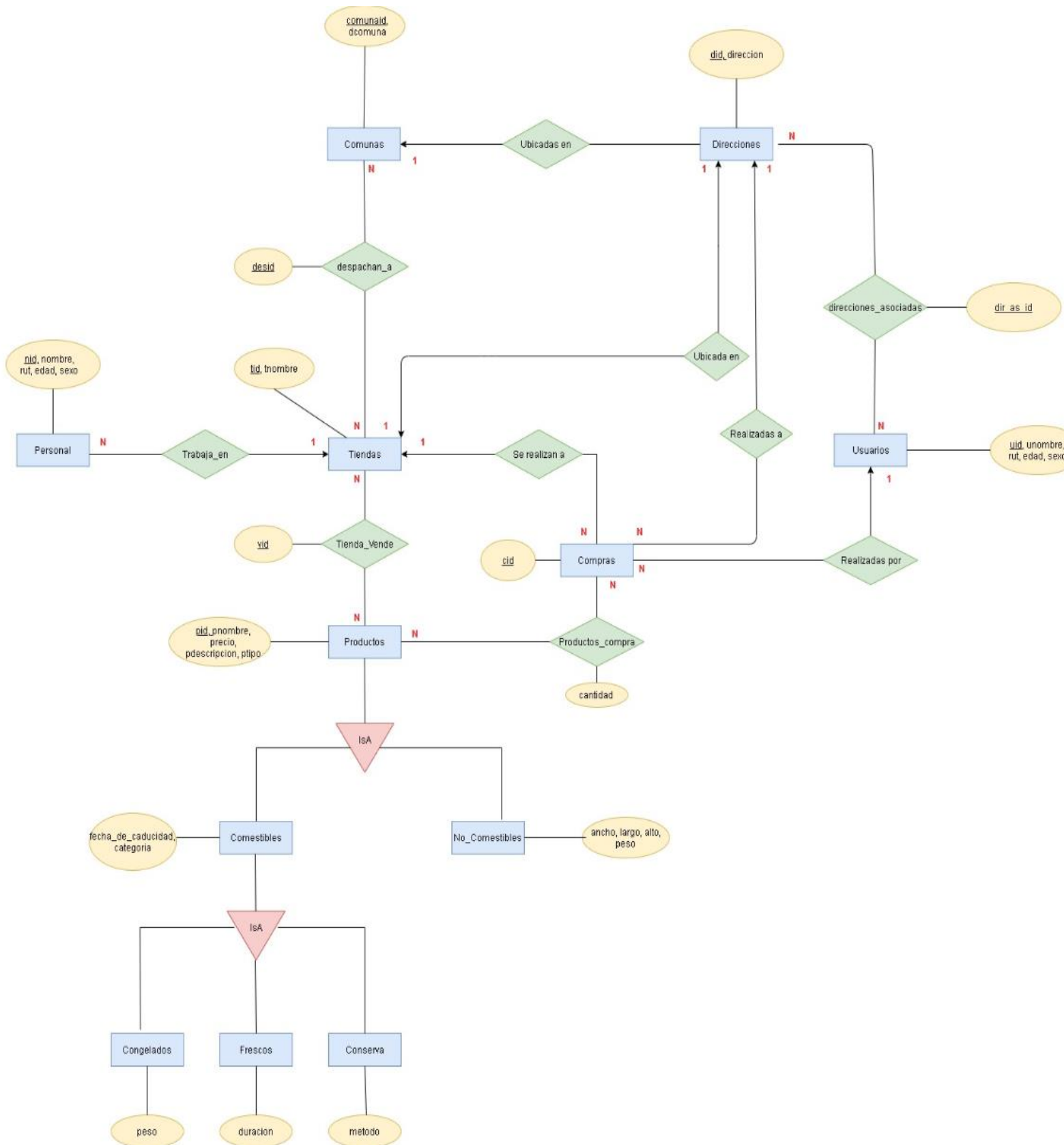


## I. Diagrama Entidad-Relación



## II. Esquema Relacional

**tiendas**(tid INT PRIMARY KEY, tnombre VARCHAR(100), did INT, nid INT, FOREIGN KEY (did) REFERENCES direcciones(did), FOREIGN KEY (nid) REFERENCES personal(nid))

**despachan\_a**(desid INT PRIMARY KEY, tid INT, comunaid INT, FOREIGN KEY (tid) REFERENCES tiendas(tid), FOREIGN KEY (comunaid) REFERENCES comunas(comunaid))

**comunas**(comunaid INT PRIMARY KEY, dcomuna VARCHAR(100))

**tienda\_vende**(vid INT PRIMARY KEY, tid INT, pid INT, FOREIGN KEY (tid) REFERENCES tiendas(tid), FOREIGN KEY (pid) REFERENCES productos(pid))

**productos**(pid INT PRIMARY KEY, pnombre VARCHAR(100), precio INT, pdescpcion VARCHAR(100), ptipo VARCHAR(100))

**comestibles**(pid INT PRIMARY KEY, fecha\_de\_caducidad DATE, categoria VARCHAR(100), FOREIGN KEY (pid) REFERENCES productos(pid))

**congelados**(pid INT PRIMARY KEY, peso FLOAT, FOREIGN KEY (pid) REFERENCES productos(pid))

**frescos**(pid INT PRIMARY KEY, duracion INT, FOREIGN KEY (pid) REFERENCES productos(pid))

**conserva**(pid INT PRIMARY KEY, metodo VARCHAR(100), FOREIGN KEY (pid) REFERENCES productos(pid))

**no\_comestibles**(pid INT PRIMARY KEY, largo INT, alto INT, ancho INT, peso FLOAT, FOREIGN KEY (pid) REFERENCES productos(pid))

**personal**(nid INT PRIMARY KEY, nombre VARCHAR(100), rut VARCHAR(100), edad INT, sexo VARCHAR(100), tid INT, FOREIGN KEY (tid) REFERENCES tiendas(tid))

**direcciones**(did INT PRIMARY KEY, direccion VARCHAR(100), comunaid INT, FOREIGN KEY (comunaid) REFERENCES comunas(comunaid))

**usuarios**(uid INT PRIMARY KEY, unombre VARCHAR(100), rut VARCHAR(100), edad INT, sexo VARCHAR(100))

**direcciones\_asociadas**(dir\_as\_id INT PRIMARY KEY, uid INT, did INT, FOREIGN KEY (uid) REFERENCES usuarios(uid), FOREIGN KEY (did) REFERENCES direcciones(did))

**compras**(cid INT PRIMARY KEY, uid INT, did INT, tid INT, FOREIGN KEY (uid) REFERENCES usuarios(uid), FOREIGN KEY (did) REFERENCES direcciones(did), FOREIGN KEY (tid) REFERENCES tiendas(tid))

**productos\_compra**(cid INT, pid INT, cantidad INT, PRIMARY KEY(cid, pid), FOREIGN KEY (cid) REFERENCES compras(cid), FOREIGN KEY (pid) REFERENCES productos(pid))

### III. Explicación y justificación BCNF

**Comunas:** Esta tabla se crea a partir de querer tener información de todas las comunas en las cuales las tiendas trabajan. Consta de 2 atributos: comunaid y dcomuna (ejemplo: 9, Santiago). En esta tabla encontramos las siguientes dependencias:

- comunaid → dcomuna

En este caso comunaid se comporta como una super llave que determina dcomuna. La única forma de obtener dcomuna es por medio de comunaid en la tabla Comunas, por lo tanto está en BCNF.

**Direcciones:** Esta tabla nace de la necesidad de tener la información de la dirección y la comuna. Consta de los atributos did, direccion, comunaid.

En esta tabla encontramos las siguientes dependencias:

- did → dirección, comunaid

El atributo did determina todo el resto permitiendo identificar correctamente la información. Luego, la relación está en BCNF.

**Usuarios:** Esta tabla se crea para obtener la información de todos los usuarios que van a realizar las compras a las tiendas. Consta de varios atributos descriptivos. Esta tabla posee las siguientes dependencias:

- uid → unombre, rut, edad, sexo
- rut → uid, unombre, edad, sexo

Ambas relaciones se encuentran en BCNF ya que para las dos dependencias funcionales tenemos una super llave en el lado izquierdo.

**Compras:** Esta tabla se crea de la necesidad de obtener la información sobre las compras realizadas. Consta de los atributos: uid, did, tid. Esta tabla posee la siguiente dependencia:

- cid → uid, did, tid

Y mediante cid es la única forma de obtener el uid, did, tid, que es información de la compra. Por lo tanto la relación está en BCNF

**Tiendas:** Esta tabla se crea para representar todas las tiendas con la que contamos en nuestra base de datos. Consta de un identificador de la tienda (tid), el nombre de la tienda (tnombre), la dirección donde se encuentra la tienda (did) y el jefe asociado a dicha tienda (nid). Consta de las siguientes dependencias:

- tid → tnombre, did, nid

Como se puede interpretar, el tid es una super llave que determina todo lo demás. No se aprecian otras dependencias funcionales.

**Personal:** Esta tabla se hace necesaria para obtener información del personal que trabaja en las tiendas. Consta de los atributos nid, nombre, rut, edad, sexo, tid.

Consta de las siguientes dependencias:

- nid → nombre, rut, edad, sexo, tid
- rut → nombre, nid, edad, sexo, tid

Estas relaciones son las únicas que existen para obtener los datos del personal y por lo tanto respeta BCNF.

**Productos:** Esta tabla representa los productos que son los ofrecidos por las tiendas y que los usuarios desean comprar. En un comienzo constaba de varios atributos que decidimos descomponer para poder cumplir la regla BCNF. En fin, quedó con los siguientes atributos: pid, pnombre, precio, pdescripcion y ptipo. Hemos encontrado las siguientes dependencias funcionales:

- pid → pnombre, precio, pdescripcion, ptipo

Cumple la regla básica de BCNF y es la única dependencia trivial de la tabla.

**Comestibles:** Esta tabla se crea para almacenar información sobre los productos comestibles. Tiene los atributos: pid, fecha\_de\_caducidad y categoria.

Esta tabla tiene la siguiente dependencia:

- pid → fecha\_de\_caducidad, categoria

Por lo tanto cumple claramente BCNF

**No\_Comestibles:** Esta tabla se crea para almacenar información sobre los productos no comestibles. Tiene los atributos: pid, largo, alto, ancho, peso. Esta tabla tiene la siguiente dependencia:

- pid → largo, alto, ancho, peso

Cumple la regla BCNF de manera trivial.

**Congelados:** Esta tabla se crea para guardar información acerca de los productos comestibles de la subcategoría congelados. Tiene los atributos: pid, peso.

Tiene la siguiente dependencia:

- pid → peso

**Frescos:** Se crea para almacenar información sobre los productos comestibles de la subcategoría de frescos. Consta de los atributos pid y duración.

- $\text{pid} \rightarrow \text{duracion}$

Como se mencionó anteriormente, al trabajar descomposición BCNF en la tabla productos original planteada y entregada en los enunciados, se nos dan relaciones BCNF para cada una de estas tablas.

**Conserva:** Se crea para almacenar información sobre los productos comestibles de la subcategoría de conserva. Consta de los atributos pid y metodo.

- $\text{pid} \rightarrow \text{metodo}$

Por la misma razón de las tablas anteriores, se cumple la regla BCNF.

**despachan\_a:** Esta tabla se crea a partir de la relación N:N entre la tabla tiendas y comunas. Posee los atributos desid, tid (identificador tienda) y comunaid (identificador comuna). Posee las siguientes dependencias:

- $\text{desid} \rightarrow \text{tid}, \text{comunaid}$
- $\text{tid}, \text{comunaid} \rightarrow \text{desid}$

Tal como vimos en clases, se cumple BCNF ya que las dependencias cumplen con que la superllave determinan el resto evitando así la redundancia.

**tienda\_vende:** Esta tabla nace de la relación N:N entre la tabla tiendas y productos. Así, se representan los productos que vende cada tienda. Posee los atributos vid, tid, pid. Se tiene la siguiente dependencia:

- $\text{vid} \rightarrow \text{tid}, \text{pid}$
- $\text{tid}, \text{pid} \rightarrow \text{vid}$

Y por lo tanto se cumple BCNF, ya que la llave primaria determina el resto.

**direcciones\_asociadas:** Esta tabla nace de la relación N:N entre la tabla usuarios y direcciones. Posee los atributos dir\_as\_id, uid y did. Al igual que las anteriores posee una formulación similar en cuanto a las dependencias:

- $\text{dir\_as\_id} \rightarrow \text{uid}, \text{did}$
- $\text{uid}, \text{did} \rightarrow \text{dir\_as\_id}$

**productos\_compra:** Esta tabla nace de la relación N:N entre la tabla de productos y compras (se realiza para constatar las cantidades de ciertos productos y considerar que una compra puede poseer varios productos asociados). Tiene como atributos cid, pid y cantidad. Posee las siguientes dependencias:

- $\text{cid}, \text{pid} \rightarrow \text{cantidad}$

Al considerar la super llave (cid,pid) donde ambos componentes son llaves primarias que determinan la cantidad correspondiente se encuentra en BCNF (no se determinan otras dependencias).

#### IV. CONSULTAS

Para las consultas en las cuales se requiere ingresar un input por parte del usuario se describen en este apartado como \$var, mientras que las que son seleccionadas por *dropdown* se describen como \$vardp.

**1. Muestre nombre de todas las tiendas, junto con los nombres de las comunas a las cuales realizan despachos.**

```
SELECT tiendas.tid, tiendas.tnombre, comunas.dcomuna
FROM tiendas, despachan_a, comunas
WHERE tiendas.tid = despachan_a.tid AND comunas.comunaid = despachan_a.comunaid
ORDER BY tiendas.tid
```

**2. Ingrese una comuna. Muestre todos los jefes de tiendas ubicadas en dicha comuna.**

```
SELECT personal.nid, personal.nombre
FROM tiendas, personal, direcciones, comunas
WHERE tiendas.nid = personal.nid AND personal.tid = tiendas.tid AND tiendas.did =
direcciones.did AND direcciones.comunaid = comunas.comunaid AND comunas.dcomuna
LIKE '%$var%';
```

**3. Seleccione un tipo de producto. Muestre todas las tiendas que venden al menos un producto de dicha categoría.**

tipo producto = 'comestible' o "no comestible"

```
SELECT DISTINCT tiendas.tid, tiendas.tnombre
FROM tiendas, tienda_vende, productos
WHERE tienda_vende.tid = tiendas.tid AND tienda_vende.pid = productos.pid AND
productos.ptipo LIKE '%$vardp%'
```

tipo producto = "congelados", "frescos" o "conserva"

```
SELECT DISTINCT tiendas.tid, tiendas.tnombre
FROM tiendas, comestibles, tienda_vende
WHERE tienda_vende.tid = tiendas.tid AND tienda_vende.pid = comestibles.pid AND
comestibles.categoria LIKE '%$vardp%'
```

**4. Ingrese una descripción. Muestre todos los usuarios que compraron el producto con esa descripción.**

```
SELECT DISTINCT usuarios.uid, usuarios.unombre
FROM usuarios, compras, productos, productos_compra
WHERE usuarios.uid = compras.uid AND productos_compra.cid = compras.cid AND
productos_compra.pid = productos.pid AND productos.pdescripcion LIKE '%$var%'
```

**5. Ingrese el nombre de una comuna. Muestre la edad promedio de los trabajadores de tiendas en esa comuna.**

```
SELECT AVG(personal.edad)
FROM personal, tiendas, direcciones, comunas
WHERE personal.tid = tiendas.tid AND tiendas.did = direcciones.did AND
direcciones.comunaid = comunas.comunaid AND comunas.dcomuna LIKE '%$var%';
```

**6. Seleccione un tipo de producto. Muestre las tiendas que han registrado la venta de la mayor cantidad de productos del tipo seleccionado.**

tipo producto = "comestible" o "no comestible"

```
SELECT tiendas.tid, tiendas.tnombre, SUM(productos_compra.cantidad) as CC
FROM productos_compra, productos, compras, tiendas
WHERE productos.pid = productos_compra.pid AND compras.cid = productos_compra.cid
AND productos.ptipo = '$vardp' AND compras.tid = tiendas.tid
GROUP BY tiendas.tid
ORDER BY CC DESC
FETCH FIRST 10 ROWS ONLY
```

tipo producto = "congelados" o "frescos" o "conserva"

```
SELECT tiendas.tid, tiendas.tnombre, SUM(productos_compra.cantidad) as CC
FROM productos_compra, comestibles, compras, tiendas
WHERE comestibles.pid = productos_compra.pid AND compras.cid = productos_compra.cid
AND comestibles.categoria = '$vardp' AND compras.tid = tiendas.tid
GROUP BY tiendas.tid
ORDER BY CC DESC
FETCH FIRST 10 ROWS ONLY
```

## **V. SUPUESTOS Y CONSIDERACIONES**

- En productos\_compra nos percatamos de que habían 2 tuplas duplicadas (referenciadas en comprasV2.csv donde se repetían todos los datos excepto la cantidad). Tenía un cid = 223 y pid = 393. Lo preguntamos en el discord del curso y nos dieron la opción de sumar las cantidades y quedarnos así con un dato.
- En la consulta 5 se redondeó el promedio obtenido para tener un resultado más limpio.
- Las tablas finales utilizadas fueron realizadas siguiendo el modelo relacional presentado anteriormente. fueron modificadas mediante DataGrip.
- En la consulta 6 se mostrarán las primeras 10 tiendas, tal como se mencionó en una issue.