

Reporte: Práctica 3

Carlos Tonatihu Barrera Pérez
Profesor: Hernández Contreras Euler
Bases de Datos
Grupo: 2CM1

3 de marzo de 2017

Índice

1. Marco Teórico	1
2. Desarrollo	2
3. Conclusiones	12
Referencias	12

1. Marco Teórico

El lenguaje de definición de datos SQL (el cual es usado para realizar estas practicas) se utiliza para crear relaciones con esquemas especificados.[1] SQL cuenta con tres principales clausulas a la hora de hacer consultas:

- La clausula **select** se corresponde con la operación de proyección del álgebra relacional. Se usa para obtener una relación de los atributos deseados en el resultado de una consulta.
- La clausula **from** se corresponde con la operación producto cartesiano del álgebra relacional. Genera una lista de las relaciones que deben ser analizadas en la evaluación de la expresión
- La clausula **where** se corresponde con el predicado selección del álgebra relacional. Es un predicado que engloba a los atributos de las relaciones que aparecen en la clausula **from**.

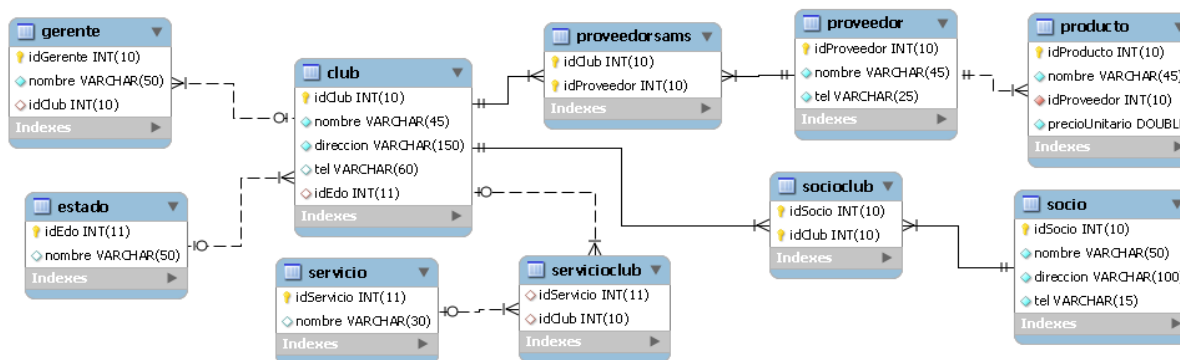
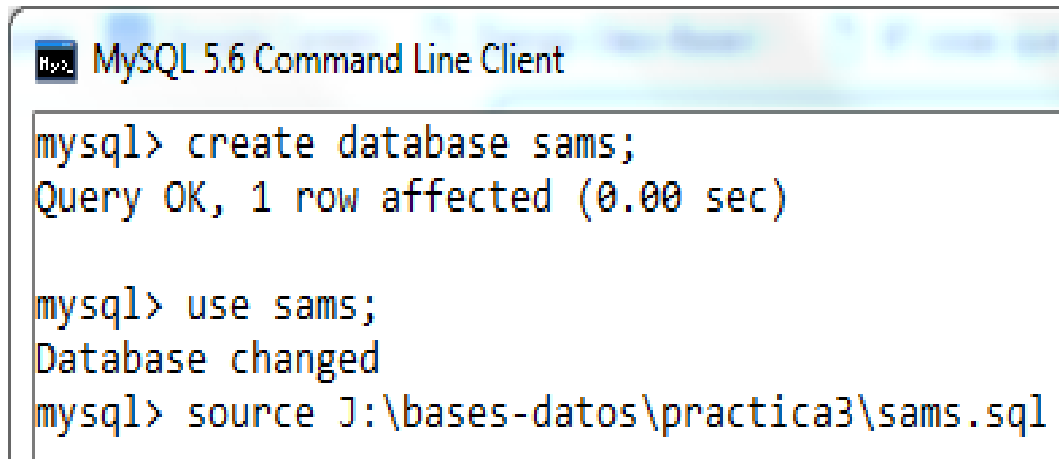


Figura 1: Diagrama de la base de datos usada en esta práctica.

2. Desarrollo

En esta practica se trabajo con un script llamado sams.sql por lo que se creo una base de datos para trabajar y despues se importo el contenido del script.

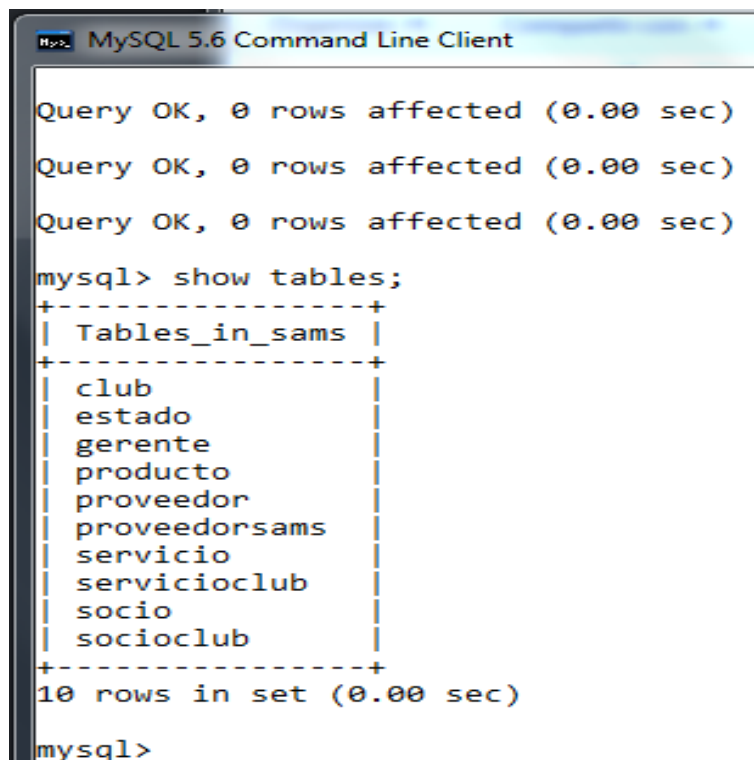


```
MySQL 5.6 Command Line Client

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

mysql> use sams;
Database changed
mysql> source J:\bases-datos\practica3\sams.sql
```

Figura 2: Creación y uso de la base.



```
MySQL 5.6 Command Line Client

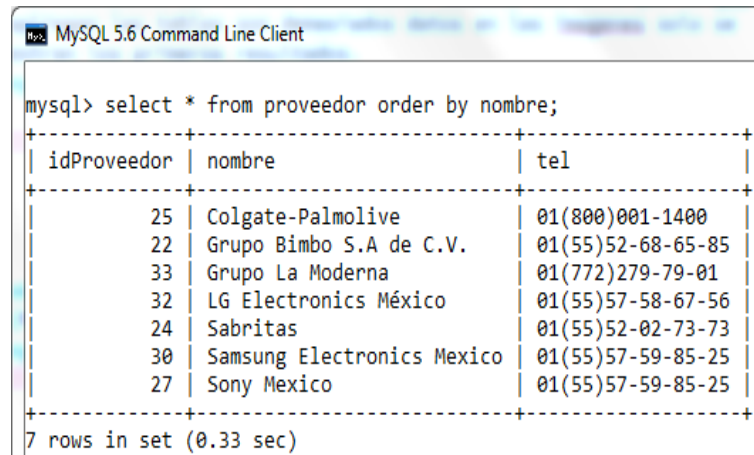
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)
Query OK, 0 rows affected (0.00 sec)

mysql> show tables;
+-----+
| Tables_in_sams |
+-----+
| club            |
| estado          |
| gerente         |
| producto        |
| proveedor       |
| proveedoresams  |
| servicio        |
| servicioclub    |
| socio           |
| socioclub       |
+-----+
10 rows in set (0.00 sec)

mysql>
```

Figura 3: Resultado de ejecutar el source.

Ahora comenzamos con el uso del comando select, debido a que lo que despliegan las tablas son demasiados datos en las imágenes solo se muestran los primeros resultados.

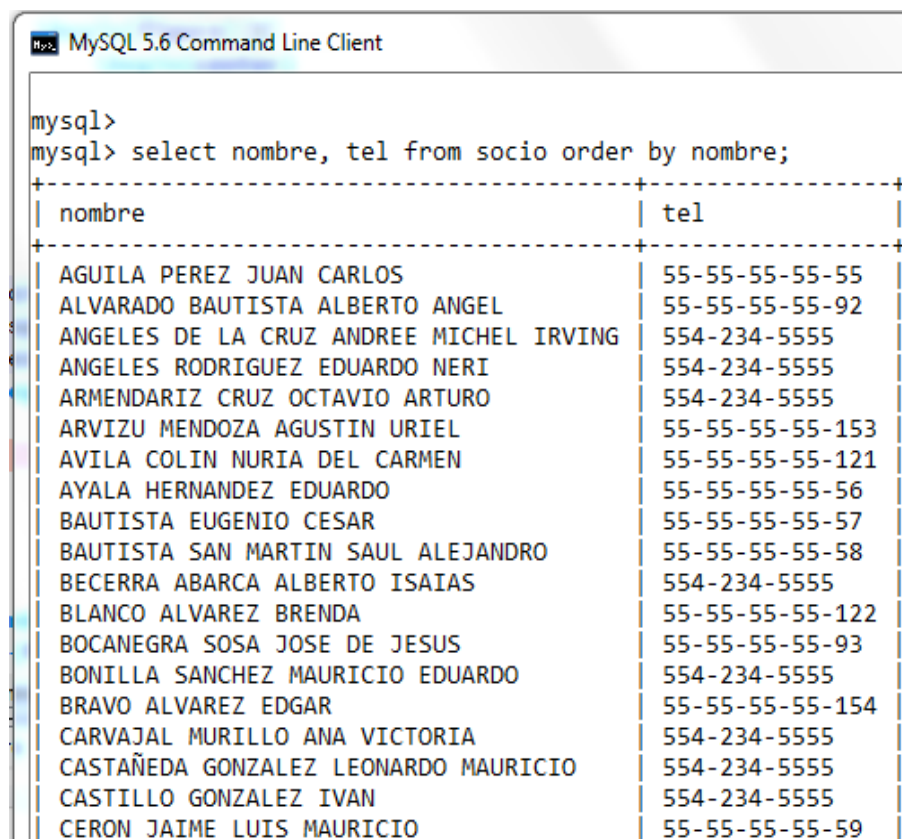


```
mysql> select * from proveedor order by nombre;
```

idProveedor	nombre	tel
25	Colgate-Palmolive	01(800)001-1400
22	Grupo Bimbo S.A de C.V.	01(55)52-68-65-85
33	Grupo La Moderna	01(772)279-79-01
32	LG Electronics México	01(55)57-58-67-56
24	Sabritas	01(55)52-02-73-73
30	Samsung Electronics Mexico	01(55)57-59-85-25
27	Sony Mexico	01(55)57-59-85-25

7 rows in set (0.33 sec)

Figura 4: Se muestran todos los campos y se ordena por nombre el contenido de proveedor.



```
mysql> select nombre, tel from socio order by nombre;
```

nombre	tel
AGUILA PEREZ JUAN CARLOS	55-55-55-55-55
ALVARADO BAUTISTA ALBERTO ANGEL	55-55-55-55-92
ANGELES DE LA CRUZ ANDREE MICHEL IRVING	554-234-5555
ANGELES RODRIGUEZ EDUARDO NERI	554-234-5555
ARMENDARIZ CRUZ OCTAVIO ARTURO	554-234-5555
ARVIZU MENDOZA AGUSTIN URIEL	55-55-55-55-153
AVILA COLIN NURIA DEL CARMEN	55-55-55-55-121
AYALA HERNANDEZ EDUARDO	55-55-55-55-56
BAUTISTA EUGENIO CESAR	55-55-55-55-57
BAUTISTA SAN MARTIN SAUL ALEJANDRO	55-55-55-55-58
BECERRA ABARCA ALBERTO ISAIAS	554-234-5555
BLANCO ALVAREZ BRENDA	55-55-55-55-122
BOCANEGRA SOSA JOSE DE JESUS	55-55-55-55-93
BONILLA SANCHEZ MAURICIO EDUARDO	554-234-5555
BRAVO ALVAREZ EDGAR	55-55-55-55-154
CARVAJAL MURILLO ANA VICTORIA	554-234-5555
CASTAÑEDA GONZALEZ LEONARDO MAURICIO	554-234-5555
CASTILLO GONZALEZ IVAN	554-234-5555
CERON JAIME LUIS MAURICIO	55-55-55-55-59

Figura 5: Muestra algunos campos y se ordenan por nombre.

```

mysql> select nombre, preciounitario from producto where preciounitario>=1000 and preciounitario<=6000;
+-----+-----+
| nombre          | preciounitario |
+-----+-----+
| Sony Bravia     | 4699           |
| Sony PSP        | 3799           |
| Horno Microondas AGE1072 | 1349           |
| Horno Microondas AGE107W | 1999           |
| Refrigerador 4 puertas | 4490           |
| LG Celular GSM ME970 | 2322.42        |
| LG Celular GSM ME230 | 4600           |
| LG Celular GSM MG160 | 5129           |
+-----+-----+
8 rows in set (0.04 sec)

mysql> select nombre, preciounitario from producto where preciounitario between 1000 and 6000 order by preciounitario;
+-----+-----+
| nombre          | preciounitario |
+-----+-----+
| Horno Microondas AGE1072 | 1349           |
| Horno Microondas AGE107W | 1999           |
| LG Celular GSM ME970 | 2322.42        |
| Sony PSP        | 3799           |
| Refrigerador 4 puertas | 4490           |
| LG Celular GSM ME230 | 4600           |
| Sony Bravia     | 4699           |
| LG Celular GSM MG160 | 5129           |
+-----+-----+
8 rows in set (0.02 sec)

```

Figura 6: Desplegamos los campos con un precio unitario entre 1000 y 6000 de dos formas distintas.

```

mysql> select nombre, direccion from club where nombre='tepeyac' or nombre='pachuca' or nombre='toreo';
+-----+-----+
| nombre | direccion |
+-----+-----+
| Tepeyac | Av. Ferrocarril Hidalgo No. 431, Col. Guadalupe Tepeyac, C.P.07840 |
| Toreo   | Blvd. Manuel uvila Camacho No. 467, Col. Periodistas, C.P.11220 |
| Pachuca | Av. Rjo de las Avenidas No. 701, Fracc. Las Terrazas, C.P.42098, Pachuca. |
+-----+-----+
3 rows in set (0.06 sec)

```

Figura 7: Información de los club con los nombres de Tepeyac, Toreo o Pachuca.

```

MySQL 5.6 Command Line Client
mysql> select nombre as "un club chevere", direccion from club where nombre='tepeyac' or nombre='pachuca' or nombre='toreo';
+-----+-----+
| un club chevere | direccion |
+-----+-----+
| Tepeyac         | Av. Ferrocarril Hidalgo No. 431, Col. Guadalupe Tepeyac, C.P.07840 |
| Toreo          | Blvd. Manuel Mvila Camacho No. 467, Col. Periodistas, C.P.11220 |
| Pachuca        | Av. Rjo de las Avenidas No. 701, Fracc. Las Terrazas, C.P.42098, Pachuca. |
+-----+-----+
3 rows in set (0.02 sec)

```

Figura 8: Lo mismo que el anterior pero renombrando la columna nombre como 'un club chevere'.

En este punto se empezó a usar la palabra reservada 'like' que es una wildcard (comodín) para las consultas un poco más complejas.

```

MySQL 5.6 Command Line Client
mysql> select nombre from gerente where nombre like "% Hernandez %";
+-----+
| nombre |
+-----+
| BALLESTEROS HERNANDEZ YOSELIN ANDREA |
| GALVAN HERNANDEZ MARIA ISABEL |
| GRACIAN HERNANDEZ SERGIO DEMIAN |
| HERRERA HERNANDEZ JENIFFER |
| LARA HERNANDEZ ADRIANA |
| VERTIZ HERNANDEZ CARLOS |
| LOPEZ HERNANDEZ YAZMIN SARAHI |
| GUEVARA HERNANDEZ LUIS AXELL |
| PERALTA HERNANDEZ IRVING DAVID |
| DIAZ HERNANDEZ FRANCISCO JAVIER |
| GOMEZ HERNANDEZ OMAR JESUS |
| GONZALEZ HERNANDEZ SAMUEL |
+-----+
12 rows in set (0.05 sec)

```

Figura 9: Mostramos a todos aquellos que se apelliden Hernandez.

```
MySQL 5.6 Command Line Client
mysql> select nombre, tel from proveedor where nombre like "LG%" or nombre like "sabritas";
+-----+-----+
| nombre           | tel           |
+-----+-----+
| Sabritas         | 01(55)52-02-73-73 |
| LG Electronics México | 01(55)57-58-67-56 |
+-----+-----+
2 rows in set (0.00 sec)
```

Figura 10: Se muestran todos los teléfonos de los proveedores de LG y sabritas.

```
MySQL 5.6 Command Line Client
mysql> select nombre, tel from socio where nombre like "Gonzal%" order by nombre;
+-----+-----+
| nombre           | tel           |
+-----+-----+
| GONZALEZ ANGELES FABIAN GERARDO | 55-55-55-55-101 |
| GONZALEZ DIAZ MARTIN AUGUSTO    | 554-234-5555    |
| GONZALEZ ESTRADA HILDA          | 55-55-55-55-68  |
| GONZALEZ GALLEGOS ROSA ISELA    | 55-55-55-55-69  |
| GONZALEZ HERNANDEZ URIEL        | 55-55-55-55-120 |
| GONZALEZ SANCHEZ MARS YUREN     | 554-234-5555    |
| GONZALEZ ZAVALA HUGO CESAR      | 554-234-5555    |
+-----+-----+
7 rows in set (0.00 sec)
```

Figura 11: Muestra a aquellos socios con apellido paterno Gonzáles.

```
MySQL 5.6 Command Line Client
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 (0.05 sec)
```

Figura 12: Muestra el nombre de todos los servicios disponibles.

```
MySQL 5.6 Command Line Client
mysql> select count(*) from socio where nombre like "Martinez%" or nombre like "% Martinez%";
+-----+
| count(*) |
+-----+
|          9 |
+-----+
1 row in set (0.03 sec)
```

Figura 13: Cuenta cuantos socios se apellidan Martínez.


```

MySQL 5.6 Command Line Client
mysql> select nombre from socio where nombre like "% JUAN%";
+-----+
| nombre |
+-----+
| VILLA RUSSELL JUAN PABLO |
| MALDONADO HERNANDEZ JUAN RAFAEL |
| AGUILA PEREZ JUAN CARLOS |
| LINARES CERON JUAN ANTONIO |
| REYES HERNANDEZ JUAN ALFREDO |
+-----+
5 rows in set (0.00 sec)

```

Figura 14: Muestra el nombre de los socios llamados Juan.

Después de hacer algunas consultas comenzamos a modificar las relaciones comenzando renombrar la tabla gerente.

```

MySQL 5.6 Command Line Client
mysql> desc gerente;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idGerente | int(10) unsigned | NO | PRI | NULL | |
| nombre | varchar(50) | NO | | NULL | |
| idClub | int(10) unsigned | YES | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.10 sec)

mysql> alter table gerente rename as empleado;
Query OK, 0 rows affected (0.34 sec)

mysql> desc empleado;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idGerente | int(10) unsigned | NO | PRI | NULL | |
| nombre | varchar(50) | NO | | NULL | |
| idClub | int(10) unsigned | YES | MUL | NULL | |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.02 sec)

```

Figura 15: Cambio nombre.

Se hicieron cambios en la tabla estado.

```

mysql> desc estado;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idEdo | int(11)        | NO   | PRI | NULL    |       |
| nombre | varchar(50)    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.03 sec)

mysql> alter table estado change column nombre estado varchar(50);
Query OK, 0 rows affected (0.18 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc estado
-> ;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idEdo | int(11)        | NO   | PRI | NULL    |       |
| estado | varchar(50)    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.01 sec)

```

Figura 16: Cambiamos el nombre del campo nombre a estado.

De la misma forma se modifico la relación socio.

```

mysql> desc socio;
+-----+-----+-----+-----+-----+-----+
| Field      | Type              | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idSocio    | int(10) unsigned  | NO   | PRI | NULL    |       |
| nombre     | varchar(50)       | NO   |     | NULL    |       |
| direccion  | varchar(100)      | NO   |     | NULL    |       |
| tel        | varchar(15)       | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

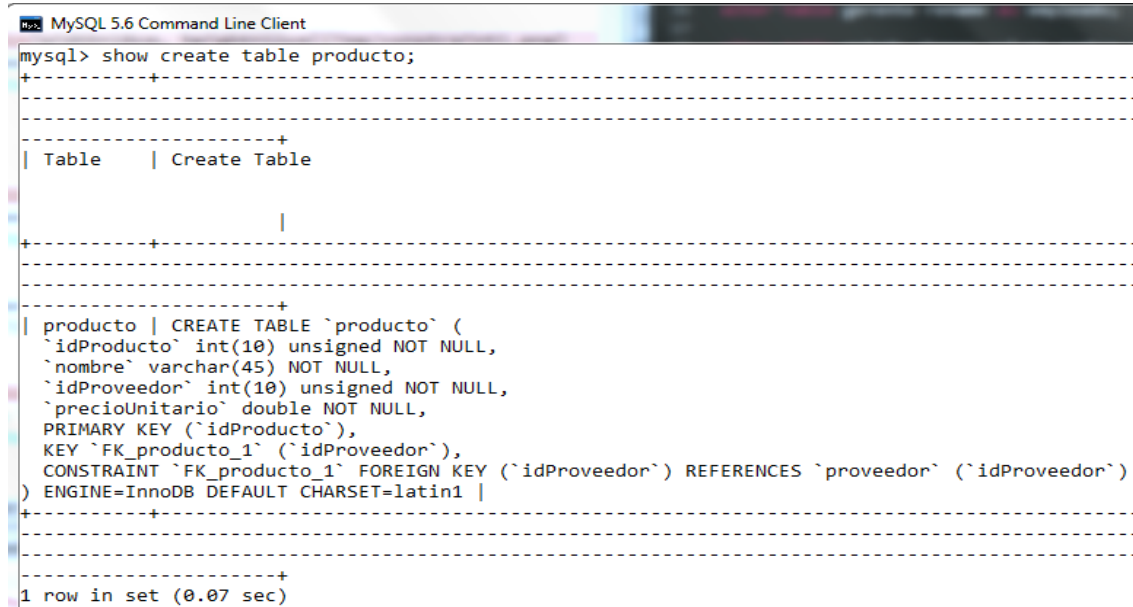
mysql> alter table socio drop column tel;
Query OK, 0 rows affected (0.75 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc socio;
+-----+-----+-----+-----+-----+-----+
| Field      | Type              | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idSocio    | int(10) unsigned  | NO   | PRI | NULL    |       |
| nombre     | varchar(50)       | NO   |     | NULL    |       |
| direccion  | varchar(100)      | NO   |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

```

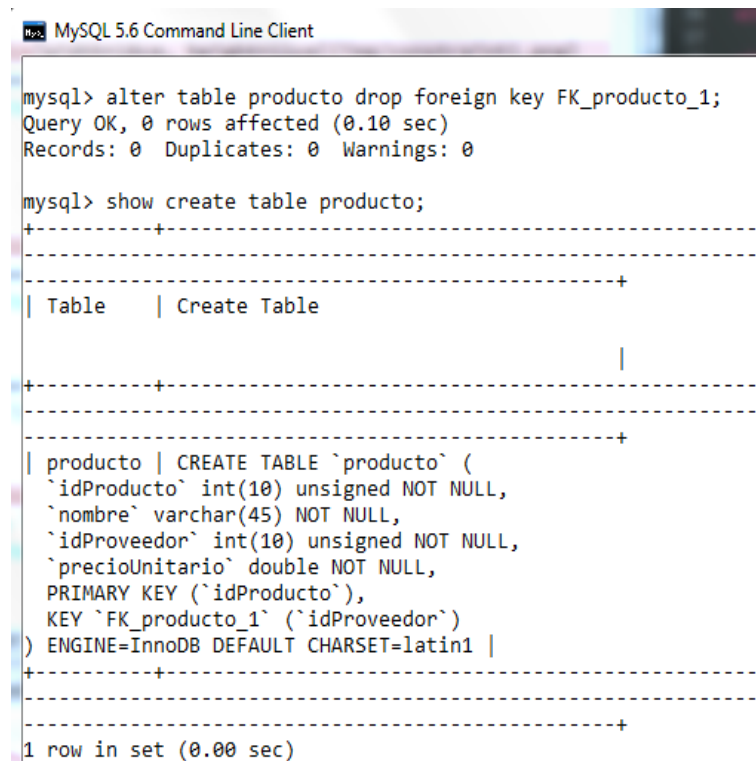
Figura 17: Se elimino la columna tel y su contenido de socio.

A continuación se procedió a eliminar la llave foranea de la tabla producto.



```
mysql> show create table producto;
+-----+
| Table | Create Table
+-----+
| producto | CREATE TABLE `producto` (
  `idProducto` int(10) unsigned NOT NULL,
  `nombre` varchar(45) NOT NULL,
  `idProveedor` int(10) unsigned NOT NULL,
  `precioUnitario` double NOT NULL,
  PRIMARY KEY (`idProducto`),
  KEY `FK_producto_1` (`idProveedor`),
  CONSTRAINT `FK_producto_1` FOREIGN KEY (`idProveedor`) REFERENCES `proveedor` (`idProveedor`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+
```

Figura 18: Buscamos el constraint.



```
mysql> alter table producto drop foreign key FK_producto_1;
Query OK, 0 rows affected (0.10 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> show create table producto;
+-----+
| Table | Create Table
+-----+
| producto | CREATE TABLE `producto` (
  `idProducto` int(10) unsigned NOT NULL,
  `nombre` varchar(45) NOT NULL,
  `idProveedor` int(10) unsigned NOT NULL,
  `precioUnitario` double NOT NULL,
  PRIMARY KEY (`idProducto`),
  KEY `FK_producto_1` (`idProveedor`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
+-----+
```

Figura 19: Borramos la relación.

Creamos una llave primaria compuesta para empleado para hacer esto primero se elimina la llave primaria anterior.

```
MySQL 5.6 Command Line Client
mysql> alter table empleado drop primary key;
Query OK, 200 rows affected (1.68 sec)
Records: 200 Duplicates: 0 Warnings: 0

mysql> alter table empleado add primary key(idGerente, nombre);
Query OK, 0 rows affected (1.05 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc empleado
-> ;
+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idGerente  | int(10) unsigned    | NO   | PRI | NULL    |       |
| nombre     | varchar(50)         | NO   | PRI | NULL    |       |
| idClub     | int(10) unsigned    | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)
```

Figura 20: Nueva llave primaria.

Luego se agrego el campo el campo salario de tipo double a la tabla empleado.

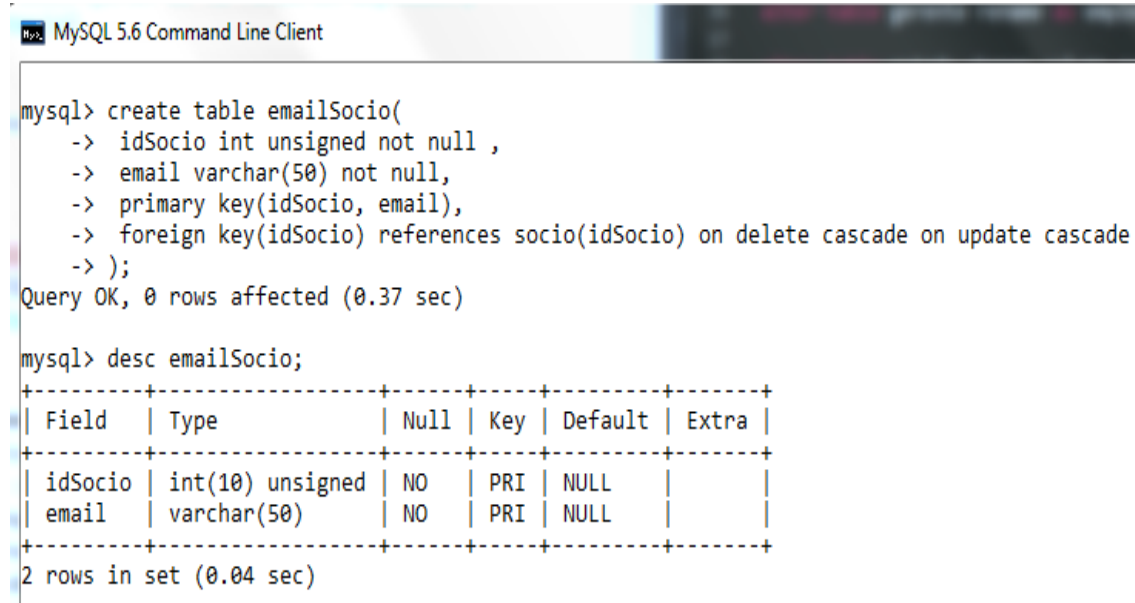
```
MySQL 5.6 Command Line Client
mysql>
mysql>
mysql>
mysql> desc empleado;
+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idGerente  | int(10) unsigned    | NO   | PRI | NULL    |       |
| nombre     | varchar(50)         | NO   | PRI | NULL    |       |
| idClub     | int(10) unsigned    | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.01 sec)

mysql> alter table empleado add column salario double;
Query OK, 0 rows affected (0.81 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc empleado;
+-----+-----+-----+-----+-----+-----+
| Field      | Type                | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idGerente  | int(10) unsigned    | NO   | PRI | NULL    |       |
| nombre     | varchar(50)         | NO   | PRI | NULL    |       |
| idClub     | int(10) unsigned    | YES  | MUL | NULL    |       |
| salario    | double              | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

Figura 21: Nueva descripción de la relación empleado.

Finalmente agregamos una nueva relación para poder manejar múltiples correos del socio.



```
mysql> create table emailSocio(
-> idSocio int unsigned not null ,
-> email varchar(50) not null,
-> primary key(idSocio, email),
-> foreign key(idSocio) references socio(idSocio) on delete cascade on update cascade
-> );
Query OK, 0 rows affected (0.37 sec)

mysql> desc emailSocio;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idSocio | int(10) unsigned | NO   | PRI | NULL    |       |
| email   | varchar(50)      | NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.04 sec)
```

Figura 22: Creación de la nueva tabla emailSocio.

3. Conclusiones

En esta practica se observo de una manera muy simple el funcionamiento de algunos comandos y diversas opciones que tienen para desplegar el contenido de las tablas lo cual en proximos proyectos nos brindara las herramientas necesarias para buscar información y darle más funcionamiento a las aplicaciones que realicemos ofreciendo un producto más completo a nuestros usuarios.

Referencias

- [1] H. F. K. Abraham Silberschatz, *Fundamentos de Diseño de Bases de Datos*. McGraw-Hill, 2007.