



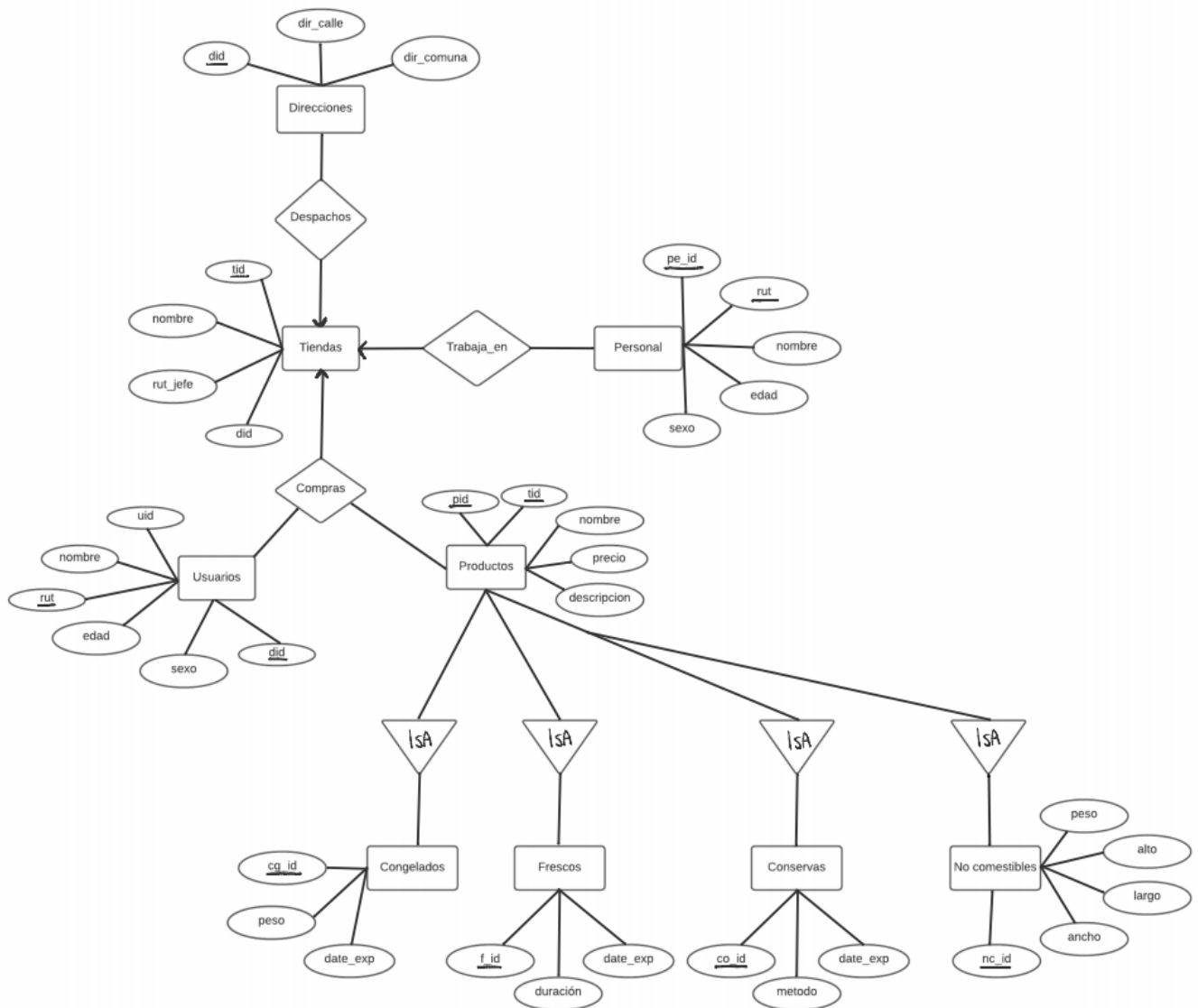
PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
Escuela de Ingeniería  
Departamento de Ingeniería Informática y Computacional  
**IIC2413 Bases de Datos**

---

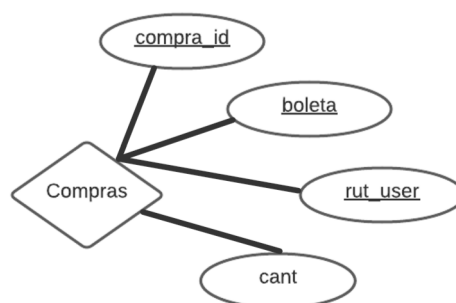
# Entrega 02

Belén Silva Cisternas  
Camilo Oyanadel Fernández

## Diagrama E/R



Por temas de optimización de espacio y que el diagrama fuera más legible optamos por completar la tabla Compras del diagrama anterior en la siguiente imagen:



## Esquema relacional

- Tiendas( tid int, rut\_jefe int, did int, nombre varchar, **foreign key(rut\_jefe) references Personal(rut), foreign key(did) references Direcciones(did)** )
- Despachos( tid int, dir\_comuna varchar, **foreign key(tid) references Tiendas(tid)** )
- Direcciones( did int, dir\_calle varchar, dir\_comuna varchar )
- Personal( pe\_id int, rut varchar, nombre varchar, edad int, sexo varchar )
- Trabaja\_en( rut varchar, tid int, **foreign key(rut) references Personal(rut), foreign key(tid) references Tiendas(tid)** )
- Usuarios( uid int, nombre varchar, rut varchar, edad int, sexo varchar, did int, **foreign key(did) references Direcciones(did)** )
- Compras( boleto int, compra\_id int, tid int, rut\_user varchar, pid int, did int, cant int, **foreign key(pid, tid) references Productos(pid, tid), foreign key(rut\_user, did) references Usuarios(rut, did)** )
- Productos( pid int, tid int, nombre varchar, precio int, descripcion varchar, **foreign key(tid) references Tiendas(tid)** )
- No\_comestibles( nc\_id int, tid int, pid int, ancho int, largo int, alto int, peso int, **foreign key(tid) references Tiendas(tid), foreign key(pid) references Productos(pid)** )
- Congelados( cg\_id int, tid int, pid int, peso int, exp\_date date, **foreign key(tid) references Tiendas(tid), foreign key(pid) references Productos(pid)** )
- Conservas( co\_id int, tid int, pid int, metodo varchar, exp\_date date, **foreign key(tid) references Tiendas(tid), foreign key(pid) references Productos(pid)** )
- Frescos( f\_id int, tid int, pid int, duracion int, exp\_date date, **foreign key(tid) references Tiendas(tid), foreign key(pid) references Productos(pid)** )

## Justificación BCNF

- Tiendas( tid int, rut\_jefe int, did, nombre varchar )

tid -> rut\_jefe, did, nombre

tid es super llave ya que me determina al resto de los atributos, por lo tanto Tiendas está en BCNF.

- Despachos( tid int, dir\_comuna varchar )

La única llave son todos los atributos, por lo tanto Despachos está en BCNF.

- Direcciones( did int, dir\_calle varchar, dir\_comuna varchar )

did -> dir\_calle, dir\_comuna

did es super llave ya que me determina al resto de los atributos, por lo tanto Despachos está en BCNF.

- Personal( pe\_id int, rut int, nombre varchar, edad int, sexo varchar )

pe\_id, rut -> nombre, edad, sexo

rut -> nombre, edad, sexo, pe\_id  
pe\_id -> nombre, edad, sexo, rut

Tenemos que los lados izquierdos son super llaves, por lo tanto se cumple que Personal está en BCNF.

- Trabaja\_en( rut int, tid int )

La única llave son todos los atributos, por lo tanto Trabaja\_en está en BCNF.

- Usuarios( uid int, nombre varchar, rut int, edad int, sexo varchar, did int )

rut, did -> uid, nombre, edad, sexo  
rut -> uid, nombre, edad, sexo

Tenemos que rut, did son super llave y rut es PK, por lo tanto se cumple que Usuarios está en BCNF. [1]

- Compras( bolea int, compra\_id int, tid int, rut\_user int, pid int, did int, cant int )

bolea, compra\_id, rut\_user -> tid, pid, did, cant  
bolea -> compra\_id, tid, rut\_user, pid, did, cant

Tenemos que los lados izquierdos son super llaves, por lo tanto se cumple que Compras está en BCNF.

- Productos( pid int, tid int, nombre varchar, precio int, descripcion varchar )

pid, tid -> nombre, precio, descripcion  
pid -> nombre, precio, descripcion

Tenemos que pid, tid son super llave y pid es PK, por lo tanto se cumple que Productos está en BCNF.

- No\_comestibles( nc\_id int, tid int, pid int, ancho int, largo int, alto int, peso int )

nc\_id, tid, pid -> ancho, largo, alto, peso  
nc\_id -> tid, pid, ancho, largo, alto, peso  
pid -> ancho, largo, alto, peso

Tenemos que nc\_id, tid, pid son super llave, nc\_id es PK y pid es PK. por lo tanto se cumple que No\_comestibles está en BCNF.

- Congelados( cg\_id int, tid int, pid int, peso int, exp\_date date )

cg\_id, tid, pid -> peso, exp\_date  
cg\_id -> tid, pid, peso, exp\_date

pid -> peso, exp\_date

Tenemos que cg\_id, tid, pid son super llave, cg\_id es PK y pid es PK. por lo tanto se cumple que Congelados está en BCNF.

- Conservas( co\_id int, tid int, pid int, metodo varchar, exp\_date date )

co\_id, tid, pid -> metodo, exp\_date  
co\_id -> tid, pid, metodo, exp\_date  
pid -> metodo, exp\_date

Tenemos que co\_id, tid, pid son super llave, co\_id es PK y pid es PK. por lo tanto se cumple que Conservas está en BCNF.

- Frescos( f\_id int, tid int, pid int, duracion int, exp\_date date )

f\_id, tid, pid -> duracion, exp\_date  
f\_id -> tid, pid, duracion, exp\_date  
pid -> duracion, exp\_date

Tenemos que f\_id, tid, pid son super llave, f\_id es PK y pid es PK, por lo tanto se cumple que Frescos está en BCNF.

- Por lo tanto concluimos que nuestro modelo se encuentra en BCNF.

## Consultas SQL

1. SELECT t.nombre, d.dir\_comuna  
FROM Tiendas AS t  
JOIN Despachos AS d  
ON t.id = d.tid  
ORDER BY t.nombre;
2. SELECT p.nombre, p.rut, p.edad, p.sexo  
FROM Personal AS p, Tiendas AS t, Direcciones as d  
WHERE p.rut = t.rut\_jefe  
AND t.did = d.did  
AND LOWER(d.dir\_comuna) = LOWER('\$comuna');
3. **Opción: “No Comestible”**  
SELECT t.tid, t.rut\_jefe, t.did, t.nombre  
FROM Productos AS p, Tiendas AS t, No\_Comestibles AS nc  
WHERE p.tid = t.tid  
AND p.pid = nc.pid  
GROUP BY t.tid, t.rut\_jefe, t.did, t.nombre;

### **Opción: “Congelado”**

SELECT t.tid, t.rut\_jefe, t.did, t.nombre

```

FROM Productos AS p, Tiendas AS t, Congelados AS nc
WHERE p.tid = t.tid
AND p.pid = nc.pid
GROUP BY t.tid, t.rut_jefe, t.did, t.nombre;

```

**Opción: “Conserva”**

```

SELECT t.tid, t.rut_jefe, t.did, t.nombre
FROM Productos AS p, Tiendas AS t, Conservas AS nc
WHERE p.tid = t.tid
AND p.pid = nc.pid
GROUP BY t.tid, t.rut_jefe, t.did, t.nombre;

```

**Opción: “Fresco”**

```

SELECT t.tid, t.rut_jefe, t.did, t.nombre
FROM Productos AS p, Tiendas AS t, Frescos AS nc
WHERE p.tid = t.tid
AND p.pid = nc.pid
GROUP BY t.tid, t.rut_jefe, t.did, t.nombre;

```

**Opción: “Comestible”**

```

SELECT t.tid, t.rut_jefe, t.did, t.nombre
FROM Productos AS p, Tiendas AS t, Congelados AS cg
WHERE p.tid = t.tid
AND p.pid = cg.pid
GROUP BY t.tid, t.rut_jefe, t.did, t.nombre
UNION SELECT t.tid, t.rut_jefe, t.did, t.nombre
FROM Productos AS p, Tiendas AS t, Conservas AS co
WHERE p.tid = t.tid
AND p.pid = co.pid
GROUP BY t.tid, t.rut_jefe, t.did, t.nombre
UNION SELECT t.tid, t.rut_jefe, t.did, t.nombre
FROM Productos AS p, Tiendas AS t, Frescos AS f
WHERE p.tid = t.tid
AND p.pid = f.pid
GROUP BY t.tid, t.rut_jefe, t.did, t.nombre;

```

4. SELECT u.uid, u.rut, u.nombre, u.sexo, u.edad  
FROM Usuarios AS u, Productos AS p, Compras AS c  
WHERE u.rut = c.rut\_user  
AND c.pid = p.pid  
AND LOWER(p.descripcion) LIKE LOWER('%\$descripcion%')  
GROUP BY u.uid, u.rut, u.nombre, u.sexo, u.edad;
5. SELECT d.dir\_comuna, AVG(p.edad)  
FROM Personal AS p, Direcciones AS d, Trabaja\_en AS te, Tiendas as t  
WHERE te.rut = p.rut  
AND te.tid = d.pid  
AND LOWER(d.dir\_comuna) = LOWER('\$comuna')

GROUP BY d.dir\_comuna;

**6. Opción: “No Comestible”**

```
SELECT t.tid, t.rut_jefe, t.did, t.nombre, SUM(c.cant) as cant
FROM Tiendas AS t, Compras AS c, No_Comestibles AS nc
WHERE t.tid = c.tid
AND c.pid = nc.pid
GROUP BY t.tid, t.rut_jefe, t.did, t.nombre
ORDER BY cant DESC
LIMIT 3;
```

**Opción: “Congelado”**

```
SELECT t.tid, t.rut_jefe, t.did, t.nombre, SUM(c.cant) as cant
FROM Tiendas AS t, Compras AS c, Congelados AS nc
WHERE t.tid = c.tid
AND c.pid = nc.pid
GROUP BY t.tid, t.rut_jefe, t.did, t.nombre
ORDER BY cant DESC
LIMIT 3;
```

**Opción: “Conserva”**

```
SELECT t.tid, t.rut_jefe, t.did, t.nombre, SUM(c.cant) as cant
FROM Tiendas AS t, Compras AS c, Conservas AS nc
WHERE t.tid = c.tid
AND c.pid = nc.pid
GROUP BY t.tid, t.rut_jefe, t.did, t.nombre
ORDER BY cant DESC
LIMIT 3;
```

**Opción: “Fresco”**

```
SELECT t.tid, t.rut_jefe, t.did, t.nombre, SUM(c.cant) as cant
FROM Tiendas AS t, Compras AS c, Frescos AS nc
WHERE t.tid = c.tid
AND c.pid = nc.pid
GROUP BY t.tid, t.rut_jefe, t.did, t.nombre
ORDER BY cant DESC
LIMIT 3;
```

**Opción: “Comestible”**

```
SELECT x.tid,x.rut_jefe,x.did,x.nombre, SUM(x.Ventas)
FROM (SELECT t.tid, t.rut_jefe, t.did, t.nombre, SUM(c.cant) AS Ventas
FROM Tiendas AS t, Compras AS c, Congelados AS cg
WHERE t.tid = c.tid
AND c.pid = cg.pid
GROUP BY t.tid, t.rut_jefe, t.did, t.nombre
UNION ALL SELECT t.tid, t.rut_jefe, t.did, t.nombre, SUM(c.cant) AS Ventas
FROM Tiendas AS t, Compras AS c, Conservas AS co
```

```

WHERE t.tid = c.tid
AND c.pid = co.pid
GROUP BY t.tid, t.rut_jefe, t.did, t.nombre
UNION ALL SELECT t.tid, t.rut_jefe, t.did, t.nombre, SUM(c.cant) AS Ventas
FROM Tiendas AS t, Compras AS c, Frescos AS f
WHERE t.tid = c.tid
AND c.pid = f.pid
GROUP BY t.tid, t.rut_jefe, t.did, t.nombre) AS x
GROUP BY x.tid, x.rut_jefe, x.did, x.nombre
ORDER BY SUM(x.Ventas) DESC;

```

## Supuestos

1. Para nuestro esquema uid no tiene alguna función particular, pero para evitar pérdida de información la agregamos en esta tabla.
2. El uso de ORDER BY en la primera consulta es para facilidad de lectura y no afecta la esencia de la solicitud en sí.
3. No se repiten tiendas con los mismos atributos.
4. No se repiten comunas dentro de las comunas a donde se realizan despachos.
5. Todas las personas asociadas a usuarios y personal poseen un rut único e irrepetible.
6. Cada tienda solo puede tener un jefe de tienda y debe pertenecer al personal que trabaja dentro de dicha tienda.
7. Dentro de productos cada producto de cada tienda es unico, esto por que para nosotros dicha tabla actúa más como un definidor antes que almacenador de productos en sí.
8. Las consultas funcionan escribiendo lo que se pide, no mediante identificador.