

**Bases de Datos.**

Juan José Gracia Gómez

Ingeniería de Sistemas Y computación  
Universidad de Cundinamarca

Profesor: William Alexander Matallana Porras

Chía

2025

## Contenido

Introducción.....	3
Objetivo:.....	4
¿Qué es un dato? .....	5
¿Qué es información?.....	5
¿Qué son tablas? .....	5
Componentes de un Tabla: .....	5
Registros: .....	5
Campos: .....	6
Clave o Llave Primaria:.....	6
Clave o Llave Foránea: .....	6
Relaciones entre tablas .....	6
Una a Una (1:1):.....	6
Relación de uno a muchos (1:N):.....	7
Relación de muchos a muchos (N:M):.....	8
¿Qué significa SQL?.....	9
¿Qué es DDL en SQL? .....	9
¿Qué es DML en SQL?.....	9
¿Qué es un Sistema Gestor de Bases de Datos? .....	10
SGBD más utilizados:.....	10
¿Qué es la normalización de una base de datos? .....	10
Tipos de datos en una base de datos. ....	11
Numéricos:.....	11
Texto: .....	11
Fecha y Hora: .....	11
Booleanos: .....	12
Binarios:.....	12
Conclusiones. ....	13
Referencias:.....	14

### **Introducción.**

Las bases de datos se usan para guardar y organizar información de forma ordenada. Gracias a ellas, es posible trabajar con datos de manera eficiente y precisa. A través de tablas, relaciones, tipos de datos y el uso del lenguaje SQL, podemos insertar, modificar o consultar información fácilmente. Además, herramientas como los Sistemas Gestores de Bases de Datos (SGBD) y la normalización ayudan a que los datos estén bien organizados y sin errores repetidos.

**Objetivo:**

Aprender cómo funcionan las bases de datos, cómo se organizan los datos y cómo usar SQL para trabajar con esa información de forma correcta.

### ¿Qué es un dato?

Un dato es una unidad mínima de información que, por sí sola, carece de significado. Representa un hecho, una medición, una característica o un símbolo sin ningún contexto que lo interprete. Según la AWS, "un dato en una base de datos es una unidad básica de información que se almacena y organiza dentro del sistema." (s. f.).

Ejemplo;

Numero “37”, el cual es un dato, pero no tiene contexto; es decir no indica algo en específico.

### ¿Qué es información?

Hace referencia a datos los cuales han sido procesados y organizados de manera que adquieren significado y son útiles para la toma de decisiones.

Cristina tiene 45 años.

En este caso la información que se proporciona da un contexto claro para el dato “45”.

### ¿Qué son tablas?

Es una estructura fundamental para organizar y almacenar datos relacionados. Se compone de filas y columnas, similar a una hoja de cálculo. Como se menciona en ComparaSoftware “son objetos o estructuras que contienen los datos organizados en filas y columnas. Cada fila representa un registro único, y cada columna un campo dentro del registro.” (s. f.).

Productos		
Id	Nombre	Precio
1	Laptop	5000
2	Teclado	200

En este caso la tabla sería de productos tecnológicos y cada columna pertenece a una característica, y cada fila a un producto.

### Componentes de un Tabla:

#### Registros:

Corresponde a las filas de la tabla. Cada fila contiene la información de algo específico y este se compone de datos en los campos.

#### Ejemplo:

En la tabla “Productos”, un registro sería la fila con los datos de un producto, como: (1, “Teclado”, “200”)

**Campos:**

Son las columnas de la tabla. Cada campo en este caso representa una característica o atributo de la entidad que estemos manejando.

**Ejemplo:**

En la tabla “Productos”, los campos son ID, Nombres, Precio.

**Clave o Llave Primaria:**

Es un campo que permite identificar cada registro, pero de manera única en una tabla. En dicho campo ningún valor puede repetirse para garantizar que cada registro es único.

**Ejemplo:**

En la tabla “Productos”, el campo de “Id” podría ser la llave primaria, ya que cada producto tiene su Id único.

**Clave o Llave Foránea:**

Es un campo del cual se hace uso para poder crear una relación entre dos tablas. Hace referencia la clave primaria de otra tabla, lo que permite vincular diferentes registros.

**Ejemplo:**

En una Tabla “Compras”, el campo “Productos\_Id” podría ser una clave foránea que se refiere al campo “Id” de la tabla “Productos”.

**Relaciones entre tablas****Una a Una (1:1):**

Un solo registro de una tabla se asocia con un único registro en otra tabla. Aunque no es muy común puede llegar a ser útil cuando los datos de una tabla deben ser almacenados por separado, pero aun deben estar vinculados de manera exclusiva a una fila en otra tabla.

**Ejemplo:**

En el caso de tener dos tablas; Productos y Ventas. Cada producto tiene un único Id y cada Id pertenece a un único Producto.

En la tabla Productos, el registro de un producto tiene un campo llamado “Id”.

En la tabla Ventas, cada registro tiene un campo llamado “Id” que hace referencia al “Id” de la tabla Productos.

```
CREATE TABLE Productos (
```

```

    Id INT PRIMARY KEY,
    Nombre VARCHAR(100),
    Precio DECIMAL(10, 2)
);

```

```

CREATE TABLE Ventas (
    ID_Venta INT PRIMARY KEY,
    Id INT,
    Cantidad INT,
    FOREIGN KEY (Id) REFERENCES Productos(Id)
);

```

### **Relación de uno a muchos (1:N):**

Se da cuando un registro en una tabla puede estar asociado con varios registros en otra tabla, pero cada uno de esos registros en la segunda tabla solo se asocia con un único registro de la primera tabla.

### **Ejemplo:**

En una tabla Clientes cada cliente tendría un identificado único (Id\_Cliente).

En una tabla Ventas cada venta tendría un campo llamado “Id\_Cliente” que hace referencia al “Id\_Cliente” de la tabla Clientes.

```

CREATE TABLE Clientes (
    ID_Cliente INT PRIMARY KEY,
    Nombre VARCHAR(100),
    Correo VARCHAR(100)
);

```

```

CREATE TABLE Ventas (
    ID_Venta INT PRIMARY KEY,
    Fecha DATE,

```

```
ID_Cliente INT,
FOREIGN KEY (ID_Cliente) REFERENCES Clientes(ID_Cliente)
);
```

En este caso un cliente podría hacer muchas compras que se verían reflejadas en la tabla de Ventas, pero cada venta estaría asociada a un único cliente.

### **Relación de muchos a muchos (N:M):**

Sucede cuando varios registros en una tabla se pueden asociar con varios registros de otra tabla. Generalmente se hace uso de una tabla intermedia que contiene claves foráneas de ambas tablas relacionadas. Dicha tabla sirve para hacer la conexión entre los registros de ambas tablas.

### **Ejemplo:**

En un caso de dos tablas Estudiantes y Clases. Un estudiante puede estar unscrito en varias clases, y un curso puede tener varios estudiantes inscritos. Para manejar esta relación, se crearía una tabla intermedia que se llame Inscripciones.

```
CREATE TABLE Estudiantes (
    ID_Estudiante INT PRIMARY KEY,
    Nombre VARCHAR(100)
);
```

```
CREATE TABLE Clases (
    ID_Clase INT PRIMARY KEY,
    Nombre_Clase VARCHAR(100)
);
```

```
CREATE TABLE Inscripciones (
    ID_Estudiante INT,
    ID_Clase INT,
    PRIMARY KEY (ID_Estudiante, ID_Clase),
    FOREIGN KEY (ID_Estudiante) REFERENCES Estudiantes(ID_Estudiante),
```



```
FOREIGN KEY (ID_Clase) REFERENCES Clases(ID_Clase)
);
```

En este caso hacemos uso de la tabla Inscripciones para hacer la relación de muchos a muchos entre Estudiantes y Clases

### ¿Qué significa SQL?

SQL (Structured Query Language) es un lenguaje usado para gestionar bases de datos relacionales. Permite realizar consultas, insertar, modificar o eliminar datos de forma eficiente. Es clave en áreas como desarrollo de software, análisis de datos y administración de sistemas. Según IBM, "SQL es el lenguaje estándar para comunicarse con un sistema de gestión de bases de datos relacional" (IBM, 2025).

#### Ejemplo:

```
SELECT * FROM Productos;
```

Si quisiéramos ver todos los productos en la tabla “Productos”, se escribiría una consulta SQL como esta.

### ¿Qué es DDL en SQL?

El Lenguaje de Definición de Datos (Data Definition Language) en SQL se utiliza para crear y modificar la estructura de una base de datos, como tablas, índices o esquemas. Algunos comandos que corresponden a esta categoría son: CREATE, ALTER, DROP y TRUNCATE. Según Market Tay, "DDL son los comandos que permiten definir la estructura de una base de datos, como la creación y modificación de tablas" (Market Tay, s.f.).

#### Ejemplo:

```
CREATE TABLE Profesores (ID INT, Nombre VARCHAR(100));
```

En este caso se crea una tabla Profesores

### ¿Qué es DML en SQL?

El Lenguaje de Manipulación de Datos (Data Manipulation Language) permite trabajar con los datos almacenados dentro de las tablas, es decir, insertar, actualizar, eliminar o consultar información. Entre los comandos más utilizados están INSERT, UPDATE, DELETE y SELECT. Como indica Market Tay, "DML son los comandos que permiten gestionar los datos contenidos en las tablas" (Market Tay, s.f.).

#### Ejemplo:

```
INSERT INTO Profesores (ID, Nombre) VALUES (1, 'Martin Gutierrez');
```

Se inserta un profesor en la tabla “Profesores”

### ¿Qué es un Sistema Gestor de Bases de Datos?

Es un software que permite crear, gestionar y manipular bases de datos. Facilita el almacenamiento, recuperación y protección de datos, asegurando su integridad y seguridad. Según InterSystems, "un SGBD nos permite almacenar, recuperar y manipular datos, actuando como interfaz entre los usuarios y las bases de datos" (InterSystems, s.f.).

#### SGBD más utilizados:

- **MySQL:** Usado principalmente en aplicaciones web, de código abierto.
- **PostgreSQL:** Conocido por su robustez y cumplimiento de estándares.
- **Oracle Database:** Ideal para grandes empresas por su escalabilidad y soporte.
- **Microsoft SQL Server:** Integrado con otros productos de Microsoft, adecuado para entornos empresariales.
- **SQLite:** Ligero y embebido, común en aplicaciones móviles y de escritorio.

### ¿Qué es la normalización de una base de datos?

Es un proceso en el cual se organizan datos para reducir la redundancia y evitar problemas de actualización. Consiste en aplicar reglas que estructuran la información en tablas relacionadas, asegurando que cada dato se almacene una sola vez. Según ProgramaciónPro, "la normalización de base de datos es un proceso que consiste en organizar los datos de una base de datos de manera que se reduzca la redundancia y se mejore la integridad de la información" (ProgramaciónPro, s.f.).

#### Ejemplo:

Alumnos Inscritos				
<b>Id_Alumno</b>	<b>Nombre</b>	<b>Apellido</b>	<b>Id_Curso</b>	<b>Descripción</b>
1	María	Rodríguez	A200	Algebra
2	Luis	Colmenares	B300	Ingles
3	Felipe	Ruiz	A400	Programación

Alumno		
<b>Id</b>	<b>Nombre</b>	<b>Apellido</b>
1	María	Rodríguez
2	Luis	Colmenares
3	Felipe	Ruiz

Matricula	
Id_Alumno	Id_Curso
1	A200
2	B300
3	A400

Curso	
Id_Curso	Descripción
A200	Algebra
B300	Ingles
A400	Programación

En este caso la tabla principal se separo en varias tablas para evitar la redundancia.

### **Tipos de datos en una base de datos.**

Indican qué tipo de valores puede almacenar una columna en una base de datos. Algunos tipos comunes son:

#### **Numéricos:**

Almacena números, pueden ser de tipo (INT, DECIMAL, FLOAT).

#### **Ejemplo:**

```
CREATE TABLE Empleados (
    ID_Empleado INT PRIMARY KEY,
    Edad INT
);
```

En este caso “Edad” almacenaría valores como 10, 40, etc.

#### **Texto:**

Se utiliza para almacenar cadenas de texto; como (CHAR, VARCHAR Y TEXT).

#### **Ejemplo:**

```
CREATE TABLE Estudiantes (
    ID_Estudiante INT PRIMARY KEY,
    Nombre VARCHAR(100)
);
```

Un campo “Nombre” en este caso de tipo VARCHAR(100) podría almacenar nombres como “Luis Martinez”.

#### **Fecha y Hora:**

Se puede usar para representar fechas y horas; se usan (DATE, TIME DATETIME Y TIMESTAMP)

**Ejemplo:**

```
CREATE TABLE Empleados (  
    ID_Empleado INT PRIMARY KEY,  
    Fecha_Nacimiento DATE  
);
```

En el campo “Fecha\_Nacimiento” de tipo Date se podría almacenar fechas como “2009-12-28”

**Booleanos:**

Almacenan valores lógicos, generalmente TRUE o FALSE.

**Ejemplo:**

```
CREATE TABLE Usuarios (  
    ID_Usuario INT PRIMARY KEY,  
    Activo BOOLEAN  
);
```

Un campo “Activo” de tipo BOOLEAN puede almacenar valores como TRUE O FALSE para indicar si una cuenta de usuarios esta activa o no.

**Binarios:**

Para datos no textuales, como imágenes o archivos, se utilizan tipos como BLOB y VARBINARY.

**Ejemplo:**

```
CREATE TABLE Productos (  
    ID_Producto INT PRIMARY KEY,  
    Imagen BLOB  
);
```

Si se tiene una tabla para almacenar imágenes de productos, en una columna tipo BLOB se podría almacenar imágenes del producto.

### **Conclusiones.**

- Las bases de datos permiten transformar datos aislados en información útil mediante estructuras organizadas como tablas y registros.
- El uso de SQL, junto con los lenguajes DDL y DML, facilita la creación, modificación y gestión de los datos almacenados.
- La normalización y los SGBD son clave para garantizar integridad, evitar redundancia y mantener la eficiencia del sistema.

### **Uso de IA**

El uso de inteligencia artificial para este documento fue de un 27% el cual corresponde a la ayuda al momento de crear algunos de los ejemplos que sirvieron para uso de ese trabajo

**Referencias:**

- ComparaSoftware. (2023, 20 de noviembre). *¿Qué es una tabla en base de datos? Definición y tipos.* <https://blog.comparasoftware.com/que-es-tabla-en-base-de-datos/>
- Amazon Web Services, Inc. (s.f.). *¿Qué es una base de datos?*. <https://aws.amazon.com/es/what-is/database/>
- IBM. (2025). *¿Qué es SQL (lenguaje de consulta estructurada)?* <https://www.ibm.com/mx-es/think/topics/structured-query-language>
- Market Tay. (s.f.). *Diferencia entre DDL y DML en DBMS (con tabla de comparación).* <https://markettay.com/diferencia-entre-ddl-y-dml-en-dbms/>
- InterSystems. (s.f.). *¿Qué es un sistema de gestión de bases de datos (SGBD)?*. <https://www.intersystems.com/es/recursos/que-es-sistema-de-gestion-de-bases-datos-sgbd/>
- ProgramaciónPro. (s.f.). *¿Qué es la normalización de base de datos y por qué es importante?* <https://programacionpro.com/que-es-la-normalizacion-de-base-de-datos-y-por-que-es-importante/>