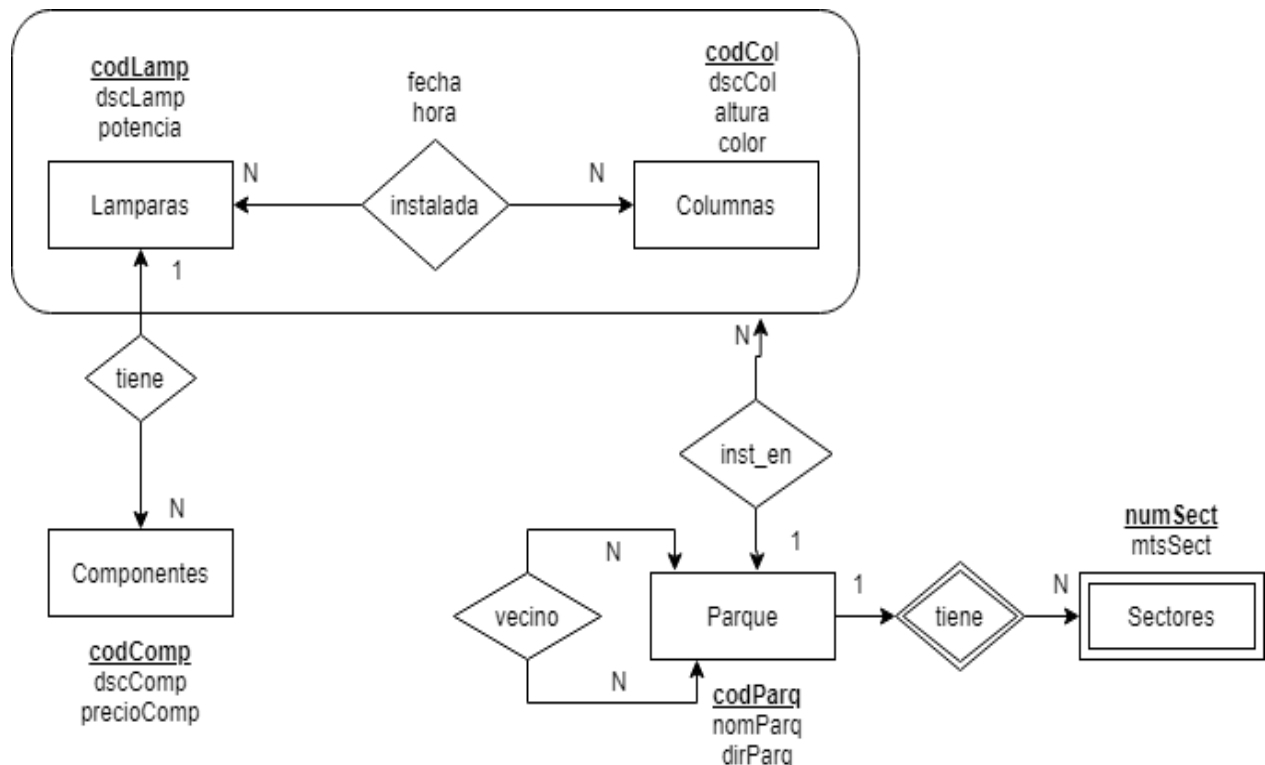


EVALUACION	Examen	GRUPO		FECHA	
MATERIA	Bases de Datos 1				
CARRERA	Analista Programador/Analista en Tecnologías de la Infomración				
CONDICIONES	- Duración: 3 horas - Sin material				

1. Dado el siguiente MER y las reglas del negocio, realizar el pasaje a tablas en 3NF indicando claves primarias y claves foráneas.



3. Utilizando DDL agregar una tabla *Ciudades*(codCiud, nomCiud), agregar a la tabla Parque el código de ciudad y crear la restricción relacional correspondiente.
4. Con las tablas obtenidas resolver los siguientes ejercicios:
 - a. Mostrar descripción y potencia de las lámparas que fueron instaladas en columnas de color gris en el mes de enero de 2023.
 - b. Mostrar nombre de los parques que son vecinos entre sí.
 - c. Mostrar los datos de los componentes que poseen las lámparas que fueron instaladas en mas de 10 parques diferentes.
 - d. Para cada parque, mostrar su nombre, el total de lámparas instaladas en columnas de dicho parque y la potencia total de dichas lámparas, solo mostrar los datos si el total de lámparas instaladas es mayor a 5.

solución

1.

Lámpara(codLamp, dscLamp, potencia)
Columna(codCol, dscCol, altura, color)
Parque(codParq, nomParq, dirParq)
Sector(numcSect, codParq, mtsSect)
Instalada(codLamp, codCol, fecha_hora, codParq)
Componente(codComp, dscComp, precioComp, codLamp)
Vecino(codParq_1, codParq_2)

2.

```
CREATE TABLE Lampara(codLamp int not null,  
                      dscLamp varchar(30) not null,  
                      potencia int,  
                      CONSTRAINT PK_Lampara PRIMARY KEY(codLamp),  
                      CONSTRAINT CK_PotenciaLampara CHECK (potencia >=10))  
  
CREATE TABLE Columna(codCol int not null,  
                      dscCol varchar(30) not null,  
                      altura decimal(10,2),  
                      color varchar(10) not null,  
                      CONSTRAINT PK_Columna PRIMARY KEY(codCol),  
                      CONSTRAINT CK_ColorColumna CHECK(color IN ('Gris','Blanco','Amarillo')))  
  
CREATE TABLE Parque(codParq int not null,  
                      nomParq varchar(30) not null,  
                      dirParq varchar(50),  
                      CONSTRAINT PK_Parque PRIMARY KEY(codParq),  
                      CONSTRAINT UK_NomPARq UNIQUE(nomParq))  
  
CREATE TABLE Sector(numSect int not null,  
                     codParq int not null,  
                     mtsSect decimal(10,2),  
                     CONSTRAINT PK_Sector PRIMARY KEY(numSect,codParq))  
  
CREATE TABLE Instalada(codLamp int not null,  
                        codCol int not null,  
                        fecha_hora datetime,  
                        codParq int not null,  
                        CONSTRAINT PK_Instalada PRIMARY KEY(codLamp,codCol,fecha_hora),  
                        CONSTRAINT FK_LampInstal FOREIGN KEY(codLamp) REFERENCES Lampara(codLamp),  
                        CONSTRAINT FK_ColInstal FOREIGN KEY(codCol) REFERENCES Columna(codCol),  
                        CONSTRAINT FK_ParqInstal FOREIGN KEY(codParq) REFERENCES Parque(codParq))  
  
CREATE TABLE Componente(codComp int not null,  
                          dscComp varchar(30) not null,  
                          precioComp money,  
                          codLamp int,  
                          CONSTRAINT PK_Componente PRIMARY KEY(codComp),  
                          CONSTRAINT FK_LampComo FOREIGN KEY(codLamp) REFERENCES Lampara(codLamp),  
                          CONSTRAINT CK_PrecioComp CHECK(precioComp BETWEEN 10 and 200))  
  
CREATE TABLE Vecino(codParq_1 int not null,  
                     codParq_2 int not null,  
                     CONSTRAINT PK_Vecino PRIMARY KEY(codParq_1,codParq_2),  
                     CONSTRAINT FK_Parq1 FOREIGN KEY(codParq_1) REFERENCES Parque(codParq),  
                     CONSTRAINT FK_Parq2 FOREIGN KEY(codParq_2) REFERENCES Parque(codParq))
```

3.

```
CREATE TABLE Ciudades(codCiud int not null,  
                        nomCiud varchar(30) not null,  
                        CONSTRAINT PK_Ciudad PRIMARY KEY(codCiud))  
  
ALTER TABLE Parque ADD codCiud int  
ALTER TABLE Parque ADD CONSTRAINT FK_CiudadParque FOREIGN KEY(codCiud) REFERENCES Ciudades(codCiud)
```

4.

```
/* a. Mostrar descripción y potencia de las lámparas que fueron instaladas en columnas de color gris en el mes de enero de 2023. */  
SELECT l.dscLamp, l.potencia  
FROM Lampara l, Instalada i, Columna c  
WHERE l.codLamp = i.codLamp and  
      c.codCol = i.codCol and  
      i.fecha_hora BETWEEN '01/01/2023' and '31/12/2023' and  
      c.color='Gris'  
  
/* b. Mostrar nombre de los parques que son vecinos entre sí. */  
SELECT DISTINCT p1.nomParq, p2.nomParq  
FROM Parque p1, Parque p2, Vecino v  
WHERE v.codParq_1 = p1.codParq and  
      v.codParq_2 = p2.codParq  
  
/* c. Mostrar los datos de los componentes que poseen las lámparas que fueron instaladas en mas de 10 parques diferentes. */  
SELECT c.codComp, c.dscComp, c.precioComp  
FROM Componente c, Lampara l, Instalada i  
WHERE c.codLamp = l.codLamp and  
      l.codLamp = i.codLamp  
GROUP BY c.codComp, c.dscComp, c.precioComp  
HAVING COUNT(DISTINCT(i.codParq)) > 10  
  
/* d. Para cada parque, mostrar su nombre, el total de lámparas instaladas en columnas de dicho parque y la potencia total de dichas lámparas, solo mostrar los datos si el total de lámparas instaladas es mayor a 5 */  
SELECT p.nomParq, COUNT(i.codLamp) as Totallamparas, SUM(l.potencia) as PotenciaTotal  
FROM Parque p, Lampara l, Instalada i  
WHERE p.codParq = i.codParq and  
      l.codLamp = i.codLamp  
GROUP BY p.nomParq  
HAVING COUNT(i.codLamp) > 5
```