ENTREGA 2

GRUPO 60
Integrantes:
Joaquín Cáceres
Tomás García

Esquema:

Usuario(idu INT, nombre VARCHAR(300), fecha_nac DATE, correo VARCHAR(300), nacionalidad VARCHAR(60), PRIMARY KEY (idu))

Registro(regid INT, fecha_ent DATE, fecha_sal DATE, estado VARCHAR(60), idu INT, ids INT, PRIMARY KEY (regid), FOREIGN KEY (idu) REFERENCES Usuario, FOREIGN KEY (ids) REFERENCES Sendero)

Sendero(ids INT, nombre VARCHAR(300), largo FLOAT, dificultad VARCHAR(60), duracion INT, PRIMARY KEY (ids))

ParqueSendero(ids INT, idp INT, PRIMARY KEY (ids), FOREIGN KEY (ids) REFERENCES Sendero, FOREIGN KEY (idp) REFERENCES ParquesNacionales)

ParquesNacionales(idp INT, nombre VARCHAR(300), hectareas FLOAT, descripcion VARCHAR(300), tarifa FLOAT, PRIMARY KEY (idp))

ParqueAtractivo(idp INT, ida INT, PRIMARY KEY (ida), FOREIGN KEY (idp) REFERENCES ParquesNacionales, FOREIGN KEY (ida) REFERENCES Atractivo)

Atractivo(idr INT, nombre VARCHAR(300), comentario VARCHAR(300), PRIMARY KEY (idr))

ParqueRegion(idp INT, idr INT, PRIMARY KEY (idp), FOREIGN KEY (idp) REFERENCES ParquesNacionales, FOREIGN KEY (idr) REFERENCES Region)

Region(idr INT, nombre VARCHAR(300), comentario VARCHAR(300), PRIMARY KEY (idr))

RegionVina(idr INT, idv INT, PRIMARY KEY (idv), FOREIGN KEY (idr) REFERENCES Region, FOREIGN KEY (idv) REFERENCES Vina)

Vina(idv INT, nombre VARCHAR(300), telefono_contacto VARCHAR(300), descripcion VARCHAR(300), PRIMARY KEY (idv))

VinaVino(idv INT, idvino INT, PRIMARY KEY (idvino), FOREIGN KEY (idv) REFERENCES Vina, FOREIGN KEY (idvino) REFERENCES Vino)

Vino(idvino INT, nombre VARCHAR(300), descripcion VARCHAR(300), cepa VARCHAR(300), precio FLOAT, PRIMARY KEY (idvino))

VinoTour(eid INT, idvino INT, PRIMARY KEY (eid, idvino), FOREIGN KEY (eid) REFERENCES Tour, FOREIGN KEY (idvino) REFERENCES Vino)

Tour(eid INT, nombre VARCHAR(300), precio FLOAT, PRIMARY KEY (eid))

VinaTour(eid INT, idv INT, PRIMARY KEY (eid, idv), FOREIGN KEY (eid) REFERENCES Tour, FOREIGN KEY (idv) REFERENCES Vina)

Dependencias:

idu -> nombre, fecha_nac, correo, nacionalidad

regid -> fecha_ent, fecha_sal, idu, ids, estado

ids -> nombre, largo, dificultad, duración

idp -> nombre, hectáreas, teléfono, descripción, tarifa

ida ->nombre, descripcion, url

codigo -> nombre, reseña

idv -> nombre, telefono_contacto, descripcion

idvino -> nombre, descripcion, cepa, precio

eid -> nombre, precio

Explicación:

En todas las dependencias funcionales no triviales X -> Y, X es llave por lo tanto el esquema está en BCNF.

Todas las restricciones de llaves primarias y foráneas se encuentran en la creación de las tablas en SQL.

Crear tablas:

```
DROP TABLE IF EXISTS Usuario;

CREATE TABLE Usuario(idu INT, nombre VARCHAR(300), fecha_nac DATE, correo VARCHAR(300), nacionalidad VARCHAR(60), PRIMARY KEY (idu));

DROP TABLE IF EXISTS Sendero;

CREATE TABLE Sendero(ids INT, nombre VARCHAR(300), largo FLOAT, dificultad VARCHAR(60), duracion INT, PRIMARY KEY (ids));

DROP TABLE IF EXISTS ParquesNacionales;

CREATE TABLE ParquesNacionales(idp INT, nombre VARCHAR(300), hectareas FLOAT, descripcion VARCHAR(300), tarifa FLOAT, PRIMARY KEY (idp));

DROP TABLE IF EXISTS Atractivo;

CREATE TABLE Atractivo(ida INT, nombre VARCHAR(300), descripcion VARCHAR(300), url VARCHAR(300), PRIMARY KEY (ida));

DROP TABLE IF EXISTS Region;

CREATE TABLE Region(idr INT, nombre VARCHAR(300), comentario VARCHAR(300), PRIMARY KEY (idr));

DROP TABLE IF EXISTS Vina;
```

```
CREATE TABLE Vina(idv INT, nombre VARCHAR(300), telefono_contacto VARCHAR(300), descripcion VARCHAR(300), PRIMARY KEY (idv));

DROP TABLE IF EXISTS Vino;

CREATE TABLE Vino(idvino INT, nombre VARCHAR(300), descripcion VARCHAR(300), cepa VARCHAR(300), precio FLOAT, PRIMARY KEY (idvino));

DROP TABLE IF EXISTS Tour;

CREATE TABLE Tour(eid INT, nombre VARCHAR(300), precio FLOAT, PRIMARY KEY (eid));

DROP TABLE IF EXISTS Registro;

CREATE TABLE Registro(regid INT, fecha_ent DATE, fecha_sal DATE, estado VARCHAR(60), idu INT, ids INT, PRIMARY KEY (regid), FOREIGN KEY (idu) REFERENCES Usuario, FOREIGN KEY (ids) REFERENCES Sendero);

DROP TABLE IF EXISTS ParqueSendero;

CREATE TABLE ParqueSendero(ids INT, idp INT, PRIMARY KEY (ids), FOREIGN KEY (ids) REFERENCES Sendero, FOREIGN KEY (idp) REFERENCES ParquesNacionales);

DROP TABLE IF EXISTS ParqueAtractivo;

CREATE TABLE ParqueAtractivo(idp INT, ida INT, PRIMARY KEY (ida), FOREIGN KEY (idp) REFERENCES ParquesNacionales, FOREIGN KEY (ida) REFERENCES Atractivo);

DROP TABLE IF EXISTS ParqueRegion;

CREATE TABLE ParqueRegion(idp INT, idr INT, PRIMARY KEY (idp), FOREIGN KEY (idp) REFERENCES ParquesNacionales, FOREIGN KEY (idr) REFERENCES Region);

DROP TABLE IF EXISTS RegionVina;

CREATE TABLE RegionVina(idr INT, idv INT, PRIMARY KEY (idv), FOREIGN KEY (idr) REFERENCES Region, FOREIGN KEY (idv) REFERENCES Vina);

DROP TABLE IF EXISTS VinaVino;

CREATE TABLE VinaVino(idv INT, idvino INT, PRIMARY KEY (idvino), FOREIGN KEY (idv) REFERENCES Vina, FOREIGN KEY (idvino) REFERENCES Vino);

DROP TABLE IF EXISTS VinoTour;

CREATE TABLE VinoTour(eid INT, idvino INT, PRIMARY KEY (eid, idvino), FOREIGN KEY (eid) REFERENCES Tour, FOREIGN KEY (idvino) REFERENCES Vino);
```

```
DROP TABLE IF EXISTS VinaTour;

CREATE TABLE VinaTour(eid INT, idv INT, PRIMARY KEY (eid, idv), FOREIGN KEY (eid)
REFERENCES Tour, FOREIGN KEY (idv) REFERENCES Vina);
```

Consultas:

```
1)  SELECT S.nombre
     FROM Sendero AS S, Registro AS R, Usuario AS U
      WHERE R.idu = i AND S.ids = R.ids AND R.idu = U.idu

2)  SELECT PN.nombre
     FROM ParquesNacionales AS PN, ParqueRegion AS PR
     WHERE PN.idp = PR.idp AND PN.idr = 6
     UNION
     SELECT V.nombre
     FROM Vina As V, RegionVina AS RV
     WHERE RV.idv = V.idv AND V.idr = 6

3)
SELECT V.nombre
FROM Vino AS V, VinoTur AS VT, Tour AS T
WHERE V.idvino = VT.idvino AND  VT.eid = T.eid AND
T.nombre = “nombre” AND V.cepa = “cepa”

4)
SELECT Vina.nombre, V.*
FROM (SELECT *
FROM Vino
WHERE precio = (SELECT MAX(precio)
                        FROM VINO)) AS V, VinaVino AS VV, Vina
WHERE V.idvino = VV.idvino AND VV.idv = Vina.idv

5) SELECT Registro.idu
   FROM Registro, (SELECT ids
   FROM Sendero
   WHERE largo = (SELECT MAX(largo)
                       FROM Sendero)) AS sendero_largo
   WHERE sendero_largo.ids = Registro.ids AND Registro.estado = “en ruta”
6)
SELECT Sendero.nombre
FROM (SELECT ids, COUNT(*) AS total FROM Registro WHERE estado = 'perdido'
GROUP BY ids) AS senderos_perdidos, Sendero
```

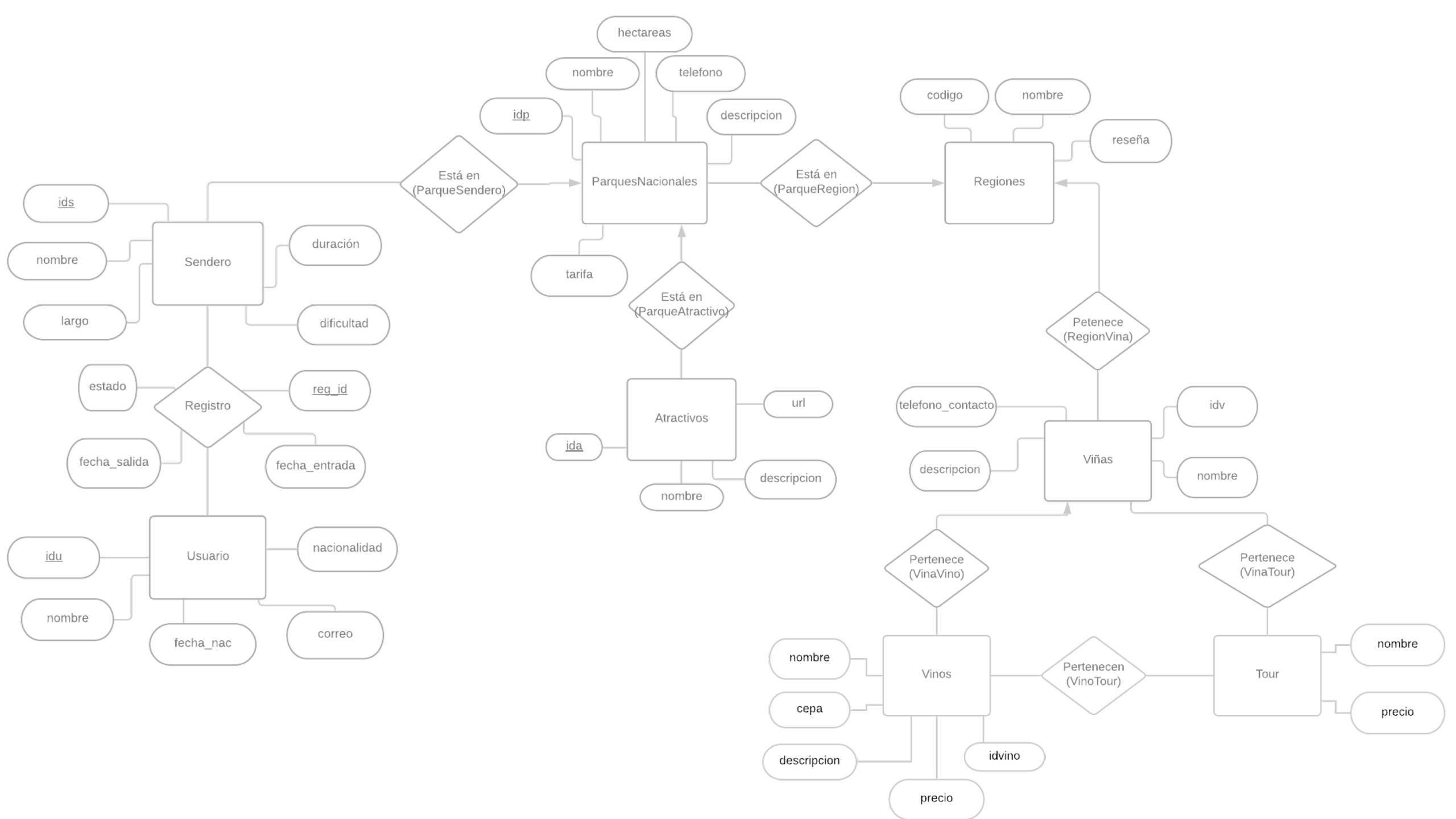
```
WHERE Sendero.ids = senderos_perdidos.ids AND senderos_perdidos.total = (SELECT
MAX(total) FROM (SELECT ids, COUNT(*) AS total FROM Registro WHERE estado =
'perdido' GROUP BY ids) AS q)
```

7)

```
SELECT PN.nombre, COUNT(S.ids), SUM(S.largo)
FROM ParquesNacionales AS PN, Senderos AS S, ParqueSendero AS PS
WHERE PN.idp = PS.idp AND S.ids = PS.ids AND PN.idp = i
GROUP BY PN.nombre
```

8)

```
 SELECT S1.nombre
     FROM Sendero as S1
     WHERE (SELECT COUNT(*)
                 FROM Sendero AS S2
                 WHERE S2.largo > S1.largo) = i-1
```


hectareas
nombre
telefono
idp
descripcion
ParquesNacionales
tarifa
Está en (ParqueSendero)
Está en (ParqueRegion)
Está en (ParqueAtractivo)
codigo
nombre
reseña
Regiones
ids
nombre
largo
Sendero
duración
dificultad
estado
Registro
reg_id
fecha_salida
fecha_entrada
Atractivos
ida
url
descripcion
nombre
Petenece (RegionVina)
telefono_contacto
descripcion
Viñas
idv
nombre
idu
Usuario
nacionalidad
nombre
fecha_nac
correo
Pertenece (VinaVino)
Pertenece (VinaTour)
nombre
cepa
descripcion
Vinos
precio
idvino
Pertenecen (VinoTour)
Tour
nombre
precio