

Introducción a SQL y MySQL

Base de Datos Relacionales

¡Hola! 🙌 Te damos la bienvenida a base de datos relacionales con MySQL 🙌

En el ámbito empresarial de hoy en día, donde las empresas acumulan una gran cantidad de información, a veces puede ser un desafío poner todas las piezas en su lugar.

Aquí es donde entran en juego las bases de datos relacionales. Son como organizadores especiales que ayudan a colocar cada pieza en su posición correcta y a ver la imagen completa. Piensa en ellas como las estanterías en las que organizas tus libros por categorías para encontrar lo que necesitas de manera más fácil.

En esta guía exploraremos juntos cómo las bases de datos relacionales actúan como organizadores eficientes, cómo ayudan a las empresas a manejar su información y cómo puedes aprender a utilizarlas de manera efectiva. ¡Prepárate para sumergirte en el mundo de las bases de datos relacionales y descubrir un nuevo mundo de posibilidades!

¡Que comience el viaje! 🚀

Introducción

Una base de datos relacional es un tipo de base de datos que organiza los datos en tablas relacionadas entre sí. Cada tabla consta de una serie de filas y columnas, y cada fila de una tabla representa un único registro, mientras que las columnas representan los campos de ese registro.

Las bases de datos relacionales se utilizan por varias razones. En primer lugar, permiten una organización eficiente y flexible de los datos. En segundo lugar,

aseguran la integridad de los datos a través de restricciones y reglas. Por último, facilitan la recuperación y manipulación de datos a través de un lenguaje de consulta estructurado (SQL), lo que permite realizar consultas complejas.

Comprendiendo SQL

SQL, qué significa “Structured Query Language” o en español “Lenguaje de Consulta Estructurado”, es el lenguaje estándar utilizado para interactuar con las bases de datos relacionales.

SQL es importante porque proporciona una forma de acceder y manipular grandes conjuntos de datos de manera eficiente. A diferencia de los lenguajes de programación convencionales como Java o Python, SQL se diseñó específicamente para trabajar con bases de datos, lo que lo hace especialmente adecuado para tareas como la filtración, clasificación y agregación de datos.

Subconjuntos de Lenguajes en SQL

SQL incluye varios sublenguajes, cada uno diseñado para un propósito específico:

DDL (Data Definition Language, o Lenguaje de Definición de Datos): Este subconjunto de SQL se utiliza para definir y modificar la estructura de las tablas y otras estructuras en una base de datos. Los comandos típicos de DDL incluyen **CREATE**, **ALTER** y **DROP**.

DML (Data Manipulation Language, o Lenguaje de Manipulación de Datos): Este subconjunto de SQL se utiliza para manipular los datos dentro de las tablas de una base de datos. Los comandos típicos de DML incluyen **SELECT**, **INSERT**, **UPDATE** y **DELETE**.

DCL (Data Control Language, o Lenguaje de Control de Datos): Este subconjunto de SQL se utiliza para controlar los permisos y el acceso a los datos en una base de datos. Los comandos típicos de DCL incluyen **GRANT** y **REVOKE**.


TCL (Transaction Control Language, o Lenguaje de Control de Transacciones): Este subconjunto de SQL se utiliza para manejar transacciones dentro de una

base de datos para asegurar la integridad de los datos. Los comandos típicos de TCL incluyen COMMIT, ROLLBACK y SAVEPOINT.

¿Qué es un sistema de gestión de base de datos (DBMS)?

Un Sistema de Gestión de Bases de Datos, abreviado como DBMS por sus siglas en inglés (Database Management System), es un software o conjunto de programas que permite a los usuarios interactuar con una base de datos de manera eficiente y segura. Su función principal es administrar y controlar el acceso, la organización, la manipulación y la seguridad de los datos almacenados en una base de datos.

Además, los DBMS relacionales implementan el estándar SQL lo que permite ejecutar los "scripts" escritos en dicho lenguaje.

 Se le dice "script" a todo archivo ejecutable con un conjunto de instrucciones o comandos escritos en un lenguaje de programación o secuencia de comandos que se utiliza para realizar una serie de tareas.

En nuestro caso, el DBMS que utilizaremos se llama MySQL. Es un sistema de gestión de bases de datos de código abierto y muy popular que se utiliza comúnmente en aplicaciones web y empresariales.

Servidor y Base de Datos

Es esencial comprender la distinción entre un servidor y una base de datos.

El servidor MySQL es el administrador principal que controla el acceso y la gestión de una o más bases de datos, mientras que las bases de datos son como carpetas que almacenan y organizan los datos en tablas.

Tablas y sus Características Principales:

Las tablas son elementos esenciales en una base de datos relacional. Aquí están algunas de sus características clave:

- **Filas y Columnas:** Las tablas constan de filas y columnas. Cada fila representa una entrada de datos única, mientras que cada columna describe un atributo específico o campo de datos.
- **Estructura Predefinida:** Antes de insertar datos en una tabla, debes definir su estructura, especificando qué columnas contendrá y qué tipo de datos pueden almacenar.
- **Relaciones:** Las tablas pueden relacionarse entre sí mediante claves primarias y foráneas, lo que permite representar datos interconectados y facilitar la consulta.
- **Integridad de los Datos:** Las bases de datos aplican restricciones y reglas para garantizar la integridad de los datos almacenados en las tablas. Esto ayuda a mantener la precisión y consistencia de la información.

Aquí tienes un ejemplo de una tabla "clientes" de una base de datos:

cliente_id	nombre	apellido	correo_electronico	telefono	fecha_registro
1	Juan	Pérez	juan@example.com	555123456	2023-09-26
2	María	López	maria@example.com	555789012	2023-09-26
3	Carlos	Rodríguez	carlos@example.com	555345678	2023-09-25
4	Ana	González	ana@example.com	555678901	2023-09-25
5	Laura	Martínez	laura@example.com	555234567	2023-09-24

Explicación de las partes de la tabla:

- cliente_id: Esta columna es un identificador único para cada cliente. Se utiliza como clave primaria, lo que significa que cada valor en esta columna es único y se utiliza para identificar de manera exclusiva a cada cliente en la tabla.
- nombre: Esta columna almacena el nombre del cliente.
- apellido: Almacena el apellido del cliente.
- correo_electronico: Aquí se registran las direcciones de correo electrónico de los clientes.
- telefono: Esta columna contiene los números de teléfono de los clientes.
- fecha_registro: Registra la fecha en que cada cliente se registró en el sistema.

Esta tabla de ejemplo representa una entidad llamada "clientes" y contiene información sobre los clientes, como sus nombres, apellidos, información de contacto y fechas de registro. Cada fila en la tabla representa un cliente individual, y cada columna almacena un atributo específico de los clientes.

Sobre la estructura predefinida, las relaciones y la integridad de los datos lo explicaremos en profundidad en las próximas clases.

Sublenguaje DML (Lenguaje de Manipulación de Datos):

El sublenguaje DML se utiliza para conocer el contenido de un servidor y sus bases de datos e interactuar con los datos almacenados en sus tablas. Aquí están algunos comandos DML clave que utilizaremos en la parte práctica:

SHOW DATABASES: Este comando se utiliza para listar todas las bases de datos disponibles en el servidor MySQL. Proporciona información sobre las bases de datos a las que un usuario tiene acceso.

SHOW TABLES: Este comando se utiliza para listar todas las tablas en la base de datos actualmente seleccionada. Muestra las tablas disponibles en la base de datos que está siendo utilizada.

SELECT: Utilizado para recuperar datos de una tabla o conjunto de tablas. Un comando SELECT básico se ve así: `SELECT * FROM nombre_tabla;`. Esto devuelve todas las filas y columnas de la tabla especificada.

INSERT: Agrega nuevos registros a una tabla. Por ejemplo, puedes utilizar `INSERT INTO tabla (columna1, columna2) VALUES (valor1, valor2);` para insertar datos en una tabla específica.

UPDATE: Modifica registros existentes en una tabla. El comando se ve como: `UPDATE nombre_tabla SET columna = nuevo_valor WHERE condición;`. Esto permite actualizar datos en función de ciertos criterios.

DELETE: Elimina registros de una tabla. Por ejemplo, `DELETE FROM nombre_tabla WHERE condición;` borra filas que cumplan con ciertos requisitos.

Estos comandos DML son fundamentales para interactuar con un servidor y sus bases de datos y permiten desde recuperar información hasta modificar y eliminar registros según sea necesario.