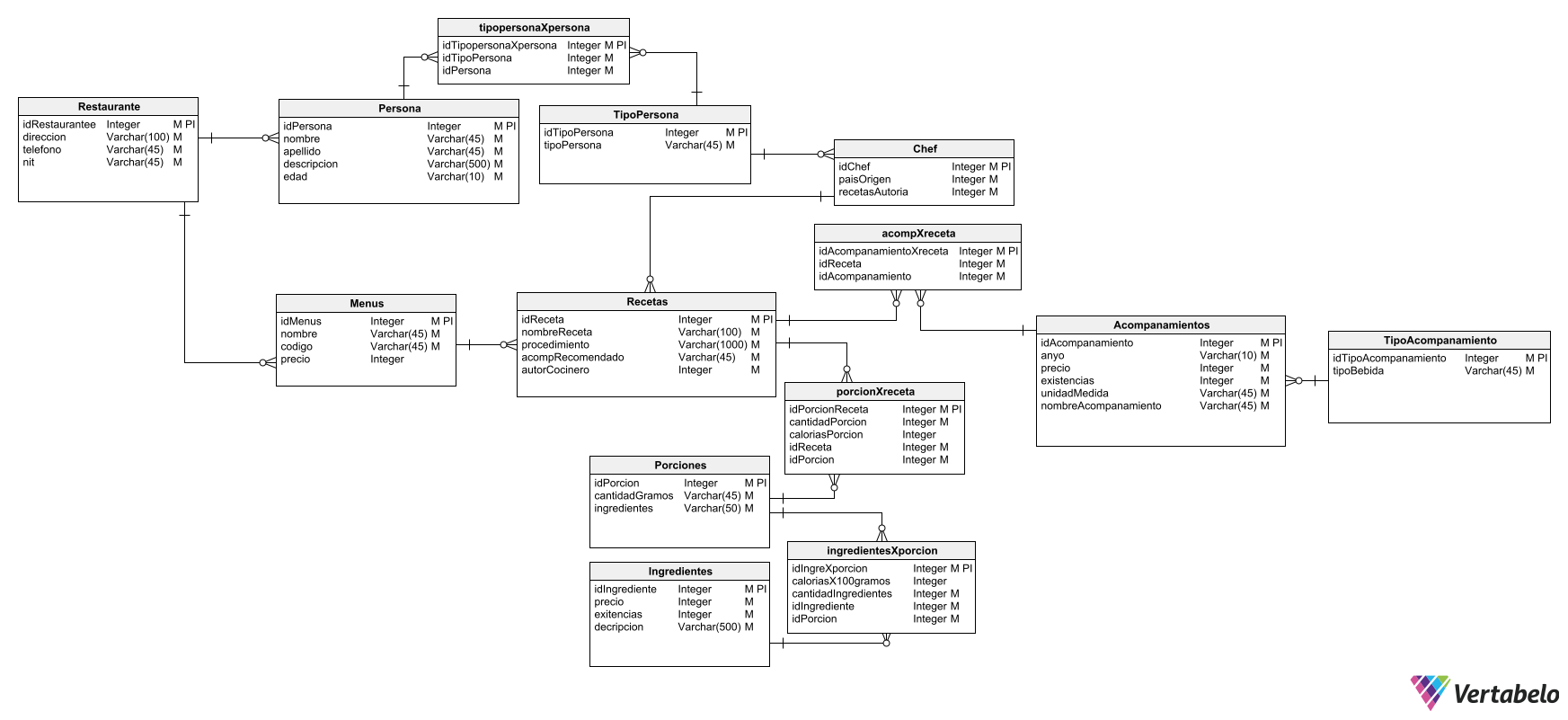
TALLER BASES DE DATOS:

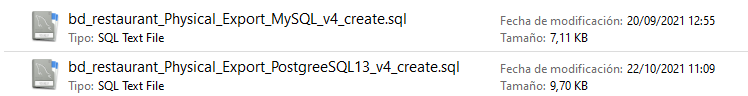
Casos de aplicación de la sentencia DDL\_ALTER en MySQL y PostgreeSQL.

Ing. Luis Felipe Narváez Gómez. E-mail: [luis.narvaez@usantoto.edu.co](mailto:luis.narvaez@usantoto.edu.co). Cod: 2312660. Facultad de Ingeniería de Sistemas.

La base de datos que tenemos de restaurantes corresponde a “db\_restaurant”, su estructura en el diagrama entidad relación es el siguiente:



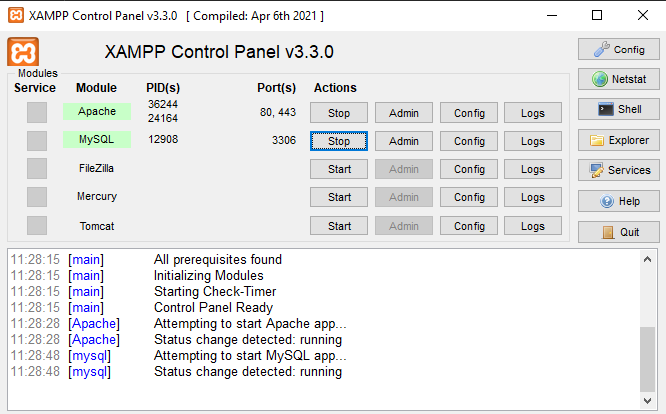
Esta base de datos fue generada en Vertabelo, este recurso web tiene la capacidad y función de generar el modelo y físico y a partir del generar código SQL respectivo para crear la base de datos en diferentes motores como IBM DB2, PostgreSQL, Oracle Database, Microsoft SQL Server, MySQL, HSQLDB, SQLite, Amazon Redshift, BigQuery y Snowflake. Utilizando esta herramienta podemos tener para este ejercicio la Database para trabajarla con MySQL Xampp y PostgreeSQL PGAdmin 4.



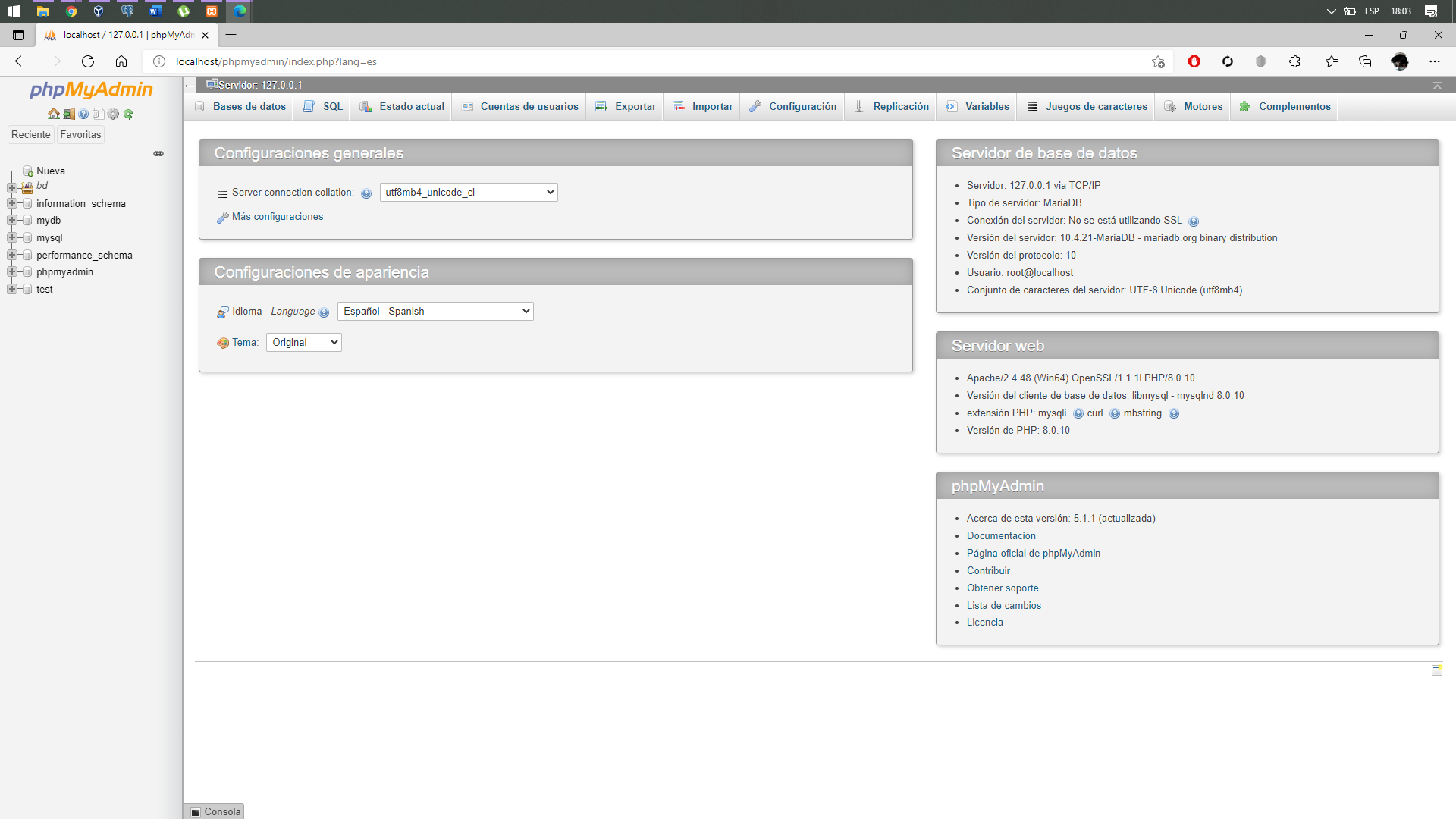
De esta manera este Ejercicio se dividirá en dos fases, la solución dentro de MySQL y el desarrollo dentro de PostgreSQL.

**MySQL XAMPP v3.3.0**

Haciendo uso de la GUI de Xampp podemos montar esta base de datos, para esto debemos lazar primeramente el servicio dentro de nuestra computadora.



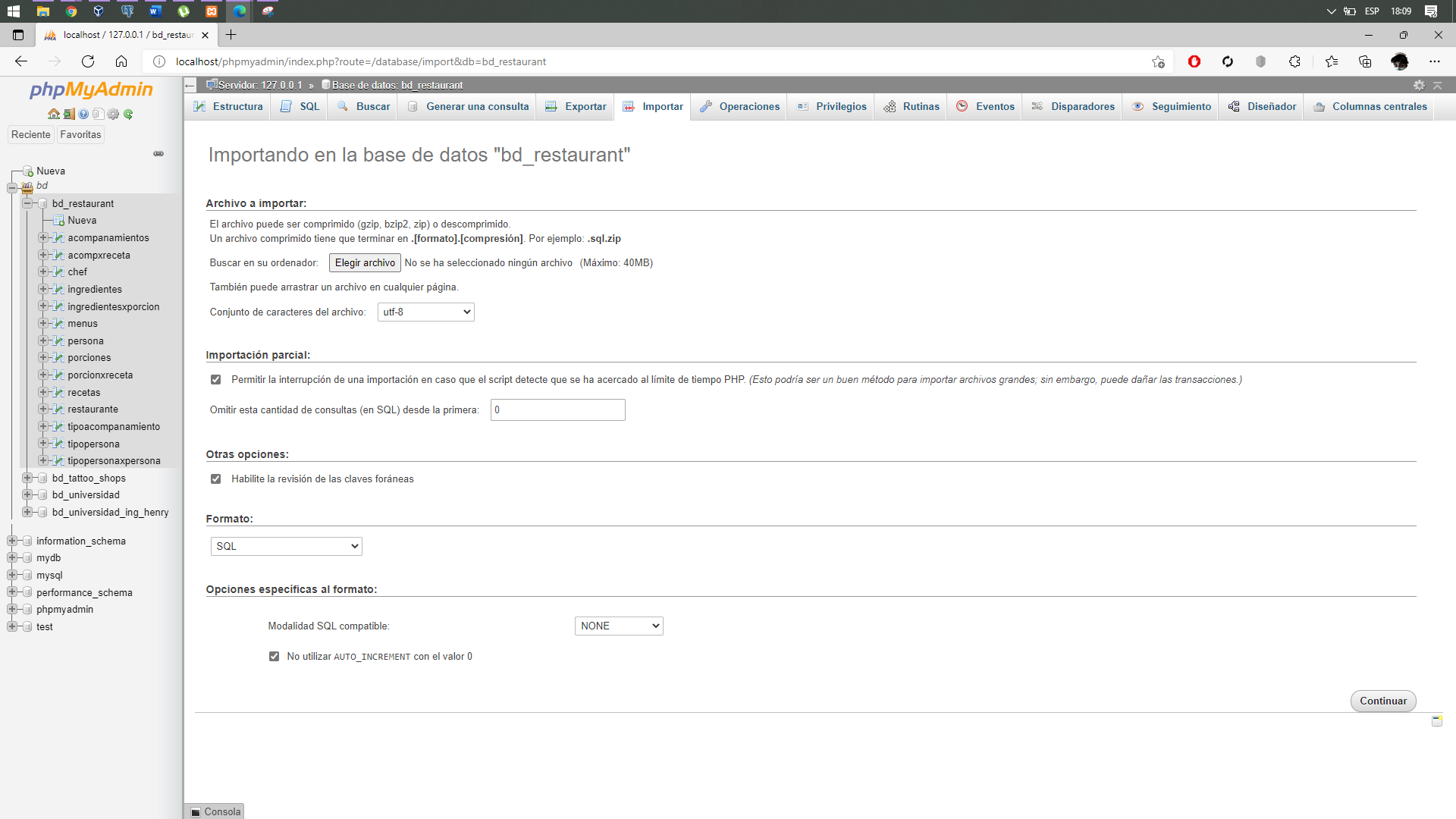
Luego por medio de algún navegador, podemos entrar a la dirección local de este servicio siendo “[localhost / 127.0.0.1 | phpMyAdmin 5.1.1](http://localhost/phpmyadmin/)” la cual nos mostrara lo siguiente:



Para crear la nueva base de datos, basta con ir al menú lateral de la pantalla, seleccionar la herramienta vertical que dice “nueva” y escribir el nombre de la base de datos y un cotejamiento de “utf8mb4\_general\_ci” el cual nos permitirá ingresar datos que contengan caracteres especiales de algunos idiomas, como el español, portugués, francés, etc. Lenguas románticas o del árbol de lenguas nórdicas, las cuales utilizan tildes, la “ñ”, aspectos como “¨”, etc.



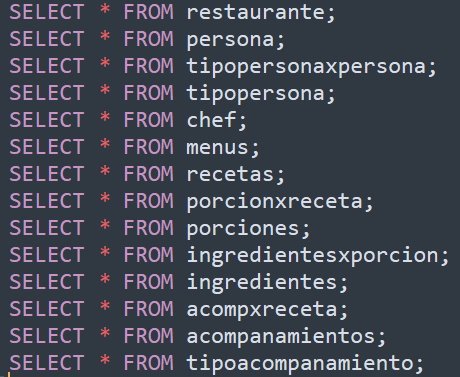
Una vez creada, podemos seleccionarla en el mismo menú lateral en vertical de la pantalla y dentro de ella, seleccionamos en la barra de menús superior la opción de “importar”. En ella solo debemos elegir nuestro archivo de extensión “SQL” y que el formato este seleccionado precisamente en este tipo de extensión.



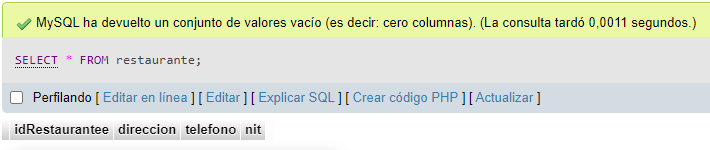
Luego de esto, solo queda ejecutar el código y se montaran las tablas y restricciones propias de la Base de Datos. Podemos comprobar cada una de las existencias de estas tablas y sus atributos o columnas mediante consultas SQL, la siguiente imagen corresponde a las tablas existentes dentro de la base de datos en el Motor MySQL de Xampp:

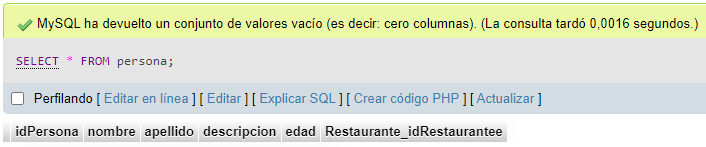


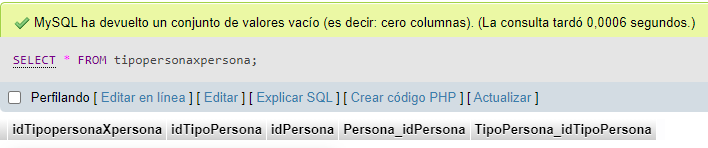
La siguiente Imagen Corresponde a las consultas realizadas para consultar los elementos dentro de las tablas. Ya que estas no están pobladas, solo se mostrarán los nombres de las columnas o los nombres de los atributos en cada tabla:

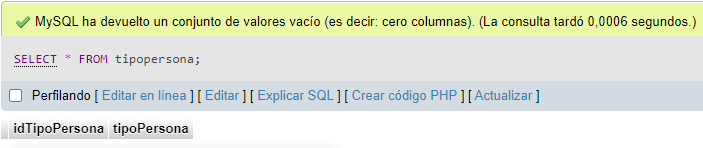


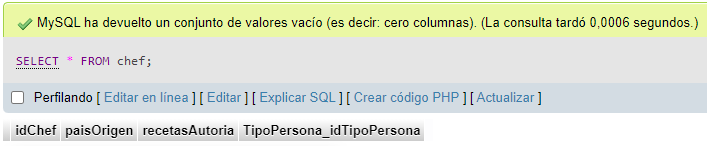
Estos son los resultados de cada una de las consultas:

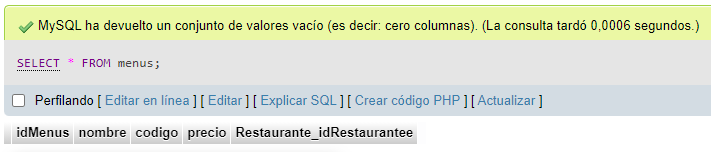


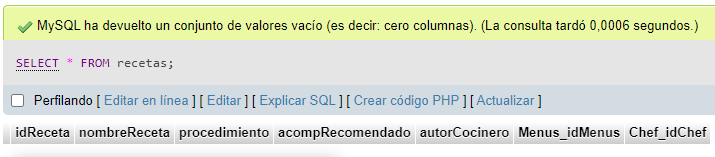


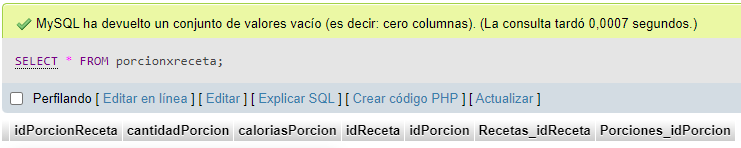


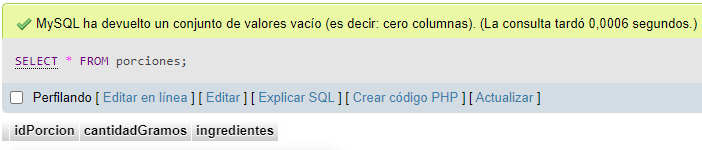


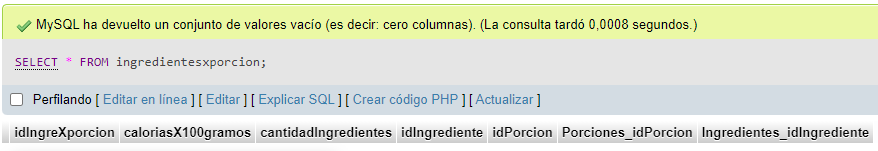


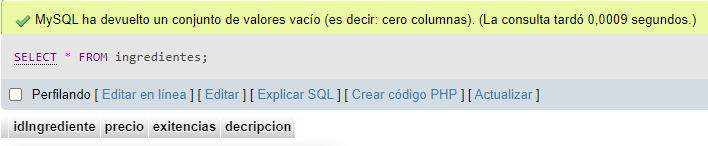


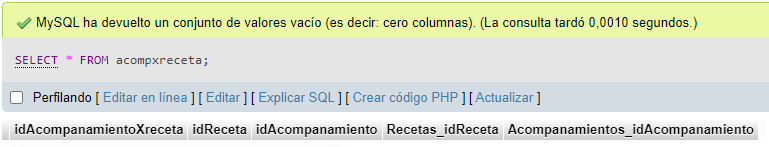


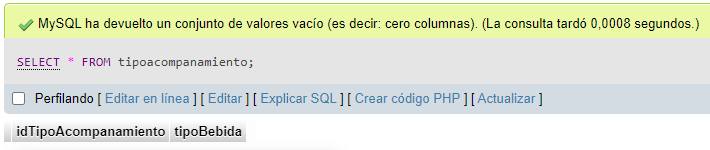
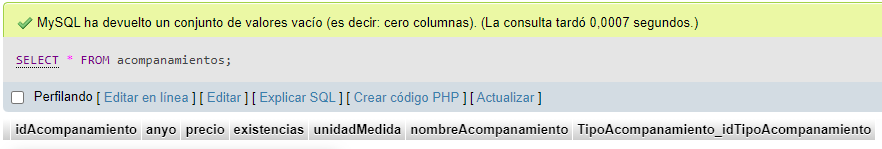




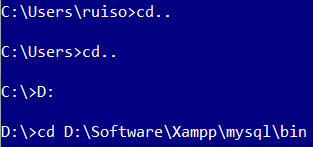




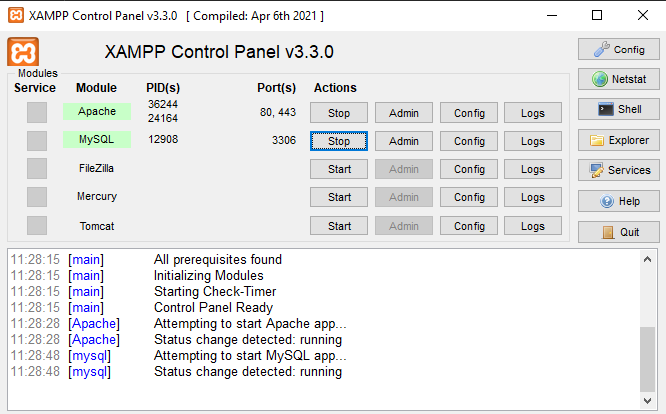




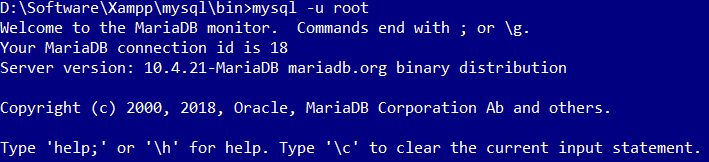
Ahora bien, para movernos con mayor libertad por las tablas y la misma base de datos podemos transportarnos a un entorno de trabajo distinto a Xampp como lo es la Consola de Comandos. Para esto debemos primero acceder a la ruta donde tenemos instalado Xampp y abrir el directorio de bin.



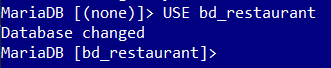
Una vez hecho esto, ejecutamos nuestro Xampp y activamos tanto Apache como MySQL.



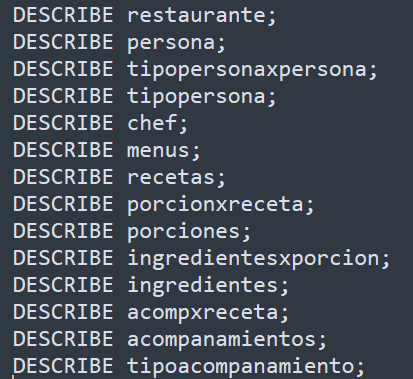
Una vez hecho esto, podemos iniciar el motor de MySQL para trabajar desde la consola de comandos.



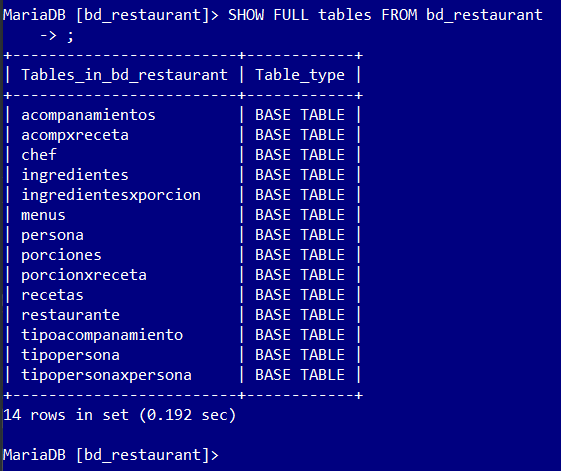
Ahora si podemos cambiar a la base de datos que estamos trabajando con el comando “USE nombre\_database”, de esta manera empezaremos a hacer tanto consultas como transformaciones a la base de datos.

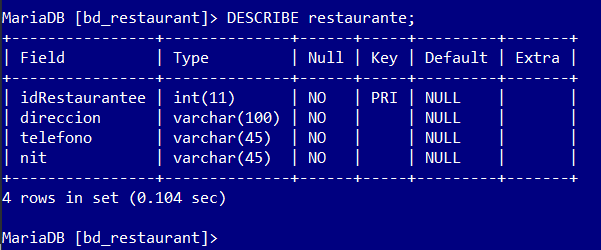


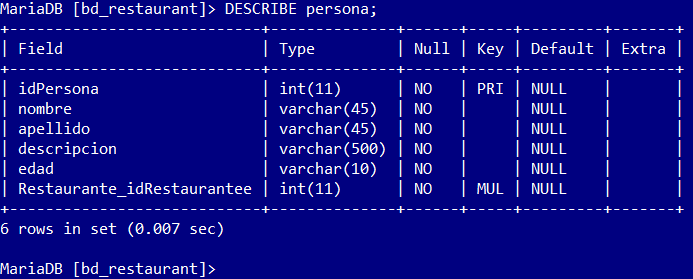
Ahora bien, podemos realizar las consultas de los datos de las tablas para cerciorarnos que estas tengan los atributos que estamos utilizando.

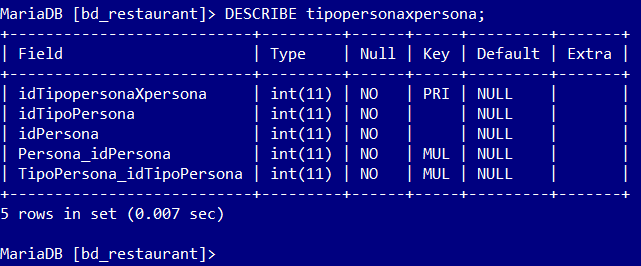


Su resultado sería el siguiente:

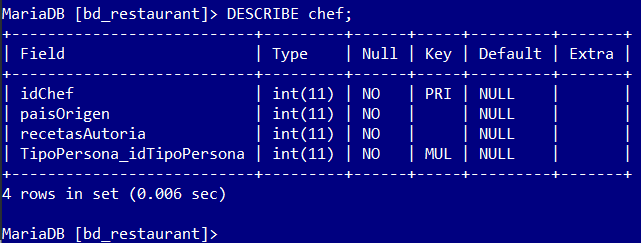




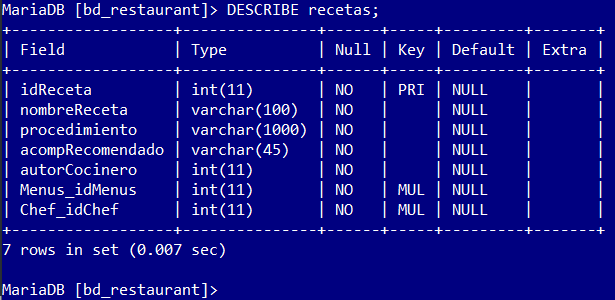


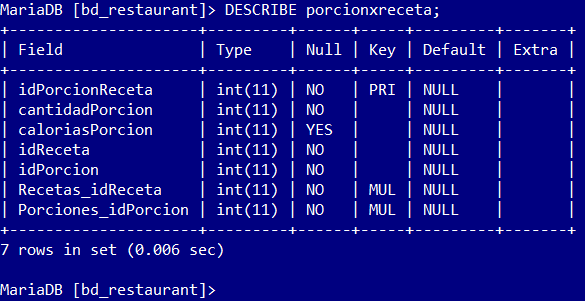


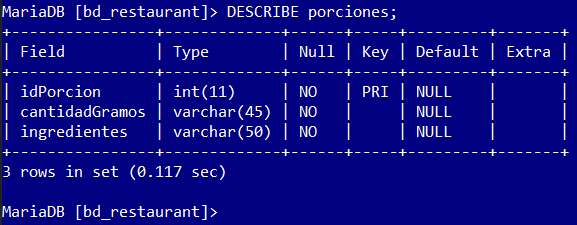


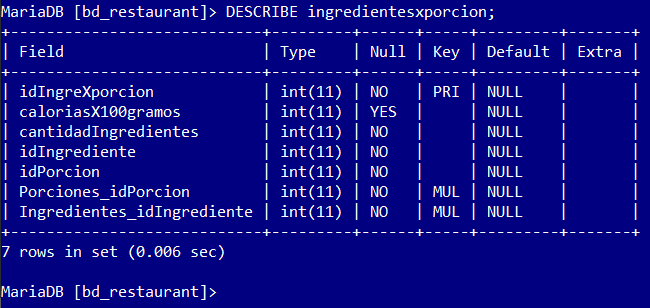


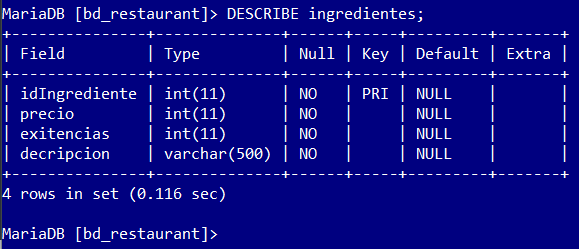


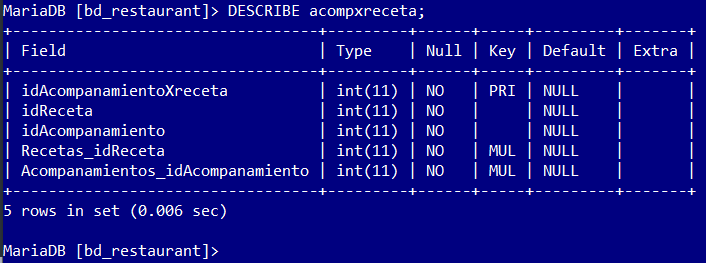


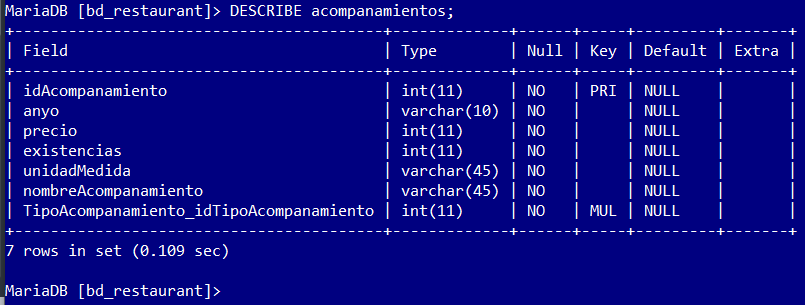


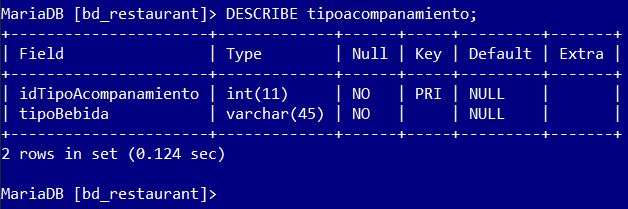




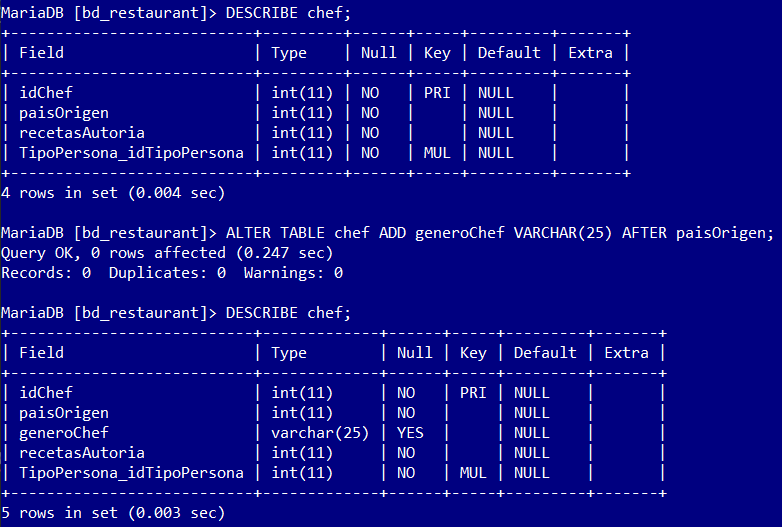








**AGREGAR CAMPO:** Ahora supongamos que, en nuestra base de datos, El Gerente del restaurante nos pide añadir un nuevo atributo a la tabla de Chef, el cual será Genero. Genero para facilitar la interacción podrá ser definido como carácter simple.

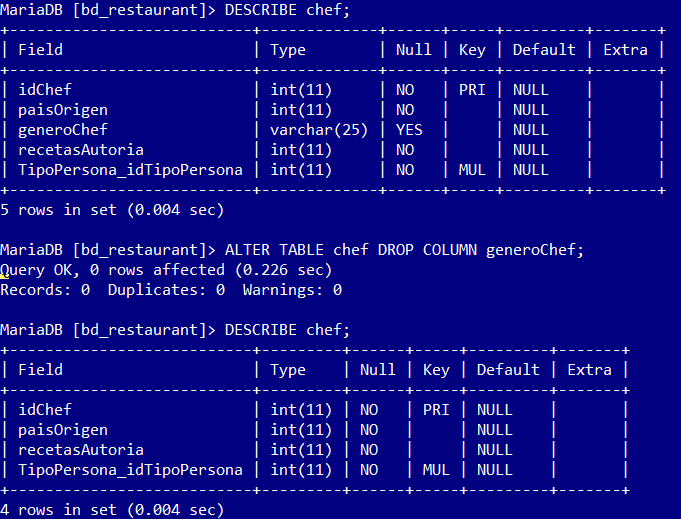


El comando utilizado fue el siguiente:

ALTER TABLE chef ADD generoChef VARCHAR(25) AFTER paisOrigen;

Lo que traduce este comando sería algo como: transfórmame la tabla “chef” añadiéndome una nueva columna o atributo llamado “generoChef” el cual será de tipo “varchar” que tendrá “25 caracteres” y este después del atributo “paísOrigen”. Al no especificarle nada más, hará que la misma por defecto sea un campo que si pueda ser nulo y no sea una llave primaria.

**QUITAR UN CAMPO:** Ahora supongamos que, tras una queja de los empleados, la decisión del Gerente de añadir a la base de datos el campo de Genero, fue mal recibida, en su argumento defienden que los comensales no les debe importar que genero tiene la persona que prepare un platillo especifico. Por tal motivo se nos pide borrar este campo nuevamente de la tabla “chef”.



El comando utilizado es:

ALTER TABLE chef DROP COLUMN generoChef;

Lo que traduce este comando es algo como: transfórmame la tabla “chef” eliminando la columna “generoChef”.

**RENOMBRAR UN CAMPO:** trabajando con la base de datos nos damos cuenta que la llave foránea “TipoPersona\_idTipoPersona” posee un nombre muy largo para trabajar con ella y no es fácilmente identificable, por tal motivo se resuelve renombrar el campo.

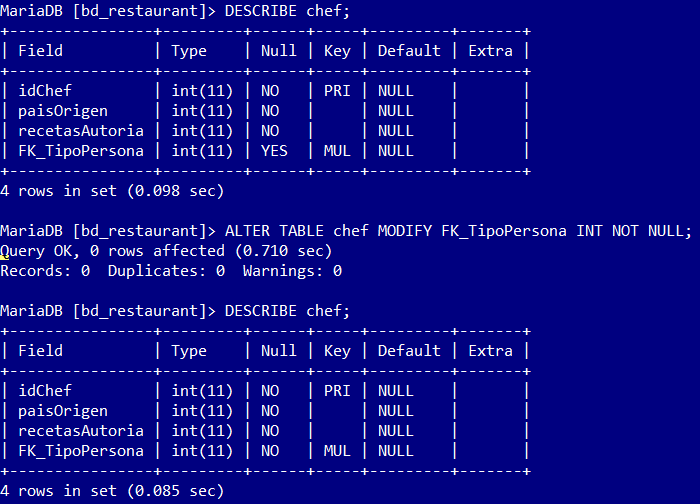


El comando utilizado es:

ALTER TABLE chef CHANGE TipoPersona\_idTipoPersona FK\_TipoPersona INT;

Este anterior comando traduce algo como: Transformame una tabla con nombre “chef” y cambia el atributo “TipoPersona\_idTipoPersona” por (marcado por un espacio en el comando) “FK\_TipoPersona” que será de tipo “INT”, este reemplazo se dará en la misma columna sin afectar los datos ya existentes en la misma, manteniendo la ubicación. Si no es dado otro aspecto más, se tomará por defecto el estado de “Key” previamente establecido y dejando que si puedan existir datos nulos.

**CAMBIAR EL TIPO DE DATO A UNO EXISTENTE:** con el ultimo cambio que hemos realizado, hemos dejado la cualidad a “FK\_TipoPersona” de contener datos nulos, este aspecto no es requerido en nuestra base de datos pues una persona registrada dentro del restaurante debe poderse identificar dentro de uno de los diferentes roles en la empresa, sea cual sea. Por tal motivo es necesario cambiar el tipo de dato de este atributo.

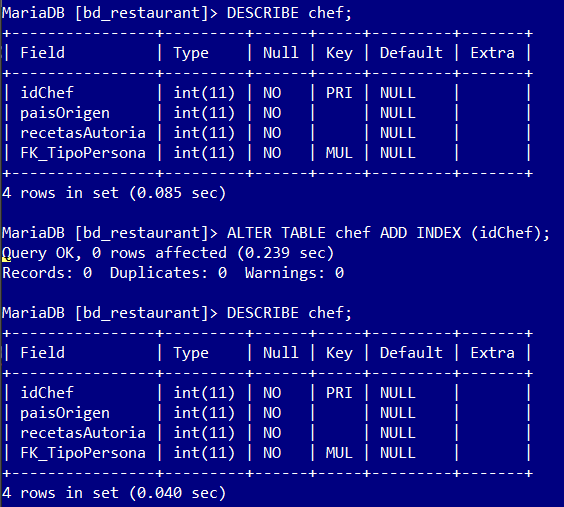


El comando utilizado fue:

ALTER TABLE chef MODIFY FK\_TipoPersona INT NOT NULL;

Este comando traducido dicta algo como: transfórmame la tabla denominada “chef” modificando el atributo “FK\_TipoPersona” haciendo que sea de tipo “INT” y no pueda albergar datos nulos.

**CREAR UN ÍNDICE:** El Gerente del restaurante solicita que podamos encontrar los datos de los cocineros de manera más eficiente, para la forma más sencilla es asignar un número de identificación único para cada persona, esto se logra manteniendo un índice para la tabla de chef.



El comando utilizado es:

ALTER TABLE chef ADD INDEX (idChef);

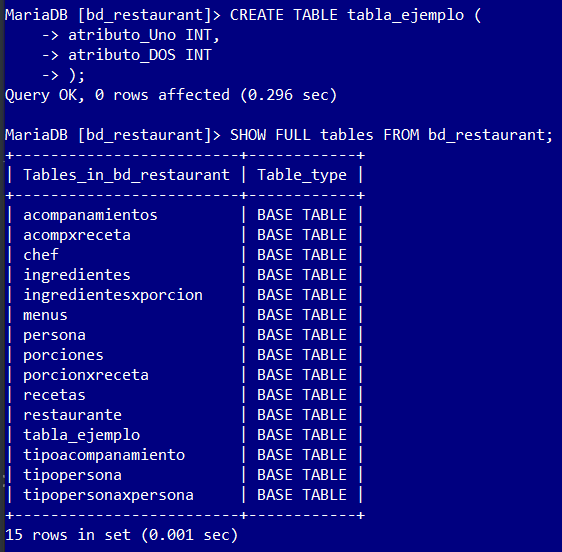
Este comando añade un índice que crea un código único ordenado por cada dato ingresado al momento de poblar la tabla, esto generalmente se hace con la columna que es la llave primaria de la tabla.

El comando traducido se da como: Transfórmame una tabla llamada “chef” y añádeme un índice en la columna “idChef”.

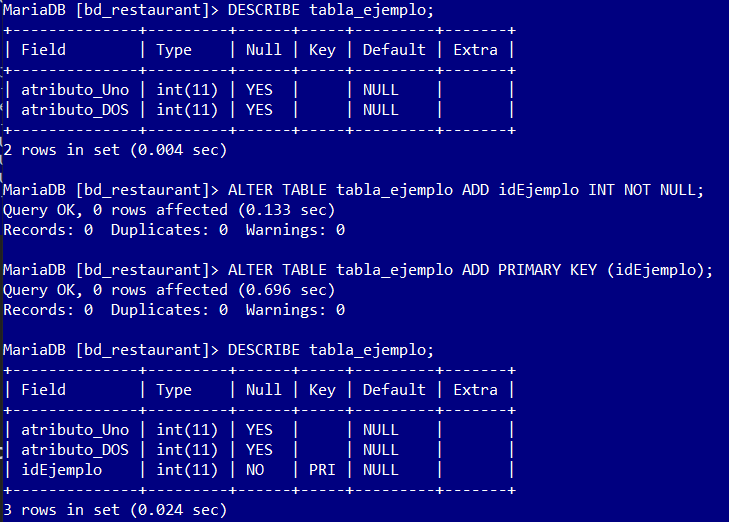
Ahora bien, crearemos una tabla flotante dentro de la base de datos para hacer ejercicios de llave primaria (crear y eliminar) y ejercicios de llave foránea (crear y eliminar). Las tablas actuales son las siguientes:



**CREAR NUEVA TABLA:** CREATE TABLE tabla\_ejemplo ( atributo\_Uno INT, atributo\_DOS INT );



**CREAR LLAVE PRIMARIA:** La tabla como podemos ver a continuación, solo posee dos atributos no una llave primaria, por tanto, debemos crear una.

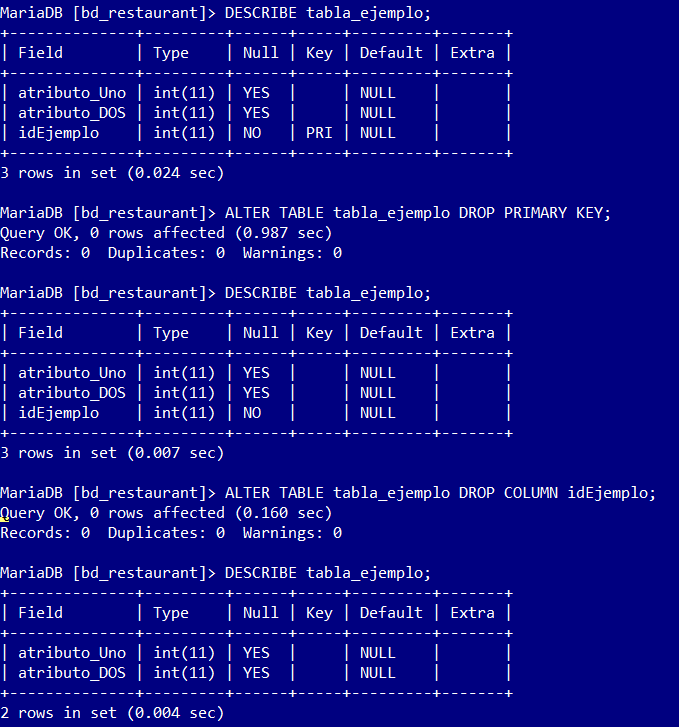


Aquí se utilizaron dos comandos, uno para crear el atributo y otro para que este sea una llave primaria o PRIMARY KEY.

ALTER TABLE tabla\_ejemplo ADD idEjemplo INT NOT NULL;

ALTER TABLE tabla\_ejemplo ADD PRIMARY KEY (idEjemplo);

**BORRAR LLAVE PRIMARIA:** como observamos, se cometió un “error”, realmente no es necesario que la llave primaria este en una determinada posición dentro de la tabla, sin embargo, por convenciones de trabajado con las bases de datos, como una regla no escrita, la llave primaria se deja como el primer atributo dentro de una tabla. Por tanto, podemos eliminar el campo previamente creado como la llave primaria y volverlo a crear cerciorándonos que tenga la posición inicial.



El primer comando utilizado quita la etiqueta que identifica un campo como llave primaria, el segundo simplemente elimina el atributo de la tabla.

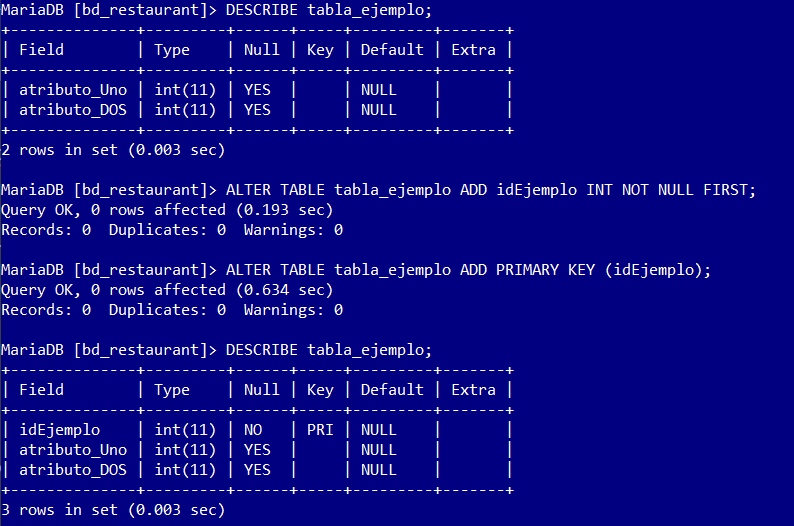
ALTER TABLE tabla\_ejemplo DROP PRIMARY KEY;

ALTER TABLE tabla\_ejemplo DROP COLUMN idEjemplo;

Para crear de nuevo el campo, pero ahora que este en primer lugar utilizaremos los siguientes comandos:

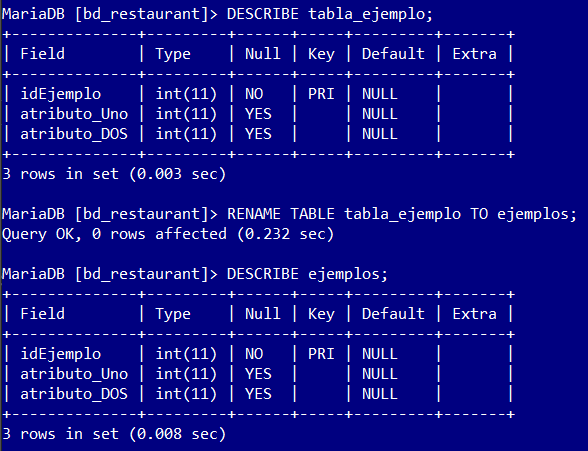
ALTER TABLE tabla\_ejemplo ADD idEjemplo INT NOT NULL FIRST;

ALTER TABLE tabla\_ejemplo ADD PRIMARY KEY (idEjemplo);

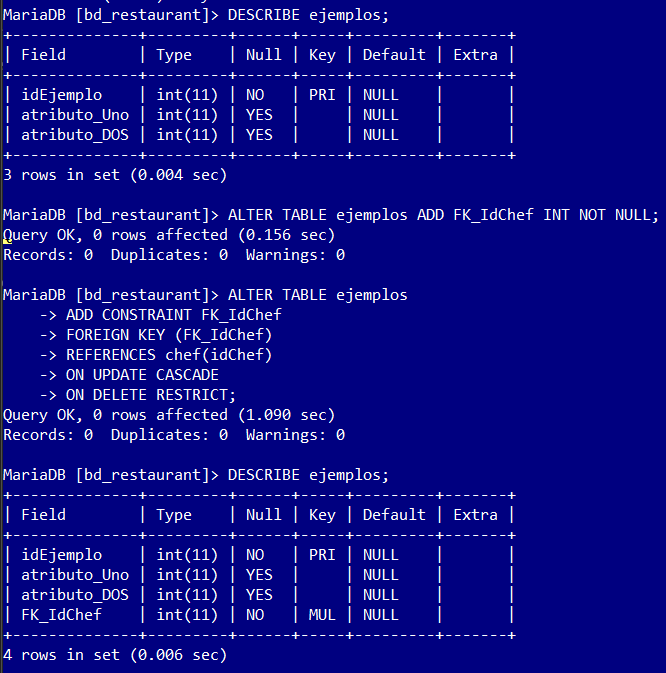


**RENOMBRAR TABLA:** Para facilitar un poco más el manejo de la tabla podemos renombrarla. Esto lo que hace es crear una nueva tabla con un “CREATE TABLE” con el nombre de la tabla que hemos dado, con los atributos de la tabla original de la que estamos solicitando y movemos todos los datos hacia la nueva tabla. La tabla original se elimina y queda la nueva reemplazando la anterior.

RENAME TABLE tabla\_ejemplo TO ejemplos;



**CREAR LLAVES FORANEAS:** las llaves foráneas nos permiten traer ciertos datos que están enlazados con la información que tenemos en una tabla que por su naturaleza propia pertenecen a otra tabla.



Los comandos utilizados son dos, el primero que crea el campo donde se destinara la llave foránea y el segundo corresponde a toda la configuración de la llave foránea.

ALTER TABLE ejemplos ADD FK\_IdChef INT NOT NULL;

El siguiente código se traduce como: Transfórmame una tabla llamada “ejemplos” en ella añádeme una restricción que se denominará “FK\_IdChef” que será de tipo llave foránea, la misma estará en la columna “FK\_IdChef” de la tabla “ejemplos” y tendrá como referencia los datos de la columna “idChef” de la tabla “Chef”. El comportamiento de actualización de datos se dará en cascada a partir de las hijas de la tabla “ejemplos” y en caso de ser eliminada su comportamiento será “restrictivo”.

ALTER TABLE ejemplos

ADD CONSTRAINT FK\_IdChef

FOREIGN KEY (FK\_IdChef)

REFERENCES chef(idChef)

ON UPDATE CASCADE

ON DELETE RESTRICT;

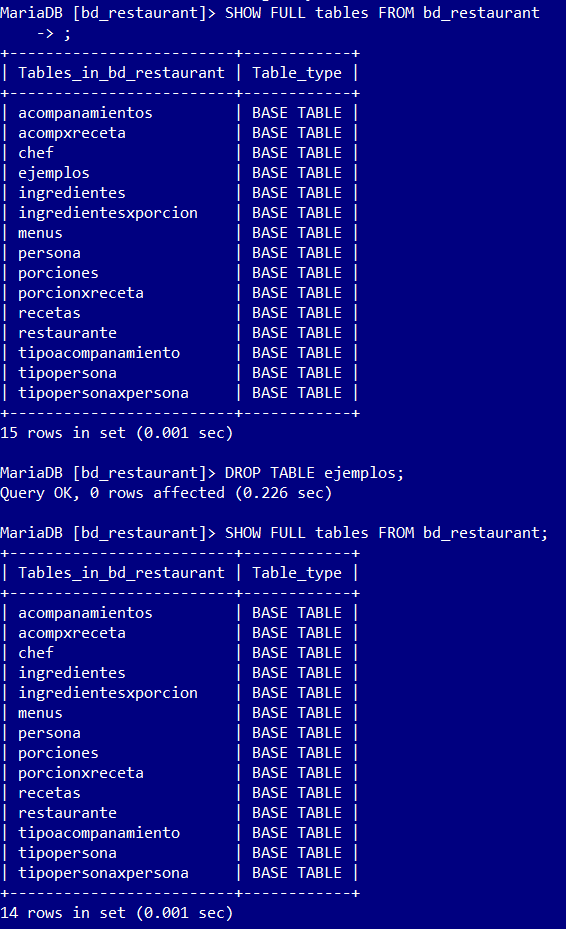
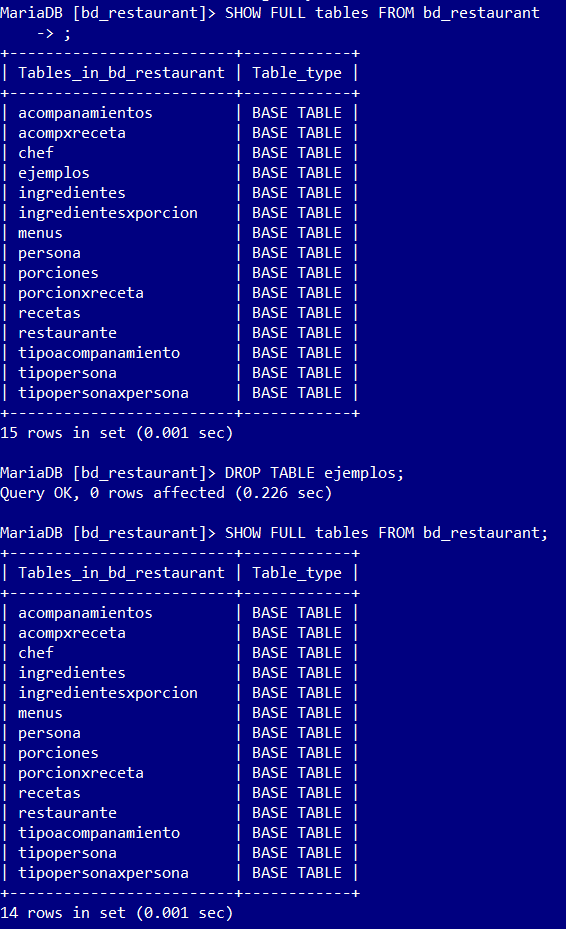
**ELIMINAR TABLA:** De esta manera ahora tenemos las siguientes tablas:



Volveremos a su estado original eliminando la tabla “ejemplos”.

DROP TABLE ejemplos;

SHOW FULL tables FROM bd\_restaurant;

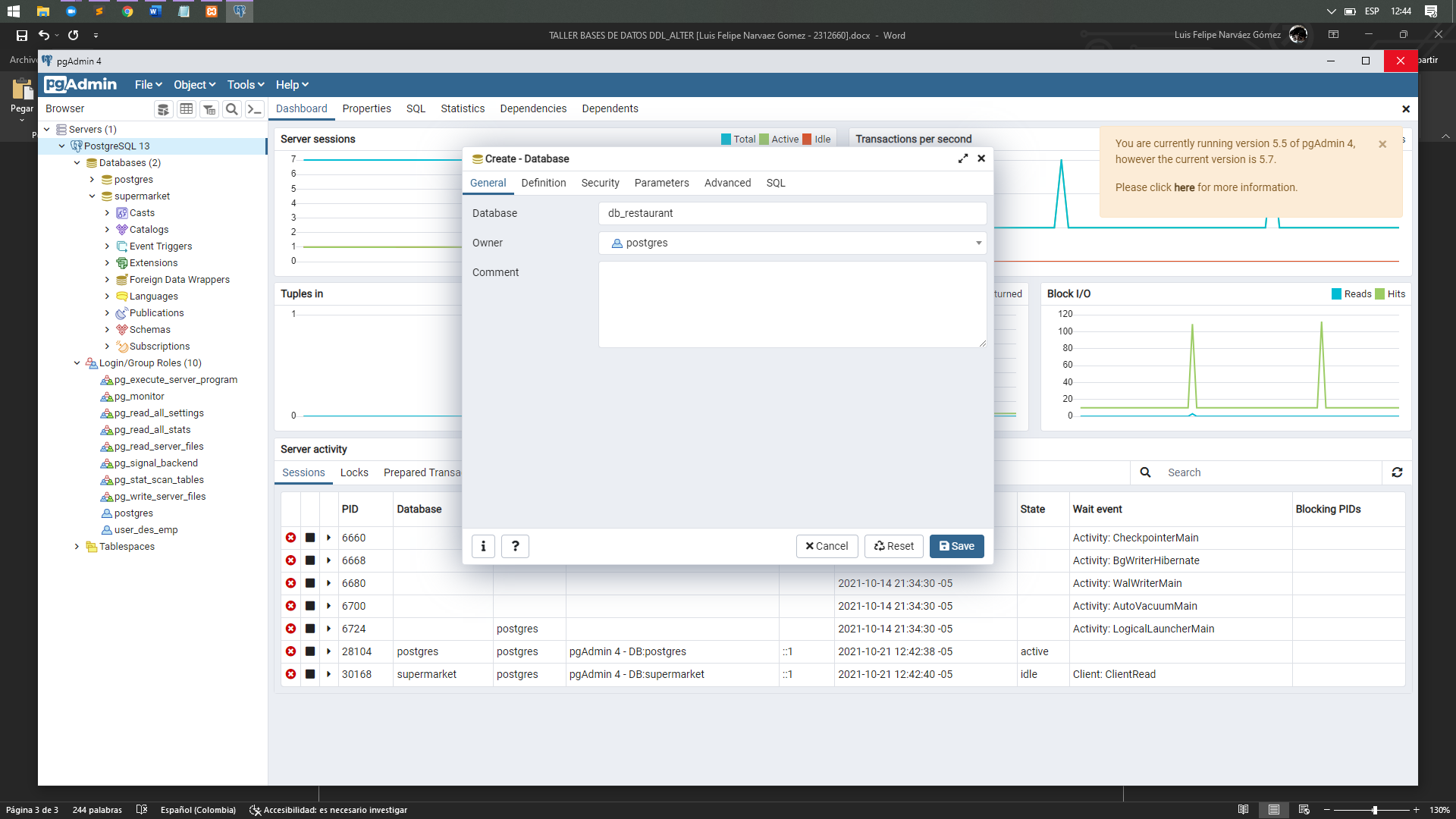


**PostgreSQL PgAdmin 4.**

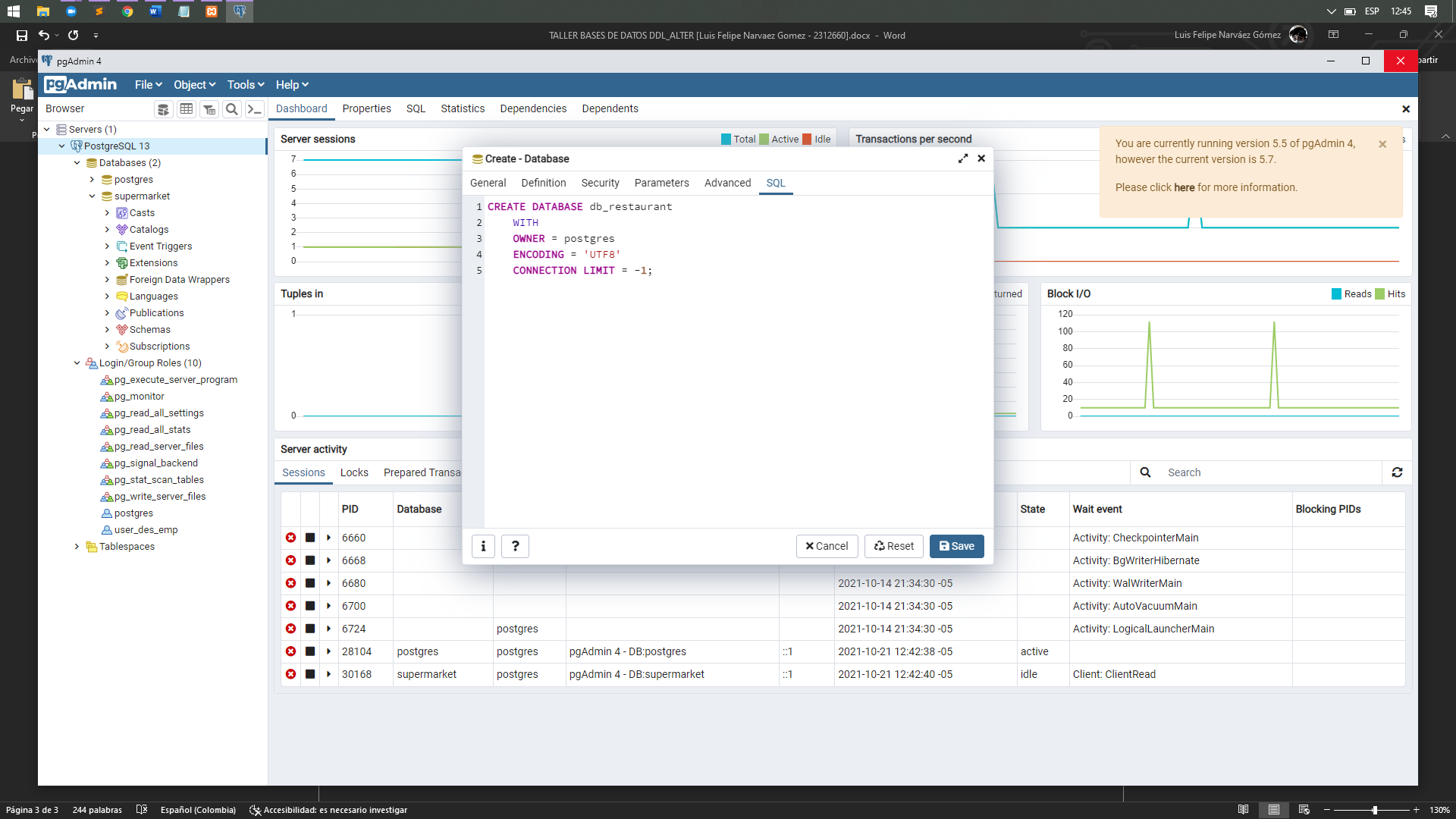
Ahora bien, si en cambio yo quisiera trabajar en PostgreSQL PGAdmin 4, tendría que crear la database. Lo primero que debemos hacer es dirigirnos al servidor que tenemos abierto, colocado en el menú lateral de la pantalla, seleccionamos “PostgreSQL 13”, dando clic derecho en el aparecerá un menú de tareas vertical donde podremos ubicar la opción “create” y en ella la opción de “Database”, la cual seleccionaremos al final.

Dado lo último se desplegará como aparece a continuación, una ventana donde podremos configurar esta base de datos. Para hacer mas sencillo este procedimiento, solo ubicaremos el nombre de la base de datos y el usuario que se encargara de ella, esto dentro de la pestaña “General”.

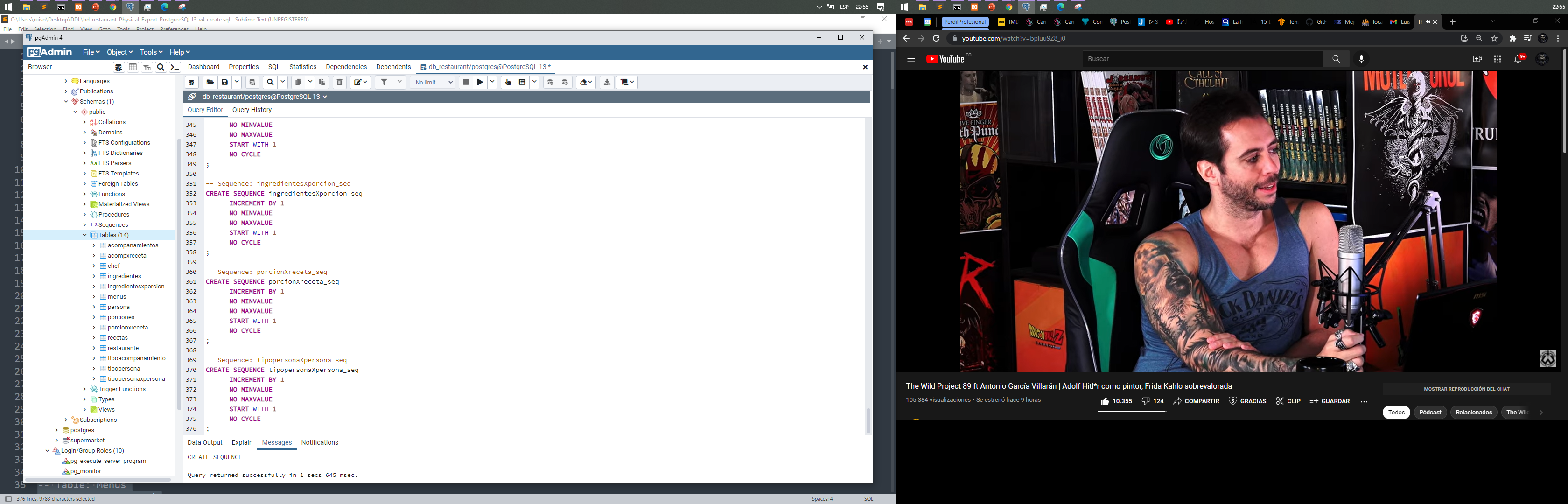
El nombre de la base de datos figura como un espacio continuo del atributo “Database” y el usuario que elegiremos para delegársela esta en el atributo de “Owner”. En este ultimo paso debemos tener cuidado en que no es recomendable que utilicemos el super usuario o el usuario administrador de PostgreSQL denominado “postgres”, con la finalidad de que configuraciones dadas específicamente para la base de datos en cuestión, no alteren el funcionamiento general de PostgreSQL.



También podemos visualizar que en la pestaña “SQL” se nos dará el código SQL para crear esta misma base de datos que estamos haciendo por el método de GUI y que es internamente lo que ejecutara PgAdmin 4.

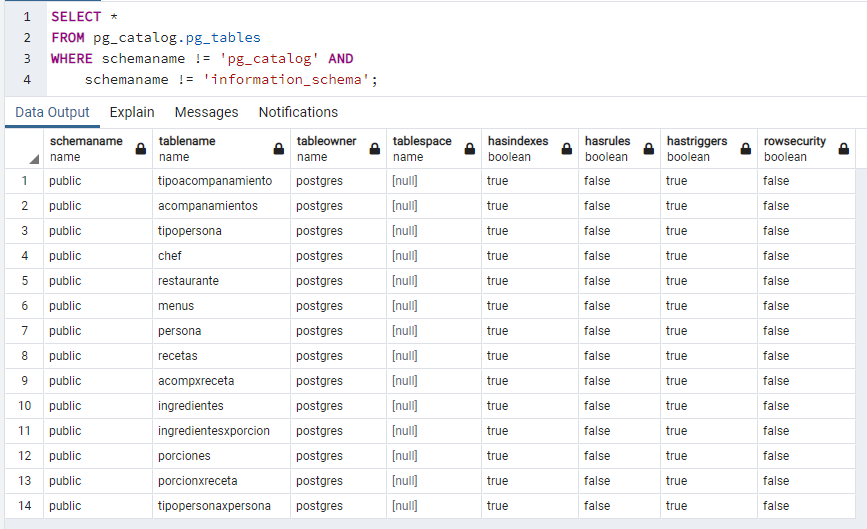


Una vez creada la base de datos debemos entrar a la barra de menús, el menú Tool y seleccionar la herramienta Query, en ella podremos traspasar las sentencias SQL que crearan nuestra base de datos. Las sentencias SQL las podemos hallar en el interior de nuestro archivo generado por Vertabelo, basta solo con abrirlo con cualquier editor de texto o programa que trabaje con texto plano tal como “Bloc de Notas”. Pasaremos todas las instrucciones al espacio Query y las ejecutaremos, esperando que en el espacio inmediatamente inferior no salga ningún problema.



Podremos observar también de la anterior imagen que, en la parte lateral en vertical, en la sección de “Schemas” y las “tables”, estarán todas las entidades que hemos creado anteriormente. Exactamente los mismos ejemplos que desarrollamos en MySQL los desarrollaremos en PostgreSQL. El espacio de comandos aquí será el mismo dado dentro de la GUI de PgAdmin, la opción de Query que antes vimos, dentro de ella podemos hacer todas las consultas y transformación DDL y DML que queramos para este ejercicio.

**MOSTRAR TODAS LAS TABLAS:** Aquí es donde algunos comandos serán distintos entre MySQL y PostgreSQL, aunque en la mayoría son pequeños arreglos de sintaxis lo que diferenciara el SQL entre ambos motores de base de datos, pero en general es casi lo mismo en ambos.

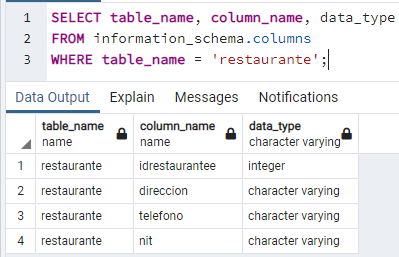


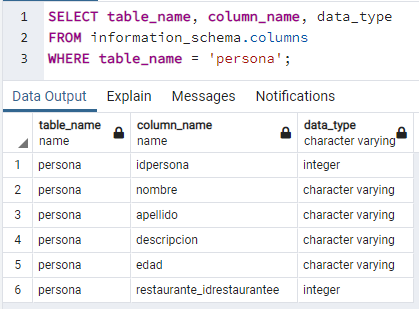
SELECT \*

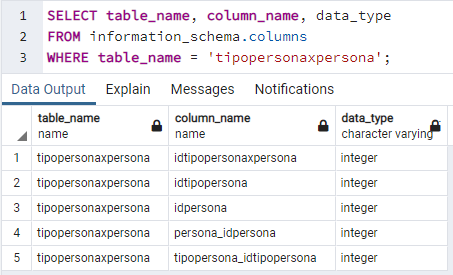
FROM pg\_catalog.pg\_tables

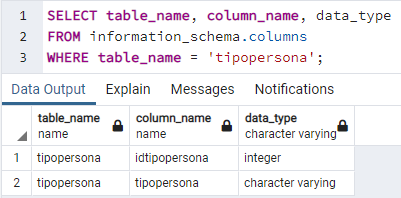
WHERE schemaname != 'pg\_catalog' AND schemaname != 'information\_schema';

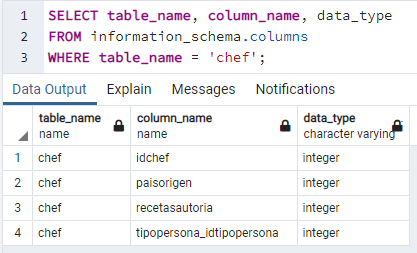
**MOSTRAR ATRIBUTOS DE LAS TABLAS:**

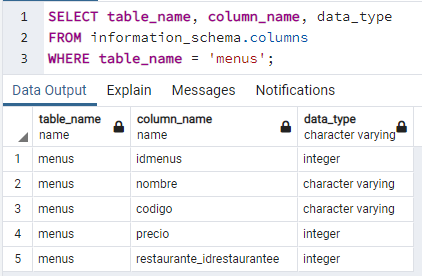


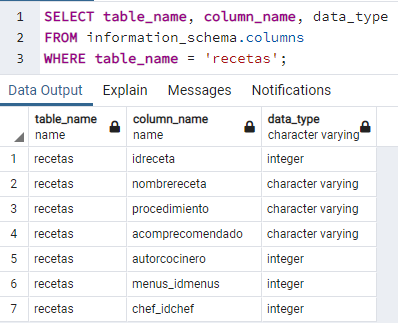


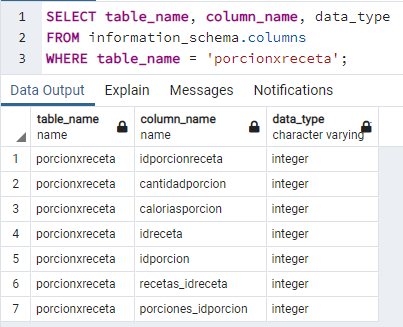


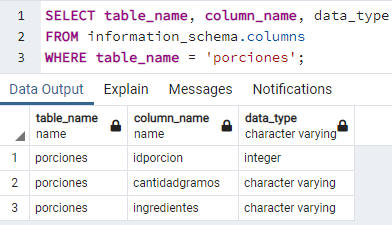


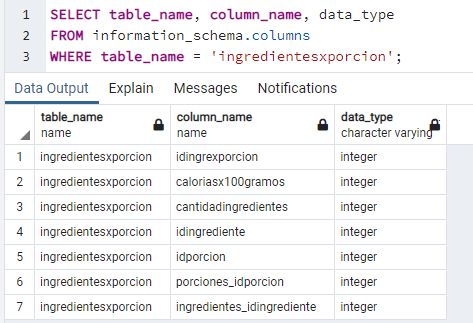




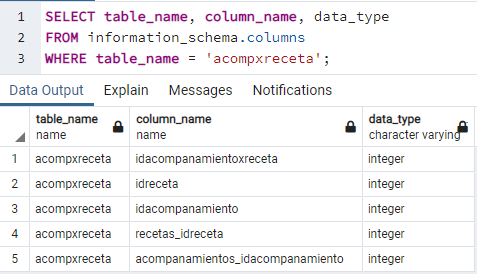


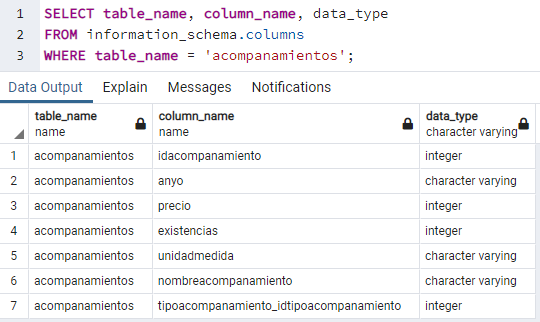


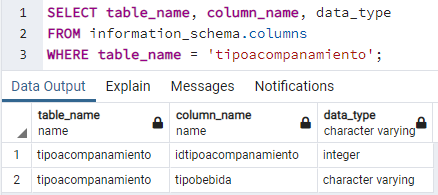




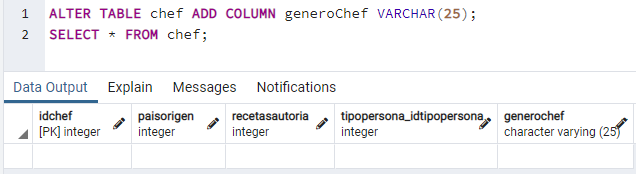








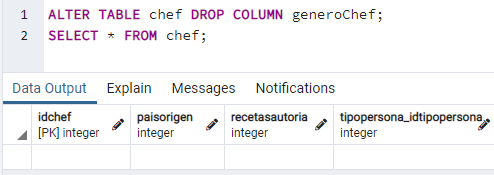
**AGREGAR UN CAMPO:**



El comando utilizado fue:

ALTER TABLE chef ADD COLUMN generoChef VARCHAR(25);

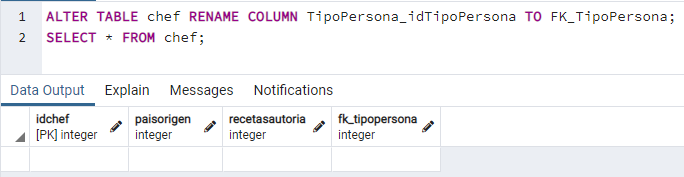
**QUITAR UN CAMPO:**



El comando utilizado fue:

ALTER TABLE chef DROP COLUMN generoChef;

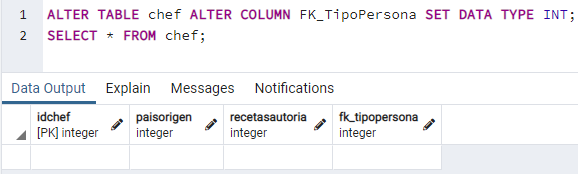
**RENOMBRAR UN CAMPO:**



El comando utilizado fue:

ALTER TABLE chef RENAME COLUMN TipoPersona\_idTipoPersona TO FK\_TipoPersona;

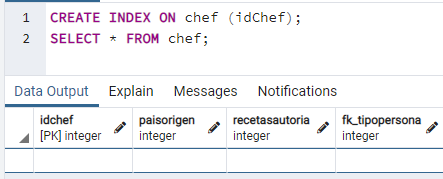
**CAMBIAR EL TIPO DE DATO A UNO EXISTENTE:**



El comando utilizado fue:

ALTER TABLE chef ALTER COLUMN FK\_TipoPersona SET DATA TYPE INT;

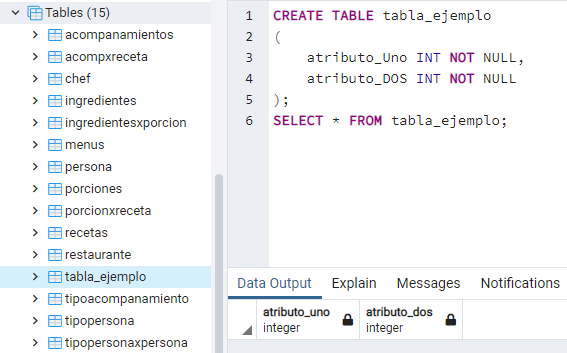
**CREAR UN ÍNDICE:**



El comando utilizado fue:

CREATE INDEX ON chef (idChef);

**CREAR UNA TABLA NUEVA:**



El comando utilizado fue:

CREATE TABLE tabla\_ejemplo (atributo\_Uno INT NOT NULL, atributo\_DOS INT NOT NULL);

**CREAR LLAVE PRIMARIA:**

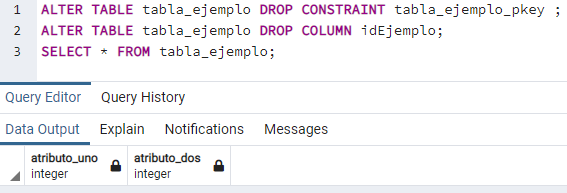


Los comandos utilizados fueron:

ALTER TABLE tabla\_ejemplo ADD COLUMN idEjemplo INT;

ALTER TABLE tabla\_ejemplo ADD PRIMARY KEY (idEjemplo);

**BORRAR LLAVE PRIMARIA:**



Los comandos utilizados fueron:

ALTER TABLE tabla\_ejemplo DROP CONSTRAINT tabla\_ejemplo\_pkey ;

ALTER TABLE tabla\_ejemplo DROP COLUMN idEjemplo;

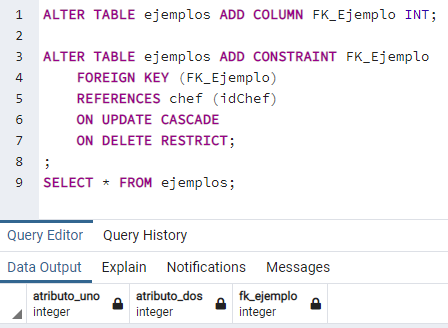
**RENOMBRER TABLA:**



El comando utilizado es:

ALTER TABLE tabla\_ejemplo RENAME TO ejemplos;

**CREAR LLAVE FORANEA:**



Los comandos utilizados fueron:

ALTER TABLE ejemplos ADD COLUMN FK\_Ejemplo INT;

ALTER TABLE ejemplos ADD CONSTRAINT FK\_Ejemplo

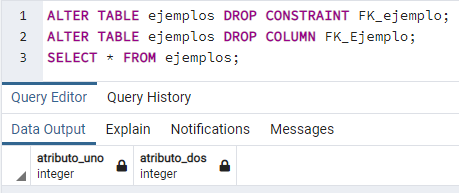
FOREIGN KEY (FK\_Ejemplo)

REFERENCES chef (idChef)

ON UPDATE CASCADE

ON DELETE RESTRICT;

**ELIMINAR LLAVE FORANEA:**

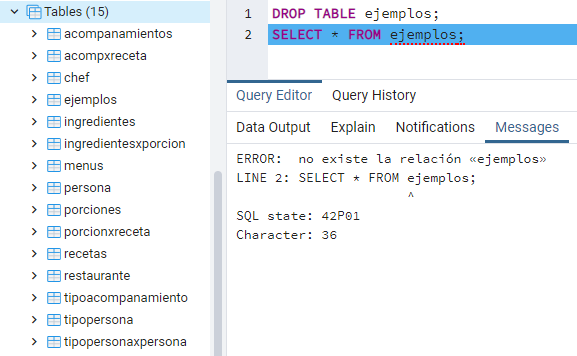


Los comandos utilizados fueron:

ALTER TABLE ejemplos DROP CONSTRAINT FK\_ejemplo;

ALTER TABLE ejemplos DROP COLUMN FK\_Ejemplo;

**ELIMINA UNA TABLA:**



El comando utilizado fue:

DROP TABLE ejemplos;