

GUIA PRACTICA DE SQL UTILIZANDO MYSQL WORKBENCH



Autor: Ing. Ochoa Enciso, José Luis

2023



Tabla de contenido

1. Sistema Gestor de Base de Datos (SGBD)	3
2. Introducción a MySQL	3
2.1. Características:	4
2.2. Instalación	4
3. Crear una base de datos	7
3.1. Base de datos	7
3.2. Datos estructurados	7
3.3. Lenguaje SQL	8
4. Tablas y tipos de datos	8
4.1. Tablas	8
4.1.1. Campos:	9
4.1.2. Registro:	9
4.1.3. Dato:	9
4.2. Tipos de datos	9
4.3. Atributos de los campos	11
5. Llaves en BBDD relacionales	11
5.1. Primary Key (Llave primaria)	12
5.2. Foreign Key (Llave foránea)	12
6. Sentencias SQL	12
6.1. Lenguaje de Definición de Datos (DDL)	13
6.2. Lenguaje de Manipulación de Datos (DML)	13
6.3. Lenguaje de Control de Datos (DCL)	14
6.4. Lenguaje de Control de Transacción (TCL)	14
7. Anexos	15
7.1. Normalización de Bases de Datos	15

1. Sistema Gestor de Base de Datos (SGBD)

Un Sistema Gestor de Base de Datos (SGBD) o DataBase Managenent System (DBMS) es un sistema que permite la creación, gestión y administración de bases de datos, así como la elección y manejo de las estructuras necesarias para el almacenamiento y búsqueda de información del modo más eficiente posible.



En la actualidad, existen multitud de SGBD y pueden ser clasificados según la forma en que administran los datos en:

1.1. SGBD Relacionales (SQL)

Este modelo se basa fundamentalmente en establecer relaciones o vínculos entre los datos, imaginando una tabla aparte por cada relación existente con sus propios registros y atributos.

1.2. SGBD No relacionales (NoSQL)

Una base de datos no relacional (NoSQL) es aquella base de datos que:

- No requiere de estructuras de datos fijas como tablas
- No garantiza completamente las características ACID.
- Escala muy bien horizontalmente.

Se utilizan en entornos distribuidos que han de estar siempre disponibles y operativos y que gestionan un importante volumen de datos.

2. Introducción a MySQL

MySQL es un sistema de gestión de bases de datos relacionales (RDBMS) de código abierto respaldado por Oracle y basado en el lenguaje de consulta estructurado (SQL). MySQL funciona prácticamente en todas las plataformas, incluyendo Linux, UNIX y Windows. Aunque puede utilizarse en una amplia gama de aplicaciones, MySQL se asocia más a menudo con las aplicaciones web y la publicación en línea.



✓ ¿Cómo funciona?

MySQL se basa en un modelo cliente-servidor. El núcleo de MySQL es el servidor MySQL, que maneja todas las instrucciones (o comandos) de la base de datos. El servidor MySQL está disponible como un programa independiente para su uso en un entorno de red cliente-servidor y como una biblioteca que puede ser incrustada (o enlazada) en aplicaciones independientes.

2.1. Características:

- Velocidad y rendimiento.
- Bajo consumo de recursos
- Herramientas de administración
- Portabilidad y mantenimiento de datos
- Cuenta con paquetes de instalación como XAMPP (Apache + PHP + MySQL).

2.2. Instalación

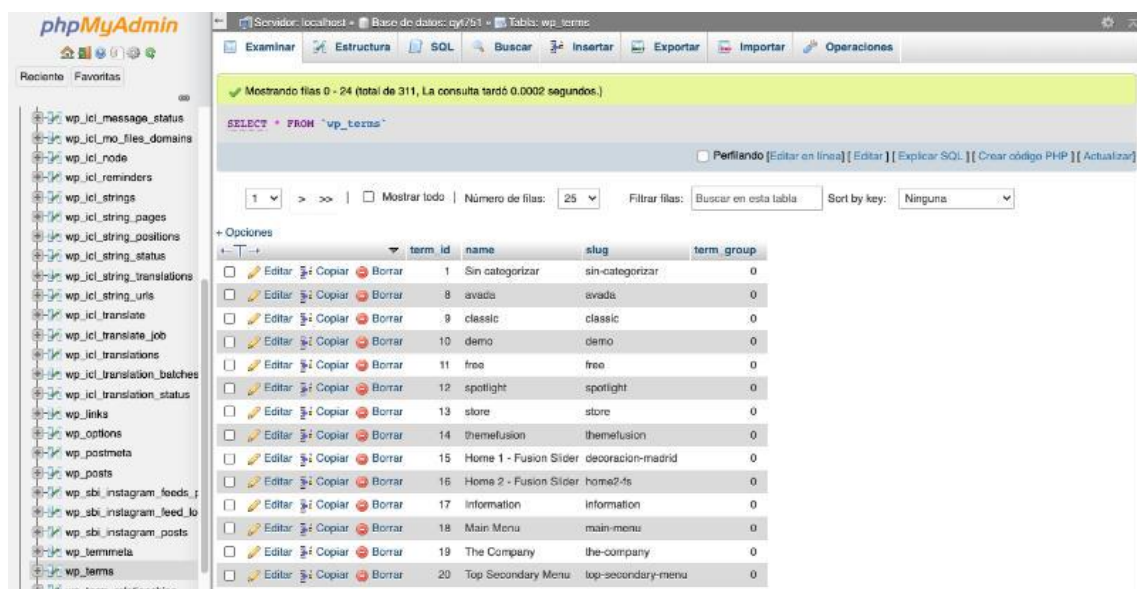
▪ XAMPP

Una de las formas clásicas de poder trabajar con MySQL como gestor de base de datos es instalando XAMPP, el cual es una distribución de Apache completamente gratuita y fácil de instalar que contiene MariaDB, PHP y Perl. El paquete de instalación de XAMPP ha sido diseñado para ser increíblemente fácil de instalar y usar.

Link: <https://www.apachefriends.org/es/index.html>



Una vez instalada XAMPP puedes ingresar al localhost (previamente el puerto debe ser activado en Apache) para poder empezar a trabajar con MySQL.

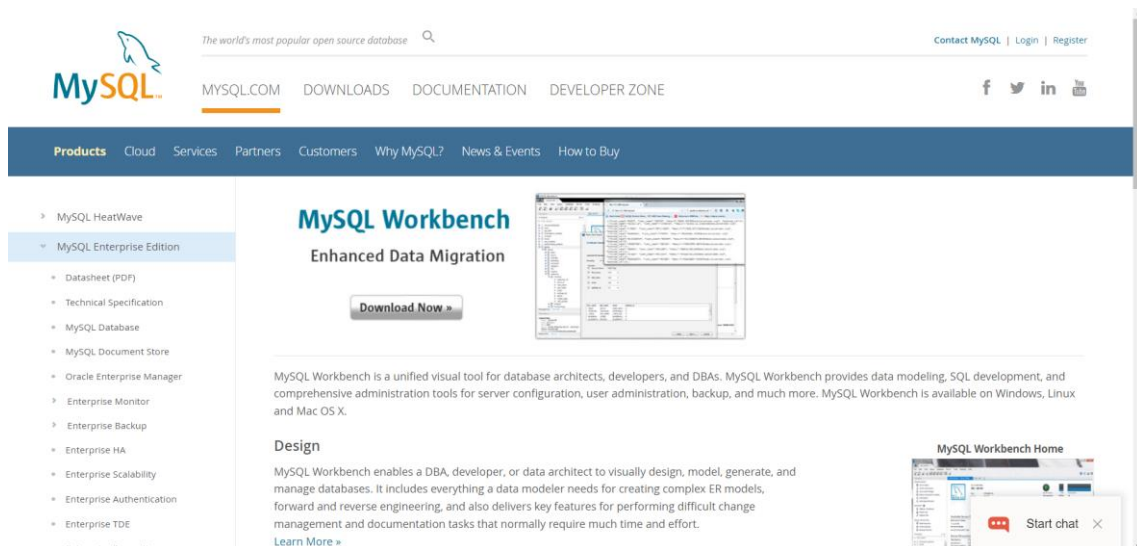


MySQL Workbench

El software MySQL Workbench es un entorno de MySQL gráfico de diseño de bases de datos, servidores, administración y mantenimiento para el sistema MySQL. Además, esta herramienta gráfica fue desarrollada y distribuida por la compañía de desarrollo de nube y locales Oracle Corporation. Para el desarrollo de esta guía utilizaremos este software.

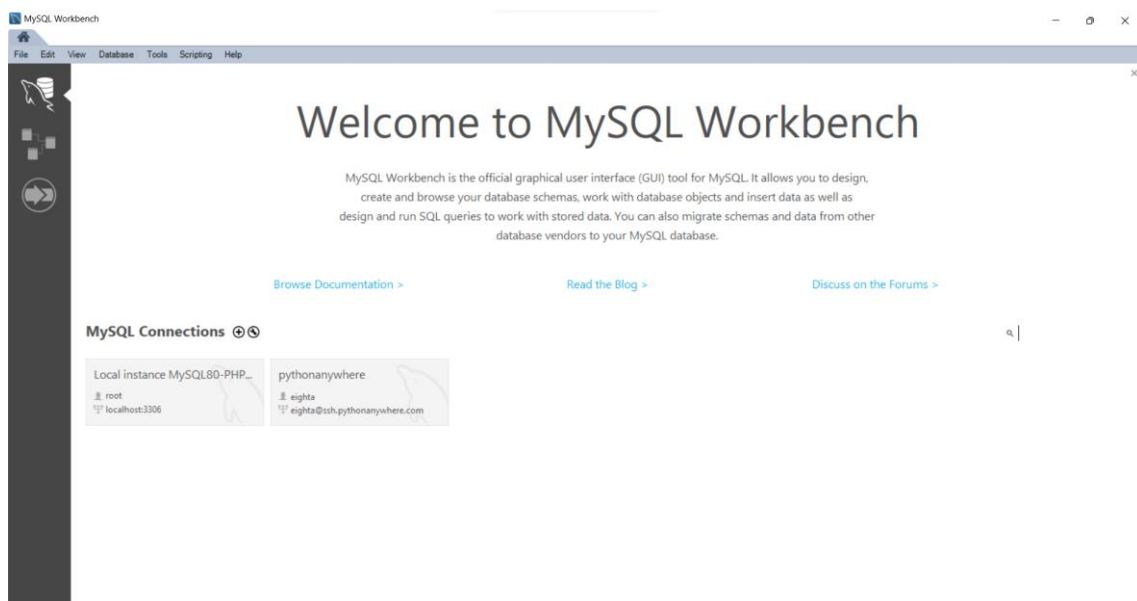
Recomendación: Estar registrado en la plataforma de Oracle, para poder descargar el software a usar.

Link: <https://www.mysql.com/products/workbench/>

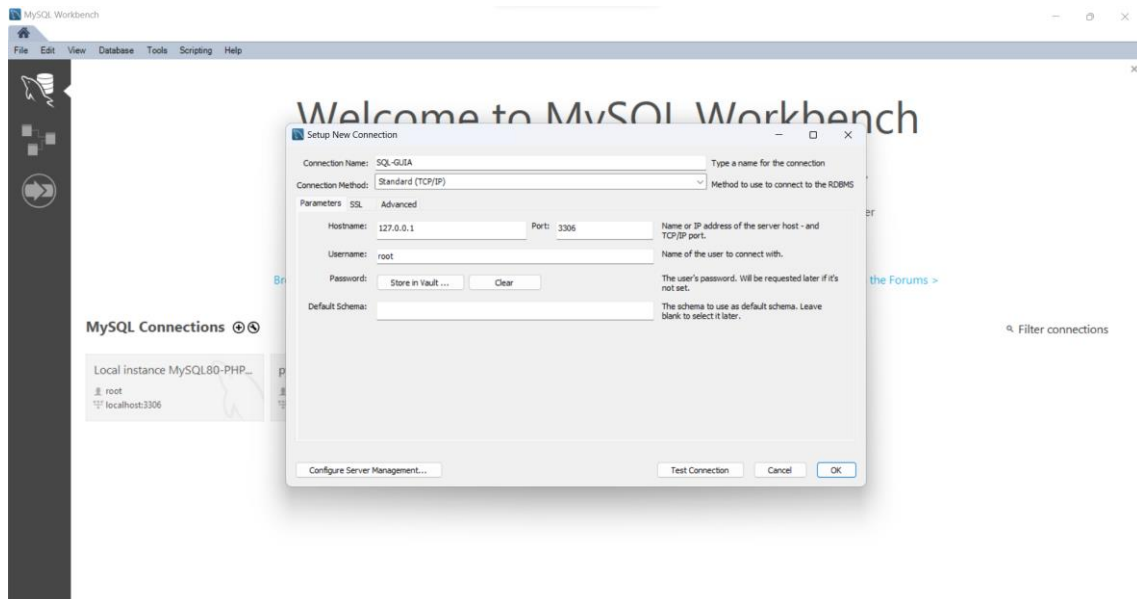


A continuación, comparto una guía para la instalación de MySQL Community Server.

Link: http://www.factureya.com/web/descargas/MANUAL_MYSQL_FY.pdf



Una vez logremos instalar MySQL Workbench podemos crear nuestra primera conexión, para así empezar a crear nuestra primera base de datos.



3. Crear una base de datos

Antes de continuar, con la creación de una base de datos definamos algunos conceptos.

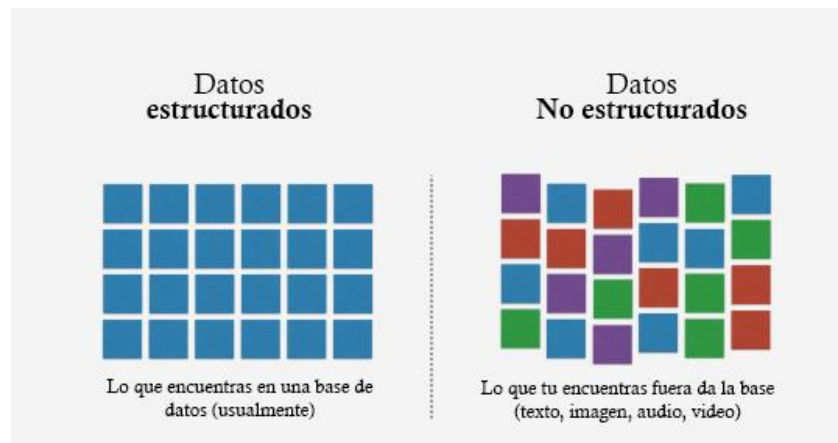
3.1. Base de datos

Una base de datos es simplemente una colección de datos estructurados. Imagina que tomas una foto: presionas un botón y capturas una imagen de ti mismo. Tu foto es información y la galería de tu teléfono es la base de datos. Una base de datos es un lugar en el que los datos son almacenados y organizados. La palabra «relacional» significa que los datos almacenados en el conjunto de datos son organizados en forma de tablas. Cada tabla se relaciona de alguna manera definida de acuerdo a la interacción de la información.



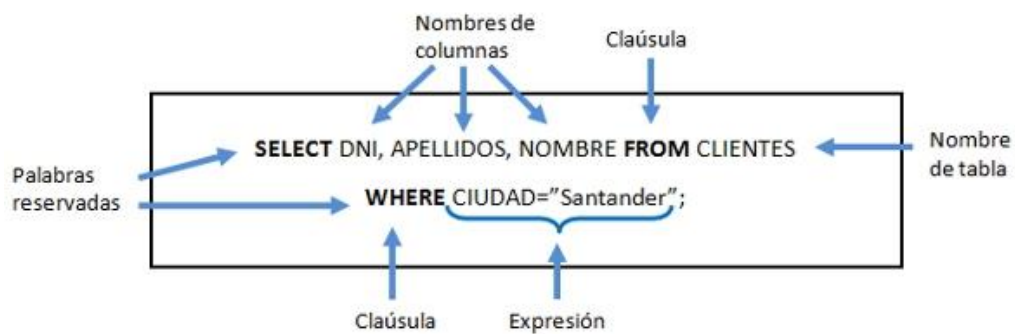
3.2. Datos estructurados

Es aquella información que se almacena en forma de bases de datos relacionales, generalmente conocidas como bases de datos SQL.



3.3. Lenguaje SQL

Es el lenguaje de consulta, usado como interfaz para comunicarse con bases de datos y realizar operaciones de acceso y manipulación de la información almacenada.



Ahora que ya conocemos estos conceptos, crearemos nuestra primera base de datos.

```
CREATE DATABASE MYDATABASE;
```

```
SHOW DATABASES;
```

```
USE MYDATABASE;
```

```
DROP DATABASE <name>
```

4. Tablas y tipos de datos

4.1. Tablas

Las tablas son como contenedores para la base de datos, para dividirla en partes más pequeñas con el fin de almacenar la información de manera separada y más organizada.

Tabla: Persona

CAMPOS

ID	Nombre	Apellido	Teléfono	Dirección
1	Nicolás	Morales	6956315439	Plaza Víctor, 938, Bajos
2	Sandra	Padrón	4634665466	Avenida Noelia, 59
3	Beatriz	Pacheco	3534768456	Camino real 74, 4º 2ª

DATO

REGISTRO **TABLA**

4.1.1. Campos:

Dentro de cada tabla, tenemos la información dividida en columnas, también llamadas campos. Esto representa el tipo información contendrá la tabla. Por ejemplo, *Nombre*.

4.1.2. Registro:

Son cada una de las filas que contienen datos. Una fila, un registro. Los registros, como verás están sujetos a los campos, cada registro, completa información sobre cada campo. Por ejemplo, 2 – Sandra – Padrón – 4634665466 – Avenida Noelia, 59.

4.1.3. Dato:

Cada porción de información en un registro es un dato. Por ejemplo, *Morales*.

Un punto importante a revisar, son los tipos de datos soportados en nuestro SGBD para poder definir el campo de una tabla.

4.2. Tipos de datos

Por la naturaleza de cada dato, en MySQL podemos definir cinco grandes grupos al momento de tipificar los datos que se van a almacenar en nuestra base. Cada uno de ellos contiene subtipos que especializan el tipo de datos para definirlo completa y unívocamente.

4.2.1. Numérico

INT	Permite números desde -2147483648 hasta 2147483647. Si se define como UNSIGNED (sin signo) permite números desde 0 hasta 4294967295.
BEGINT	Permite números desde -9223372036854775808 hasta 9223372036854775807. Si se define como UNSIGNED (sin signo) permite números desde 0 hasta 18446744073709551615.
BOOL	Un número entero que puede ser 0 ó 1. Valores que se utilizan para representar los estados True o False.
Para los siguientes tipos se puede especificar el número máximo de dígitos y el número de decimales (max. Dígitos, max. Decimales).	
FLOAT	Permite almacenar números decimales de punto flotante pequeños, por lo que sus cálculos son aproximados. Tiene una precisión de 7 dígitos.
DOUBLE	Permite almacenar números decimales de punto flotante grandes, por lo que sus cálculos son aproximados. Un valor del tipo double tiene una precisión entre 15-16 dígitos

DECIMAL	Permite almacenar números decimales de punto fijo grandes, por lo que sus cálculos son exactos. Un valor del tipo decimal tiene una precisión de 28-29 dígitos.
----------------	---

4.2.2. Fecha y marca temporal

DATE	Permite almacenar una fecha con el formato YYYY-MM-DD (4 dígitos para el año, 2 dígitos para el mes y 2 dígitos para el día).
DATETIME	Permite almacenar fecha y hora con el formato YYYY-MM-DD HH:MM:SS (4 dígitos para el año, 2 dígitos para el mes, 2 dígitos para el día, 2 dígitos para las horas, 2 dígitos para los minutos y 2 dígitos para los segundos). <i>El rango soportado es de '1000-01-01 00:00:00' a '9999-12-31 23:59:59'.</i>
TIMESTAMP	Permite almacenar fecha y hora con el formato YYYY-MM-DD HH:MM:SS (4 dígitos para el año, 2 dígitos para el mes, 2 dígitos para el día, 2 dígitos para las horas, 2 dígitos para los minutos y 2 dígitos para los segundos). <i>El rango soportado es de '1970-01-01 00:00:01' UTC a '2038-01-19 03:14:07' UTC.</i>
TIME	Permite almacenar la hora con el formato HH:MM:SS (2 dígitos para las horas, 2 dígitos para los minutos y 2 dígitos para los segundos).
YEAR	Permite almacenar el año con el formato YYYY (4 dígitos para los años).

4.2.3. Cadena

CHAR	Permite almacenar una cadena de datos con longitud fija. Siempre reservará espacio para la longitud definida aunque no se utilice. La longitud máxima es de 255.
VARCHAR	Permite almacenar una cadena de datos (caracteres, números y caracteres especiales) con longitud variable. No reserva el espacio de la longitud máxima definida, ya que ocupa espacio del tamaño real de los datos. La longitud máxima es de 255.
TINYTEXT	Permite almacenar una cadena de datos (solo caracteres; no admite números ni caracteres especiales) de una longitud máxima de 255 caracteres.
TEXT	Permite almacenar una cadena de caracteres de longitud máxima de 65,535.
BLOB	Es un campo que guarda información en formato binario y se utiliza cuando desde PHP se almacena en la base de datos el contenido de un archivo binario (típicamente, una imagen o un archivo comprimido ZIP).
ENUM	Su nombre es la abreviatura de "enumeración". Este campo nos permite establecer cuáles serán los valores posibles que se le podrán insertar.
SET	Su nombre significa "conjunto". De la misma manera que ENUM, debemos especificar una lista, pero de hasta 64 opciones solamente. La carga de esos valores es idéntica a la de ENUM, una lista de valores entre comillas simples, separados por comas. Pero, a diferencia de ENUM, sí podemos llegar a dejarlo vacío, sin elegir ninguna opción de las posibles.

4.3. Atributos de los campos

Los campos pueden poseer ciertos modificadores o "condiciones" que se pueden especificar al crear el campo, y que nos brindan la posibilidad de controlar con mayor exactitud qué se podrá almacenar en ese campo, cómo lo almacenaremos y otros detalles.

NULL – NOT NULL	Algunas veces tendremos la necesidad de tener que agregar registros sin que los valores de todos sus campos sean completados, es decir, dejando algunos campos vacíos (al menos provisoriamente).
VALOR PREDETERMINADO (DEFAULT)	Podemos establecer valores por defecto para los campos cuando creamos la tabla, esta es una muy buena decisión para evitar incompatibilidades entre distintas versiones de MySQL. Para ello utilizamos "default" al definir el campo.
BINARY	Definir un campo de texto CHAR o VARCHAR como BINARY (binario) sólo afecta en el ordenamiento de los datos: en vez de ser indiferentes a mayúsculas y minúsculas, un campo BINARY se ordenará teniendo en cuenta esta diferencia, por lo cual, a igualdad de letra, primero aparecerán los datos que contengan esa letra en mayúsculas.
INDICES	El objetivo de crear un índice es mantener ordenados los registros por aquellos campos que sean frecuentemente utilizados en búsquedas, para así agilizar los tiempos de respuesta.
PRIMARY KEY Y AUTO_INCREMENT	Siempre, en toda la tabla, uno de los campos (por convención, el primero, y también por convención usualmente llamado id –por "identificador"-), debe ser de definido como clave primaria o Primary Key. Otro atributo típico, que es Auto_Increment, es decir, que no nos preocupamos por darle valor a ese campo: al agregar un registro.
UNIQUE	Si especificamos que el valor de un campo sea Unique, estaremos obligando a que su valor no pueda repetirse en más de un registro, pero no por eso el campo se considerará clave primaria de cada registro.

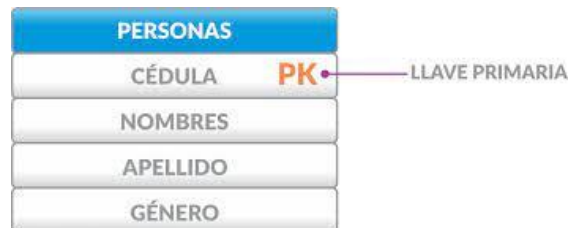
Ahora que ya conocemos estos conceptos acerca de las tablas, podemos empezar a crear nuestra primera tabla.

5. Llaves en BBDD relacionales

Las llaves son elementos dentro de una tabla que permiten identificar de manera única una entidad de un conjunto de entidades. En el caso de las bases de datos relacionales, las llaves primarias y foráneas permiten entrelazar las distintas tablas. A continuación, revisaremos los conceptos.

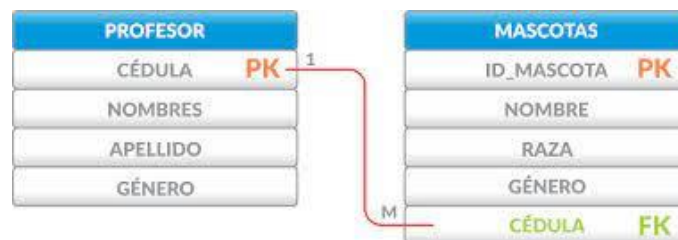
5.1. Primary Key (Llave primaria)

Una llave 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 llaves primarias permitiendo que haya sólo una fila con un valor de clave primaria específico en una tabla.



5.2. Foreign Key (Llave foránea)

Una llave foránea es una columna o un conjunto de columnas en una tabla cuyos valores corresponden a los valores de la clave primaria de otra tabla. Para poder añadir una fila con un valor de llave foránea específico, debe existir una fila en la tabla relacionada con el mismo valor de clave primaria.

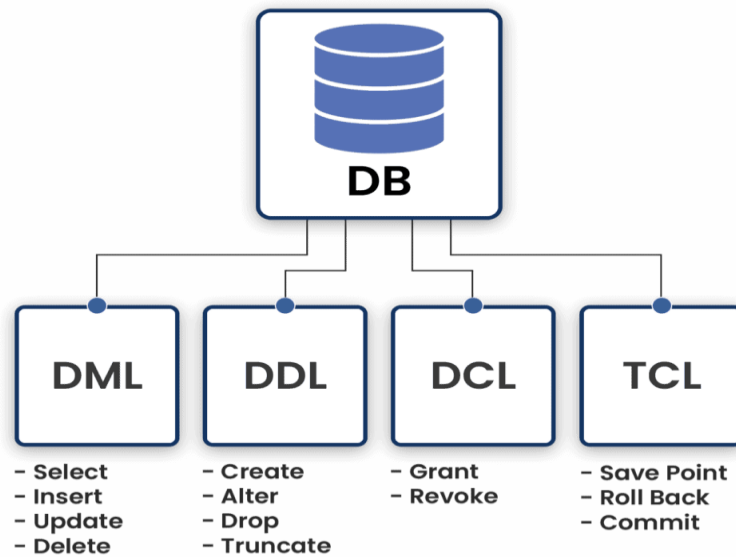


Nota: En los anexos podrás encontrar teoría y ejemplos sobre normalización y modelamiento de datos.

¡Ahora sí!, a realizar nuestra primera relación entre tablas en MySQL.

6. Sentencias SQL

Analizaremos los diferentes conjuntos de lenguaje SQL. Analizaremos las diferencias entre DDL, DML, DCL y TCL.



6.1. Lenguaje de Definición de Datos (DDL)

Se utilizan para definir la estructura de base de datos o esquema.

Es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de definición de las estructuras que almacenarán los datos, así como de los procedimientos o funciones que permitan consultarlos.

Un Data Definition Language o *Lenguaje de descripción de datos (DDL)* es un lenguaje de programación para definir estructuras de datos.

CREATE	Para crear objetos en la base de datos.
ALTER	Altera la estructura de la base de datos.
DROP	Elimina los objetos de la base de datos.
TRUNCATE	Eliminar todos los registros de una tabla, incluyendo todos los espacios asignados a los registros se eliminan.
COMMENT	Agregar comentarios al diccionario de datos.
RENAME	Cambiar el nombre de un objeto.

6.2. Lenguaje de Manipulación de Datos (DML)

Se utilizan para la gestión de datos dentro de los objetos de esquema.

Es un idioma proporcionado por los sistemas gestores de bases de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o modificación de los datos contenidos en las Bases de Datos del Sistema Gestor de Bases de Datos. El lenguaje de manipulación de datos o *Data manipulation Language (DML)* más popular hoy día es SQL, usado para recuperar y manipular datos en una base de datos relacional.

SELECT	Recuperar datos de la base de datos.
INSERT	Insertar datos en una tabla.
UPDATE	Actualizaciones de datos existentes en una tabla.

DELETE	Elimina todos los registros de una tabla.
MERGE	Operación upsert (inserción o actualización).
CALL	Llama a una query o subprograma.
EXPLAIN PLAN	Explicar la ruta de acceso a los datos.
LOCK TABLE	Concurrencias de control.

6.3. Lenguaje de Control de Datos (DCL)

Un Lenguaje de Control de Datos (DCL por sus siglas en inglés: *Data Control Language*) es un lenguaje proporcionado por el Sistema de Gestión de Base de Datos que incluye una serie de comandos SQL que permiten al administrador controlar el acceso a los datos contenidos en la Base de Datos.

GRANT	Permite dar permisos a uno o varios usuarios o roles para realizar tareas determinadas.
REVOKE	Permite eliminar permisos que previamente se han concedido con GRANT.

Las consultas sobre las que se pueden conceder o denegar permisos son las siguientes: CONNECT, SELECT, INSERT, UPDATE, DELETE, USAGE.

6.4. Lenguaje de Control de Transacción (TCL)

Es un lenguaje de programación y un subconjunto de SQL, que se utiliza para controlar el procesamiento de transacciones en una base de datos.

Una transacción es una unidad lógica de trabajo que comprende una o más sentencias SQL, por lo general un grupo de *Data Manipulation Language* (DML).

COMMIT	Guarda el trabajo realizado
SAVEPOINT	Identifica un punto en una transacción a la que más tarde se puede volver.
ROLLBACK	Restaurar la base de datos a la original, hasta el último commit.
SET TRANSACTION	Cambia las opciones de transacción como nivel de aislamiento y qué segmento de cancelación utiliza.

Referencias.

Claves Foráneas: <https://www.ibm.com/docs/es/ida/9.1.2?topic=entities-primary-foreign-keys>

Sentencias SQL : <http://www.tierradelazaro.com/wp-content/uploads/2016/12/DDL-DML-DCL-TCL.pdf>

Tipo de dato: <https://blog.hubspot.es/website/tipos-de-datos-mysql>

Atributos de los campos: <https://disenowebakus.net/tipos-de-datos-mysql.php>

Normalización. <https://explodat.cl/Analytics/desarrollo-de-software/normalizacion-de-bases-de-datos/>

7. Anexos

7.1. Normalización de Bases de Datos

La normalización es un proceso clave para diseñar bases de datos relacionales. Usualmente consiste en aplicar una serie de reglas (formas normales) para convertir un modelo entidad-relación a un modelo relacional.

Gracias a esto se pueden prevenir errores y mejorar la eficiencia de consultas.

La normalización tiene como objetivo optimizar los datos y brindar integridad, los beneficios específicos son:

- Evitar la redundancia de los datos.
- Prevenir problemas de actualización.
- Proteger la integridad de los datos.
- Facilitar el acceso e interpretación de los datos.
- Reducir el tiempo y complejidad de revisión de las bases de datos.
- Optimizar el espacio de almacenamiento.
- Prevenir borrados indeseados.

El proceso consta de etapas bien ordenadas llamadas formas normales (FN), se parte en la primera forma normal (1FN) pudiendo llegar hasta la sexta (6FN), aunque lo tradicional es llegar hasta la tercera forma normal.

Para realizar nuestro ejemplo de normalización, utilizaremos datos de un almacén.

ALMACEN

Modelo	Fabricacion	Memoria	Cantidad
Iphone 13 -Apple	China	128gb	20
Samsung Galaxy S23	Corea del sur	256 gb	30
Huawei P30	China	256gb	15
Xiaomi Redmi Note 11	China	128gb	22
Iphone 13 -Apple	USA	256gb	18
Xiaomi Poco X5	China	128gb	13

7.1.1. Primera Forma Normal 1FN

Una tabla está en primera forma normal si y sólo si:

- No existen filas repetidas.
- Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son simples e indivisibles.

id	Modelo	Marca	Fabricacion	Memoria	Cantidad
1	Iphone 13	Apple	USA	128gb	20
2	Galaxy S23	Samsung	Corea del sur	256 gb	30
3	P30	Huawei	China	256gb	15
4	Redmi Note 11	Xiaomi	China	128gb	22
5	Iphone 13	Apple	USA	256gb	18
6	Poco X5	Xiaomi	China	128gb	13

7.1.2. Segunda Forma Normal 2FN

Una tabla está en segunda forma normal si y sólo si:

- Cumple con las reglas de 1FN.
- Todos los atributos que no forman parte de la Clave Principal tienen dependencia funcional completa de ella.
- Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros.
- Relacionar las tablas mediante claves externas.

Modelo

id	Modelo	idMarca
1	Iphone 13	1
2	Galaxy S23	2
3	P30	3
4	Redmi Note 11	4
5	Poco X5	4

Marca

id	Marca	Fabricacion
1	Apple	USA
2	Samsung	Corea del sur
3	Huawei	China
4	Xiaomi	China

Memoria

id	Memoria
1	128gb
2	256 gb
3	512 gb

Modelo - cantidad

id	idModelo	idMemoria	Cantidad
1	1	1	20
2	2	2	30
3	3	2	15
4	4	1	22
5	1	2	18
6	5	1	13

7.1.3. Tercera Forma Normal 3FN

Una tabla está en tercera forma normal si y sólo si:

- Cumple con las reglas de 2FN.
- No existen dependencias transitivas (eliminar columnas que no dependen de la clave principal).

id	Modelo	idMarca
1	Iphone 13	1
2	Galaxy S23	2
3	P30	3
4	Redmi Note 11	4
5	Poco X5	4

Memoria

id	Memoria
1	128gb
2	256 gb
3	512 gb

Marca

id	Marca	idFabricacion
1	Apple	1
2	Samsung	2
3	Huawei	3
4	Xiaomi	3

Fabricacion

id	Pais
1	USA
2	Corea del sur
3	China

Modelo - cantidad

id	idModelo	idMemoria	Cantidad
1	1	1	20
2	2	2	30
3	3	2	15
4	4	1	22
5	1	2	18
6	5	1	13

Modelamiento de datos.

<https://bookdown.org/paranedagarcia/database/modelamiento-de-datos.html>

<https://www.tecnologias-informacion.com/modelos-datos.html>