Proyecto2: Base de Datos

28 de Mayo de 2021

Cristhian Rojas
Pontificia Universidad Católica de Chile
Licenciatura en astronomía

Sebastián Lepe
Pontificia Universidad Católica de Chile
Licenciatura en astronomía

1. Diagrama E/R

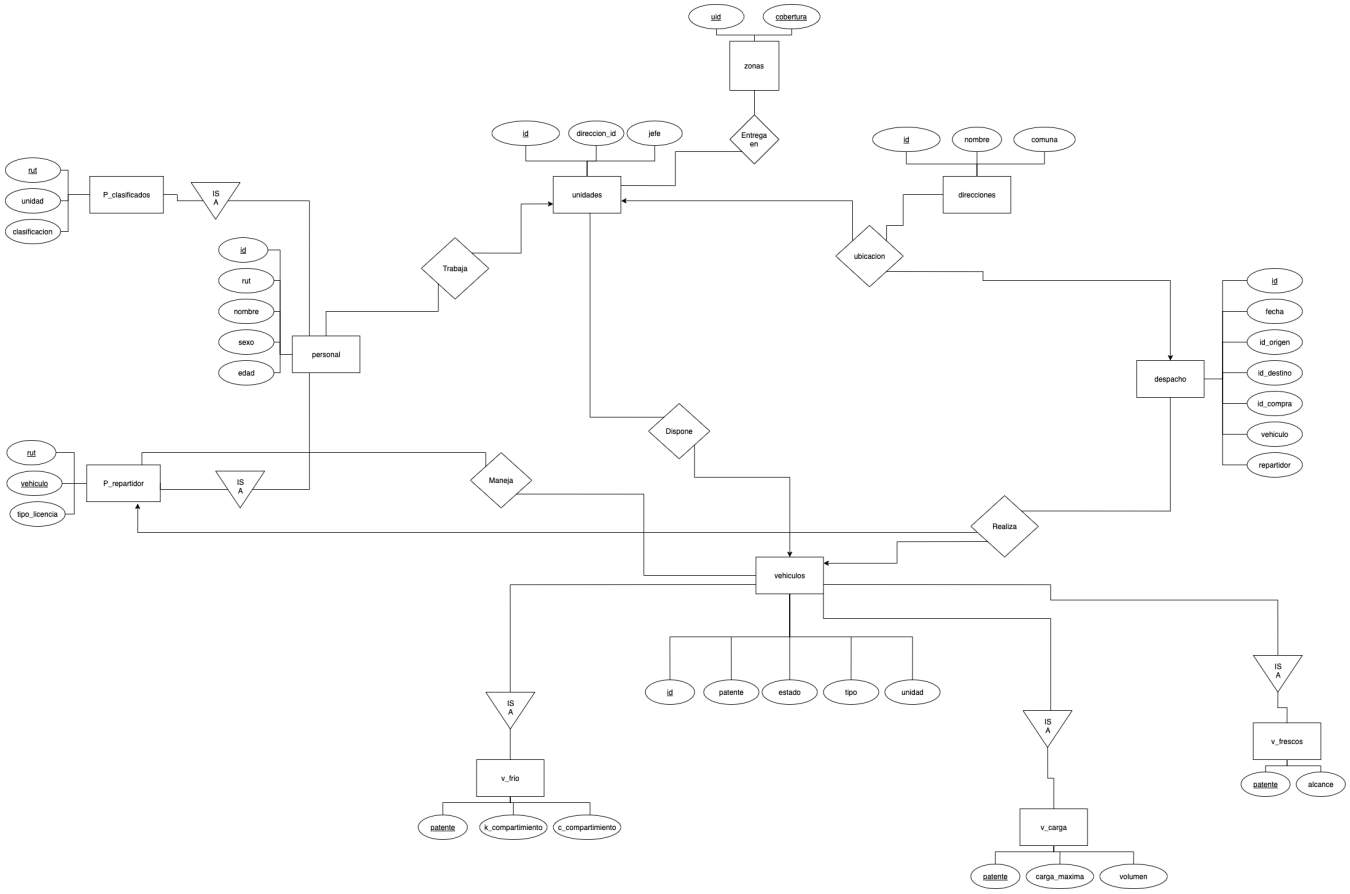

Figura 1: Diagrama E/R

2. Esquema Relacional

Los atributos subrayados corresponden a PRIMARY KEYS, para las foreign keys se especificó el atributo y a que tabla hace referencia.

Unidades {

- uid int
- direccion int
- jefe int

}

Zonas {

- uid int
- cobertura varchar
- FOREIGN KEY(uid) REFERENCES Unidades(uid)

}

Direcciones {

- id int
- nombre varchar
- comuna varchar

}

Personal {

- id int
- rut varchar
- nombre varchar
- sexo varchar
- edad int

}

P_clasificados {

- rut varchar
- unidad int
- clasificacion varchar
- FOREIGN KEY(rut) REFERENCES Personal(rut)

}

P_repartidor {

- rut varchar
- vehiculo int
- tipo licencia varchar
- FOREIGN KEY(rut) REFERENCES Personal(rut)

}

Vehiculos {

- id int
- patente varchar
- estado varchar
- tipo varchar
- unidad int

}

V_carga {

- patente varchar
- carga_maxima float
- volumen float
- FOREIGN KEY(patente) REFERENCES Vehiculos(patente)

}

V_frio {

- patente varchar
- k_compartimiento int
- c_compartimiento int
- FOREIGN KEY(patente) REFERENCES Vehiculos(patente)

}

V_frescos {

- patente varchar
- alcance int
- FOREIGN KEY(patente) REFERENCES Vehiculos(patente)

}

Despacho {

- id int
- fecha timestamp
- id_origen int
- id_destino int
- id_compra int
- vehiculo int
- repartidor int

}

3. Justificacion BCNF

- Unidades esta normalizado y en BCNF ya que se tienen las dependencias:

$$uid \rightarrow direccion, jefe$$

$$jefe \rightarrow uid, direccion$$

$$(uid, jefe) \rightarrow direccion$$

  por lo tanto se tienen estas combinaciones de uid y jefe como super-keys de las cuales elegimos la llave candidata uid como Primary key.

- Zonas esta normalizado y en BCNF ya que se tienen las dependencias:

$$(uid, cobertura) \rightarrow uid$$

$$(uid, cobertura) \rightarrow cobertura$$

  Donde se eligio la llave candidata (uid,cobertura) como Primary key

- Direcciones esta normalizado y en BCNF ya que se tienen las dependencias:

$$id \rightarrow nombre, comuna$$

  en donde id es Primary key.

- Personal esta normalizado y en BCNF ya que se tienen las dependencias:

$$id \rightarrow rut, nombre, sexo, edad$$

$$rut \rightarrow id, nombre, sexo, edad$$

$$(id, rut) \rightarrow nombre, sexo, edad$$

  por lo tanto se tienen estas combinaciones de id y rut como super-keys de las cuales elegimos la llave candidata id como Primary key.

- P_clasificados esta normalizado y en BCNF ya que se tienen las dependencias:

$$rut \rightarrow unidad, clasificacion$$

  en donde rut es Primary key.

- P_repartidor esta normalizado y en BCNF ya que se tienen las dependencias:

$$rut \rightarrow vehiculo, tipo$$

  en donde rut es Primary key.

- Vehiculos esta normalizado y en BCNF ya que tiene las dependencias:

$$id \rightarrow patente, estado, tipo, unidad$$

$$patente \rightarrow id, estado, tipo, unidad$$

$$(id, patente) \rightarrow estado, tipo, unidad$$

  en donde estas combinaciones de id y patente corresponden a super-keys, de las cuales se eligió a la llave candidata id como Primary key

- V_carga esta normalizado y en BCNF ya que se tienen las dependencias:

$$patente \rightarrow carga, volumen$$

  en donde patente es Primary key.

- V_frio esta normalizado y en BCNF ya que se tienen las dependencias:

$$patente \rightarrow k\_compartimiento, c\_compartimiento$$

en donde patente es Primary key.

- V_frescos esta normalizado y en BCNF ya que se tienen las dependencias:

$$patente \rightarrow alcance$$

en donde patente es Primary key.

- Despacho esta normalizada y en BCNF ya que se tienen las dependencias:

$$id \rightarrow fecha, id\_origen, id\_destino, id\_compra, vehiculo, repartidor$$
$$id\_compra \rightarrow id, fecha, id\_origen, id\_destino, vehiculo, repartidor$$
$$(id, id\_compra) \rightarrow fecha, id\_origen, id\_destino, vehiculo, repartidor$$

en donde estas combinaciones de id e id_compra corresponden a las Super-keys de las cuales se eligió la llave candidata id como Primary key.

4. Supuestos

Para la justificación del BCNF nos basamos en lo expresado por el profesor en una issue [1] (ver figura 2) y además de lo obtenido en la pagina [2].

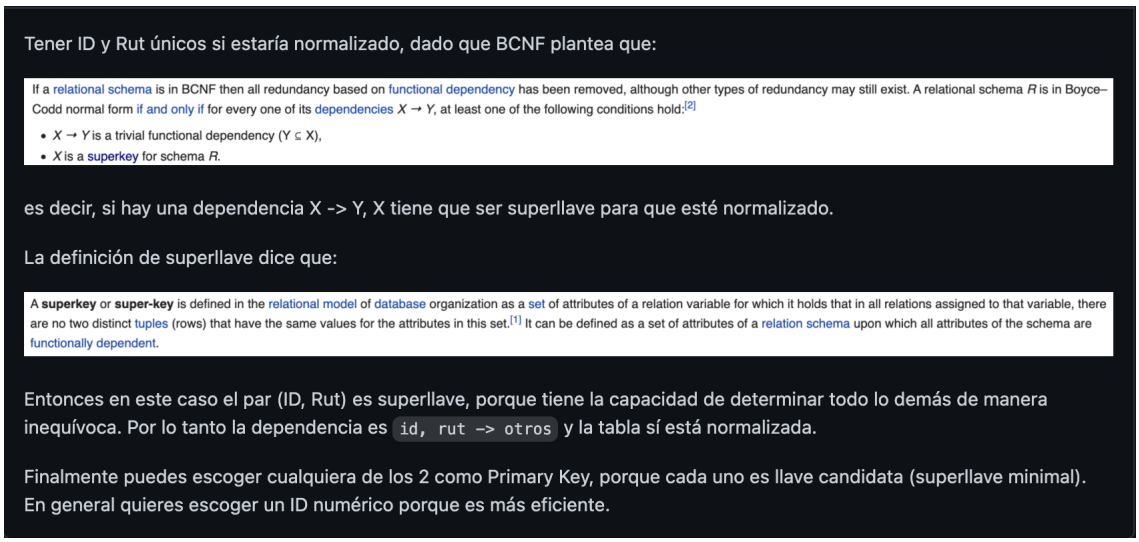

Figura 2: Base de nuestra justificacion BCNF

Para la realización de nuestro proyecto se tomaron en cuenta algunos supuestos.

Al momento de procesar los datos CSV, se noto que para el personal existían elementos erroneos, con distintas edades, y elemenos repetidos, exactamente iguales. Adjuntamos ejemplos.

```
In [220]: for i in personal:
              if i[0] == '683':
                  print(i)

['683', 'Piotr Glenn', '81881771-1', 'mujer', '31', '', '', 'auto', '105\n']
['683', 'Piotr Glenn', '81881771-1', 'mujer', '31', '', '', 'moto', '8\n']
['683', 'Piotr Glenn', '81881771-1', 'mujer', '31', '', '', 'auto', '105\n']
['683', 'Piotr Glenn', '81881771-1', 'mujer', '30', '', '', 'no es necesaria', '29\n']
```

```
['80885102-4', '53', 'no es necesaria']
['80885102-4', '53', 'no es necesaria']
```

Figura 3: Errores en CSV personal

Por ello se decidió dejar las edades minimas de cada persona y eliminar todos los elementos redundantes.

Junto con esto, se obervó que los camiones no pertenecían a ningun grupo de vehículos (frío, frescos o de carga) ya que no poseían ninguno de los atributos necesarios, por lo que se dejaron solamente como parte de Vehículos.

5. Consultas

5.1. Muestre las direcciones de todas las unidades de la empresa de despachos.

Solución:

SELECT direcciones.nombre FROM unidades,direcciones WHERE unidades.direccion_id = direcciones.id

5.2. Ingrese una comuna. Muestre todos los vehículos de las unidades ubicadas en esa comuna.

Solución: Para la variable ingresada se tomó: $comuna_elegida, la comuna ingresada.

SELECT * FROM vehiculos, (SELECT uid FROM direcciones, unidades WHERE unidades.direccion_id = direcciones.id AND direcciones.comuna = '$comuna_elegida') as unidades WHERE vehiculos.unidad = unidades.uid

5.3. Ingrese una comuna y seleccione un ano. Muestre todos los vehículos que hayan realizado un despacho a dicha comuna durante ese ano

Solución: Para las variables ingresadas se tomaron: $comuna, la comuna ingresada; $año, el año ingresado.

SELECT * FROM vehiculos,(SELECT vehiculo FROM direcciones,(SELECT * FROM despacho WHERE CAST(fecha AS text) LIKE '$año%') AS año_vehi WHERE año_vehi.id_destino = direcciones.id AND direcciones.comuna = '$comuna') AS v_año WHERE vehiculos.id = v_año.vehiculo

5.4. Ingrese un tipo de vehículo y seleccione dos números. Muestre todos los despachos realizados por un vehículo del tipo ingresado, y cuyo repartidor tiene una edad entre el rango seleccionado

Solución: Para las variables ingresadas se tomaron: $tipo, tipo de vehiculo; $edad1, primer número elegido; $edad2, segundo numero elegido.

SELECT * FROM vehiculos, despacho, personal WHERE despacho.vehiculo = vehiculos.id AND vehiculos.tipo = '$tipo' AND despacho.repartidor = personal.id AND personal.edad BETWEEN $edad1 AND $edad2

5.5. Ingrese dos comunas. Encuentre los jefes de las unidades qué realizan despachos a ambas comunas.

Solución: Para las variables ingresadas se tomaron: $comuna1, la primera comuna; $comuna2, la segunda comuna

SELECT * FROM personal, (SELECT jefe FROM unidades, (SELECT uid FROM zonas WHERE cobertura = '$comuna1' INTERSECT SELECT uid FROM zonas WHERE cobertura = '$comuna2') AS u WHERE unidades.uid = u.uid) AS jefes WHERE personal.id = jefes.jefe

5.6. Ingrese un tipo de vehículo. Encuentre la unidad qué maneja más vehículos de ese tipo.

Solución: Para la variable ingresadas se tomó: $tipo, tipo de vehículo.

SELECT * FROM (SELECT * FROM unidades, (SELECT count(tipo), unidad FROM vehiculos WHERE tipo = '$tipo' GROUP BY unidad) AS cantidad WHERE cantidad.unidad = unidades.uid) AS und, (SELECT max(cantidad.count) AS max_c FROM unidades, (SELECT count(tipo), unidad FROM vehiculos WHERE tipo = '$tipo' GROUP BY unidad) AS cantidad WHERE cantidad.unidad = unidades.uid) AS max WHERE und.count = max.max_c

Referencias

[1] `https://github.com/IIC2413/Syllabus-2021-1/issues/201`

[2] `https://beginnersbook.com/2015/04/super-key-in-dbms/`