a) **Especificación de requisitos:**

Queremos implementar una tienda de ropa donde tienen una cantidad específica de prendas en la que se especifique el tipo, la talla, el color, el precio y un número de serie único.

La tienda tiene un nombre y se encuentra en una dirección específica.

De los clientes que compren dichos artículos deberemos saber su nombre, apellido, su número de teléfono y su DNI.

Los empleados que trabajan para la empresa tienen un id de empleado para cada uno de ellos, un salario y una experiencia.

Cada empleado pertenece a un departamento el cual tiene un nombre y un sector diferente del resto.

Los empleados que trabajan en la tienda no pueden trabajar en otras pero la tienda tiene varios trabajadores.

Los clientes no están obligados a comprar prendas siempre que entren a la tienda, los empleados no pueden pertenecer a más de un departamento y los departamentos están formados por uno o más trabajadores.

La tienda suele tener todo en stock pero hay veces que tiene alguna prenda agotada, si es el caso, puedes buscarla en cualquier otra tienda.

Por último, las compras realizadas por la tienda tienen un número de factura, un precio y una fecha de compra.

b) Diseño Conceptual:

Diagrama entregado por el otro grupo sin corregir:

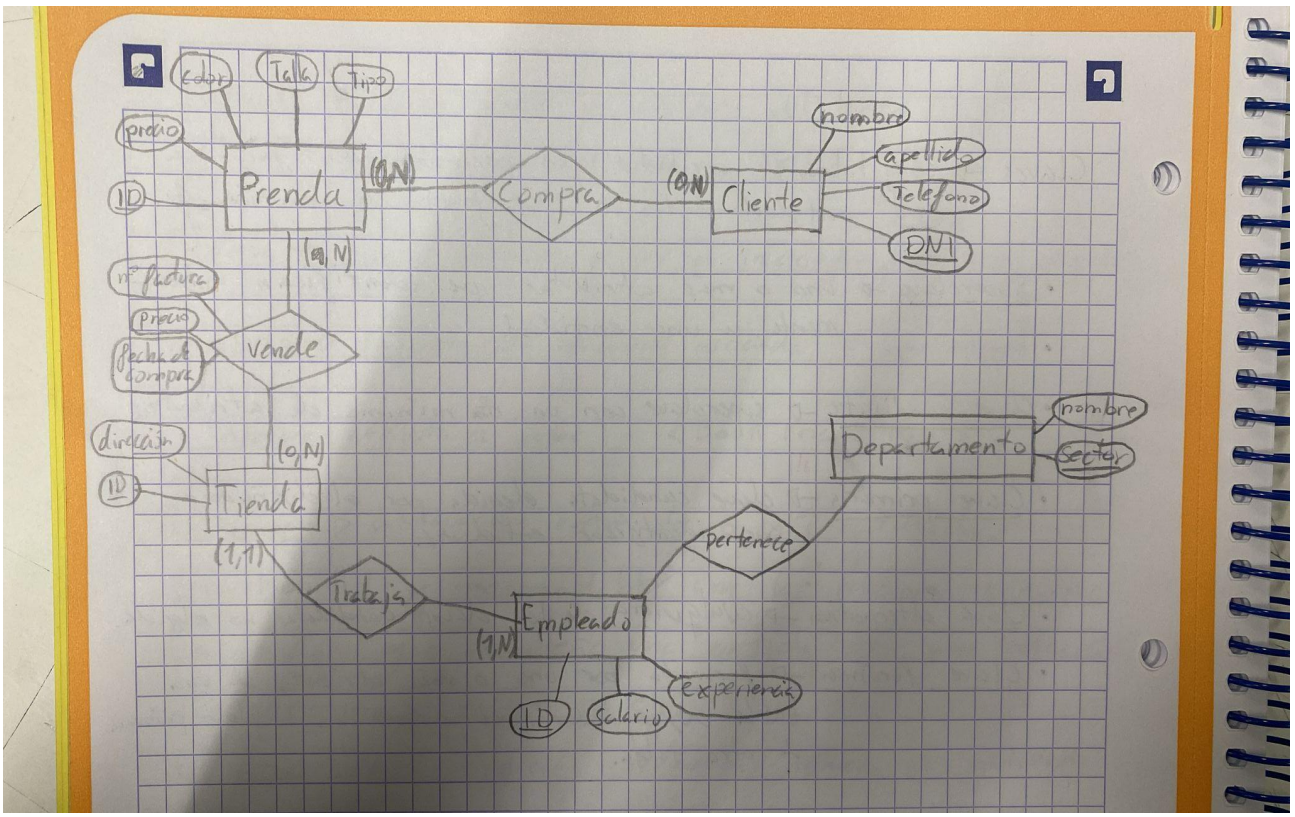
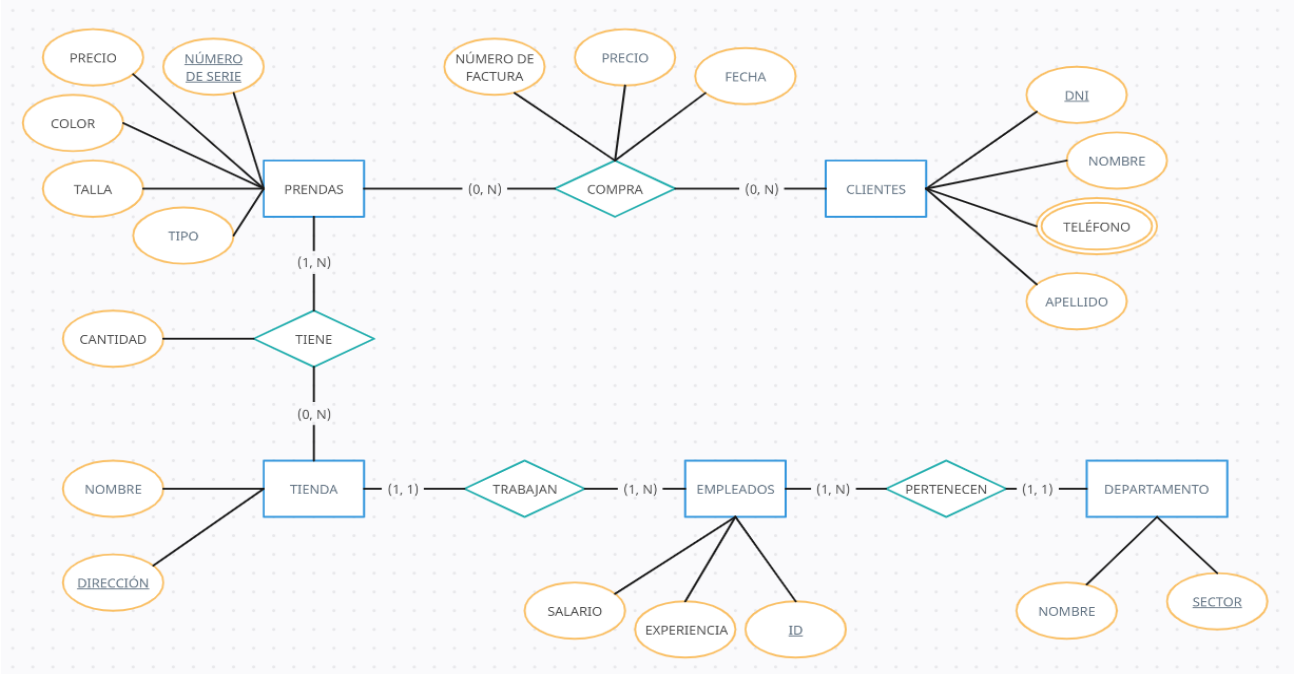


Diagrama corregido tras presentarlo en clase:



c) Diseño Lógico:

- Paso a tablas:

Hemos tenido que realizar unos pocos cambios.
El primero es debido a que coincidía que muchos de los atributos de las entidades se llamaban ID, esto no es erróneo como tal, pero para que se vean de manera más clara las claves primarias de cada una de estas entidades hemos decidido cambiarlo.
También hemos cambiado la relación “vende” por “tiene” ya que no tenía mucho sentido que tienda venda prendas y que las prendas las compran los clientes ya que viene a ser lo mismo. Los atributos de “vende” ahora pasan a ser los de “compra”.
Por último, hemos añadido las cardinalidades que faltaban en el primer diagrama entidad-relación.

Prenda	(NumSerie, precio, color, talla, tipo)
Tienda	(Dirección, nombre)
Cliente	(DNI, nombre, apellido, telefono)
Empleado	(ID, salario, experiencia, **Direccion_tienda, Sector_departamento**)
Departamento	(Sector, nombre)
Prenda_cliente	(**NumSerie_prenda, DNI_cliente**, numFactura, precio, fechaDeCompra)
Tienda_prenda	(**Dirección_tienda, numSerie_prenda**, cantidad)

d) Diseño Físico:

Hemos creado las tablas utilizando los tipos de datos TEXT para almacenar solo texto, INTEGER que almacena números, VARCHAR que sirve para almacenar datos alfanuméricos (tanto números como letras), DATE para almacenar fechas y DECIMAL el cual nos permite añadir valores con números decimales.
Cada tabla cuenta con su clave primaria y aquellos que estén relacionados con otras tablas con su debida clave foránea.
También hemos añadido para cada clave foránea la sentencia ON DELETE CASCADE la cual se utiliza para que si una clave es borrada o modificada también se actualice dicha acción en las claves ajenas que están en las otras tablas

```
CREATE TABLE prenda (
        numSerie INTEGER PRIMARY KEY AUTOINCREMENT,
        precio DECIMAL(10,2),
        color TEXT,
        talla VARCHAR,
        tipo TEXT);
```

```
CREATE TABLE tienda (
                direccion VARCHAR PRIMARY KEY,
                nombre TEXT);

CREATE TABLE cliente (
                dni VARCHAR PRIMARY KEY,
                nombre TEXT,
                apellido TEXT,
                telefono INTEGER);

CREATE TABLE departamento (
                    sector TEXT PRIMARY KEY,
                    nombre TEXT);

CREATE TABLE empleado (
                id INTEGER PRIMARY KEY,
                salario DECIMAL(10,2),
                experiencia TEXT,
                direccion_tienda VARCHAR,
                sector_departamento TEXT,
                FOREIGN KEY (direccion_tienda) REFERENCES tienda(direccion) ON
                DELETE CASCADE,
                FOREIGN KEY (sector_departamento) REFERENCES
                departamento(sector) ON DELETE CASCADE);

CREATE TABLE prenda_cliente (numSerie_prenda INTEGER,
                    dni_cliente VARCHAR,
                    numFactura INTEGER,
                    precio DECIMAL(10,2),
                    fechaDeCompra DATE,
                    PRIMARY KEY (numSerie_prenda, dni_cliente),
                    FOREIGN KEY (numSerie_prenda) REFERENCES prenda(numSerie)
ON DELETE CASCADE,
                    FOREIGN KEY (dni_cliente) REFERENCES cliente(dni) ON DELETE
CASCADE);

CREATE TABLE tienda_prenda (direccion_tienda VARCHAR,
                    numSerie_prenda INTEGER,
                    cantidad INTEGER,
                    PRIMARY KEY (direccion_tienda, numSerie_prenda),
                    FOREIGN KEY (direccion_tienda) REFERENCES tienda(direccion) ON
DELETE CASCADE,
                    FOREIGN KEY (numSerie_prenda) REFERENCES prenda(numSerie)
ON DELETE CASCADE);
```

e) Consultas:

- Creación de vistas para utilizarlas posteriormente en las consultas:

```
sqlite> CREATE VIEW view_empleados_baratos AS SELECT *
   ...> FROM empleado
   ...> WHERE salario < 1300.00;
```

```
sqlite> CREATE VIEW view_ventas_por_cliente AS SELECT prenda_cliente.dni_cliente, prenda_cliente.numSeri
e_prenda,
   ...> prenda_cliente.fechaDeCompra, cliente.nombre, cliente.apellido
   ...> FROM prenda_cliente
   ...> JOIN cliente
   ...> ON prenda_cliente.dni_cliente = cliente.dni;
```

- Utilización de sentencia de actualización/modificación en 3 consultas:

En esta primera consulta lo que pretendemos es cambiar el precio de una prenda que tenga un número de serie determinado por lo que utilizaremos un UPDATE para seleccionar la tabla en la que queremos modificar la información, un SET para indicar lo que queremos cambiar y por último añadiremos un WHERE para seleccionar el parámetro que queremos.

```
sqlite> UPDATE prenda SET precio = 17.50 WHERE numSerie = 2856;
sqlite> select * from prenda;
numSerie  precio  color     talla  tipo
--------  ------  --------  -----  ----------
1947      25.99   verde     2XL    chaqueta
2756      35.95   verde     M      calcetines
2856      17.5    rojo      XL     camiseta
3846      15.35   morado    2XL    chaqueta
3856      9.99    amarillo  S      camiseta
4757      60.91   azul      L      pantalon
7574      80.28   amarillo  L      camiseta
9274      40.99   azul      M      pantalon
9285      25.85   negro     XL     pantalon
9375      76.97   rojo      L      chaqueta
```

En la segunda consulta queremos subirle el sueldo a los empleados, por lo que utilizaremos un UPDATE para seleccionar la tabla donde está la información que queremos modificar, un SET para cambiar el contenido que queramos y por último usaremos un WHERE para añadir un parámetro para saber que fila y que columna cambiar.

```
sqlite> UPDATE empleado SET salario = salario + 100 WHERE sector_departamento = 'cinco';
sqlite> select * from empleado;
id     salario  experiencia  direccion_tienda  sector_departamento
-----  -------  -----------  ----------------  -------------------
20947  1200.37  alta         catellana 42      cuatro
28467  998.98   baja         catellana 42      cuatro
28826  1223.67  media        gran via 4        cinco
36575  1200.37  media        fuencarral 23     cinco
47264  1700     muy alta     gran via 4        cinco
73648  800.38   baja         gran via 4        tres
76937  1200.28  alta         fuencarral 23     cuatro
83850  903.37   baja         catellana 42      cuatro
86736  1284.28  alta         fuencarral 23     tres
94764  1500.25  muy alta     gran via 4        tres
```

Para finalizar con las sentencias de actualización/modificación haremos una consulta donde modificaremos la cantidad de prendas que tiene un tienda ubicada en una dirección específica para que el stock sea mayor de lo que ya era.

```
sqlite> UPDATE tienda_prenda SET cantidad = 76 WHERE direccion_tienda = 'castellana 42';
sqlite> select * from tienda_prenda;
direccion_tienda  numSerie_prenda  cantidad
----------------  ---------------  --------
gran via 4        1947             3
fuencarral 23     9274             2
castellana 42     2856             76
castellana 42     2756             76
gran via 4        3846             248
fuencarral 23     3856             293
castellana 42     4757             76
gran via 4        7574             169
fuencarral 23     9375             285
castellana 42     9285             76
```

- Utilización de 3 sentencias de consultas involucrando a más de una tabla:

En este primer caso queremos ver una tabla donde aparezcan el id y el salario de los empleados, en que tienda trabajan y el nombre del sector al que pertenecen.

```
sqlite> SELECT e.id, e.salario, t.nombre AS nombre_tienda, d.nombre AS nombre_sector
   ...> FROM empleado AS e
   ...> JOIN tienda AS t ON e.direccion_tienda = t.direccion
   ...> JOIN departamento AS d ON e.sector_departamento = d.sector;
id     salario  nombre_tienda  nombre_sector
-----  -------  -------------  -------------------
28826  1223.67  shoefaceGV     atencion al cliente
36575  1200.37  shoefaceF      atencion al cliente
47264  1700     shoefaceGV     atencion al cliente
73648  800.38   shoefaceGV     contabilidad
86736  1284.28  shoefaceF      contabilidad
94764  1500.25  shoefaceGV     contabilidad
```

Calcula el gasto promedio por cliente y muestra solo aquellos cuyo gasto promedio sea superior a 200

```
sqlite> SELECT pc.dni_cliente, AVG(pc.precio) AS gasto_promedio
   ...> FROM prenda_cliente AS pc
   ...> GROUP BY pc.dni_cliente HAVING AVG(pc.precio) > 200;
dni_cliente  gasto_promedio
-----------  --------------
12123142R    285.36
35245234X    1784.37
35423565H    496.28
35432344L    684.19
41353466P    1783.66
53734234N    396.59
86285638A    274.27
```

Suma los salarios de los empleados por departamento limitando los resultados a los dos primeros departamentos.

```
sqlite> SELECT d.nombre AS departamento, SUM(e.salario) AS total_salarios
   ...> FROM empleado AS e
   ...> JOIN departamento AS d ON e.sector_departamento = d.sector GROUP BY d.nombre LIMIT 2;
departamento         total_salarios
-------------------  --------------
atencion al cliente  4124.04
contabilidad         3584.91
```

Consultas Subordinadas:

Esta consulta muestra todas las tiendas que ha vendido la prenda con el número de serie 1947

```
sqlite> SELECT * FROM tienda WHERE direccion IN(SELECT direccion_tienda FROM tienda_prenda WHERE numSerie_prenda = 1947);
direccion   nombre
----------  ----------
gran via 4  shoefaceGV
```

Muestra los datos del empleado cuyo salario sea el salario máximo registrado en la tabla.

```
sqlite> SELECT * FROM empleado WHERE salario = (SELECT MAX(salario) FROM empleado);
id     salario  experiencia  direccion_tienda  sector_departamento
-----  -------  -----------  ----------------  -------------------
47264  1700     muy alta     gran via 4        cinco
```

Algunas de las consultas utilizan funciones de agregación diferentes como MAX(), AVG() y SUM()

Consultas de Borrado:
Borra al cliente cuyo dni es igual a 12123142R

dni	nombre	apellido	telefono
18462846S	Maximo	Dutti	693739573
38563967T	Margaret	Astor	638569476
86285638A	Emidio	Tucci	674658375
12123142R	Gonzalo	Lopez	645632834
35423565H	Pablo	Hernandez	612343235
35432344L	Alejandro	Lillo	676432133
53734234N	Daniel	Candela	697421456
35264653M	Javier	Colina	676545334
35245234X	Tomas	Garcia	665534323
41353466P	Fernando	Sanchez	643543332

```
sqlite> DELETE FROM cliente WHERE dni = '12123142R';
sqlite> select * from cliente;
dni         nombre      apellido    telefono
----------  ----------  ----------  ----------
18462846S   Maximo      Dutti       693739573
38563967T   Margaret    Astor       638569476
86285638A   Emidio      Tucci       674658375
35423565H   Pablo       Hernandez   612343235
35432344L   Alejandro   Lillo       676432133
53734234N   Daniel      Candela     697421456
35264653M   Javier      Colina      676545334
35245234X   Tomas       Garcia      665534323
41353466P   Fernando    Sanchez     643543332
```

Elimina aquellos empleados que tienen una experiencia baja y a su vez su salario es menor a 950 euros.

```
sqlite> select * from empleado;
id     salario  experiencia  direccion_tienda  sector_departamento
-----  -------  -----------  ----------------  -------------------
20947  1200.37  alta         catellana 42      cuatro
28467  998.98   baja         catellana 42      cuatro
28826  1223.67  media        gran via 4        cinco
36575  1200.37  media        fuencarral 23     cinco
47264  1700     muy alta     gran via 4        cinco
73648  800.38   baja         gran via 4        tres
76937  1200.28  alta         fuencarral 23     cuatro
83850  903.37   baja         catellana 42      cuatro
86736  1284.28  alta         fuencarral 23     tres
94764  1500.25  muy alta     gran via 4        tres
sqlite> DELETE FROM empleado WHERE experiencia = 'baja' AND salario < 950;
sqlite> select * from empleado;
id     salario  experiencia  direccion_tienda  sector_departamento
-----  -------  -----------  ----------------  -------------------
20947  1200.37  alta         catellana 42      cuatro
28467  998.98   baja         catellana 42      cuatro
28826  1223.67  media        gran via 4        cinco
36575  1200.37  media        fuencarral 23     cinco
47264  1700     muy alta     gran via 4        cinco
76937  1200.28  alta         fuencarral 23     cuatro
86736  1284.28  alta         fuencarral 23     tres
94764  1500.25  muy alta     gran via 4        tres
```

Borra aquellas tiendas las cuales tienen una cantidad de prendas inferior a la media de todas las prendas de las tiendas.

```
direccion_tienda  numSerie_prenda  cantidad
----------------  ---------------  --------
gran via 4        1947             3
fuencarral 23     9274             2
castellana 42     2856             5
castellana 42     2756             100
gran via 4        3846             248
fuencarral 23     3856             293
castellana 42     4757             183
gran via 4        7574             169
fuencarral 23     9375             285
castellana 42     9285             937
```

```
sqlite> DELETE FROM tienda_prenda WHERE cantidad < (SELECT AVG(cantidad) FROM tienda_prenda WHERE direccion_tienda = 'gran via 4');
sqlite> select * from tienda_prenda;
direccion_tienda  numSerie_prenda  cantidad
----------------  ---------------  --------
gran via 4        3846             248
fuencarral 23     3856             293
gran via 4        7574             169
fuencarral 23     9375             285
```