Tomar alguna de las tablas (TablaX, TablaY, TablaZ) de la base de datos bd\_grupo\_xyz y realizar las siguientes operaciones.

1. Mostrar **todos los registros** de la TablaX

* INSTRUCCIÓN SQL:

SELECT \* FROM TablaY

* Explicación:

Muestra los registros de la tabla

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. **Mostrar el valor solo de 2 columnas** de todos los registros de la TablaX

* INSTRUCCIÓN SQL:

SELECT nombre, correo\_electronico FROM Usuario;

* Explicación:

Muestra los valores de las columnas nombre y correo\_electronico de todos los registros de la tabla Usuario

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Mostrar el valor de 2 columnas de todos los registros de la TablaX, pero en el resultado de la consulta el **nombre de esas dos columnas debe ser** Columna1, Column2 respectivamente.

* INSTRUCCIÓN SQL:

Muestra los valores de las columnas nombre y correo\_electronico de la tabla Usuario, con nombres Column1 y Column2

* Explicación:

SELECT nombre AS Columna1, correo\_electronico AS Column2 FROM Usuario;

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Mostrar el **número de registros** almacenados en la TablaX

* INSTRUCCIÓN SQL:

SELECT COUNT(\*) FROM Usuario;

* Explicación:

Muestra el número de registros almacenados en la tabla Usuario

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Mostrar todos los registros de la TablaX **ordenados descendentemente**

* INSTRUCCIÓN SQL:

SELECT \* FROM Articulo ORDER BY precio\_inicial DESC;

* Explicación:

Muestra todos los registros de la tabla Articulo ordenados descendentemente por precio\_inicial

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Mostrar el **número de registros** almacenados en la TablaX, pero el valor del resultado debe estar almacenado en una columna llamada EL\_CONTEO.

* INSTRUCCIÓN SQL:

SELECT COUNT(\*) AS EL\_CONTEO FROM Articulo;

* Explicación:

Muestra el número de registros almacenados en la tabla Articulo, con el alias EL\_CONTEO

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta en la TabalX o la Tabla Y o la TablaZ en la que se pueda obtener una **sumatoria** de valores de alguna columna numérica, pero el valor del resultado debe estar almacenado en una columna llamada LA\_SUMATORIA.

* INSTRUCCIÓN SQL:

SELECT SUM(precio\_inicial) AS LA\_SUMATORIA FROM Articulo;

* Explicación:

Obtiene una sumatoria de los precios iniciales de los artículos, con el alias LA\_SUMATORIA.

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ en la que se pueda obtener el **promedio** de valores de alguna columna numérica, pero el valor del resultado debe estar almacenado en una columna llamada El\_PROMEDIO.

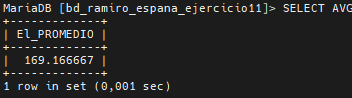
* INSTRUCCIÓN SQL:

SELECT AVG(precio\_inicial) AS El\_PROMEDIO FROM Articulo;

* Explicación:

Obtiene el promedio de los precios iniciales de los artículos, con el alias El\_PROMEDIO

* Recorte de pantalla del resultado:



1. Realizar una consulta en la TablaX o la TablaY o la TablaZ en la que se pueda obtener el **el valor máximo** de alguna columna numérica, pero el valor del resultado debe estar almacenado en una columna llamada El\_MAXIMO.

* INSTRUCCIÓN SQL:

SELECT MAX(precio\_inicial) AS El\_MAXIMO FROM Articulo;

* Explicación:

Obtiene el valor máximo del precio inicial de los artículos, con el alias El\_MAXIMO

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ en la que se pueda obtener el **el valor mínimo** de alguna columna numérica, pero el valor del resultado debe estar almacenado en una columna llamada El\_MINIMO.

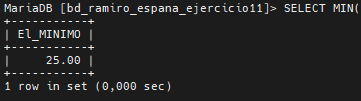
* INSTRUCCIÓN SQL:

SELECT MIN(precio\_inicial) AS El\_MINIMO FROM Articulo;

* Explicación:

tener el valor mínimo del precio inicial de los artículos, con el alias EL\_MINIMO

* Recorte de pantalla del resultado:



1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar los valores de dos columnas **concatenados** en una sola columna, pero el valor del resultado debe estar almacenado en una columna llamada VALORES\_UNIDOS.

* INSTRUCCIÓN SQL:

SELECT CONCAT(nombre, ' - ', descripcion) AS VALORES\_UNIDOS FROM Articulo;

* Explicación:

Obtiene el valor mínimo del precio inicial de los artículos, con el alias VALORES\_UNIDOS

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar el **MAYÚSCULAS** los valores de una columna, y en **Minúsculas** los valores de otra columna, pero en el resultado estas columnas deben llamarse EN\_MAYUSCULA, EN\_MINUSCULA respectivamente.

* INSTRUCCIÓN SQL:

SELECT UPPER(nombre) AS EN\_MAYUSCULA, LOWER(correo\_electronico) AS EN\_MINUSCULA FROM Usuario;

* Explicación:

Muestra en mayúsculas los valores de una columna y en minúsculas los de otra columna con los aliases EN\_MAYUSCULA y EN\_MINUSCULA.

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **el AÑO** de alguna columna Tipo Fecha, pero en el resultado esta columna debe llamarse EL\_AÑO.

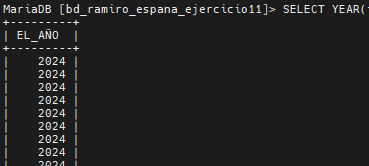
* INSTRUCCIÓN SQL:

SELECT YEAR(fecha\_registro) AS EL\_AÑO FROM Usuario;

* Explicación:

Muestra solo el año de una columna tipo fecha, con el alias EL\_AÑO

* Recorte de pantalla del resultado:



1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **el MES** de alguna columna Tipo Fecha, pero en el resultado esta columna debe llamarse EL\_MES

* INSTRUCCIÓN SQL:

SELECT MONTH(fecha\_registro) AS EL\_MES FROM Usuario;

* Explicación:

Muestra solo el mes de una columna tipo fecha, con el alias EL\_MES

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **el DIA** de alguna columna Tipo Fecha, pero en el resultado esta columna debe llamarse EL\_DIA.

* INSTRUCCIÓN SQL:

SELECT DAY(fecha\_registro) AS EL\_DIA FROM Usuario;

* Explicación:

Muestra solo el día de una columna tipo fecha, con el alias EL\_DIA

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **la fecha en formato personalizado,** así **(dia@mes@año)** de alguna columna Tipo Fecha, pero en el resultado esta columna debe llamarse FECHA\_RARA

* INSTRUCCIÓN SQL:

SELECT CONCAT(DAY(fecha\_registro), '@', MONTH(fecha\_registro), '@', YEAR(fecha\_registro)) AS FECHA\_RARA FROM Usuario;

* Explicación:

Muestra la fecha en formato (día@mes@año), con el alias FECHA\_PARA

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar sólo **la FECHA ACTUAL** en una nueva columna**,** pero en el resultado esta columna debe llamarse FECHA\_ACTUAL.

* INSTRUCCIÓN SQL:

SELECT CURRENT\_DATE() AS FECHA\_ACTUAL;

* Explicación:

Muestra la fecha actual en una nueva columna con el alias FECHA\_ACTUAL

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta en la TablaX o la TablaY o la TablaZ que permita mostrar en 3 nuevas columnas el resultado de **SUMAR**, **RESTAR** Y **MULTIPLICAR** los valores de 2 columnas Tipo numérico, pero en el resultado esta columna debe llamarse LA\_SUMA, LA\_RESTA, LA\_MULTIPLICACION respectivamente.

* INSTRUCCIÓN SQL:

SELECT

id\_oferta + precio\_ofertado AS LA\_SUMA,

id\_oferta - precio\_ofertado AS LA\_RESTA,

id\_oferta \* precio\_ofertado AS LA\_MULTIPLICACION

FROM Oferta;

* Explicación:

Muestra en nuevas columnas la suma, resta y multiplicación de id\_oferta y precio ofertado de la tabla Oferta con los aliases LA\_SUMA, LARESTA, LA\_MULTIPLICACION.

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Investigar cual es el objetivo de la Palabra DISTINCT de SQL

La palabra clave DISTINCT se utiliza en SQL para eliminar duplicados en los resultados de una consulta. Es útil cuando solo se quieren valores únicos de una columna o combinación de columnas.

1. Seleccionar alguna de las tablas TablaX o la TablaY o la TablaZ y realizar una consulta donde se pueda apreciar el uso de la palabra **DISTINCT**

* INSTRUCCIÓN SQL:

SELECT DISTINCT id\_tipo\_usuario FROM Usuario;

* Explicación:

Muestra todos los tipos únicos de estado del usuario (id\_estado\_usuario)

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta en la TablaX que permita mostrar los datos de un registro cuya columna llave primaria sea **igual** al valorY

* INSTRUCCIÓN SQL:

SELECT \* FROM Usuario WHERE id\_usuario = 5;

* Explicación:

Muestra los datos de un registro cuya columna llave primaria sea igual a 5 de la tabla Usuario.

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **igual** datoZ

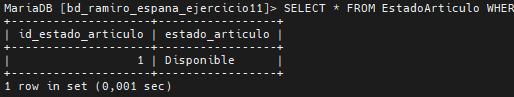
* INSTRUCCIÓN SQL:

SELECT \* FROM EstadoArticulo WHERE estado\_articulo = 'Disponible';

* Explicación:

Muestra registros que cumplan con la condición estado\_articulo igual a Disponible

* Recorte de pantalla del resultado:



1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **diferente**  datoZ

* INSTRUCCIÓN SQL:

SELECT \* FROM EstadoArticulo WHERE estado\_articulo != 'Disponible';

* Explicación:

Muestra registros que cumplan con la condición estado\_articulo es igual a No Disponible

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **menor**  datoZ

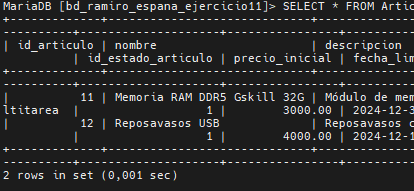
* INSTRUCCIÓN SQL:

SELECT \* FROM Articulo WHERE precio\_inicial < 5000;

* Explicación:

Muestra los registros que cumplan con la condición precio\_final menor a 5000

* Recorte de pantalla del resultado:



1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **mayor** datoZ

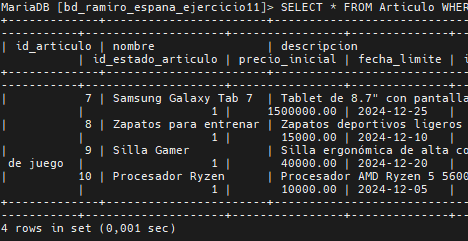
* INSTRUCCIÓN SQL:

SELECT \* FROM Articulo WHERE precio\_inicial > 5000;

* Explicación:

Muestra registros que cumplan con la condición precio\_inicial mayor a 5000

* Recorte de pantalla del resultado:



1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **entre valor** datoX y valorZ

* INSTRUCCIÓN SQL:

SELECT \* FROM Articulo WHERE precio\_inicial BETWEEN 1000 AND 5000;

* Explicación:

Muestra registros que cumplan con la condición precio\_inicial entre 1000 y 5000

* Recorte de pantalla del resultado:

Imagen que contiene Texto

Descripción generada automáticamente

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **Sea NULO**

* INSTRUCCIÓN SQL:

SELECT \* FROM Articulo WHERE descripcion IS NULL;

* Explicación:

Muestra registros que cumplan con la condición descripcion sea NULO

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **No Sea NULO**

* INSTRUCCIÓN SQL:

SELECT \* FROM Articulo WHERE descripcion IS NOT NULL;

* Explicación:

Muestra registros que cumplan con la condición descripcion no sea NULO

* Recorte de pantalla del resultado:

Interfaz de usuario gráfica

Descripción generada automáticamente

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **Contenga el valor** DatoX en su interior

* INSTRUCCIÓN SQL:

SELECT \* FROM Articulo WHERE descripcion LIKE '%deportivo%';

* Explicación:

Muestra registros que cumplen con la condición descripcion contiene el valor “deportivo”

* Recorte de pantalla del resultado:

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamente

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **Inicie con el valor** DatoX

* INSTRUCCIÓN SQL:

SELECT \* FROM Articulo WHERE nombre LIKE 'Laptop%';

* Explicación:

Muestra registros que cumplan con la condición descripcion inicie con el valor Laptop

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente con confianza media

1. Realizar una consulta en la TablaX que permita mostrar los registros con que cumplan con la conducción columnaY **Termine con el valor** DatoX

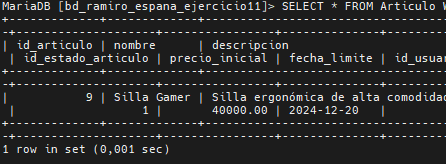
* INSTRUCCIÓN SQL:

SELECT \* FROM Articulo WHERE descripcion LIKE '%Juego';

* Explicación:

Muestra registros que cumplan con la condición descripcion termine con el valor Juego

* Recorte de pantalla del resultado:



1. Investigar cual es el objetivo del comando IN.

El comando IN en SQL se utiliza para especificar múltiples valores en una condición WHERE. Permite verificar si una columna tiene un valor que coincide con cualquiera de los valores dentro de un conjunto proporcionado. Es equivalente a usar múltiples condiciones OR, pero con sintaxis más compacta.

1. Realizar una consulta en la TablaX que permita mostrar el uso y aplicación de la palabra **IN**

* INSTRUCCIÓN SQL:

SELECT \* FROM EstadoArticulo WHERE estado\_articulo IN ('Disponible', 'Reservado');

* Explicación:

Muestra registros que cumplen con la condición de Disponible o Reservado

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Investigar cual es el objetivo del comando IN.

El comando IN se utiliza para comparar una columna con una lista de valores posibles, simplificando la consulta y mejorando la legibilidad al reemplazar múltiples condiciones OR.

Para realizar las siguientes consultas se asume que TablaA, TablaB, TablaC, …, TablaZ, representan relacionadas entre sí, mediante llaves foráneas, que se describen en El Modelo Relacional del ejercicio que han seleccionado.

Asumimos que TablaA y TablaB están relacionados así:

[**TablaA**] ->----<**R**>----|- [**TablaB**] ->----<**R**>----|- [**TablaC**]

Es decir, [**TablaA**] tiene una llave foránea llamada por ejemplo **tablaB\_id**, la cual se conecta con la llave primaria **id** de la [**TablaB**], mientras que [**TablaB**] tiene una llave foránea llamada por ejemplo **tablaC\_id** la cual se conecta con la llave primaria **id** de la [**TablaC**],

1. Realizar una consulta que permitan mostrar los registros de la **TablaA que están relacionados** con la **TablaB**

* INSTRUCCIÓN SQL:

SELECT A.\* FROM Articulo A JOIN Categoria C ON A.id\_categoria = C.id\_categoria;

* Explicación:

Muestra los registros de Articulo relacionados con Categoria

* Recorte de pantalla del resultado:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

1. Realizar una consulta que permitan mostrar los registros de la **TablaA** que **NO están relacionados** con la **TablaB**

* INSTRUCCIÓN SQL:

SELECT A.\* FROM Articulo A LEFT JOIN Categoria C ON A.id\_categoria = C.id\_categoria WHERE C.id\_categoria IS NULL;

* Explicación:

Muestra registros de la Tabla Articulo que NO están relacionados con la Tabla Categoria

* Recorte de pantalla del resultado:



1. Realizar una consulta que permitan mostrar los registros de la **TablaB** que **están relacionados** con la **TablaA**

* INSTRUCCIÓN SQL:

SELECT C.\* FROM Categoria C JOIN Articulo A ON C.id\_categoria = A.id\_categoria;

* Explicación:

Muestra registros de la Tabla Categoria relacionados con la Tabla Articulo

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta que permitan mostrar los registros de la **TablaB** que **NO están relacionados** con la **TablaA**

* INSTRUCCIÓN SQL:

SELECT C.\* FROM Categoria C LEFT JOIN Articulo A ON C.id\_categoria = A.id\_categoria WHERE A.id\_categoria IS NULL;

* Explicación:

Muestra registros de la Tabla Categoria que NO están relacionados con la Tabla Articulo

* Recorte de pantalla del resultado:



1. Realizar una consulta que permitan mostrar los registros de la **TablaB** que **NO están relacionados** con la **TablaA**

* INSTRUCCIÓN SQL:

SELECT C.\* FROM Categoria C LEFT JOIN Articulo A ON C.id\_categoria = A.id\_categoria LEFT JOIN Subasta S ON A.id\_articulo = S.id\_articulo WHERE S.id\_articulo IS NULL;

* Explicación:

Muestra registros de la Tabla Categoria que NO están relacionados con la Tabla Subasta.

* Recorte de pantalla del resultado:

Texto

Descripción generada automáticamente

1. Realizar una consulta que permitan mostrar los registros de la **TablaA que están relacionados** con la **TablaC** mediante la **TablaB**

* INSTRUCCIÓN SQL:

SELECT U.\* FROM Usuario U JOIN Articulo A ON U.id\_usuario = A.id\_usuario JOIN Subasta S ON A.id\_articulo = S.id\_articulo;

* Explicación:

Muestra registros de la Tabla Usuario relacionados con la Tabla Subasta mediante la Tabla Articulo

* Recorte de pantalla del resultado:



1. Supongamos que necesitamos buscar un registro en [**TablaA]** cuya columna **llave primaria** tiene el **valor X,** pero también necesitamos obtener los registros de [**TablaC**] que están relacionados con ese registro de [**TablaA**].

* INSTRUCCIÓN SQL:

SELECT A.\*, C.\* FROM Usuario A JOIN Subasta B ON A.id\_usuario = B.id\_usuario JOIN Operacion C ON B.id\_subasta = C.id\_subasta WHERE A.id\_usuario = 3;

* Explicación:

Muestra registros de la Tabla Usuario y los registros relacionados en la Tabla Subasta

* Recorte de pantalla del resultado:



**DESCRIPCIÓN DE LA ACTIVIDAD**

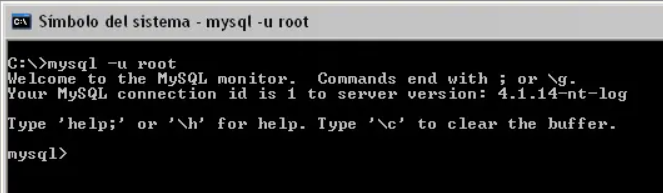
**CONDICIONES PARA EL DESARROLLO DE LA ACTIVIDAD:**

* Para interactuar con el Servidor de BD se debe única y exclusivamente la **Terminal de Línea de Comando (CLI o Consola)**  cada Motor de Bases de Datos. (NO SE ADMITE EL USO DE APLICACIONES CON INTERFAZ GRÁFICA)
* De debe usar este mismo documento de word para dar respuesta a cada uno de los ITEM, colocando como respuesta:
  + La instrucción SQL correcta en formato texto
  + Un recorte del fragmento de la terminal CLI donde fue ingresada y ejecutada la instrucción y su respectiva salida. No colocar la pantalla completa porque se hace ilegible la imagen. Una explicación sobre el la instrucción y la salida del mismo
  + Por ejemplo:

INSTRUCCIÓN SQL:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

RECORTE DE PANTALLA



EXPLICACIÓN.

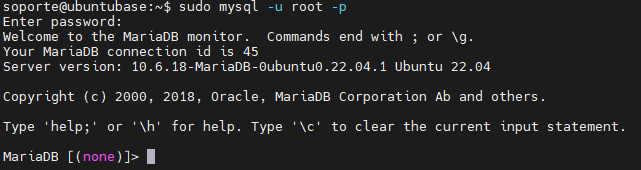
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Entrar al Motor de Bases** de datos usando la **Terminal (consola)** de Línea de Comandos usando el usuario creado durante la instrucción o el usuario por defecto con rol de super usuario.

* INSTRUCCIÓN SQL:

\_sudo mysql -u root -p\_\_\_\_\_\_\_\_\_

* RECORTE DE PANTALLA



* EXPLICACIÓN.

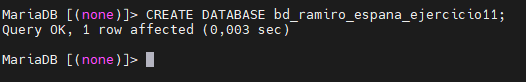
Esta orden indica que se debe iniciar la consola del motor especificando que se hace con el superadministrador (root) y que valida la contraseña antes de ingresar

1. **Crear una base de datos** (tablespace en oracle) llamada **bd\_ramiro\_espana\_ejercicio11**.

* INSTRUCCIÓN SQL:

CREATE DATABASE bd\_ramiro\_espana\_ejercicio11;

* RECORTE DE PANTALLA



* EXPLICACIÓN.

La orden CREATE DATABASE crea la estructura de una nueva base de datos o simplemente una base de datos vacía.

1. **Crear** un nuevo **usuario** para poder entrar al Motor de BD, los datos del nuevo usuario deben ser:

Nombre: **user\_ramiro\_espana**

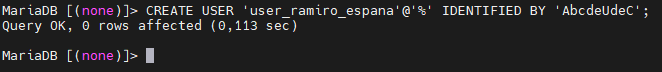
Clave: **AbcdeUdeC**

El nuevo usuario podrá conectarse desde cualquier dirección IP

* INSTRUCCIÓN SQL:

CREATE USER 'user\_ramiro\_espana'@'%' IDENTIFIED BY 'AbcdeUdeC';

* RECORTE DE PANTALLA



* EXPLICACIÓN.

Crea un nuevo usuario con el nombre user\_ramiro\_espana y la contraseña AbcdeUdeC. El símbolo % permite que el usuario se conecte desde cualquier dirección IP.

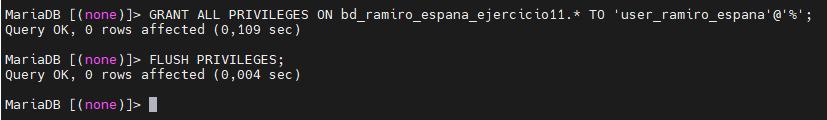
1. Al nuevo usuario **user\_ramiro\_espana** creado anteriormente , se le deben **asignar Privilegios de Superusuario** sobre la base de datos **bd\_ramiro\_espana\_ejercicio11**

* INSTRUCCIÓN SQL:

GRANT ALL PRIVILEGES ON bd\_ramiro\_espana\_ejercicio11.\* TO 'user\_ramiro\_espana'@'%';

FLUSH PRIVILEGES;

* RECORTE DE PANTALLA



* EXPLICACIÓN.

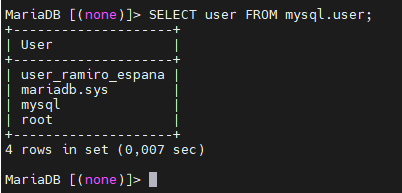
se le otorgan todos los permisos sobre la base de datos bd\_ramiro\_espana\_ejercicio11 al usuario user\_ramiro\_espana. FLUSH PRIVILEGES recarga los privilegios.

1. **Mostrar los usuarios** existentes

* INSTRUCCIÓN SQL:

SELECT user FROM mysql.user;

* RECORTE DE PANTALLA



* EXPLICACIÓN

Muestra la lista de todos los usuarios existentes en el sistema de bases de datos MariaDB.

1. **Salir de la terminal** o consola del Motor de BD

* INSTRUCCIÓN SQL:

exit

* RECORTE DE PANTALLA



* EXPLICACIÓN

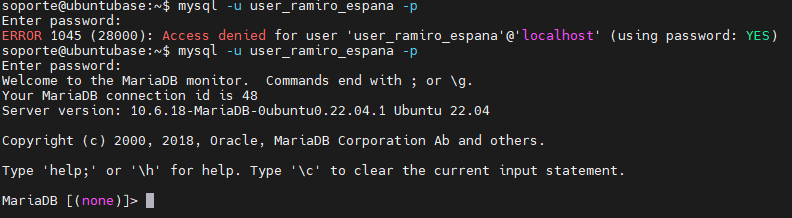
Termina la sesión en la terminal del motor de bases de datos y cierra la conexión.

1. Volver a **entrar** la Motor de Bases de datos **desde la terminal o consola** pero ahora usando las c**redenciales del nuevo usuario** **user\_ramiro\_espana** creado anteriormente-

* INSTRUCCIÓN SQL:

mysql -u user\_ramiro\_espana -p

* RECORTE DE PANTALLA



* EXPLICACIÓN

Inicia sesión en MariaDB utilizando el nuevo usuario user\_ramiro\_espana

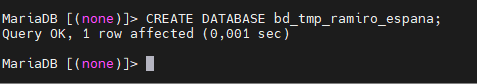
Las siguientes operaciones SQL deben ser realizadas utilizando el nuevo usuario **user\_ramiro\_espana**.

1. **Crear** una **nueva BD** llamada **bd\_tmp\_ramiro\_espana**

* INSTRUCCIÓN SQL:

CREATE DATABASE bd\_tmp\_ramiro\_espana;

* RECORTE DE PANTALLA



* EXPLICACIÓN

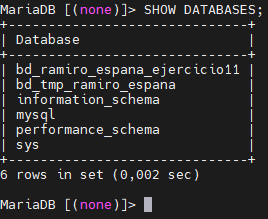
Crea una nueva base de datos llamada bd\_tmp\_ramiro\_espana

1. **Mostrar** las **Bases de datos** que pertenecen al **user\_ramiro\_espana**

* INSTRUCCIÓN SQL:

SHOW DATABASES;

* RECORTE DE PANTALLA



* EXPLICACIÓN

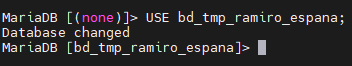
muestra todas las bases de datos a las que tiene acceso el usuario user\_ramiro\_espana actualmente conectado.

1. Entrar o usar la BD **bd\_tmp\_ramiro\_espana**

* INSTRUCCIÓN SQL:

USE bd\_tmp\_ramiro\_espana;

* RECORTE DE PANTALLA



* EXPLICACIÓN

Selecciona la base de datos bd\_tmp\_ramiro\_espana como la base de datos activa para realizar operaciones.

1. **Crear una tabla** llamada **bd\_tmp\_ramiro\_espana** con las siguiente estructura

**tabla\_tempora**

**id** auto\_incremental y clave primera

**columna2** de tipo Texto, no nula y con longitud de 100

**columna3** de tipo Entero

**columna4** de tipo Real

**columna5** de tipo Fecha

**columna6** de tipo FechaHora

**columna7** de tipo hora

**columna8** de tipo FechaHora automatica

**columna9** de tipo Texto con longitud variable

**columna10** de tipo archivo binario

* INSTRUCCIÓN SQL:

CREATE TABLE tabla\_tempora (

id INT AUTO\_INCREMENT PRIMARY KEY,

columna2 VARCHAR(100) NOT NULL,

columna3 INT,

columna4 FLOAT,

columna5 DATE,

columna6 DATETIME,

columna7 TIME,

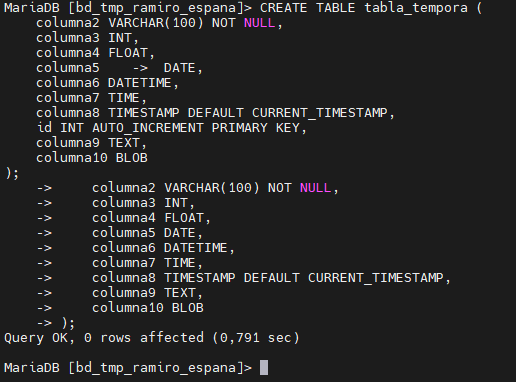
columna8 TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

columna9 TEXT,

columna10 BLOB

);

* RECORTE DE PANTALLA



* EXPLICACIÓN

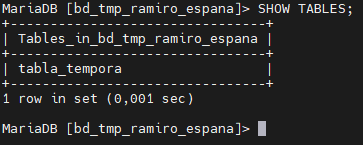
crea una tabla con 10 columnas, incluyendo un campo id autoincremental y varios tipos de datos como texto, enteros, fechas y binarios

1. **Mostrar las tablas** de la **bd\_tmp\_ramiro\_espana**

* INSTRUCCIÓN SQL:

SHOW TABLES;

* RECORTE DE PANTALLA



* EXPLICACIÓN

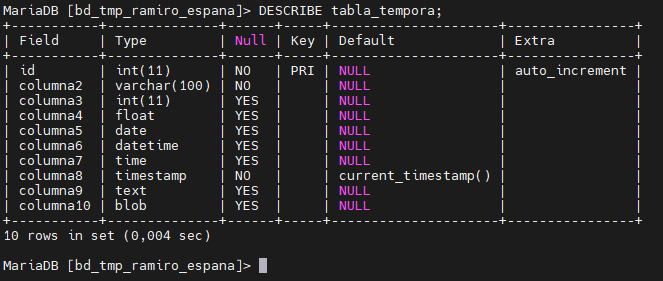
lista todas las tablas existentes dentro de la base de datos bd\_tmp\_ramiro\_espana.

1. **Describir** la estructura de la **tabla\_tempora**

* INSTRUCCIÓN SQL:

DESCRIBE tabla\_tempora;

* RECORTE DE PANTALLA



* EXPLICACIÓN

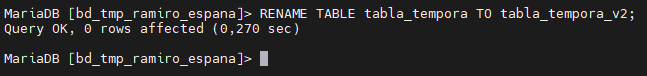
muestra la estructura de la tabla tabla\_tempora, incluyendo los nombres de columnas, tipos de datos y restricciones.

1. **Cambiar** el **nombre** de la **tabla\_tempora** por **tabla\_tempora\_v2**

* INSTRUCCIÓN SQL:

RENAME TABLE tabla\_tempora TO tabla\_tempora\_v2;

* RECORTE DE PANTALLA



* EXPLICACIÓN

Renombra la tabla tabla\_tempora como tabla\_tempora\_v2.

1. **Mostrar** las **tablas** de la **bd\_temporal**

INSTRUCCIÓN SQL:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

RECORTE DE PANTALLA

EXPLICACIÓN

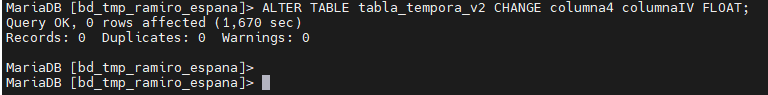
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. En la tabla **tabla\_tempora\_v2 cambiar el nombre** de la **columna4** por **columnaIV**

* INSTRUCCIÓN SQL:

ALTER TABLE tabla\_tempora\_v2 CHANGE columna4 columnaIV FLOAT;

* RECORTE DE PANTALLA



* EXPLICACIÓN

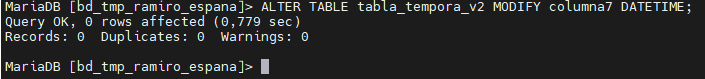
cambia el nombre de columna4 a columnaIV, conservando el tipo de dato como FLOAT.

1. En la tabla **tabla\_tempora\_v2 cambiar el tipo** de dato de la **columna7** por el tipo **FechaHora**

* INSTRUCCIÓN SQL:

ALTER TABLE tabla\_tempora\_v2 MODIFY columna7 DATETIME;

* RECORTE DE PANTALLA



* EXPLICACIÓN

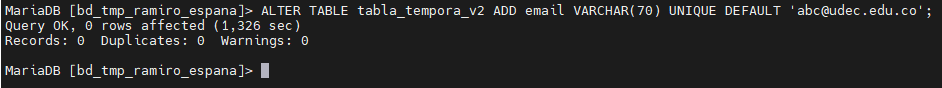
Cambia el tipo de la columna columna7 de TIME a DATETIME.

1. En la tabla **tabla\_tempora\_v2 agregar una columna** nueva llamada **email** de tipo Texto de longitud 70, que sea **única** y con valor por defecto **abc@udec.edu.co**

* INSTRUCCIÓN SQL:

ALTER TABLE tabla\_tempora\_v2 ADD email VARCHAR(70) UNIQUE DEFAULT 'abc@udec.edu.co';

* RECORTE DE PANTALLA



* EXPLICACIÓN

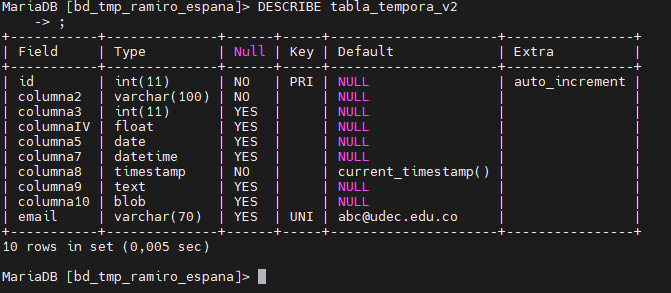
Agrega una columna email con una longitud de 70 caracteres, valor por defecto y restricción de unicidad.

1. **Describir** la Estructura de la **tabla\_tempora\_v2** para observar los cambios

* INSTRUCCIÓN SQL:

DESCRIBE tabla\_tempora\_v2

* RECORTE DE PANTALLA



* EXPLICACIÓN

Describe la estructura de la tabla\_tempora2

1. **Eliminar** la **columna6**

* INSTRUCCIÓN SQL:

ALTER TABLE tabla\_tempora\_v2 DROP COLUMN columna6;

* RECORTE DE PANTALLA



* EXPLICACIÓN

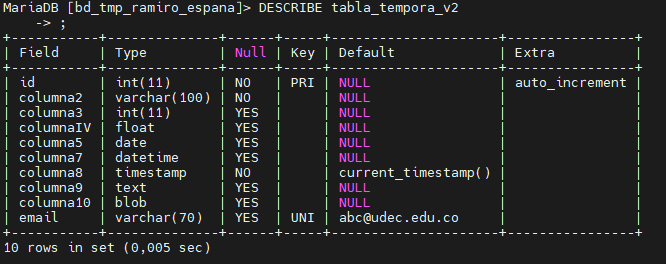
Elimina la columna columna6 de la tabla tabla\_tempora\_v2.

1. **Describir** la Estructura de la **tabla\_tempora\_v2** para observar los cambios

INSTRUCCIÓN SQL:

DESCRIBE tabla\_tempora\_v2

RECORTE DE PANTALLA



EXPLICACIÓN

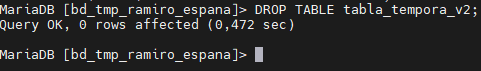
Describe el estado actual de la tabla, después de las modificaciones

1. **Eliminar** la **tabla\_tempora\_v2**

* INSTRUCCIÓN SQL:

DROP TABLE tabla\_tempora\_v2;

* RECORTE DE PANTALLA



* EXPLICACIÓN

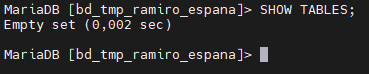
Elimina por completo la tabla tabla\_tempora\_v2 de la base de datos.

1. **Mostrar** las **tablas** de la **bd\_tmp\_ramiro\_espana**

* INSTRUCCIÓN SQL:

SHOW TABLES;

* RECORTE DE PANTALLA



* EXPLICACIÓN

Muestra las tablas (si existen) en la base de datos seleccionada.

1. **Cambiarle** el **nombre** de la **base de datos** **bd\_tmp\_ramiro\_espana** por **bd\_tmp\_ramiro\_espana\_v2**

Según el motor de BD utilizado esta operación no se realiza por CLI, si ese es su caso, entonces, debe explicar el proceso necesario para cambiar el nombre a la BD.

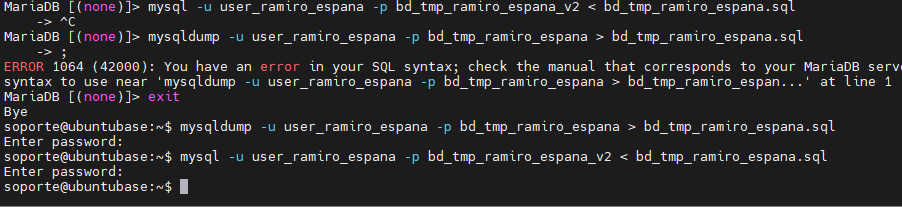
* INSTRUCCIÓN SQL:

*exportación*: mysqldump -u user\_ramiro\_espana -p bd\_tmp\_ramiro\_espana > bd\_tmp\_ramiro\_espana.sql

*creación*: CREATE DATABASE bd\_tmp\_ramiro\_espana\_v2;

*importación*: mysql -u user\_ramiro\_espana -p bd\_tmp\_ramiro\_espana\_v2 < bd\_tmp\_ramiro\_espana.sql

* RECORTE DE PANTALLA



* EXPLICACIÓN

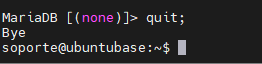
En MariaDB, no se puede renombrar una base de datos directamente con una instrucción SQL. El procedimiento implica crear una nueva base de datos, exportar los datos de la base de datos antigua, importarlos en la nueva base y eliminar la base de datos original. Por lo que primero se hace una exportación de datos de la base, luego se crea la base nueva y se hace la importación hacia la nueva base datos.

1. Salid de la CLI de la terminal del motor de BD

* INSTRUCCIÓN SQL:

quit;

* RECORTE DE PANTALLA



* EXPLICACIÓN

Esta instrucción ejecuta la salida de la consola inmediatamente después de pulsar enter (Igualmente se puede utilizar el comando general de salida de consola del sistema base: *exit*)

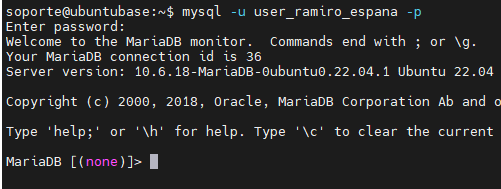


1. Entrar al CLI de la terminal del motor de BD pero ahora utilizando las credenciales del usuario **user\_ramiro\_espana**

* INSTRUCCIÓN SQL:

mysql -u user\_ramiro\_espana -p

* RECORTE DE PANTALLA



* EXPLICACIÓN

Esta orden hace el ingreso a la consola del administrador de bases de datos con el usuario *user\_ramiro\_espana* con la contraseña previamente asignada.

1. **Mostrar** las **Bases de datos** que pertenecen al **user\_ramiro\_espana**

* INSTRUCCIÓN SQL:

SHOW DATABASES;

* RECORTE DE PANTALLA

Texto

Descripción generada automáticamente

* EXPLICACIÓN

Esta orden muestra todas las bases de datos a las que tiene acceso el usuario *user\_ramiro\_espana*

1. Eliminar el usuario **user\_tmp\_ramiro\_espana**

* INSTRUCCIÓN SQL:

DROP USER 'user\_tmp\_ramiro\_espana'@'%';

* RECORTE DE PANTALLA



* EXPLICACIÓN

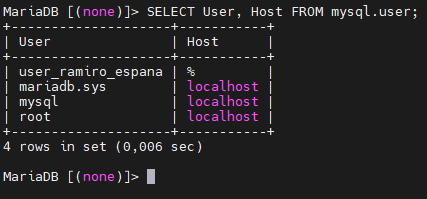
La orden intenta eliminar el usuario user\_tmp\_ramiro\_espana, pero al no existir este usuario arroja un error del tipo falla.

1. Mostrar los usuarios activos en el motor de bases de datos.

* INSTRUCCIÓN SQL:

SELECT User, Host FROM mysql.user;

* RECORTE DE PANTALLA



* EXPLICACIÓN

Utilizando la orden *SELECT User, Host FROM mysql.user;* se puede validar el listado actual de usuarios activos o disponibles.

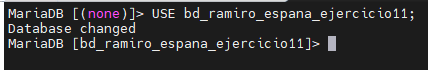
Las siguientes operaciones SQL se deben realizar tomando como base **Modelo Relacional** del ejercicio que le fue asignado:

1. **Entrar** o usar la base de datos **bd\_tu\_ramiro\_espana\_ejercicio11** creada anteriormente.

* INSTRUCCIÓN SQL:

USE bd\_ramiro\_espana\_ejercicio11;

* RECORTE DE PANTALLA



* EXPLICACIÓN

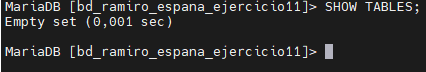
Esta orden selecciona como bases de datos de trabajo única y exclusivamente *bd\_ramiro\_espana\_ejercicio11*, lo que indica que cualquier que se realice de auí en adelante solo afectará a esta base de datos.

1. **Mostrar** las **tablas** de la **bd\_tu\_ramiro\_espana\_ejercicio11**

* INSTRUCCIÓN SQL:

SHOW TABLES;

* RECORTE DE PANTALLA



* EXPLICACIÓN

La orden SHOW TABLES; se ejecuta para mostrar todas las tablas que existan en la base de datos en la que se está trabajando actualmente.

1. **Crear** las tablasde acuerdo a lo reflejado en el Modelo Relacional del ejercicio.

* INSTRUCCIONES SQL PARA CREAR LAS TABLAS:

CREATE TABLE Usuario (

id\_usuario INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(50),

apellidos VARCHAR(50),

direccion VARCHAR(100),

correo\_electronico VARCHAR(100),

fecha\_registro DATE,

id\_estado\_usuario INT,

id\_tipo\_usuario INT,

contrasena VARCHAR(100),

preferencias TEXT

);

CREATE TABLE TipoUsuario (

id\_tipo\_usuario INT PRIMARY KEY AUTO\_INCREMENT,

tipo\_usuario VARCHAR(50)

);

CREATE TABLE EstadoUsuario (

id\_estado\_usuario INT PRIMARY KEY AUTO\_INCREMENT,

estado\_usuario VARCHAR(50)

);

CREATE TABLE Articulo (

id\_articulo INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(50),

descripcion TEXT,

id\_estado\_articulo INT,

precio\_inicial DECIMAL(10,2),

fecha\_limite DATE,

id\_usuario INT,

id\_categoria INT

);

CREATE TABLE EstadoArticulo (

id\_estado\_articulo INT PRIMARY KEY AUTO\_INCREMENT,

estado\_articulo VARCHAR(50)

);

CREATE TABLE Subasta (

id\_subasta INT PRIMARY KEY AUTO\_INCREMENT,

id\_articulo INT,

fecha\_inicio DATE,

fecha\_limite DATE,

id\_estado\_subasta INT,

id\_usuario INT

);

CREATE TABLE EstadoSubasta (

id\_estado\_subasta INT PRIMARY KEY AUTO\_INCREMENT,

estado\_subasta VARCHAR(50)

);

CREATE TABLE Oferta (

id\_oferta INT PRIMARY KEY AUTO\_INCREMENT,

id\_subasta INT,

id\_usuario INT,

precio\_ofertado DECIMAL(10,2),

fecha\_oferta DATE

);

CREATE TABLE Categoria (

id\_categoria INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(50)

);

CREATE TABLE Comentario (

id\_comentario INT PRIMARY KEY AUTO\_INCREMENT,

id\_usuario INT,

id\_articulo INT,

texto\_comentario TEXT,

fecha\_comentario DATE,

id\_comentario\_tipo INT

);

CREATE TABLE ComentarioTipo (

id\_comentario\_tipo INT PRIMARY KEY AUTO\_INCREMENT,

comentario\_tipo VARCHAR(50)

);

CREATE TABLE Operacion (

id\_operacion INT PRIMARY KEY AUTO\_INCREMENT,

id\_subasta INT,

id\_usuario INT,

fecha\_operacion DATE,

precio\_final DECIMAL(10,2)

);

CREATE TABLE Soporte (

id\_caso INT PRIMARY KEY AUTO\_INCREMENT,

id\_usuario INT,

asunto VARCHAR(100),

descripcion TEXT,

fecha\_solicitud DATE,

id\_estado\_caso INT

);

CREATE TABLE EstadoCaso (

id\_estado\_caso INT PRIMARY KEY AUTO\_INCREMENT,

estado\_caso VARCHAR(50)

);

CREATE TABLE Satisfaccion (

id\_satisfaccion INT PRIMARY KEY AUTO\_INCREMENT,

id\_usuario INT,

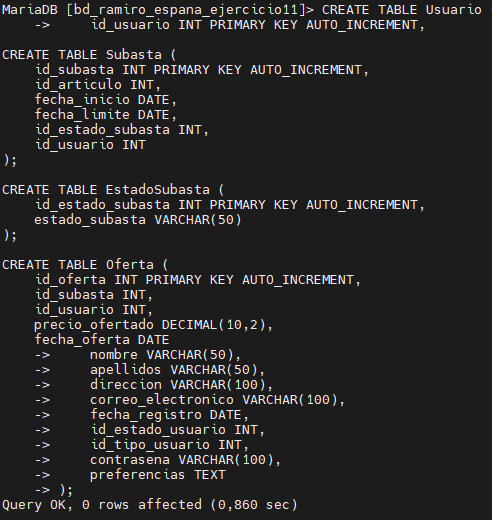
puntuacion INT,

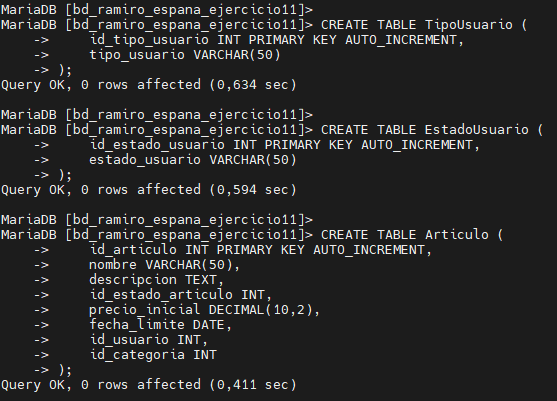
comentario TEXT,

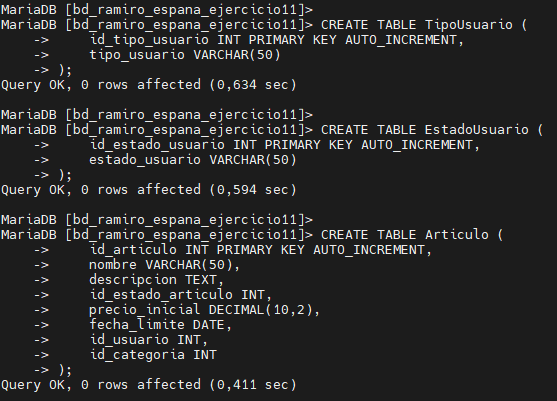
fecha\_satisfaccion DATE

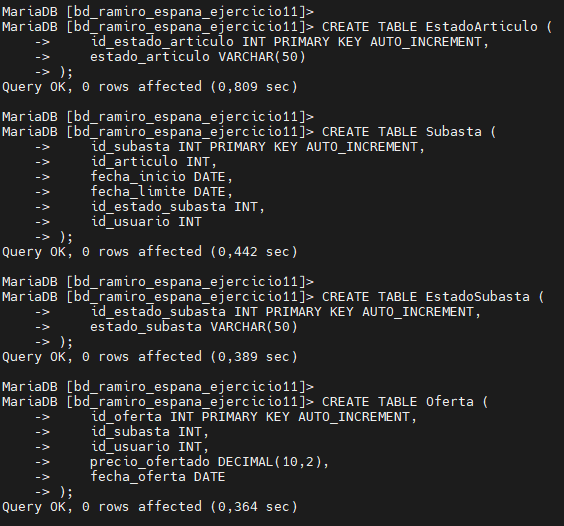
);

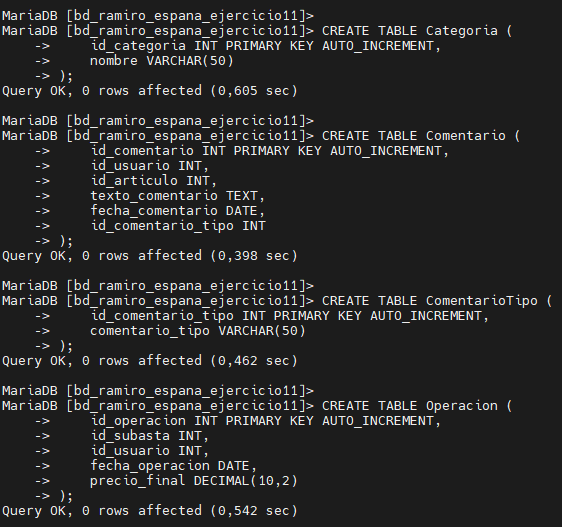
* RECORTES DE PANTALLAS











Texto

Descripción generada automáticamente

* EXPLICACIÓN

Estas instrucciones crean las tablas en la base de datos de acuerdo al Modelo Relacional que se desarrolló desde la actividad 11, reflejando las entidades y la asignación arbitraria de las características (tamaño) de los atributos.

1. **Describir** cada una de las **tablas** creadas creadas en la BD **bd\_ramiro\_espana\_ejercicio11**

* INSTRUCCIONES SQL PARA DESCRIBIR LAS TABLAS:

DESCRIBE Usuario;

DESCRIBE TipoUsuario;

DESCRIBE EstadoUsuario;

DESCRIBE Articulo;

DESCRIBE EstadoArticulo;

DESCRIBE Subasta;

DESCRIBE EstadoSubasta;

DESCRIBE Oferta;

DESCRIBE Categoria;

DESCRIBE Comentario;

DESCRIBE ComentarioTipo;

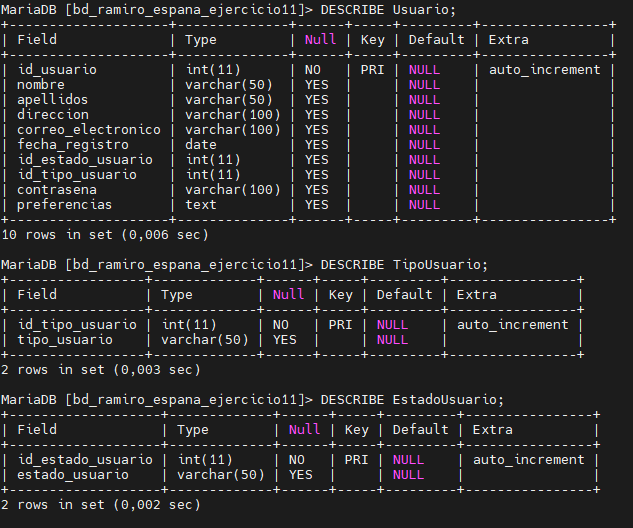
DESCRIBE Operacion;

DESCRIBE Soporte;

DESCRIBE EstadoCaso;

DESCRIBE Satisfaccion;

* RECORTES DE PANTALLAS



Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

* EXPLICACIÓN

Cada comando DESCRIBE proporciona la estructura de cada una de las tablas creadas, incluyendo los nombres de columnas, tipos de datos, y restricciones (si las hubiera). Tener en cuenta que la orden solo se puede ejecutar una tabla a la vez, no funcionará *DESCRIBE tabla1, tabla2, etc;*

1. Si las tablas aún no tienen las columnas que deben actuar como llaves foráneas, entonces, usted debe agregar dichas columnas a en respectivas tablas.

* INSTRUCCIONES SQL PARA AGREGAR LAS COLUMNAS:

ALTER TABLE Usuario ADD id\_estado\_usuario INT;

ALTER TABLE Usuario ADD id\_tipo\_usuario INT;

ALTER TABLE Articulo ADD id\_estado\_articulo INT;

ALTER TABLE Articulo ADD id\_usuario INT;

ALTER TABLE Articulo ADD id\_categoria INT;

ALTER TABLE Subasta ADD id\_articulo INT;

ALTER TABLE Subasta ADD id\_estado\_subasta INT;

ALTER TABLE Subasta ADD id\_usuario INT;

ALTER TABLE Oferta ADD id\_subasta INT;

ALTER TABLE Oferta ADD id\_usuario INT;

ALTER TABLE Comentario ADD id\_usuario INT;

ALTER TABLE Comentario ADD id\_articulo INT;

ALTER TABLE Comentario ADD id\_comentario\_tipo INT;

ALTER TABLE Operacion ADD id\_subasta INT;

ALTER TABLE Operacion ADD id\_usuario INT;

ALTER TABLE Soporte ADD id\_usuario INT;

ALTER TABLE Soporte ADD id\_estado\_caso INT;

ALTER TABLE Satisfaccion ADD id\_usuario INT;

* RECORTES DE PANTALLAS

Texto

Descripción generada automáticamente

* EXPLICACIÓN

Dado que las tablas en la orden creación ya tenían las columnas identificadas como llaves foráneas, se recibe el error 1060 que indica que hay columnas duplicadas

1. **Agregar** los **índices** en las columnas que actuarán como llaves foráneas

* INSTRUCCIONES SQL PARA CREAR ÍNDICES:

CREATE INDEX idx\_estado\_usuario ON Usuario(id\_estado\_usuario);

CREATE INDEX idx\_tipo\_usuario ON Usuario(id\_tipo\_usuario);

CREATE INDEX idx\_estado\_articulo ON Articulo(id\_estado\_articulo);

CREATE INDEX idx\_usuario\_articulo ON Articulo(id\_usuario);

CREATE INDEX idx\_categoria\_articulo ON Articulo(id\_categoria);

CREATE INDEX idx\_articulo\_subasta ON Subasta(id\_articulo);

CREATE INDEX idx\_estado\_subasta ON Subasta(id\_estado\_subasta);

CREATE INDEX idx\_usuario\_subasta ON Subasta(id\_usuario);

CREATE INDEX idx\_subasta\_oferta ON Oferta(id\_subasta);

CREATE INDEX idx\_usuario\_oferta ON Oferta(id\_usuario);

CREATE INDEX idx\_usuario\_comentario ON Comentario(id\_usuario);

CREATE INDEX idx\_articulo\_comentario ON Comentario(id\_articulo);

CREATE INDEX idx\_comentario\_tipo ON Comentario(id\_comentario\_tipo);

CREATE INDEX idx\_subasta\_operacion ON Operacion(id\_subasta);

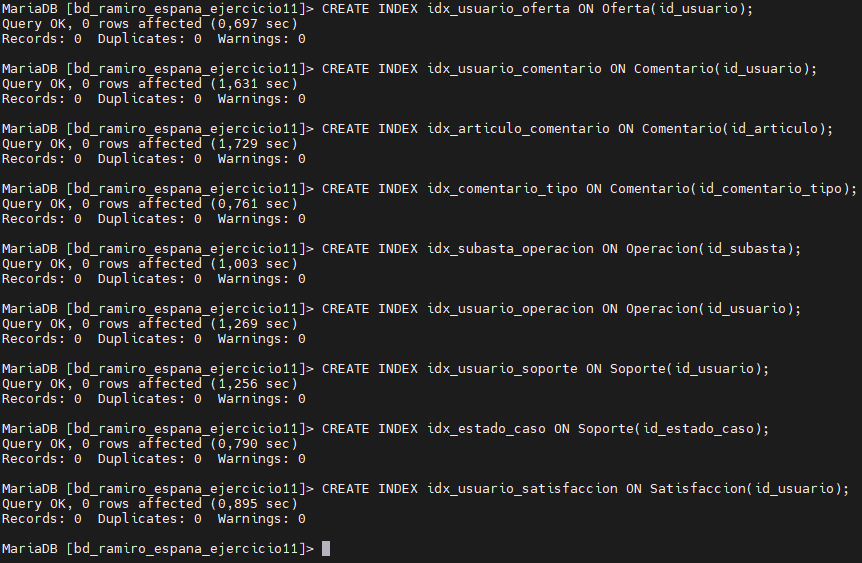
CREATE INDEX idx\_usuario\_operacion ON Operacion(id\_usuario);

CREATE INDEX idx\_usuario\_soporte ON Soporte(id\_usuario);

CREATE INDEX idx\_estado\_caso ON Soporte(id\_estado\_caso);

CREATE INDEX idx\_usuario\_satisfaccion ON Satisfaccion(id\_usuario);

* RECORTES DE PANTALLAS

  
NOTA: para prevenir el exceso de imágenes adjuntas, se agrega solo una que muestra que la orden se ha ejecutado sin novedad sobre errores o fallas.

* EXPLICACIÓN

La orden CREATE INDEX crea índices con el objetivo de optimizar el acceso a las columnas que actuarán como llaves foráneas.

1. **Agregar** restricciones de **llave foránea** en cada una de las tablas a las que haya lugar según el **Modelo Relacional**, cada llave foránea debe aplicar restricción de integridad referencial Restrictiva para operaciones de Eliminar y Cascada para Operaciones de Actualizar.

**(Estudiar e investigar qué es integridad referencial, como se aplica esto en las llaves foráneas)**

* INSTRUCCIONES SQL PARA AGREGAR LAS LLAVES FORÁNEAS:

ALTER TABLE Usuario ADD CONSTRAINT fk\_estado\_usuario FOREIGN KEY (id\_estado\_usuario) REFERENCES EstadoUsuario(id\_estado\_usuario) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Usuario ADD CONSTRAINT fk\_tipo\_usuario FOREIGN KEY (id\_tipo\_usuario) REFERENCES TipoUsuario(id\_tipo\_usuario) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Articulo ADD CONSTRAINT fk\_estado\_articulo FOREIGN KEY (id\_estado\_articulo) REFERENCES EstadoArticulo(id\_estado\_articulo) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Articulo ADD CONSTRAINT fk\_usuario\_articulo FOREIGN KEY (id\_usuario) REFERENCES Usuario(id\_usuario) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Articulo ADD CONSTRAINT fk\_categoria\_articulo FOREIGN KEY (id\_categoria) REFERENCES Categoria(id\_categoria) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Subasta ADD CONSTRAINT fk\_articulo\_subasta FOREIGN KEY (id\_articulo) REFERENCES Articulo(id\_articulo) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Subasta ADD CONSTRAINT fk\_estado\_subasta FOREIGN KEY (id\_estado\_subasta) REFERENCES EstadoSubasta(id\_estado\_subasta) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Subasta ADD CONSTRAINT fk\_usuario\_subasta FOREIGN KEY (id\_usuario) REFERENCES Usuario(id\_usuario) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Oferta ADD CONSTRAINT fk\_subasta\_oferta FOREIGN KEY (id\_subasta) REFERENCES Subasta(id\_subasta) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Oferta ADD CONSTRAINT fk\_usuario\_oferta FOREIGN KEY (id\_usuario) REFERENCES Usuario(id\_usuario) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Comentario ADD CONSTRAINT fk\_usuario\_comentario FOREIGN KEY (id\_usuario) REFERENCES Usuario(id\_usuario) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Comentario ADD CONSTRAINT fk\_articulo\_comentario FOREIGN KEY (id\_articulo) REFERENCES Articulo(id\_articulo) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Comentario ADD CONSTRAINT fk\_comentario\_tipo FOREIGN KEY (id\_comentario\_tipo) REFERENCES ComentarioTipo(id\_comentario\_tipo) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Operacion ADD CONSTRAINT fk\_subasta\_operacion FOREIGN KEY (id\_subasta) REFERENCES Subasta(id\_subasta) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Operacion ADD CONSTRAINT fk\_usuario\_operacion FOREIGN KEY (id\_usuario) REFERENCES Usuario(id\_usuario) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Soporte ADD CONSTRAINT fk\_usuario\_soporte FOREIGN KEY (id\_usuario) REFERENCES Usuario(id\_usuario) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Soporte ADD CONSTRAINT fk\_estado\_caso FOREIGN KEY (id\_estado\_caso) REFERENCES EstadoCaso(id\_estado\_caso) ON DELETE RESTRICT ON UPDATE CASCADE;

ALTER TABLE Satisfaccion ADD CONSTRAINT fk\_usuario\_satisfaccion FOREIGN KEY (id\_usuario) REFERENCES Usuario(id\_usuario) ON DELETE RESTRICT ON UPDATE CASCADE;

* RECORTES DE PANTALLAS

Texto

Descripción generada automáticamente

* EXPLICACIÓN

Las instrucciones (una por cada tabla) agregan las llaves foráneas con las restricciones de integridad referencial restrictivo para eliminar y cascada para actualizar. Profundizando en los conceptos aplicados; la restricción restrictiva para eliminar evita que se elimine una fila en la tabla padre si existe alguna relación con la tabla hija, protegiendo la integridad de los datos cuando la tabla ya esté poblada. Mientras que la restricción en cascada para actualizar se usa en caso de que una clave primaria en la tabla padre se actualiza, este cambio se propaga automáticamente en la tabla hija algo así como una modificación recursiva.

1. Utilizar alguna herramienta para visualizar el Modelo Relacional a partir de la BD ya creada. Los tres motores de BD que usaron para el desarrollo de esta actividad ofrecen una herramienta que nos permite ingresar instrucciones SQL al motor de BD de forma rápida e intuitiva, gracias a que están desarrolladas usando Interfaz Gráfica de Usuario (GUI), también nos permiten gestionar cómodamente todos los componentes del servidor y los elementos de cualquier BD dentro de dicho motor, así como también, nos permiten generar automáticamente el MR (Modelo Relacional) seleccionado una BD previamente creará, o viceversa, nos permiten crear la BD a partir de un MR.

COLOCAR AQUÍ LA IMAGEN DEL MODELO RELACIONAL GENERADO AUTOMÁTICAMENTE

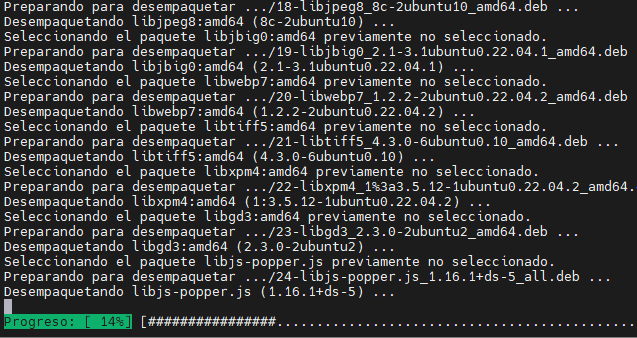
PASOS PARA GENERAR EL MODELO RELACIONAL A PARTIR DE LA BD CREADA EN EL RESPECTIVO MOTOR DE BD.

Haciendo uso de lo recomendado, se procede a instalar el administrador de bases de datos phpmyadmin (seleccionado por familiaridad y por previos usos personal y laboralmente), investigando sobre la posibilidad de poder crear el MR a partir de la base de datos se ha encontrado suficiente información para trabajar con esta herramienta. (guía de referencia <https://parzibyte.me/blog/2020/09/28/generar-diagrama-base-datos-phpmyadmin/> )

Breve guía de instalación (oficial): <https://tecadmin.net/how-to-install-phpmyadmin-on-ubuntu-22-04/>

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente



Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Sitio web

Descripción generada automáticamente

Se ingresa con el usuario user\_ramiro\_espana

Interfaz de usuario gráfica

Descripción generada automáticamente

MR Generado

Interfaz de usuario gráfica

Descripción generada automáticamente

Enlace público

<https://drive.google.com/file/d/1ps9gHNsFEbpXMl9nIV23P-dKYzJU74Mv/view?usp=sharing>

1. Crear un **Backup** solo **de la estructura** de la **bd\_tu\_ramiro\_espana\_ejercicio11** y guardar el resultado en un archivo llamado **backup\_bd\_ramiro\_espana\_ejercicio11.sql**

PASOS PARA GENERAR EL BACKUP DE LA BD CREADA EN EL RESPECTIVO MOTOR DE BD. (IDEAL UTILIZAR EL COMANDO SQL APROPIADO SEGÚN CADA MOTOR DE BD Y OPCIONALMENTE TAMBIÉN PUEDEN HACERLO Y EXPLICARLO USANDO EL MODO GRÁFICO)

Primero se ejecuta la orden para ingresar (reingresar) a la consola de mysql

Texto

Descripción generada automáticamente

luego se identifica la base de datos objetivo

Texto

Descripción generada automáticamente

Se sale de la consola cli y en la terminal de servidor se ejecuta la orden para crear la copia solo de la estructura con:

*mysqldump -u user\_ramiro\_espana -p --no-data bd\_ramiro\_espana\_ejercicio11 > backup\_bd\_ramiro\_espana\_ejercicio11.sql;*

**Si se ejecuta correctamente no muestra ningún mensaje de confirmación, caso contrario muestra algún mensaje de error.**

Se valida con la comando **ls** en el directorio actual y efectivamente se valida que la copia de seguridad ha sido creada

Texto

Descripción generada automáticamente

1. Eliminar la Base de datos **bd\_tu\_nombre\_tu\_apellido\_xyz**

* INSTRUCCIÓN SQL :

DROP DATABASE bd\_ramiro\_espana\_ejercicio11;

* RECORTES DE PANTALLAS

Texto

Descripción generada automáticamente

* EXPLICACIÓN

Esta orden elimina completamente la base de datos bd\_ramiro\_espana\_ejercicio11, borrando todas las tablas y datos almacenados en ella.

1. Mostrar las bases de datos del usuario **bd\_tu\_nombre\_tu\_apellido\_xyz**

* INSTRUCCIÓN SQL :

SHOW DATABASES;

* RECORTES DE PANTALLAS

Texto

Descripción generada automáticamente

* EXPLICACIÓN

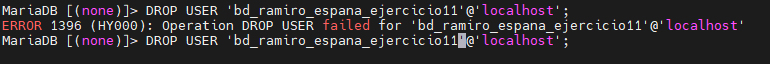
Esta instrucción muestra todas las bases de datos disponibles para el usuario, y se puede evidenciar que la base de datos previamente eliminada ya no está en el listado.

1. Eliminar el Usuario **bd\_tu\_nombre\_tu\_apellido\_xyz**

* INSTRUCCIÓN SQL :

DROP USER 'bd\_ramiro\_espana\_ejercicio11'@'localhost';

* RECORTES DE PANTALLAS



* EXPLICACIÓN

El comando debe eliminar el usuario discriminado y revocar todos sus derechos, pero al no existir un usuario con esa estructura se produce un mensaje de error tipo falla.

1. Utilizar el archivo de Backup para realizar restauración de la BD previamente eliminada.

* INSTRUCCIÓN SQL :

CREATE DATABASE bd\_ramiro\_espana\_ejercicio11;

*mysql -u user\_ramiro\_espana -p bd\_ramiro\_espana\_ejercicio11 < backup\_bd\_ramiro\_espana\_ejercicio11.sql*

SHOW DATABASES;

USE bd\_ramiro\_espana\_ejercicio11;

DESCRIBE Usuario;

* RECORTES DE PANTALLAS

Texto

Descripción generada automáticamente



Texto

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

* EXPLICACIÓN

Primero se ingresa a la consola de mysql para crear una base de datos vacía que pueda recibir la copia de seguridad hecha.

Luego se sale de la consola CLI y en la consola del servidor se ejecuta la orden para descargar toda la copia de seguridad en la nueva base de datos vacía, la dirección del símbolo < indica cual es el origen y cual es el destino. (siempre se toma un tiempo sin avisar a menos que haya surgido algún error. Luego se valida si efectivamente la base de datos ha sido restaurada completamente ingresando al CLI del motor de base de datos, seleccionando la base y tomando como referencia la tabla Usuario, para que muestre su estructura.

1. De acuerdo a los dos motores de BD que usted ha elegido, al finalizar esta actividad usted debe estar ne la capacidad de diligenciar la siguiente tabla:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OPERACIÓN SQL | MySQL | MariaDB | PostgreSQL | SQLServer |
| Entrar a la terminal | mysql -u usuario -p | mysql -u usuario -p | psql -U usuario | SQL Server Management Studio (SSMS) o sqlcmd |
| Crear un usuario | CREATE USER 'nombre'@'host' IDENTIFIED BY 'contraseña'; | CREATE USER 'nombre'@'host' IDENTIFIED BY 'contraseña'; | CREATE USER nombre WITH PASSWORD 'contraseña'; | CREATE LOGIN nombre WITH PASSWORD = 'contraseña'; |
| Asignar permisos a un usuario | GRANT ALL PRIVILEGES ON base\_de\_datos.\* TO 'nombre'@'host'; | GRANT ALL PRIVILEGES ON base\_de\_datos.\* TO 'nombre'@'host'; | GRANT ALL PRIVILEGES ON DATABASE base\_de\_datos TO nombre; | GRANT CONTROL ON DATABASE::base\_de\_datos TO nombre; |
| Cambiar la clave a un usuario | ALTER USER 'nombre'@'host' IDENTIFIED BY 'nueva\_contraseña'; | ALTER USER 'nombre'@'host' IDENTIFIED BY 'nueva\_contraseña'; | ALTER USER nombre WITH PASSWORD 'nueva\_contraseña'; | ALTER LOGIN nombre WITH PASSWORD = 'nueva\_contraseña'; |
| Quitar los permisos a un usuario | REVOKE ALL PRIVILEGES ON base\_de\_datos.\* FROM 'nombre'@'host'; | REVOKE ALL PRIVILEGES ON base\_de\_datos.\* FROM 'nombre'@'host'; | REVOKE ALL PRIVILEGES ON DATABASE base\_de\_datos FROM nombre; | REVOKE CONTROL ON DATABASE::base\_de\_datos FROM nombre; |
| Eliminar un usuario | DROP USER 'nombre'@'host'; | DROP USER 'nombre'@'host'; | DROP USER nombre; | DROP LOGIN nombre; |
| Mostrar los usuarios activos en el Motor de BD | SELECT user FROM mysql.user; | SELECT user FROM mysql.user; | \du | SELECT name FROM sys.syslogins; |
| Salir de la terminal | exit | exit | \q | exit o quit |
| Mostrar las BD | SHOW DATABASES; | SHOW DATABASES; | \l o SELECT datname FROM pg\_database; | SELECT name FROM sys.databases; |
| Entrar o usar una BD | USE base\_de\_datos; | USE base\_de\_datos; | \c base\_de\_datos | USE base\_de\_datos; |
| Eliminar una BD | DROP DATABASE base\_de\_datos; | DROP DATABASE base\_de\_datos; | DROP DATABASE base\_de\_datos; | DROP DATABASE base\_de\_datos; |
| Crear una tabla | CREATE TABLE tabla (...); | CREATE TABLE tabla (...); | CREATE TABLE tabla (...); | CREATE TABLE tabla (...); |
| Mostrar la estructura de una tabla | DESCRIBE tabla; o SHOW COLUMNS FROM tabla; | DESCRIBE tabla; o SHOW COLUMNS FROM tabla; | \d tabla | sp\_help 'tabla'; |
| Cambiar el nombre de una tabla | RENAME TABLE tabla TO nueva\_tabla; | RENAME TABLE tabla TO nueva\_tabla; | ALTER TABLE tabla RENAME TO nueva\_tabla; | sp\_rename 'tabla', 'nueva\_tabla'; |
| Agregar una columna a una tabla | ALTER TABLE tabla ADD columna tipo; | ALTER TABLE tabla ADD columna tipo; | ALTER TABLE tabla ADD columna tipo; | ALTER TABLE tabla ADD columna tipo; |
| Cambiar el tipo de datos a una columna | ALTER TABLE tabla MODIFY columna nuevo\_tipo; | ALTER TABLE tabla MODIFY columna nuevo\_tipo; | ALTER TABLE tabla ALTER COLUMN columna TYPE nuevo\_tipo; | ALTER TABLE tabla ALTER COLUMN columna nuevo\_tipo; |
| Cambiar el nombre de una columna | ALTER TABLE tabla CHANGE columna nuevo\_nombre tipo; | ALTER TABLE tabla CHANGE columna nuevo\_nombre tipo; | ALTER TABLE tabla RENAME COLUMN columna TO nuevo\_nombre; | sp\_rename 'tabla.columna', 'nuevo\_nombre', 'COLUMN'; |
| Eliminar una columna | ALTER TABLE tabla DROP COLUMN columna; | ALTER TABLE tabla DROP COLUMN columna; | ALTER TABLE tabla DROP COLUMN columna; | ALTER TABLE tabla DROP COLUMN columna; |
| Tipos de datos Numéricos Enteros | INT, TINYINT, SMALLINT | INT, TINYINT, SMALLINT | INTEGER, SMALLINT, BIGINT | INT, SMALLINT, BIGINT |
| Tipos de datos numéricos reales o decimales | FLOAT, DOUBLE, DECIMAL | FLOAT, DOUBLE, DECIMAL | NUMERIC, REAL, DOUBLE PRECISION | FLOAT, REAL, DECIMAL |
| Tipos de datos caracteres de longitud fija | CHAR(n) | CHAR(n) | CHAR(n) | CHAR(n) |
| Tipos de datos caracteres de longitud variable | VARCHAR(n) | VARCHAR(n) | VARCHAR(n) | VARCHAR(n) |
| Tipos de datos Boolean | TINYINT(1) o BOOL | TINYINT(1) o BOOL | BOOLEAN | BIT |
| Tipos de datos Fecha | DATE, DATETIME, TIMESTAMP | DATE, DATETIME, TIMESTAMP | DATE, TIMESTAMP | DATE, DATETIME |
| Tipos de datos para guardar datos binarios grandes | BLOB | BLOB | BYTEA | VARBINARY, IMAGE |
| Tipos de datos conjunto | SET | SET | No soportado | No soportado |
| Tipos de datos enumeración | ENUM | ENUM | No soportado | No soportado |
| Columnas con valor autoincremental, secuencial o serial | AUTO\_INCREMENT | AUTO\_INCREMENT | SERIAL | IDENTITY |
| Eliminar una tabla | DROP TABLE tabla; | DROP TABLE tabla; | DROP TABLE tabla; | DROP TABLE tabla; |
| Agregar una Llave foránea al crear la tabla | CONSTRAINT fk\_name FOREIGN KEY (columna) REFERENCES otra\_tabla(columna); | CONSTRAINT fk\_name FOREIGN KEY (columna) REFERENCES otra\_tabla(columna); | FOREIGN KEY (columna) REFERENCES otra\_tabla(columna); | FOREIGN KEY (columna) REFERENCES otra\_tabla(columna); |
| Agregar una llave foránea después de creada la tabla | ALTER TABLE tabla ADD CONSTRAINT fk\_name FOREIGN KEY (columna) REFERENCES otra\_tabla(columna); | ALTER TABLE tabla ADD CONSTRAINT fk\_name FOREIGN KEY (columna) REFERENCES otra\_tabla(columna); | ALTER TABLE tabla ADD FOREIGN KEY (columna) REFERENCES otra\_tabla(columna); | ALTER TABLE tabla ADD CONSTRAINT fk\_name FOREIGN KEY (columna) REFERENCES otra\_tabla(columna); |
| Crear un índice | CREATE INDEX idx\_name ON tabla(columna); | CREATE INDEX idx\_name ON tabla(columna); | CREATE INDEX idx\_name ON tabla(columna); | CREATE INDEX idx\_name ON tabla(columna); |
| Mostrar los índices creados | SHOW INDEX FROM tabla; | SHOW INDEX FROM tabla; | \di o SELECT \* FROM pg\_indexes WHERE tablename = 'tabla'; | sp\_helpindex 'tabla'; |
| Mostrar las restricciones de llave foránea de una tabla | SHOW CREATE TABLE tabla; | SHOW CREATE TABLE tabla; | SELECT conname FROM pg\_constraint WHERE conrelid = 'tabla'::regclass; | SELECT \* FROM sys.foreign\_keys WHERE parent\_object\_id = OBJECT\_ID('tabla'); |
| Eliminar una restricción de llave foránea o cualquier otra restricción | ALTER TABLE tabla DROP FOREIGN KEY fk\_name; | ALTER TABLE tabla DROP FOREIGN KEY fk\_name; | ALTER TABLE tabla DROP CONSTRAINT fk\_name; | ALTER TABLE tabla DROP CONSTRAINT fk\_name; |
| Hacer un backup | mysqldump -u usuario -p base\_de\_datos > archivo.sql | mysqldump -u usuario -p base\_de\_datos > archivo.sql | pg\_dump base\_de\_datos > archivo.sql | BACKUP DATABASE base\_de\_datos TO DISK = 'archivo.bak'; |
| Restaurar la BD | mysql -u usuario -p base\_de\_datos < archivo.sql | mysql -u usuario -p base\_de\_datos < archivo.sql | psql -U usuario -d base\_de\_datos -f archivo.sql | RESTORE DATABASE base\_de\_datos FROM DISK = 'archivo.bak'; |