

TEXT CLASS REVIEW

TEMAS A TRATAR EN EL CUE:

- Definición de base de datos.
- Modelos de bases de datos.
- Bases de Datos Relacionales.
- Componentes de una base de datos relacional.
- RDBMS (Relational Database Management System)
- SQL (Structured Query Language).

DEFINICIÓN DE BASE DE DATOS

Desde un punto de vista general, una base de datos es todo aquel repositorio o almacén de datos que pertenecen a un mismo contexto, y desde donde se puede extraer información.

Desde el punto de vista informático, una base de datos se define como un sistema capaz de administrar y procesar datos, y que cuenta con una arquitectura de cliente - servidor.



Ilustración 1. Arquitectura básica de una base de datos.

- **Cliente:** todo aquel componente externo al SGBD (Sistema Gestor de Base de Datos) que realiza cualquier acción sobre los datos.

- **SGBD:** componentes de software que permite realizar cualquier acción sobre los datos, pero no de forma directa, sino a través de una capa de acceso.
- **Datos:** datos físicos almacenados en la BD.
- **Esquema:** estructura de una BD.

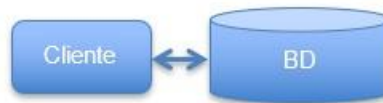


Ilustración 2. Arquitectura básica simplificada de una base de datos.

La ilustración 2 es la representación simplificada, y más conocida, de la arquitectura de una BD, donde la BD encapsula al SGBD y Datos de la ilustración 1.

MODELOS DE BASES DE DATOS

Existen varias, siendo las más conocidas:

- **Relacionales:** son una colección de elementos de datos, organizados en un conjunto de tablas formalmente descritas, desde donde se puede acceder a los datos o volver a montarlos de muchas maneras diferentes, sin tener que reorganizar las tablas de la base. La interfaz estándar de programa de usuario, y aplicación a una base de datos relacional, es el Lenguaje de Consultas Estructuradas (SQL). Algunos ejemplos, son: PostgreSQL, MySQL, Oracle y SQL Server.
- **No Relacionales:** también conocidas como NoSQL, pensadas para modelos de datos específicos, con esquemas flexibles, y ampliamente reconocidas porque son fáciles de desarrollar, por su funcionalidad y el rendimiento a escala. Algunos ejemplos, son: Cassandra, Redis, MongoDB y CouchDB.
- **Jerárquicas:** útiles en aplicaciones que manejan un gran volumen de información. Durante años fueron uno de los modelos de gestión de bases de datos más utilizados, pero con el

tiempo y la aparición de otros más ágiles, han ido cayendo en desuso. Puedes encontrar más información al respecto en el siguiente [enlace](#).

Nos centraremos en las bases de datos relacionales, ya que este es el modelo utilizado en la actualidad para poder diseñar e implementar aplicaciones empresariales.

BASES DE DATOS RELACIONALES

Tienen dos componentes principales: la “Entidad” y las “Relaciones”

Entidad: es la unidad que representa lógicamente un conjunto finito de datos del mismo tipo, por ejemplo: Clientes, Ventas, Pagos, entre otros; donde cada uno de estos son entidades.



Ilustración 3. Representación de las entidades de una BD.

Relación: une lógicamente dos o más tablas, para indicar qué dependencias pueden existir entre ellas, es decir, indica la forma en que se relacionan las tablas.



Ilustración 4. Representación de las entidades y relaciones que existen.

COMPONENTES DE UNA BASE DE DATOS RELACIONAL

Encontramos los siguientes:

- Base de datos.
- Tabla.
- Atributos.
- Relaciones.
- Índices.

RDBMS (RELATIONAL DATABASE MANAGEMENT SYSTEM)

Es un programa que nos permite gestionar bases de datos relacionales. Para entenderlo mejor, previo a los RDBMS estaban los DBMS (Relational Database Management System) para gestionar base de datos. Se compone principalmente de:

- Interfaz: permite al usuario comunicarse con los DBMS.
- Database language engine: motor de base de datos para interpretar las consultas, y realizar las acciones necesarias usando su lenguaje.
- Query optimizer: para optimizar el lenguaje de consulta, con el objetivo de poder ejecutar las consultas lo más rápido posible.
- Database engine: para gestionar objetos de la base de datos, como lo son las tablas o vistas DBMS Management Component. Son componentes de gestión encargados de realizar copias de seguridad, monitorización de rendimiento, seguridad, entre otros.

Los **RDBMS** son una mejora de los **DBMS** al introducir el modelo relacional, permitiendo la relación entre tablas, y usando claves primarias, claves foráneas e índices.

Como principales diferencias entre **DBMS** y **RDBMS**, se pueden destacar:

Los **DBMS** están orientados para aplicaciones que gestionen pocos datos, pues puede aumentar la complejidad y lentitud a mayor volumen de éstos, debido a que es muy común la presencia de datos redundantes. Entre estos sistemas se encuentran: *LibreOffice Base* y *FoxPro*.

Mientras que los **RDBMS**, están orientados a gestionar un mayor volumen de datos gracias al modelo relacional, permitiendo realizar consultas más rápidas, y también evitar los datos redundantes mediante el uso de claves de índices. Entre estos: *PostgreSQL*, *SQL Server*, *Oracle*, *MySQL*, *MariaDB*, entre otros.

¿QUÉ ES SQL?

Structured Query Language (Lenguaje estructurado de consultas), es un lenguaje creado para la definición y manipulación de bases de datos relacionales. Su beneficio es que facilita la administración de datos almacenados.



EL ROL DE LAS BASES DE DATOS RELACIONALES EN ORGANIZACIONES

Según IBM, las bases de datos relacionales pueden ayudar a las personas y empresas a "comprender mejor las relaciones entre todos los datos disponibles, y obtener nuevos conocimientos para tomar mejores decisiones o identificar nuevas oportunidades". Por ejemplo: imagina que una empresa mantiene una tabla de clientes, la cual contiene datos sobre cada cuenta de cliente, y una o más tablas transacciones que contienen datos para describir operaciones individuales.

Utilizando todos estos datos, las aplicaciones como los generadores de informes, pueden ejecutar consultas a bases de datos para crear informes formales. De hecho, gran parte de los documentos que ejecutan las empresas para realizar un seguimiento del inventario, las ventas, las finanzas, o incluso realizar proyecciones financieras, provienen del uso de una base de datos relacional.

¿CUÁNDO UTILIZAR SQL O NOSQL?

Tanto SQL como NoSQL, son tipos de bases de datos recomendadas para utilizar a la hora de comenzar a desarrollar tus proyectos, y cada una de ellas presenta ventajas y desventajas.

 SQL	 NoSQL
Cuando el volumen de mis datos no crece o lo hace poco a poco.	Cuando el volumen de mis datos crece muy rápidamente en momentos puntuales.
Cuando las necesidades de proceso se pueden asumir en un sólo servidor.	Cuando las necesidades de proceso no se pueden prever.
Cuando no tenemos picos de uso del sistema por parte de los usuarios más allá de los previstos.	Cuando tenemos picos de uso del sistema por parte de los usuarios en múltiples ocasiones.

Por ejemplo, los sistemas contables, o de inventario, requieren transacciones de varias filas, y para este tipo de trabajos la mejor opción son las bases de datos SQL (PostgreSQL). Mientras que, si los sistemas son de gestión de contenido, aplicaciones móviles, de análisis en tiempo real, bases de datos con un crecimiento rápido, con un esquema descentralizado, son preferibles las bases de datos NOSQL (MongoDB).