**Actividad Programación**

Mónica Yulihet Poveda Carrasco

**Profesor**

William Alexander Matallana Porras

Universidad de Cundinamarca, Extensión Chía

Faculta de Ingenieria

Programación I

2025

# Tabla de Contenido

[1. Tabla de Contenido 2](#_Toc196405532)

[2. Introducción 2](#_Toc196405533)

[3. Objetivo 3](#_Toc196405534)

[4. Desarrollo 3](#_Toc196405535)

[1. Que es base de datos 3](#_Toc196405536)

[2. que es un dato 3](#_Toc196405537)

[3. que es información 3](#_Toc196405538)

[4. Que son las tablas 3](#_Toc196405539)

[5. Que son los registros 3](#_Toc196405540)

[6. Que es una clave o llave primaria 3](#_Toc196405541)

[7. Que es una clave o llave foranea 3](#_Toc196405542)

[8. Cuales son las relaciones entre tablas 3](#_Toc196405543)

[8.1 1- 1 3](#_Toc196405544)

[8.2 1- muchos 3](#_Toc196405545)

[8.3 muchos -muchos 3](#_Toc196405546)

[9. que es SQL 3](#_Toc196405547)

[10. Que son las consultas DDL & DML 3](#_Toc196405548)

[11. Que es un sistema gestor de datos 3](#_Toc196405549)

[12. Que es la normalización de una base de datos 3](#_Toc196405550)

[13. Cuales son los tipos de datos en una base de datos 3](#_Toc196405551)

[5. Conclusiones 3](#_Toc196405552)

[6. Referencias bibliográficas 3](#_Toc196405553)

# Introducción

# Objetivo

# Desarrollo

## Que es una base de datos

Una base de datos es una herramienta para recopilar y organizar información. Las bases de datos pueden almacenar información sobre personas, productos, pedidos u otras cosas. Muchas bases de datos comienzan como una lista en una hoja de cálculo o en un programa de procesamiento de texto. A medida que la lista aumenta su tamaño, empiezan a aparecer redundancias e inconsistencias en los datos. Cada vez es más difícil comprender los datos en forma de lista y los métodos de búsqueda o extracción de subconjuntos de datos para revisión son limitados. Una vez que estos problemas empiezan a aparecer, es una buena idea transferir los datos a una base de datos creada por un sistema de administración de bases de datos (DBMS), como Access.

Usando Access, se puede:

* Agregar nuevos datos a una base de datos, como un nuevo artículo en un inventario.
* Modificar datos existentes en la base de datos, por ejemplo, cambiar la ubicación actual de un artículo.
* Eliminar información, por ejemplo, si un artículo se vende o se descarta.
* Organizar y ver los datos de diferentes formas.
* Compartir los datos con otras personas mediante informes, correo electrónico, intranet o Internet.

El propósito principal de una base de datos es facilitar la manipulación y el análisis de la información. Al ser almacenada electrónicamente, esta persiste en el tiempo y puede ser consultada y manipulada en cualquier momento. Esto, a su vez, brinda una gran flexibilidad para trabajar con grandes volúmenes de datos y obtener resultados de manera rápida y precisa.

EJEMPLO:

## Tipos de bases de datos

Existen diversos tipos de bases de datos, clasificadas según su estructura y uso en el ámbito informático. Pueden clasificarse según su uso, pero también siguiendo otros criterios. Cada tipo de base de datos tiene características y ventajas particulares que se adaptan a diferentes necesidades y contextos, lo que las hace perfectas a la hora de gestionar cualquier tipo de proyecto digital de forma completamente fiable, independientemente de cuáles sean su naturaleza o sus objetivos.

Las bases de datos se pueden clasificar en función de diferentes criterios, como su tipo o el uso que se les da en distintos contextos. A continuación, se enumeran las principales clasificaciones de las bases de datos:

**Bases de datos según su tipo:**

* **Bases de datos relacionales:** Son el tipo más común y se basan en el modelo relacional. Utilizan tablas para almacenar y organizar los datos, con relaciones establecidas entre las tablas mediante claves primarias y claves externas.
* **Bases de datos jerárquicas:**Organizan los datos en una estructura jerárquica, similar a un árbol, donde cada registro tiene una relación padre-hijo. Estas bases de datos son eficientes para manejar datos con relaciones fuertemente definidas.
* **Bases de datos de red:** Similares a las jerárquicas, pero permiten relaciones más complejas. Utilizan un modelo de grafo para conectar registros mediante enlaces, lo que brinda mayor flexibilidad en las relaciones de los datos.
* **Bases de datos orientadas a objetos:**Almacenan datos en forma de objetos, que pueden contener atributos y métodos. Permiten almacenar estructuras de datos complejas y mantener una relación directa con los lenguajes de programación orientados a objetos.
* [**Bases de datos NoSQL**](https://www.godaddy.com/resources/latam/digitalizacion/nosql-que-es)**:**Se refieren a un conjunto diverso de sistemas de bases de datos que no se ajustan al modelo relacional. Estas bases de datos están diseñadas para manejar grandes volúmenes de datos no estructurados o semi estructurados, como documentos, gráficos y datos en tiempo real.

**Bases de datos según su uso:**

* **Bases de datos transaccionales:** Son utilizadas en aplicaciones empresariales que requieren el manejo de transacciones, donde se garantiza la atomicidad, consistencia, aislamiento y durabilidad (ACID) de las operaciones. Ejemplos incluyen sistemas de gestión de bases de datos (SGBD) para bancos y sistemas de reservas.
* **Bases de datos analíticas:** Están optimizadas para consultas complejas y análisis de datos. Se utilizan en aplicaciones de inteligencia empresarial y data warehousing, donde se busca extraer información valiosa a partir de grandes volúmenes de datos.
* **Bases de datos espaciales:** Estas bases de datos se especializan en el almacenamiento y la consulta de datos relacionados con la ubicación geográfica. Permiten la manipulación de datos geoespaciales, como mapas, coordenadas y áreas de interés.
* **Bases de datos temporales:**Mantienen un historial de los cambios realizados en los datos a lo largo del tiempo. Son útiles en aplicaciones como auditoría, seguimiento de versiones y gestión de datos históricos.
* **Bases de datos multimedia:**Almacenan y gestionan datos multimedia, como imágenes, audio y video. Están diseñadas para permitir la indexación, búsqueda y recuperación eficiente de contenido multimedia.

Tenemos que decir que estas clasificaciones no son excluyentes, y una base de datos puede pertenecer a más de una categoría según sus características y usos específicos. La elección de la base de datos adecuada depende de los requisitos y objetivos de cada proyecto o aplicación.

## que es un dato

Un dato en una base de datos es una unidad básica de información que se almacena y organiza dentro del sistema. Así, cada dato representa un valor concreto, como un número, una palabra, una fecha o cualquier otro tipo de información relevante para el contexto de la base de datos en cuestión.

En una base de datos, los datos se estructuran en tablas en las que cada dato se coloca en una celda específica de la tabla, correspondiente a su columna y fila. Así, cada dato se identifica de manera única dentro de la base de datos, lo que permite su identificación y posterior recuperación. Además, los datos en una base de datos pueden estar relacionados entre sí mediante claves y relaciones, lo que permite establecer conexiones y asociaciones significativas entre distintas entidades o tablas. Esto posibilita realizar consultas complejas y obtener información interrelacionada con mucha más profundidad.

Por otro lado, hay que decir que la precisión y la integridad de los datos son fundamentales en una base de datos realmente útil y fiable. Los datos deben ser consistentes, confiables y estar siempre libres de errores o inconsistencias. Para garantizar esto, se aplican reglas y restricciones en la base de datos como la definición de tipos de datos, la validación de campos y la imposición de restricciones de integridad.

EJEMPLO:

1. que es información

La información se puede definir como un conjunto de datos procesados y que tienen un significado (relevancia, propósito y contexto), y que por lo tanto son de utilidad para quién debe tomar decisiones, al disminuir su incertidumbre. Los datos se pueden transforman en información añadiéndoles valor:

* **Contextualizando**: se sabe en qué contexto y para qué propósito se generaron.
* **Categorizando**: se conocen las unidades de medida que ayudan a interpretarlos.
* **Calculando**: los datos pueden haber sido procesados matemática o estadísticamente.
* **Corrigiendo**: se han eliminado errores e inconsistencias de los datos.
* **Condensando**: los datos se han podido resumir de forma más concisa (agregación).

Por tanto, la información es la comunicación de conocimientos o inteligencia, y es capaz de cambiar la forma en que el receptor percibe algo, impactando sobre sus juicios de valor y sus comportamientos.

***Información = Datos + Contexto (añadir valor) + Utilidad (disminuir la incertidumbre)***

EJEMPLO:

## Que son las tablas

Una tabla de base de datos es similar en apariencia a una hoja de cálculo en cuanto a que los datos se almacenan en filas y columnas. Por ende, es bastante fácil importar una hoja de cálculo en una tabla de base de datos. La principal diferencia entre almacenar los datos en una hoja de cálculo y almacenarlos en una base de datos es la forma en la que están organizados los datos.

Para aprovechar al máximo la flexibilidad de una base de datos, los datos deben organizarse en tablas para que no se produzcan redundancias. Por ejemplo, si quiere almacenar información sobre los empleados, cada empleado debe especificarse solo una vez en la tabla que está configurada para los datos de los empleados. Los datos sobre los productos se almacenarán en su propia tabla y los datos sobre las sucursales se almacenarán en otra tabla. Este proceso se denomina normalización.

Cada fila de una tabla se denomina registro. En los registros se almacena información. Cada registro está formado por uno o varios campos. Los campos equivalen a las columnas de la tabla. Por ejemplo, puede tener una tabla llamada "Empleados" donde cada registro (fila) contiene información sobre un empleado distinto y cada campo (columna) contiene otro tipo de información como nombre, apellido, dirección, etc. Los campos deben designarse como un determinado tipo de datos, ya sea texto, fecha u hora, número o algún otro tipo.

EJEMPLO:

## Que son los registros

Un registro es una unidad de información que contiene un conjunto de campos relacionados. Cada registro representa una entidad o instancia única dentro de una tabla o colección de datos.

En la actualidad, los registros siguen siendo la unidad fundamental de almacenamiento y organización de datos en una amplia gama de sistemas de gestión de bases de datos, desde los tradicionales modelos relacionales hasta los más recientes enfoques NoSQL y de almacenamiento de datos en la nube.

Las aplicaciones más relevantes de los registros incluyen el almacenamiento de información sobre clientes, transacciones, inventarios, registros médicos y cualquier otro tipo de datos estructurados que requieran ser gestionados y consultados de manera eficiente. Además, se han integrado con técnicas avanzadas de indexación, particionamiento y replicación para mejorar el rendimiento y la escalabilidad de las bases de datos.

La importancia de los registros radica en su papel fundamental para organizar y acceder a la información de manera estructurada y eficiente. Su uso adecuado, junto con una arquitectura de base de datos bien diseñada, es esencial para el desarrollo de aplicaciones que requieren el almacenamiento y la recuperación de grandes volúmenes de datos.

EJEMPLO:

## Que es una clave o llave primaria

Las tablas se relacionan con otras tablas mediante una relación de clave primaria o de clave foránea. Las relaciones de claves primarias y foráneas se utilizan en las bases de datos relacionales para definir relaciones de muchos a uno entre tablas.

Las relaciones de claves primarias y foráneas entre tablas en un esquema de estrella o copo de nieve, a veces llamadas relaciones de muchos a uno, representan las vías de acceso a través de las cuales las tablas relacionadas se unen en la base de datos. Estas vías de acceso de unión son la base para formar consultas de datos históricos. Para obtener más información sobre las relaciones de muchos a uno, consulte Relaciones de muchos a uno.

**Claves primarias**

Una clave primaria es una columna o un conjunto de columnas en una tabla cuyos valores identifican de forma exclusiva una fila de la tabla. Una base de datos relacional está diseñada para imponer la exclusividad de las claves primarias permitiendo que haya sólo una fila con un valor de clave primaria específico en una tabla.

**La clave primaria debe cumplir tres condiciones:**

* El campo o campos que forman la clave principal de una tabla no puede contener valores nulos. Es decir, siempre tiene que tomar un valor para cada fila de la tabla.
* No pueden haber dos filas en la tabla con el mismo valor en el campo o campos de la clave principal. Es decir, dicho valor no puede repetirse en ninguna fila.
* Sólo puede haber una clave principal por tabla.

IMPORTANTE

* TODAS LAS TABLAS DEBEN TENER LLAVES PRIMARIAS
* Cuando un campo cumple estas dos propiedades (sin nulos y sin repetidos) se le llama Clave Primaria o Clave Principal y toda tabla debe tener una.
* Cuando intentemos insertar una nueva fila con valores que infrinjan estas dos reglas, el sistema no nos deja crear la nueva fila y nos devuelve un error.

EJEMPLO:

## Que es una clave o llave foránea

Una clave foránea es una columna o conjunto de columnas en una tabla que establece un vínculo con la clave primaria de otra tabla dentro de una base de datos relacional. Este vínculo crea una relación referencial, lo que permite que la información esté coherentemente distribuida y relacionada entre diferentes tablas.

Para que una fila pueda contener un valor específico en una clave foránea, debe existir previamente una fila en la tabla referenciada que tenga ese mismo valor como clave primaria. De esta forma, se garantiza la integridad referencial, es decir, que las relaciones entre los datos sean válidas y consistentes.

Este mecanismo es fundamental para evitar la duplicación de datos, organizar la información eficientemente y mantener la coherencia lógica en sistemas complejos donde múltiples entidades están relacionadas entre sí.

EJEMPLO:

El uso adecuado de llaves primarias y foráneas también contribuye a la optimización del rendimiento en una base de datos relacional. Al establecer relaciones eficientes entre tablas, se evita la redundancia innecesaria de datos, lo que conduce a un diseño más limpio y a un menor espacio de almacenamiento requerido.

Además, la normalización, un proceso que organiza la estructura de las tablas para minimizar la redundancia y dependencias funcionales, se ve facilitada por la implementación adecuada de llaves primarias y foráneas. Al reducir la duplicación de datos, se mejora la coherencia y la eficacia en la gestión de información.

# Cuales son las relaciones entre tablas

Existen tres tipos de relaciones de tablas en Access:

## 8.1 UNO -UNO

En una relación uno a uno, cada registro de la primera tabla solo puede tener un registro coincidente en la segunda tabla y viceversa. Esta relación no es común porque la mayor parte de las veces la información relacionada de este modo se almacena en la misma tabla. Puede usar la relación uno a uno para dividir una tabla con muchos campos, para aislar parte de una tabla por razones de seguridad o para almacenar información que solo se aplica a un subconjunto de la tabla principal. Cuando identifique esta relación, ambas tablas deben compartir un campo común.

## UNO- MUCHOS

Usemos como ejemplo una base de datos de seguimiento de pedidos que incluya una tabla Clientes y una tabla Pedidos. Un cliente puede realizar cualquier número de pedidos. Por cada cliente representado en la tabla Clientes puede haber representados muchos pedidos en la tabla Pedidos. Por lo tanto, la relación entre la tabla Clientes y la tabla Pedidos es una relación de uno a varios.

Para representar una relación uno a varios en el diseño de la base de datos, tome la clave principal del lado "uno" de la relación y agréguela como campo o campos adicionales a la tabla en el lado "varios" de la relación. En este caso, por ejemplo, agregaría un nuevo campo (el campo Id. de la tabla Clientes) a la tabla Pedidos y lo denominaría Id. de cliente. Así, Access puede usar el número de identificador del cliente de la tabla Pedidos para dar con al cliente correcto de cada producto.

## MUCHOS-MUCHOS

Centrémonos ahora en la relación entre una tabla Productos y una tabla Pedidos. Un solo pedido puede incluir varios productos. Por otra parte, un único producto puede aparecer en muchos pedidos. Por tanto, por cada registro de la tabla Pedidos puede haber varios registros en la tabla Productos. Además, por cada registro de la tabla Productos puede haber varios registros en la tabla Pedidos. Esta relación se denomina relación de varios a varios. Tenga en cuenta que para detectar las relaciones de varios a varios existentes entre las tablas, es importante que considere ambas partes de la relación.

Para representar una relación de varios a varios, debe crear una tercera tabla, a menudo denominada tabla de unión, que divide la relación de varios a varios en dos relaciones uno a varios. Debe insertar la clave principal de cada una de las dos tablas en la tercera. Como resultado, la tercera tabla registra cada ocurrencia, o instancia, de la relación. Por ejemplo, la tabla Pedidos y la tabla Productos tienen una relación varios a varios que se define mediante la creación de dos relaciones uno a varios con la tabla Detalles de pedidos. Un pedido puede incluir muchos productos, y cada producto puede aparecer en muchos pedidos.

## ¿Qué es SQL?

El lenguaje de consulta estructurada (SQL) es un lenguaje de programación para almacenar y procesar información en una base de datos relacional. Una base de datos relacional almacena información en forma de tabla, con filas y columnas que representan diferentes atributos de datos y las diversas relaciones entre los valores de datos. Puede usar las instrucciones SQL para almacenar, actualizar, eliminar, buscar y recuperar información de la base de datos. También puede usar SQL para mantener y optimizar el rendimiento de la base de datos.

## ¿Por qué es importante SQL?

El lenguaje de consulta estructurada (SQL) es un lenguaje de consulta popular que se usa con frecuencia en todos los tipos de aplicaciones. Los analistas y desarrolladores de datos aprenden y usan SQL porque se integra bien con los diferentes lenguajes de programación. Por ejemplo, pueden incrustar consultas SQL con el lenguaje de programación Java para crear aplicaciones de procesamiento de datos de alto rendimiento con los principales sistemas de bases de datos SQL, como Oracle o MS SQL Server. Además, SQL es muy fácil de aprender, ya que en sus instrucciones se utilizan palabras clave comunes en inglés.

## ¿Cuáles son los componentes de un sistema SQL?

Los sistemas de administración de bases de datos relacionales utilizan un lenguaje de consulta estructurada (SQL) para almacenar y administrar datos. El sistema almacena varias tablas de bases de datos que se relacionan entre sí. MS SQL Server, MySQL o MS Access son ejemplos de sistemas de administración de bases de datos relacionales. Los siguientes son los componentes de un sistema de este tipo.

**Tabla SQL**

Una tabla SQL es el elemento básico de una base de datos relacional. La tabla de la base de datos SQL se compone de filas y columnas. Los ingenieros de bases de datos crean relaciones entre varias tablas de bases de datos para optimizar el espacio de almacenamiento de datos.

Por ejemplo, el ingeniero de bases de datos crea una tabla SQL para los productos de una tienda:

Interfaz de usuario gráfica

El contenido generado por IA puede ser incorrecto.

A continuación, el ingeniero de bases de datos vincula la tabla de productos a la tabla de colores con el ID de color

Tabla

El contenido generado por IA puede ser incorrecto.

## Instrucciones SQL

Las instrucciones SQL, o consultas SQL, son instrucciones válidas que los sistemas de administración de bases de datos relacionales entienden. Los desarrolladores de software crean instrucciones SQL a partir de diferentes elementos del lenguaje SQL. Los elementos del lenguaje SQL son componentes como identificadores, variables y condiciones de búsqueda que forman una instrucción SQL correcta.

EJEMPLO:

La siguiente instrucción SQL utiliza un comando SQL INSERT para almacenar la Marca del colchón A, con un precio de 499 USD, en una tabla llamada Mattress\_table, (Tabla\_colchón) con los nombres de columna brand\_name (nombre\_marca) y cost (costo):

INSERT INTO Mattress\_table (brand\_name, cost)

VALUES(‘A’,’499’);

**Procedimientos almacenados**

Los procedimientos almacenados son una colección de una o más instrucciones SQL almacenadas en la base de datos relacional. Los desarrolladores de software utilizan los procedimientos almacenados para mejorar la eficiencia y el rendimiento. Por ejemplo, pueden crear un procedimiento almacenado para actualizar las tablas de ventas en vez de escribir la misma instrucción SQL en diferentes aplicaciones.

## ¿Cómo funciona SQL?

La implementación del lenguaje de consulta estructurada (SQL) implica una máquina servidor que procese las consultas de la base de datos y devuelva los resultados. El proceso SQL pasa por varios componentes de software, incluidos los siguientes.

**Analizador**

El analizador comienza por tokenizar o reemplazar algunas de las palabras de la instrucción SQL con símbolos especiales. A continuación, comprueba si la instrucción tiene lo siguiente:

**Corrección**

El analizador verifica que la instrucción SQL se ajuste a la semántica o reglas de SQL que garantizan la corrección de la instrucción de la consulta. Por ejemplo, el analizador comprueba si el comando SQL termina con punto y coma. Si falta el punto y coma, el analizador devuelve un error.

**Autorización**

El analizador también valida que el usuario que ejecuta la consulta tenga la autorización necesaria para manipular los datos correspondientes. Por ejemplo, solo los usuarios administradores podrían tener derecho a eliminar datos.

**Motor relacional**

El motor relacional, o procesador de consultas, crea un plan para recuperar, escribir o actualizar los datos correspondientes de la manera más eficaz. Por ejemplo, busca consultas similares, vuelve a utilizar métodos de manipulación de datos anteriores o crea uno nuevo. Escribe el plan en una representación de nivel intermedio de la instrucción SQL llamada código de bytes. Las bases de datos relacionales utilizan el código de bytes para hacer búsquedas y modificaciones en ellas.

**Motor de almacenamiento**

El motor de almacenamiento, o motor de base de datos, es el componente de software que procesa el código de bytes y ejecuta la instrucción SQL deseada. Lee y almacena los datos en los archivos de la base de datos del almacenamiento en disco físico. Al finalizar, el motor de almacenamiento devuelve el resultado a la aplicación que lo solicita.

# Que son las consultas DDL & DML

## Que es un sistema gestor de datos

## Que es la normalización de una base de datos

## Cuáles son los tipos de datos en una base de datos

# Que son los campos

# Conclusiones

# Referencias bibliográficas