



ESCUELA SUPERIOR DE CÓMPUTO
INSTITUTO POLITÉCNICO NACIONAL



CABALLERO HUESCA CARLOS EDUARDO

Reporte de Practica



Practica 4

BASES DE DATOS
2CM5

Hernández Rubio Erika

I. INTRODUCCIÓN

1. Dado el siguiente diagrama relacional, generar el esquema de la base de datos correspondiente. Considerar la cardinalidad de las relaciones para crear las llaves foráneas correctamente. **Al crear la tabla CinemexEmpleado utilizar ON UPDATE CASCADE y ON DELETE CASCADE en su definición**
2. Agregar 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. Renombrar la tabla empleado, quedando como **asociado**.
5. Aumentar en 50 el tamaño del tipo de dato del campo dirección en la tabla asociado.
6. Redefinir la PK de la relación cinemex quedando de la siguiente manera PK(idCinemex,nombre).
7. Crea la siguiente tabla: **Cartelera**(idCartelera, nombre, fechainicio, fechafin, clasificación) y relaciónala con la tabla **Cinemex**.
8. Generar un respaldo del esquema final de su bases de datos

II. DESARROLLO

1. Dado el anterior diagrama relacional, generar el esquema de la base de datos correspondiente.

MySQL 5.7 Command Line Client

Database changed

```
mysql> CREATE TABLE Cinemex(  
-> idCinemex int NOT NULL PRIMARY KEY,  
-> nombre varchar(45) NOT NULL,  
-> direccion varchar(45) NOT NULL,  
-> tel varchar(20),  
-> email varchar(50)  
-> );
```

Query OK, 0 rows affected (0.23 sec)

mysql>

```
mysql> CREATE TABLE Gerente(  
-> idGerente INT NOT NULL PRIMARY KEY,  
-> nombre VARCHAR(45) NOT NULL,  
-> turno VARCHAR(15) NOT NULL,  
-> noCel INT,  
-> salario DOUBLE NOT NULL,  
-> Cinemex_idCinemex INT NOT NULL,  
-> FOREIGN KEY(Cinemex_idCinemex) REFERENCES Cinemex(idCinemex)  
-> );
```

Query OK, 0 rows affected (0.28 sec)

mysql>

```
mysql> CREATE TABLE Empleado(  
-> idEmpleado VARCHAR(20) NOT NULL PRIMARY KEY,  
-> nombre VARCHAR(50) NOT NULL,  
-> direccion VARCHAR(100) NOT NULL,  
-> tel VARCHAR(20),  
-> sex VARCHAR(1)  
-> );
```

Query OK, 0 rows affected (0.19 sec)

MySQL 5.7 Command Line Client

```
-> );  
Query OK, 0 rows affected (0.19 sec)
```

mysql>

```
mysql> 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  
-> );
```

Query OK, 0 rows affected (0.21 sec)

mysql> _

2. Agregar las columnas correspondientes para almacenar el salario y el correo electrónico de los empleados

```
MySQL 5.7 Command Line Client
-> FOREIGN KEY(Empleado_idEmpleado) REFERENCES Empleado(idEmpleado) ON DELETE CASCADE ON UPDATE CASCADE
-> );
Query OK, 0 rows affected (0.21 sec)

mysql> ALTER TABLE Empleado ADD COLUMN
-> salario DOUBLE;
Query OK, 0 rows affected (0.46 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
mysql> ALTER TABLE Empleado ADD COLUMN
-> correo VARCHAR(50);
Query OK, 0 rows affected (0.42 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> _
```

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).

```
MySQL 5.7 Command Line Client

mysql> ALTER TABLE Gerente MODIFY noCel VARCHAR(30);
Query OK, 0 rows affected (0.67 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

4. Renombrar la tabla empleado, quedando como **asociado**.

```
MySQL 5.7 Command Line Client
Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE empleado rename as asociado;
Query OK, 0 rows affected (0.38 sec)

mysql> _
```

5. Aumentar en 50 el tamaño del tipo de dato del campo dirección en la tabla asociado.

```
MySQL 5.7 Command Line Client
Query OK, 0 rows affected (0.38 sec)

mysql> ALTER TABLE Asociado MODIFY direccion VARCHAR(150);
Query OK, 0 rows affected (0.42 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql>
```

6. Redefinir la PK de la relación cinemex quedando de la siguiente manera PK(idCinemex,nombre).

```
MySQL 5.7 Command Line Client
Query OK, 0 rows affected (0.42 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> SHOW CREATE TABLE Gerente;
+-----+-----+
| Table | Create Table |
+-----+-----+
| Gerente | CREATE TABLE `gerente` (
  `idGerente` int(11) NOT NULL,
  `nombre` varchar(45) NOT NULL,
  `turno` varchar(15) NOT NULL,
  `noCel` varchar(30) DEFAULT NULL,
  `salario` double NOT NULL,
  `Cinemex_idCinemex` int(11) NOT NULL,
  PRIMARY KEY (`idGerente`),
  KEY `Cinemex_idCinemex` (`Cinemex_idCinemex`),
  CONSTRAINT `gerente_ibfk_1` FOREIGN KEY (`Cinemex_idCinemex`) REFERENCES `cinemex` (`idCinemex`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 |
+-----+-----+
1 row in set (0.02 sec)

mysql>
mysql> ALTER TABLE Gerente DROP FOREIGN KEY gerente_ibfk_1;
Query OK, 0 rows affected (0.38 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

MySQL 5.7 Command Line Client

1 row in set (0.02 sec)

mysql>

mysql> ALTER TABLE Gerente DROP FOREIGN KEY gerente_ibfk_1;

Query OK, 0 rows affected (0.38 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql>

mysql> SHOW CREATE TABLE Cinemex_Empleado;

ERROR 1146 (42S02): Table 'cinemex.cinemex_empleado' doesn't exist

mysql>

mysql> ALTER TABLE CinemexEmpleado DROP FOREIGN KEY cinemexempleado_ibfk_1;

Query OK, 0 rows affected (0.07 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql>

mysql> ALTER TABLE Cinemex DROP PRIMARY KEY;

Query OK, 0 rows affected (0.57 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql>

mysql> ALTER TABLE Cinemex ADD PRIMARY KEY(idCinemex, Nombre);

Query OK, 0 rows affected (0.43 sec)

Records: 0 Duplicates: 0 Warnings: 0

MySQL 5.7 Command Line Client

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

5 rows in set (0.00 sec)

mysql> ALTER TABLE cinemexEmpleado ADD COLUMN

-> nombre_nombreCinemex VARCHAR(45) NOT NULL;

Query OK, 0 rows affected (0.47 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> ALTER TABLE cinemexEmpleado ADD FOREIGN KEY (Cinemex_idCinemex, nombre_nombreCinemex) REFERENCES cinemex(idCinemex, nombre) ON DELETE CASCADE ON UPDATE CASCADE;

Query OK, 0 rows affected (0.82 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> DESC cinemexEmpleado;

Field	Type	Null	Key	Default	Extra
Cinemex_idCinemex	int(11)	NO	PRI	NULL	
Empleado_idEmpleado	varchar(20)	NO	PRI	NULL	
nombre_nombreCinemex	varchar(45)	NO		NULL	

3 rows in set (0.00 sec)

mysql> DESC Cinemex;

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

5 rows in set (0.00 sec)

mysql>

7. Crea la siguiente tabla: **Cartelera**(idCartelera, nombre, fechainicio, fechafin, clasificación) y relaciónala con la tabla **Cinemex**.

MySQL 5.7 Command Line Client

5 rows in set (0.00 sec)

```
mysql>
mysql> CREATE TABLE Cartelera(
-> idCartelera INT NOT NULL PRIMARY KEY,
-> nombre VARCHAR(50) NOT NULL,
-> fechaInicio VARCHAR(20) NOT NULL,
-> fechaFin VARCHAR(20) NOT NULL,
-> clasificacion VARCHAR(30) NOT NULL,
-> id_CinemexCartelera INT NOT NULL,
-> nombreCinemexCartelera VARCHAR(45) NOT NULL,
-> FOREIGN KEY(id_CinemexCartelera, nombreCinemexCartelera) REFERENCES Cinemex(idCinemex,nombre)
-> );
Query OK, 0 rows affected (0.30 sec)
```

mysql>

III. CONCLUSIONES

- Al eliminar una **llave foránea** primero necesitamos ver como se creó la tabla y al código que le asigno a la llave foránea la podemos eliminar con **drop**
- Antes de redefinir una relación compuesta, tenemos que eliminar todas las relaciones de la tabla, realizar la relación compuesta y volver a hacer las relaciones, pero ahora es compuesta.
- ON UPDATE/DELETE CASCADE. Cuando eliminas alguna cosa en la relación padre, se verá afectada también la relación hijo