Introducción a las bases de datos

Las bases de datos son una herramienta fundamental para el almacenamiento, organización y gestión de información en cualquier ámbito. Desde pequeñas empresas hasta grandes corporaciones, pasando por proyectos personales y aplicaciones móviles, todos ellos utilizan bases de datos para almacenar y gestionar información de manera eficiente.

¿Qué es una base de datos?

Una base de datos es un conjunto de datos organizados y relacionados entre sí que se almacenan en un dispositivo de almacenamiento. Estos datos pueden ser de cualquier tipo: números, textos, imágenes, audios, videos, etc.

Tipos de bases de datos

1. **Bases de datos relacionales**: son las más utilizadas y se basan en la organización de los datos en tablas, que se relacionan entre sí mediante claves primarias y claves foráneas.
2. **Bases de datos NoSQL**: son aquellas que no utilizan tablas, sino que almacenan los datos en estructuras más complejas como documentos, grafos o clave-valor.
3. **Bases de datos orientadas a objetos**: son aquellas que almacenan los datos en forma de objetos, y se utilizan principalmente en programación orientada a objetos.

Lenguajes de bases de datos

- **SQL**: es un lenguaje de consulta estructurado, que se utiliza principalmente para manipular bases de datos relacionales.
- **NoSQL**: aunque no es propiamente un lenguaje, NoSQL hace referencia a un conjunto de bases de datos que utilizan otros lenguajes de programación y estructuras de datos.

Fases para el diseño de bases de datos

1. **Análisis de requisitos:**: La primera fase es la de análisis de requisitos, en la que se identifican y documentan las necesidades de la base de datos, incluyendo los tipos de datos que se almacenarán, la cantidad de registros esperados, los patrones de acceso a los datos y las restricciones de integridad de los datos.
2. **Diseño conceptual**: En la segunda fase, se crea un modelo conceptual de la base de datos, que describe los datos y sus relaciones de alto nivel. Esto se hace utilizando herramientas como diagramas de entidad-relación (ER), que muestran las entidades, atributos y relaciones entre los datos.
3. **Diseño lógico**: En la tercera fase, se traduce el modelo conceptual a un modelo lógico, que es una representación más detallada de la base de datos en términos de tablas, campos, claves y relaciones. En esta fase se establecen las reglas de normalización y se crean las tablas y campos necesarios para almacenar los datos.
4. **Diseño físico**: En la cuarta fase, se traduce el modelo lógico a un diseño físico, que describe cómo se implementará la base de datos en el sistema de gestión de bases de datos (DBMS). Esto incluye la definición de las estructuras de almacenamiento, como las tablas, los índices y las vistas.

5. **Implementación**: En la quinta fase, se implementa el diseño físico de la base de datos en el DBMS elegido. Esto implica la creación de las tablas, índices, restricciones y otros objetos de la base de datos, así como la carga inicial de datos.
6. **Pruebas y ajustes**: En la sexta fase, se realizan pruebas y ajustes para asegurarse de que la base de datos funcione correctamente y cumpla con los requisitos establecidos. Esto incluye pruebas de rendimiento, integridad de datos y seguridad.
7. **Mantenimiento**: En la última fase, se lleva a cabo el mantenimiento continuo de la base de datos, incluyendo la realización de copias de seguridad, la optimización del rendimiento y la implementación de nuevas características y funcionalidades.

Modelos de bases de datos

Existen varios modelos de bases de datos que se utilizan en la industria. Aquí hay una lista de algunos de los modelos de bases de datos más comunes:

- **Modelo jerárquico**: en este modelo, los datos se organizan en una estructura jerárquica de árbol, donde cada nodo padre puede tener varios nodos hijos, pero cada nodo hijo solo puede tener un nodo padre. Este modelo se utiliza principalmente para sistemas de información de gestión (SIG) y sistemas de información geográfica (SIG).
- **Modelo de red:**: en este modelo, los datos se organizan en una estructura de red, donde cada registro puede tener varios registros secundarios relacionados y viceversa. Este modelo se utiliza principalmente en aplicaciones de biblioteca y sistemas de facturación.
- **Modelo relacional (entidad/relación)**:este es el modelo de bases de datos más común en uso actualmente. Se basa en el uso de tablas para organizar los datos, con cada tabla representando una entidad en el mundo real y cada fila en la tabla representando un registro en esa entidad. Este modelo utiliza claves primarias y relaciones entre tablas para garantizar la integridad de los datos y permitir la realización de consultas complejas.
- **Modelo orientado a objetos**: este modelo trata los datos como objetos, con propiedades y métodos asociados con cada objeto. Este modelo es útil para aplicaciones que manejan datos complejos y relacionales, como la gestión de redes sociales y los sistemas de información geográfica.
- **Modelo documental**: este modelo almacena los datos en documentos estructurados, como archivos JSON o XML. Este modelo es útil para aplicaciones que manejan grandes cantidades de datos no estructurados, como el almacenamiento de información de sensores o la gestión de archivos multimedia.

Partes de una base de datos

- **Tablas**: las tablas son como hojas de papel donde se escribe la información. Cada tabla tiene un nombre y varias columnas donde se escriben los datos.
- **Campos**: cada columna de una tabla se llama campo. Los campos pueden tener diferentes tipos de datos como texto, números o fechas.
- **Registros**: cada fila de una tabla se llama registro. Cada registro contiene la información de un elemento o persona en particular.
- **Claves**: las claves son como etiquetas especiales que se usan para identificar cada registro. Cada registro tiene una clave única que lo diferencia de los demás registros.
- **Relaciones**: las relaciones son como líneas que unen las tablas entre sí. Las relaciones se usan para conectar la información que está en diferentes tablas.

Tipos de relaciones

En las bases de datos, se pueden establecer distintos tipos de relaciones entre las tablas para reflejar la interconexión de los datos. Algunos de los tipos más comunes de relaciones son:

- **Relación uno a uno (1:1)**: En este tipo de relación, un registro de una tabla está relacionado con exactamente uno de otra tabla y viceversa. Por ejemplo, una tabla de "empleados" puede estar relacionada con una tabla de "direcciones" a través de un campo común como "id_empleado".
- **Relación uno a muchos (1:N)**: En este tipo de relación, un registro de una tabla está relacionado con varios registros de otra tabla, pero cada registro de la segunda tabla solo está relacionado con uno de la primera tabla. Por ejemplo, una tabla de "clientes" puede estar relacionada con una tabla de "pedidos" a través de un campo común como "id_cliente". Un cliente puede tener varios pedidos, pero cada pedido está relacionado con un solo cliente.
- **Relación muchos a uno (N:1)**: En este tipo de relación, varios registros de una tabla están relacionados con un solo registro de otra tabla. Por ejemplo, una tabla de "pedidos" puede estar relacionada con una tabla de "clientes" a través de un campo común como "id_cliente". Varios pedidos pueden estar relacionados con un solo cliente.
- **Relación muchos a muchos (N:N)**: En este tipo de relación, varios registros de una tabla están relacionados con varios registros de otra tabla. Por ejemplo, una tabla de "estudiantes" puede estar relacionada con una tabla de "clases" a través de una tabla intermedia que contiene los campos comunes de ambas tablas como "id_estudiante" y "id_clase". Un estudiante puede estar inscrito en varias clases y cada clase puede tener varios estudiantes.

Tipos de claves

Las claves son utilizadas para identificar y relacionar registros en diferentes tablas. Hay varios tipos de claves que se pueden utilizar en una base de datos, incluyendo:

- **Clave primaria (Primary key)**: Una clave primaria es un campo o un conjunto de campos que identifican de manera única cada registro en una tabla. La clave primaria se utiliza para garantizar que cada registro en una tabla sea único y para relacionar registros en diferentes tablas. Cada tabla debe tener una clave primaria.
- **Clave foránea (Foreign key)**: Una clave foránea es un campo en una tabla que se relaciona con la clave primaria de otra tabla. La clave foránea se utiliza para establecer una relación entre dos tablas y asegurar la integridad referencial de la base de datos.
- **Clave única (Unique key)**: Una clave única es un campo o un conjunto de campos que garantizan que cada registro en una tabla sea único. A diferencia de la clave primaria, una tabla puede tener varias claves únicas.
- **Clave candidata (Candidate key)**: Una clave candidata es un campo o un conjunto de campos que pueden ser utilizados como clave primaria. Una tabla puede tener varias claves candidatas, pero solo una de ellas puede ser seleccionada como clave primaria.
- **Clave compuesta (Composite key)**: Una clave compuesta es un conjunto de campos que se utilizan juntos como clave primaria. En lugar de utilizar un solo campo como clave primaria, se utiliza una combinación de campos.
- **Clave natural (Natural key)**: Una clave natural es un campo o un conjunto de campos que ya existen en los datos de la tabla

Tipos de campos

Los campos son los elementos más básicos de una base de datos. Los campos definen el tipo de datos que se pueden almacenar en una tabla. A continuación, te explico los tipos de campos más comunes en las bases de datos:

- **Cadena de caracteres (VARCHAR o TEXT)**: Estos campos se usan para almacenar números, como edades, cantidades o precios. INTEGER se usa para números enteros, mientras que DECIMAL se usa para números con decimales.
- **Número (INTEGER o DECIMAL)**: Estos campos se usan para almacenar números, como edades, cantidades o precios. INTEGER se usa para números enteros, mientras que DECIMAL se usa para números con decimales.
- **Fecha y hora (DATE o TIMESTAMP)**: Estos campos se usan para almacenar fechas y horas. DATE se usa para fechas, mientras que TIMESTAMP se usa para fechas y horas con precisión de segundos.
- **Booleano (BOOLEAN)**: Este campo se usa para almacenar valores verdadero/falso o 0/1. Es útil para campos que solo tienen dos opciones posibles, como "activo/inactivo".
- **BLOB (Binary Large OBject)**: Este campo se usa para almacenar datos binarios, como imágenes o archivos de audio. Los BLOBs son útiles para almacenar grandes cantidades de datos binarios.
- **Enumeración (ENUM)**: Este campo se usa para campos que solo pueden tener un conjunto limitado de valores. Por ejemplo, un campo "género" solo puede tener valores "masculino" o "femenino".

Expresiones y operadores

En las bases de datos, las expresiones son combinaciones de valores, operadores y funciones que se utilizan para realizar cálculos, comparaciones y transformaciones de datos en una consulta SQL. Los operadores son símbolos o palabras clave que se utilizan en estas expresiones para indicar qué operación se debe realizar entre los valores.

- **Operadores aritméticos**: se utilizan para realizar operaciones matemáticas en los valores de las columnas.
  - `+`: suma
  - `-`: resta
  - `*` : multiplicación
  - `/` : división
  - `%` : módulo (resto de la división)
- **Operadores de comparación**: se utilizan para comparar valores en las columnas.
  - `=`: diferente
  - `<>` o `!=`: diferente
  - `<` : menor que
  - `>` : mayor que
  - `<=` : menor o igual que
  - `>=` : mayor o igual que
- **Operadores lógicos**: se utilizan para combinar expresiones lógicas y producir un resultado verdadero o falso
  - `AND` : operador lógico "y" (ambas condiciones deben ser verdaderas)
  - `OR` : operador lógico "o" (al menos una condición debe ser verdadera)
  - `NOT` : operador lógico "no" (invierte el resultado de la expresión)
- **Expresiones de cadena**: se utilizan para realizar operaciones matemáticas en los valores de las columnas. Los operadores aritméticos son los siguientes:

- `||`: concatenación de cadenas (une dos cadenas en una sola)
- `LIKE`: comparación de cadenas (compara una cadena con un patrón)
- `SUBSTR` : extracción de subcadenas (extrae una porción de una cadena)
- `UPPER` : conversión a mayúsculas (convierte una cadena a mayúsculas)
- `LOWER` : conversión a minúsculas (convierte una cadena a minúsculas)

Sistema gestor de base de datos (SGBD o DBMS)

Un sistema gestor de bases de datos (SGBD o DBMS, por sus siglas en inglés) es un software que se utiliza para administrar y gestionar bases de datos. Su función principal es permitir la creación, el acceso, la manipulación y el mantenimiento de grandes cantidades de datos almacenados en una base de datos. Entre las tareas que realiza un SGBD se encuentran:

- Definir la estructura y el esquema de la base de datos
- Controlar el acceso y la seguridad de la base de datos
- Almacenar y recuperar datos de la base de datos
- Permitir la manipulación de los datos mediante operaciones de lectura, escritura y actualización
- Realizar consultas y búsquedas de datos en la base de datos
- Administrar el rendimiento y la optimización de la base de datos
- Realizar copias de seguridad y restauración de la base de datos

Los SGBD pueden ser de diferentes tipos, según su modelo de datos, su arquitectura y su capacidad de procesamiento. Algunos de los SGBD más comunes son MySQL, Oracle, Microsoft SQL Server, PostgreSQL y MongoDB.

MYSQL básicos

A continuación, te presento algunos de los comandos básicos de MySQL:

1. Conexión a la base de datos `mysql -u usuario -p contraseña`
2. Mostrar base de datos `show databases:`
3. Selección de una base de datos `USE nombre_de_la_base_de_datos;`
4. Mostrar tablas de la base de datos: `show tables;`
5. Creación de una tabla

```
CREATE TABLE nombre_de_la_tabla (
columna1 tipo_de_dato,
columna2 tipo_de_dato,
...
);
```

6. Inserción de datos en una tabla `INSERT INTO nombre_de_la_tabla (columna1, columna2, ...) VALUES (valor1, valor2, ...);`
7. Consulta de datos de una tabla `SELECT * FROM nombre_de_la_tabla;`