



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

Marco Teórico 3

Instrucciones 4

 1ª Parte 4

 2ª Parte 7

Screenshots 8

 1ª Parte 8

 2ª Parte 12

Conclusiones 15

Referencias..... 15

En general a las operaciones básicas de manipulación de datos que podemos realizar con SQL se les denomina **operaciones CRUD** (de *Create, Read, Update and Delete*, 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:
 - a. En Windows: Abrir la consola afín al programa *"MySQL Command Line Client"*
 - b. 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 );
```

6. Cambiar tipo de dato en gerente, noCel a varchar

```
alter table gerente modify column noCel varchar(15);
```

7. Renombrar empleado por asociado

```
alter table empleado rename as asociado;
```

8. Aumentar de tamaño el campo direccion en asociado a 350

```
alter table asociado modify column direccion varchar(350);
```

9. 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;
```

10. 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);

11. Cómo crear un respaldo

a. En Windows:

- i. Ingresar a la terminal (cmd)
- ii. Buscar la ruta (path) de la carpeta MySQL/bin
- iii. Colocarnos en la ruta bin desde la terminal por medio del comando *cd*
- iv. 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

b. En Linux:

- i. Ingresaremos a la terminal
- ii. 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:

- a. `select * from club;` // mostrar todo el contenido
- b. `select idclub, nombre from club;` // limitar a los nombres del club
- c. `select idclub, nombre from club order by 2;` // ordenar por el segundo campo
- d. `select idclub, nombre from club order by 2 DESC;`
- e. `select COUNT(*) from proveedor;` // proveedores
- f. `select COUNT(*) as proveedor from proveedor;` // renombrar contador
- g. `select COUNT(*) as "numero de proveedores" from proveedor;` // renombrar mas de una palabra
- h. `select nombre as socios from socio;` // consultar nombre socios
- i. `select nombre as socios from socio order by nombre;`
- j. `select nombre from servicio;` // conocer los servicios
- k. `select nombre from servicio order by 1;`
- l. `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
- m. `select * from gerente;` // mostrar todo el contenido de los gerentes
- n. `select * from gerente order by nombre;`

Screenshots

1ª Parte

```
Enter password: ****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3
Server version: 5.7.19-log MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> create database e2;
Query OK, 1 row affected (0.02 sec)

mysql> use e2;
Database changed
```

```
mysql> create table empleado(
-> idEmp int not null primary key,
-> nombre varchar(10),
-> direccion varchar(300),
-> tel int,
-> genero varchar(10)
-> );
Query OK, 0 rows affected (0.07 sec)

mysql>
mysql> create table cinemex(
-> idCinemex int not null primary key,
-> nombre varchar(30),
-> dir varchar(300),
-> tel int,
-> email varchar(30)
-> );
Query OK, 0 rows affected (0.08 sec)

mysql>
mysql> 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
-> );
Query OK, 0 rows affected (0.04 sec)

mysql> show tables;
+-----+
| Tables_in_e2 |
+-----+
| cinemex      |
| ec           |
| empleado     |
+-----+
3 rows in set (0.00 sec)
```

```
mysql> desc empleado;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idEmp | int(11) | NO | PRI | NULL | |
| nombre | varchar(10) | YES | | NULL | |
| direccion | varchar(300) | YES | | NULL | |
| tel | int(11) | YES | | NULL | |
| genero | varchar(10) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> alter table empleado add column email varchar(60);
Query OK, 0 rows affected (0.12 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
mysql> alter table empleado add column salario double;
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc empleado;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idEmp | int(11) | NO | PRI | NULL | |
| nombre | varchar(10) | YES | | NULL | |
| direccion | varchar(300) | YES | | NULL | |
| tel | int(11) | YES | | NULL | |
| genero | varchar(10) | YES | | NULL | |
| email | varchar(60) | YES | | NULL | |
| salario | double | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```



```
mysql> show tables;
+-----+
| Tables_in_e2 |
+-----+
| cinemex      |
| ec           |
| empleado     |
+-----+
3 rows in set (0.00 sec)

mysql> 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
  -> );
Query OK, 0 rows affected (0.04 sec)

mysql> show tables;
+-----+
| Tables_in_e2 |
+-----+
| cinemex      |
| ec           |
| empleado     |
| gerente      |
+-----+
4 rows in set (0.00 sec)
```

```
mysql> desc gerente;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idGerente  | int(11)   | NO   | PRI | NULL    |       |
| nombre     | varchar(50) | YES  |     | NULL    |       |
| turno      | varchar(20) | YES  |     | NULL    |       |
| salario    | double    | YES  |     | NULL    |       |
| noCel      | int(11)   | YES  |     | NULL    |       |
| idCinemex  | int(11)   | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql> alter table gerente modify column noCel varchar(15);
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc gerente;
+-----+-----+-----+-----+-----+-----+
| Field      | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idGerente  | int(11)   | NO   | PRI | NULL    |       |
| nombre     | varchar(50) | YES  |     | NULL    |       |
| turno      | varchar(20) | YES  |     | NULL    |       |
| salario    | double    | YES  |     | NULL    |       |
| noCel      | varchar(15) | YES  |     | NULL    |       |
| idCinemex  | int(11)   | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

```
mysql> show tables;
+-----+
| Tables_in_e2 |
+-----+
| cinemex      |
| ec           |
| empleado     |
| gerente      |
+-----+
4 rows in set (0.00 sec)

mysql> alter table empleado rename as asociado;
Query OK, 0 rows affected (0.02 sec)

mysql> show tables;
+-----+
| Tables_in_e2 |
+-----+
| asociado     |
| cinemex      |
| ec           |
| gerente      |
+-----+
4 rows in set (0.00 sec)
```

```
mysql> desc asociado;
```

Field	Type	Null	Key	Default	Extra
idEmp	int(11)	NO	PRI	NULL	
nombre	varchar(10)	YES		NULL	
direccion	varchar(300)	YES		NULL	
tel	int(11)	YES		NULL	
genero	varchar(10)	YES		NULL	
email	varchar(60)	YES		NULL	
salario	double	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> alter table asociado modify column direccion varchar(350);
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> desc asociado;
```

Field	Type	Null	Key	Default	Extra
idEmp	int(11)	NO	PRI	NULL	
nombre	varchar(10)	YES		NULL	
direccion	varchar(350)	YES		NULL	
tel	int(11)	YES		NULL	
genero	varchar(10)	YES		NULL	
email	varchar(60)	YES		NULL	
salario	double	YES		NULL	

7 rows in set (0.00 sec)

```
mysql> show create table ec;
```

Table	Create Table
ec	<pre>CREATE TABLE `ec` (`idEmp` int(11) NOT NULL, `idCinemex` int(11) NOT NULL, PRIMARY KEY (`idEmp`,`idCinemex`), KEY `idCinemex` (`idCinemex`), CONSTRAINT `ec_ibfk_1` FOREIGN KEY (`idEmp`) REFERENCES `asociado` (`idEmp`) ON DELETE CASCADE ON UPDATE CASCADE, CONSTRAINT `ec_ibfk_2` FOREIGN KEY (`idCinemex`) REFERENCES `cinemex` (`idCinemex`) ON DELETE CASCADE ON UPDATE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=utf8</pre>

1 row in set (0.01 sec)

```
mysql> alter table ec drop foreign key ec_ibfk_2;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> show create table gerente;
```

Table	Create Table
gerente	<pre>CREATE TABLE `gerente` (`idGerente` int(11) NOT NULL, `nombre` varchar(50) DEFAULT NULL, `turno` varchar(20) DEFAULT NULL, `salario` double DEFAULT NULL, `noCel` varchar(15) DEFAULT NULL, `idCinemex` int(11) DEFAULT NULL, PRIMARY KEY (`idGerente`), KEY `idCinemex` (`idCinemex`), CONSTRAINT `gerente_ibfk_1` FOREIGN KEY (`idCinemex`) REFERENCES `cinemex` (`idCinemex`) ON DELETE CASCADE ON UPDATE CASCADE) ENGINE=InnoDB DEFAULT CHARSET=utf8</pre>

1 row in set (0.00 sec)

```
mysql> alter table gerente drop foreign key gerente_ibfk_1;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> alter table Cinemex drop primary key;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table Cinemex add primary key(idCinemex,nombre);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc cinemex;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idCinemex  | int(11)       | NO   | PRI | NULL    |       |
| nombre     | varchar(30)   | NO   | PRI | NULL    |       |
| dir        | varchar(300)  | YES  |     | NULL    |       |
| tel        | int(11)       | YES  |     | NULL    |       |
| email      | varchar(30)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> alter table ec add column nombreCinemex varchar(30);
Query OK, 0 rows affected (0.10 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
mysql> alter table ec add foreign key(idCinemex,nombreCinemex) references Cinemex(idCinemex,nombre) on delete cascade on update cascade;
Query OK, 0 rows affected (0.07 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
mysql> alter table gerente add column nombreCinemex varchar(30);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
mysql> alter table gerente add foreign key(idCinemex,nombreCinemex) references Cinemex(idCinemex,nombre) on delete cascade on update cascade;
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> show tables;
+-----+
| Tables_in_e2 |
+-----+
| asociado     |
| cinemex      |
| ec           |
| gerente      |
+-----+
4 rows in set (0.00 sec)

mysql> 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
  -> );
Query OK, 0 rows affected (0.05 sec)

mysql> show tables;
+-----+
| Tables_in_e2 |
+-----+
| asociado     |
| cartelera    |
| cinemex      |
| ec           |
| gerente      |
+-----+
5 rows in set (0.00 sec)
```

2ª Parte

```
mysql> create database c1;
Query OK, 1 row affected (0.00 sec)

mysql> use c1;
Database changed
mysql> source C:\Users\manue\Documents\Escuela\3er Semestre\Bases de Datos\sams.sql
```

```
mysql> select * from club;
+-----+-----+-----+
| idClub | nombre                | direccion                |
|-----|-----|-----|
| 1 | Aguascalientes        | Av. Independencia No. 2351, Col. La Concepci n, C.P.20110 |
| 2 | Santa M nica          | Av. Siglo XXI No. 3980, Col. Rancho Santa M nica, C.P.20289, Aguascalientes. |
| 3 | Macro Plaza Tijuana    | Blvd. Insurgentes No. 18015, Col. La Mesa c.P.22105 |
| 4 | Mexicali              | Blvd. L zaro C rdenas 1801, Col. Ex ejido de Zacatecas C.P.21090 |
| 5 | Tijuana               | Blvd. S  nchez Taboada 4005, Zona R jo, c.P.22320 |
```

```
108 rows in set (0.00 sec)
```

```
mysql> select idclub, nombre from club;
+-----+-----+
| idclub | nombre                |
|-----|-----|
| 1 | Aguascalientes        |
| 2 | Santa M nica          |
| 3 | Macro Plaza Tijuana    |
| 4 | Mexicali              |
| 5 | Tijuana               |
| 6 | Los Cabos             |
| 7 | Cola de Ballena       |
```

```
mysql> select idclub, nombre from club order by 2;
+-----+-----+
| idclub | nombre                |
|-----|-----|
| 34 | Acapulco              |
| 35 | Acapulco Diamante     |
| 20 | Acoxta                |
| 1 | Aguascalientes        |
| 8 | Campeche              |
| 80 | Canc n                |
| 45 | Cd. Jard n            |
```

```
mysql> select idclub, nombre from club order by 2 DESC;
+-----+-----+
| idclub | nombre                |
|-----|-----|
| 59 | Zumpango              |
| 108 | Zacatecas             |
| 105 | Xalapa                |
| 76 | V ja Capu             |
| 94 | Villahermosa          |
| 29 | Villa Jard n          |
| 104 | Veracruz              |
| 61 | Uruapan               |
```

```
mysql> select COUNT(*) from proveedor;
+-----+
| COUNT(*) |
+-----+
|          7 |
+-----+
1 row in set (0.02 sec)

mysql> select COUNT(*) as proveedor from proveedor;
+-----+
| proveedor |
+-----+
|          7 |
+-----+
1 row in set (0.00 sec)

mysql> select COUNT(*) as "numero de proveedores" from proveedor;
+-----+
| numero de proveedores |
+-----+
|          7 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> select nombre as socios from socio;
+-----+
| socios |
+-----+
| ANGELES DE LA CRUZ ANDREE MICHEL IRVING |
| ARMENDARIZ CRUZ OCTAVIO ARTURO |
| CARVAJAL MURILLO ANA VICTORIA |
| CASTAÑEDA GONZALEZ LEONARDO MAURICIO |
| CASTILLO GONZALEZ IVAN |
| CRUZ CHAVEZ EDGAR DANIEL |
| FLORES GALICIA OMAR ISAIAS |
| GARCIA CERVANTES ANEL |
```

```
mysql> select nombre as socios from socio order by nombre;
+-----+
| socios |
+-----+
| AGUILA PEREZ JUAN CARLOS |
| ALVARADO BAUTISTA ALBERTO ANGEL |
| ANGELES DE LA CRUZ ANDREE MICHEL IRVING |
| ANGELES RODRIGUEZ EDUARDO NERI |
| ARMENDARIZ CRUZ OCTAVIO ARTURO |
| ARVIZU MENDOZA AGUSTIN URIEL |
| AVILA COLIN NURIA DEL CARMEN |
| AYALA HERNANDEZ EDUARDO |
```

```
mysql> select nombre from servicio;
+-----+
| nombre |
+-----+
| Farmacia |
| Foto Centro |
| Joyería |
| Óptica |
| TMA Centro Llantero |
| Service Deli |
| BWM Banco Walt-Mart |
| Punta del Cielo |
| Servicio a Domicilio |
| Apple shop |
+-----+
10 rows in set (0.00 sec)

mysql> select nombre from servicio order by 1;
+-----+
| nombre |
+-----+
| Apple shop |
| BWM Banco Walt-Mart |
| Farmacia |
| Foto Centro |
| Joyería |
| Punta del Cielo |
| Service Deli |
| Servicio a Domicilio |
| TMA Centro Llantero |
| Óptica |
+-----+
10 rows in set (0.00 sec)
```

```
mysql> select nombre, precioUnitario from producto
-> where precioUnitario between 100 and 3000
-> order by 2;
```

nombre	precioUnitario
Horno Microondas AGE1072	1349
Horno Microondas AGE107W	1999
LG Celular GSM ME970	2322.42

```
3 rows in set (0.02 sec)
```

```
mysql> select * from gerente;
```

idGerente	nombre	idClub
101	AGUIRRE MARTINEZ ISRAEL	1
102	BARUCH BALDERRAMA ANGEL	2
103	DIAZ GUTIERREZ PEDRO	3
104	FERNANDEZ BENITEZ ELIZABETH	4
105	GONZALEZ LOPEZ ALAN	5
106	LARA CARDENAS URIEL MAURICIO	6

```
mysql> select * from gerente order by nombre;
```

idGerente	nombre	idClub
242	ACEVES RESENDIZ JOSE EDUARDO	33
238	ACOSTA CAUDILLO LIDIA	29
243	ACOSTA GOMEZ ALEJANDRO	34
167	AGUILERA LUNA ORESTES JOB	66
272	AGUIRRE ENCISO GANDHI	63
101	AGUIRRE MARTINEZ ISRAEL	1

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). Aulaclíc.es. Retrieved 4 September 2017, from http://www.aulaclíc.es/sql/t_2_1.htm

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