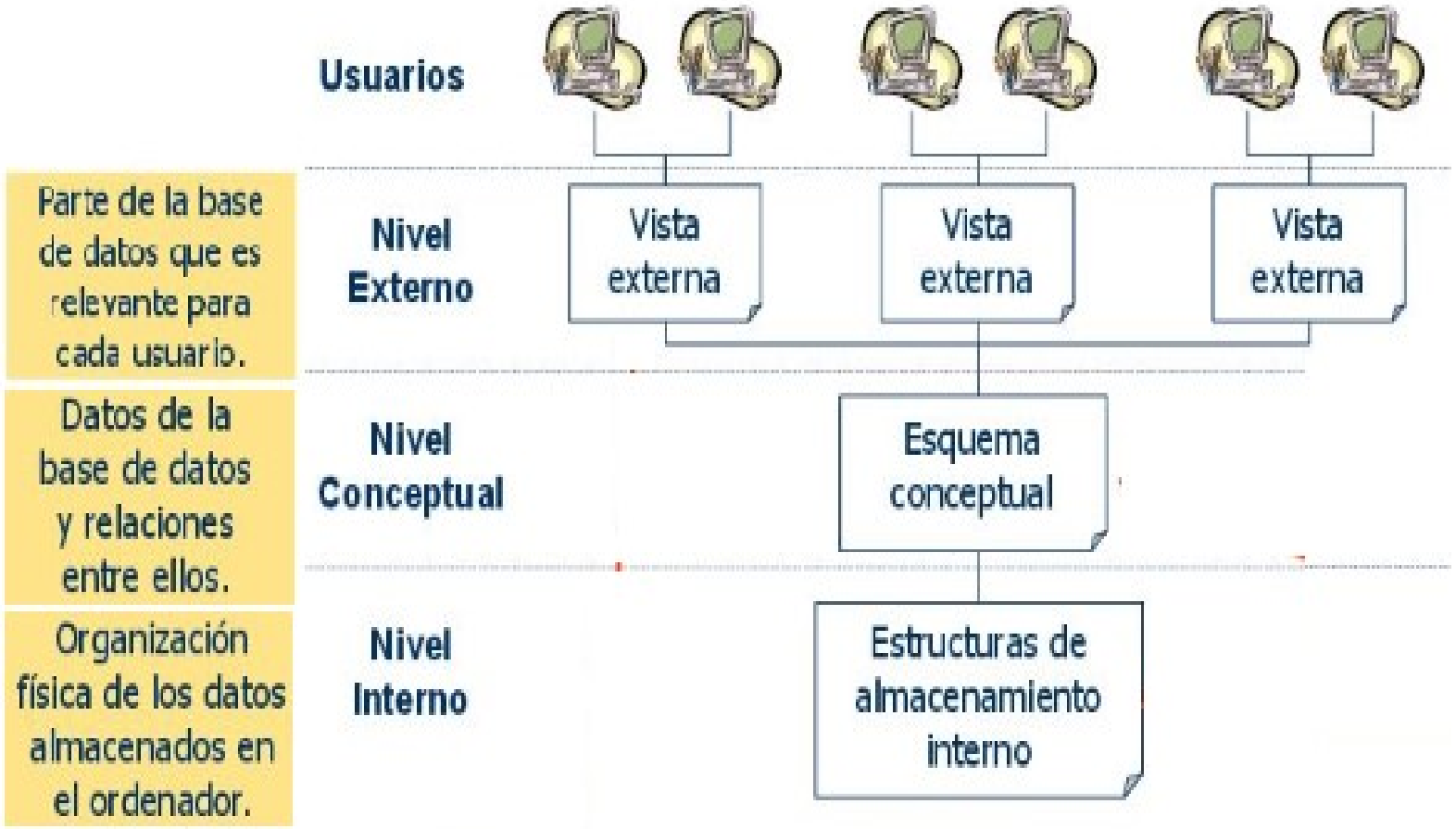
1.1. Conceptos básicos - Objetivos

Seguridad: Proteger los datos contra accesos no autorizados y garantizar la confidencialidad, integridad y disponibilidad de la información, implementan mecanismos de control de acceso, autenticación, autorización y cifrado de datos para proteger la información sensible, políticas de seguridad, auditorías y prevenir accesos no autorizados puedan realizar operaciones específicas en la base de datos.

Flexibilidad y Escalabilidad: Proveer un modelo flexible y escalable para gestionar datos que pueda adaptarse a las necesidades cambiantes. El modelo relacional permite a los diseñadores de BD crear esquemas que pueden evolucionar con el tiempo sin necesidad de rehacer la base de datos desde cero, permitiendo agregar nuevas tablas, modificar esquemas existentes y gestionar grandes volúmenes de datos de manera eficiente.

Recuperación y Manipulación de Datos: Diseñadas para realizar consultas complejas y manipulación de datos de manera optimizadas y eficiente, BD relacionales utilizan índices, planes de consulta optimizados y técnicas de gestión de almacenamiento para mejorar el rendimiento de las operaciones de lectura y escritura.

1.1. Conceptos básicos - Abstracción



1.1. Conceptos básicos - Abstracción

1. Nivel Interno (Físico):

Es el nivel más bajo de abstracción, en él se especifica la forma en que se va a almacenar los datos en los dispositivos de almacenamiento.

Es aquí donde el almacenamiento de datos se maneja de manera eficiente, optimizando recursos y garantizando la rapidez en las operaciones de bases de datos.

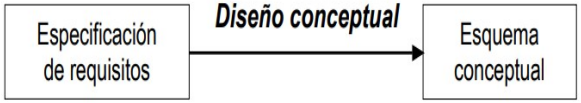
Se detalla el cómo y el dónde se almacenan los datos, la independencia de los datos y los métodos de acceso, su confiabilidad, disponibilidad, integridad y seguridad de la información que son críticos para el rendimiento general del sistema.

1.1. Conceptos básicos - Abstracción

2. Nivel Conceptual (Núcleo lógico):

Es el nivel medio de abstracción, en él vemos qué datos son almacenados en la base de datos y las relaciones que existen entre estos y su estructura.

Existen tres características importantes inherentes son la integración, la consistencia y la independencia de los datos, en este nivel trabajan los administradores de bases de datos, y son los que deben decidir qué información se va a guardar en la base de datos.



Esquema conceptual → Descripción de alto nivel del contenido de información de la base de datos, independiente del SGBD que se vaya a utilizar.

Modelo conceptual → Lenguaje que se utiliza para describir esquemas conceptuales.

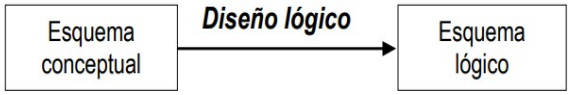
Propósito → Obtener un esquema completo que lo exprese todo.

1.1. Conceptos básicos - Abstracción

3. Nivel lógico (Perspectiva usuario):

El primer nivel, o esquema externo o vistas, está enfocado en cómo los usuarios finales visualizarán la información, describiendo cada parte de la Base de Datos que nos interesa, cada esquema externo a un grupo de usuarios y múltiples vistas de información.

En este pueden operar los administradores de Bases de Batos (BD), desarrolladores o el personal no técnico, cada uno con sus necesidades y permisos



Esquema lógico → Descripción de la estructura de la base de datos según el modelo del SGBD que se vaya a utilizar.

Modelo lógico → Lenguaje que se utiliza para describir esquemas lógicos; hay varios modelos lógicos: de red, relacional, orientado a objetos, ...

Propósito → Obtener una representación que use de la manera más eficiente los recursos disponibles en el modelo lógico para estructurar datos y modelar restricciones.

El diseño lógico depende del **modelo de BD** que soporta el SGBD.

1.1. Conceptos básicos - Los SI de Gestión

Hay dos tipos de sistemas de información de gestión de datos:

- Sistemas de información orientados al proceso o sistema clásico de ficheros: En estos hay distintas aplicaciones para gestionar diferentes aspectos del sistema de información. Este Sistema tiene muchos inconvenientes, y como única ventaja destacaría que al ser los procesos independientes la modificación de uno no afecta al resto.

- Sistemas de información orientados al dato o Sistema de Bases de datos: Tiene grandes ventajas frente al anterior, entre ellas la independencia de los datos, menor duplicidad de los datos, menor espacio de almacenamiento, entre otras.

  Pero también encontramos inconvenientes en este sistema, ya que su instalación es costosa, se requiere un personal calificado, su implantación es larga y difícil y se tiene una ausencia de estándares reales

1.1. Conceptos básicos - Enfoques

Tradicional

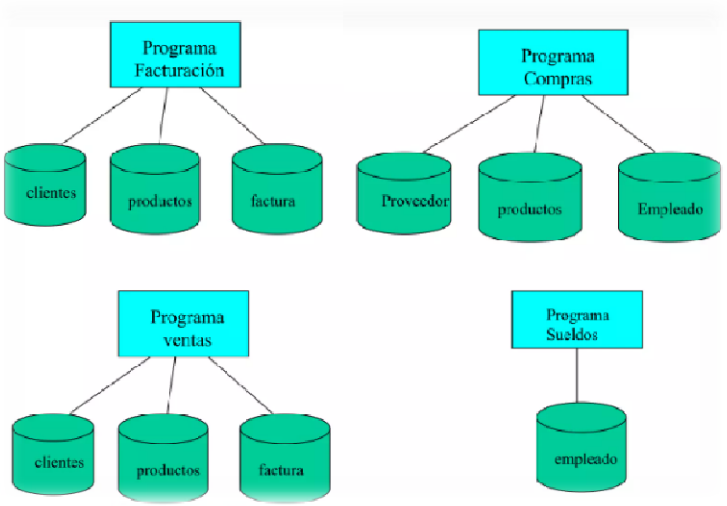
- El enfoque tradicional utilizado en el desarrollo de Sistemas de Información para el tratamiento de los datos, se relaciona con el procesamiento de datos por unidad organizacional
- Cada sistema desarrollado es diseñado para satisfacer las necesidades de un departamento o grupo de usuarios, no existiendo una planificación corporativa o un modelo que guíe el desarrollo de aplicaciones
- Este enfoque es conocido como enfoque por agregación
- El nombre por agregación, representa a un proceso evolutivo que se presenta al ir acoplando a un sistema de información nuevas funciones (nuevos requerimientos que no habían sido considerado en el momento del diseño inicial del sistema)

1.1. Conceptos básicos - Enfoques

Tradicional

Por ejemplo, el Sistema de Procesamiento de Archivos

- Cada nueva aplicación es diseñada con su propio conjunto de archivos de datos
- Muchos de esos datos pueden ya existir en archivos de otras aplicaciones, pero para ser usados en la nueva aplicación requerirían de reestructuración
- Es más simple diseñar nuevos archivos para cada aplicación



1.1. Conceptos básicos

Tradicional

Desventajas

- Redundancia no controlada: Pérdida de espacio, tiempo perdido en actualizaciones.
- Escasa posibilidad de compartir datos: Con nuevas aplicaciones que requieren nuevos archivos, causando un aumento de datos duplicados.
- Inconsistencia de datos: Errores en las aplicaciones ,reportes inconsistentes, pérdida de la confianza en la integridad del sistema de información)
- Inflexibilidad: Sumamente rígido y es lenta la la evolución del sistema de información, frustración de los usuarios

  Pobre estandarización (inconsistencia por sinónimos y por homónimos, dificultad en las mantenciones de la aplicación.
- Baja productividad del programador (mayor costo del software)
- Excesiva mantención (cerca del 80% programación es ocupado en esta tarea)

1.1. Conceptos básicos - Enfoques

Base de Datos

- Al final de los años sesenta nacen las bases de datos, para almacenar los datos utilizados por los usuarios, empresas, etc. A partir de allí surge el ENFOQUE DE BASE DE DATOS.
- Una base da datos (BD) en un conjunto de datos que permiten satisfacer las necesidades de información de una organización.
- Desde una perspectiva organizacional, se puede definir como un conjunto de datos operacionales relevantes para la toma de decisiones permitiendo satisfacer diversos requerimientos de información.
- Los datos son visualizados como un recurso que debe ser compartido entre diferentes usuarios
- Cada usuario puede contar con una visión propia de la base de datos, de acuerdo a sus requerimientos de información
- Los datos son almacenados de tal independientes del programa que los acede y los usa

1.1. Conceptos básicos - Enfoques

Base de Datos

- Se tiene un control centralizado de las operaciones de protección, ingreso, modificación, eliminación y recuperación de datos, a través de un software específico(Sistema de Administración de Base de Datos, DBMS)
- Función manipulación de datos (permite almacenar, modificar y recuperar los datos de la Base).
- Función seguridad de datos (permite controlar el acceso y la concurrencia de los usuarios y provee mecanismos de respaldo y recuperación de la base de datos)
- Herramientas CASE (apoyan el desarrollo de software, especialmente en lo que respecta al diseño de la base de datos y sus programas de aplicación)

1.1. Conceptos básicos

Base de Datos

Ventajas

- Esfuerzo por estandarización (se establece la función de administración de datos)
- Facilita el desarrollo de aplicaciones (el programador no necesita cargar con las tareas de diseño, construcción y mantención de archivos maestros)
- Controles de seguridad, privacidad e integridad (se establece la función de administración de datos)
- Flexibilidad en el acceso (múltiples trayectorias de recuperación de cada dato, satisfacción de requerimientos ad-hoc mediante SQL)
- Independencia de los datos (separación de las descripciones de datos de los programas de aplicaciones que usan los datos)
- Reducción del mantenimiento de programas (los datos son independientes de los programas y se reduce la necesidad de modificar los programas)

1.1. Conceptos básicos

?Que es Modelo de Base de Datos?

Un modelo de datos consiste en simbolizar la información del mundo real para que sea más fácil su utilización. Este modelo debe describir la estructura de los datos, como las relaciones existentes entre estos, o sus restricciones y su significado.

Un modelo de datos intenta plasmar información del mundo real, que será almacenada y utilizada por un Sistema Gestor de Bases de Datos, conocido por sus siglas SGBD.

Una vez hecho este análisis de los datos, debemos realizar un esquema, para esto tenemos que llevarlo al modelo conceptual.

Este esquema no debe tener ninguna correlación con la implementación del esquema final de visión o lógico, ni debe incluir elemento relacionado con la futura base de datos

1.2. Evolución de las Bases de Datos

Origen de Bases de Datos

Origen Bases de Datos: Modelos Jerárquicos y de Red (60-70)

- Primeros modelos de bases de datos.
- IBM's IMS (Information Management System).
- Conceptos de registros y conjuntos.

El Surgimiento del Modelo Relacional: Modelo Relacional, E.F. Codd (70-80)

- Introducción del modelo relacional.
- Organización de datos en tablas.
- SQL (Structured Query Language) se vuelve estándar.

Bases de Datos Orientadas a Objetos: programación Objetos (80-90)

- Integración de conceptos de programación orientada a objetos.
- Manejo de datos complejos y estructuras jerárquicas.
- Ejemplos: ObjectDB, db4o.

1.2. Evolución de las Bases de Datos

Actualidad en Bases de Datos

Big Data y NoSQL (2000 -2010)

- Creciente volumen, velocidad y variedad de datos.
- Tecnologías como Hadoop y Spark.
- Mayor flexibilidad y escalabilidad.
- Modelos clave-valor, columnares, documentales y de grafos.
- Ejemplos: MongoDB, Cassandra, Redis.

Bases de Datos y Computación en la Nube , Azure (2010-2020)

- Servicios de bases de datos como Amazon RDS, Google Cloud SQL.
- Escalabilidad y disponibilidad mejoradas.
- Modelos de pago por uso.

Años 2020: Bases de Datos Autónomas y Multimodelo

- Autogestión, autoseguridad y autorreparación.
- Ejemplo: Oracle Autonomous Database.
- Soporte para múltiples modelos de datos en una sola base de datos.
- Ejemplos: ArangoDB, Microsoft Azure Cosmos DB.

1.2. Evolución de las Bases de Datos

Tendencias en Bases de Datos

Inteligencia Artificial y Machine Learning en Bases de Datos:

- Optimización de consultas y gestión de datos.
- Predicciones y análisis en tiempo real.
- Bases de datos con capacidades de aprendizaje automático integradas.

Blockchain y Bases de Datos Descentralizadas

- Bases de datos distribuidas y seguras.
- Aplicaciones en finanzas, contratos inteligentes y cadenas de suministro.
- Hyperledger, Ethereum.

Bases de Datos y Procesamiento en Tiempo Real

- Análisis y respuesta instantánea a eventos.
- Aplicaciones en comercio electrónico, finanzas y telecomunicaciones.
- Ejemplos: Apache Kafka, Redis Streams.

Bases de Datos y Computación Quantum

- Potencial de revolución en el procesamiento de datos.
- Investigación en algoritmos cuánticos para bases de datos.
- Proyectos de investigación en Google y IBM.

1.3 Evolución de las Bases de Datos

Independencia Física de Datos

La independencia física de datos se refiere a la capacidad de cambiar la forma en que los datos son almacenados físicamente en el hardware sin afectar la forma en que los datos son accedidos y manipulados por los sistemas de software.

- **Aislamiento de cambios físicos:** Los cambios en la estructura física no afectan el esquema lógico.
- **Flexibilidad:** Permite la optimización del rendimiento sin interrumpir las aplicaciones que usan la base de datos.
- **Mantenimiento:** Facilita las tareas de administración y mantenimiento de la base de datos.

1.3. Evolución de las Bases de Datos

Independencia Física de Datos

Por ejemplo:

Una empresa decide migrar su Base de Datos de un servidor local a una nube privada para mejorar el rendimiento y la escalabilidad.

Sin Independencia Física:

- Las aplicaciones deben modificarse para adaptarse a la nueva ubicación y estructura de almacenamiento de datos.
- Posible interrupción de servicios y tiempo de inactividad.

Con Independencia Física:

- La base de datos se migra sin necesidad de cambiar las aplicaciones.
- Las aplicaciones siguen funcionando sin modificaciones, accediendo a los datos de la misma forma.

1.3. Evolución de las Bases de Datos

Independencia Física de Datos

Beneficios

- **Eficiencia Operacional:** Simplifica la administración de la base de datos.
- **Reducción de Costos:** Menos necesidad de modificar aplicaciones.
- **Mejora de Desempeño:** Permite optimizaciones en el almacenamiento físico.
- **Escalabilidad:** Facilita la adaptación a nuevas tecnologías de almacenamiento.

La independencia física de datos permitió a la empresa mejorar su infraestructura de almacenamiento sin necesidad de modificar las aplicaciones existentes. Esto demostró cómo los cambios en la capa física de la base de datos no afectan la capa lógica, asegurando una transición suave y eficiente.

1.3 Evolución de las Bases de Datos

Independencia Lógica de Datos

La independencia lógica de datos se refiere a la capacidad de modificar el esquema lógico de una base de datos sin tener que alterar las aplicaciones que interactúan con ella. Esto incluye cambios en tablas, vistas y relaciones sin afectar la capa de aplicación.

- **Aislamiento de cambios lógicos:** Los cambios en el esquema lógico no afectan el esquema externo.
- **Flexibilidad:** Permite la evolución del modelo de datos sin interrupción del servicio.
- **Mantenimiento:** Facilita la actualización y mejora del modelo de datos.

1.3 Evolución de las Bases de Datos

Independencia Lógica de Datos

Por ejemplo:

Una empresa de comercio electrónico decide agregar nuevas funcionalidades a su sistema de base de datos, incluyendo el almacenamiento de reseñas de productos por parte de los clientes.

Sin Independencia Lógica:

- Las aplicaciones deben actualizarse para adaptarse al nuevo esquema.
- Posible interrupción de servicios y tiempo de inactividad.

Con Independencia Lógica:

- **Identificación de la Necesidad:** Se requiere almacenar reseñas de productos.
- **Modificación del Esquema Lógico:** Se agrega una nueva tabla "Reseñas" relacionada con la tabla "Productos".
- **Uso de Vistas:** Se crean vistas para mantener la compatibilidad con las aplicaciones existentes.

1.3. Lenguaje de Manipulación de Datos

El lenguaje DML

- Un lenguaje de manipulación de datos (Data Manipulation Language, o DML en inglés) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.
- El lenguaje de manipulación de datos más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional. Otros ejemplos de DML son los usados por bases de datos IMS/DL1, CODASYL u otras.
- Consta del os siguientes elementos: Select, Insert, Delete y Update.
- Se clasifican en dos grandes grupos: lenguajes de consulta procedimentales y lenguajes de consulta no procedimentales

1.3. Lenguaje de Manipulación de Datos

Clasificación

Lenguajes de consulta procedimentales.

- En este tipo de consulta de lenguaje el usuario da instrucciones al sistema para que realice una serie de procedimientos u operaciones en la base de datos para calcular un resultado final.

Ejemplo:

```
DECLARE
  CURSOR emp_cursor IS
    SELECT employee_id, first_name, last_name, salary
    FROM employees
    WHERE salary > 50000;
BEGIN
  FOR emp_record IN emp_cursor LOOP
    DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_record.employee_id || ', Name: ' || emp_record.first_name ||
' ' || emp_record.last_name || ', Salary: ' || emp_record.salary);
  END LOOP;
END;
```

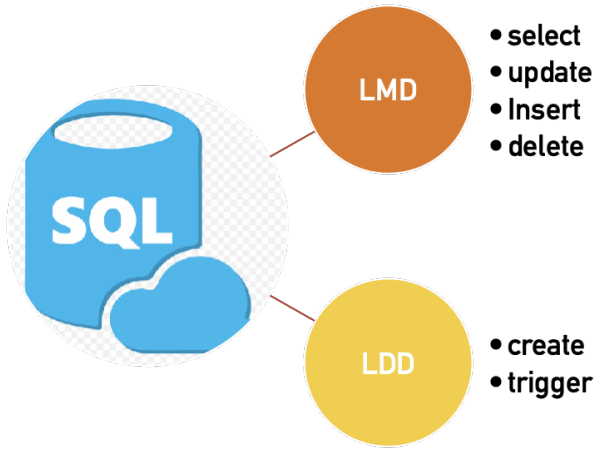
1.3. Lenguaje de Manipulación de Datos

Clasificación

Lenguajes de consulta no procedimentales

- El usuario describe la información deseada sin un procedimiento específico para obtener esa información. Es la base del lenguaje de consulta SQL.
- Ejemplo:

```
SELECT employee_id, first_name, last_name, salary
FROM employees
WHERE salary > 50000;
```



1.3. Lenguaje de Manipulación de Datos

El lenguaje LDD

- El lenguaje LDD en SQL se refiere a Lenguaje de Definición de Datos (conocido como DDL, Data Definition Language en inglés).
- Este lenguaje se utiliza para definir y gestionar las estructuras de las bases de datos, como las tablas, índices, esquemas, vistas y restricciones, el lenguaje más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional. Otros ejemplos de DML son los usados por bases de datos IMS/DL1, CODASYL u otras.
- Create: Crea nuevos objetos en la base de datos, como tablas, índices, vistas, esquemas, etc.
- Alter: Modifica la estructura de un objeto existente en la base de datos, como una tabla o un índice.
- Drop: Elimina un objeto de la base de datos, como una tabla, vista, índice, etc
- Truncate: Elimina todos los registros de una tabla, pero mantiene la estructura de la tabla para su uso futuro.

1.3. Lenguaje de Manipulación de Datos

El lenguaje LDD

- Rename: Cambia el nombre de un objeto de la base de datos.

Características del LDD (DDL)

- Definición de Estructura: Permite definir y modificar la estructura de los objetos de la base de datos.
- Ejecución Inmediata: Los comandos DDL son ejecutados inmediatamente y no pueden ser revertidos (a menos que se esté utilizando un sistema de transacciones con soporte para DDL).
- No Manejo de Datos: A diferencia del DML (Data Manipulation Language), el DDL no se utiliza para manipular datos dentro de las tablas, sino para definir y modificar la estructura de las tablas y otros objetos de la base de datos.
- El LDD (DDL) es esencial para la creación y mantenimiento de la estructura de una base de datos, permitiendo a los administradores y desarrolladores definir cómo se almacenan y organizan los datos.