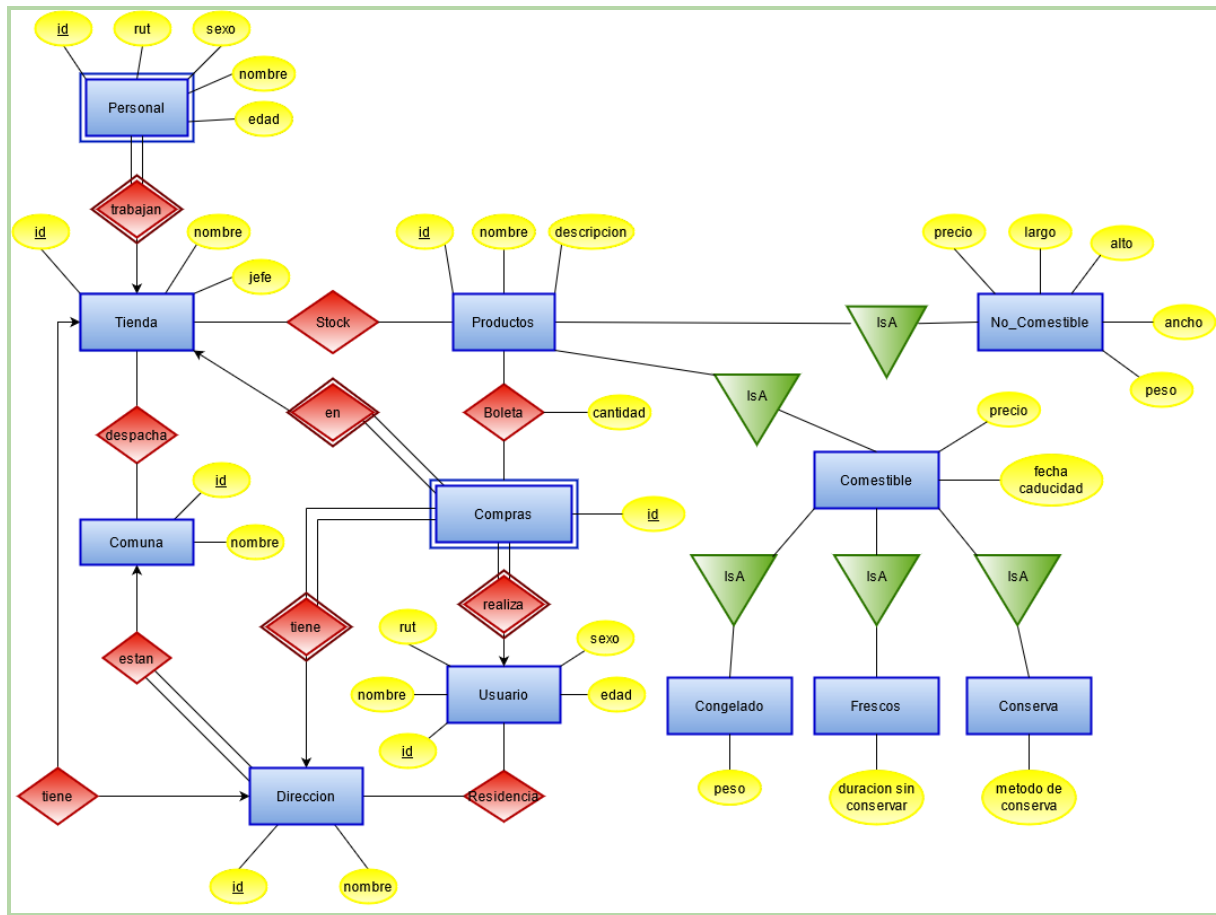


Reporte entrega 2

Grupo 123
Alonso Venegas González
Manuel Cifuentes Cabello

Diagrama



Esquema relacional

Personal(id int,nombre varchar(30) NOT NULL,rut varchar(30) NOT NULL,edad int NOT NULL,sexo varchar(10) NOT NULL,tienda int NOT NULL,PRIMARY KEY (id),FOREIGN KEY(tienda) REFERENCES Tienda(id) ON DELETE CASCADE)

id → nombre, rut, edad, sexo, tienda

rut → id, nombre, edad, sexo, tienda

BCNF

Tienda(id int,nombre varchar(50) NOT NULL,dirección int UNIQUE, jefe int NOT NULL,PRIMARY KEY (id),FOREIGN KEY(dirección) REFERENCES Direccion(id))

id → nombre, jefe, direccion

direccion → id, nombre, jefe

BCNF

Despacha(id_tienda int NOT NULL,id_comuna int NOT NULL,PRIMARY KEY (id_tienda, id_comuna),FOREIGN KEY(id_tienda) REFERENCES Tienda(id)FOREIGN KEY(id_comuna) REFERENCES Comuna(id))

Comuna(id int,nombre varchar(30) NOT NULL,PRIMARY KEY(id))

id → nombre

BCNF

Direccion(id int,nombre varchar(50) NOT NULL,comuna int NOT NULL, PRIMARY KEY (id),FOREIGN KEY(comuna) REFERENCES Comuna(id))

id → nombre, comuna

BCNF

Residencia(id_usuario int NOT NULL,id_direccion int NOT NULL,PRIMARY KEY (id_direccion, id_usuario),FOREIGN KEY(id_direccion) REFERENCES Direccion(id), FOREIGN KEY(id_usuario) REFERENCES Usuario(id))

Usuario(id int,nombre varchar(50) NOT NULL,rut varchar(30) NOT NULL,edad int NOT NULL,sexo varchar(10) NOT NULL,PRIMARY KEY (id))

id → rut, sexo, nombre, edad

rut → id, sexo, nombre, edad

BCNF

Productos(id int,nombre varchar(30) NOT NULL,descripcion varchar(100),PRIMARY KEY(id))

id → nombre, descripcion

BCNF

Compras(id int,id_usuario int NOT NULL,id_direccion int NOT NULL,id_tienda int NOT NULL,PRIMARY KEY(id),FOREIGN KEY(id_tienda) REFERENCES Tienda(id) ON DELETE CASCADE,FOREIGN KEY(id_usuario) REFERENCES Usuario(id) ON DELETE CASCADE,FOREIGN KEY(id_direccion) REFERENCES Direccion(id) ON DELETE CASCADE)

id → id_tienda, id_usuario, id_direccion

BCNF

Boleta*(id_compra int NOT NULL,id_producto int NOT NULL,cantidad int NOT NULL, PRIMARY KEY (id_compra, id_producto),FOREIGN KEY(id_compra) REFERENCES Compras(id),FOREIGN KEY(id_producto) REFERENCES Productos(id))

id_compra, id_producto → cantidad

BCNF

Stock(id_producto int NOT NULL,id_tienda int NOT NULL,PRIMARY KEY (id_tienda, id_producto),FOREIGN KEY(id_tienda) REFERENCES Tienda(id),FOREIGN KEY(id_producto) REFERENCES Productos(id))

No_Comestible(id_producto int NOT NULL,precio int,largo int,alto int,ancho int,peso float,PRIMARY KEY (id_producto),FOREIGN KEY(id_producto) REFERENCES Productos(id))
id_producto → largo, alto, ancho, peso
BCNF

Comestible(id_producto int NOT NULL,precio int,fecha_caducidad DATE,PRIMARY KEY (id_producto),FOREIGN KEY(id_producto) REFERENCES Productos(id))
id_producto → fecha_caducidad
BCNF

Congelado(id_producto_congelado int NOT NULL,peso float,PRIMARY KEY (id_producto_congelado),FOREIGN KEY(id_producto_congelado) REFERENCES Comestible(id_producto))
id_producto → peso
BCNF

Frescos(id_producto_fresco int NOT NULL,duracion_sin_conservar int,PRIMARY KEY (id_producto_fresco),FOREIGN KEY(id_producto_fresco) REFERENCES Comestible(id_producto))
id_producto → duracion_sin_conservar
BCNF

Conserva(id_producto_conserva int NOT NULL,metodo_conserva varchar(20),PRIMARY KEY (id_producto_conserva),FOREIGN KEY(id_producto_conserva) REFERENCES Comestible(id_producto))
id_producto → metodo_conserva
BCNF

Los que están con negrita son entidades mientras que los que están con negrita y cursiva, son relaciones entre entidades.

*Boleta es una relación entre la entidad productos y compras, pero como tiene su propio atributo que es cantidad, la justificamos con el modelo BCNF y la escribimos sin cursiva.

Justificación

Todo nuestro esquema se encuentra en BCNF porque nos basamos principalmente en ese modelo para modelar, valga la redundancia, nuestras dependencias funcionales, ya que por lo visto en clases, creemos que es el mejor método a la hora de crear tablas.

Consultas SQL

1- select tienda.nombre,comuna.nombre from tienda,despacha,comuna where tienda.id=despacha.id_tienda and despacha.id_comuna=comuna.id order by tienda.id

2- select distinct personal.id,personal.nombre,personal.rut,personal.edad,personal.sexo from tienda,personal,despacha,comuna where tienda.jefe=personal.id and tienda.id=despacha.id_tienda and despacha.id_comuna=comuna.id and comuna.nombre like '%\$comuna_b%'

3- Usamos un dropdown que luego seleccionaba un condicional para ver cual desplegar. En el caso de Comestibles: SELECT distinct tienda.nombre from comestible,stock,tienda where stock.id_tienda=tienda.id and stock.id_producto=comestible.id_producto

En el caso de No Comestibles: SELECT distinct tienda.nombre from no_comestible,stock,tienda where stock.id_tienda=tienda.id and stock.id_producto=no_comestible.id_producto

4- select distinct usuario.id,usuario.nombre,usuario.rut,usuario.edad,usuario.sexo,productos.nombre from usuario,compras,boleto,productos where usuario.id=compras.id_usuario AND compras.id=boleto.id_compra AND boleto.id_producto=productos.id AND productos.descripcion like '%\$descripcion_b%

5- select avg(personal.edad) from personal,tienda,direccion,comuna where personal.tienda=tienda.id and tienda.dirección=direccion.id and direccion.comuna=comuna.id and comuna.nombre like '%\$comuna%

6- Al igual que en la tercera consulta usamos un dropdown que luego seleccionaba un condicional para ver cual desplegar. En el caso de comestibles: select tienda.nombre, sum(boleto.cantidad) from comestible, boleto, compras, tienda where comestible.id_producto=boleto.id_producto and boleto.id_compra=compras.id and compras.id_tienda=tienda.id group by tienda.id order by sum(boleto.cantidad) desc limit 5

En el caso no comestibles: select tienda.nombre, sum(boleto.cantidad) from no_comestible, boleto, compras, tienda where no_comestible.id_producto=boleto.id_producto and boleto.id_compra=compras.id and compras.id_tienda=tienda.id group by tienda.id order by sum(boleto.cantidad) desc limit 5

Supuestos

-Hicimos el supuesto de que las tiendas no tenían un cantidad definida de stock de un producto, ya que había veces donde habían compras que excedían la cantidad de productos que tenía una tienda, esto lo vemos en la tabla *productosV2.csv* donde habían casos de filas iguales, es decir una tienda tenía dos veces el mismo producto, lo cual llevaría a pensar que se repetían porque tenía 2 veces el mismo producto, pero como mencionamos al principio, esto significaría que aquellas compras que tienen más cantidad que las disponibles por la tienda tendrían que ser invalidadas, lo cual no era el objetivo de esta

tarea, por lo cual llegamos a la conclusión de eliminar esas filas repetidas en el csv y solo dejar una fila por producto.

-Decidimos que en el caso de que en algún campo de texto no se ingrese nada, mostraremos todos los resultados posibles para todos los inputs posibles.