### **OBSERVACIÓN IMPORTANTE:**

Si estás pensando en utilizar la IA para que te resuelva el ejercicio, es muy probable que lo logre, quizás incluso obtengas una nota alta, apruebes la asignatura, y tal vez te gradúes como Ingeniero de Software de la grandiosa UdeC. Sin embargo, al final del camino, nada de eso te servirá para avanzar como profesional ni para alcanzar tus verdaderos propósitos de vida.

Cuando decides usar la IA para realizar tus procesos básicos de aprendizaje, (lo mínimo que debes hacer como ser humano que aprende, progresa, piensa, realiza, construye, innova, resuelve, etc) creas una ilusión que solo te engaña a ti mismo, entonces, es el momento de reflexionar y preguntarte si realmente vale la pena seguir perdiendo años construyendo una burbuja que inevitablemente explotará. Y cuando eso ocurra, te enfrentarás a la realidad: esos cinco años de tu vida jamás los podrás recuperar, y el fracaso será una consecuencia directa de las decisiones equivocadas que tomaste.

Es importante que consideres qué tipo de profesional y persona quieres ser, y si estás dispuesto a invertir en tu desarrollo verdadero, en lugar de depender de atajos que sólo retrasarán tu éxito genuino.

**OBSERVACIONES**:

##### 1. Actividad Individual

1. Cada estudiante (sin excepción alguna) debe subir la actividad desde su respectiva cuenta en sima-pesad
2. Solo se evaluará y calificará a los estudiantes que hayan subido la actividad única y exclusivamente por el aula virtual de BD-1 de SIMA-PESAS, los alumnos que no entreguen la actividad serán evaluados con nota 1.0.

**FORMATO DE ENTREGA:**

Un archivo comprimido en formato ZIP **(NO USAR OTRO FORMATO) - NO SE ACEPTA**

**Nombre del archivo**:

codigo\_nombre\_apellido\_\_bd-1\_act\_1\_c**.zip**

**Por ejemplo**:

1234567\_john\_arrieta\_bd-1\_act\_1\_2024\_4.**zip**

El archivo comprimido debe contener como mínimo los siguientes archivos.

■ Documento en formato EDITABLE .DOC (**NO ES PDF**) con el d**esarrollo y explicación detallada** del desarrollo de la actividad.

■ El desarrollo y **entrega del trabajo debe contener todos los ítem** típicos y característicos de un trabajo académico.

1. Presentación

2. Tabla de contenido

3. Introducción

4. Objetivos

5. Justificación

6. Desarrollo

1. Título y Enunciado del ejercicio.
2. Desarrollo de los pasos del 1 al 10 descritos en la siguiente sección
3. Debe ser lo suficientemente explicativo, argumentativo e ilustrativo.

7. Bibliografía

Archivo comprimido con los códigos (archivo de imagen del DER, Archivo imagen del MR, documento (informe de laboratorio) con la descripción detallada, muy bien argumentada e ilustrativa sobre el desarrollo de la actividad.

**Normas APA para la elaboración de esta actividad**

<https://www.colconectada.com/normas-apa/>

**DESCRIPCIÓN DE LA ACTIVIDAD**

Realizar un detallado análisis del enunciado del ejercicio que le fue asignado para que proceda a desarrollar los siguientes puntos:

1. Identificar conjunto de entidades entidades y sus atributos

ENTIDAD\_A (atributo1, atributo2, atributo3, atributo4, atributon)

ENTIDAD\_B (atributo1, atributo2, atributo3, atributo4, atributon)

ENTIDAD\_N (atributo1, atributo2, atributo3, atributo4, atributon)

**…**

1. Identificar interrelaciones entre conjunto de entidades

ENTIDAD\_A ->---<R1>----<- ENTIDAD\_B

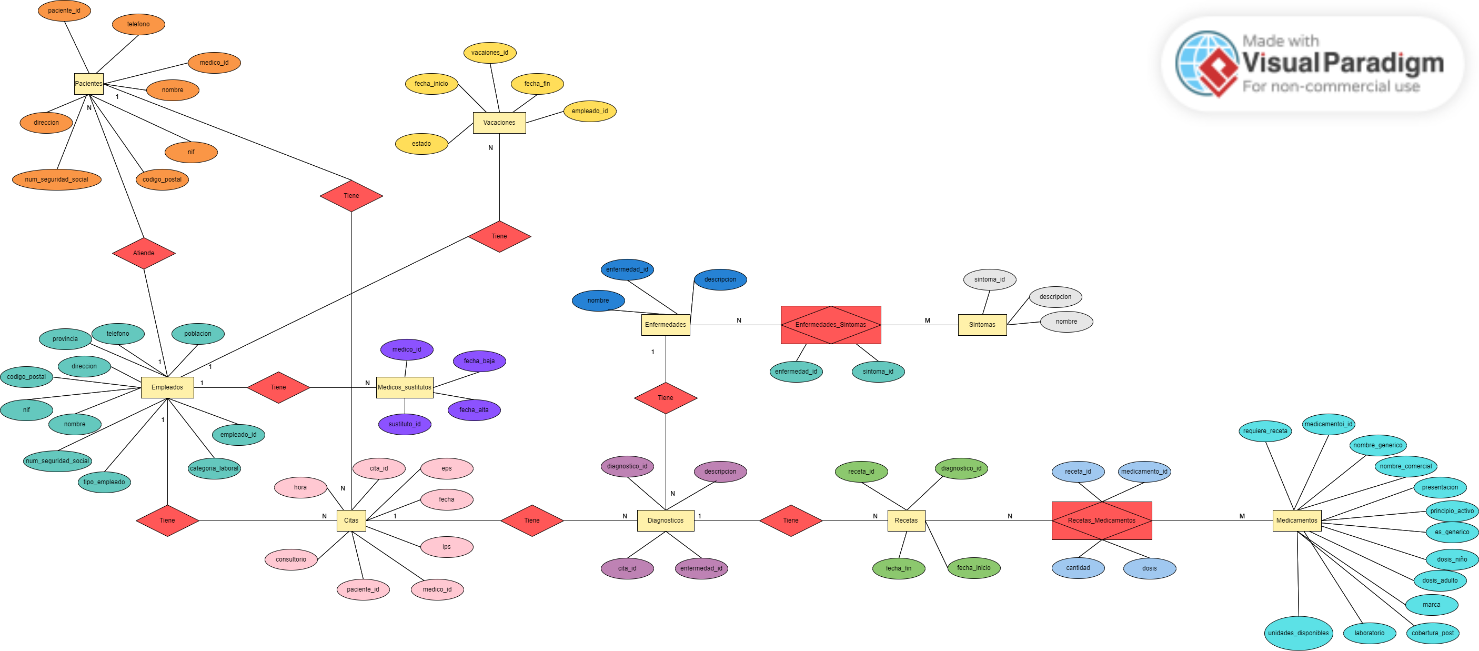
ENTIDAD\_D --|---<R2>----<- ENTIDAD\_C

ENTIDAD\_B --|---<R3>----|-- ENTIDAD\_D

**…**

1. **Diseñar los respectivos Diagramas Entidad Relación (DER)**
   1. Adjuntar la imagen como del DER en esta sección del documento y adjuntar como archivo independiente o colocar enlace para descargarla o visualizarla desde internet.

[Diagrama entidad-relacion](https://drive.google.com/file/d/13UfZu5zTzqm61Wo1CnY4uzYhBDX94yBo/view?usp=sharing)



Tomar como base los Diagramas Entidad Relación y realizar los siguientes siguientes puntos:

1. Transformar el **DER** a **Modelo Relacional (MR)**.

Por cada conjunto de entidades en el DER se debe crear una Relación en el MR

[**RELACION\_A**] (campo1, campo2, campo3, campo4, campoN)

[**RELACION\_B**] (campo1, campo2, campo3, campo4, campoN)

[**RELACION\_C**] (campo1, campo2, campo3, campo4, campoN

**…**

1. Identificar los atributos (**Multivaluados** o **Compuestos**) en el DER y transformarlos en Relaciones del MR.

[**RELACION\_X**](campo1, campo2, campo3, campo4, campoN)

[**RELACION\_Y**](campo1, campo2, campo3, campo4, campoN)

**…**

1. Identificar las **llaves candidatas** para cada Relación en MR a partir de los atributos identificativos en el DER, luego determinar cual o cuales campos del cada Relacion del MAR serán definidos como Llave primaria

[**RELACION\_A**](***campo1***, campo2, campo3, ***campo4***, campoN)

[**RELACION\_B**](campo1, ***campo2***, campo3, campo4, campoN)

[**RELACION\_C**](***campo1***, campo2, campo3, campo4, campoN)

**…**

1. Identificar las **llaves foráneas** en el **MR,** a partir de las Interrelaciones de **uno a uno en el DER.**

[**ENTIDADES\_A**]--**|**-----<**IR**>-----**|**--[**ENTIDADES\_B**]

* + - Conectar la Relacion\_A con la Relacion\_B colocando la misma cardinalidad de la interrelación <IR>

[**RELACION\_A**]--**|**----------------**|**--[**RELACION\_B]**

* + - Agregar un nuevo campo al final de la lista de campos de la Relacion\_A. Se recomienda que su nombre inicie con el nombre de la Relacion\_B y termine con la palabra id. Realiza el mismo proceso en la Relacion\_B, pero el nuevo campo corresponde al nombre de la Relacion\_A.
    - Ambos campos representan las llaves foráneas que conectarán una Tupla de la Relacion\_A con otra Tupla de la Relacion\_B y viceversa.

[**RELACION\_A**](**campo1**, campo2, campo3, campoN, *relacionB\_id*)

[**RELACION\_B**](**campo1**, campo2, campo3, campoN, *relacionA\_id*)

**…** repetirel mismo procedimiento en todas las Interrelaciones de uno a uno que existan en el DER.

1. Identificar las **llaves foráneas** en el **MR,** a partir de las Interrelaciones de **uno a muchos en el DER**.

[**ENTIDADES\_A**]--**|**-----<**IR**>-----**|<**-[**ENTIDADES\_B**]

* + - Conectar la Relacion\_A con la Relacion\_B colocando la misma cardinalidad de la interrelación <IR>

[**RELACION\_A**]--**|**----------------**|<**-[**RELACION\_B]**

* + - Agregar un nuevo campo al final de la lista de campos de la Relacion\_B. Se recomienda que su nombre inicie con el nombre de la Relacion\_A y termine con la palabra id. El nuevo campo representa la llave foránea que conectará una Tupla de la Relacion\_B con otra Tupla de la Relacion\_A.

[**RELACION\_A**](**campo1**, campo2, campo3, campoN)

[**RELACION\_B**](**campo1**, campo2, campo3, campoN, *relacionA\_id*)

**…** aplicar el mismo procedimiento en todas las Interrelaciones de uno a muchos que existan en el DER.

1. Identificar las **llaves foráneas** en el **MR,** a partir de las Interrelaciones de **muchos a muchos en el DER**

[**ENTIDADES\_A**]-**>|**---<**IR**(atr1, atr2)>---**|<**-[**ENTIDADES\_B**]

* 1. Crear una nueva Relación. Se recomienda que el nombre sea igual al nombre del Interrelación IR. [**Relacion\_IR**]
  2. Agregar un campo en la [**Relacion\_IR**] por cada atributo que tenga la Interrelacion **<IR>**, solo en el caso que **<IR>** tenga dichos atributos. [**Relacion\_IR**](campo1, campo2)
  3. Agregar un nuevo campo al final de la lista de campos de la [**Relacion\_IR**]. Se recomienda que su nombre inicie con el nombre de la **Relacion\_A** y termine con la palabra id. El nuevo campo representa la llave foránea que conectará una Tupla de la **Relacion\_A** con otra Tupla de la **Relacion\_IR**.

[**Relacion\_IR**](campo1, campo2, *relacionA\_id*).

* 1. Agregar un nuevo campo al final de la lista de campos de la [**Relacion\_IR**]. Se recomienda que su nombre inicie con el nombre de la **Relacion\_B** y termine con la palabra id. El nuevo campo representa la llave foránea que conectará una Tupla de la **Relacion\_B** con otra Tupla de la **Relacion\_IR**.

[**Relacion\_IR**](campo1, campo2, *relacionA\_id, relacionB\_id*).

* 1. La llave candidata para la [**Relacion\_IR**] se compone por el *relacionA\_id* y *relacionB\_id*
  2. Determinar si la llave primera de [**Relacion\_IR**] efectivamente corresponde a la llave candidata, o si es necesario identificar una nueva llave candidata para que esta satisfaga el concepto de llave primaria.
  3. Conectar la **Relacion\_A** con la **Relacion\_IR** con cardinalidad de uno a muchos.

[**RELACION\_A**]--**|**----------------**|<**-[**RELACION\_IR]**

* 1. Conectar la **Relacion\_B** con la **Relacion\_IR** con cardinalidad de uno a muchos.

[**RELACION\_B**]--**|**----------------**|<**-[**RELACION\_IR]**

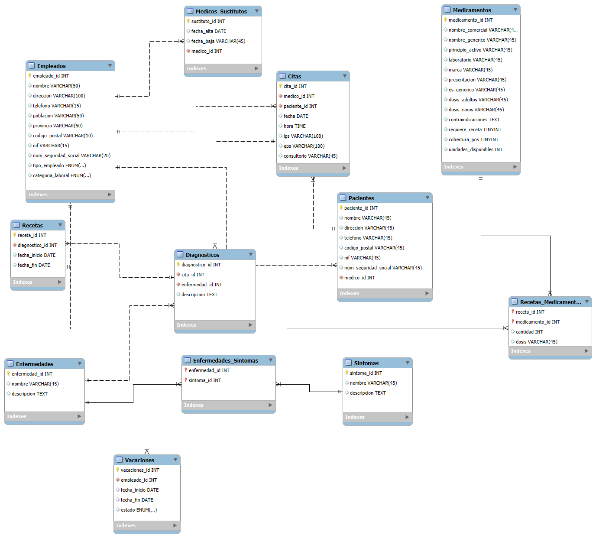
1. Identificar las **llaves foráneas** en el **MR,** a partir de las Interrelaciones de **Generalización |ISA> en DER**

Investigar y aplicar una de las tres formas de hacerlo.

1. Diseñar el respectivo Modelo Relacional (MR) a partir del DER

Adjuntar la imagen del MR en esta sección del documento y adjuntar la imagen como un archivo independiente agregado al archivo .ZIP que debes subir a la plataforma. Opcionalmente puede colocar un enlace para descargar la imagen o visualizarla desde internet.

[Modelo relacional](https://drive.google.com/file/d/1jJuJcs2LWM2HTonlGQDjc4s3d5swKX7U/view?usp=sharing)



**PARTE 2. IMPLEMENTACIÓN DE LA BASE DE DATOS**

1. INSTALAR 3 MOTORES Y SERVIDORES DE BASES DE DATOS RELACIONALES
   1. Instalar Microsoft **SQL Server**
   2. Instalar **Oracle** SQL Server
      1. (Si tiene un PC con pocas capacidades, puede si lo desea usar el siguiente  instalador:

<https://drive.google.com/file/d/136_o1Lsr80Su4fEHqfIwnHG7xr1KPbL7/view?usp=sharing>)

1. Instalar **PostgreSQL** Server
2. Como evidencia puede realizar un documento (o en su defecto un video con cámara y audio donde se vea y escuche tu voz, publicado en youtube (si desea puede ocultarlo) y anexar la URL en este documento) sobre el proceso de instalación paso a paso o idealmente.

[Instalacion de las bases de datos](https://youtu.be/mFFo8aPtxBs?si=4lEn8Oq7l-T38r7Y)

[](https://youtu.be/mFFo8aPtxBs?si=4lEn8Oq7l-T38r7Y)

1. Tomar como base el Modelo Relacional obtenido anteriormente (PARTE 1) a partir del Diagrama entidad Relación las siguientes operaciones:

Tomar como base el o los Diagramas Entidad Relación desarrollados en la Actividad 1 - Unidad 1 y realizar las siguientes operaciones:

1. INSTALAR 3 MOTORES Y SERVIDORES DE BASES DE DATOS RELACIONALES
   1. Instalar Microsoft **SQL Server**
   2. Instalar **Oracle** SQL Server
      1. (Si tiene un PC con pocas capacidades, puede si lo desea usar el siguiente instalador:

<https://drive.google.com/file/d/136_o1Lsr80Su4fEHqfIwnHG7xr1KPbL7/view?usp=sharing>)

* 1. Instalar **PostgreSQL** Server
  2. Como evidencia puede realizar un documento (o en su defecto un video con cámara y audio donde se vea y escuche tu voz, publicado en youtube (si desea puede ocultarlo) y anexar la URL en este documento) sobre el proceso de instalación paso a paso o idealmente.

[Instalacion de las bases de datos](https://youtu.be/mFFo8aPtxBs?si=4lEn8Oq7l-T38r7Y)

[](https://youtu.be/mFFo8aPtxBs?si=4lEn8Oq7l-T38r7Y)

**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

Texto

Descripción generada automáticamente

EXPLICACIÓN.

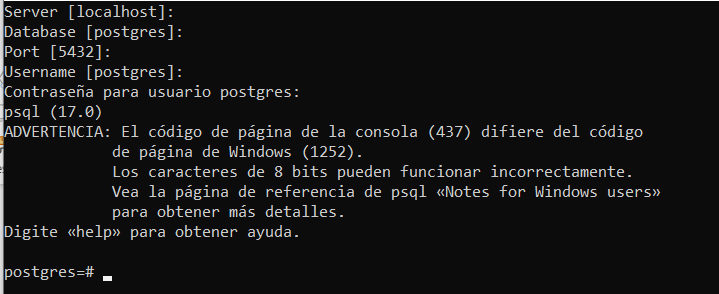
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

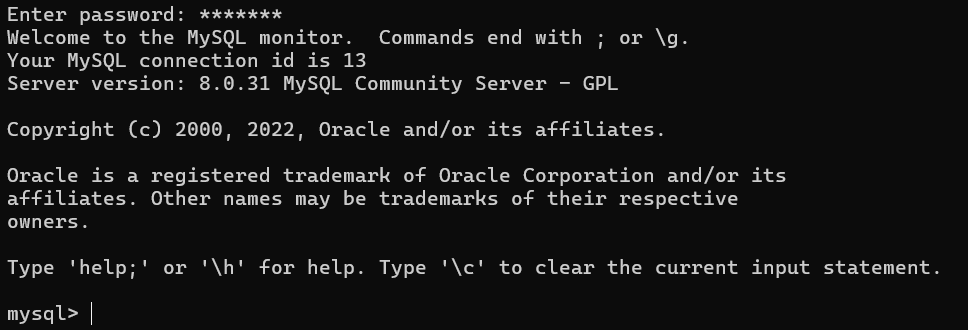
1. **Entrar al Motor de Bases** de datos usando la **Terminal (consola)** de Línea de Comandos usando el usuario creado durante la instrucción o el usuario por defecto con rol de super usuario.

INSTRUCCIÓN SQL:

Ingresando a la terminal de postgres y mysql

RECORTE DE PANTALLA





EXPLICACIÓN.

Ingresando a la terminal de postgres y mysql

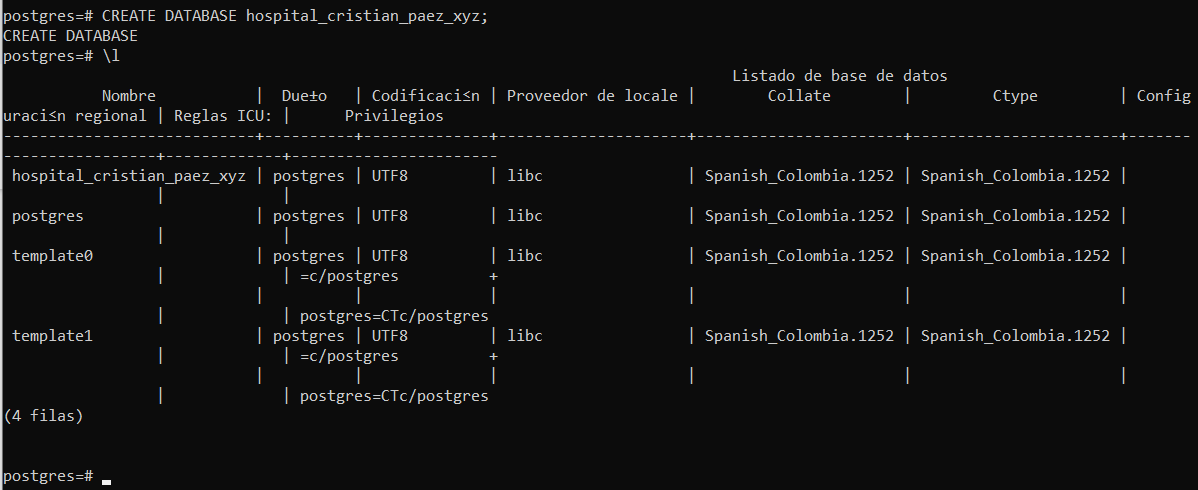
1. **Crear una base de datos** (tablespace en oracle) llamada **bd\_tu\_nombre\_tu\_apellido\_xyz**, donde debe reemplazarse numero\_nombre\_del\_archivo del ejercicio que le fue asignado.

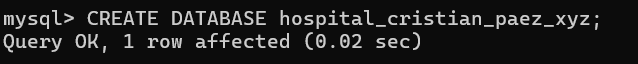
INSTRUCCIÓN SQL:

**CREATE DATABASE hospital\_cristian\_paez\_xyz;**

**\l**

RECORTE DE PANTALLA





EXPLICACIÓN.

Estoy creando la base de datos llamada hospital\_cristian\_paez\_xyz

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

Nombre: **user\_tu\_nombre\_tu\_apellido**

Clave: **AbcdeUdeC**

El nuevo usuario podrá conectarse desde cualquier dirección IP

INSTRUCCIÓN SQL:

**PostgresSQL:**

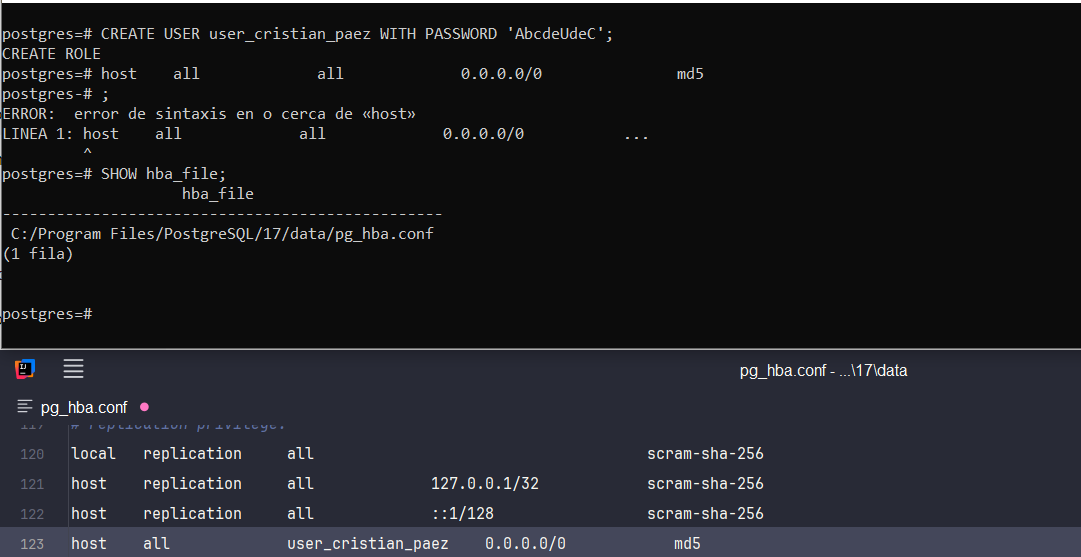
CREATE USER user\_cristian\_paez WITH PASSWORD 'AbcdeUdeC';

SHOW hba\_file;

**MySQL:**

CREATE USER 'user\_cristian\_paez'@'%' IDENTIFIED BY 'AbcdeUdeC';

RECORTE DE PANTALLA





EXPLICACIÓN.

**PostgreSQL**: Lo que hice fue crear un usuario con los comandos que se muestran allí y también le di acceso para que se pueda conectar desde cualquier lado configurando el archivo pg\_hba.conf, haciendo esto:

**host:** Especifica que se permite una conexión de red.

**all**: Se permite el acceso a todas las bases de datos.

**user\_cristian\_paez**: Especifica el nombre del usuario.

**0.0.0.0/0**: Permite conexiones desde cualquier dirección IP.

**md5**: Es el método de autenticación que requiere una contraseña.

**MySQL**: creo el usuario user\_cristian\_paez con la contraseña AbcdeUdeC y permito que se conecte desde cualquier dirección IP (% representa cualquier host)

1. Al nuevo usuario **user\_tu\_nombre\_tu\_apellido** creado anteriormente , se le deben **asignar Privilegios de Superusuario** sobre la base de datos **bd\_tu\_nombre\_tu\_apellido\_xyz**

INSTRUCCIÓN SQL:

**PostgresSQL:**

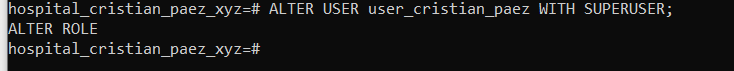
**ALTER USER user\_cristian\_paez WITH SUPERUSER;**

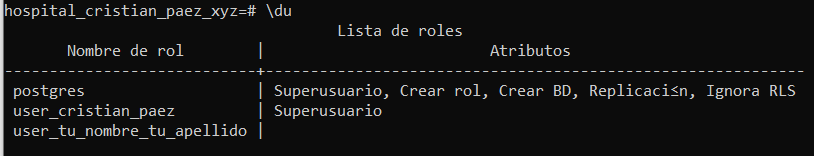
**\du**

**MySQL:**

GRANT ALL PRIVILEGES ON hospital\_cristian\_paez\_xyz.\* TO 'user\_cristian\_paez'@'%';

RECORTE DE PANTALLA







EXPLICACIÓN.

Lo primero que hago es iniciar sesión el base de datos del hospital después, estando ahí le asigno el rol de superusuario al user\_cristian\_paez, tanto en MySQL como en PosgresSQL.

1. **Mostrar los usuarios** existentes

INSTRUCCIÓN SQL:

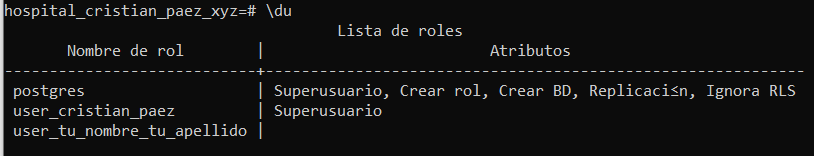
**PosgresSQL:**

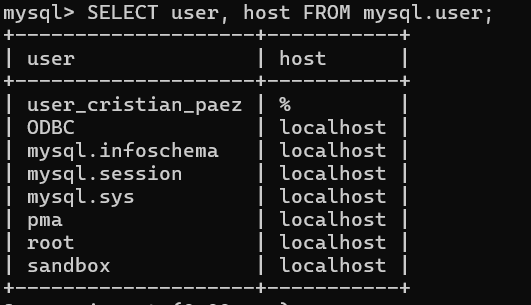
**\du**

**MySQL:**

SELECT user, host FROM mysql.user;

RECORTE DE PANTALLA





EXPLICACIÓN

Con el comando muestro a los usuarios creados con sus respectivos roles en ambas bases de datos.

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

INSTRUCCIÓN SQL:

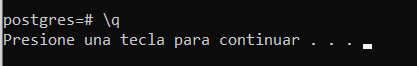
**PostgreSQL:**

**\q**

**MySQL:**

**\quit;**

RECORTE DE PANTALLA





EXPLICACIÓN

Con este comando me salgo de la consola, pulso una tecla y se cierra la consola.

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\_tu\_nombre\_tu\_apellido** creado anteriormente-

INSTRUCCIÓN SQL:

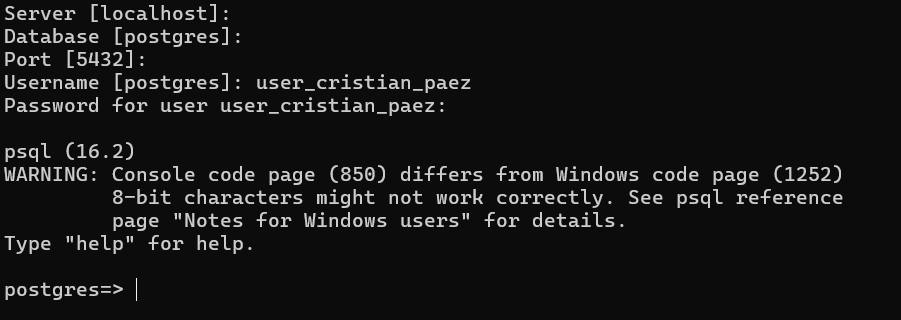
**PostgreSQL:**

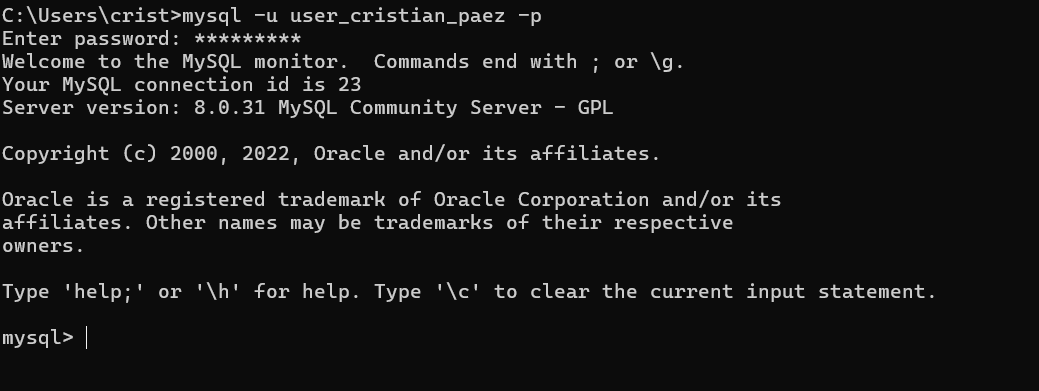
user\_critian\_paez y después la contraseña y listo

**MySQL**:

mysql -u user\_cristian\_paez -p

RECORTE DE PANTALLA





EXPLICACIÓN

Primero en postgres inicio sesión con el usuario y contraseña, en mysql uso y comando en cmd para ingresar con mi usuario y contraseña.

-u especifica el nombre de usuario.

-p indica que pedirá la contraseña después de ejecutar el comando.

Las siguientes operaciones SQL deben ser realizadas utilizando el nuevo usuario **user\_tu\_nombre\_tu\_apellido**.

1. **Crear** una **nueva BD** llamada **bd\_tmp\_tu\_nombre\_tu\_apellido**

INSTRUCCIÓN SQL:

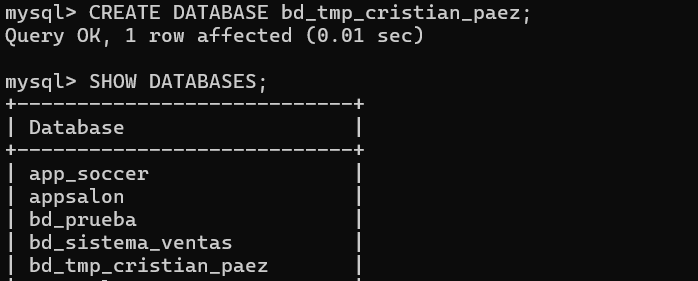
PostgreSQL:

CREATE DATABASE bd\_tmp\_cristian\_paez;

MySQL:

CREATE DATABASE bd\_tmp\_cristian\_paez;

RECORTE DE PANTALLA





EXPLICACIÓN

Se utilizó en ambos casos comandos para crear la base de datos solicitada.

1. **Mostrar** las **Bases de datos** que pertenecen al **user\_tu\_nombre\_tu\_apellido**

INSTRUCCIÓN SQL:

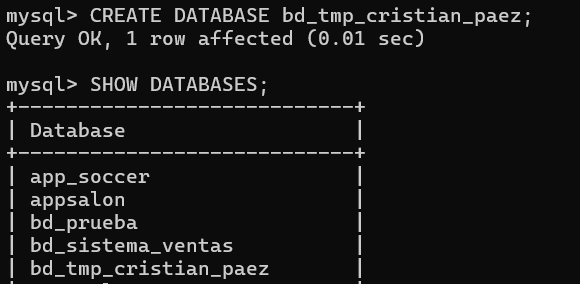
PostgreSQL:

\l

MySQL:

SHOW DATABASES;

RECORTE DE PANTALLA





EXPLICACIÓN

1. Entrar o usar la BD **bd\_tmp\_tu\_nombre\_tu\_apellido**

INSTRUCCIÓN SQL:

**MySQL:**

USE bd\_tmp\_cristian\_paez;

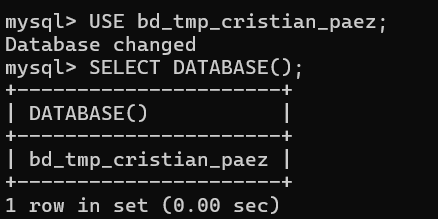
SELECT DATABASE();

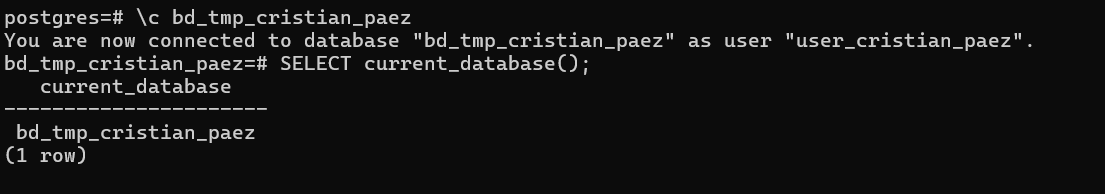
**PostgreSQL:**

\c bd\_tmp\_cristian\_paez

SELECT current\_database();

RECORTE DE PANTALLA





EXPLICACIÓN

Bueno primero en MySQL estoy usando dos comandos el primero me sirve para conectarme a la base de datos y el otro me sirve para verificar esa conexión, de igual manera lo mismo pasa con el comando de PostgreSQL.

1. **Crear una tabla** llamada **bd\_tmp\_tu\_nombre\_tu\_apellido**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:

**MySQL:**

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

);

**PostgreSQL:**

CREATE TABLE tabla\_tempora (

id SERIAL PRIMARY KEY,

columna2 VARCHAR(100) NOT NULL,

columna3 INTEGER,

columna4 REAL,

columna5 DATE,

columna6 TIMESTAMP,

columna7 TIME,

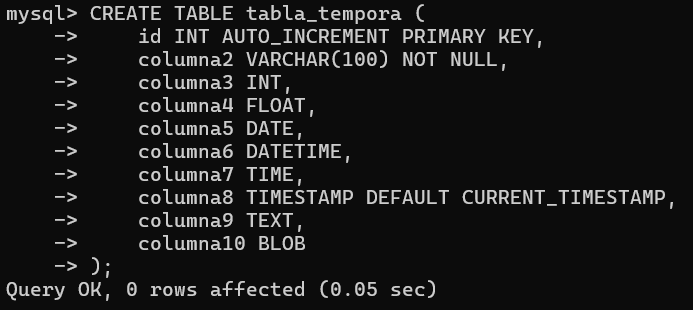
columna8 TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

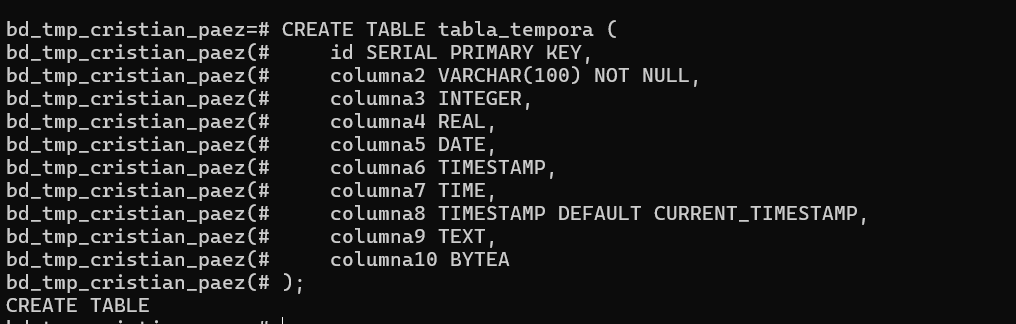
columna9 TEXT,

columna10 BYTEA

);

RECORTE DE PANTALLA





EXPLICACIÓN

Se usa el comando CREATE TABLE en ambos motores de da base de datos para hacer la creación de la misma.

1. **Mostrar las tablas** de la **bd\_tmp\_tu\_nombre\_tu\_apellido**

INSTRUCCIÓN SQL:

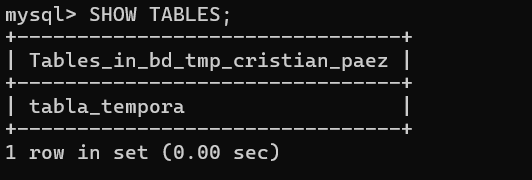
**MySQL:**

SHOW TABLES;

**PostgreSQL:**

\dt

RECORTE DE PANTALLA





EXPLICACIÓN

Estando ya dentro la base de datos con el usuario que ya se ha creado se usa esos comandos para ver las tablas de la base de datos.

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

INSTRUCCIÓN SQL:

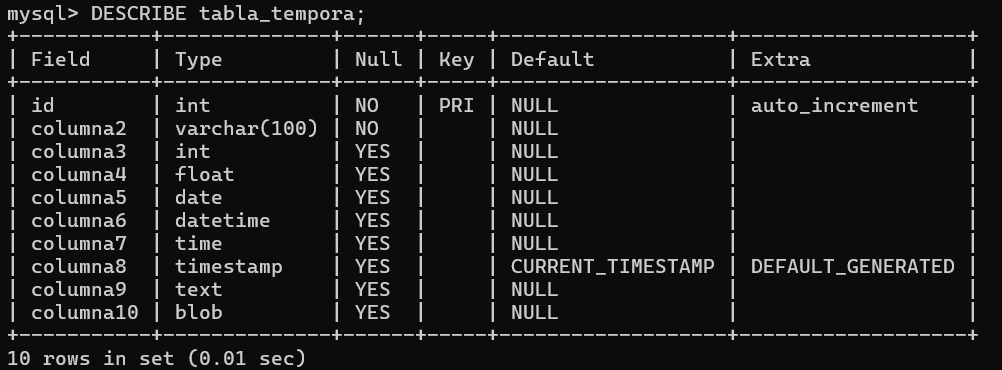
**MySQL:**

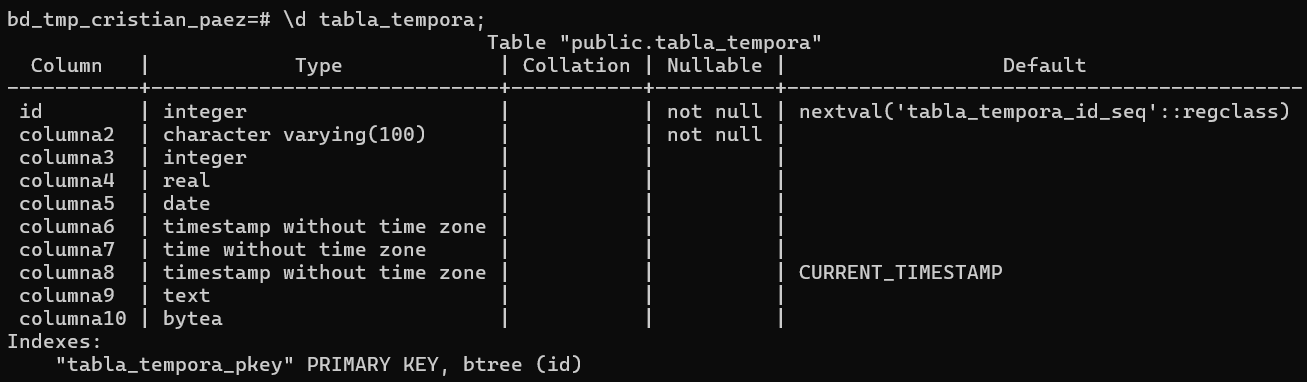
DESCRIBE tabla\_tempora;

**PostgreSQL:**

\d tabla\_tempora;

RECORTE DE PANTALLA





EXPLICACIÓN

Se ejecutaron los comandos mencionados para ver la descripción de la tabla hay que tener en cuenta que hay que estar dentro de la base de datos para poder ejecutar estos comandos.

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

INSTRUCCIÓN SQL:

MySQL:

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

PostgreSQL:

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

RECORTE DE PANTALLA





EXPLICACIÓN

Lo que se hizo fue utilizar este comando para cambiar el nombre de la tabla si se dando e cuenta difieren en muy poco pero se parecen los comandos.

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

INSTRUCCIÓN SQL:

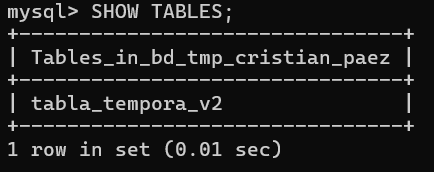
MySQL:

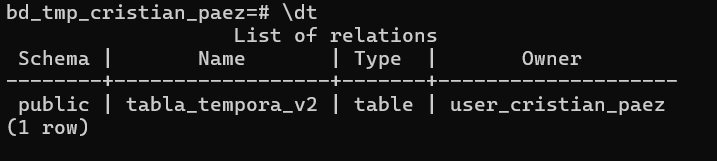
SHOW TABLES;

PostgreSQL:

\dt

RECORTE DE PANTALLA





EXPLICACIÓN

Ambos comandos muestran las tablas de la db\_temporal

1. En la tabla **tabla\_tempora\_v2 cambiar el nombre** de la **culumna4** por **culumnaIV**

INSTRUCCIÓN SQL:

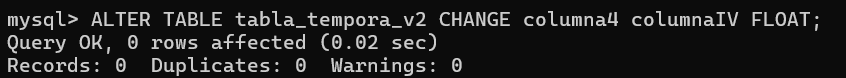
**MySQL:**

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

**PostgreSQL:**

ALTER TABLE tabla\_tempora\_v2 RENAME COLUMN columna4 TO columnaIV;

RECORTE DE PANTALLA





EXPLICACIÓN

Con estos comandos hago el cambio de nombre de una columna en ambos en MySQl y psql.

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

INSTRUCCIÓN SQL:

MySQL:

ALTER TABLE tabla\_tempora\_v2 MODIFY columna7 DATETIME;

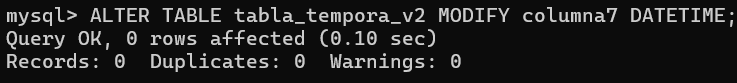
**PostgreSQL:**

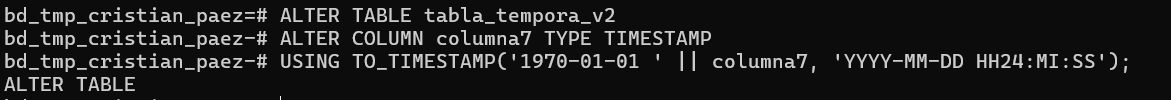
ALTER TABLE tabla\_tempora\_v2

ALTER COLUMN columna7 TYPE TIMESTAMP

USING TO\_TIMESTAMP('1970-01-01 ' || columna7, 'YYYY-MM-DD HH24:MI:SS');

RECORTE DE PANTALLA





EXPLICACIÓN

Con estos comandos tenemos modificada dicha columna tanto en MySQL como en PostgreSQL, en esta consola, se le agrego mas cosas, para que se mas especifico:

TO\_TIMESTAMP: Este es el formato que se usa para combinar una fecha fija (1970-01-01) con el valor de la columna columna7, que contiene la hora, para formar un TIMESTAMP.

'1970-01-01 ' || columna7: Concatenamos la fecha '1970-01-01 ' con el valor de columna7 para que PostgreSQL lo pueda convertir a TIMESTAMP.

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:

**MySQL:**

ALTER TABLE tabla\_tempora\_v2

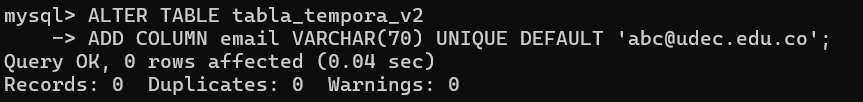
ADD COLUMN email VARCHAR(70) UNIQUE DEFAULT 'abc@udec.edu.co';

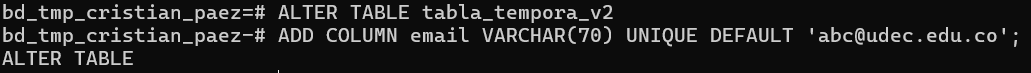
**PostgreSQL:**

ALTER TABLE tabla\_tempora\_v2

ADD COLUMN email VARCHAR(70) UNIQUE DEFAULT 'abc@udec.edu.co';

RECORTE DE PANTALLA





EXPLICACIÓN

En estos dos comandos agregamos dos comandos que nos ayudan agregar dos columnas.

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

INSTRUCCIÓN SQL:

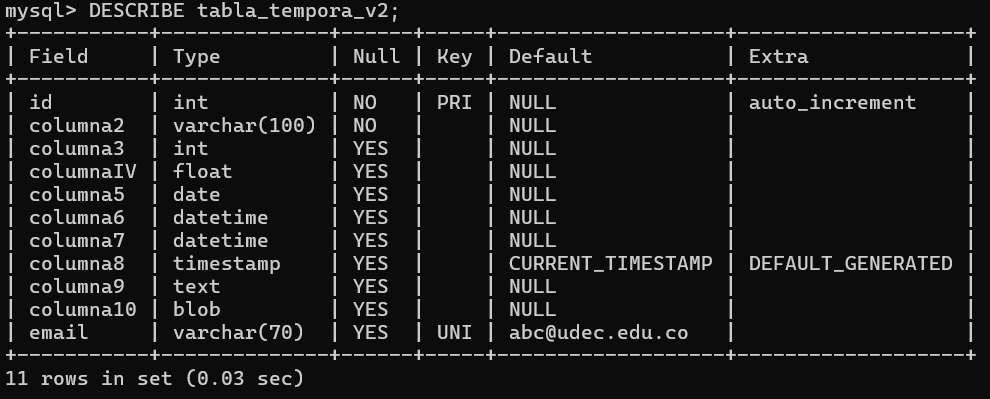
**MySQL:**

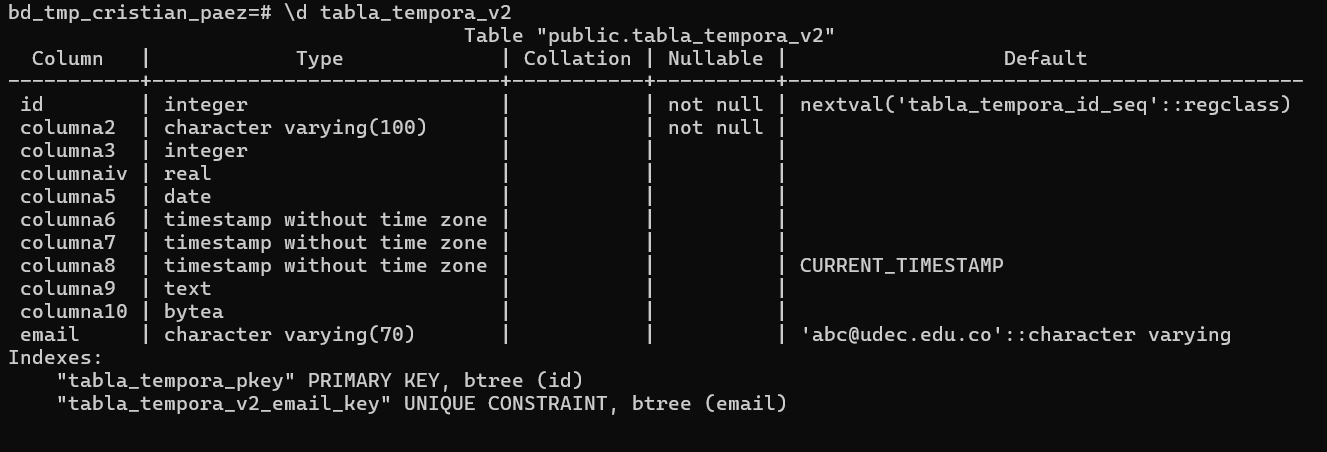
DESCRIBE tabla\_tempora\_v2;

**PostgreSQL:**

\d tabla\_tempora\_v2

RECORTE DE PANTALLA





EXPLICACIÓN

Ambos comandos de las consolas describen lo que hay en las tablas

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

INSTRUCCIÓN SQL:

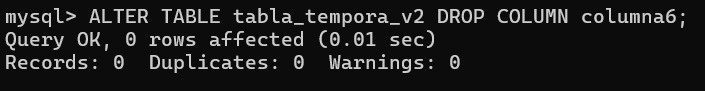
**MySQL:**

ALTER TABLE tabla\_tempora\_v2 DROP COLUMN columna6;

**PostgreSQL:**

ALTER TABLE tabla\_tempora\_v2 DROP COLUMN columna6;

RECORTE DE PANTALLA





EXPLICACIÓN

Con estos comandos de ambos motores se eliminan las columnas correspondientes.

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

INSTRUCCIÓN SQL:

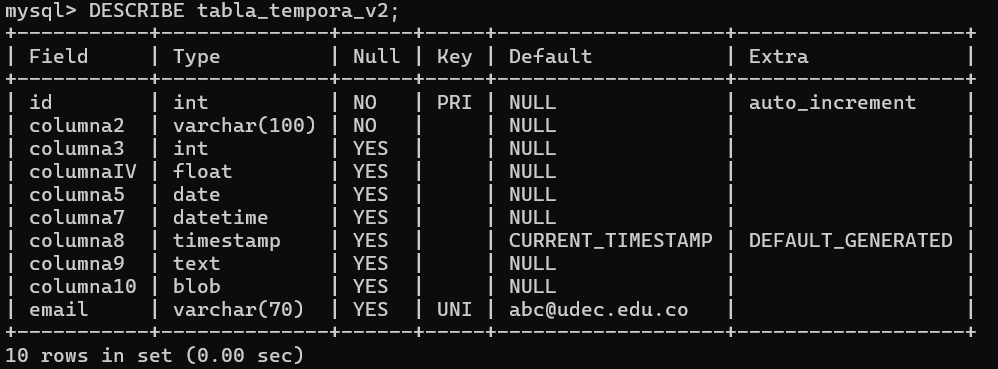
**MySQL:**

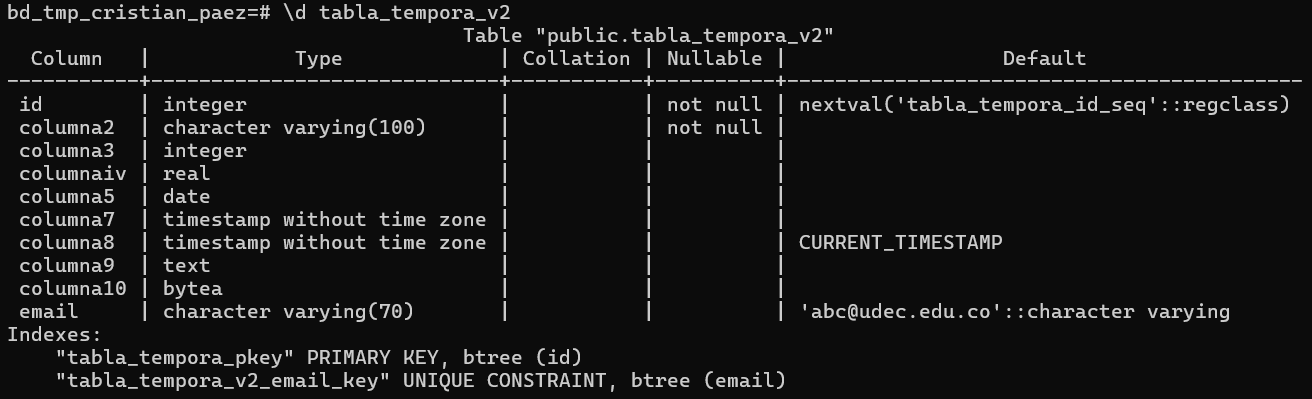
DESCRIBE tabla\_tempora\_v2;

**PostgreSQL:**

\d tabla\_tempora\_v2

RECORTE DE PANTALLA





EXPLICACIÓN

Ambos comandos nos muestran la tabla modificada

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

INSTRUCCIÓN SQL:

**MySQL y PostgreSQL:**

DROP TABLE tabla\_tempora\_v2;

DROP TABLE tabla\_tempora\_v2;

RECORTE DE PANTALLA





EXPLICACIÓN

Ambos comandos sirven para para eliminar la tabla.

1. **Mostrar** las **tablas** de la **bd\_tmp\_tu\_nombre\_tu\_apellido**

INSTRUCCIÓN SQL:

**MySQL:**

SHOW TABLES;

**PostgreSQL:**

\dt

RECORTE DE PANTALLA





EXPLICACIÓN

Ambos muestran que ya no hay nada, porque se elimino.

1. **Cambiarle** el **nombre** de la **base de datos** **bd\_tmp\_tu\_nombre\_tu\_apellido** por **bd\_tmp\_tu\_nombre\_tu\_apellido\_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:

**MySQL**:

CREATE DATABASE bd\_tmp\_cristian\_paez\_v2;

mysqldump -u user\_cristian\_paez -p bd\_tmp\_cristian\_paez > backup.sql

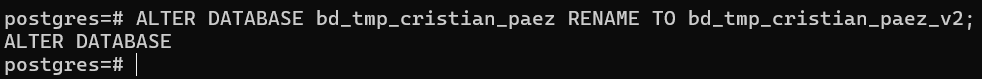
mysql -u user\_cristian\_paez -p bd\_tmp\_cristian\_paez\_v2 < backup.sql

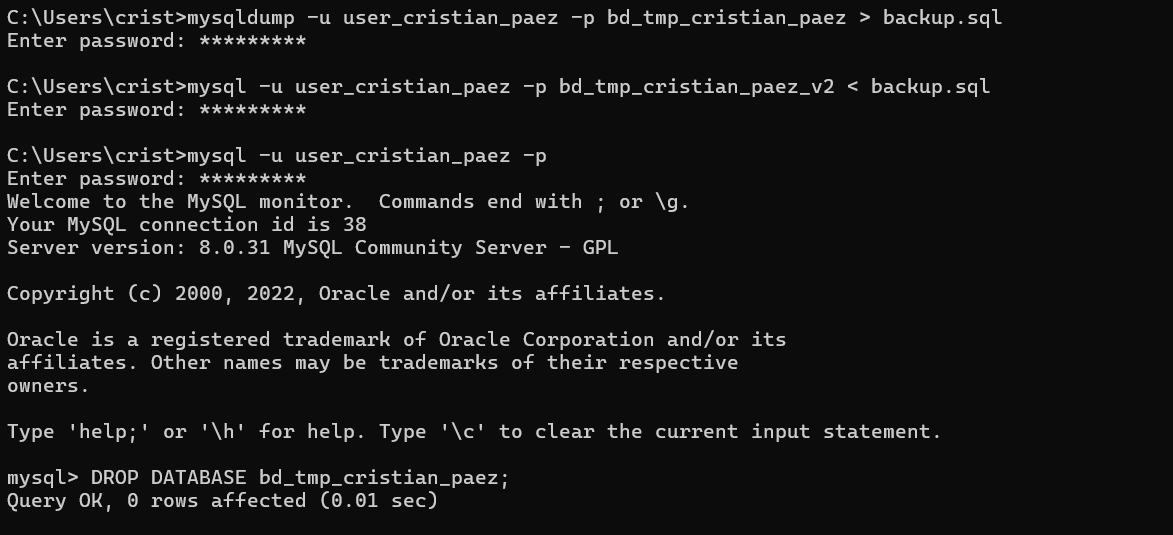
DROP DATABASE bd\_tmp\_cristian\_paez;

PostgreSQL

ALTER DATABASE bd\_tmp\_cristian\_paez RENAME TO bd\_tmp\_cristian\_paez\_v2;

RECORTE DE PANTALLA





EXPLICACIÓN

Empecemos con PostgreSQL que es un comando relativamente fácil, en cuanto a MySQL es mas complicado, ya que hay que seguir unos pasos mas adicionales que no están descritos directamente en la consola de MySQL, primero lo que hacemos es inmigrar los datos para tener el respaldo de los mismos y poderlo usar con la otra base de datos nueva.

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

INSTRUCCIÓN SQL:

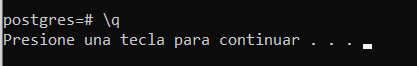
**PostgreSQL:**

**\q**

**MySQL:**

**\quit;**

RECORTE DE PANTALLA





EXPLICACIÓN

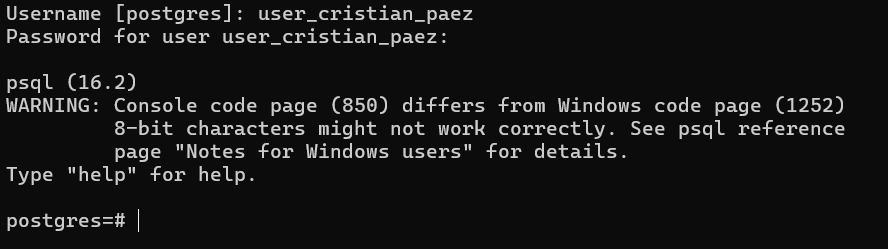
Con este comando me salgo de la consola, pulso una tecla y se cierra la consola.

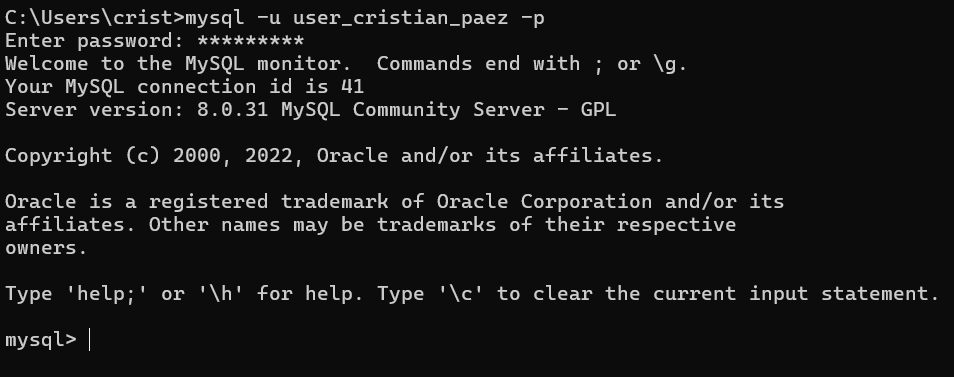
1. Entrar al CLI de la terminal del motor de BD pero ahora utilizando las credenciales del usuario **user\_tu\_nombre\_tu\_apellido**

INSTRUCCIÓN SQL:

Contraseña y el nombre de usuario

RECORTE DE PANTALLA





EXPLICACIÓN

Inicio sesión con user\_cristian\_paez

1. **Mostrar** las **Bases de datos** que pertenecen al **user\_tu\_nombre\_tu\_apellido**

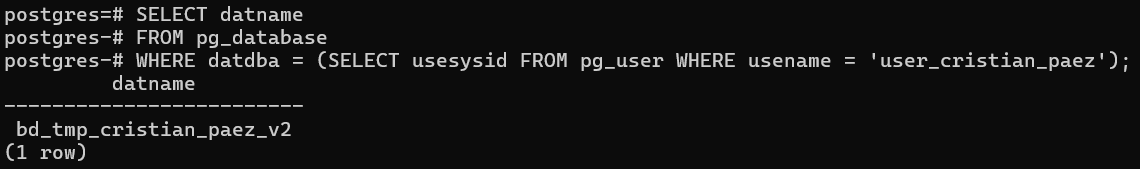
INSTRUCCIÓN SQL:

SELECT datname

FROM pg\_database

WHERE datdba = (SELECT usesysid FROM pg\_user WHERE usename = 'user\_cristian\_paez');

RECORTE DE PANTALLA



EXPLICACIÓN

Con esto podemos consultar la base de datos que tiene asignadas user\_cristian\_paez

1. Eliminar el usuario **user\_tu\_nombre\_tu\_apellido**

INSTRUCCIÓN SQL:

**MySQL:**

DROP USER 'user\_cristian\_paez'@'%';

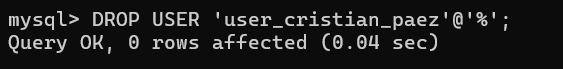
**PostgreSQL:**

ALTER DATABASE bd\_tmp\_cristian\_paez\_v2 OWNER TO postgres;

DROP ROLE user\_cristian\_paez;

RECORTE DE PANTALLA





EXPLICACIÓN

Para MySQL uso un comando que me sirve para eliminar el usuario sin ningún problema, en PostgreSQL hay que hacer un paso primero que es quitarle los roles de la base de datos que tenga asignados para que no haya ningún problema y después si lo eliminamos.

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

INSTRUCCIÓN SQL:

MySQL:

SHOW PROCESSLIST;

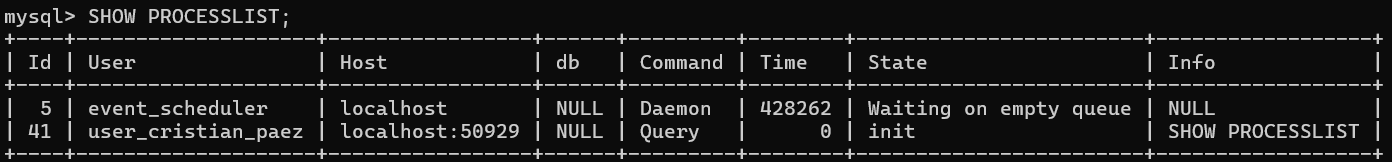
PostgreSQL:

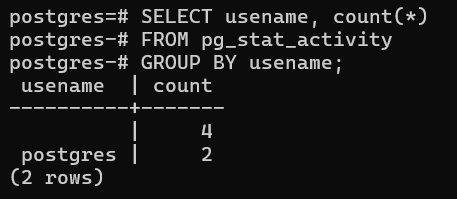
SELECT usename, count(\*)

FROM pg\_stat\_activity

GROUP BY usename;

RECORTE DE PANTALLA





EXPLICACIÓN

Con estas instrucciones podemos ver lo usuarios activos, se pueden ver los usuarios activos, todavía se pueden ver lo usuario que elimine, pero eso es normal porque cuestión de los motores de bases de datos.

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\_nombre\_tu\_apellido\_xyz** creada anteriormente.

INSTRUCCIÓN SQL:

**MySQL:**

USE hospital\_cristian\_paez\_xyz;

**PostgreSQL:**

\c hospital\_cristian\_paez\_xyz

RECORTE DE PANTALLA





EXPLICACIÓN

Con estos comandos nos conectamos a la base de datos

1. **Mostrar** las **tablas** de la **bd\_tu\_nombre\_tu\_apellido\_xyz**

INSTRUCCIÓN SQL:

**MySQL:**

SHOW TABLES;

**PostgreSQL:**

\dt

RECORTE DE PANTALLA





EXPLICACIÓN

Con estos comandos podemos ver las tablas de las bases de datos, pero como tal no hay nada.

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

INSTRUCCIONES SQL PARA CREAR LAS TABLAS:

CREATE TABLE Empleados (

empleado\_id INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(50),

direccion VARCHAR(100),

telefono VARCHAR(15),

poblacion VARCHAR(50),

provincia VARCHAR(50),

codigo\_postal VARCHAR(10),

nif VARCHAR(15),

num\_seguridad\_social VARCHAR(20),

tipo\_empleado ENUM('medico', 'ats', 'auxiliar', 'celador', 'admin'),

categoria\_laboral ENUM('titular', 'interino', 'sustituto') -- Solo para médicos

);

CREATE TABLE Medicos\_Sustitutos (

sustituto\_id INT PRIMARY KEY AUTO\_INCREMENT,

medico\_id INT,

fecha\_alta DATE,

fecha\_baja DATE,

FOREIGN KEY (medico\_id) REFERENCES Empleados(empleado\_id)

);

CREATE TABLE Pacientes (

paciente\_id INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(50),

direccion VARCHAR(100),

telefono VARCHAR(15),

codigo\_postal VARCHAR(10),

nif VARCHAR(15),

num\_seguridad\_social VARCHAR(20),

medico\_id INT,

FOREIGN KEY (medico\_id) REFERENCES Empleados(empleado\_id)

);

CREATE TABLE Citas (

cita\_id INT PRIMARY KEY AUTO\_INCREMENT,

medico\_id INT,

paciente\_id INT,

fecha DATE,

hora TIME,

ips VARCHAR(100), -- Institución Prestadora de Servicios

eps VARCHAR(100), -- Entidad Promotora de Salud

consultorio VARCHAR(50),

FOREIGN KEY (medico\_id) REFERENCES Empleados(empleado\_id),

FOREIGN KEY (paciente\_id) REFERENCES Pacientes(paciente\_id)

);

CREATE TABLE Diagnosticos (

diagnostico\_id INT PRIMARY KEY AUTO\_INCREMENT,

cita\_id INT,

enfermedad\_id INT,

descripcion TEXT,

FOREIGN KEY (cita\_id) REFERENCES Citas(cita\_id),

FOREIGN KEY (enfermedad\_id) REFERENCES Enfermedades(enfermedad\_id)

);

CREATE TABLE Enfermedades (

enfermedad\_id INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(100),

descripcion TEXT

);

CREATE TABLE Sintomas (

sintoma\_id INT PRIMARY KEY AUTO\_INCREMENT,

nombre VARCHAR(100),

descripcion TEXT

);

CREATE TABLE Enfermedades\_Sintomas (

enfermedad\_id INT,

sintoma\_id INT,

PRIMARY KEY (enfermedad\_id, sintoma\_id),

FOREIGN KEY (enfermedad\_id) REFERENCES Enfermedades(enfermedad\_id),

FOREIGN KEY (sintoma\_id) REFERENCES Sintomas(sintoma\_id)

);

CREATE TABLE Medicamentos (

medicamento\_id INT PRIMARY KEY AUTO\_INCREMENT,

nombre\_comercial VARCHAR(100),

nombre\_generico VARCHAR(100),

principio\_activo VARCHAR(100),

laboratorio VARCHAR(100),

marca VARCHAR(100),

presentacion VARCHAR(50),

es\_generico BOOLEAN,

dosis\_adultos VARCHAR(50),

dosis\_ninos VARCHAR(50),

contraindicaciones TEXT,

requiere\_receta BOOLEAN,

cobertura\_pos BOOLEAN,

unidades\_disponibles INT

);

CREATE TABLE Recetas (

receta\_id INT PRIMARY KEY AUTO\_INCREMENT,

diagnostico\_id INT,

fecha\_inicio DATE,

fecha\_fin DATE,

FOREIGN KEY (diagnostico\_id) REFERENCES Diagnosticos(diagnostico\_id)

);

CREATE TABLE Recetas\_Medicamentos (

receta\_id INT,

medicamento\_id INT,

cantidad INT,

dosis VARCHAR(50),

PRIMARY KEY (receta\_id, medicamento\_id),

FOREIGN KEY (receta\_id) REFERENCES Recetas(receta\_id),

FOREIGN KEY (medicamento\_id) REFERENCES Medicamentos(medicamento\_id)

);

CREATE TABLE Vacaciones (

vacaciones\_id INT PRIMARY KEY AUTO\_INCREMENT,

empleado\_id INT,

fecha\_inicio DATE,

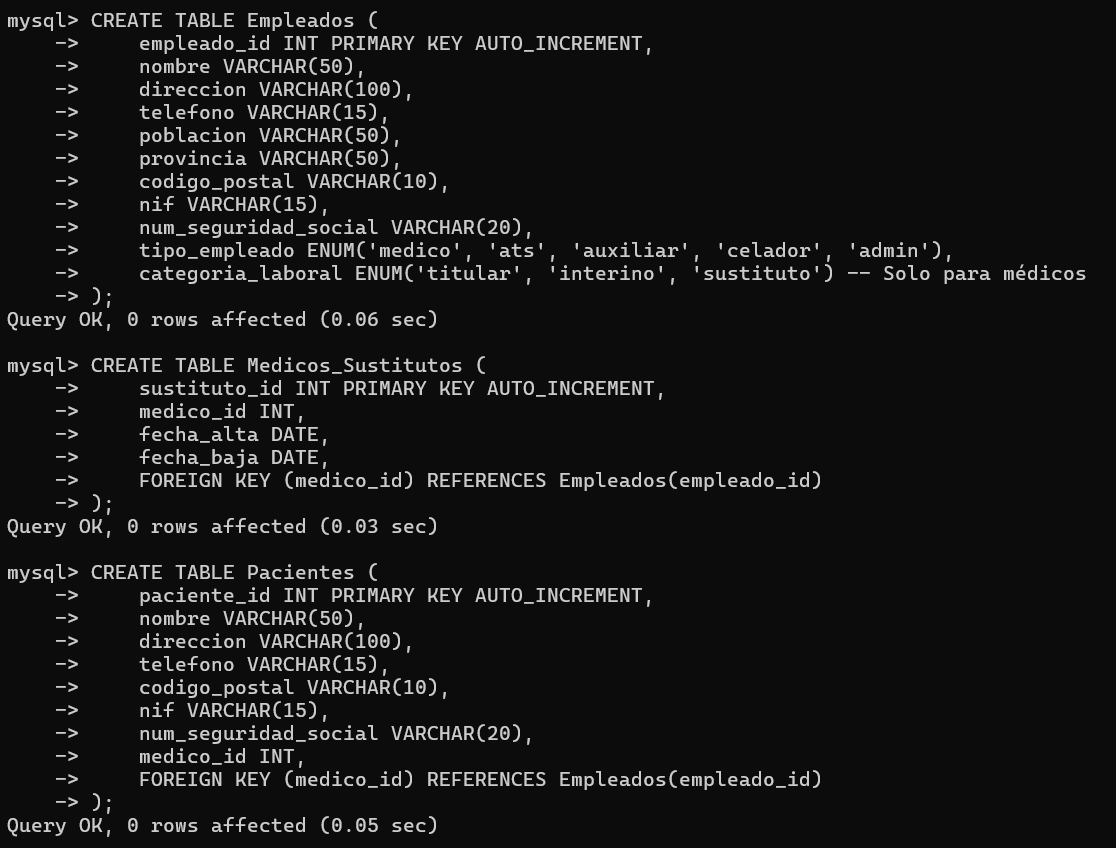
fecha\_fin DATE,

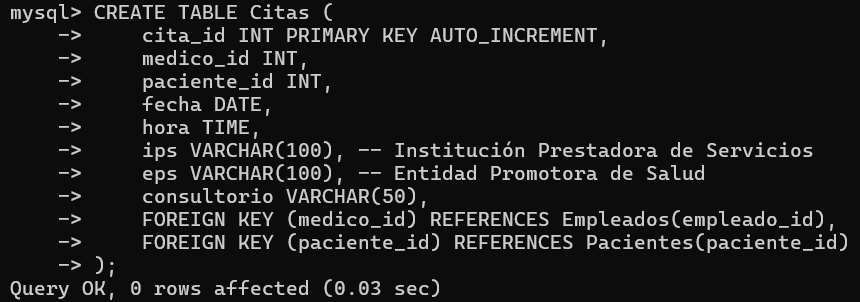
estado ENUM('planificado', 'disfrutado'),

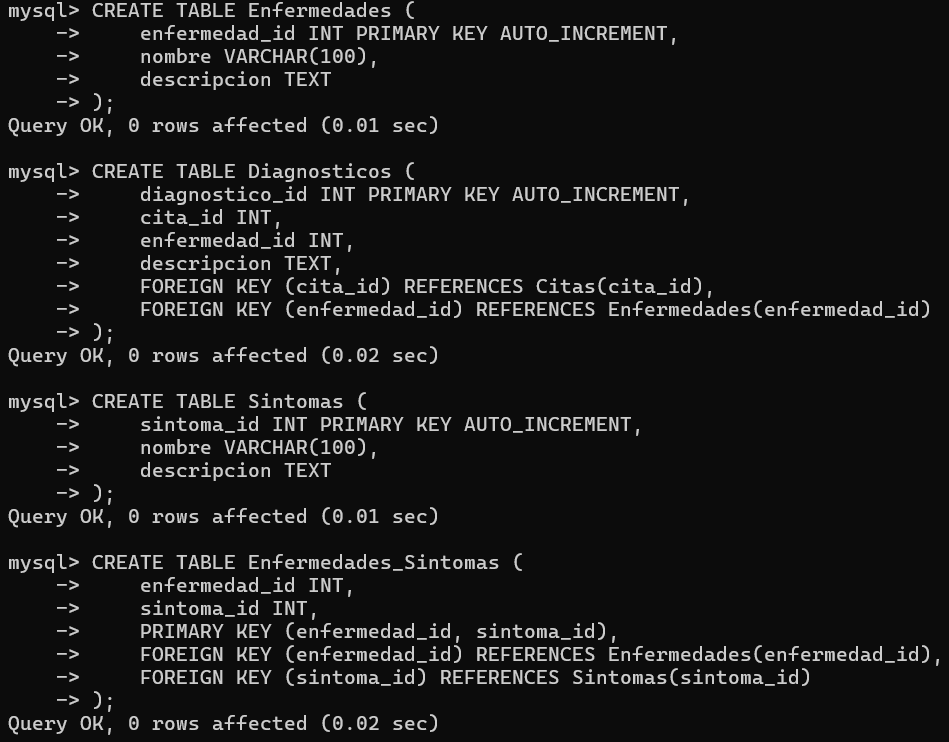
FOREIGN KEY (empleado\_id) REFERENCES Empleados(empleado\_id)

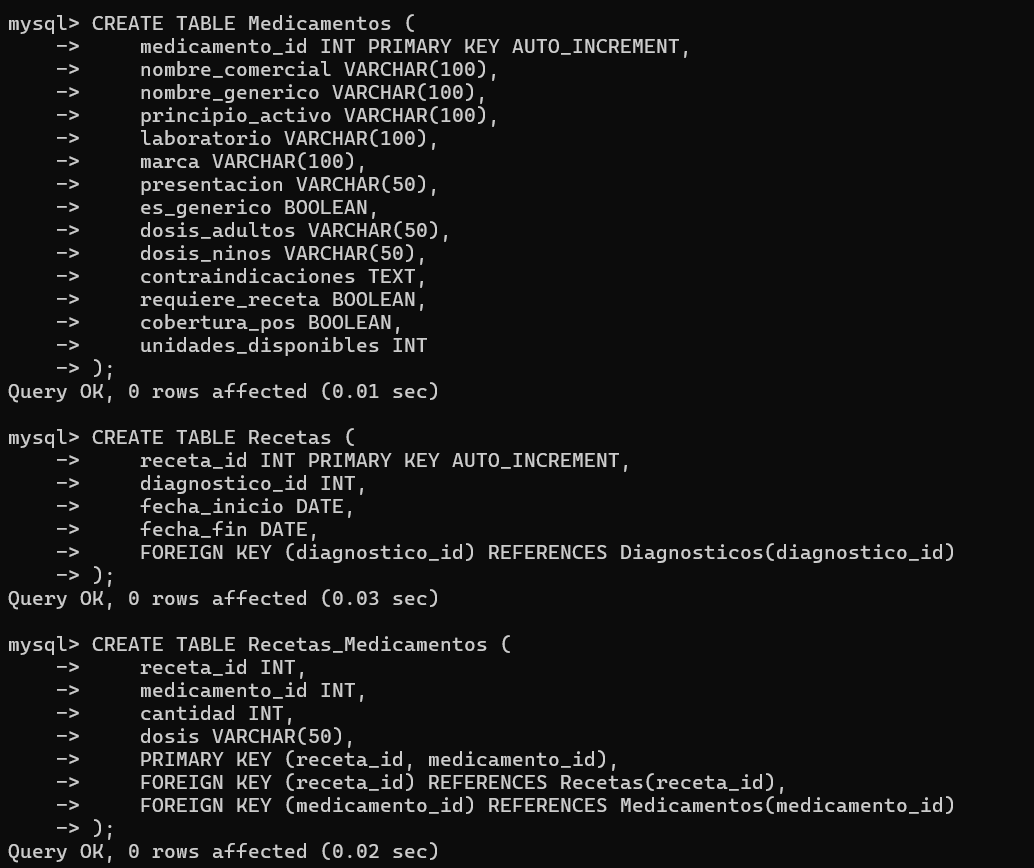
);

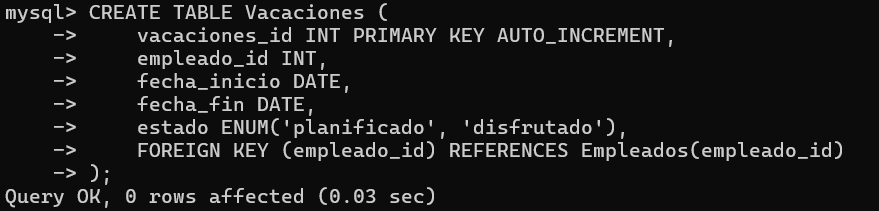
RECORTES DE PANTALLAS











EXPLICACIÓN

Todos estos comandos me ayudaron a crear las tablas de la base de datos

1. **Describir** cada una de las **tablas** creadas creadas en la BD **bd\_tu\_nombre\_tu\_apellido\_xyz**

INSTRUCCIONES SQL PARA DESCRIBIR LAS TABLAS:

DESCRIBE Empleados;

DESCRIBE Medicos\_Sustitutos;

DESCRIBE Pacientes;

DESCRIBE Citas;

DESCRIBE Diagnosticos;

DESCRIBE Enfermedades;

DESCRIBE Sintomas;

DESCRIBE Enfermedades\_Sintomas;

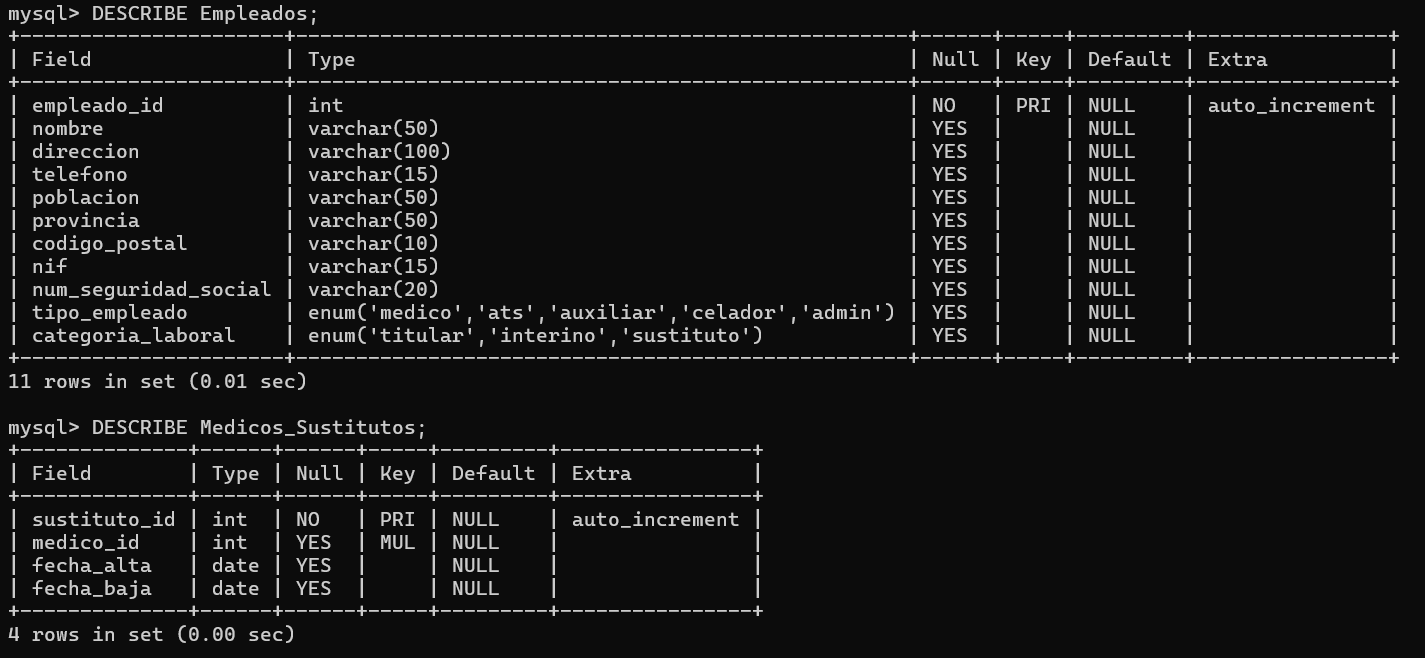
DESCRIBE Medicamentos;

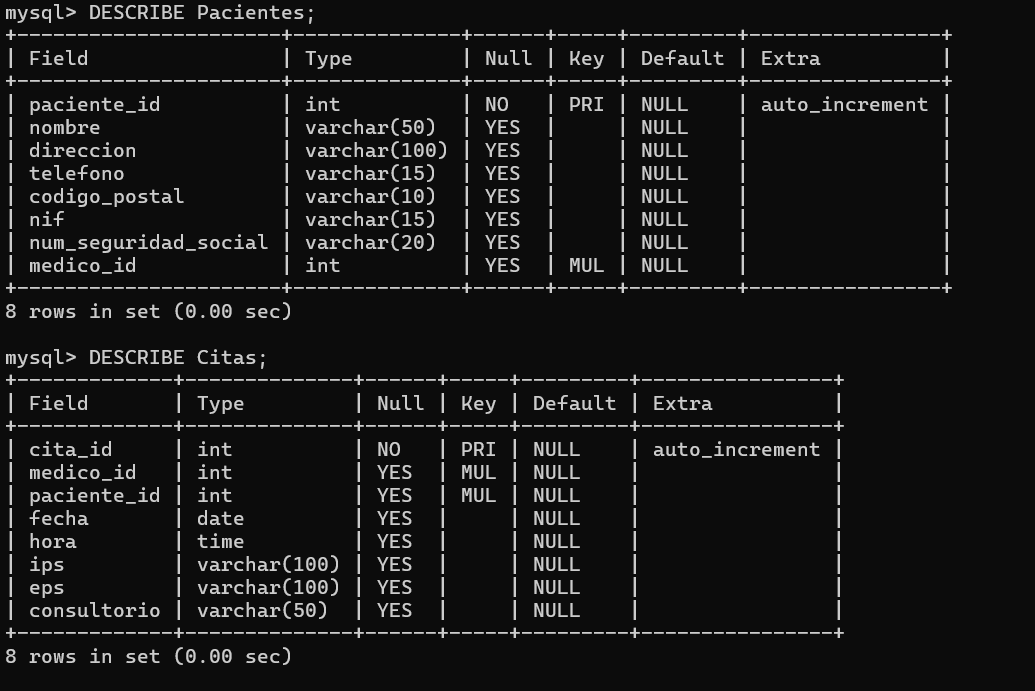
DESCRIBE Recetas

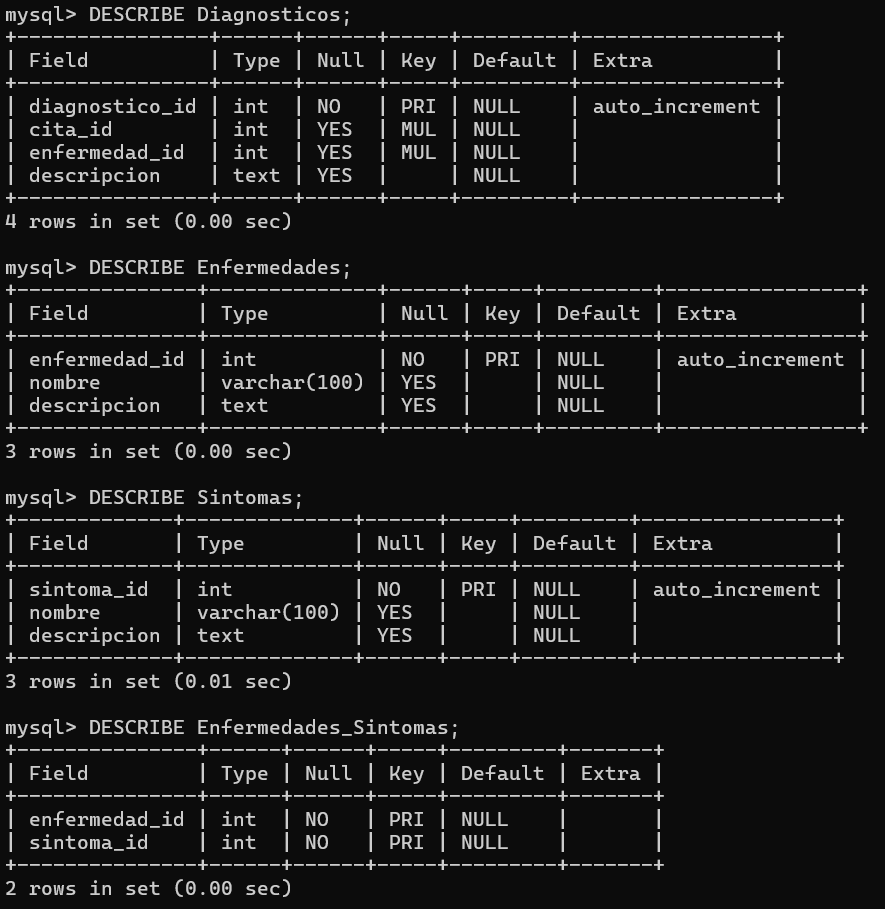
DESCRIBE Recetas\_Medicamentos;

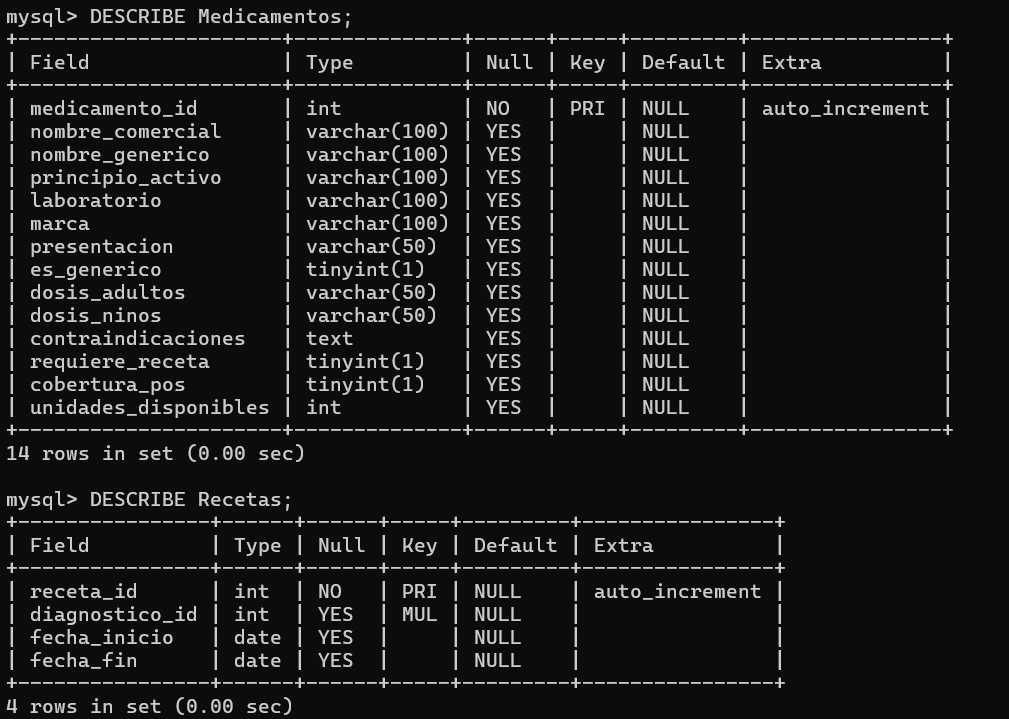
DESCRIBE Vacaciones;

RECORTES DE PANTALLAS











EXPLICACIÓN

Con estos comandos podemos ver cada detalle de las tablas creadas

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:

MySQL:

ALTER TABLE Medicos\_Sustitutos

ADD FOREIGN KEY (medico\_id) REFERENCES Empleados(empleado\_id);

ALTER TABLE Pacientes

ADD FOREIGN KEY (medico\_id) REFERENCES Empleados(empleado\_id);

ALTER TABLE Citas

ADD FOREIGN KEY (medico\_id) REFERENCES Empleados(empleado\_id),

ADD FOREIGN KEY (paciente\_id) REFERENCES Pacientes(paciente\_id);

ALTER TABLE Diagnosticos

ADD FOREIGN KEY (cita\_id) REFERENCES Citas(cita\_id),

ADD FOREIGN KEY (enfermedad\_id) REFERENCES Enfermedades(enfermedad\_id);

ALTER TABLE Recetas

ADD FOREIGN KEY (diagnostico\_id) REFERENCES Diagnosticos(diagnostico\_id);

ALTER TABLE Recetas\_Medicamentos

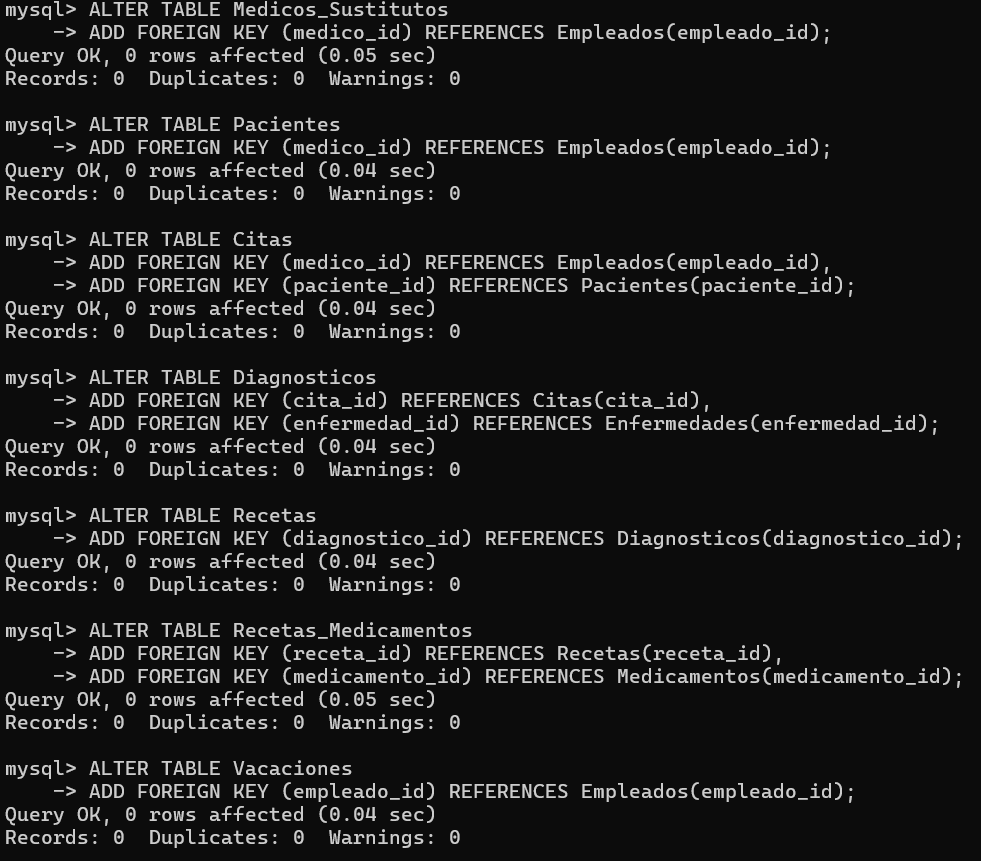
ADD FOREIGN KEY (receta\_id) REFERENCES Recetas(receta\_id),

ADD FOREIGN KEY (medicamento\_id) REFERENCES Medicamentos(medicamento\_id);

ALTER TABLE Vacaciones

ADD FOREIGN KEY (empleado\_id) REFERENCES Empleados(empleado\_id);

RECORTES DE PANTALLAS



EXPLICACIÓN

Todos estos comandos nos sirven para crear las llaves foraneas

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

INSTRUCCIONES SQL PARA CREAR ÍNDICES:

CREATE INDEX idx\_medico\_id ON Medicos\_Sustitutos(medico\_id);

CREATE INDEX idx\_medico\_id\_pacientes ON Pacientes(medico\_id);

CREATE INDEX idx\_medico\_id\_citas ON Citas(medico\_id);

CREATE INDEX idx\_paciente\_id\_citas ON Citas(paciente\_id);

CREATE INDEX idx\_cita\_id\_diagnosticos ON Diagnosticos(cita\_id);

CREATE INDEX idx\_enfermedad\_id\_diagnosticos ON Diagnosticos(enfermedad\_id);

CREATE INDEX idx\_diagnostico\_id\_recetas ON Recetas(diagnostico\_id);

CREATE INDEX idx\_receta\_id\_medicamentos ON Recetas\_Medicamentos(receta\_id);

CREATE INDEX idx\_medicamento\_id\_medicamentos ON Recetas\_Medicamentos(medicamento\_id);

CREATE INDEX idx\_empleado\_id\_vacaciones ON Vacaciones(empleado\_id);

RECORTES DE PANTALLAS



EXPLICACIÓN

Con estos comandos se crean los índices necesarios.

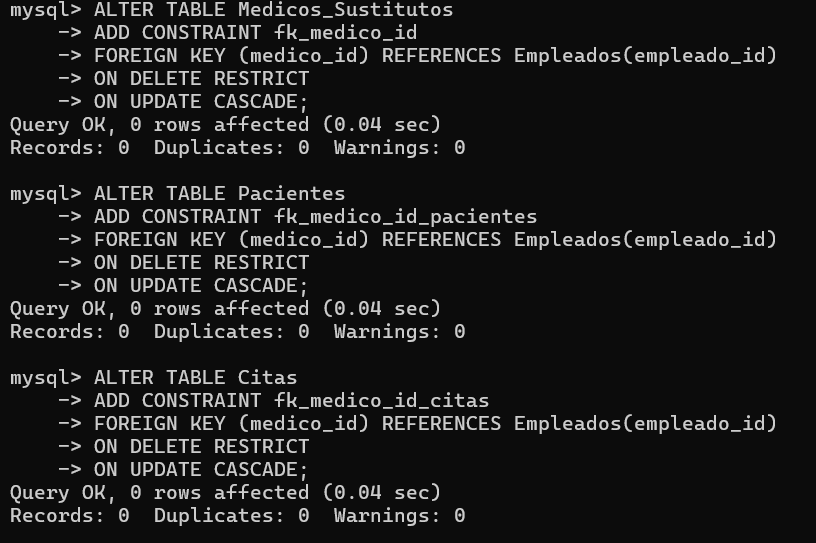
1. **Agregar** restricciones de **llave foránea** en cada una de las tablas a las que haya lugar según el **Modelo Relacional**, cada llave foránea debe aplicar restricción de integridad referencial Restrictiva para operaciones de Eliminar y Cascada para Operaciones de Actualizar.

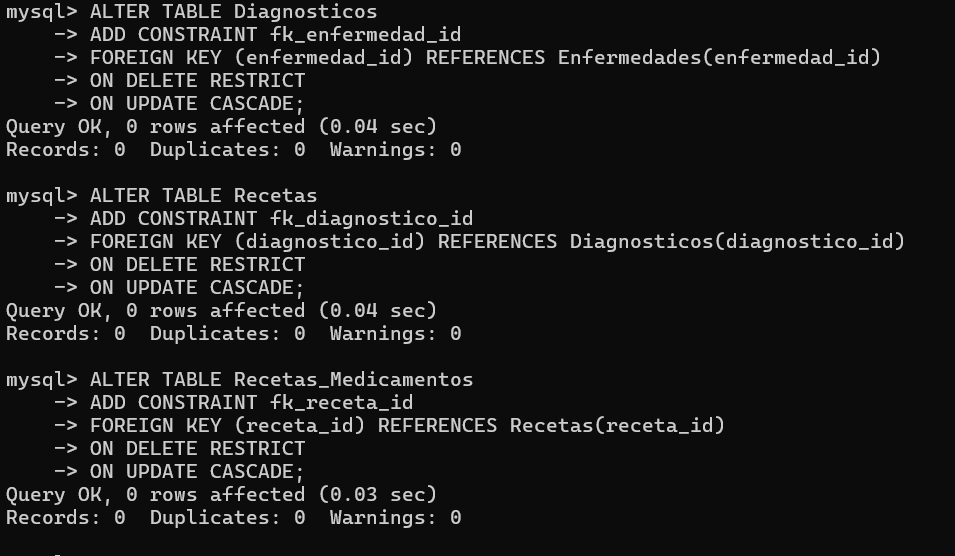
**(Estudiar e investigar qué es integridad referencial, como se aplica esto en las llaves foráneas)**

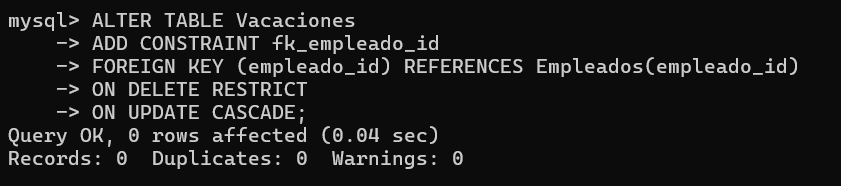
INSTRUCCIONES SQL PARA AGREGAR LAS LLAVES FORÁNEAS:

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

RECORTES DE PANTALLAS





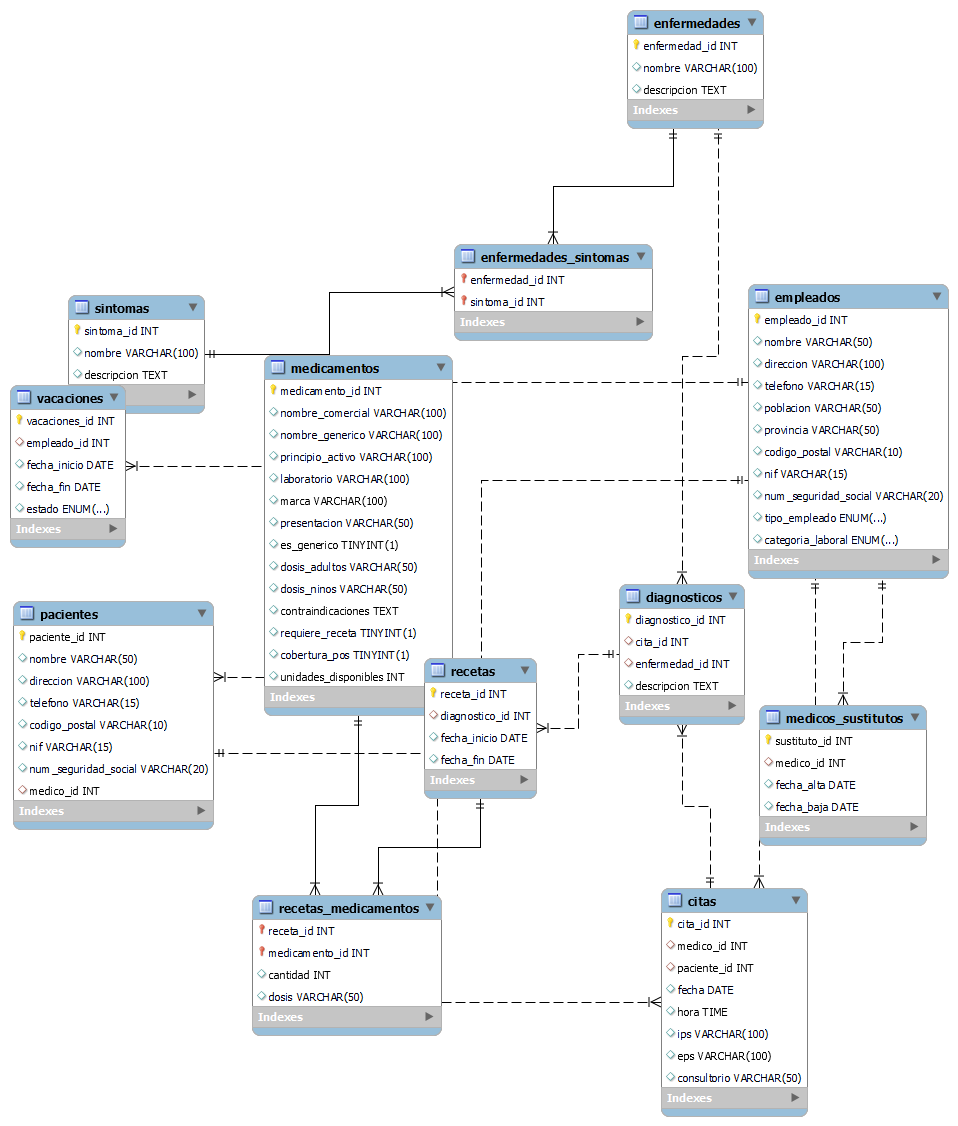


EXPLICACIÓN

Estos comandos me sirven para agregar restricciones a las llaves foraneas

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.

Abre MySQL Workbench y crea una nueva conexión a tu base de datos.

En el panel de navegación, ve a "Database" > "Reverse Engineer".

Sigue el asistente para seleccionar la base de datos que deseas visualizar.

Después se podrás ver el Modelo Relacional representado gráficamente.

1. Crear un **Backup** solo **de la estructura** de la **bd\_tu\_nombre\_tu\_apellido\_xyz** y guardar el resultado en un archivo llamado **backup\_bd\_tu\_nombre\_tu\_apellido\_xyz.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)

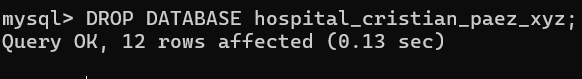
Yo lo hice con la interfaz gráfica de **MySQL Workbench,** ya estando en la interfaz grafica viendo las tablas pueden darle en File, seguido de Export, le damos en Forward Engineer SQL CREATE Script y seguimos los pasos y listo lo guardamos donde nosotros queramos.

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

INSTRUCCIÓN SQL :

DROP DATABASE hospital\_cristian\_paez\_xyz;

RECORTES DE PANTALLAS



EXPLICACIÓN

Con ese comando eliminamos la base de datos

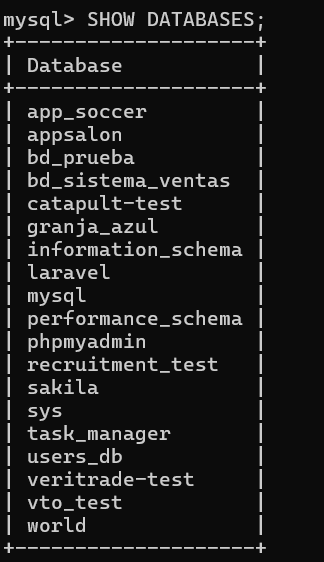
1. Mostrar las bases de datos del usuario **bd\_tu\_nombre\_tu\_apellido\_xyz**

**Ya el usuario fue eliminado anteriormente, según las instrucciones también ese usuario se llama user\_tu\_nombre\_y\_paellido y no bd\_tu\_nombre\_tu\_apellido\_xyz, entonces no lo tiene, solamente muestro las bases de datos.**

INSTRUCCIÓN SQL :

SHOW DATABASES;

RECORTES DE PANTALLAS



EXPLICACIÓN

Se muestran las bases de datos.

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

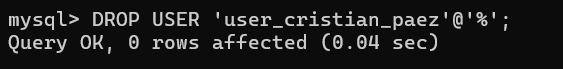
**Ya el usuario fue eliminado anteriormente, según las instrucciones también ese usuario se llama user\_tu\_nombre\_y\_paellido y no bd\_tu\_nombre\_tu\_apellido\_xyz, entonces no lo tiene, solamente muestro las bases de datos.**

**Comparto como se elimino anteriormente:**

INSTRUCCIÓN SQL :

DROP USER 'user\_cristian\_paez'@'%';

RECORTES DE PANTALLAS



EXPLICACIÓN

Con ese comando elimino el usuario.

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

INSTRUCCIÓN SQL:

mysql -u root -p hospital\_cristian\_paez\_xyz < C:\Users\crist\OneDrive\Documentos\HeidiSQL\DbHospital.sql

RECORTES DE PANTALLAS



EXPLICACIÓN

Con este comando restauro la base de datos.

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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| OPERACIÓN SQL | MySQL | Oracle | PostgreSQL | SQLServer |
| Entrar a la terminal | mysql -u usuario -p |  | psql -U usuario -d base\_de\_datos |  |
| Crear un usuario | CREATE USER 'user\_tu\_nombre\_tu\_apellido'@'%' IDENTIFIED BY 'AbcdeUdeC'; |  | CREATE USER user\_tu\_nombre\_tu\_apellido WITH PASSWORD 'AbcdeUdeC'; |  |
| Asignar permisos a un usuario | GRANT ALL PRIVILEGES ON \*.\* TO 'user\_tu\_nombre\_tu\_apellido'@'%'; |  | GRANT ALL PRIVILEGES ON DATABASE hospital\_cristian\_paez\_xyz TO user\_tu\_nombre\_tu\_apellido; |  |
| Cambiar la clave a un usuario | ALTER USER 'user\_tu\_nombre\_tu\_apellido'@'%' IDENTIFIED BY 'nueva\_clave'; |  | ALTER USER user\_tu\_nombre\_tu\_apellido WITH PASSWORD 'nueva\_clave'; |  |
| Quitar los permisos a un usuario | REVOKE ALL PRIVILEGES ON \*.\* FROM 'user\_tu\_nombre\_tu\_apellido'@'%'; |  | REVOKE ALL PRIVILEGES ON DATABASE hospital\_cristian\_paez\_xyz FROM user\_tu\_nombre\_tu\_apellido; |  |
| Eliminar un usuario | DROP USER 'user\_tu\_nombre\_tu\_apellido'@'%'; |  | DROP USER user\_tu\_nombre\_tu\_apellido; |  |
| Mostrar los usuarios activos en el Motor de BD | SELECT User FROM mysql.user; |  | \du |  |
| Salid de la terminal | exit; o Ctrl + D |  | \q |  |
| Mostrar las BD | SHOW DATABASES; |  | \l |  |
| Entrar o usar una BD | USE nombre\_base\_datos; |  | \c nombre\_base\_datos |  |
| Eliminar una BD | DROP DATABASE nombre\_base\_datos; |  | DROP DATABASE nombre\_base\_datos; |  |
| Crear la tabla del punto # 12 | CREATE TABLE nombre\_tabla (campo1 tipo, campo2 tipo, ...); |  | CREATE TABLE nombre\_tabla (campo1 tipo, campo2 tipo, ...); |  |
| Mostrar la estructura de una tabla | DESCRIBE nombre\_tabla; |  | \d nombre\_tabla |  |
| Cambiar el nombre de una tabla | RENAME TABLE nombre\_tabla\_a TO nombre\_tabla\_b; |  | ALTER TABLE nombre\_tabla RENAME TO nombre\_tabla\_b; |  |
| Agregar una columna a una tabla | ALTER TABLE nombre\_tabla ADD columna tipo; |  | ALTER TABLE nombre\_tabla ADD COLUMN columna tipo; |  |
| Cambiar el tipo de datos a una columna | ALTER TABLE nombre\_tabla MODIFY COLUMN columna nuevo\_tipo; |  | ALTER TABLE nombre\_tabla ALTER COLUMN columna TYPE nuevo\_tipo; |  |
| Cambiar el nombre de una columna | ALTER TABLE nombre\_tabla CHANGE columna viejo\_nombre nuevo\_nombre; |  | ALTER TABLE nombre\_tabla RENAME COLUMN viejo\_nombre TO nuevo\_nombre; |  |
| Eliminar una columna | ALTER TABLE nombre\_tabla DROP COLUMN columna; |  | ALTER TABLE nombre\_tabla DROP COLUMN columna; |  |
| Tipos de datos Numéricos Enteros | INT, TINYINT, SMALLINT, MEDIUMINT, BIGINT |  | INTEGER, SMALLINT, BIGINT, SERIAL |  |
| Tipos de datos numéricos reales o decimales | FLOAT, DOUBLE, DECIMAL |  | REAL, DOUBLE PRECISION, NUMERIC |  |
| Tipos de datos caracteres de longitud fija | CHAR(n) |  | CHAR(n) |  |
| Tipos de datos caracteres de longitud variable | VARCHAR(n) |  | VARCHAR(n) |  |
| Tipos de datos Boolean | BOOLEAN o TINYINT(1) |  | BOOLEAN |  |
| Tipos de datos Fecha | DATE, DATETIME, TIMESTAMP |  | DATE, TIMESTAMP, TIMESTAMPTZ |  |
| Tipos de datos para guardar datos binarios grandes | BLOB, MEDIUMBLOB, LONGBLOB |  | BYTEA |  |
| Tipos de datos conjunto | SET |  | N/A |  |
| Tipos de datos enumeración | ENUM |  | N/A |  |
| Columnas con valor autoincremental, secuencial o serial | AUTO\_INCREMENT |  | SERIAL |  |
| Cómo Eliminar una tabla | DROP TABLE nombre\_tabla; |  | DROP TABLE nombre\_tabla; |  |
| Agregar una Llave foránea al crear la tabla | CREATE TABLE nombre\_tabla (columna1 tipo, columna2 tipo, FOREIGN KEY (columna1) REFERENCES otra\_tabla(columna2)); |  | CREATE TABLE nombre\_tabla (columna1 tipo, columna2 tipo, CONSTRAINT fk\_nombre FOREIGN KEY (columna1) REFERENCES otra\_tabla(columna2)); |  |
| Agregar una llave foránea después de creada la tabla | ALTER TABLE nombre\_tabla ADD CONSTRAINT fk\_nombre FOREIGN KEY (columna1) REFERENCES otra\_tabla(columna2); |  | ALTER TABLE nombre\_tabla ADD CONSTRAINT fk\_nombre FOREIGN KEY (columna1) REFERENCES otra\_tabla(columna2); |  |
| Crear un índice | CREATE INDEX nombre\_indice ON nombre\_tabla(columna); |  | CREATE INDEX nombre\_indice ON nombre\_tabla(columna); |  |
| Mostrar los índice creados | SHOW INDEX FROM nombre\_tabla; |  | \di |  |
| Mostrar las restricciones de llave foránea de una tabla | SHOW CREATE TABLE nombre\_tabla; |  | \d nombre\_tabla |  |
| Eliminar una restricción de llave foránea o cualquier otra restricción | ALTER TABLE nombre\_tabla DROP FOREIGN KEY nombre\_restriccion; |  | ALTER TABLE nombre\_tabla DROP CONSTRAINT nombre\_restriccion; |  |
| Hacer un backup | mysqldump -u usuario -p nombre\_base\_datos > backup.sql |  | pg\_dump nombre\_base\_datos > backup.sql |  |
| Restaurar la BD | mysql -u usuario -p nombre\_base\_datos < backup.sql |  | psql -U usuario -d nombre\_base\_datos < backup.sql |  |