

# Reporte: Práctica 1

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	8
Referencias	9

## 1. Marco Teórico

Un **sistema gestor de bases de datos (SGBD)** consiste en un conjunto de datos interrelacionados y en un conjunto de programas para tener acceso a esos datos. Los datos describen una empresa concreta.

El objetivo principal de un **SGBD** es proporcionar un entorno que sea tanto conveniente como eficiente para las personas que lo usan para la recuperación y almacenamiento de información.

Uno de los principales propósitos de los sistemas de bases de datos es ofrecer a los usuarios una visión abstracta de los datos. Es decir, el sistema oculta ciertos detalles de la manera en que los datos se almacenan y mantienen.[1]

Un concepto importante a mencionar son las **tablas** a las cuales se les asigna un nombre exclusivo. De manera burda una tabla es un conjunto de entidades y cada fila es una entidad es por esto que existe una correspondencia entre el término tabla y el concepto matemático de relación.[1]

Además, es importante mencionar otros conceptos a tratar en esta práctica:

- **Atributos.** Un atributo hace referencia a las columnas de las tablas.
- **Clave primaria.** Es un campo, o grupo de campos que identifica en forma única un registro. La llave primaria se utiliza para distinguir un registro con el fin de que se pueda tener acceso a ellos, organizarlos y manipularlos.
- **Clave foránea.** Es una limitación referencial entre dos tablas la cual identifica una columna o grupo de columnas en una tabla que se refiere a una columna o grupo de columnas en otra tabla.

## 2. Desarrollo

En esta practica se comenzó creando la base de datos y seleccionándola para poder usarla. Para realizar esto se usan los siguientes comandos.

---

```
create database trabajos_terminales;  
use trabajos_terminales;
```

---

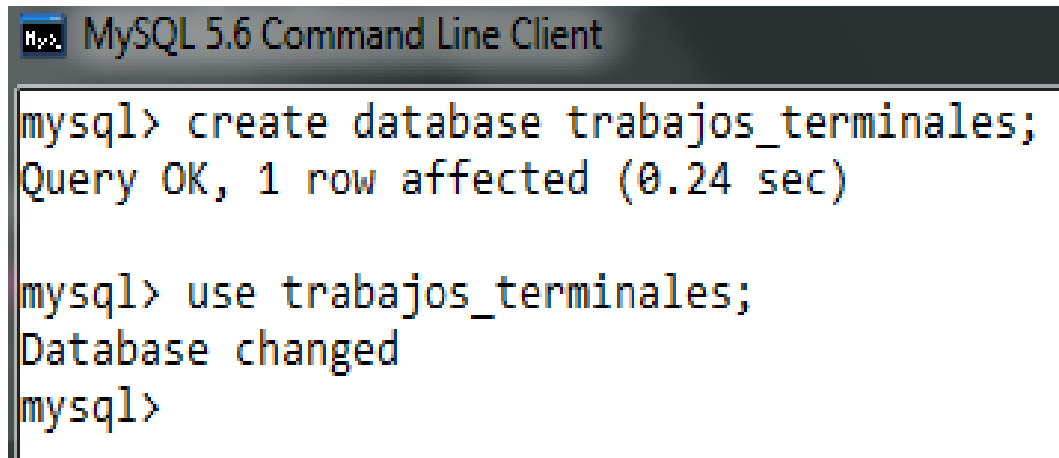


Figura 1: Creación y uso de la base.

Se crearon las relaciones con las que se planeaba trabajar, el orden en el que se crean es importante para evitar errores debido a la clave foránea que se usa en dirige y profesor.

---

```
create table tt(  
    nott int not null primary key,  
    titulo varchar(100)  
);  
  
create table depto(  
    idDepto int not null primary key,  
    nombre varchar(50)  
);  
  
create table profesor(  
    idProf int not null primary key,  
    nombre varchar(20),  
    ap varchar(30),  
    am varchar(30),  
    academia varchar(50),  
    salario double,  
    idDepto int,  
    foreign key(idDepto) references depto(idDepto) on delete cascade on  
        update cascade
```

```
);

create table presentacion(
    idPresentacion int not null primary key,
    fecha date,
    califRevisor float,
    califSinodales float,
    tipo varchar(30)
);

create table dirige(
    idProf int not null,
    nott int not null,
    primary key(idProf, nott),
    foreign key(idProf) references profesor(idProf) on delete cascade on
        update cascade,
    foreign key(nott) references tt(nott) on delete cascade on update cascade
);
```

---

```

MySQL 5.6 Command Line Client
mysql> desc tt;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| nott  | int(11)       | NO   | PRI | NULL    |       |
| titulo | varchar(100)  | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.16 sec)

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

mysql> desc profesor;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idProf | int(11)       | NO   | PRI | NULL    |       |
| nombre | varchar(20)   | YES  |     | NULL    |       |
| ap     | varchar(30)   | YES  |     | NULL    |       |
| am     | varchar(30)   | YES  |     | NULL    |       |
| academia | varchar(50) | YES  |     | NULL    |       |
| salario | double       | YES  |     | NULL    |       |
| idDepto | int(11)      | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.01 sec)

mysql> desc presentacion;
+-----+-----+-----+-----+-----+-----+
| Field | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idPresentacion | int(11) | NO   | PRI | NULL    |       |
| fecha          | date    | YES  |     | NULL    |       |
| califRevisor   | float   | YES  |     | NULL    |       |
| califSinodales | float   | YES  |     | NULL    |       |
| tipo           | varchar(30) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.01 sec)

```

Figura 2: Descripción de las tablas después de crearlas.

En este punto se empezó a modificar la descripción de las tablas usando el comando alter.

---

```

alter table profesor rename as catedratico; -- Esto cambia el nombre de la
      tabla profesor a catedratico
alter table presentacion add column dictamen varchar(30); --Agregamos un
      nueva columna a esta tabla
alter table depto change column nombre depto varchar(50) not null; -- Se
      cambia el nombre de la columna
alter table catedratico add column tel int; -- Agrega una nueva columna
alter table catedratico change column tel tel varchar(30); -- Cambia el tipo
      de dato

```

---

```

MySQL 5.6 Command Line Client
mysql> alter table profesor rename as catedratico;
Query OK, 0 rows affected (0.25 sec)

mysql> alter table presentacion add column dictamen varchar(30);
Query OK, 0 rows affected (0.73 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table depto change column nombre depto varchar(50) not null;
Query OK, 0 rows affected (0.52 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table catedratico add column tel int;
Query OK, 0 rows affected (0.68 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table catedratico change column tel tel varchar(30);
Query OK, 0 rows affected (0.93 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc catedratico;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idProf | int(11) | NO | PRI | NULL | |
| nombre | varchar(20) | YES | | NULL | |
| ap | varchar(30) | YES | | NULL | |
| am | varchar(30) | YES | | NULL | |
| academia | varchar(50) | YES | | NULL | |
| salario | double | YES | | NULL | |
| idDepto | int(11) | YES | MUL | NULL | |
| tel | varchar(30) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.01 sec)

```

Figura 3: Cambios en la base.

```

MySQL 5.6 Command Line Client
8 rows in set (0.01 sec)

mysql> desc presentacion;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idPresentacion | int(11) | NO | PRI | NULL | |
| fecha | date | YES | | NULL | |
| califRevisor | float | YES | | NULL | |
| califSinodales | float | YES | | NULL | |
| tipo | varchar(30) | YES | | NULL | |
| dictamen | varchar(30) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)

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

```

Figura 4: Más cambios realizados.

Ahora se procedió a agregar una llave foránea a la tabla presentación, para esto primero se creo la columna que seria la futura FK y se especifica con que tabla tiene relación.

---

```

alter table presentacion add column nott int; -- Campo necesario para la
foreign key
alter table presentacion add FOREIGN KEY(nott) references tt(nott)

```

on delete cascade on update cascade; -- Creacion de la clave foranea

---

```
MySQL 5.6 Command Line Client
mysql> alter table presentacion add column nott int;
Query OK, 0 rows affected (0.70 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table presentacion add FOREIGN KEY(nott) references tt(nott)
-> on delete cascade on update cascade;
Query OK, 0 rows affected (0.95 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc presentacion
-> ;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idPresentacion | int(11)       | NO   | PRI | NULL    |       |
| fecha          | date         | YES  |     | NULL    |       |
| califRevisor   | float        | YES  |     | NULL    |       |
| califSinodales | float        | YES  |     | NULL    |       |
| tipo           | varchar(30)   | YES  |     | NULL    |       |
| dictamen       | varchar(30)   | YES  |     | NULL    |       |
| nott           | int(11)       | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.04 sec)
```

Figura 5: Creación de la llave foránea

Después, se modifico la llave primaria haciéndola compuesta para lograr esto se borra la anterior PK y se agrega la nueva.

```
alter table presentacion drop primary key; --Borramos nuestra anterior PK
alter table presentacion add primary key(idPresentacion, fecha); -- Nueva
primary key compuesta
```

---

```
MySQL 5.6 Command Line Client
mysql> alter table presentacion drop primary key;
Query OK, 0 rows affected (0.93 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> alter table presentacion add primary key(idPresentacion, fecha);
Query OK, 0 rows affected (0.65 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> desc presentacion;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default          | Extra |
+-----+-----+-----+-----+-----+-----+
| idPresentacion | int(11)       | NO   | PRI | NULL             |       |
| fecha          | date         | NO   | PRI | 0000-00-00       |       |
| califRevisor   | float        | YES  |     | NULL             |       |
| califSinodales | float        | YES  |     | NULL             |       |
| tipo           | varchar(30)   | YES  |     | NULL             |       |
| dictamen       | varchar(30)   | YES  |     | NULL             |       |
| nott           | int(11)       | YES  | MUL | NULL             |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.04 sec)
```

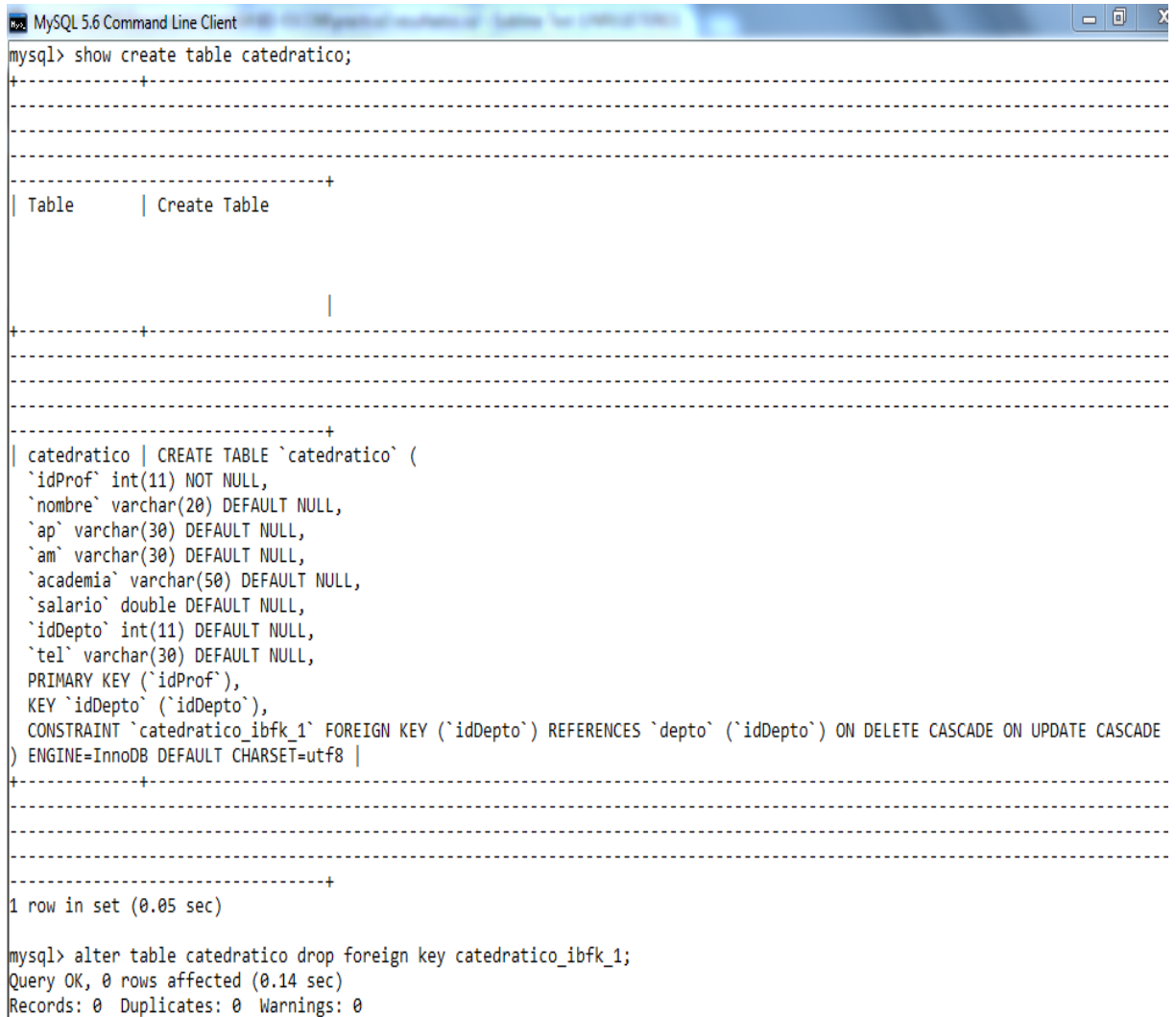
Figura 6: Creación de la llave primaria compuesta.

Finalmente se elimino la clave foránea de una tabla, esto es mas complejo que la llave primaria ya que se debe observar la descripción de la relación para obtener el constraint que tiene asignado dicha clave.

---

```
show create table catedratico; -- Lo usamos para observar el id de la llave
foranea
alter table catedratico drop foreign key catedratico_ibfk_1; -- Ahora si
podemos borrar esta llave foranea
```

---



```
mysql> show create table catedratico;
+-----+-----+
| Table | Create Table |
+-----+-----+
| catedratico | CREATE TABLE `catedratico` (
  `idProf` int(11) NOT NULL,
  `nombre` varchar(20) DEFAULT NULL,
  `ap` varchar(30) DEFAULT NULL,
  `am` varchar(30) DEFAULT NULL,
  `academia` varchar(50) DEFAULT NULL,
  `salario` double DEFAULT NULL,
  `idDepto` int(11) DEFAULT NULL,
  `tel` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`idProf`),
  KEY `idDepto` (`idDepto`),
  CONSTRAINT `catedratico_ibfk_1` FOREIGN KEY (`idDepto`) REFERENCES `depto` (`idDepto`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.05 sec)

mysql> alter table catedratico drop foreign key catedratico_ibfk_1;
Query OK, 0 rows affected (0.14 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

Figura 7: Se identifico y borro la clave foránea



```
MySQL 5.6 Command Line Client
Records: 0  Duplicates: 0  Warnings: 0

mysql> show create table catedratico;
+-----+-----+
| Table          | Create Table
+-----+-----+
| catedratico     | CREATE TABLE `catedratico` (
  `idProf` int(11) NOT NULL,
  `nombre` varchar(20) DEFAULT NULL,
  `ap` varchar(30) DEFAULT NULL,
  `am` varchar(30) DEFAULT NULL,
  `academia` varchar(50) DEFAULT NULL,
  `salario` double DEFAULT NULL,
  `idDepto` int(11) DEFAULT NULL,
  `tel` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`idProf`),
  KEY `idDepto` (`idDepto`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.00 sec)
```

Figura 8: Este fue el resultado de borrarlo.

Para finalizar se hizo un respaldo, es importante estar en la ruta que se muestra en la imagen para poder realizar esto

```
C:\Program Files\MySQL\MySQL Server 5.6\bin>mysqldump -u root -p trabajos_terminales>C:\Users\USER\Desktop\bases\respaldo1.sql
Enter password: *****

C:\Program Files\MySQL\MySQL Server 5.6\bin>
```

Figura 9: Se logro hacer el respaldo sin errores.

### 3. Conclusiones

Esta primera práctica me permitió empezar a conocer el como se trabaja con una base de datos a un nivel básico pero fundamental también es importante destacar que para dominar esta técnica es necesario seguir practicando y repasando las practicas que se hagan de aquí en adelante.

## Referencias

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