



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA
DEPARTAMENTO DE CIENCIA DE LA COMPUTACIÓN

IIC2413 — Bases de Datos — 1' 2021

Entrega 2 – Pregunta 2

Supuestos y acotaciones

- Los repartidores tienen una sola licencia (issue #54)
- En la relación 1:1 de Jefatura, no es necesario que un Administrador sea jefe de una Unidad. En la clase 5 minuto 27, el profesor dijo que según nuestra convención esto puede ser así.
- El punto anterior se repite para la relación 1:1 de Unidades con Direcciones.
- Para evitar los nulos y como no se asegura que los administradores no pueden ser repartidores, decidimos modelar la herencia de personal mediante tablas múltiples.
- En la corrección de la entrega 1 nos dijeron que era mejor que las primary keys sean int por temas de optimización, es por esto que decidimos que los ruts, nombres y patentes no sean primarys keys y les pusimos un id a todas las relaciones.
- Como los repartidores, a parte de los atributos en personal, no se diferencian más que por sus licencias y vehículos, no es necesario hacer una tabla que los contenga, ya que no hay información para agregar que no esté en las tablas licencias o relación VR. Es por esto que aunque sean una Entidad, no tienen tabla propia ya que sería información redundante.
- Los únicos que pueden ser jefes son los administradores, pero como ellos deben pertenecer a una Unidad de por si, preferimos que en id_jefe se refiera al id en personal para no generar ciclos que no son compatibles.

Esquemas de Relaciones

Personal: id int (primary key), nombre varchar, rut varchar, sexo varchar, edad int, tipo varchar

Dependencias:

id → nombre, rut, sexo, edad, tipo

rut → id, nombre, sexo, edad, tipo

Como id y rut son super llaves, esta relación está en BCNF

Administrativos: id int (primary key) (foreign key references Personal(id)), id_unidad int (foreign key references Unidades(id)), clasificación varchar

Dependencias:

id → id_unidad, clasificación

Como id es una super llave, esta relación está en BCNF

Licencias: id int (primary key), id_personal int (foreign key references Personal(id)), licencia varchar

Dependencias:

id → id_personal, licencia

id_personal, licencia → id

Como id y (id_personal, licencia) son super llaves, esta relación está en BCNF

Unidades: id int (primary key), id_direccion (foreign key references Direcciones(id)), id_jefe int (unique) (foreign key references Personal(id))

Dependencias:

id → id_direccion, id_jefe

id_jefe → id, id_direccion

id_direccion → id, id_jefe (si es que no es posible que dos unidades se encuentren en la misma dirección, si fuera posible, no existiría esta dependencia por lo que no afectaría el BCNF)

Como id, id_jefe e id_direccion son super llaves, esta relación está en BCNF

Cobertura: id int (primary key), id_unidad (foreign key references Unidades(id)), id_comuna (foreign key references Comuna(id))

Dependencias:

id → id_unidad, id_comuna id_unidad, id_comuna → id

Como id e (id_unidad, id_comuna) son super llaves, esta relación está en BCNF

Vehiculos: id int (primary key), patente varchar, id_unidad int (foreign key references Unidades(id)), estado varchar, tipo varchar, categoria varchar

Dependencias:

id → patente, id_unidad, estado, tipo, categoria

patente → id, id_unidad, estado, tipo, categoria

Como id y patente son super llaves, esta relación está en BCNF

Frescos: id int (primary key) (foreign key references Vehiculos(id)), alcance int

Dependencias:

id → alcance Como id es una super llave, esta relación está en BCNF

Frios: id int (primary key) (foreign key references Vehiculos(id)), compartimentos int, capacidad int

Dependencias:

id → compartimentos, capacidad

Como id es una super llave, esta relación está en BCNF

Carga: id int (primary key) (foreign key references Vehiculos(id)), volumen int, toneladas int

Dependencias:

id → volumen, toneladas

Como id es una super llave, esta relación está en BCNF

RelacionVR: id int (primary key), id_vehiculo int (foreign key references Vehiculos(id)), id_repartidor int (foreign key references Personal(id))

Dependencias:

id → id_vehiculo, id_repartidor

id_vehiculo, id_repartidor → id

Como id e (id_vehiculo, id_repartidor) son super llaves, la relación está en BCNF

Comunas: id int, comuna varchar

Dependencias:

id → comuna

comuna → id

Como id y comuna son super llaves, la relación está en BCNF

Direcciones: id int, nombre varchar, id_comuna int (foreign key references Comunas(id))

Dependencias:

$id \rightarrow \text{nombre}, id_comuna$

$\text{nombre}, id_comuna \rightarrow id$ (debido a que no se repiten direcciones en la misma comuna issue #265)

Como id y $(\text{nombre}, id_comuna)$ son super llaves, la relación está en BCNF

Despachos: id int (primary key), fecha date, id_origen int (foreign key references Direcciones(id)), id_destino int (foreign key references Direcciones(id)), id_compra int, id_vehiculo (foreign key references Vehiculos(id)), id_repartidor int (foreign key references Personal(id))

Dependencias:

$id \rightarrow \text{fecha}, id_origen, id_destino, id_compra, id_vehiculo, id_repartidor$

$id_compra \rightarrow id, \text{fecha}, id_origen, id_destino, id_vehiculo, id_repartidor$ (asumiendo que una compra tiene un solo despacho. Si este no fuera el caso, la dependencia no existiría por lo que no afectaría el BCNF)

*(Se asume que dos despachos pueden tener $id_origen, id_destino, id_fecha, id_vehiculo, id_repartidor$ iguales.

Un caso así sería que una persona compre dos cosas en compras diferentes y la empresa para optimizar los despachos, las envía juntos en un viaje. Al asumir esto, no existen más dependencias.)

Como id e id_compra son super llaves, la relación está en BCNF.

Por último, cómo todas las relaciones están en BCNF el esquema relacional está en BCNF.

Consultas Álgebra Relacional

1. *Todas la direcciones de las unidades de despacho.*

```
SELECT Unidades.id, Direcciones.nombre, Comunas.comuna
FROM Unidades, Direcciones, Comunas
WHERE Unidades.id_direccion = Direcciones.id
AND Direcciones.id_comuna = Comunas.id
ORDER BY Unidades.id
```

2. *Todos los vehículos de las unidades ubicadas en la comuna '\$comuna'.*

```
SELECT Vehiculos.id, Vehiculos.patente, Vehiculos.estado, Vehiculos.tipo
FROM Vehiculos, Unidades, Direcciones, Comunas
WHERE Vehiculos.id_unidad = Unidades.id
AND Unidades.id_direccion = Direcciones.id
AND Direcciones.id_comuna = Comunas.id
AND Comunas.comuna LIKE '%$comuna %'
```

3. *Todos los vehículos que hayan realizado un despacho a la comuna '\$comuna' en el año '\$año'.*

```
SELECT Vehiculos.id, Vehiculos.patente, Vehiculos.estado, Vehiculos.tipo
FROM Vehiculos, Despachos, Direcciones, Comunas
WHERE DATE_PART('year', Despachos.fecha) = $año
AND Vehiculos.id = Despachos.id_vehiculo
AND Despachos.id_destino = Direcciones.id
AND Direcciones.id_comuna = Comunas.id
AND Comunas.comuna LIKE '%$comuna %'
```

4. *Todos los despachos realizados por un vehículo de tipo '\$tipo', y cuyo repartidor tiene una edad entre el rango '\$edad1' y '\$edad2'.*

```
SELECT Despachos.id, Despachos.fecha, Despachos.id_origen, Despachos.id_destino, Despachos.id_compra,
Despachos.id_vehiculo, Despachos.id_repartidor, Personal.edad
FROM Despachos, Vehiculos, RelacionVR, Personal
WHERE Vehiculos.tipo = '$tipo'
AND Personal.edad BETWEEN $edad1 AND $edad2
AND Vehiculos.id = Despachos.id_vehiculo
AND Vehiculos.id = RelacionVR.id_vehiculo
AND Despachos.id_repartidor = Personal.id
AND RelacionVR.id_repartidor = Personal.id
ORDER BY Despachos.id
```

5. *Los jefes de la unidades que realizan despachos a la comuna '\$comuna1' y a la '\$comuna2'.*

```
SELECT Unidades.id, Personal.id, Personal.nombre, Personal.rut, Personal.sexo, Personal.edad
FROM Personal, Unidades, Cobertura, Comunas
WHERE Cobertura.id_comuna = Comunas.id
AND Personal.id = Unidades.id_jefe
AND Unidades.id = Cobertura.id_unidad
AND Comunas.comuna LIKE '%$comuna2%'
AND Unidades.id IN
    (SELECT Unidades.id
     FROM Unidades, Cobertura, Comunas
     WHERE Unidades.id = Cobertura.id_unidad
     AND Comunas.id = Cobertura.id_comuna
     AND Comunas.comuna LIKE '%$comuna1 %'
     GROUP BY Unidades.id)
GROUP BY (Unidades.id, Personal.id, Personal.nombre, Personal.rut, Personal.sexo, Personal.edad)
```

La consulta 6 esta en la siguiente página porque no cabía en el espacio que queda :(

6. La unidad que maneja mas vehículos del tipo '\$tipo'.

```
SELECT Cantidades.id, Cantidades.cantidad
FROM (
    SELECT Unidades.id, COUNT(Vehiculos.id) as cantidad
    FROM Unidades, Vehiculos
    WHERE Vehiculos.id_unidad = Unidades.id
    AND Vehiculos.tipo LIKE '%$tipo%'
    GROUP BY Unidades.id) as Cantidades, (
    SELECT MAX(cantidad2) as mayor
    FROM (
        SELECT Unidades.id, COUNT(Vehiculos.id) as cantidad2
        FROM Unidades, Vehiculos
        WHERE Vehiculos.id_unidad = Unidades.id
        AND Vehiculos.tipo LIKE '%$tipo%'
        GROUP BY Unidades.id
        ORDER BY Unidades.id) as Tabla1) as Maximo
WHERE Cantidades.cantidad = Maximo.mayor
```

Una forma de entender esta consulta es:

R1 = Tabla Cantidades (nos entrega una relación que contiene la unidad con el número de vehículos del tipo '\$tipo')

```
SELECT Unidades.id, COUNT(Vehiculos.id) as cantidad
FROM Unidades, Vehiculos
WHERE Vehiculos.id_unidad = Unidades.id
AND Vehiculos.tipo LIKE '%$tipo%'
GROUP BY Unidades.id
```

R2 = Tabla Máximo (nos entrega una relación con un solo dato (mayor): el número más grande entre la columna cantidad de R1)

```
SELECT MAX(cantidad2) as mayor
FROM (R1) as Tabla1
```

Consulta final

```
SELECT Cantidades.id, Cantidades.cantidad
FROM (R2) as mayor, (R1) as Cantidades
WHERE Cantidades.cantidad = Maximo.mayor
```

Es lo mejor que pudimos explicarlo :)

