Instituto Politécnico Nacional

Escuela Superior de Cómputo

Bases de Datos

Práctica no. 2:

Consultas

Profesor: Euler Hernández Contreras

Alumno: Calva Hernández José Manuel

Grupo: 2CM12

Fecha de realización: 23/Agosto/17

Fecha de entrega: 07/Septiembre/17

## Índice

[Índice 2](#_Toc492314614)

[Marco Teórico 3](#_Toc492314615)

[Instrucciones 4](#_Toc492314616)

[1ª Parte 4](#_Toc492314617)

[2ª Parte 7](#_Toc492314618)

[Screenshots 8](#_Toc492314619)

[1ª Parte 8](#_Toc492314620)

[2ª Parte 12](#_Toc492314621)

[Conclusiones 15](#_Toc492314622)

[Referencias 15](#_Toc492314623)

## Marco Teórico

En general a las operaciones básicas de manipulación de datos que podemos realizar con SQL se les denomina **operaciones CRUD** (de **C***reate,***R***ead,***U***pdate and***D***elete*, o sea, *Crear, Leer, Actualizar y Borrar*, sería CLAB en español, pero no se usa). Lo verás utilizado de esta manera en muchos sitios, así que apréndete ese acrónimo.

Hay cuatro instrucciones para realizar estas tareas:

* **INSERT**: Inserta filas en una tabla. Se corresponde con la “C” de CRUD.
* **SELECT**: muestra información sobre los datos almacenados en la base de datos. Dicha información puede pertenecer a una o varias tablas. Es la “R”.
* **UPDATE**: Actualiza información de una tabla. Es, obviamente, la “U”.
* **DELETE**: Borra filas de una tabla. Se corresponde con la “D”.

Ahora nos vamos a centrar en la “R” de CRUD, es decir, en cómo recuperar la información que nos interesa de dentro de una base de datos, usando para ello el lenguaje de consulta o SQL.  Ya nos preocuparemos luego de cómo llegamos a introducir los datos primeramente.

Para realizar consultas sobre las tablas de las bases de datos disponemos de la instrucción SELECT. Con ella podemos consultar una o varias tablas. Es sin duda el comando más versátil del lenguaje SQL.

Existen muchas cláusulas asociadas a la sentencia SELECT (GROUP BY, ORDER, HAVING, UNION). También es una de las instrucciones en la que con más frecuencia los motores de bases de datos incorporan cláusulas adicionales al estándar.

**SELECT.-** Permite seleccionar las columnas que se van a mostrar y en el orden en que lo van a hacer. Simplemente es la instrucción que la base de datos interpreta como que vamos a solicitar información.

Se consideran consultas simples dado que involucran solamente una tabla, a medida que subimos de complejidad, iremos involucrando un mayor número de tablas a las consultas.

A la hora de realizar una consulta en un lenguaje de alto nivel, como SQL, primero se debe pasar por un examen o análisis léxico, un análisis sintáctico y una validación.

* **Analizador léxico**: identifica los componentes del lenguaje en el texto de la consulta.
* **Analizador sintáctico**: revisa la sintaxis de la consulta para determinar si está formulada de acuerdo a las reglas sintácticas de lenguaje de la consulta.
* **Validación**: Ha de comprobarse que todos los nombres de atributos y de relaciones sean válidos y tengan sentido desde el punto de vista semántico.

A continuación de esto se crea una representación interna de consulta, por lo regular en forma de estructura de datos de árbol o grafo. En seguida, el SGBD debe crear una estrategia de ejecución para obtener el resultado de la consulta a partir de los archivos internos de la base de datos, el proceso de elegir la más adecuada de estas se conoce como optimización.

## Instrucciones

### 1ª Parte

La práctica se dividió en dos partes, la primera de ellas fue similar a la anterior, siendo:

1. Primero debemos iniciar la terminal de MySQL para comenzar a ingresar los comandos relacionados, para ello tendremos dos opciones:
   1. En Windows: Abrir la consola afín al programa *“MySQL Command Line Client”*
   2. En Linux: Escribir en la terminal el siguiente comando “*mysql -u root -p -enter-“*
2. Crear la base de datos: Para ello escribiremos los siguiente comandos en la terminal abierta anteriormente  
     
   create database \_\_\_\_\_\_; // Creará la base de datos en el sistema  
   use ejemplo1; // Seleccionaremos la base de datos a usar
3. Construiremos en primer lugar las relaciones propietarias de nuestra base de datos  
     
   create table empleado(idEmp int not null primary key, nombre varchar(10), direccion varchar(300), tel int, genero varchar(10) );  
     
   create table cinemex(idCinemex int not null primary key, nombre varchar(30), dir varchar(300), tel int, email varchar(30) );  
     
   create table ec(idEmp int not null, idCinemex int not null, primary key(idEmp,idCinemex), foreign key(idEmp) references empleado(idEmp) on delete cascade on update cascade, foreign key(idCinemex) references cinemex(idCinemex) on delete cascade on update cascade );
4. Agregar email y salario en empleado  
     
   alter table empleado add column email varchar(60);  
     
   alter table empleado add column salario double;
5. Crear relacion y asociar con cinemex  
     
   gerente(idGerente, nombre, turno, salario, noCel)  
     
   create table gerente(idGerente int not null primary key, nombre varchar(50), turno varchar(20), salario double, noCel int, idCinemex int, foreign key(idCinemex) references Cinemex(idCinemex) on delete cascade on update cascade );

1. Cambiar tipo de dato en gerente, noCel a varchar  
     
   alter table gerente modify column noCel varchar(15);
2. Renombrar empleado por asociado  
     
   alter table empleado rename as asociado;
3. Aumentar de tamaño el campo direccion en asociado a 350  
     
   alter table asociado modify column direccion varchar(350);
4. Modificar la definicion de la llave primaria en cinemex, teniendo una nueva PK(idCinemex,nombre)  
     
    a) eliminar las FK de las relaciones ec y gerente  
     
    show create table ec;  
     
    alter table ec drop foreign key ec\_ibfk\_2; // eliminar asociacion con referencia de show create table  
     
    show create table gerente;  
     
    alter table gerente drop foreign key gerente\_ibfk\_1;  
     
    b) eliminar llave primaria de Cinemex  
     
    alter table Cinemex drop primary key;  
     
    c) agregar la nueva llave primaria compuesta por idCinemex y nombre  
     
    alter table Cinemex add primary key(idCinemex,nombre);  
     
    d) establecer las nuevas relaciones, creando primero los nuevos campos para las nuevas relaciones  
     
    alter table ec add column nombreCinemex varchar(30);  
     
    alter table ec add foreign key(idCinemex,nombreCinemex) references Cinemex(idCinemex,nombre) on delete cascade on update cascade;  
     
    alter table gerente add column nombreCinemex varchar(30);  
     
    alter table gerente add foreign key(idCinemex,nombreCinemex) references Cinemex(idCinemex,nombre) on delete cascade on update cascade;
5. Crear la nueva relación y asociarla con Cinemex  
     
   cartelera(idCartelera, nombre, fechaInicio, fechaFin, clasificacion, idCinemex, nombreCinemex)  
     
   create table cartelera(idCartelera int not null primary key, nombre varchar(60), fechaInicio date, fechaFin date, clasificacion varchar(5), idCinemex int, nombreCinemex varchar(30), foreign key(idCinemex,nombreCinemex) references Cinemex(idCinemex,nombre) on delete cascade on update cascade );
6. Cómo crear un respaldo
   1. En Windows:
      1. Ingresar a la terminal (cmd)
      2. Buscar la ruta (path) de la carpeta MySQL/bin
      3. Colocarnos en la ruta bin desde la terminal por medio del comando *cd*
      4. Ingresar el siguiente comando  
         mysqldump -u root -p \_\_\_\_\_\_\_\_\_>\_\_\_\_\_\_\_\_\_\_\_  
         Donde en el primer espacio ingresaremos el nombre de la base de datos, y en el segundo la ruta donde se guardará, seguido del nombre que le pondremos al archivo con terminación .sql
   2. En Linux:
      1. Ingresaremos a la terminal
      2. Ejecutaremos el siguiente comando  
         mysqldump -u root -p \_\_\_\_\_\_\_\_\_>\_\_\_\_\_\_\_\_\_\_\_  
         Donde en el primer espacio ingresaremos el nombre de la base de datos, y en el segundo la ruta donde se guardará, seguido del nombre que le pondremos al archivo con terminación .sql

Comandos adicionales:

* show tables; // mostrará las tablas que contiene nuestra base de datos
* desc/describe \_\_\_; // muestra la tabla especificada, así como sus componentes
* show create table \_\_\_; // mostrará las relaciones específicas de una tabla
* drop database; // desecha la tabla actual, nos permitirá volver a construirla en caso de error
* select database; // muestra el nombre de la base de datos actual

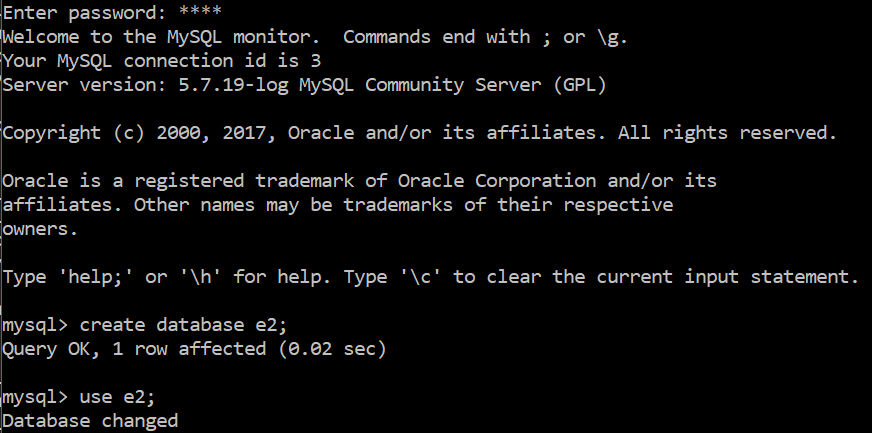
### 2ª Parte

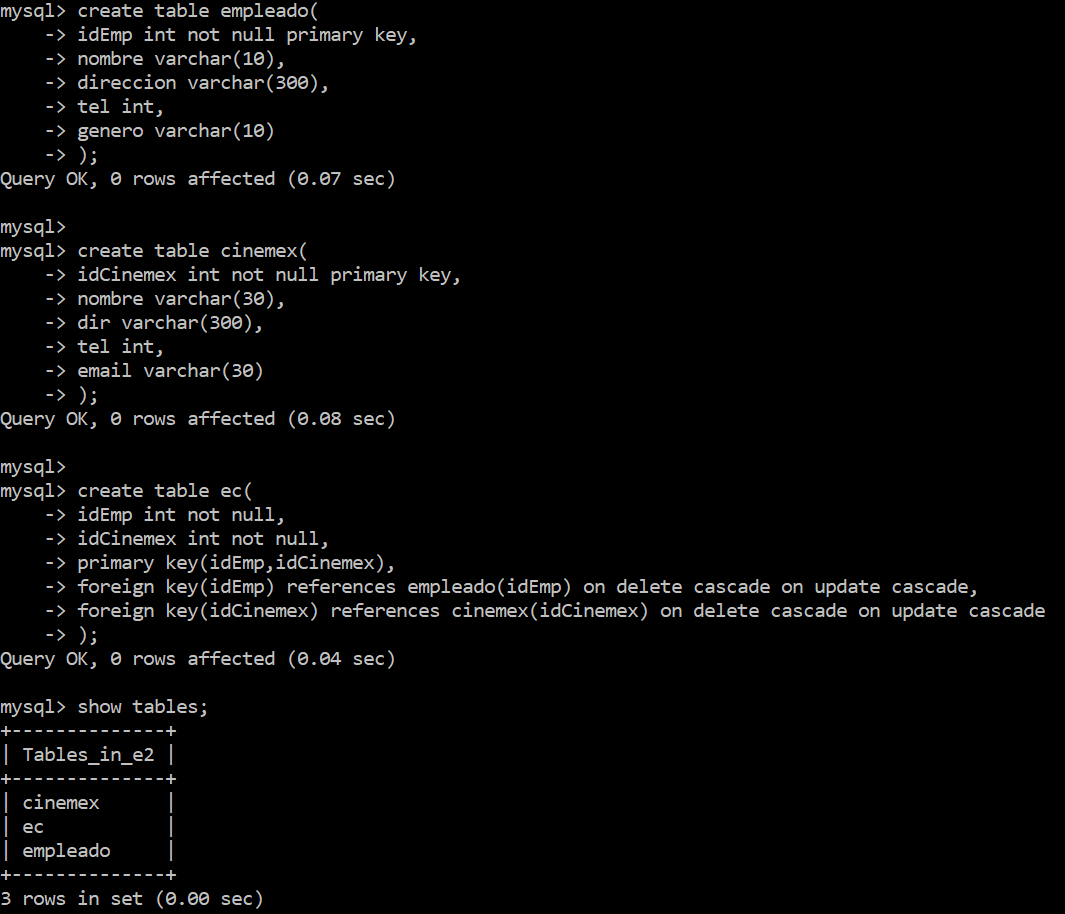
La segunda parte fue una serie de consultas sobre una base de datos proporcionada:

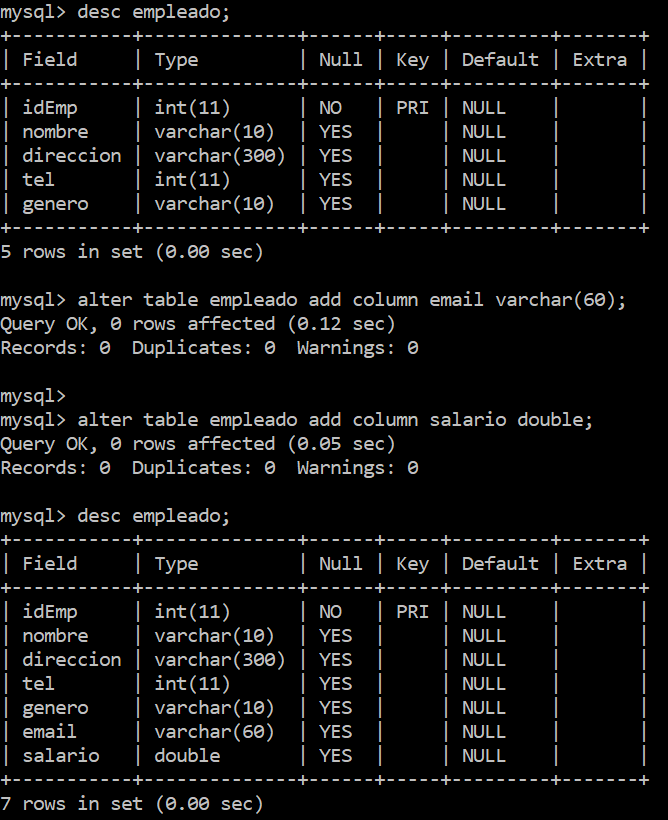
1. Cargar el script sams.sql  
     
    a) Crear una bd  
    create database s1;  
     
    b) use  
    use s1;  
     
    c) source...   
    source // arrastrar a la terminal (en ubuntu quitar comillas) para la ruta
2. Consultas:
   1. select \* from club; // mostrar todo el contenido
   2. select idclub, nombre from club; // limitar a los nombres del club
   3. select idclub, nombre from club order by 2; // ordenar por el segundo campo
   4. select idclub, nombre from club order by 2 DESC;
   5. select COUNT(\*) from proveedor; // proveedores
   6. select COUNT(\*) as proveedor from proveedor; // renombrar contador
   7. select COUNT(\*) as "numero de proveedores" from proveedor; // renombrar mas de una palabra
   8. select nombre as socios from socio; // consultar nombre socios
   9. select nombre as socios from socio order by nombre;
   10. select nombre from servicio; // conocer los servicios
   11. select nombre from servicio order by 1;
   12. select nombre, precioUnitario from producto where precioUnitario between 100 and 3000 order by 2; // consulta compuesta para conocer nombre y precio unitario de los productos que valen entre 100 a 3000
   13. select \* from gerente; // mostrar todo el contenido de los gerentes
   14. select \* from gerente order by nombre;

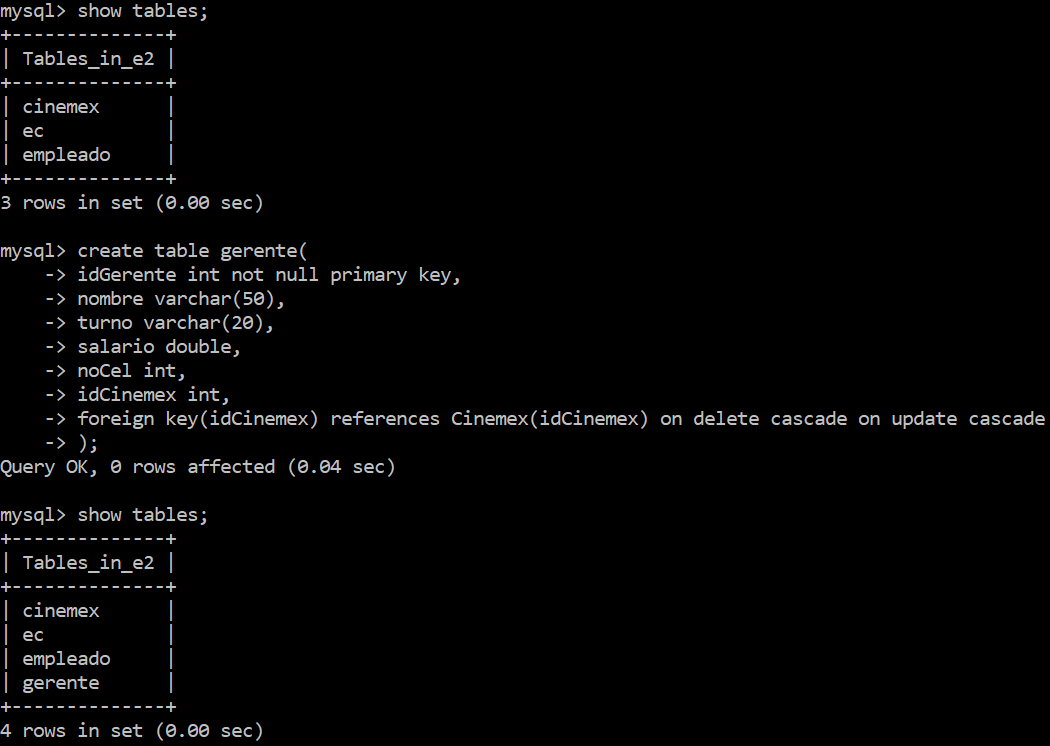
## Screenshots

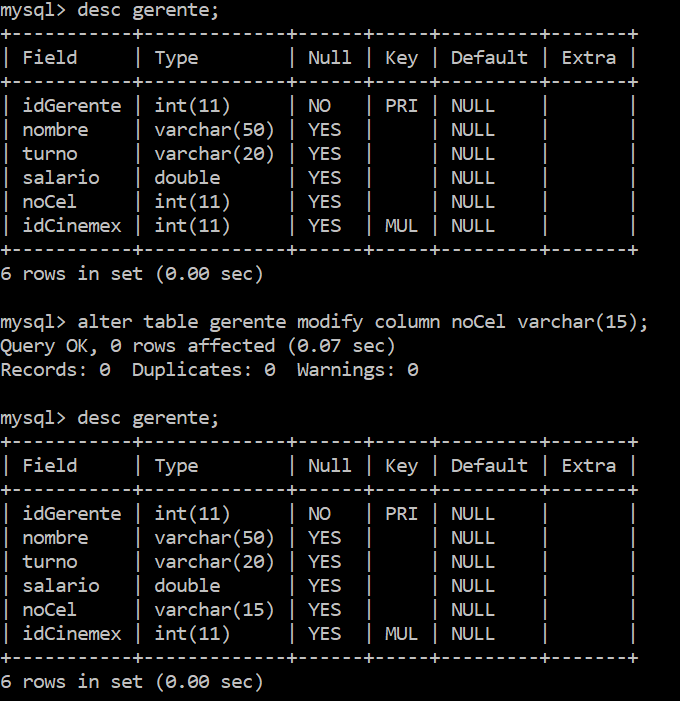
### 1ª Parte

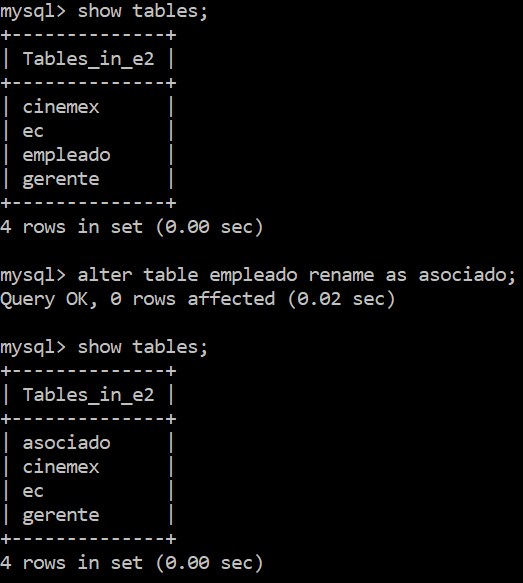


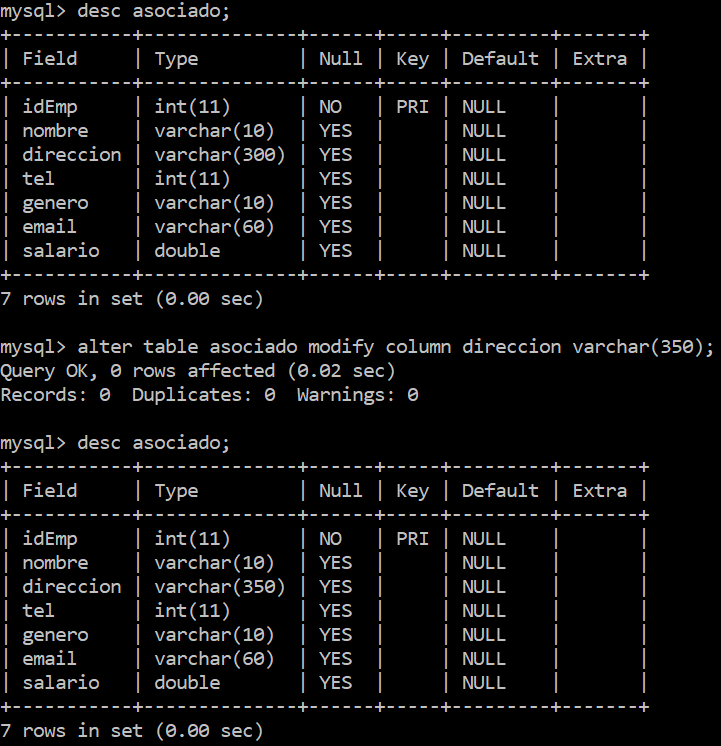


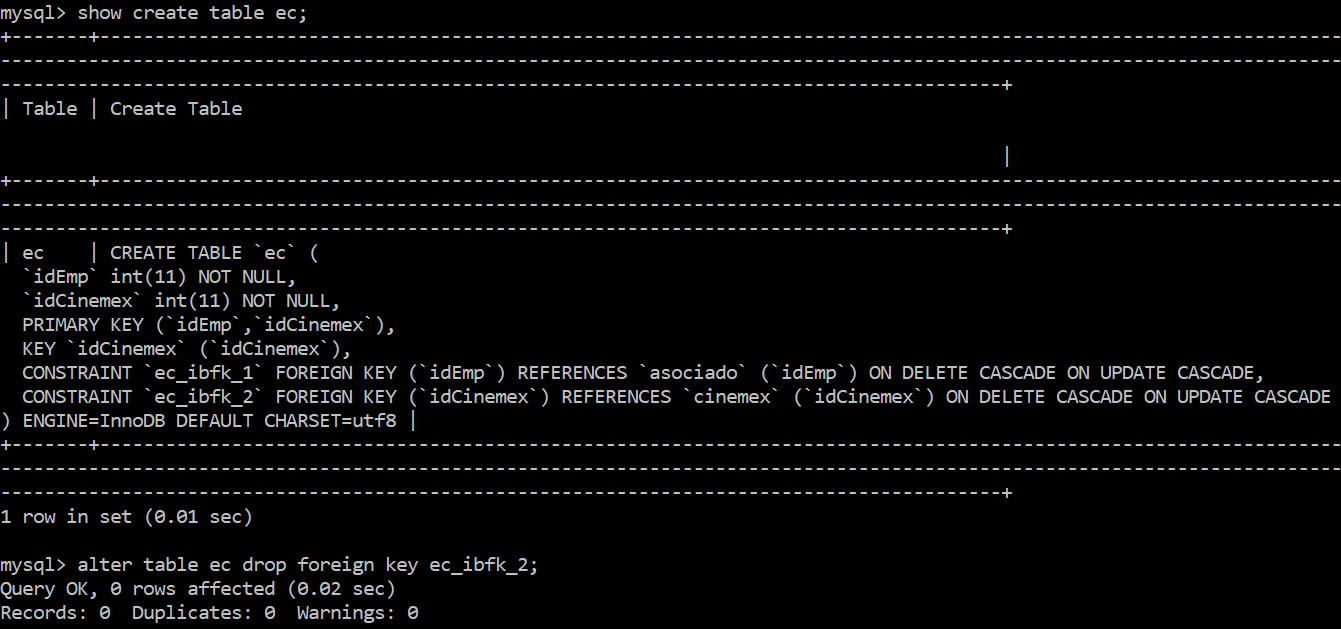


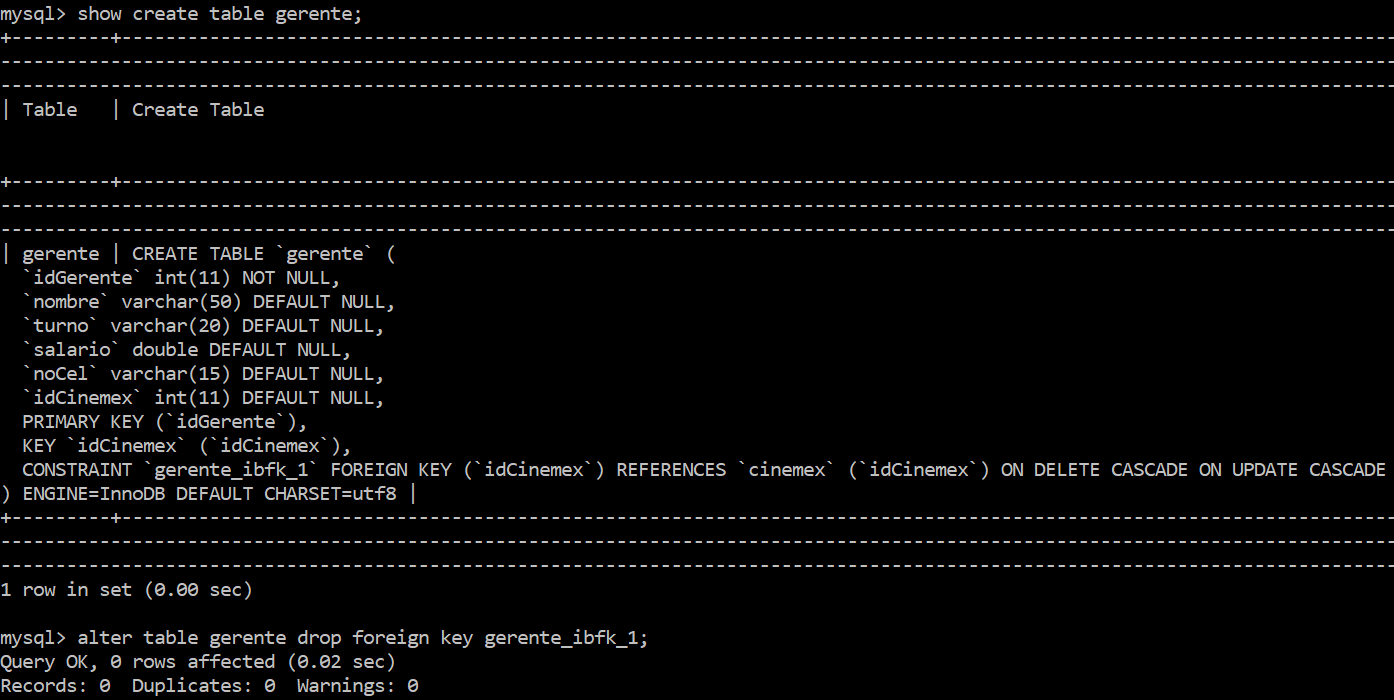


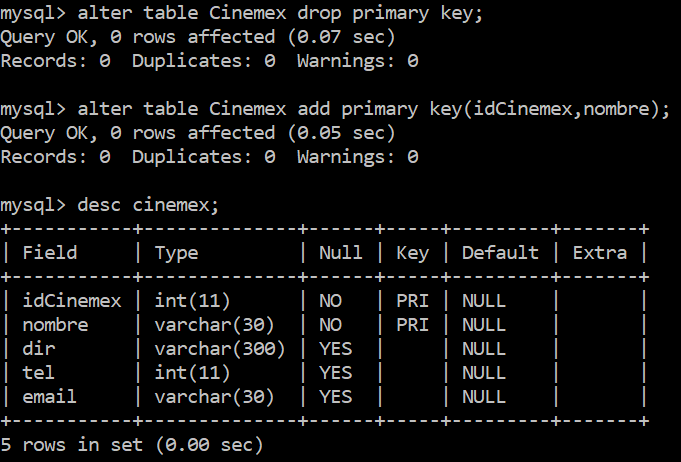


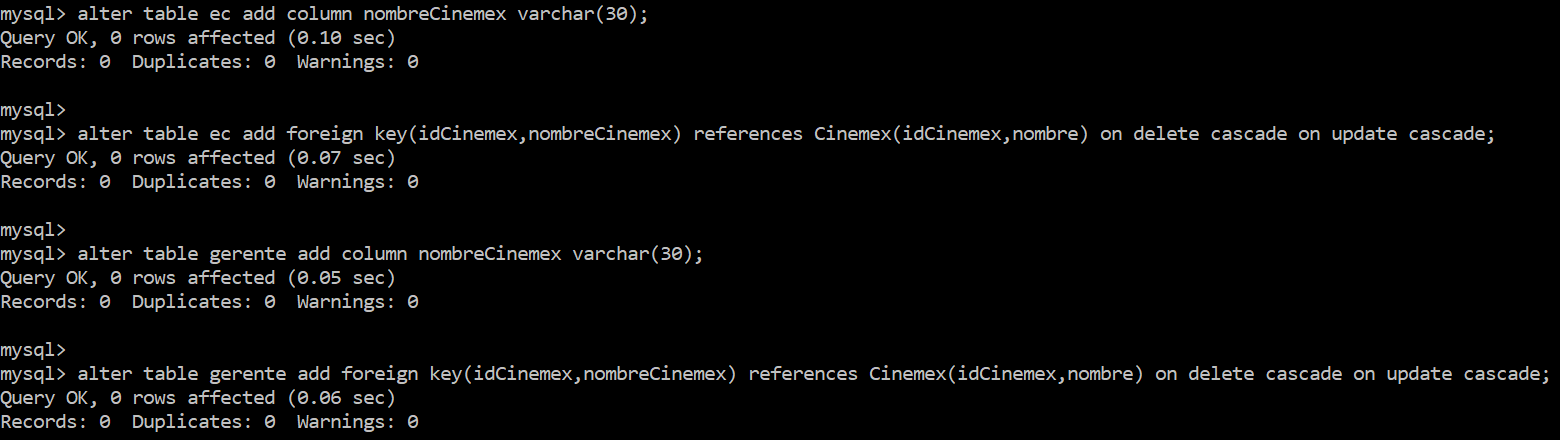


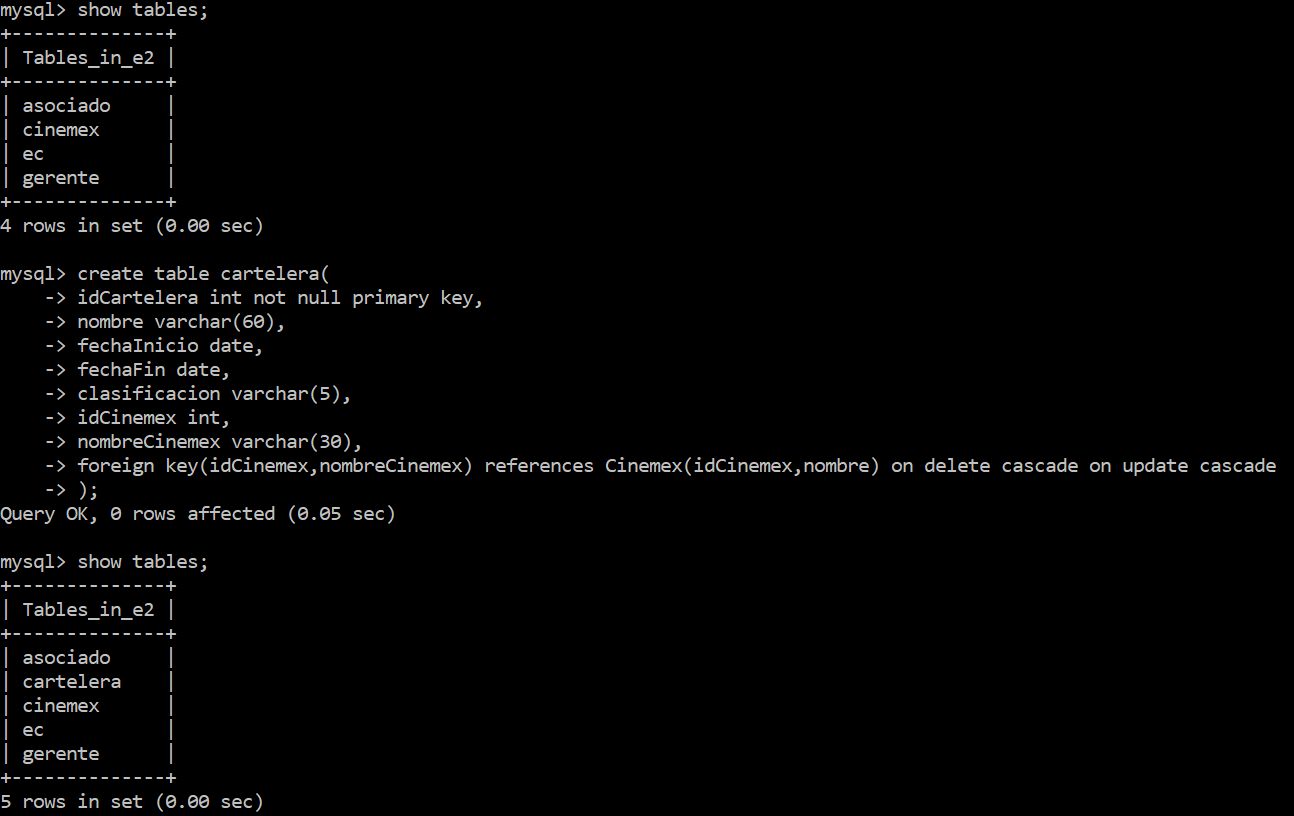




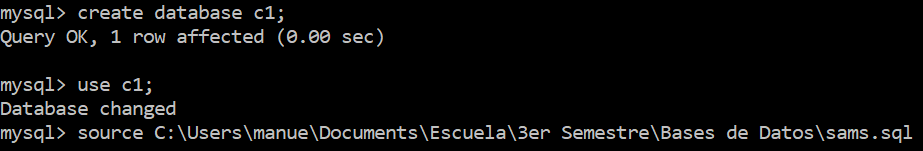


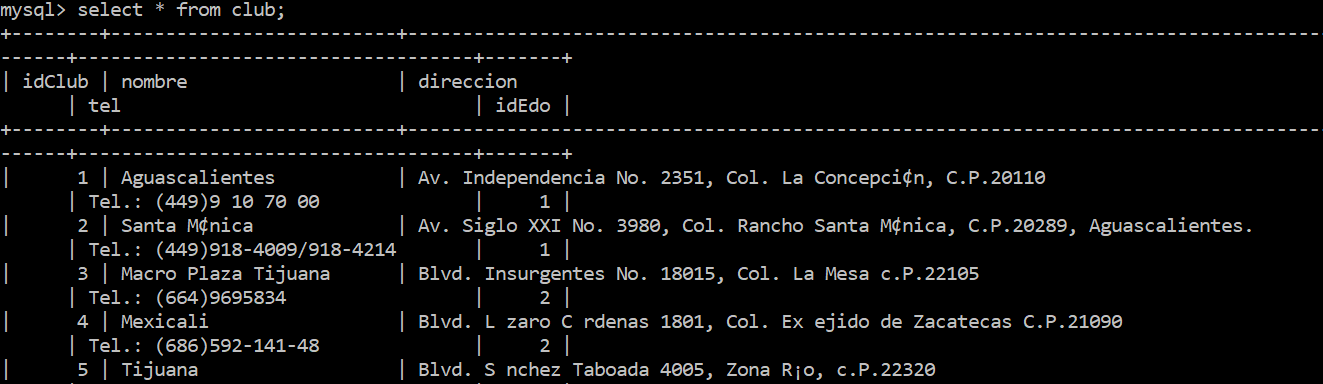




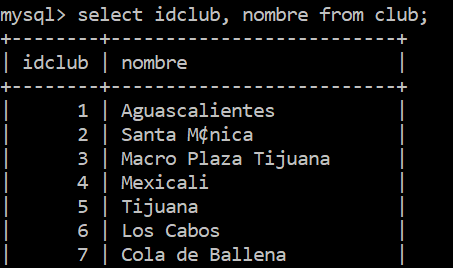


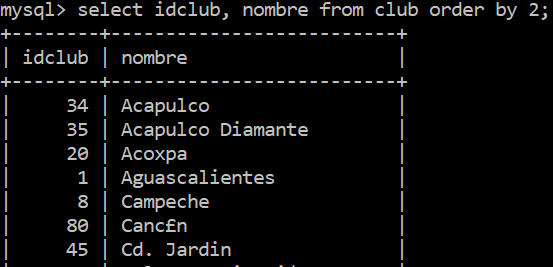
### 2ª Parte

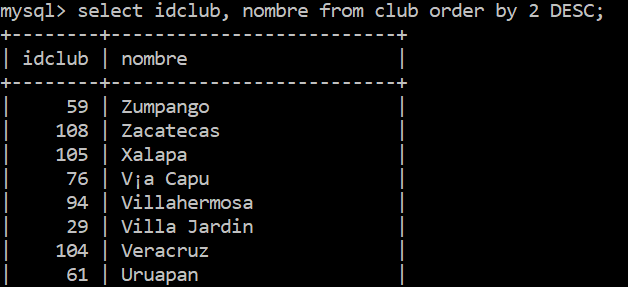


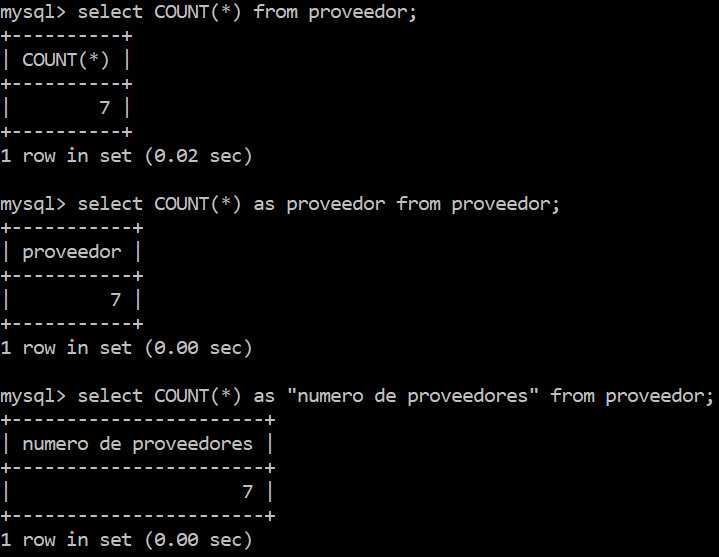


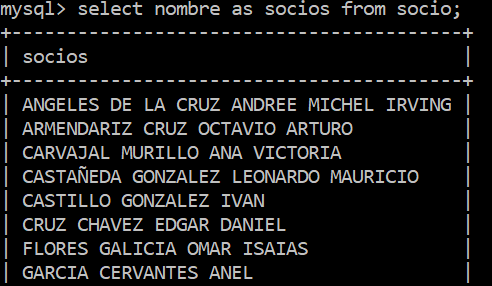
https://i.gyazo.com/752085282633d51ac8a4442bad4aa408.png

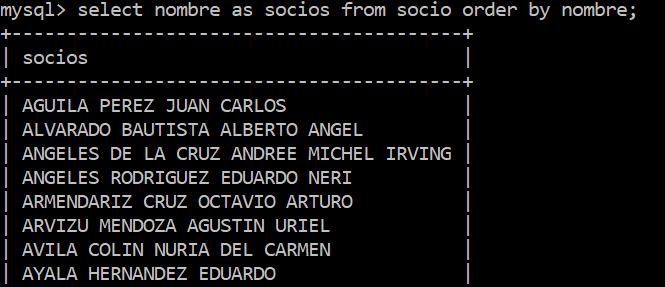


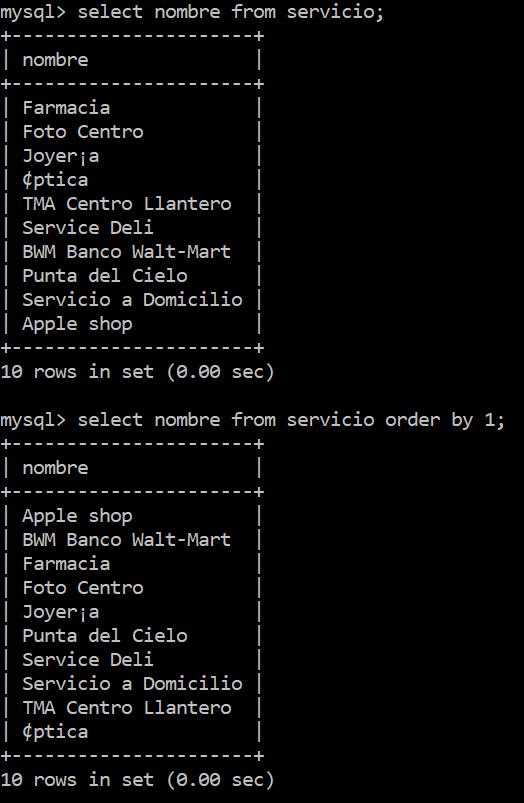


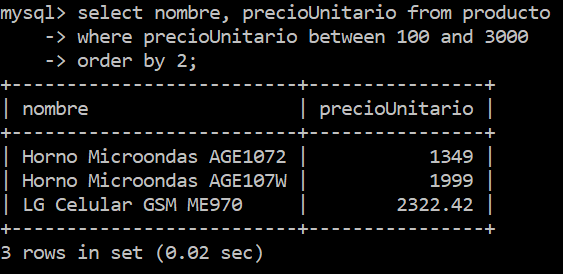


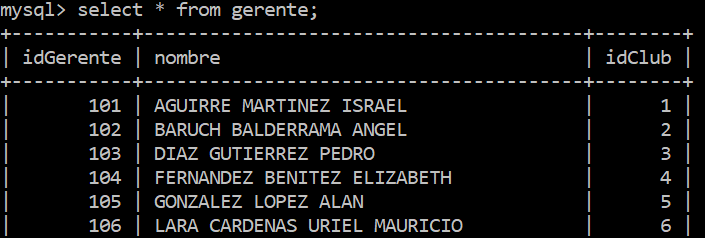


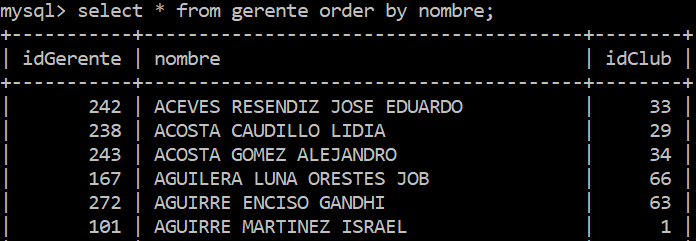












## Conclusiones

El inicio de esta práctica no distó mucho de la anterior, sin embargo, conocimos algunos métodos nuevos como la forma de agregar una PK compuesta, que a simple vista puede parecer sencillo, pero a la hora de intentarlo debemos tomar en cuenta todas sus posibles relaciones en la base de datos.

Por otra parte, la segunda sección me pareció bastante interesante, ya que pudimos comenzar a realizar consultas, algo nuevo para mí; y los múltiples comandos usados me parecieron bastante más sencillos de lo que esperaba, nuevamente fue algo bastante claro ya que las órdenes ejecutadas no tienen mayor dificultad. Sin embargo, se debe de tomar práctica en las consultas para poder realizarlas con una mayor facilidad, y de manera más fluida.

## Referencias

*Fundamentos de SQL: Cómo realizar consultas simples con SELECT* - campusMVP.es. (2017). campusMVP.es. Retrieved 4 September 2017, from https://www.campusmvp.es/recursos/post/Fundamentos-de-SQL-Como-realizar-consultas-simples-con-SELECT.aspx

*Tema 2 Las consultas simples.* (2017). Aulaclic.es. Retrieved 4 September 2017, from http://www.aulaclic.es/sql/t\_2\_1.htm

Ramez, E., & Navathe, S. (2000). *Sistemas de Bases de Datos: Conceptos Fundamentales* (1st ed.). México: Pearson Educación.