Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ciencia de la Computación

IIC2413 — Bases de Datos — 2' 2019

Entrega 2

Matías Reyes
Nicolás Van Sint Jan

1. Diagrama E/R

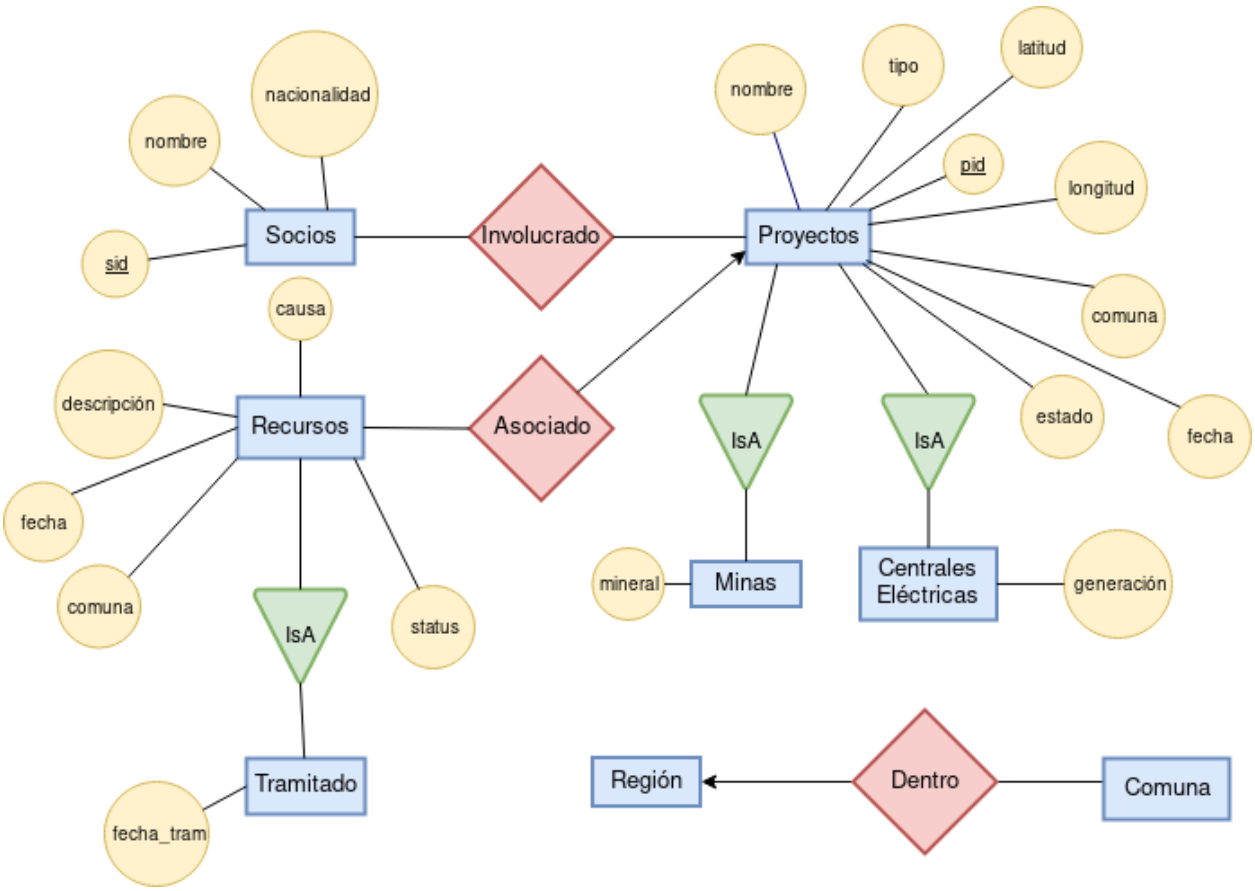

Figura 1: Diagrama Entidad/Relación

2. Esquema

Notación

- subrrayado: Llave primaria
- *cursiva*: Llave foránea

Esquema

- `Socios(ssombre: string, apellido: string, nacionalidad: string)`
- `Proyectos(pnombre: string, latitud: float, longitud: float, comuna: string, fecha_apertura: string, operativo: string, tipo: string)`
- `Recursos(numero: string, causa: string, area: float, descripcion: string, fecha_apertura: string, comuna: string, status: string)`
- `RecursosProyectos(numero: string, pnombre: string)`
- `SociosProyectos(snombre: string, apellido: string, pnombre: string)`
- `Mineras(pnombre: string, mineral: string)`
- `Centrales(pnombre: str, generacion: string)`
- `Tramitados(numero: string, fecha_dictamen: string)`
- `Comunas(comuna: string, region: string)`

2.1. BCNF

El esquema anterior está casi en 3NF porque los atributos que no son llave en cada tabla no dependen de un atributo que no sea llave, con la única excepción del atributo `comuna` que depende de la `latitud` y `longitud` (`latitud,longitud` → `comuna`) y `latitud,comuna` no es llave candidata en `Proyectos`. No consideramos este caso por ser de precisión geográfica exacta, hacer una tabla que rompa la dependencia sería redundante y consumiría espacio de más (tener un proyecto localizado en el mismo lugar que otro proyecto no tiene mucho sentido). Aún así, para poder obtener de manera correcta 3NF debemos crear una nueva relación `Localizacion(latitud: string, longitud:string, comuna:string)` y redifinir `Proyectos` quitando el atributo `comuna`.

Con el cambio anterior, el esquema además está en BCNF dado que está en 3NF y no hay dependencias que no sean superconjuntos de llaves candidatas. Esto primero se da porque todas las relaciones tienen o una llave candidata o ninguna con la excepción de `Socios`; como ya vimos que están en 3NF, todas los atributos dependen de alguna llave, luego es trivial que se cumpla que son superconjuntos de las llave candidata (pues tienen a lo más un elemento). El único caso especial es en la relación `Socios`, pero en este caso la dependencia es que `snombre,apellido` → `nacionalidad`, es decir, es superconjunto de la llave candidata.

3. Comandos usados para crear tablas

```
DROP TABLE IF EXISTS Socios;
CREATE TABLE Socios(
snombre VARCHAR(100),
apellido VARCHAR(100),
nacionalidad VARCHAR(100));
```

```
\COPY Socios FROM 'psql_socios.csv' DELIMITERS ',' CSV HEADER;


DROP TABLE IF EXISTS Proyectos;
CREATE TABLE Proyectos(
tipo VARCHAR(20),
pnombre VARCHAR(100) PRIMARY KEY,
latitud FLOAT,
longitud FLOAT,
comuna VARCHAR(50),
fecha_apertura DATE,
operativo BOOLEAN);
\COPY Proyectos FROM 'psql_proyectos.csv' DELIMITERS ',' CSV HEADER;


DROP TABLE IF EXISTS Recursos;
CREATE TABLE Recursos(
numero CHAR(14) PRIMARY KEY,
causa VARCHAR(100),
area FLOAT,
descripcion TEXT,
fecha_apertura DATE,
comuna VARCHAR(50),
status VARCHAR(20));
\COPY Recursos FROM 'psql_recursos.csv' DELIMITERS ',' CSV HEADER;

DROP TABLE IF EXISTS Mineras;
CREATE TABLE Mineras(
mineral VARCHAR(50),
pnombre VARCHAR(100) PRIMARY KEY,
FOREIGN KEY (pnombre) REFERENCES Proyectos (pnombre) ON DELETE CASCADE);
\COPY Mineras FROM 'psql_mineras.csv' DELIMITERS ',' CSV HEADER;

DROP TABLE IF EXISTS Comunas;
CREATE TABLE Comunas(
comuna VARCHAR(50) PRIMARY KEY,
region VARCHAR(100));
\COPY Comunas FROM 'psql_comunas.csv' DELIMITERS ',' CSV HEADER;

DROP TABLE IF EXISTS Centrales;
CREATE TABLE Centrales(
pnombre VARCHAR(100) PRIMARY KEY,
generacion VARCHAR(50),
FOREIGN KEY (pnombre) REFERENCES Proyectos (pnombre) ON DELETE CASCADE);
\COPY Centrales FROM 'psql_centrales.csv' DELIMITERS ',' CSV HEADER;

DROP TABLE IF EXISTS Tramitados;
CREATE TABLE Tramitados(
fecha_dictamen DATE,
numero CHAR(14) PRIMARY KEY,
FOREIGN KEY (numero) REFERENCES Recursos (numero) ON DELETE CASCADE);
```

```
\COPY Tramitados FROM 'psql_tramitados.csv' DELIMITERS ',' CSV HEADER;

DROP TABLE IF EXISTS SociosProyectos;
CREATE TABLE SociosProyectos(
apellido VARCHAR(100),
pnombre VARCHAR(100),
snombre VARCHAR(100),
FOREIGN KEY (apellido, snombre) REFERENCES Socios (apellido, snombre) ON DELETE CASCADE,
FOREIGN KEY (pnombre) REFERENCES Proyectos (pnombre) ON DELETE CASCADE);
\COPY SociosProyectos FROM 'psql_syp.csv' DELIMITERS ',' CSV HEADER;

DROP TABLE IF EXISTS RecursosProyectos;
CREATE TABLE RecursosProyectos(
numero CHAR(14) PRIMARY KEY,
pnombre VARCHAR(100),
FOREIGN KEY (pnombre) REFERENCES Proyectos (pnombre) ON DELETE CASCADE,
FOREIGN KEY (numero) REFERENCES Recursos (numero) ON DELETE CASCADE);
\COPY RecursosProyectos FROM 'psql_ryp.csv' DELIMITERS ',' CSV HEADER;
```

4. Consultas

1. ```
   SELECT pnombre FROM Centrales
   WHERE generacion='termoeléctrica';
   ```

2. ```
   SELECT pnombre FROM Proyectos NATURAL JOIN Comunas
   WHERE tipo='vertedero'
   AND region LIKE '%Metropolitana%';
   ```

3. ```
   SELECT numero FROM
   Recursos NATURAL JOIN RecursosProyectos NATURAL JOIN Mineras
   WHERE fecha_apertura >= '1990-01-01' AND fecha_apertura <= '2010-12-31';
   ```

4. ```
   SELECT DISTINCT region FROM
   Recursos NATURAL JOIN Comunas
   WHERE status='en trámite';
   ```

5. ```
   SELECT apellido, snombre, pnombre, count FROM
   (SELECT pnombre, COUNT(*) AS count FROM
   Recursos NATURAL JOIN RecursosProyectos
   WHERE status='en trámite'
   GROUP BY pnombre) AS np NATURAL JOIN SociosProyectos
   ORDER BY
   apellido,
   count DESC;
   ```

6. ```
   SELECT DISTINCT Proyectos.pnombre FROM Recursos, RecursosProyectos, Proyectos
   WHERE Proyectos.pnombre=RecursosProyectos.pnombre
   AND Recursos.numero=RecursosProyectos.numero
   AND Recursos.status='aprobado'
   AND Proyectos.operativo=TRUE;
   ```

5. Supuestos extra

- Dado que existían mineras con múltiples minerales, se decidió dejar solo uno (el primero de la lista en los datos originales) para poder usar la llave primaria del nombre de la minera (como se sugiere en la issue #50).