

Contexto

En una empresa de ventas, es necesario gestionar un inventario de productos que se van almacenando en diferentes proveedores, y a su vez, realizar transacciones de compra y venta. Para esto, se utiliza una base de datos relacional (RDBMS) para organizar toda la información de manera estructurada. El sistema debe permitir agregar productos, actualizar el inventario, registrar compras y ventas, y consultar los datos de manera eficiente.

1. Describir los componentes básicos de una base de datos relacional: tablas, registros, campos, claves primarias y foráneas.

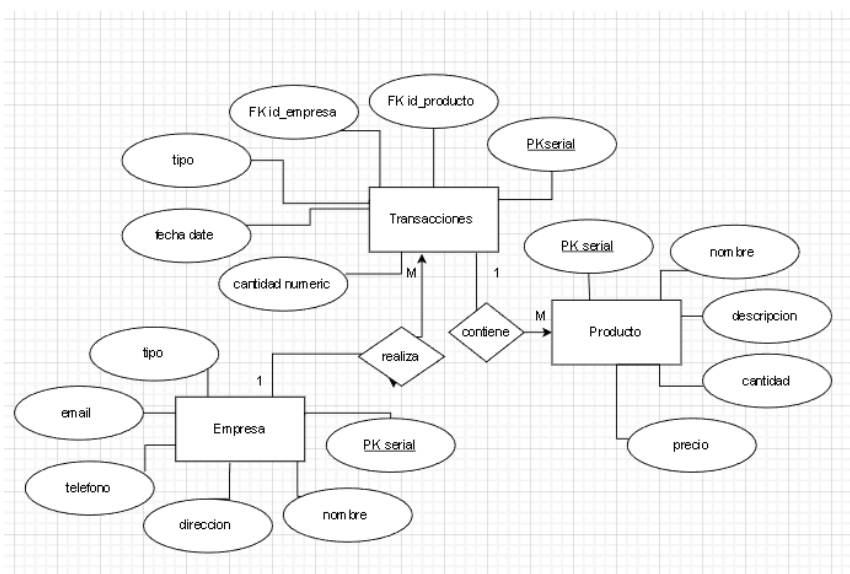
Explicar cómo se gestionan y almacenan los datos en tablas y cómo se establece la relación entre ellas para satisfacer necesidades organizacionales.

Para crear la base de datos se piden tres tablas, Transacciones, Productos y Empresa, que contengan los datos e información de cada registro. Cada tabla tendrá sus respectivos atributos, como *nombre*, *descripción* o *cantidad* e incluyendo en cada una de ellas, una clave primaria para identificarlas de manera única.

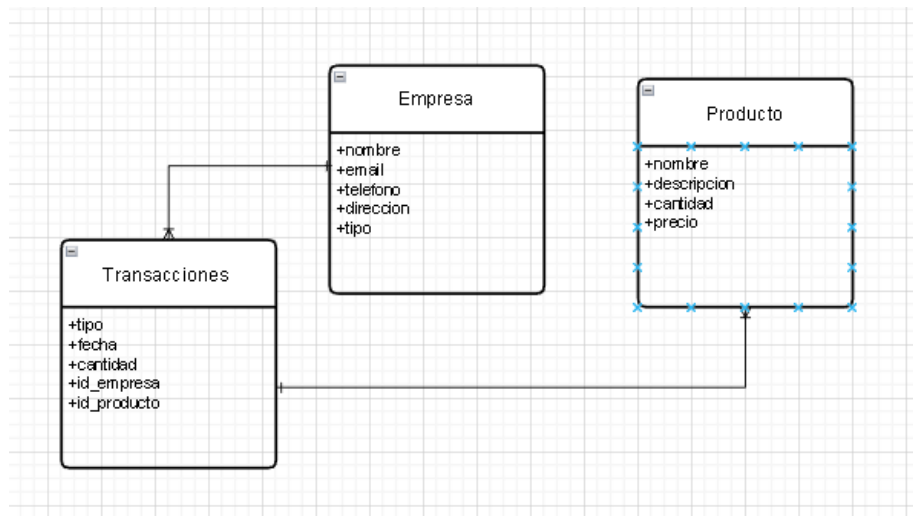
En este caso, para relacionar la información de Productos y Empresa, manteniendo la simplicidad de los datos, se crea una tabla extra, Transacciones. Esta incluye atributos similares, pero además llaves foráneas como forma de relación con las otras tablas. También se establece que cada empresa puede hacer muchas transacciones, pero cada transacción está asociada a una empresa y una cantidad determinada de productos.

5. Elaborar un modelo de datos de acuerdo a los estándares de modelamiento para resolver un problema de baja complejidad.

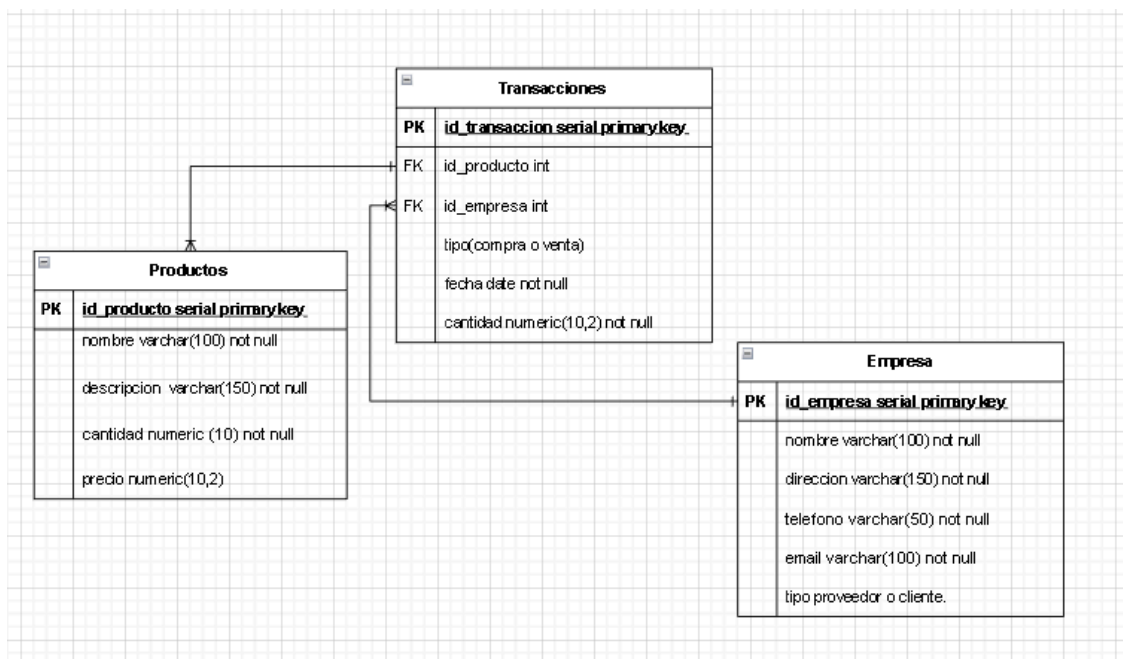
Crear un diagrama entidad-relación (ER) para representar el modelo de datos antes de implementarlo en una base de datos.



Modelo conceptual de Entidad-Relación.



Modelo lógico relacional.



Modelo físico.

4. Implementar estructuras de datos relacionales utilizando lenguaje de definición de datos (DDL) a partir de un modelo de datos para la creación y mantención de las definiciones de los objetos de una base de datos.

Utilizar el lenguaje DDL para crear, modificar y eliminar tablas, índices y otros objetos dentro de una base de datos.

Aquí se muestra la creación de las tablas respectivas con sus atributos y restricciones para no aceptar datos nulos, ni números negativos.

```
Users > fcsan > Desktop > BOOTCAMP > CLASES ASINCRONICAS > TRABAJO INDIVIDUAL > M5 > EVALUACION M5 > AE5.sql

--2.
create database evaluacion_m5;
\c evaluacion_m5

--Crear tabla Empresa
create table empresa (id_empresa serial primary key, nombre varchar(100) not null, direccion varchar(150) not null,
telefono varchar(50) not null, email varchar(100) not null, tipo varchar(20) not null);

-- Crear tabla Productos
create table productos (id_producto serial primary key, nombre varchar(100) not null, descripcion varchar(150) not null,
cantidad int not null, precio numeric(10,2) not null);

--Crear tabla Transacciones
create table transacciones (id_transaccion serial primary key, id_producto int, id_empresa int,
tipo varchar(20) not null, fecha date not null, cantidad int not null, constraint fk_producto foreign key(id_producto) references productos(id_producto),
constraint fk_empresa foreign key (id_empresa) references empresa(id_empresa));
```

```
-- 8. Agregar restricciones para ingresar datos validos
alter table productos add constraint precio_positivo check (precio > 0);

alter table productos add constraint inventario_positivo check (cantidad >= 0);
```

3. Utilizar lenguaje de manipulación de datos (DML) para la modificación de los datos existentes en una base de datos dando solución a un problema planteado.

Implementar consultas de inserción (INSERT), actualización (UPDATE) y eliminación (DELETE) para modificar los datos dentro de las tablas.

Aquí se hicieron 10 inserciones en la tabla Empresa, y se distinguen entre 'proveedor' o 'cliente' junto con otros datos de contacto e identificación única.

```
--Insertar datos a Empresa
insert into empresa(nombre, direccion, telefono, email, tipo) values
('Verduras del Valle', 'Av Campestre 123', '5552220058', 'contacto@valleverde.com', 'proveedor'),
('DulceMania', 'Calle Central 45', '+5698487589', 'info@dulcemanía.cl', 'cliente'),
('Minimarket Miriam', 'Conde de Castellar 321', '+56987895581', 'mmarketm@contacto.com', 'proveedor'),
('Verduras La Finca', 'Av. Norte 77', '+56988775123', 'verduras@info.cl', 'cliente'),
('Confites Arcoiris', 'Av. Simpson 101', '+652898569', 'confites@gmail.com', 'proveedor'),
('Mayorista NaturalFood', 'Calle Mercado 88', '+56987878745', 'contact@naturalfood.com', 'proveedor'),
('Tiendita Martina', 'Av. 25 de Mayo 190', '63220014', 'martina@tienda.com', 'cliente'),
('Supermercado AlPaso', 'Av. Libertad 10', '+56998580010', 'ventas@alpasso.com', 'cliente'),
('Kiosco La Esquina', 'Calle 12', '56320005', 'esquina@kiosco.com', 'proveedor'),
('Distribuidora Confites', 'Zona Industrial 88', '+56985520006', 'admin@confiteplus.com', 'proveedor');
```

Luego se insertaron datos a Productos, y por último se creó la tabla Transacciones. Esta tabla contiene llaves foráneas que la relacionan a los datos de Productos y Empresa, es por eso que no se puede crear mientras las otras tablas no existan y no pueda hacer referencia a ellas en su creación. También se pedía actualizar la cantidad de inventario de un producto después de una venta o compra, por lo que al insertar una transacción se actualizaba el inventario con UPDATE de la tabla Productos.

Tabla Producto.

```
--Insertar datos a Producto
insert into productos(nombre, descripcion, cantidad, precio) values
('Manzana fuji', 'Fruta fresca de temporada', 80, 1.50),
('Platano', 'Banana madura por unidad', 100, 1.20),
('Zanahoria', 'Zanahoria orgánica por kilo', 50, 1.50),
('Lechuga', 'Lechuga crespas frescas', 90, 1.40),
('Tomate', 'Tomate rojo por kilo', 30, 1.70),
('Caramelos acidos', 'Caramelos sabor fresa', 10, 1.75),
('Lolipop', 'Chupetín sabor surtido', 300, 1.00),
('Gomitas', 'Gomitas de frutas surtidas', 150, 4 ),
('Chocolate Leche', 'Mini barra de chocolate', 200, 1.80),
('Palomitas Mantequilla', 'Bolsa de palomitas caramelizadas', 15, 0.9);
```

Tabla Transacciones.

También aquí se utilizó la transacción para no comprometer los datos de forma permanente con begin y commit;

```
--4. 5. Actualiza la cantidad de inventario de un producto después de una venta o compra.
begin;
insert into transacciones(id_producto, id_empresa, tipo, fecha, cantidad) values
(1, 1, 'compra', '2025-10-03', 3);
update productos set cantidad = cantidad + 3 where id_producto = 1;

insert into transacciones(id_producto, id_empresa, tipo, fecha, cantidad) values
(2, 3, 'venta', '2025-05-25', 10);
update productos set cantidad = cantidad - 10 where id_producto = 2;

insert into transacciones(id_producto, id_empresa, tipo, fecha, cantidad) values
(3, 2, 'compra', '2025-07-14', 8);
update productos set cantidad = cantidad + 8 where id_producto = 3;

insert into transacciones(id_producto, id_empresa, tipo, fecha, cantidad) values
(4, 6, 'compra', '2025-08-21', 80);
update productos set cantidad = cantidad + 80 where id_producto = 4;

insert into transacciones(id_producto, id_empresa, tipo, fecha, cantidad) values
(2, 5, 'venta', '2025-02-08', 15);
update productos set cantidad = cantidad - 15 where id_producto = 2;

insert into transacciones(id_producto, id_empresa, tipo, fecha, cantidad) values
(5, 5, 'venta', '2025-10-25', 20);
update productos set cantidad = cantidad - 20 where id_producto = 5;

commit;
```

2. Utilizar Lenguaje Estructurado de Consultas (SQL) para la obtención de información que satisfaga los requerimientos planteados a partir de un modelo de datos dado.

Desarrollar consultas SQL para obtener información específica de las tablas, utilizando cláusulas como `SELECT`, `WHERE`, `JOIN`, `GROUP BY`, entre otras.

Aquí se pide recuperar todos los proveedores que suministran un producto como la lechuga, y qué cantidad hay en el inventario. Para eso se integran las tres tablas a través de un `JOIN` y sus respectivas llaves.

```
select empresa.nombre as proveedor, productos.nombre as producto, productos.cantidad as cantidad_inventario
from empresa join transacciones on empresa.id_empresa = transacciones.id_empresa
join productos on productos.id_producto = transacciones.id_producto
where productos.nombre ilike 'Lechuga' and transacciones.tipo = 'compra' and empresa.tipo = 'proveedor';
```

proveedor	producto	cantidad_inventario
Mayorista NaturalFood	Lechuga	170

(1 fila)

Aquí se pide sumar todas las compras del mes de octubre del 2025 y a cuánto dinero equivale. Para esto se utiliza la función `SUM()`, y creando como condición un rango en la fecha de transacción, además de relacionarlo con la tabla `Productos` para poder multiplicar el valor del producto con la cantidad.

```
select sum(transacciones.cantidad) AS total_compras, sum(transacciones.cantidad * productos.precio) as valor_total
from transacciones join productos on transacciones.id_producto = productos.id_producto
where transacciones.fecha >= '2025-01-01' and transacciones.fecha <= '2025-10-31'
and transacciones.tipo = 'compra';
```

total_compras	valor_total
91	128.50

(1 fila)

Se pide contar las compras y ventas del mes de octubre y agrupar los datos por nombre de empresa, fecha y tipo de transacción. Se utiliza la función `COUNT()` para contar las transacciones y se hizo un `JOIN` con la tabla `Empresa` para relacionar el nombre de la empresa con la transacción.

```
select count(transacciones.cantidad) as total_productos, transacciones.fecha, transacciones.tipo, empresa.nombre
from transacciones join empresa on transacciones.id_empresa = empresa.id_empresa
where transacciones.fecha >= '2025-10-01' and transacciones.fecha <= '2025-10-31'
group by empresa.nombre, transacciones.fecha, transacciones.tipo;
```

total_productos	fecha	tipo	nombre
1	2025-10-25	venta	Confites Arcoíris
1	2025-10-03	compra	Verduras del Valle

(2 filas)

