



PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE  
FACULTAD DE INGENIERÍA  
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
IIC2413 - BASES DE DATOS  
PRIMER SEMESTRE 2021

# Entrega 2

## - Grupo 71 -

---

### Esquema

1. Usuarios(id INT PRIMARY KEY, nombre VARCHAR(100), rut VARCHAR(10), edad INT, sexo VARCHAR(10))
2. Comunas(id INT PRIMARY KEY, nombre VARCHAR(50))
3. Productos(id INT PRIMARY KEY, nombre VARCHAR(130), precio INT, descripcion VARCHAR(200))
4. Direcciones(id INT PRIMARY KEY, nombre VARCHAR(150), id\_comuna INT, FOREIGN KEY(id\_comuna) REFERENCES Comunas(id) ON DELETE SET NULL)
5. Trabajadores(id INT PRIMARY KEY, nombre VARCHAR(100), rut VARCHAR(10), edad INT, sexo VARCHAR(10))
6. Tiendas(id INT PRIMARY KEY, nombre VARCHAR(100), id\_direccion INT, id\_jefe INT, FOREIGN KEY(id\_direccion) REFERENCES Direcciones(id) ON DELETE SET NULL, FOREIGN KEY(id\_jefe) REFERENCES Trabajadores(id) ON DELETE SET NULL)
7. Compras(id INT PRIMARY KEY, id\_usuario INT, id\_direccion INT, id\_tienda INT, FOREIGN KEY(id\_usuario) REFERENCES Usuarios(id) ON DELETE SET NULL, FOREIGN KEY(id\_direccion) REFERENCES Direcciones(id) ON DELETE SET NULL, FOREIGN KEY(id\_tienda) REFERENCES Tiendas(id) ON DELETE SET NULL)
8. ComprasPorProducto(id\_compra INT, id\_producto INT, cantidad INT, PRIMARY KEY(id\_compra, id\_producto), FOREIGN KEY(id\_compra) REFERENCES Compras(id) ON DELETE CASCADE, FOREIGN KEY(id\_producto) REFERENCES Productos(id) ON DELETE SET NULL)

- 
9. TiendasTrabajadores(id\_tienda INT, id\_trabajador INT, PRIMARY KEY(id\_tienda, id\_trabajador), FOREIGN KEY(id\_tienda) REFERENCES Tiendas(id) ON DELETE CASCADE, FOREIGN KEY(id\_trabajador) REFERENCES Trabajadores(id) ON DELETE CASCADE)
  10. TiendasComunasCobertura(id\_tienda INT, id\_comuna INT, PRIMARY KEY(id\_tienda, id\_comuna), FOREIGN KEY(id\_tienda) REFERENCES Tiendas(id) ON DELETE CASCADE, FOREIGN KEY(id\_comuna) REFERENCES Comunas(id) ON DELETE CASCADE)
  11. DireccionesUsuarios(id\_usuario INT, id\_direccion INT, PRIMARY KEY(id\_usuario, id\_direccion), FOREIGN KEY(id\_usuario) REFERENCES Usuarios(id) ON DELETE CASCADE, FOREIGN KEY(id\_direccion) REFERENCES Direcciones(id) ON DELETE CASCADE)
  12. PerteneceA(id\_producto INT, id\_tienda INT, PRIMARY KEY(id\_producto, id\_tienda), FOREIGN KEY(id\_producto) REFERENCES Productos(id) ON DELETE CASCADE, FOREIGN KEY(id\_tienda) REFERENCES Tiendas(id) ON DELETE CASCADE)
  13. ProductosNoComestibles(id INT PRIMARY KEY, ancho FLOAT, alto FLOAT, largo FLOAT, peso FLOAT, FOREIGN KEY(id) REFERENCES Productos(id) ON DELETE CASCADE)
  14. ProductosComestiblesEnConservas(id INT PRIMARY KEY, metodo\_conservacion VARCHAR(100), fecha\_expiracion DATE, FOREIGN KEY(id) REFERENCES Productos(id) ON DELETE CASCADE)
  15. ProductosComestiblesCongelados(id INT PRIMARY KEY, peso FLOAT, fecha\_expiracion DATE, FOREIGN KEY(id) REFERENCES Productos(id) ON DELETE CASCADE)
  16. ProductosComestiblesFrescos(id INT PRIMARY KEY, duracion\_sin\_refrigeracion FLOAT, fecha\_expiracion DATE, FOREIGN KEY(id) REFERENCES Productos(id) ON DELETE CASCADE)

---

## Supuestos y consideraciones

1. El id es único para cada entidad.
2. Se pre-procesaron los archivos .csv de tal manera de generar nuevos archivos .csv (uno para cada tabla) sin tuplas repetidas. Para eliminar los elementos repetidos se hizo uso de sets.
3. Se eliminaron 2 tuplas de ComprasV2.csv porque tenían repetido el par (id\_compra,id\_producto).

## Justificación esquema BCNF

1. Dependencias funcionales Tiendas:  
 $id \rightarrow nombre, id\_direccion, id\_jefe$

Única dependencia funcional, donde el lado izquierdo es candidato a llave, de hecho es la llave primaria en esta relación. Por lo que se considera la relación en BCNF.

2. Dependencias funcionales Trabajadores:  
 $id \rightarrow nombre, rut, edad, sexo$   
 $rut \rightarrow nombre, edad, sexo$

En ambas dependencias, el lado izquierdo (id y rut) son llave, por lo que la relación se considera BCNF.

3. Dependencias funcionales Comunas:  
 $id \rightarrow nombre$   
 $nombre \rightarrow id$

En ambas dependencias, el lado izquierdo (id y nombre) son llave, por lo que la relación se considera BCNF.

4. Dependencias funcionales Direcciones:  
 $id \rightarrow nombre, id\_comuna$

Única dependencia funcional, donde el lado izquierdo es candidato a llave, de hecho es la llave primaria en esta relación. Por lo que se considera la relación en BCNF.

---

5. Dependencias funcionales Usuarios:

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

$rut \rightarrow nombre, edad, sexo$

En ambas dependencias, el lado izquierdo (id y rut) son llave, por lo que la relación se considera BCNF.

6. Dependencias funcionales Compras:

$id \rightarrow id\_usuario, id\_direccion, id\_tienda$

Única dependencia funcional, donde el lado izquierdo es candidato a llave, de hecho es la llave primaria en esta relación. Por lo que se considera la relación en BCNF.

7. Dependencias funcionales Productos:

$id \rightarrow nombre, precio, descripcion$

Única dependencia funcional, donde el lado izquierdo es candidato a llave, de hecho es la llave primaria en esta relación. Por lo que se considera la relación en BCNF.

8. Dependencias funcionales ProductosNoComestibles:

$id\_producto \rightarrow ancho, alto, largo, peso$

Única dependencia funcional, donde el lado izquierdo es candidato a llave, de hecho es la llave primaria en esta relación. Por lo que se considera la relación en BCNF.

9. Dependencias funcionales ProductosComestiblesEnConservas:

$id\_producto \rightarrow metodo\_conservacion, fecha\_expiracion$

Única dependencia funcional, donde el lado izquierdo es candidato a llave, de hecho es la llave primaria en esta relación. Por lo que se considera la relación en BCNF.

10. Dependencias funcionales ProductosComestiblesCongelados:

$id\_producto \rightarrow peso, fecha\_expiracion$

Única dependencia funcional, donde el lado izquierdo es candidato a llave, de hecho es la llave primaria en esta relación. Por lo que se considera la relación en BCNF.

- 
11. Dependencias funcionales ProductosComestiblesFrescos:  
 $id\_producto \rightarrow duracion\_sin\_refrigeracion, fecha\_expiracion$

Única dependencia funcional, donde el lado izquierdo es candidato a llave, de hecho es la llave primaria en esta relación. Por lo que se considera la relación en BCNF.

12. Dependencias funcionales TiendasComunasCobertura:  
 $id\_tienda, id\_comuna \rightarrow id\_tienda, id\_comuna$

Una dependencia funcional, donde el par se define a si mismo. Al mismo tiempo el par es llave primaria de la relación. Así la relación está en BCNF.

13. Dependencias funcionales DireccionesUsuarios:  
 $id\_usuario, id\_direccion \rightarrow id\_usuario, id\_direccion$

Una dependencia funcional, donde el par se define a si mismo. Al mismo tiempo el par es llave primaria de la relación. Así la relación está en BCNF.

14. Dependencias funcionales PerteneceA:  
 $id\_producto, id\_tienda \rightarrow id\_producto, id\_tienda$

Una dependencia funcional, donde el par se define a si mismo. Al mismo tiempo el par es llave primaria de la relación. Así la relación está en BCNF.

15. Dependencias funcionales ComprasPorProducto:  
 $id\_compra, id\_producto \rightarrow cantidad$

Una dependencia funcional, donde el par se define al resto de atributos y a si mismo. Al mismo tiempo el par es llave primaria de la relación. Así la relación está en BCNF.

16. Dependencias funcionales PerteneceA:  
 $id\_tienda, id\_trabajador \rightarrow id\_tienda, id\_trabajador$

Una dependencia funcional, donde el par se define a si mismo. Al mismo tiempo el par es llave primaria de la relación. Así la relación está en BCNF.

---

## Consultas en SQL

1. Muestre nombre de todas las tiendas, junto con los nombres de las comunas a las cuales realizan despachos. La consulta asociada se muestra a continuación:

```
SELECT DISTINCT Tiendas.id, Tiendas.nombre, Comunas.nombre FROM Tiendas,
Comunas, TiendasComunasCobertura WHERE Tiendas.id = TiendasComunasCobertura.id_tienda
AND Comunas.id = TiendasComunasCobertura.id_comuna ORDER BY Tiendas.id;
```

2. Ingrese una comuna. Muestre todos los jefes de tiendas ubicadas en dicha comuna. Suponiendo que el string asociado a la comuna ingresada se almacena en la variable `nombre_comuna_ingresado` (`nombre_comuna_ingresado = input(str)`), la consulta queda como sigue:

```
SELECT DISTINCT Trabajadores.id, Trabajadores.nombre, Trabajadores.rut, Trabajadores.edad,
Trabajadores.sexo FROM Trabajadores, Tiendas, Direcciones, TiendasTrabajadores,
Comunas WHERE TiendasTrabajadores.id_tienda = Tiendas.id AND Tiendas.id_jefe
= TiendasTrabajadores.id_trabajador AND Tiendas.id_direccion = Direcciones.id AND
Direcciones.id_comuna = Comunas.id AND Comunas.nombre = nombre_comuna_ingresado;
```

3. Seleccione un tipo de producto. Muestre todas las tiendas que venden al menos un producto de dicha categoría. A continuación se muestran las consultas dependiendo del tipo de producto seleccionado. Las opciones a elegir entre tipos de productos son:  
A = ProductosNoComestibles  
B = ProductosComestiblesEnConservas  
C = ProductosComestiblesCongelados  
D = ProductosComestiblesFrescos  
E = ProductosComestibles

Cabe destacar que las consultas A, B, C y D corresponden a una consulta con estructura similar. Además, por su lado, la consulta E resulta ser una unión de las consultas B, C y D.

-Si se selecciona A, la consulta es como sigue:

```
SELECT DISTINCT Tiendas.id, Tiendas.nombre, Tiendas.id_direccion, Tiendas.id_jefe
FROM Tiendas, PerteneceA, Productos, ProductosNoComestibles WHERE Productos.id
= ProductosNoComestibles.id AND Productos.id = PerteneceA.id_producto AND Tiendas.id
= PerteneceA.id_tienda;
```

- Si se selecciona B, la consulta es como sigue:

```
SELECT DISTINCT Tiendas.id, Tiendas.nombre, Tiendas.id_direccion, Tiendas.id_jefe
FROM Tiendas, PerteneceA, Productos, ProductosComestiblesEnConservas WHERE
Productos.id = ProductosComestiblesEnConservas.id AND Productos.id = PerteneceA.id_producto
```

---

AND Tiendas.id = PerteneceA.id\_tienda;

- Si se selecciona C, la consulta es como sigue:

```
SELECT DISTINCT Tiendas.id, Tiendas.nombre, Tiendas.id_direccion, Tiendas.id_jefe
FROM Tiendas, PerteneceA, Productos, ProductosComestiblesCongelados WHERE
Productos.id = ProductosComestiblesCongelados.id AND Productos.id = PerteneceA.id_producto
AND Tiendas.id = PerteneceA.id_tienda;
```

- Si se selecciona D, la consulta es como sigue:

```
SELECT DISTINCT Tiendas.id, Tiendas.nombre, Tiendas.id_direccion, Tiendas.id_jefe
FROM Tiendas, PerteneceA, Productos, ProductosComestiblesFrescos WHERE Productos.id
= ProductosComestiblesFrescos.id AND Productos.id = PerteneceA.id_producto AND
Tiendas.id = PerteneceA.id_tienda;
```

-Finalmente, si se selecciona E, la consulta es como sigue:

```
(SELECT DISTINCT Tiendas.id, Tiendas.nombre, Tiendas.id_direccion, Tiendas.id_jefe
FROM Tiendas, PerteneceA, Productos, ProductosComestiblesEnConservas WHERE
Productos.id = ProductosComestiblesEnConservas.id AND Productos.id = PerteneceA.id_producto
AND Tiendas.id = PerteneceA.id_tienda) UNION (SELECT DISTINCT Tiendas.id,
Tiendas.nombre, Tiendas.id_direccion, Tiendas.id_jefe FROM Tiendas, PerteneceA,
Productos, ProductosComestiblesCongelados
WHERE Productos.id = ProductosComestiblesCongelados.id AND Productos.id =
PerteneceA.id_producto AND Tiendas.id = PerteneceA.id_tienda) UNION (SELECT
DISTINCT Tiendas.id, Tiendas.nombre, Tiendas.id_direccion, Tiendas.id_jefe FROM
Tiendas, PerteneceA, Productos, ProductosComestiblesFrescos WHERE Productos.id
= ProductosComestiblesFrescos.id AND Productos.id = PerteneceA.id_producto AND
Tiendas.id = PerteneceA.id_tienda);
```

4. Ingrese una descripción. Muestre todos los usuarios que compraron el producto con esa descripción. Suponiendo que el string asociado a la descripción ingresada se almacena en la variable descripcion (descripcion = input(str)), la consulta queda como sigue:  

```
SELECT DISTINCT Usuarios.id, Usuarios.nombre, Usuarios.rut, Usuarios.edad, Usuarios.sexo
FROM Compras, ComprasPorProducto, Usuarios, Productos WHERE
ComprasPorProducto.id_producto = Productos.id AND ComprasPorProducto.id_compra
= Compras.id AND Compras.id_usuario = Usuarios.id AND Productos.descripcion =
descripcion;
```
5. Ingrese el nombre de una comuna. Muestre la edad promedio de los trabajadores de tiendas en esa comuna. Suponiendo que el string asociado al nombre de la comuna ingresado se almacena en la variable nombre\_comuna\_ingresado (nombre\_comuna\_ingresado = input(str)), la consulta queda como sigue: 

```
SELECT AVG(Trabajadores.edad)
FROM Trabajadores, TiendasTrabajadores, Tiendas, Direcciones, Comunas WHERE
```

---

TiendasTrabajadores.id\_tienda = Tiendas.id AND Tiendas.id\_direccion = Direcciones.id  
AND Direcciones.id\_comuna = Comunas.id AND TiendasTrabajadores.id\_trabajador  
= Trabajadores.id AND Comunas.nombre = nombre\_comuna\_ingresado GROUP BY  
Comunas.nombre;

6. Seleccione un tipo de producto. Muestre las tiendas que han registrado la venta de la mayor cantidad de productos del tipo seleccionado. A continuación se muestran las consultas dependiendo del tipo de producto seleccionado. Las opciones a elegir entre tipos de productos son:

A = ProductosNoComestibles

B = ProductosComestiblesEnConservas

C = ProductosComestiblesCongelados

D = ProductosComestiblesFrescos

E = ProductosComestibles





