

Clase N° 10: Bases de datos



¿Qué son las bases de datos?

- Conjunto organizado de información o datos estructurados
- Se almacenan electrónicamente en un sistema computacional
- Permite gestionar, buscar y actualizar datos de manera eficiente
- Se gestionan mediante software llamado Sistemas de Gestión de Bases de Datos (DBMS).
- Puede ser accedida por múltiples usuarios al mismo tiempo

Tipos de bases de datos

Existen múltiples tipos de bases de datos, cada una de ellas se adapta mejor a nuestra necesidad de negocios

- Relacionales
- NoSQL
- Objetos
- En memoria



BDD Relacionales

Son las más utilizadas en la industria. Están diseñadas para almacenar datos estructurados en TABLAS relacionadas entre sí.

Las tablas se organizan en filas y columnas, y utilizan claves (PK y FK) para relacionar datos entre ellas.

Son muy escalables y ofrecen flexibilidad a la hora de consultar los datos



BDD NoSQL

Las bases de datos NoSQL son una alternativa a las bases de datos relacionales.

No utilizan tablas y claves como las bases de datos relacionales, sino que utilizan una variedad de estructuras de datos, como documentos, gráficos y pares clave-valor.

Las bases de datos NoSQL son muy escalables y ofrecen un rendimiento excelente para grandes conjuntos de datos no estructurados, como datos de redes sociales o de Internet de las cosas (IoT).



DBB Objetos

Guarda archivos en un entorno de datos plano, o grupo de almacenamiento, como un objeto autónomo que contiene todos los datos, un identificador único y metadatos detallados que contienen información sobre los datos, permisos, políticas y otras contingencias.

El almacenamiento de objetos funciona mejor para el almacenamiento estático, en especial con los datos no estructurados, en los que escribes datos una vez, pero que quizás necesitas leerlos muchas veces.



BDD Relacionales

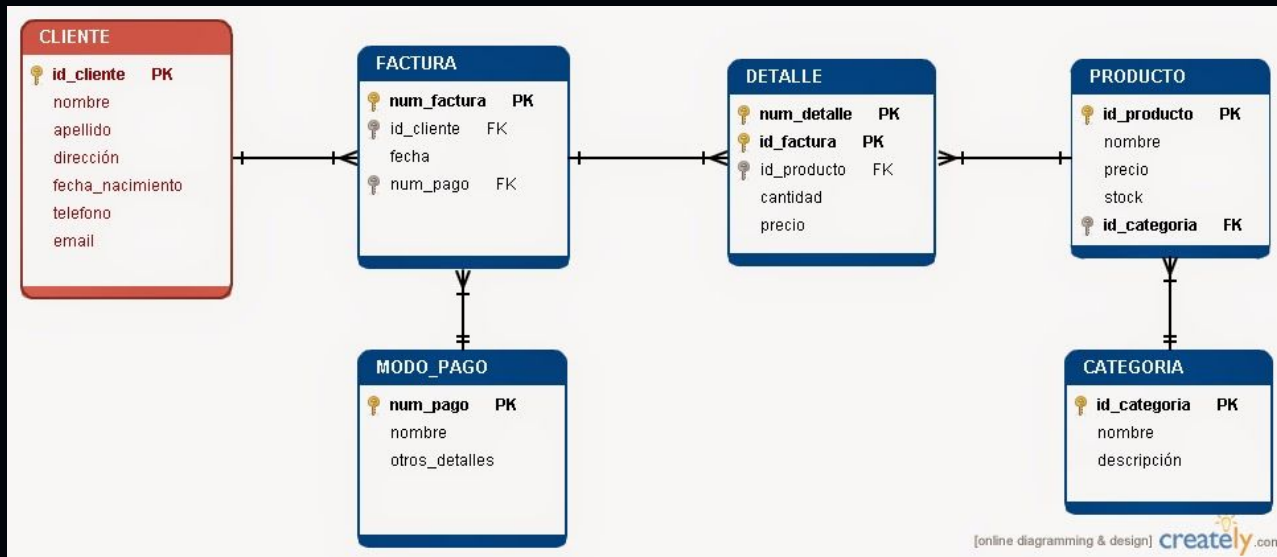
Una base de datos relacional organiza los datos en tablas con **filas** y **columnas**, donde cada fila representa un **registro** y cada columna, un atributo del mismo. Utiliza relaciones entre tablas para vincular datos de manera lógica, lo que permite realizar consultas complejas y obtener información de manera eficiente.

Ejemplo:

Clientes			Pedidos		
id	nombre	edad	id	total	id_cliente
1	Jose	29	1	350	2
2	Pedro	34	2	935.5	2

BDD Relacionales: Tablas

Son estructuras fundamentales dentro de una base de datos relacional. Representan un conjunto organizado de datos relacionados, organizados en filas (registros) y columnas (campos).



MySQL: Instalación

Windows:

- Mysql Workbench: <https://dev.mysql.com/downloads/workbench/>

Linux:

- `sudo apt install mysql-server`
- `sudo systemctl status mysql`
- Dbeaver: <https://dbeaver.io/download/>



Lenguaje de consultas SQL

SQL (Structured Query Language) es un lenguaje estándar utilizado para gestionar y manipular bases de datos relacionales. Es un lenguaje declarativo que permite a los usuarios interactuar con las bases de datos para realizar tareas de búsqueda, actualización y eliminación de datos

Es el lenguaje que nos permite comunicarnos con el motor de base de datos

```
53 USE DatabaseName;  
54 GO  
55  
56 CREATE PROCEDURE ProcedureName  
57     @FirstName type,  
58     @LastName type....  
59 AS  
60  
61     //Your SQL query here  
62     Select FirstName, LastName  
63     FROM Employee  
64     WHERE FirstName = @FirstName AND LastName = @LastName  
65 GO
```

SQL: tipos de comandos

DDL: Permite definir y modificar la estructura de las bases de datos

- CREATE: Crea bases de datos, tablas, vistas, etc.
- ALTER: Modifica la estructura de una base de datos o tabla.
- DROP: Elimina bases de datos, tablas, vistas, etc.

DML: Se usa para gestionar los datos dentro de las tablas

- SELECT: Recupera datos de la base de datos.
- INSERT: Agrega nuevos registros en una tabla.
- UPDATE: Modifica registros existentes en una tabla.
- DELETE: Elimina registros de una tabla.

DDL: CREATE

```
-- Crear la tabla Clientes
CREATE TABLE Clientes (
    ID_Cliente INT AUTO_INCREMENT PRIMARY KEY,
    Nombre VARCHAR(100),
    Email VARCHAR(100) UNIQUE,
    Fecha_Registro DATE
);
```

DDL: ALTER TABLE

```
ALTER TABLE Clientes  
MODIFY COLUMN Nombre VARCHAR(200);
```

```
ALTER TABLE Clientes  
CHANGE COLUMN Fecha_Registro Fecha_Inscripcion DATE;
```

```
ALTER TABLE Clientes  
DROP COLUMN Telefono;
```

DDL: DROP

```
DROP TABLE Clientes;
```

```
DROP DATABASE nombre_base_de_datos;
```

DML: SELECT

```
-- Obtener los clientes con su id, nombre, edad  
SELECT * FROM Clientes
```

```
-- Obtener el ID de los clientes  
SELECT id FROM Clientes  
  
-- Obtener nombre y edad de los clientes  
SELECT nombre, edad FROM Clientes
```

Clientes

id	nombre	edad
1	Jose	29
2	Pedro	34

DML: INSERT

```
INSERT INTO Clientes VALUES ("Pablo", 25)
```

Clientes

id	nombre	edad
1	Jose	29
2	Pedro	34
3	Pablo	25

DML: UPDATE

```
UPDATE Clientes SET nombre="Carlos" WHERE id=3
```

Usamos la cláusula "WHERE" para filtrar el conjunto de datos que queremos actualizar

Clientes

id	nombre	edad
1	Jose	29
2	Pedro	34
3	Pablo Carlos	25

DML: DELETE

```
DELETE FROM Clientes WHERE id=3
```

Usamos la cláusula “WHERE” para filtrar el conjunto de datos que queremos eliminar

Clientes

id	nombre	edad
1	Jose	29
2	Pedro	34
3	Carlos	25

WHERE

Es una cláusula en SQL que se utiliza para filtrar registros en una consulta, mostrando sólo aquellos que cumplen con una condición específica.

```
SELECT columnas  
FROM tabla  
WHERE condición;
```

```
SELECT *  
FROM empleados  
WHERE salario > 50000;
```

Importante: no olvides usar WHERE en conjunto UPDATE Y DELETE, ya que de lo contrario se eliminarán/actualizarán todos los registros de la tabla y dicha acción es irreversible

Relaciones entre tablas

Las relaciones en SQL permiten conectar datos entre diferentes tablas mediante claves primarias y foráneas, facilitando la organización y consulta eficiente en bases de datos relacionales.

Tipos:

- Uno a Uno: un registro de una tabla está asociado a un único registro de otra tabla
 - Ej: Usuario – Perfil Un usuario tiene un único perfil
- Uno a Muchos: un registro en una tabla está asociado con varios registros en otra.
 - Ej: Usuario → Publicaciones. El usuario puede tener múltiples publicaciones.
- Muchos a Muchos: Varios registros en una tabla están asociados con varios registros en otra. Se implementa usando una tabla intermedia.
 - Ej: Productos ↔ Ordenes. La orden tiene múltiples productos, y el producto puede estar en múltiples órdenes

Clave primaria y Clave Foránea

Las claves son elementos esenciales para establecer relaciones y garantizar la integridad de los datos en una base de datos relacional.

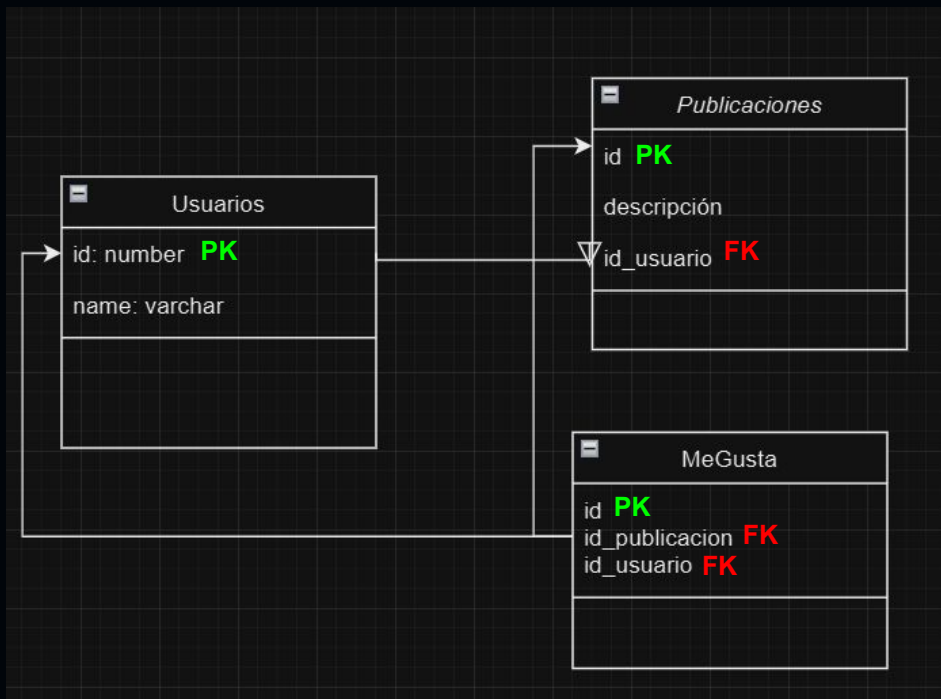
Clave Primaria (Primary Key): Es un identificador único para cada registro en una tabla.

- No puede tener valores duplicados.
- No permite valores nulos (NULL).
- Cada tabla debe tener una única clave primaria.

Clave Foránea (Foreign Key): Es una columna en una tabla que se refiere a la clave primaria de otra tabla, creando una relación entre ambas.

- Garantiza que los valores coincidan con los de la clave primaria en la tabla referenciada.
- Ayuda a mantener la integridad referencial (evitar registros huérfanos).

Clave primaria y Clave Foránea



Práctica grupal

Una tienda de libros necesita gestionar su base de datos. La tienda vende libros de diferentes autores y cada libro pertenece a una categoría. Ayuda a construir y realizar consultas sobre la base de datos.

Preguntas:

¿Que ideas se les ocurren?

¿Qué tablas necesitamos?

¿Cuales son las relaciones?

Práctica

- Crea una base de datos para una red social
 - Crea las tablas publicaciones, me_gustas, comentarios, usuarios
 - Carga las tablas con datos, puedes usar GPT para generar los INSERTS
 - Crea las relaciones entre las FK y PK para cada tabla según corresponda
 - Armar las consultas SQL para obtener la siguiente información:
-
- Obtén todos los usuarios registrados.
 - Muestra los nombres y correos electrónicos de los usuarios registrados en el último mes
 - Lista todas las publicaciones junto con su título y fecha de creación.
 - Encuentra todos los comentarios relacionados con una publicación específica.
 - Crea una nueva publicación para un usuario específico.
 - Añade un comentario a una publicación por parte de un usuario específico.
 - Cambia la fecha de un comentario a la fecha actual.
 - Incrementa el contador de "me gusta" de una publicación agregando un registro.
 - Marca una publicación como "archivada" (requiere agregar un campo ficticio).