

## ENTREGA 2

### INTEGRANTES:

Lucas Salcedo Muñoz

Martín Ponce Carrasco

### TABLAS UTILIZADAS:

PRODUCTOS
id, INT, PK
nombre, VARCHAR (100)
precio, INT
descripción, VARCHAR (100)
tipo, VARCHAR (30)

PRODUCTOS_NO_COMESTIBLES
producto_id, INT, PK, FK productos(id)
largo, INT
alto, INT
ancho, INT
peso, INT

COMPRAS
id, INT, PK
usuario_id, INT, FK usuarios(id)
direccion_id, INT, FK direcciones(id)
tienda_id, INT, FK tiendas(id)

DESPACHO
tienda_id, INT, FK tiendas(id)
comuna_de_cobertura, VARCHAR (30)

DIRECCION_USUARIOS
direccion_id, INT, PK, FK direcciones(id)
usuario_id, INT, FK usuarios(id)

DIRECCIONES
id, INT, PK
nombre, VARCHAR (100)
comuna, VARCHAR (30)

PRODUCTOS_CONGELADOS
producto_id, INT, PK, FK productos(id)
peso, INT
fecha_caducidad, DATE

PRODUCTOS_CONSERVA
producto_id, INT, PK, FK productos(id)
fecha_caducidad, DATE
tipo_de_conserva, VARCHAR (30)

PRODUCTOS_FRESCOS
producto_id, INT, PK, FK productos(id)
fecha_caducidad, DATE
duración_sin_refrigerar, INT

PRODUCTOS_EN_COMPRA
compra_id, INT, FK compras(id)
producto_id, INT, FK productos(id)
cantidad, INT

PRODUCTOS_EN_TIENDA
tienda_id, INT, FK tiendas(id)
producto_id, INT, FK productos(id)

TIENDAS
id, INT, PK
nombre, VARCHAR (100)
dirección_id, INT, FK direcciones(id)
jefe_id, INT, FK trabajadores(id)

TRABAJADORES_EN_TIENDA
tienda_id, INT, FK tiendas(id)
trabajador_id, INT, FK trabajadores(id)

TRABAJADORES
id, INT, PK
nombre, VARCHAR (100)
rut, VARCHAR (30)
edad, INT
sexo, VARCHAR (30)

USUARIOS
id, INT, PK
nombre, VARCHAR (100)
rut, VARCHAR (30)
edad, INT
sexo, VARCHAR (30)

### JUSTIFICACIÓN MODELO:

Modelamos nuestra base de datos a partir de la forma BCNF, pues cada una de nuestras tablas están en la forma BCNF. Procedemos a justificar cada relación construida, mostrando sus dependencias funcionales y normalización, a excepción de las tablas que poseen 2 columnas, pues por definición deben estar en BCNF.

-Usuarios: La dependencia que existe aquí es de la forma (id, rut) → (nombre, sexo, edad) y como (id, rut) es superllave (determina los demás atributos) la tabla se encuentra en BCNF.

-Trabajadores: La dependencia que existe aquí es de la forma (id, rut) → (nombre, sexo, edad) y como (id, rut) es superllave (determina los demás atributos) la tabla se encuentra en BCNF.

-Compras: La dependencia que existe aquí es de la forma (id) → (usuario\_id, dirección\_id, tienda\_id) y como (id) es llave primaria (determina los demás atributos) la tabla se encuentra en BCNF.

-Tiendas: La dependencia que existe aquí es de la forma (id) → (nombre, dirección\_id, jefe\_id) y como (id) es llave primaria (determina los demás atributos) la tabla se encuentra en BCNF.

-Direcciones: La dependencia que existe aquí es de la forma (id) → (nombre, comuna) y como (id) es llave primaria (determina los demás atributos) la tabla se encuentra en BCNF.

-Productos: La dependencia que existe aquí es de la forma (id) → (nombre, precio, descripcion) y como (id) es llave primaria (determina los demás atributos) la tabla se encuentra en BCNF.

-Productos\_no\_comestibles: La dependencia que existe aquí es de la forma (producto\_id) → (largo, ancho, alto, peso) y como (producto\_id) es llave primaria (determina los demás atributos) la tabla se encuentra en BCNF.

-Productos\_congelados: La dependencia que existe aquí es de la forma (producto\_id) → (fecha\_caducidad, peso) y como (producto\_id) es llave primaria (determina los demás atributos) la tabla se encuentra en BCNF.

-Productos\_frescos: La dependencia que existe aquí es de la forma (producto\_id) → (fecha\_caducidad, duracion\_sin\_refrigerar) y como (producto\_id) es llave primaria (determina los demás atributos) la tabla se encuentra en BCNF.

-Productos\_conserva: La dependencia que existe aquí es de la forma (producto\_id) → (fecha\_caducidad, tipo\_de\_conserva) y como (producto\_id) es llave primaria (determina los demás atributos) la tabla se encuentra en BCNF.

-Productos\_en\_compra: No hay una dependencia funcional en la tabla, por ende, la tabla está en BCNF.

### **SUPUESTOS:**

-Según lo que se estipulaba en el enunciado, asumimos que ingresar era tipear el atributo y seleccionar era escogerlo dentro del deslizador que aparece en pantalla.

-En la consulta 6 mostramos las 10 tiendas con la mayor cantidad de productos del tipo seleccionado ordenadas por cantidad del mismo.

-En las consultas de la 1 a la 4 se ordenan alfabéticamente los resultados.

### **CONSULTAS:**

Nombre comuna:

```
SELECT tiendas.nombre, trabajadores.nombre, direcciones.comuna FROM tiendas,  
direcciones, trabajadores WHERE direcciones.comuna LIKE '%$nombre%' AND  
direcciones.id=tiendas.direccion_id AND tiendas.jefe_id=trabajadores.id ORDER BY  
tiendas.nombre;
```

Descripción:

SELECT DISTINCT usuarios.nombre FROM productos, usuarios, compras, productos\_en\_compra WHERE usuarios.id=compras.usuario\_id AND productos.descripcion LIKE '%\$descripcion%' AND compras.id=productos\_en\_compra.compra\_id AND productos\_en\_compra.producto\_id=productos.id ORDER BY usuarios.nombre;

Nombre Tienda:

SELECT DISTINCT tiendas.nombre, despacho.comuna\_de\_cobertura FROM tiendas, despacho WHERE tiendas.id=despacho.tienda\_id ORDER BY tiendas.nombre;

Tipo: SELECT DISTINCT tiendas.nombre FROM tiendas, productos, productos\_en\_tienda WHERE productos.tipo='\$var' AND tiendas.id=productos\_en\_tienda.tienda\_id AND productos.id=productos\_en\_tienda.producto\_id ORDER BY tiendas.nombre;

Mayor venta:

SELECT DISTINCT tiendas.nombre, SUM(productos\_en\_compra.cantidad) as cantidad FROM tiendas, productos, productos\_en\_compra, compras WHERE productos.tipo='\$var' AND tiendas.id=compras.tienda\_id AND compras.id=productos\_en\_compra.compra\_id AND productos.id=productos\_en\_compra.producto\_id GROUP BY tiendas.id ORDER BY cantidad DESC LIMIT 10;

Edad promedio:

Nombre comuna: SELECT tiendas.nombre, trabajadores.nombre, direcciones.comuna FROM tiendas, direcciones, trabajadores WHERE direcciones.comuna LIKE '%\$nombre%' AND direcciones.id=tiendas.direccion\_id AND tiendas.jefe\_id=trabajadores.id ORDER BY tiendas.nombre; Descripción: SELECT DISTINCT usuarios.nombre FROM productos, usuarios, compras, productos\_en\_compra WHERE usuarios.id=compras.usuario\_id AND productos.descripcion LIKE '%\$descripcion%' AND compras.id=productos\_en\_compra.compra\_id AND productos\_en\_compra.producto\_id=productos.id ORDER BY usuarios.nombre; Nombre Tienda: SELECT DISTINCT tiendas.nombre, despacho.comuna\_de\_cobertura FROM tiendas, despacho WHERE tiendas.id=despacho.tienda\_id ORDER BY tiendas.nombre; Tipo: SELECT DISTINCT tiendas.nombre FROM tiendas, productos, productos\_en\_tienda WHERE productos.tipo='\$var' AND tiendas.id=productos\_en\_tienda.tienda\_id AND productos.id=productos\_en\_tienda.producto\_id ORDER BY tiendas.nombre; Mayor venta: SELECT DISTINCT tiendas.nombre, SUM(productos\_en\_compra.cantidad) as cantidad FROM tiendas, productos, productos\_en\_compra, compras WHERE productos.tipo='\$var' AND tiendas.id=compras.tienda\_id AND compras.id=productos\_en\_compra.compra\_id AND productos.id=productos\_en\_compra.producto\_id GROUP BY tiendas.id ORDER BY cantidad DESC LIMIT 10; Edad promedio: SELECT direcciones.comuna, AVG(trabajadores.edad) FROM trabajadores, trabajadores\_en\_tienda, tiendas, direcciones

```
WHERE direcciones.comuna LIKE '%$comuna%' AND direcciones.id=tiendas.direccion_id
AND tiendas.id=trabajadores_en_tienda.tienda_id AND
trabajadores_en_tienda.trabajador_id=trabajadores.id GROUP BY
direcciones.comuna;SELECT direcciones.comuna, AVG(trabajadores.edad) FROM
trabajadores, trabajadores_en_tienda, tiendas, direcciones WHERE direcciones.comuna
LIKE '%$comuna%' AND direcciones.id=tiendas.direccion_id AND
tiendas.id=trabajadores_en_tienda.tienda_id AND
trabajadores_en_tienda.trabajador_id=trabajadores.id GROUP BY direcciones.comuna;
```

**ESQUEMA:**

