

PROYECTO FINAL

REALIZAR E IMPLEMENTAR UNA BASE DE DATOS

Lucía González Martín
BASES DE DATOS | 1º DAW

Índice de contenidos

1) Enunciado del caso real objeto de la base de datos	2
2) Modelo entidad – relación	4
3) Modelo relacional	5
Normalización	6
4) Implementación de la base de datos.....	10
5) Consultas	18
Consultas multitable.....	18
Consultas de resumen de datos	21
Subconsultas	23
6) Rutinas.....	25
Procedimientos almacenados	25
Funciones almacenadas	31
Triggers.....	35

1) Enunciado del caso real objeto de la base de datos

Una cadena de supermercados decide informatizar la gestión de sus pedidos y de las nóminas de sus empleados.

- De cada uno de los supermercados se guarda: CIF (obligatorio y único), dirección (formada por calle, número, código postal y población) y teléfono. Cada supermercado realiza uno o varios pedidos y cada pedido es realizado por un único supermercado.
- Para cada pedido: Cada pedido se identifica por un código y, también, se guarda la fecha en la que se realizó. Un pedido tiene varias líneas y cada línea solo pertenece a un pedido. Las líneas se identifican por el código del pedido y el número de línea. Cada línea consta de un producto y cada producto sólo aparece en una línea. Además, se guarda la cantidad que se pide de cada producto.

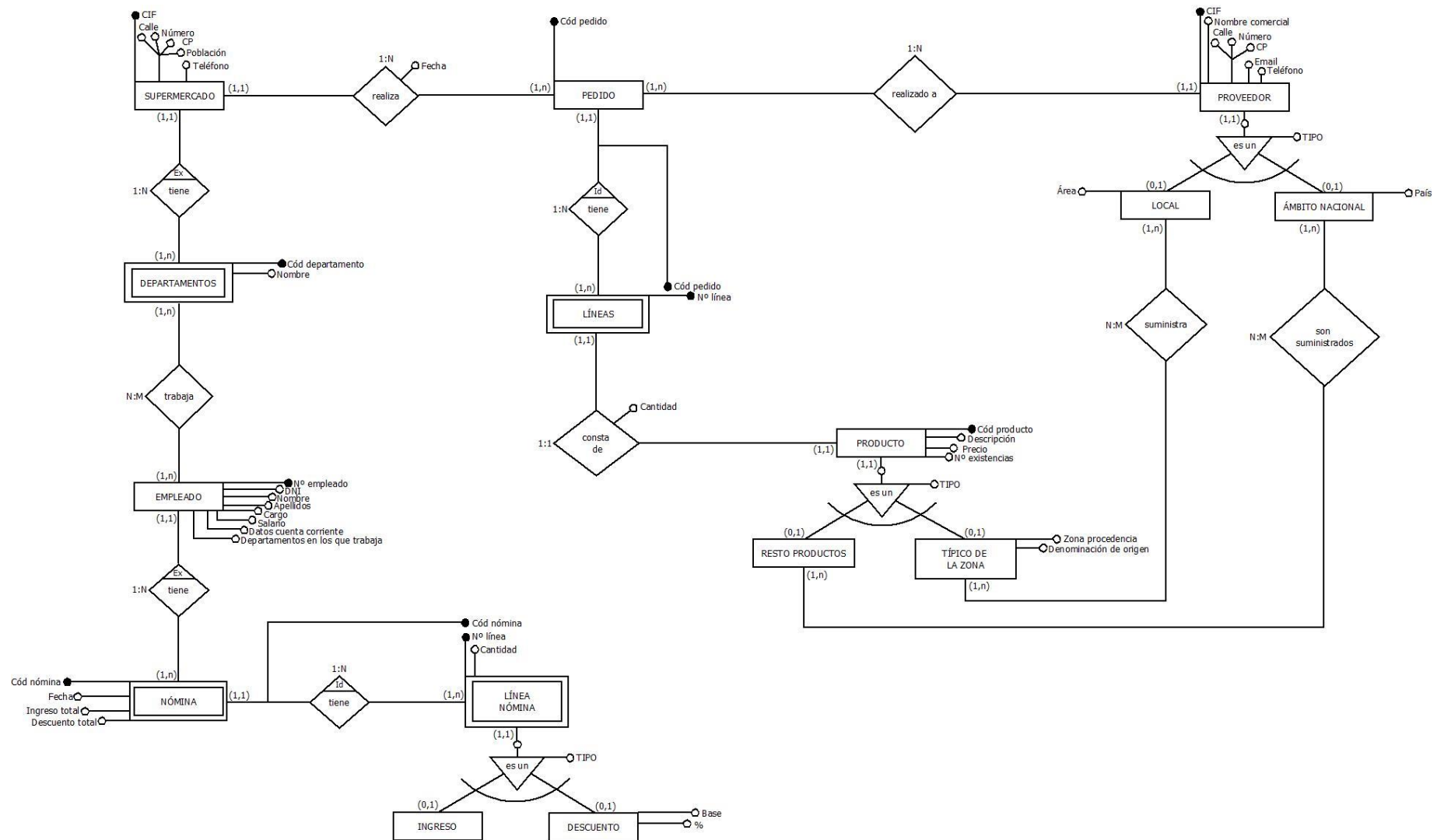
Cada pedido sólo puede ser realizado a un único proveedor y un proveedor puede recibir uno o varios pedidos.

- Para cada proveedor: CIF (obligatorio y único), nombre comercial, dirección (formada por calle, número y código postal), email y teléfono. Los proveedores pueden ser locales o de ámbito nacional. De los proveedores locales guardamos el área donde se encuentran y de los proveedores nacionales guardamos el país. Los proveedores locales suministran productos típicos de cada zona y los de ámbito nacional suministran el resto de productos. Cada proveedor local o nacional puede suministrar varios productos y cada producto puede ser suministrado por uno o varios proveedores.
- De cada producto se desea guardar un código, una descripción, el precio y el número de existencias. Los productos pueden ser típicos de cada zona o no. De los productos típicos se guarda la zona de procedencia y si tienen denominación de origen.
- Los supermercados pueden tener uno o varios departamentos y cada departamento está asociado a un único supermercado. De cada departamento se guarda su código y su nombre.
- A cada empleado se le asignará un número de empleado que será único. Además, se guarda su DNI, nombre, apellidos, cargo que ocupa en la empresa, salario, datos de su cuenta corriente y los departamentos en los que trabaja. Cada empleado puede trabajar en varios departamentos y en cada departamento trabajan varios empleados.

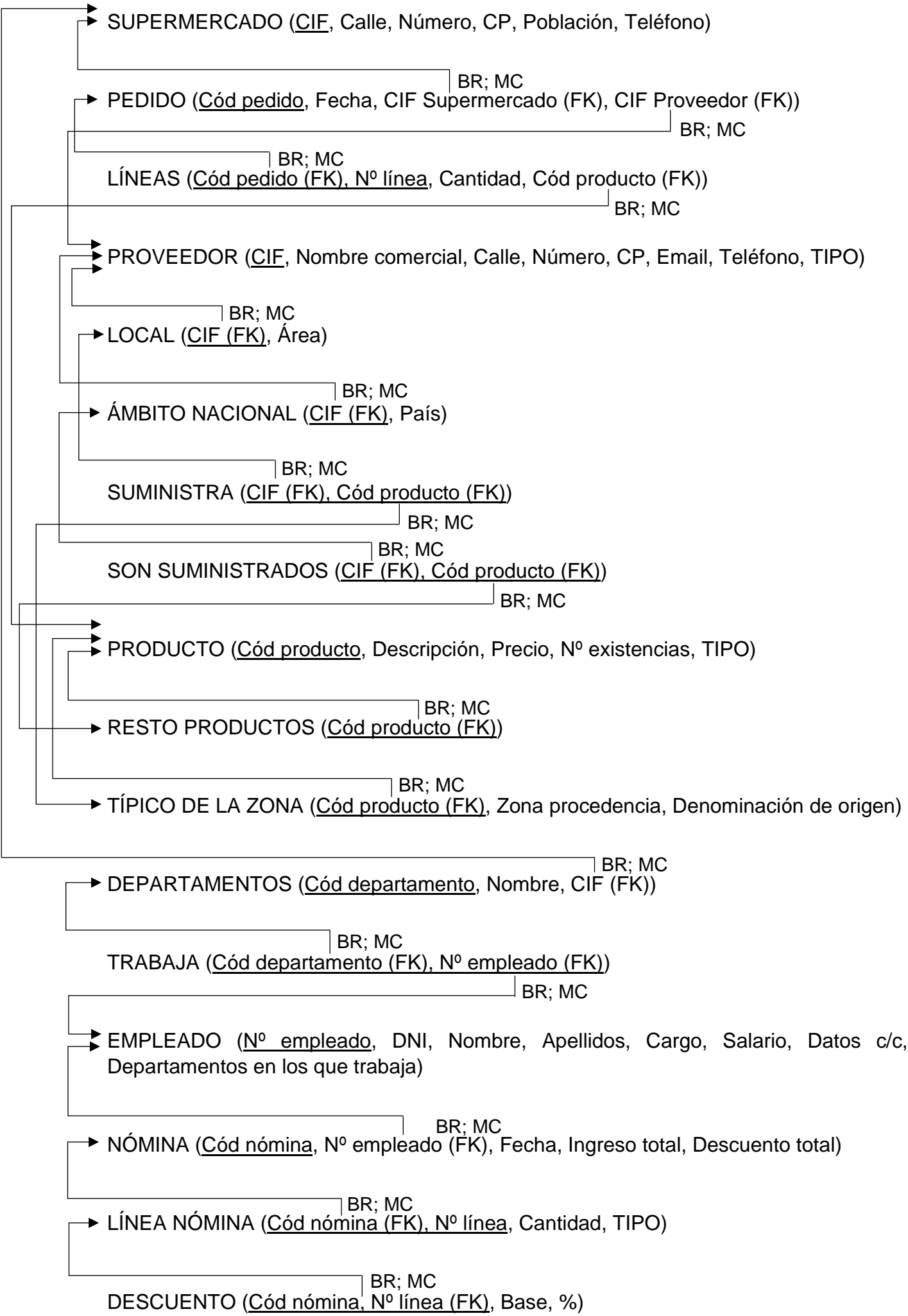
- Los empleados reciben una serie de nóminas (al menos una mensualmente). De las nóminas se guarda el ingreso total percibido por el empleado y el descuento total aplicado. Además, guardaremos la fecha. La distinción entre las nóminas se hará mediante un código que será único para cada nómina.
- Cada nómina consta de varias líneas. Las líneas pueden ser de ingreso o de descuento. En ambos casos, se recoge la cantidad que corresponde a la línea. Si se trata de un descuento se guarda la base sobre la que se aplica y el porcentaje que se aplica.

Cada línea se identifica con el código de la nómina y el número de línea.

2) Modelo entidad – relación



3) Modelo relacional



Normalización

SUPERMERCADO (CIF, Calle, Número, CP, Población, Teléfono)

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN? Sí, la clave primaria es siempre, por lo que, todos los atributos dependen completamente de ella.

¿Está en 3 FN? No, hay DF transitivas.

$$\left. \begin{array}{l} \text{CIF} \rightarrow \text{CP} \\ \text{CP} \rightarrow \text{Población} \end{array} \right\} \text{CIF} \rightarrow \text{Población}$$

SUPERMERCADO' (CIF, Calle, Número, Teléfono, CP (FK))

POBLACIONES_SUPERMERCADOS (CP, Población)

PEDIDO (Cód pedido, Fecha, CIF Supermercado (FK), CIF Proveedor (FK))

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN? Sí, la clave primaria es siempre, por tanto, todos los atributos dependen completamente de ella.

¿Está en 3 FN? Sí, no hay DF transitivas.

LÍNEAS (Cód pedido (FK), N° línea, Cantidad, Cód producto (FK))

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN? Sí, todos los atributos dependen completamente de la clave primaria formada por cód pedido y n° línea.

¿Está en 3 FN? Sí, no hay DF transitivas.

PROVEEDOR (CIF, Nombre comercial, Calle, Número, CP, Email, Teléfono, TIPO)

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN? Sí, la clave primaria es siempre, por tanto, todos los atributos dependen completamente de ella.

¿Está en 3 FN? Sí, no hay DF transitivas.

LOCAL (CIF (FK), Área)

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN? Sí, la clave primaria es siempre, por tanto, todos los atributos dependen completamente de ella.

¿Está en 3 FN? Sí, no hay DF transitivas.

ÁMBITO NACIONAL (CIF (FK), País)

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN? Sí, la clave primaria es siempre, por tanto, todos los atributos dependen completamente de ella.

¿Está en 3 FN? Sí, no hay DF transitivas.

SUMINISTRA (CIF (FK), Cód producto (FK))

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN y 3 FN? Sí, no hay atributos que no formen parte de la clave, por ello, no hay posibilidad de que existan DF no completas o transitivas.

SON SUMINISTRADOS (CIF (FK), Cód producto (FK))

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN y 3 FN? Sí, no hay atributos que no formen parte de la clave, por ello, no hay posibilidad de que existan DF no completas o transitivas.

PRODUCTO (Cód producto, Descripción, Precio, N° existencias, TIPO)

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN? Sí, la clave primaria es siempre, por tanto, todos los atributos dependen completamente de ella.

¿Está en 3 FN? Sí, no hay DF transitivas.

RESTO PRODUCTOS (Cód producto (FK))

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN y 3 FN? Sí, no hay atributos que no formen parte de la clave, por ello, no hay posibilidad de que existan DF no completas o transitivas.

TÍPICO DE LA ZONA (Cód producto (FK), Zona procedencia, Denominación de origen)

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN? Sí, la clave primaria es siempre, por tanto, todos los atributos dependen completamente de ella.

¿Está en 3 FN? Sí, no hay DF transitivas.

DEPARTAMENTOS (Cód departamento, Nombre, CIF (FK))

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN? Sí, la clave primaria es siempre, por tanto, todos los atributos dependen completamente de ella.

¿Está en 3 FN? Sí, no hay DF transitivas.

TRABAJA (Cód departamento (FK), Nº empleado (FK))

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN y 3 FN? Sí, no hay atributos que no formen parte de la clave, por ello, no hay posibilidad de que existan DF no completas o transitivas.

EMPLEADO (Nº empleado, DNI, Nombre, Apellidos, Cargo, Salario, Datos c/c, Departamentos en los que trabaja)

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN? Sí, la clave primaria es siempre, por tanto, todos los atributos dependen completamente de ella.

¿Está en 3 FN? Sí, no hay DF transitivas.

NÓMINA (Cód nómina, Nº empleado (FK), Fecha, Ingreso total, Descuento total)

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN? Sí, todos los atributos dependen completamente de la clave primaria.

¿Está en 3 FN? Sí, no hay DF transitivas.

LÍNEA NÓMINA (Cód nómina (FK), N° línea, Cantidad, TIPO)

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN? Sí, todos los atributos dependen completamente de la clave primaria.

¿Está en 3 FN? Sí, no hay DF transitivas.

DESCUENTO (Cód nómina, N° línea (FK), Base, %)

¿Está en 1 FN? Sí, no hay atributos multivaluados.

¿Está en 2 FN? Sí, todos los atributos dependen completamente de la clave primaria.

¿Está en 3 FN? Sí, no hay DF transitivas.

4) Implementación de la base de datos

```
drop database if exists CADENA_SUPERMERCADOS;  
create database CADENA_SUPERMERCADOS collate utf8mb4_bin;  
use CADENA_SUPERMERCADOS;
```

```
create table SUPERMERCADO  
(  
    CIF char(9) primary key,  
    CALLE varchar(30) not null,  
    NUMERO varchar(10) not null,  
    TELEFONO varchar(12) unique not null,  
    CP int not null  
);
```

```
create table POBLACIONES_SUPERMERCADOS  
(  
    CP int primary key,  
    POBLACION varchar(30) not null  
);
```

```
create table PEDIDO  
(  
    COD_PEDIDO int primary key,  
    FECHA date not null,  
    CIF_SUPERMERCADO char(9) not null,  
    CIF_PROVEEDOR char(9) not null  
);
```

```
create table LINEAS
(
  COD_PEDIDO int not null,
  NUM_LINEA int not null,
  CANTIDAD int not null,
  COD_PRODUCTO int not null,
  constraint LIN_PK primary key (COD_PEDIDO, NUM_LINEA)
);
```

```
create table PROVEEDOR
(
  CIF char(9) primary key,
  NOM_COMERCIAL varchar(50) not null,
  CALLE varchar(50) not null,
  NUMERO varchar(10) not null,
  CP int not null,
  EMAIL varchar(50),
  TELEFONO char(9) unique not null,
  TIPO varchar(15) not null check (TIPO in ('Local', 'Ámbito
nacional'))
);
```

```
create table LOCAL
(
  CIF char(9) primary key,
  AREA varchar(50) not null
);
```

```
create table AMBITO_NACIONAL
```

```
(
```

```
CIF char(9) primary key,
```

```
PAIS varchar(50) not null
```

```
);
```

```
create table SUMINISTRA
```

```
(
```

```
CIF char(9) not null,
```

```
COD_PRODUCTO int not null,
```

```
constraint SUM_PK primary key (CIF, COD_PRODUCTO)
```

```
);
```

```
create table SON_SUMINISTRADOS
```

```
(
```

```
CIF char(9) not null,
```

```
COD_PRODUCTO int not null,
```

```
constraint SONSU_PK primary key (CIF, COD_PRODUCTO)
```

```
);
```

```
create table PRODUCTO
```

```
(
```

```
COD_PRODUCTO int primary key,
```

```
DESCRIPCION varchar(300),
```

```
PRECIO decimal(6,2) not null,
```

```
NUM_EXISTENCIAS int not null,
```

```
TIPO varchar(20) not null check (TIPO in ('Resto productos',  
'Típico zona'))
```

```
);
```

```
create table RESTO_PRODUCTOS
(
COD_PRODUCTO int primary key
);
```

```
create table TIPICO_ZONA
(
COD_PRODUCTO int primary key,
ZONA_PROCEDENCIA varchar(50) not null,
DENOMINACION_ORIGEN char(2) not null check (DENOMINACION_ORIGEN
in ('Si', 'No'))
);
```

```
create table DEPARTAMENTOS
(
COD_DEPARTAMENTO int primary key,
NOMBRE varchar(50) not null,
CIF char(9) not null
);
```

```
create table TRABAJA
(
COD_DEPARTAMENTO int not null,
NUM_EMPLEADO int not null,
constraint TRA_PK primary key (COD_DEPARTAMENTO, NUM_EMPLEADO)
);
```

```
create table EMPLEADO
(
  NUM_EMPLEADO int primary key,
  DNI char(9) unique not null,
  NOMBRE varchar(20) not null,
  APELLIDOS varchar(50) not null,
  CARGO varchar(20) not null,
  SALARIO_NETO decimal(8,2) not null,
  DATOS_CUENTA char(30) unique not null,
  DEPARTAMENTOS_TRABAJA varchar(50) not null
);
```

```
create table NOMINA
(
  COD_NOMINA int primary key,
  NUM_EMPLEADO int not null,
  FECHA date not null,
  INGRESO_TOTAL decimal(8,2) not null,
  DESCUENTO_TOTAL decimal(6,2) not null
);
```

```
create table LINEA_NOMINA
(
  COD_NOMINA int not null,
  NUM_LINEA int not null,
  CANTIDAD decimal(8,2) not null,
  TIPO varchar(10) check (TIPO in ('Ingreso', 'Descuento')),
  constraint LINEANOM_PK primary key (COD_NOMINA, NUM_LINEA)
);
```

```
create table DESCUENTO
(
  COD_NOMINA int not null,
  NUM_LINEA int not null,
  BASE decimal(6,2) not null,
  PORCENTAJE decimal(4,2) not null,
  constraint DES_PK primary key (COD_NOMINA, NUM_LINEA)
);
```

```
-- CLAVES FORÁNEAS --
```

```
alter table SUPERMERCADO add constraint CP_FK foreign key (CP)
references POBLACIONES_SUPERMERCADOS (CP) on delete restrict on
update cascade;
```

```
alter table PEDIDO add constraint SUP_FK foreign key
(CIF_SUPERMERCADO) references SUPERMERCADO (CIF) on delete
restrict on update cascade, add constraint PRO_FK foreign key
(CIF_PROVEEDOR) references PROVEEDOR (CIF) on delete restrict
on update cascade;
```

```
alter table LINEAS add constraint PED_FK foreign key
(COD_PEDIDO) references PEDIDO (COD_PEDIDO) on delete restrict
on update cascade, add constraint PROD_FK foreign key
(COD_PRODUCTO) references PRODUCTO (COD_PRODUCTO) on delete
restrict on update cascade;
```

```
alter table LOCAL add constraint PROV_FK foreign key (CIF)
references PROVEEDOR (CIF) on delete restrict on update cascade;
```



```
alter table AMBITO_NACIONAL add constraint PROVE_FK foreign key  
(CIF) references PROVEEDOR (CIF) on delete restrict on update  
cascade;
```

```
alter table SUMINISTRA add constraint LOC_FK foreign key (CIF)  
references LOCAL (CIF) on delete restrict on update cascade, add  
constraint PRODU_FK foreign key (COD_PRODUCTO) references  
TIPICO_ZONA (COD_PRODUCTO) on delete restrict on update cascade;
```

```
alter table SON_SUMINISTRADOS add constraint NAC_FK foreign key  
(CIF) references AMBITO_NACIONAL (CIF) on delete restrict on  
update cascade, add constraint PRODUC_FK foreign key  
(COD_PRODUCTO) references RESTO_PRODUCTOS (COD_PRODUCTO) on  
delete restrict on update cascade;
```

```
alter table RESTO_PRODUCTOS add constraint PRODUCT_FK foreign  
key (COD_PRODUCTO) references PRODUCTO (COD_PRODUCTO) on delete  
restrict on update cascade;
```

```
alter table TIPICO_ZONA add constraint PRODUCTO_FK foreign key  
(COD_PRODUCTO) references PRODUCTO (COD_PRODUCTO) on delete  
restrict on update cascade;
```

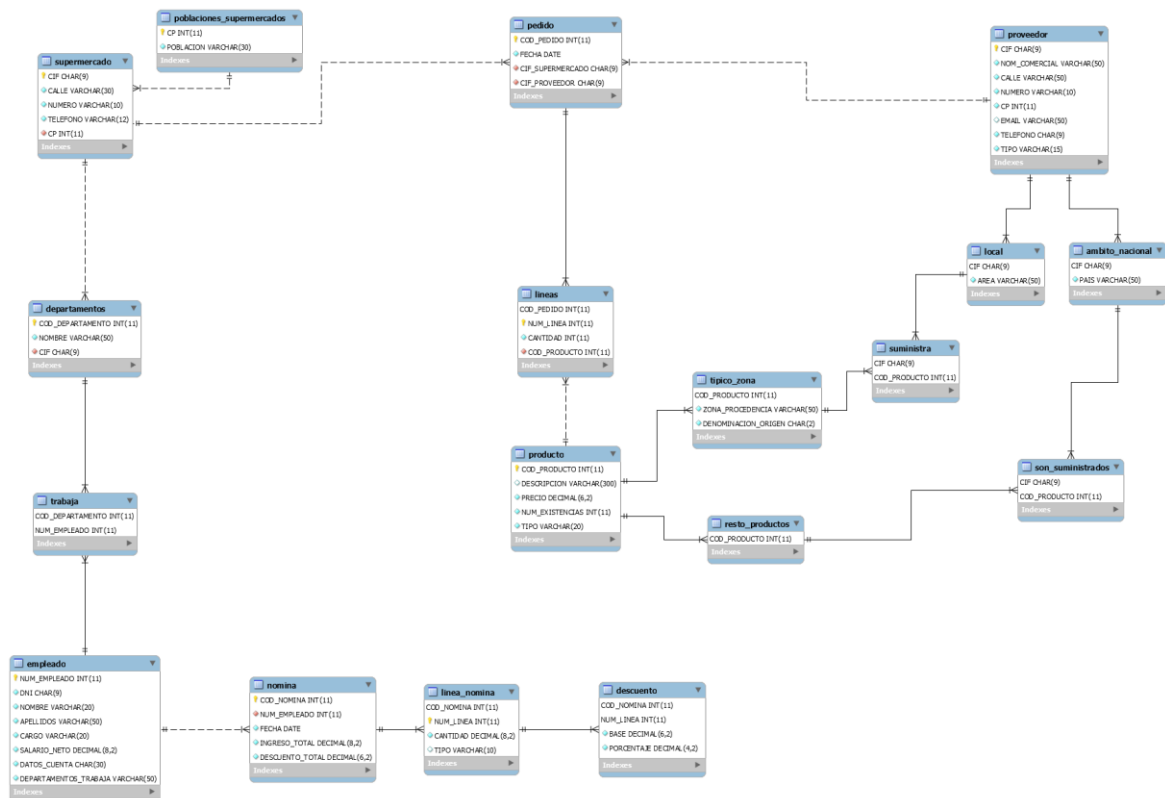
```
alter table DEPARTAMENTOS add constraint SUPE_FK foreign key  
(CIF) references SUPERMERCADO (CIF) on delete restrict on update  
cascade;
```

```
alter table TRABAJA add constraint DEP_FK foreign key  
(COD_DEPARTAMENTO) references DEPARTAMENTOS (COD_DEPARTAMENTO)  
on delete restrict on update cascade, add constraint EMP_FK  
foreign key (NUM_EMPLEADO) references EMPLEADO (NUM_EMPLEADO) on  
delete restrict on update cascade;
```

```
alter table NOMINA add constraint EMPL_FK foreign key  
(NUM_EMPLEADO) references EMPLEADO (NUM_EMPLEADO) on delete  
restrict on update cascade;
```

```
alter table LINEA_NOMINA add constraint NOM_FK foreign key
(COD_NOMINA) references NOMINA (COD_NOMINA) on delete restrict
on update cascade;
```

```
alter table DESCUENTO add constraint NOMI_FK foreign key
(COD_NOMINA, NUM_LINEA) references LINEA_NOMINA (COD_NOMINA,
NUM_LINEA) on delete restrict on update cascade;
```

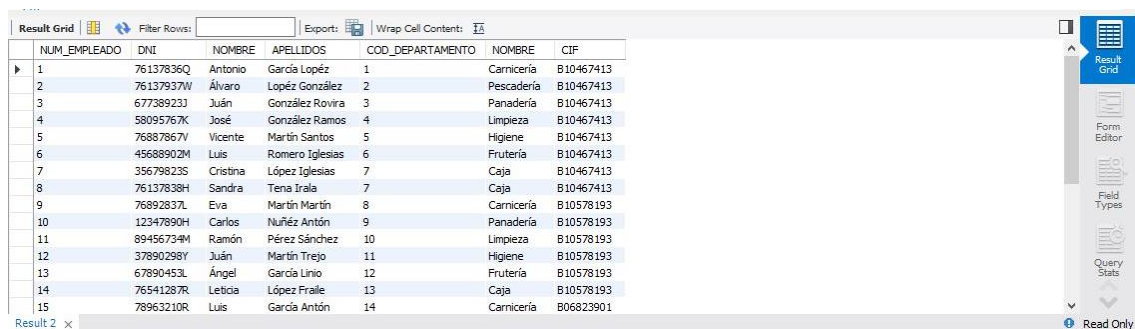


5) Consultas

Consultas multitable

1. Listar el NUM_EMPLEADO, DNI, NOMBRE y APELLIDOS de los empleados y, para cada uno de ellos, los datos del departamento donde trabaja, ordenando el resultado por NUM_EMPLEADO ascendente y además por CIF, también ascendente.

```
select EMPLEADO.NUM_EMPLEADO, EMPLEADO.DNI, EMPLEADO.NOMBRE,
EMPLEADO.APELLIDOS, DEPARTAMENTOS.* from EMPLEADO inner join
TRABAJA using(NUM_EMPLEADO) inner join DEPARTAMENTOS
using(COD_DEPARTAMENTO) order by EMPLEADO.NUM_EMPLEADO,
DEPARTAMENTOS.CIF;
```



The screenshot shows a database query result grid with 15 rows and 7 columns. The columns are: NUM_EMPLEADO, DNI, NOMBRE, APELLIDOS, COD_DEPARTAMENTO, NOMBRE, and CIF. The data is sorted by NUM_EMPLEADO and then by CIF. The departments are: Carnicería, Pescadería, Panadería, Limpieza, Higiene, Frutería, and Caja.

NUM_EMPLEADO	DNI	NOMBRE	APELLIDOS	COD_DEPARTAMENTO	NOMBRE	CIF
1	76137836Q	Antonio	García López	1	Carnicería	B10467413
2	76137937W	Álvaro	López González	2	Pescadería	B10467413
3	67738923J	Juán	González Rovira	3	Panadería	B10467413
4	58095767K	José	González Ramos	4	Limpieza	B10467413
5	76887867V	Vicente	Martín Santos	5	Higiene	B10467413
6	45688902M	Luis	Romero Iglesias	6	Frutería	B10467413
7	35679823S	Cristina	López Iglesias	7	Caja	B10467413
8	76137838H	Sandra	Tena Irala	7	Caja	B10467413
9	76892837L	Eva	Martín Martín	8	Carnicería	B10578193
10	12347890H	Carlos	Núñez Antón	9	Panadería	B10578193
11	89456734M	Ramón	Pérez Sánchez	10	Limpieza	B10578193
12	37890298Y	Juán	Martín Trejo	11	Higiene	B10578193
13	67890453L	Ángel	García Lirio	12	Frutería	B10578193
14	76541287R	Leticia	López Fraile	13	Caja	B10578193
15	78963210R	Luis	García Antón	14	Carnicería	B06823901

2. Mostrar un listado del CIF y el CP de los supermercados que han realizado pedidos, indicando para cada pedido el COD_PEDIDO, la FECHA, el CIF y el NOMBRE del proveedor. Ordenando el resultado por CIF del supermercado, COD_PEDIDO y FECHA.

```
select SUPERMERCADO.CIF as "CIF SUPERMERCADO", SUPERMERCADO.CP
as "CP SUPERMERCADO", PEDIDO.COD_PEDIDO, PEDIDO.FECHA,
PROVEEDOR.CIF as "CIF PROVEEDOR", PROVEEDOR.NOM_COMERCIAL as
"NOMBRE PROVEEDOR" from SUPERMERCADO inner join PEDIDO on
SUPERMERCADO.CIF=PEDIDO.CIF_SUPERMERCADO inner join PROVEEDOR on
PEDIDO.CIF_PROVEEDOR=PROVEEDOR.CIF order by SUPERMERCADO.CIF,
PEDIDO.COD_PEDIDO, PEDIDO.FECHA;
```

CIF SUPERMERCADO	CP SUPERMERCADO	COD_PEDIDO	FECHA	CIF PROVEEDOR	NOMBRE PROVEEDOR
B06673892	6800	13	2019-10-24	B06245783	Alimentacion San Servan S.L.
B06673892	6800	14	2019-11-08	B06256784	Dehesa De Alimentacion Y Bebidas S.L
B06673892	6800	15	2019-11-24	A82676501	Nuñez Y Marcos Distribuidores S.A.
B06673892	6800	16	2019-11-10	A00676736	footloose-vintage S.A.
B06823901	6010	9	2019-10-30	B06345467	Jarole Alimentacion S.L
B06823901	6010	10	2019-11-09	B06342894	Alimentacion Servipan S.L.
B06823901	6010	11	2019-11-29	A28679491	Galsos Distribuidores Ibericos S.A.
B06823901	6010	12	2019-12-04	A82676501	Nuñez Y Marcos Distribuidores S.A.
B10467413	10002	1	2019-10-26	B10879411	Chinata De Alimentacion S.L.
B10467413	10002	2	2019-11-02	B10368120	Alimentacion Frialto S.L.
B10467413	10002	3	2019-11-30	A28679491	Galsos Distribuidores Ibericos S.A.
B10467413	10002	4	2020-01-04	A82676501	Nuñez Y Marcos Distribuidores S.A.
B10578193	10600	5	2019-10-05	B10562387	Alimentacion Lateral S.L.
B10578193	10600	6	2019-10-28	B10542690	Compañia Oleicola S Xxi S.L.
B10578193	10600	7	2019-11-07	A28679491	Galsos Distribuidores Ibericos S.A.
B10578193	10600	8	2019-11-25	A00676736	footloose-vintage S.A.

3. Listar los proveedores locales, indicando para cada uno de ellos su CIF, NOM_COMERCIAL y AREA, así como los datos de los productos que suministra, ordenando el resultado por CIF del proveedor.

```
select PROVEEDOR.CIF, PROVEEDOR.NOM_COMERCIAL, LOCAL.AREA,
PRODUCTO.* from PROVEEDOR inner join LOCAL using(CIF) inner join
SUMINISTRA using(CIF) inner join TIPICO_ZONA using(COD_PRODUCTO)
inner join PRODUCTO using(COD_PRODUCTO) order by PROVEEDOR.CIF;
```

CIF	NOM_COMERCIAL	AREA	COD_PRODUCTO	DESCRIPCION	PRECIO	NUM_EXISTENCIAS	TIPO
B06245783	Alimentacion San Servan S.L.	Mérida	20	Aceite de Oliva Virgen Extra Garrafa	5.95	32	Típico zona
B06256784	Dehesa De Alimentacion Y Bebidas S.L	Mérida	28	Mermelada artesana ESENCIA IBERICA de picot...	2.77	28	Típico zona
B06342894	Alimentacion Servipan S.L.	Badajoz	23	Miel de Azahar - La Virgen de Extremadura - 1kg	8.35	6	Típico zona
B06345467	Jarole Alimentacion S.L	Badajoz	8	Jamón de bellota ibérico 75 % raza ibérica D.O.	503.75	18	Típico zona
B10368120	Alimentacion Frialto S.L.	Aldehuela del Jerte	3	La Chinata - Pimentón de la Vera - Pimentón Ah...	6.17	6	Típico zona
B10542690	Compañia Oleicola S Xxi S.L.	Plasencia	18	Cabecera de lomo embuchado - 400/450 gr	16.45	29	Típico zona
B10542690	Compañia Oleicola S Xxi S.L.	Plasencia	26	Mermelada de cereza picota - CAMPO & TIERRA...	1.20	10	Típico zona
B10562387	Alimentacion Lateral S.L.	Plasencia	12	Cerezas caja de 2,5Kg Calibre +28	16.25	20	Típico zona
B10562387	Alimentacion Lateral S.L.	Plasencia	14	Espárragos verdes ecológicos	6.99	10	Típico zona
B10879411	Chinata De Alimentacion S.L.	Malpartida de Plasencia	6	Cañas con azúcar La Cacerña - 500g	4.25	10	Típico zona

4. Listar los proveedores nacionales, indicando para cada uno de ellos su CIF, NOM_COMERCIAL y PAIS, así como los datos de los productos que suministra, ordenando el resultado por CIF del proveedor.

```
select PROVEEDOR.CIF, PROVEEDOR.NOM_COMERCIAL,
AMBITO_NACIONAL.PAIS, PRODUCTO.* from PROVEEDOR inner join
AMBITO_NACIONAL using(CIF) inner join SON_SUMINISTRADOS
using(CIF) inner join RESTO_PRODUCTOS using(COD_PRODUCTO) inner
join PRODUCTO using(COD_PRODUCTO) order by PROVEEDOR.CIF;
```

CIF	NOM_COMERCIAL	PAIS	COD_PRODUCTO	DESCRIPCION	PRECIO	NUM_EXISTENCIAS	TIPO
A00676736	footloose-vintage S.A.	Francia	7	Vino - Ruffino Chianti Classico	7.55	50	Resto productos
A00676736	footloose-vintage S.A.	Francia	13	Tomato Puree	5.74	37	Resto productos
A00676736	footloose-vintage S.A.	Francia	17	Vino - Vouvray Cuvee Domaine	7.32	30	Resto productos
A28679491	Galsos Distribuidores Ibericos S.A.	España	1	Manzana Golden Marlene	0.49	53	Resto productos
A28679491	Galsos Distribuidores Ibericos S.A.	España	2	Pasta al Curry - Green Masala - 200g	2.71	44	Resto productos
A28679491	Galsos Distribuidores Ibericos S.A.	España	4	Gallina Blanca - Caldo Casero de Carne - 100% ...	7.70	2	Resto productos
A28679491	Galsos Distribuidores Ibericos S.A.	España	5	Knorr - Sopa De Ave Con Fideos Finos 61g - So...	3.86	13	Resto productos
A28679491	Galsos Distribuidores Ibericos S.A.	España	9	Scottex Original Papel Higiénico - 12 Rollos	7.35	48	Resto productos
A28679491	Galsos Distribuidores Ibericos S.A.	España	10	Ace WC + Tubería Gel, Fórmula Autoactiva Des...	1.75	83	Resto productos
A28679491	Galsos Distribuidores Ibericos S.A.	España	11	Brillante Arroz Integral - Pack de 2 x 125g - Tot...	1.10	91	Resto productos
A28679491	Galsos Distribuidores Ibericos S.A.	España	15	Salsa - Balsamic Viniagrette	2.02	27	Resto productos
A28679491	Galsos Distribuidores Ibericos S.A.	España	16	Bifrutas Zumos de Frutas Tropical - Pack de 6 x 2...	1.58	33	Resto productos
A82676501	Nuñez Y Marcos Distribuidores S.A.	España	19	Pascual Yogur Griego con Frutas del Bosque - 6 ...	14.33	10	Resto productos
A82676501	Nuñez Y Marcos Distribuidores S.A.	España	21	Ace Lejía - 2l	1.55	13	Resto productos
A82676501	Nuñez Y Marcos Distribuidores S.A.	España	22	Mimosin - Intense Estallido de Pasión Suavizant...	3.46	15	Resto productos
A82676501	Nuñez Y Marcos Distribuidores S.A.	España	24	Albal Film Transparente Corte Fácil, 20 Metros ...	1.89	20	Resto productos
A82676501	Nuñez Y Marcos Distribuidores S.A.	España	25	Dodot Sensitive - Toallitas para bebé, 54 unidades	4.80	13	Resto productos
A82676501	Nuñez Y Marcos Distribuidores S.A.	España	27	Lays Patatas Fritas - 300g	1.84	10	Resto productos
A82676501	Nuñez Y Marcos Distribuidores S.A.	España	29	Oral-B Pro-Expert Protección Profesional Pasta ...	5.10	32	Resto productos
A82676501	Nuñez Y Marcos Distribuidores S.A.	España	30	Litoral Garbanzos De La Abuela - 440g	1.04	83	Resto productos

5. Mostrar un listado con los datos de los pedidos, indicando para cada línea de pedido, el NUM_LINEA, la CANTIDAD, el COD_PRODUCTO, la DESCRIPCIÓN y el PRECIO. El resultado se ordena por COD_PEDIDO, NUM_LINEA y FECHA.

```
select PEDIDO.*, LINEAS.NUM_LINEA, LINEAS.CANTIDAD,
PRODUCTO.COD_PRODUCTO, PRODUCTO.DESCRIPCION, PRODUCTO.PRECIO
from PEDIDO inner join LINEAS using(COD_PEDIDO) inner join
PRODUCTO using(COD_PRODUCTO) order by PEDIDO.COD_PEDIDO,
LINEAS.NUM_LINEA, PEDIDO.FECHA;
```

COD_PEDIDO	FECHA	CIF_SUPERMERCADO	CIF_PROVEEDOR	NUM_LINEA	CANTIDAD	COD_PRODUCTO	DESCRIPCION	PRECIO
1	2019-10-26	B10467413	B10879411	1	20	6	Cañas con azúcar La Cacería - 500g	4.25
2	2019-11-02	B10467413	B10368120	1	10	3	La Chinata - Pimentón de la Vera - Pimentón Ah...	6.17
3	2019-11-30	B10467413	A28679491	1	30	1	Manzana Golden Marlene	0.49
3	2019-11-30	B10467413	A28679491	2	40	2	Pasta al Curry - Green Masala - 200g	2.71
3	2019-11-30	B10467413	A28679491	3	50	4	Gallina Blanca - Caldo Casero de Carne - 100% ...	7.70
3	2019-11-30	B10467413	A28679491	4	15	5	Knorr - Sopa De Ave Con Fideos Finos 61g - So...	3.86
3	2019-11-30	B10467413	A28679491	5	4	9	Scottex Original Papel Higiénico - 12 Rollos	7.35
4	2020-01-04	B10467413	A82676501	1	26	19	Pascual Yogur Griego con Frutas del Bosque - 6 ...	14.33
4	2020-01-04	B10467413	A82676501	2	6	21	Ace Lejía - 2l	1.55
4	2020-01-04	B10467413	A82676501	3	17	22	Mimosin - Intense Estallido de Pasión Suavizant...	3.46
4	2020-01-04	B10467413	A82676501	4	45	24	Albal Film Transparente Corte Fácil, 20 Metros ...	1.89
4	2020-01-04	B10467413	A82676501	5	20	25	Dodot Sensitive - Toallitas para bebé, 54 unidades	4.80
4	2020-01-04	B10467413	A82676501	6	24	27	Lays Patatas Fritas - 300g	1.84
4	2020-01-04	B10467413	A82676501	7	28	29	Oral-B Pro-Expert Protección Profesional Pasta ...	5.10
4	2020-01-04	B10467413	A82676501	8	30	30	Litoral Garbanzos De La Abuela - 440g	1.04
5	2019-10-05	B10578193	B10562387	1	20	12	Cerezas caja de 2,5Kg Calibre +28	16.25
5	2019-10-05	B10578193	B10562387	2	15	14	Espárragos verdes ecológicos	6.99
6	2019-10-28	B10578193	B10542690	1	20	26	Mermelada de cereza picota - CAMPO & TIERRA...	1.20
6	2019-10-28	B10578193	B10542690	2	30	18	Cabecera de lomo embuchado - 400/450 gr	16.45
7	2019-11-07	B10578193	A28679491	1	20	2	Pasta al Curry - Green Masala - 200g	2.71
7	2019-11-07	B10578193	A28679491	2	20	4	Gallina Blanca - Caldo Casero de Carne - 100% ...	7.70
7	2019-11-07	B10578193	A28679491	3	20	10	Ace WC + Tubería Gel, Fórmula Autoactiva Des...	1.75

Consultas de resumen de datos

6. Calcular el número de departamentos que tiene cada supermercado, ordenando el resultado por el CIF del supermercado.

```
select CIF, count(COD_DEPARTAMENTO) as "NUMERO DEPARTAMENTOS"
from DEPARTAMENTOS group by CIF order by CIF;
```

CIF	NUMERO DEPARTAMENTOS
B06673892	6
B06823901	7
B10467413	7
B10578193	6

7. Listar para cada pedido el COD_PEDIDO, el CIF_SUPERMERCADO, la FECHA, la cantidad de líneas que tiene y la cantidad total de productos pedida.

```
select PEDIDO.COD_PEDIDO, PEDIDO.CIF_SUPERMERCADO, PEDIDO.FECHA,
count(LINEAS.NUM_LINEA) as "NUMERO LINEAS PEDIDO",
sum(LINEAS.CANTIDAD) as "CANTIDAD PRODUCTOS PEDIDA" from PEDIDO
inner join LINEAS using(COD_PEDIDO) group by PEDIDO.COD_PEDIDO;
```

COD_PEDIDO	CIF_SUPERMERCADO	FECHA	NUMERO LINEAS PEDIDO	CANTIDAD PRODUCTOS PEDIDA
1	B10467413	2019-10-26	1	20
2	B10467413	2019-11-02	1	10
3	B10467413	2019-11-30	5	139
4	B10467413	2020-01-04	8	196
5	B10578193	2019-10-05	2	35
6	B10578193	2019-10-28	2	50
7	B10578193	2019-11-07	6	120
8	B10578193	2019-11-25	3	33
9	B06823901	2019-10-30	1	10
10	B06823901	2019-11-09	1	10
11	B06823901	2019-11-29	4	74
12	B06823901	2019-12-04	4	65
13	B06673892	2019-10-24	1	20
14	B06673892	2019-11-08	1	5
15	B06673892	2019-11-24	8	85
16	B06673892	2019-11-10	3	33

8. Contar el número de productos cuyo TIPO es 'Típico zona' y su NUM_EXISTENCIAS es mayor que 10.

```
select count(COD_PRODUCTO) as "NUMERO PRODUCTOS" from PRODUCTO
where TIPO = 'Típico zona' and NUM_EXISTENCIAS > 10;
```

NUMERO PRODUCTOS
5

9. Calcular el precio medio y el número de existencias totales de cada TIPO de producto.

```
select TIPO, round(avg(PRECIO),2) as "PRECIO MEDIO",  
sum(NUM_EXISTENCIAS) as "EXISTENCIAS TOTALES" from PRODUCTO  
group by TIPO;
```

TIPO	PRECIO MEDIO	EXISTENCIAS TOTALES
Resto productos	4,16	707
Típico zona	57,21	169

10. Mostrar el NOMBRE del DEPARTAMENTO y el número de empleados que trabaja en cada DEPARTAMENTO cuyo CIF es 'B06673892' y tienen más de 1 empleado.

```
select DEPARTAMENTOS.NOMBRE, count(EMPLEADO.NUM_EMPLEADO) as  
"NUMERO EMPLEADOS" from DEPARTAMENTOS inner join TRABAJA  
using(COD_DEPARTAMENTO) inner join EMPLEADO using(NUM_EMPLEADO)  
where DEPARTAMENTOS.CIF = 'B06673892' group by  
DEPARTAMENTOS.NOMBRE having count(EMPLEADO.NUM_EMPLEADO) > 1;
```

NOMBRE	NUMERO EMPLEADOS
Carnicería y pescadería	2
Caja	2

Subconsultas

11. Mostrar los datos de los productos cuyo número de existencias sea el mayor.

```
select * from PRODUCTO where NUM_EXISTENCIAS = (select
max(NUM_EXISTENCIAS) from PRODUCTO);
```

COD_PRODUCTO	DESCRIPCION	PRECIO	NUM_EXISTENCIAS	TIPO
11	Brillante Arroz Integral - Pack de 2 x 125g - Tot...	1.10	91	Resto productos

12. Listar los datos de los proveedores cuyo CP sea igual al CP del PROVEEDOR con CIF 'B06245783'.

```
select * from PROVEEDOR where CP=(select CP from PROVEEDOR where
CIF='B06245783') and CIF!='B06245783';
```

CIF	NOM_COMERCIAL	CALLE	NUMERO	CP	EMAIL	TELEFONO	TIPO
B06256784	Dehesa De Alimentacion Y Bebidas S.L	Calle Logrono	12	6800	dehesaalimentacionybebidas@telefonica.net	924378663	Local

13. Contar el número de proveedores locales que suministran algún producto.

```
select count(CIF) as "Nº TOTAL PROVEEDORES LOCALES" from LOCAL
where exists (select CIF from SUMINISTRA where
LOCAL.CIF=SUMINISTRA.CIF);
```

Nº TOTAL PROVEEDORES LOCALES
8

14. Listar los datos de los pedidos de los cuales la cantidad que se ha pedido en cualquiera de sus líneas no sea mayor que 10.

```
select * from PEDIDO where not exists (select * from LINEAS
where PEDIDO.COD_PEDIDO = LINEAS.COD_PEDIDO and CANTIDAD > 10);
```


Result Grid				
Filter Rows:				
	COD_PEDIDO	FECHA	CIF_SUPERMERCADO	CIF_PROVEEDOR
2	2019-11-02	810467413	B10368120	
9	2019-10-30	B06823901	B06345467	
10	2019-11-09	B06823901	B06342894	
14	2019-11-08	B06673892	B06256784	
Result Grid				

15. Listar los datos de los empleados cuyo cargo sea el mismo que el cargo del empleado con NOMBRE 'Francisco' y APELLIDOS 'Álvarez García'.

```
select * from EMPLEADO where CARGO = (select CARGO from EMPLEADO
where NOMBRE='Francisco' and APELLIDOS= 'Álvarez García') and
NOMBRE!='Francisco' and APELLIDOS!='Álvarez García';
```

Result Grid							
Filter Rows:							
	NUM_EMPLEADO	DNI	NOMBRE	APELLIDOS	CARGO	SALARIO_NETO	DATOS_CUENTA
1	76137836Q	Antonio	García López	Carnicero	975.50	ES7645347856872357890246	Carnicería
9	76892837L	Eva	Marín Martín	Carnicero	975.50	ES1234547856072097800230	Carnicería
15	78963210R	Luis	García Antón	Carnicero	975.50	ES1234347856872357890246	Carnicería
Result Grid							

6) Rutinas

Procedimientos almacenados

1. Procedimiento almacenado que muestra los empleados que trabajan en cada supermercado según el CIF del supermercado que recibe como parámetro.

```
delimiter //
drop procedure if exists procedure1 //
create procedure procedure1(in v_cif_supermercado char(9))
begin
    case v_cif_supermercado
        when 'B10467413' then
            select EMPLEADO.* from EMPLEADO inner join
            TRABAJA using (NUM_EMPLEADO) inner join DEPARTAMENTOS using
            (COD_DEPARTAMENTO) inner join SUPERMERCADO using(CIF) where
            SUPERMERCADO.CIF = 'B10467413';
        when 'B10578193' then
            select EMPLEADO.* from EMPLEADO inner join
            TRABAJA using (NUM_EMPLEADO) inner join DEPARTAMENTOS using
            (COD_DEPARTAMENTO) inner join SUPERMERCADO using(CIF) where
            SUPERMERCADO.CIF = 'B10578193';
        when 'B06823901' then
            select EMPLEADO.* from EMPLEADO inner join
            TRABAJA using (NUM_EMPLEADO) inner join DEPARTAMENTOS using
            (COD_DEPARTAMENTO) inner join SUPERMERCADO using(CIF) where
            SUPERMERCADO.CIF = 'B06823901';
        when 'B06673892' then
            select EMPLEADO.* from EMPLEADO inner join
            TRABAJA using (NUM_EMPLEADO) inner join DEPARTAMENTOS using
            (COD_DEPARTAMENTO) inner join SUPERMERCADO using(CIF) where
            SUPERMERCADO.CIF = 'B06673892';
        else
```

```
select 'No existe ningún supermercado con ese
CIF' as MENSAJE;
```

```
end case;
```

```
end; //
```

```
delimiter ;
```

```
call procedure1('B10467413');
```

NUM_EMPLEADO	DNI	NOMBRE	APELLIDOS	CARGO	SALARIO_NETO	DATOS_CUENTA	DEPARTAMENTOS_TRABAJA
1	76137836Q	Antonio	García Lopéz	Carnicero	975.50	ES7645347856872357890246	Carnicería
2	76137937W	Álvaro	Lopéz González	Panadero	995.75	ES7645347856872350890230	Panadería
3	67738923J	Juán	González Rovira	Pescadero	1100.50	ES6745347056872357890230	Pescadería
4	58095767K	José	González Ramos	Reponedor	1000.50	ES6745347856872357890230	Limpieza
5	76887867V	Vicente	Martín Santos	Reponedor	1000.50	ES6745987856872357890230	Higiene
6	45688902M	Luis	Romero Iglesias	Reponedor	1000.50	ES6745347850072357890230	Frutería
7	35679823S	Cristina	López Iglesias	Cajero	1250.00	ES6745007856872997890230	Caja
8	76137838H	Sandra	Tena Irala	Cajero	1250.00	ES6745547856072097800239	Caja

```
call procedure1('B10578193');
```

NUM_EMPLEADO	DNI	NOMBRE	APELLIDOS	CARGO	SALARIO_NETO	DATOS_CUENTA	DEPARTAMENTOS_TRABAJA
9	76892837L	Eva	Martín Martín	Carnicero	975.50	ES1234547856072097800230	Carnicería
10	12347890H	Carlos	Nuñez Antón	Panadero	995.75	ES9805547856072097800230	Panadería
11	89456734M	Ramón	Pérez Sánchez	Reponedor	1000.50	ES6745340056872357890230	Limpieza
12	37890298Y	Juán	Martín Trejo	Reponedor	1000.50	ES5745987800872357890230	Higiene
13	67890453L	Ángel	García Linio	Reponedor	1000.50	ES9045347850072357894567	Frutería
14	76541287R	Leticia	López Fraile	Cajero	1250.00	ES4545007880872997890230	Caja

```
call procedure1('B06823901');
```

NUM_EMPLEADO	DNI	NOMBRE	APELLIDOS	CARGO	SALARIO_NETO	DATOS_CUENTA	DEPARTAMENTOS_TRABAJA
15	78963210R	Luis	García Antón	Carnicero	975.50	ES1234347856872357890246	Carnicería
16	07854578D	Marta	Lopéz Martín	Panadero	995.75	ES709847856872357890230	Panadería
17	68948798L	Enrique	García López	Reponedor	1000.50	ES1235347856872877890230	Limpieza
18	76879809D	Pepe	Martín Sánchez	Reponedor	1000.50	ES1234587856872357890230	Higiene
19	79813423L	Alicia	García Caballero	Reponedor	1000.50	ES3467347850072567890230	Frutería
20	78123456T	Bea	Martín Muñoz	Cajero	1250.00	ES4567007856872007890224	Caja

```
call procedure1('B06673892');
```

NUM_EMPLEADO	DNI	NOMBRE	APELLIDOS	CARGO	SALARIO_NETO	DATOS_CUENTA	DEPARTAMENTOS_TRABAJA
21	98129020A	Francisco	Álvarez García	Carnicero	975.50	ES4567347850072357890246	Carnicería y pescadería
23	12890020S	Santiago	Márquez Durán	Pescadero	1100.50	ES3451347856872357000230	Carnicería y pescadería
22	45678901G	Damián	Velarde González	Panadero	995.75	ES7645300856872300890230	Panadería
24	07761378M	Rodrigo	González Martín	Reponedor	1000.50	ES6745340056872300890230	Limpieza
25	23458900G	Antonio	Rey Santos	Reponedor	1000.50	ES44745987856800357890230	Higiene
26	45678902M	Blanca	Batuecas Iglesias	Reponedor	1000.50	ES9245347850072357890200	Frutería
27	65137891M	Manuel	Fernández López	Cajero	1250.00	ES2445007856872557890230	Caja
28	45870012P	Héctor	Gómez Sánchez	Cajero	1250.00	ES4945547856882097830245	Caja

```
call procedure1('B07673800');
```

MENSAJE
No existe ningún supermercado con ese CIF

2. Procedimiento para añadir un nuevo registro a la tabla DEPARTAMENTOS haciendo el tratamiento de errores utilizando handler.

```
delimiter //

drop procedure if exists procedure2//

create procedure procedure2(in v_cod_departamento int, in
v_nombre varchar(50), in v_cif_supermercado char(9))
begin
    declare error1 boolean default false;
    declare error2 boolean default false;
    declare error3 boolean default false;
    declare continue handler for 1146
        begin
            set error1 = true;
        end;
    declare continue handler for sqlstate '23000'
        begin
            set error2 = true;
        end;
    declare continue handler for 1452
        begin
            set error3 = true;
        end;

    insert into DEPARTAMENTOS values (v_cod_departamento,
v_nombre, v_cif_supermercado);

    if error1 then
        select 'La tabla DEPARTAMENTOS no existe en la
base de datos' as MENSAJE;
    else
        if error2 then
```

```
                select 'Código de departamento duplicado'
as MENSAJE;
            else
                if error3 then
                    select 'No existe ningún supermercado
con ese CIF' as MENSAJE;
                else
                    select 'Fila añadida' as MENSAJE;
                end if;
            end if;
        end if;
    end if;
end; //
delimiter ;
```

```
call procedure2(1, 'Carnicería', 'B10467413');
```

Result Grid		Filter Rows:	
	MENSAJE		
▶	Código de departamento duplicado		

```
call procedure2(60, 'Carnicería', 'B00467493');
```

Result Grid		Filter Rows:	Export
	MENSAJE		
▶	No existe ningún supermercado con ese CIF		

3. Procedimiento que muestra para cada supermercado, el nombre, los apellidos y el salario de los 3 empleados con mayor salario. Se utiliza un cursor que recorre los distintos supermercados que aparecen en la tabla SUPERMERCADOS. Además, para que el procedimiento muestre en una sola ventana el resultado de todos los supermercados, se crea una tabla dentro del procedimiento para guardar los resultados obtenidos de cada uno.

```
delimiter //
drop procedure if exists cursor1_sp//
create procedure cursor1_sp()
begin
    declare fintabla int default 0;
    declare cif_supermercado char(9);
    declare c cursor for select distinct CIF from
SUPERMERCADO;
    declare continue handler for not found set fintabla=1;
    create table if not exists EMPLEADOSMAYORSALARIO
(MENSAJE varchar(255));
    delete from EMPLEADOSMAYORSALARIO;
    open c;
    repeat
        fetch c into cif_supermercado;
        if not fintabla then
            insert into EMPLEADOSMAYORSALARIO select
concat('Empleados con mayor salario del supermercado con CIF ',
cif_supermercado);
            insert into EMPLEADOSMAYORSALARIO select
'_____';
            insert into EMPLEADOSMAYORSALARIO select
concat(concat_ws(' ', EMPLEADO.NOMBRE, EMPLEADO.APELLIDOS, 'con
un salario de', EMPLEADO.SALARIO_NETO, 'euros')) from EMPLEADO
inner join TRABAJA using (NUM_EMPLEADO) inner join DEPARTAMENTOS
using (COD_DEPARTAMENTO) inner join SUPERMERCADO using(CIF)
where SUPERMERCADO.CIF = cif_supermercado order by
SUPERMERCADO.CIF, EMPLEADO.SALARIO_NETO desc limit 3;
```

```

insert into EMPLEADOSMAYORSALARIO select
'
end if;

until fintabla
end repeat;
close c;

end; //
delimiter ;

call cursor1_sp();
select * from EMPLEADOSMAYORSALARIO;

```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	MENSAJE			
▶	Empleados con mayor salario del supermercado con CIF B06673892			
	Manuel Fernández López con un salario de 1250.00 euros			
	Héctor Gómez Sánchez con un salario de 1250.00 euros			
	Santiago Márquez Durán con un salario de 1100.50 euros			
	Empleados con mayor salario del supermercado con CIF B06823901			
	Bea Martín Muñoz con un salario de 1250.00 euros			
	Alicia García Caballero con un salario de 1000.50 euros			
	Enrique García López con un salario de 1000.50 euros			
	Empleados con mayor salario del supermercado con CIF B10467413			
	Sandra Tena Irala con un salario de 1250.00 euros			
	Cristina López Iglesias con un salario de 1250.00 euros			
	Juán González Rovira con un salario de 1100.50 euros			
	Empleados con mayor salario del supermercado con CIF B10578193			
	Leticia López Fraile con un salario de 1250.00 euros			
	Ángel García Linio con un salario de 1000.50 euros			

Funciones almacenadas

```
set global log_bin_trust_function_creators = 1;
```

1. Función que calcula la media de salarios de los empleados de cada supermercado y retorna un mensaje diciendo cual es el supermercado con mayor media de salarios.

```
delimiter //  
drop function if exists funcion1 //  
create function funcion1()  
returns varchar(200)  
begin  
    declare v_media_supermercado1 decimal (7,2);  
    declare v_media_supermercado2 decimal (7,2);  
    declare v_media_supermercado3 decimal (7,2);  
    declare v_media_supermercado4 decimal (7,2);  
    declare v_mensaje varchar(200);  
  
    select round(avg(EMPLEADO.SALARIO_NETO), 2) into  
v_media_supermercado1 from EMPLEADO inner join TRABAJA using  
(NUM_EMPLEADO) inner join DEPARTAMENTOS using (COD_DEPARTAMENTO)  
inner join SUPERMERCADO using(CIF) where SUPERMERCADO.CIF =  
'B10467413';  
  
    select round(avg(EMPLEADO.SALARIO_NETO), 2) into  
v_media_supermercado2 from EMPLEADO inner join TRABAJA using  
(NUM_EMPLEADO) inner join DEPARTAMENTOS using (COD_DEPARTAMENTO)  
inner join SUPERMERCADO using(CIF) where SUPERMERCADO.CIF =  
'B10578193';  
  
    select round(avg(EMPLEADO.SALARIO_NETO), 2) into  
v_media_supermercado3 from EMPLEADO inner join TRABAJA using  
(NUM_EMPLEADO) inner join DEPARTAMENTOS using (COD_DEPARTAMENTO)  
inner join SUPERMERCADO using(CIF) where SUPERMERCADO.CIF =  
'B06823901';  
  
    select round(avg(EMPLEADO.SALARIO_NETO), 2) into  
v_media_supermercado4 from EMPLEADO inner join TRABAJA using
```



```
(NUM_EMPLEADO) inner join DEPARTAMENTOS using (COD_DEPARTAMENTO)
inner join SUPERMERCADO using(CIF) where SUPERMERCADO.CIF =
'B06673892';
```

```
    if ((v_media_supermercado1 > v_media_supermercado2) and
(v_media_supermercado1 > v_media_supermercado3) and
(v_media_supermercado1 > v_media_supermercado4)) then
```

```
        select concat_ws(' ', 'Los empleados del supermercado
con CIF B10467413 son los que tienen un salario mayor, con una
media de', v_media_supermercado1, 'euros') into v_mensaje;
```

```
    else
```

```
        if ((v_media_supermercado2 > v_media_supermercado1)
and (v_media_supermercado2 > v_media_supermercado3) and
(v_media_supermercado2 > v_media_supermercado4)) then
```

```
            select concat_ws(' ', 'Los empleados del
supermercado con CIF B10578193 son los que tienen un salario
mayor, con una media de', v_media_supermercado2, 'euros') into
v_mensaje;
```

```
        else
```

```
            if ((v_media_supermercado3 >
v_media_supermercado1) and (v_media_supermercado3 >
v_media_supermercado2) and (v_media_supermercado3 >
v_media_supermercado4)) then
```

```
                select concat_ws(' ', 'Los empleados del
supercercado con CIF B06823901 son los que tienen un salario
mayor, con una media de', v_media_supermercado3, 'euros') into
v_mensaje;
```

```
            else
```

```
                select concat_ws(' ', 'Los empleados del
supermercado con CIF B06673892 son los que tienen un salario
mayor, con una media de', v_media_supermercado4, 'euros') into
v_mensaje;
```

```
            end if;
```

```
        end if;
```

```
    end if;
```

```
    return v_mensaje;
```

```
end; //
```

```
delimiter ;
```

```
select funcion1() as MENSAJE;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:
MENSAJE			
Los empleados del supermercado con CIF B06673892 son los que tienen un salario mayor, con una media de 1071.66 euros			

2. Función que calcula la cantidad que pide un supermercado de un producto y, si esta es mayor que 0, calcula el stock del producto que tendrá ese supermercado cuando se reciba el pedido. Si no se ha pedido ese producto, se muestra un mensaje indicándolo.

```
delimiter //
```

```
drop function if exists funcion2//
```

```
create function funcion2(v_cif_supermercado char(9),  
v_cod_producto int)
```

```
returns varchar(200)
```

```
begin
```

```
    declare v_num_existencias int;
```

```
    declare v_cantidad_pedida int default 0;
```

```
    declare v_stock int;
```

```
    declare v_mensaje varchar(200);
```

```
    select NUM_EXISTENCIAS into v_num_existencias from PRODUCTO  
where COD_PRODUCTO = v_cod_producto;
```

```
    select sum(LINEAS.CANTIDAD) into v_cantidad_pedida from  
LINEAS inner join PEDIDO using(COD_PEDIDO) where  
PEDIDO.CIF_SUPERMERCADO =v_cif_supermercado and  
LINEAS.COD_PRODUCTO = v_cod_producto;
```

```
    if (v_cantidad_pedida > 0) then
```

```
        set v_stock = v_num_existencias + v_cantidad_pedida;
```

```
        select concat_ws(' ','Cuando se reciba el pedido, el  
número de existencias del producto', v_cod_producto, 'del
```

```
supermercado con CIF', v_cif_supermercado, 'será:', v_stock)
into v_mensaje;

    else

        select 'No se ha realizado ningún pedido de ese
producto' into v_mensaje;

    end if;

    return v_mensaje;

end; //

delimiter ;
```

```
select funcion2('B10467413', 6) as MENSAJE;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	MENSAJE			
▶	Cuando se reciba el pedido, el número de existencias del producto 6 del supermercado con CIF B10467413 será: 30			

```
select funcion2('B10467413', 12) as MENSAJE;
```

Result Grid		Filter Rows:	Export:
	MENSAJE		
▶	No se ha realizado ningún pedido de ese producto		

Triggers

1. Trigger para mantener sincronizada una copia de seguridad de los nuevos pedidos realizados.

```
create table REPLICA_PEDIDO
(
COD_PEDIDO int primary key,
FECHA date not null,
CIF_SUPERMERCADO char(9) not null,
CIF_PROVEEDOR char(9) not null
);

delimiter //

drop trigger if exists trigger1//

create trigger trigger1
before insert on PEDIDO
for each row
begin
    insert into REPLICA_PEDIDO values (NEW.COD_PEDIDO,
NEW.FECHA, NEW.CIF_SUPERMERCADO, NEW.CIF_PROVEEDOR);
end; //

delimiter ;

insert into PEDIDO values (17, '2020/05/18', 'B06673892',
'A00676736');

select * from PEDIDO;
```

Result Grid				
		Filter Rows:		
		Edit:		
		Export/Import		
	COD_PEDIDO	FECHA	CIF_SUPERMERCADO	CIF_PROVEEDOR
	9	2019-10-30	B06823901	B06345467
	10	2019-11-09	B06823901	B06342894
	11	2019-11-29	B06823901	A28679491
	12	2019-12-04	B06823901	A82676501
	13	2019-10-24	B06673892	B06245783
	14	2019-11-08	B06673892	B06256784
	15	2019-11-24	B06673892	A82676501
	16	2019-11-10	B06673892	A00676736
	17	2020-05-18	B06673892	A00676736
*	NULL	NULL	NULL	NULL

```
select * from REPLICA_PEDIDO;
```

Result Grid				
		Filter Rows:		
		Edit:		
		Export/Import		
	COD_PEDIDO	FECHA	CIF_SUPERMERCADO	CIF_PROVEEDOR
▶	17	2020-05-18	B06673892	A00676736
*	NULL	NULL	NULL	NULL

2. Trigger de auditoría que almacena en la tabla auxiliar NUEVOS_EMPLEADOS, el usuario que da de alta, la fecha y los datos que se indican a continuación de los nuevos empleados que se den de alta en nuestra BD.

```
create table NUEVOS_EMPLEADOS
(
  NUM_REGISTRO int primary key auto_increment,
  REGISTRO varchar(400)
);
```

```
delimiter //
drop trigger if exists trigger2//
create trigger trigger2
after insert on EMPLEADO
for each row
begin
```

```
insert into NUEVOS_EMPLEADOS(REGISTRO) values (concat_ws('
','- ALTA REALIZADA POR:',USER(),'- FECHA:', current_Date(),'-
NUEVO EMPLEADO:', 'Num', NEW.NUM_EMPLEADO, 'Nombre', NEW.NOMBRE,
NEW.APELLIDOS, 'Cargo' ,NEW.CARGO, 'Departamentos en los que
trabaja', NEW.DEPARTAMENTOS_TRABAJA));
```

```
end; //
```

```
delimiter ;
```

```
insert into EMPLEADO values (29, '75871012D', 'Marcos', 'Sánchez
Antón', 'Cajero', '1250.00', 'ES6945547856882007830240',
'Caja');
```

```
select * from EMPLEADO;
```

NUM_EMPLEADO	DNI	NOMBRE	APELLIDOS	CARGO	SALARIO_NETO	DATOS_CUENTA	DEPARTAMENTOS_TRABAJA
21	98129020A	Francisco	Álvarez García	Carnicero	975.50	ES4567347850072357890246	Carnicería y pescadería
22	45678901G	Damián	Velarde González	Panadero	995.75	ES7645300856872300890230	Panadería
23	12890020S	Santiago	Márquez Durán	Pescadero	1100.50	ES3451347856872357000230	Carnicería y pescadería
24	07761378M	Rodrigo	González Martín	Reponedor	1000.50	ES6745340056872300890230	Limpieza
25	23458900G	Antonio	Rey Santos	Reponedor	1000.50	ES44745987856800357890230	Higiene
26	45678902M	Blanca	Batuecas Iglesias	Reponedor	1000.50	ES9245347850072357890200	Frutería
27	65137891M	Manuel	Fernández López	Cajero	1250.00	ES2445007856872557890230	Caja
28	45870012P	Héctor	Gómez Sánchez	Cajero	1250.00	ES4945547856882097830245	Caja
29	75871012D	Marcos	Sánchez Antón	Cajero	1250.00	ES6945547856882007830240	Caja
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
select * from NUEVOS_EMPLEADOS;
```

NUM_REGISTRO	REGISTRO
1	- ALTA REALIZADA POR root@localhost - FECHA 2020-05-18 - NUEVO EMPLEADO Num: 29 Nombre: Marcos Sánchez Antón Cargo: Cajero Departamentos en los que trabaja: Caja
NULL	NULL