

Práctica 3

Alejandro Adolfo Perez Raya

Base de Datos

29 de agosto de 2013

Índice

1. Marco Teórico	3
1.1. Comandos Importantes:	3
2. Instrucciones	4
3. Sentencias SQL y Screenshots	5
4. Conclusiones	10

1. Marco Teórico

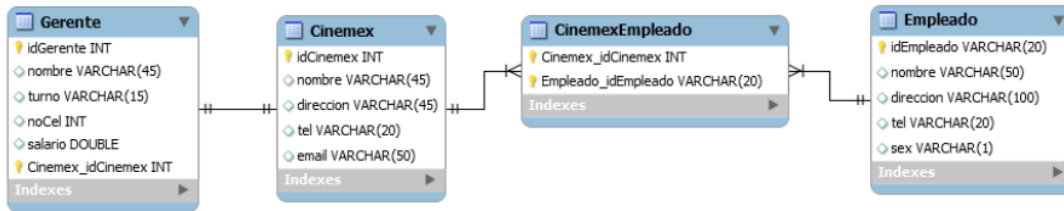
1.1. Comandos Importantes:

- *ALTER TABLE nombre_Relación OPCION COLUMN nombre_Columna tipoDato(tamaño) ;*
Sintaxis general para formular una expresión en SQL utilizando DDL para modificar un esquema de Base de datos, en especial RELACIONES.
OPCION: puede sustituirse por las principales acciones como: ADD, DROP, MODIFY, CHANGE, RENAME.
- *DROP TABLE nombre_Tabla;*
Comando para eliminar una relación.
- *ALTER TABLE nombre_Relación RENAME AS nuevoNombre;*
Comando para renombrar una relación.
- *ALTER TABLE nombre_Relación MODIFY COLUMN nombre_Columna tipoDato(tamaño);*
Comando para modificar un Tipo de Dato o bien para cambiar el tamaño de un tipo de datos.
- *ALTER TABLE nombre_Relación CHANGE COLUMN nombre_Columna nuevoNombre tipoDato(tamaño);*
Comando para renombrar una columna específica.
- *ALTER TABLE nombre_Relación DROP FOREIGN KEY identificador_FK;*
Comando para eliminar una FK.
- *ALTER TABLE nombre_Relación DROP PRIMARY KEY campoidentificador;*
Comando para eliminar una PK.
- *ALTER TABLE nombre_Relación ADD FOREIGN KEY (campoFK) REFERENCES NOMBRE_TABLA (campoIdentificador) ON DELETE CASCADE ON UPDATE CASCADE;*
Comando para agregar una FK.
- *ALTER TABLE nombre_Relación ADD PRIMARY KEY (campoid1, campo2, ... , campon);*
Comando para agregar una PK.

2. Instrucciones

Resuelve las siguientes sentencias.

1. Crea el siguiente esquema de Base de datos:



2. Agrega las columnas correspondientes para almacenar el salario y el correo electrónico de los empleados.
3. Cambia el tipo de dato del atributo noCel en gerente para que pueda almacenarse en el siguiente formato (044)-(55-55-55-55-55).
4. Renombra la relación empleado, quedando como ASOCIADO.
5. Aumenta el tamaño del tipo de dato del campo dirección en la relación asociado.
6. Elimina la asociación existente en las relaciones Gerente y Cinemex.
7. Redefine la PK de la relación cinemex quedado de la siguiente manera PK(idCinemex,nombre).
8. Crea la siguiente relación Cartelera(idCartelera, nombre, fechaincio, fechafin,clasificación) y asociarla con la relación cinemex.
9. Establece la asociación entre la relación cinemex y gerente.
10. Genera un respaldo del esquema final.

3. Sentencias SQL y Screenshots

A continuación se pondrán los comandos utilizados para realizar cada una de las instrucciones antes mencionadas.

1.

```
create table cinemex(  
  idcinemex int not null primary key,  
  nombre varchar(45),  
  direccion varchar(45),  
  tel varchar(20),  
  email varchar(50));
```

```
mysql> desc cinemex;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| idcinemex  | int(11)       | NO   | PRI | NULL    |       |  
| nombre     | varchar(45)   | YES  |     | NULL    |       |  
| direccion  | varchar(45)   | YES  |     | NULL    |       |  
| tel        | varchar(20)   | YES  |     | NULL    |       |  
| email      | varchar(50)   | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```

```
create table empleado(  
  idempleado varchar(20) not null primary key,  
  nombre varchar(50),  
  direccion varchar(100),  
  tel varchar(20),  
  sex varchar(1));
```

```
mysql> desc empleado;  
+-----+-----+-----+-----+-----+-----+  
| Field      | Type          | Null | Key | Default | Extra |  
+-----+-----+-----+-----+-----+-----+  
| idempleado | varchar(20)   | NO   | PRI | NULL    |       |  
| nombre     | varchar(50)   | YES  |     | NULL    |       |  
| direccion  | varchar(100)  | YES  |     | NULL    |       |  
| tel        | varchar(20)   | YES  |     | NULL    |       |  
| sex        | varchar(1)    | YES  |     | NULL    |       |  
+-----+-----+-----+-----+-----+-----+  
5 rows in set (0.00 sec)
```

```

create table gerente(
idgerente int not null primary key,
nombre varchar(45),
turno varchar(15),
cel int,
salario double,
cinemex_idcinemex int,
foreign key(cinemex_idcinemex) references cinemex(idcinemex)
on delete cascade on update cascade);

```

```

mysql> desc gerente;
+-----+-----+-----+-----+-----+-----+
| Field           | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idgerente       | int(11)       | NO   | PRI | NULL    |       |
| nombre          | varchar(45)   | YES  |     | NULL    |       |
| turno           | varchar(15)   | YES  |     | NULL    |       |
| cel             | int(11)       | YES  |     | NULL    |       |
| salario         | double        | YES  |     | NULL    |       |
| cinemex_idcinemex | int(11)       | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)

```

```

create table cinemexEmpleado(
cinemex_idcinemex int not null,
empleado_idEmpleado varchar(20) not null,
primary key(cinemex_idcinemex,empleado_idEmpleado),
foreign key(cinemex_idcinemex) references cinemex(idcinemex)
on delete cascade on update cascade,
foreign key(empleado_idEmpleado) references empleado(idempleado)
on delete cascade on update cascade);

```

```

mysql> desc cinemexEmpleado;
+-----+-----+-----+-----+-----+-----+
| Field           | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| cinemex_idcinemex | int(11)       | NO   | PRI | NULL    |       |
| empleado_idEmpleado | varchar(20)   | NO   | PRI | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

```

2.

```
ALTER TABLE empleado ADD COLUMN salario double;
```

```
ALTER TABLE empleado ADD COLUMN email varchar(50);
```

```
mysql> desc empleado;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idempleado | varchar(20)    | NO   | PRI | NULL    |       |
| nombre     | varchar(50)    | YES  |     | NULL    |       |
| direccion  | varchar(100)   | YES  |     | NULL    |       |
| tel        | varchar(20)    | YES  |     | NULL    |       |
| sex        | varchar(1)     | YES  |     | NULL    |       |
| salario    | double         | YES  |     | NULL    |       |
| email      | varchar(50)    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
```

3.

```
ALTER TABLE gerente MODIFY COLUMN cel varchar(20);
```

```
mysql> desc gerente;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idgerente  | int(11)        | NO   | PRI | NULL    |       |
| nombre     | varchar(45)    | YES  |     | NULL    |       |
| turno      | varchar(15)    | YES  |     | NULL    |       |
| cel        | varchar(20)    | YES  |     | NULL    |       |
| salario    | double         | YES  |     | NULL    |       |
| cinemex_idcinemex | int(11)      | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.01 sec)
```

4.

```
ALTER TABLE empleado RENAME AS asociado;
```

```
mysql> desc asociado;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idempleado | varchar(20)    | NO   | PRI | NULL    |       |
| nombre     | varchar(50)    | YES  |     | NULL    |       |
| direccion  | varchar(100)   | YES  |     | NULL    |       |
| tel        | varchar(20)    | YES  |     | NULL    |       |
| sex        | varchar(1)     | YES  |     | NULL    |       |
| salario    | double         | YES  |     | NULL    |       |
| email      | varchar(50)    | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

5.

```
ALTER TABLE asociado MODIFY COLUMN direccion VARCHAR(250);
```

```
mysql> desc asociado;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idempleado | varchar(20)    | NO   | PRI | NULL     |       |
| nombre     | varchar(50)    | YES  |     | NULL     |       |
| direccion  | varchar(250)   | YES  |     | NULL     |       |
| tel        | varchar(20)    | YES  |     | NULL     |       |
| sex        | varchar(1)     | YES  |     | NULL     |       |
| salario    | double         | YES  |     | NULL     |       |
| email      | varchar(50)    | YES  |     | NULL     |       |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

6.

```
ALTER TABLE gerente DROP FOREIGN KEY gerente_ibfk_1;
```

```
| gerente | CREATE TABLE `gerente` (
  `idgerente` int(11) NOT NULL,
  `nombre` varchar(45) DEFAULT NULL,
  `turno` varchar(15) DEFAULT NULL,
  `cel` varchar(20) DEFAULT NULL,
  `salario` double DEFAULT NULL,
  `cinemex_idcinemex` int(11) DEFAULT NULL,
  PRIMARY KEY (`idgerente`),
  KEY `cinemex_idcinemex` (`cinemex_idcinemex`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
```


7.
 - a) Eliminar la relación entre las llaves primarias con respecto a las relaciones donde aparece esa PK como FK


```
SHOW CREATE TABLE cinemexemplado; (obtener el identificador)
ALTER TABLE cinemexemplado DROP FOREIGN KEY cinemexemplado_ibfk_1;
```
 - b) Eliminar la llave primaria de Cinemex


```
ALTER TABLE cinemex DROP PRIMARY KEY;
```
 - c) Redefinir la PK con sus dos campos o atributos


```
ALTER TABLE cinemex ADD PRIMARY KEY(idcinemex,nombre);
```

```
mysql> desc cinemex;
```

Field	Type	Null	Key	Default	Extra
idcinemex	int(11)	NO	PRI	NULL	
nombre	varchar(45)	NO	PRI		
direccion	varchar(45)	YES		NULL	
tel	varchar(20)	YES		NULL	
email	varchar(50)	YES		NULL	

5 rows in set (0.00 sec)

8.


```
CREATE TABLE cartelera(
idcartelera int not null primary key,
nombre VARCHAR(300),
fechainicio DATE,
fechafin DATE,
clasificacion varchar(2),
idcinemex int,
cinemex VARCHAR(45),
FOREIGN KEY(idcinemex,cinemex) references cinemex(idcinemex,nombre)
on delete cascade on update cascade);
```

```
mysql> desc cartelera;
```

Field	Type	Null	Key	Default	Extra
idcartelera	int(11)	NO	PRI	NULL	
nombre	varchar(300)	YES		NULL	
fechainicio	date	YES		NULL	
fechafin	date	YES		NULL	
clasificacion	varchar(2)	YES		NULL	
idcinemex	int(11)	YES	MUL	NULL	
cinemex	varchar(45)	YES		NULL	

7 rows in set (0.00 sec)

9.

```
ALTER TABLE gerente ADD COLUMN CINEMEX VARCHAR(45);
```

```
ALTER TABLE gerente ADD FOREIGN KEY(cinemex_idcinemex,cinemex)
references cinemex(idcinemex,nombre) on delete cascade on update cascade;
```

```
mysql> desc gerente;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| idgerente | int(11) | NO | PRI | NULL | |
| nombre | varchar(45) | YES | | NULL | |
| turno | varchar(15) | YES | | NULL | |
| cel | varchar(20) | YES | | NULL | |
| salario | double | YES | | NULL | |
| cinemex_idcinemex | int(11) | YES | MUL | NULL | |
| CINEMEX | varchar(45) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

```
| gerente | CREATE TABLE `gerente` (
  `idgerente` int(11) NOT NULL,
  `nombre` varchar(45) DEFAULT NULL,
  `turno` varchar(15) DEFAULT NULL,
  `cel` varchar(20) DEFAULT NULL,
  `salario` double DEFAULT NULL,
  `cinemex_idcinemex` int(11) DEFAULT NULL,
  `CINEMEX` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`idgerente`),
  KEY `cinemex_idcinemex` (`cinemex_idcinemex`,`CINEMEX`),
  CONSTRAINT `gerente_ibfk_1` FOREIGN KEY (`cinemex_idcinemex`, `CINEMEX`) REFER
ENCES `cinemex` (`idcinemex`, `nombre`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1 |
```

4. Conclusiones

Para poder modificar algun campo o relación de un esquema de base de datos es necesario utilizar el comando ALTER TABLE con sus diferentes implementaciones u opciones, tambien es necesario verificar las condicones y requisitos que los comandos solicitan, por ejemplo para borrar una llave primaria primero hay que borrar cualquier relacion que exista de esta llave con otra tabla, así como conocer su identificador (CONSTRAINT) dentro de la otra tabla. Cuando creamos una llave foranea se crea un índice de multiples niveles esto se puede cuando realizamos un DESC nomTabla, en la columna de KEY nos aparece un mensaje que dice MUL, esto nos indica la existencia del índice.

Referencias

- [1] *Practica 3, Base de Datos, Modificando un esquema de base de datos.*
M. en C. Euler Hernández Contreras y M. en C. Idalia Maldonado Castillo