

SQL Básico

FreeCode2009

 @freecode2009

 @freecode2009

Contenido

- ¡Bienvenid@s!
- ¿Qué es una base de datos?
- ¿Que es una base de datos Relacional?
- ¿Qué es SQL?
- DDL, DML, DCL
- Tipos de motores de base de datos (MySQL, MariaDB, PostgreSQL)
- Instalación MySQL, MariaDB, PostgreSQL
- Clientes para las bases de datos (DBeaver)
- Tipos de datos
- Join y Left Join
- Cláusulas y operadores
- Ejemplo práctico tienda de productos

¿Qué es una base de datos?

Una base de datos es una colección de información lógicamente relacionada y organizada de manera que pueda ser fácilmente accesible, administrada y actualizada.

Las bases de datos son controladas o gestionadas a través de un motor de base de datos. En este curso utilizaremos MySQL, MariaDB y PostgreSQL.

¿Que es una base de datos relacional?

Una base de datos relacional es un tipo de base de datos que almacena y proporciona acceso a puntos de datos relacionados entre sí.

Estás bases de datos se basan en el modelo relacional, una forma intuitiva y directa de representar datos en ***tablas***.

En una base de datos relacional, cada ***fila*** en una tabla es un registro con una ***ID única***, llamada clave. Las ***columns*** de la tabla contienen los ***atributos*** de los datos y cada registro suele tener un valor para cada atributo, lo que simplifica la creación de relaciones entre los puntos de datos.

¿Qué es SQL?

Significa ***lenguaje de consulta estructurado*** y se utiliza para comunicarse con una base de datos relacional, es decir, para acceder y manipular los datos.

DDL, DML, DCL

Los comandos SQL se clasifican en las siguientes categorías:

- **DDL:** lenguaje de definición de datos. Con estos comandos podemos definir el esquema de la base de datos: ***CREATE, DROP, ALTER, TRUNCATE, COMMENT, RENAME.***
- **DML:** lenguaje de manipulación de datos. Con estos comandos podemos manipular los datos en la base de datos: ***SELECT, INSERT, UPDATE, DELETE.***
- **DCL:** se ocupan de los derechos, permisos y otros controles del sistema de base de datos: ***GRANT, REVOKE.***
- **TCL:** se ocupan de las transacciones dentro de la base de datos: ***COMMIT, ROLLBACK.***

Tipos de motores de base de datos

Las bases de datos son controladas o gestionadas a través de un motor de base de datos. En este curso utilizaremos MySQL, MariaDB y PostgreSQL.



Instalaciones

MySQL: https://www.youtube.com/playlist?list=PLyUE2hSTFifOB-zCXOu6kz4GE_y4C863R

MariaDB: <https://www.youtube.com/playlist?list=PLyUE2hSTFifMxzVzjbT17Q63oQirQBMEz>

PostgreSQL: <https://www.youtube.com/playlist?list=PLyUE2hSTFifNKppcGX0eCjRjlyj0NvMu8>

Cientes para las bases de datos

Para trabajar con el motor de base de datos es necesario contar con un cliente que permita facilitar las tareas que realizamos como:

- Consultar la base de datos
- Construir y ejecuta el código SQL
- Generar un informe
- Realizar una copia de seguridad
- Revisar problemas con algún dato en particular
- Etcétera



Tipos de datos: Numéricos

Tipo de datos	Descripción
<code>BIT</code>	puede almacenar un solo bit (0 o 1) o NULL
<code>TINYINT</code>	puede almacenar números del 0 al 255
<code>SMALLINT</code>	puede almacenar números entre -32,768 y 32,767
<code>INT</code>	puede almacenar números entre -2,147,483,648 y 2,147,483,647
<code>BIGINT</code>	puede almacenar números entre -9,223,372,036,854,775,808 y 9,223,372,036,854,775,807

Tipos de datos: Cadenas

Tipo de datos	Descripción
CHAR(x)	puede almacenar caracteres de longitud fija (máx. 8000 caracteres)
VARCHAR(x)	puede almacenar caracteres hasta una longitud dada (máx. 8000 caracteres)
TEXT	puede almacenar caracteres de hasta 2 GB de tamaño
IMAGE	puede almacenar cadenas binarias de hasta 2 GB de tamaño

Tipos de datos: Fecha y Hora

Tipo de datos	Descripción
<code>DATETIME</code>	puede almacenar la fecha del 1 de enero de 1753 al 31 de diciembre de 9999 con el tiempo
<code>DATETIME2</code>	puede almacenar la fecha del 1 de enero de 0001 al 31 de diciembre de 9999 con el tiempo
<code>DATE</code>	puede almacenar solo la fecha del 1 de enero de 0001 al 31 de diciembre de 9999
<code>TIME</code>	solo puede almacenar el tiempo

Ejemplo práctico: Tienda de productos en línea

La empresa *Celulares SAS* necesita realizar un ecommerce para vender sus productos a través de una aplicación web que les permita abarcar fidelizar a sus clientes y darse a conocer a nuevos clientes. Por esta razón contrata a **@freecode2009** para que diseñe en primera instancia la base de datos que soportará toda la aplicación. La aplicación deberá contar con los siguientes módulos:

1. Creación de productos. Los productos deberán tener definido: nombre, categoría, marca, precio, inventario y fecha de creación.
2. Clientes. Este módulo se encargará de registrar los clientes para los cuales es necesario registrar la siguiente información: nombre, nit (único por cada cliente), fecha nacimiento, tipo (natural o jurídico), teléfono, dirección y fecha de creación.
3. Ventas. Debe poder registrar las ventas por cada cliente que realiza una compra.

Cláusulas

Las cláusulas son condiciones de modificaciones utilizadas para definir los datos que desea seleccionar o manipular:

Comando	Descripción
FROM	Utilizada para especificar la tabla de la cual se van a seleccionar los registros
GROUP BY	Utilizada para separar los registros seleccionados en grupos específicos
HAVING	Utilizada para expresar condición que debe satisfacer cada grupo
ORDER BY	Utilizada para ordenar los registros seleccionados de acuerdo con un orden específico
WHERE	Utilizada para determinar los registros seleccionados en la cláusula FROM

Operadores Lógicos

Operador	Uso
AND	Es el "y" lógico. Evalúa dos condiciones y devuelve un valor de verdad sólo si ambas son ciertas.
OR	Es el "o" lógico. Evalúa dos condiciones y devuelve un valor de verdad si alguna de las dos es cierta.
NOT	Negación lógica. Devuelve el valor contrario de la expresión.

Operadores de comparación

Operador	Uso
<	Menor que
>	Mayor que
<>	Distinto de
<=	Menor o igual que
>=	Mayor o igual que
BETWEEN	Intervalo
LIKE	Comparación
In	Especificar

INNER JOIN

La sentencia SQL JOIN se utiliza para relacionar varias tablas. Nos permitirá obtener un listado de campos que tienen coincidencias en ambas tablas.

```
-- mostrar todos los productos con su respectivo nombre de categoria y marca para
select p.producto_id, p.nombre nombre_producto, p.categoria_id, c.nombre nombre_categoria,
p.marca_id, m.nombre nombre_marca, p.precio, p.inventario, p.fecha_creacion
from productos p
inner join categorias c on p.categoria_id = c.categoria_id
inner join marcas m on p.marca_id = m.marca_id;
```

productos(+) 1 x									
select p.producto_id, p.nombre nombre_producto, p.categoria_id, c.nombre nombre_categoria, m.nombre nombre_marca, p.precio, p.inventario, p.fecha_creacion									
	producto_id	nombre_producto	categoria_id	nombre_categoria	marca_id	nombre_marca	precio	inventario	fecha_creacion
1	1	Xiaomi Redmi Note 11 Dual SIM	2	[NULL]	1	Xiaomi	759.900	20	2022-10-31 15:44:50
2	2	Xiaomi Redmi 10 Dual SIM 32 GB	2	[NULL]	1	Xiaomi	950.000	10	2022-10-31 15:44:50
3	3	Samsung Galaxy A32	1	Usados	5	Samsung	889.900	20	2022-10-31 15:44:50
4	4	Moto G31 Dual SIM 128 GB	1	Usados	6	Motorola	889.900	20	2022-10-31 15:44:50
5	6	Huawei Y9a Dual SIM 128 GB	1	Usados	2	Huawei	849.900	6	2022-10-31 15:47:31
6	8	Xiaomi Redmi Note 11S Dual SIM 128 GB	1	Usados	1	Xiaomi	907.900	2	2022-10-31 15:48:46

LEFT JOIN

La sentencia LEFT JOIN nos dará el resultado anterior más los campos de la tabla de la izquierda del JOIN que no tienen coincidencias en la tabla de la derecha.

```
-- mostrar todos los productos con su respectivo nombre de categoria y marca para
-- donde la categoria sea igual a 1, pero mostrar tambien los que no coincidan
select p.producto_id, p.nombre nombre_producto, p.categoria_id, c.nombre nombre_categoria,
p.marca_id, m.nombre nombre_marca, p.precio, p.inventario, p.fecha_creacion
from productos p
left join categorias c on p.categoria_id = c.categoria_id and p.categoria_id = 1
inner join marcas m on p.marca_id = m.marca_id;
```

productos(+) 1 x

select p.producto_id, p.nombre nombre_producto, p.categoria_id, Enter a SQL expression to filter results (use Ctrl+Space)

	producto_id	nombre_producto	categoria_id	nombre_categoria	marca_id	nombre_marca	precio	inventario	fecha_creacion	
1	1	Xiaomi Redmi Note 11 Dual SIM	2	[NULL]	1	Xiaomi	759.900	20	2022-10-31 15:44:50	✓
2	2	Xiaomi Redmi 10 Dual SIM 32 GB	2	[NULL]	1	Xiaomi	950.000	10	2022-10-31 15:44:50	✓
3	3	Samsung Galaxy A32	1	Usados	5	Samsung	889.900	20	2022-10-31 15:44:50	
4	4	Moto G31 Dual SIM 128 GB	1	Usados	6	Motorola	889.900	20	2022-10-31 15:44:50	
5	6	Huawei Y9a Dual SIM 128 GB	1	Usados	2	Huawei	849.900	6	2022-10-31 15:47:31	
6	8	Xiaomi Redmi Note 11S Dual SIM 128 GB	1	Usados	1	Xiaomi	907.900	2	2022-10-31 15:48:46	

Referencias

- <https://corporatefinanceinstitute.com/resources/knowledge/data-analysis/database/>
- <https://www.oracle.com/co/database/what-is-a-relational-database/>
- <https://geekflare.com/es/multi-platform-sql-client/>
- https://postgis.readthedocs.io/es/latest/conceptos-sql/conceptos_sql.html
- <https://www.programiz.com/sql/data-types>
- <https://www.in-nomine.org/~asmodai/mysql-to-pgsql.html>