Asignatura	Bases de datos (DAW_BD)	**Creado por**	Ines Caro
Tarea	BD02 - Base de datos relacionales	**Fecha**	22/11/2020
		Fecha de entrega	06/01/2021

1. EJERCICIO:

Vamos a crear las tablas para una tienda virtual que distribuye productos agrupados en familias en varias tiendas.

Realiza un script llamado Creatienda.sql que implemente los ejercicios descritos a continuación.

Precede cada una de las sentencias SQL de los ejercicios con un comentario que incluya el enunciado del ejercicio correspondiente. Recuerda que los comentarios van precedidos del símbolo -- al inicio de la línea.

Con las sentencias DDL de SQL crea las tablas especificadas a continuación aplicando las restricciones (constraints) pedidas. Se debe cumplir la integridad referencial.

TABLA FAMILIA: Contiene las familias a las que pertenecen los productos, como por ejemplo ordenadores, impresoras,etc.

Nombre Columna	Descripción	Tipo dato	Restricciones
Codfamilia	Código que distingue una familia de otra	Numérico de 3 dígitos	Clave primaria.
Denofamilia	Denominación de la familia	Alfanumérico de 50 caracteres	No puede haber dos familias con la misma denominación. Debe tener contenido.

TABLA PRODUCTO => contendrá información general sobre los productos que distribuye la empresa a las tiendas.

Nombre Columna	Descripción	Tipo dato	Restricciones
Codproducto	Código que distingue un producto de otro	Numérico de 5 dígitos	Clave primaria.
Denoproducto	Denominación del producto	Alfanumérico de 20 caracteres	Debe tener contenido.
Descripcion	Descripción del producto	Alfanumérico de 100 caracteres	
PrecioBase	Precio base del producto	Numérico de 8 dígitos dos de ellos decimales	Mayor que 0. Debe tener contenido.
PorcReposición	Porcentaje de reposición aplicado a ese producto. Se utilizará para aplicar a las unidades mínimas y obtener el número total de unidades a reponer cuando el stock esté bajo mínimo	Numérico de 3 dígitos	Mayor que 0
UnidadesMinimas	Unidades mínimas recomendables en almacen	Numérico de 4 dígitos	Mayor que 0. Debe tener contenido.

Codfamilia	Código de la familia a la que pertenece el producto	Numérico de 3 dígitos	Clave ajena, referencia a Codfamilia de la tabla FAMILIA. Debe tener contenido.

TABLA TIENDA=> contendrá información básica sobre las tiendas que distribuyen los productos.

Nombre Columna	**Descripción**	**Tipo dato**	**Restricciones**
Codtienda	Código que distingue una tienda de otra.	Numérico de 3 dígitos	Clave primaria.
Denotienda	Denominación o nombre de la tienda.	Alfanumérico de 20 caracteres	Debe tener contenido.
Telefono	Teléfono de la tienda	Alfanumérico de 11 caracteres	
CodigoPostal	Codigo Postal donde se ubica la tienda	Alfanumérico de 5 caracteres	Debe tener contenido.
Provincia	Provincia donde se ubica la tienda	Alfanumérico de 5 caracteres	Debe tener contenido.

TABLA STOCK => Contendrá para cada tienda el número de unidades disponibles de cada producto. La clave primaria está formada por la concatenación de los campos Codtienda y Codproducto.

Nombre Columna	**Descripción**	**Tipo dato**	**Restricciones**	
Codtienda	Código de la tienda.	Numérico de 3 dígitos	Clave primaria: (Codtienda,Codproducto) Permite que un producto pueda aparecer en varias tiendas, y que en una tienda puedan haber varios productos.	Clave ajena, referencia a Codtienda de la tabla tienda. Debe tener contenido.
Codproducto	Código del producto	Numérico de 5 dígitos		Clave ajena, referencia a Codproducto de la tabla PRODUCTO. Debe tener contenido.
Unidades	Unidades de ese producto en esa tienda	Numérico de 6 dígitos.	Mayor o igual a 0. Debe tener contenido.	

```
SQL> CREATE TABLE FAMILIA(
  2  CODFAMILIA VARCHAR(3) PRIMARY KEY,
  3  DENOFAMILIA VARCHAR(50) NOT NULL);

Tabla creada.

SQL> CREATE TABLE PRODUCTO(
  2  CODPRODUCTO VARCHAR(5) PRIMARY KEY,
  3  DENOPRODUCTO VARCHAR(20) NOT NULL,
  4  DESCRIPCION VARCHAR(100),
  5  PRECIOBASE DECIMAL NOT NULL,
  6  PORCREPOSICION NUMBER(3) NOT NULL,
  7  UNIMIN NUMBER(4) NOT NULL,
  8  CODFAMILIA NUMBER(3) NOT NULL);

Tabla creada.

SQL> CREATE TABLE TIENDA(
  2  CODTIENDA NUMBER(3) PRIMARY KEY,
  3  DENOTIENDA VARCHAR(20) NOT NULL,
  4  TELEFONO VARCHAR(11),
  5  CODIGOPOSTAL VARCHAR(5) NOT NULL,
  6  PROVINCIA VARCHAR(5) NOT NULL);

Tabla creada.

SQL> CREATE TABLE STOCK(
  2  CODTIENDA NUMBER(3) NOT NULL,
  3  CODPRODUCTO VARCHAR(5) NOT NULL,
  4  UNIDADES NUMBER(6) NOT NULL);

Tabla creada.

SQL> ALTER TABLE FAMILIA ADD CONSTRAINT FK_FAMILIA_PRODUCTO
  2  FOREIGN KEY (CODFAMILIA)
  3  REFERENCES FAMILIA(CODFAMILIA);

Tabla modificada.

SQL> ALTER TABLE PRODUCTO ADD CONSTRAINT FK_PRODUCTO_STOCK
  2  FOREIGN KEY (CODPRODUCTO)
  3  REFERENCES PRODUCTO (CODPRODUCTO);

Tabla modificada.
```

```
SQL> ALTER TABLE TIENDA ADD CONSTRAINT FK_TIENDA_STOCK
  2  FOREIGN KEY (CODTIENDA)
  3  REFERENCES TIENDA(CODTIENDA);

Tabla modificada.

SQL> ALTER TABLE STOCK ADD CONSTRAINT PK_STOCK
  2  PRIMARY KEY (CODTIENDA,CODPRODUCTO);

Tabla modificada.

SQL> ALTER TABLE PRODUCTO ADD CONSTRAINT CK_PRECIOBASE_PORCREPOSICION_UNIMIN
  2  CHECK (PRECIOBASE >0);

Tabla modificada.

SQL> ALTER TABLE PRODUCTO ADD CONSTRAINT CK_PORCREPOSICION
  2  CHECK (PORCREPOSICION >0);

Tabla modificada.

SQL> ALTER TABLE PRODUCTO ADD CONSTRAINT CK_UNIMIN
  2  CHECK(UNIMIN >=0);

Tabla modificada.

SQL> aLTER TABLE STOCK ADD CONSTRAINT CK_UNIDADES
  2  CHECK (UNIDADES >=0);

Tabla modificada.

SQL>
```

2. EJERCICIO:

A) Modificar las tablas creadas en el ejercicio anterior siguiendo las indicaciones. Los ejercicios se incluirán en un script llamado ModificaTienda.sql. Cada uno de ellos, como en el ejercicio anterior, irá precedido de un comentario con el enunciado.

Añadir a la tabla STOCK

- Una columna de tipo fecha llamada FechaUltimaEntrada que por defecto tome el valor de la fecha actual.
- Una columna llamada Beneficio que contendrá el tipo de porcentaje de beneficio que esa tienda aplica en ese producto. Se debe controlar que el valor que almacene sea 1,2, 3, 4 o 5.

```
SQL> ALTER TABLE STOCK ADD(
  2  FECHAULTIMAENTRADA DATE DEFAULT SYSDATE);

Tabla modificada.

SQL> ALTER TABLE STOCK ADD(
  2  BENEFICIO NUMBER(1) CHECK(BENEFICIO BETWEEN 1 AND 5));

Tabla modificada.

SQL>
```

En la tabla PRODUCTO

- Eliminar de la tabla producto la columna Descripción.
- Añadir una columna llamada perecedero que únicamente acepte los valores: S o N.
- Modificar el tamaño de la columna Denoproducto a 50.

```
SQL> ALTER TABLE PRODUCTO DROP COLUMN DESCRIPCION;

Tabla modificada.

SQL> ALTER TABLE PRODUCTO ADD(
  2  PERECEDERO VARCHAR(1) CHECK(PERECEDERO IN('S','N')));

Tabla modificada.

SQL> ALTER TABLE PRODUCTO
  2  MODIFY DENOPRODUCTO VARCHAR(50);

Tabla modificada.
```

En la tabla FAMILIA

- Añadir una columna llamada IVA, que represente el porcentaje de IVA y únicamente pueda contener los valores 21,10,ó 4.

```
SQL> ALTER TABLE FAMILIA ADD(
  2  IVA NUMBER);

Tabla modificada.

SQL> ALTER TABLE FAMILIA ADD CONSTRAINT CK_IVA
  2  CHECK(IVA IN(4,10,21));

Tabla modificada.
```

En la tabla tienda

- La empresa desea restringir el número de tiendas con las que trabaja, de forma que no pueda haber más de una tienda en una misma zona (la zona se identifica por el código postal). Definir mediante DDL las restricciones necesarias para que se cumpla en el campo correspondiente..

```
SQL> ALTER TABLE TIENDA
  2  ADD CONSTRAINT CODIGOPOSTAL UNIQUE (CODIGOPOSTAL);

Tabla modificada.
```

B) Renombra la tabla STOCK por PRODXTIENDAS.

```
SQL> RENAME STOCK TO PRODXTIENDAS;
Nombre de tabla cambiado.
```

C) Elimina la tabla FAMILIA y su contenido si lo tuviera.

```
SQL> DROP TABLE FAMILIA CASCADE CONSTRAINT;

Tabla borrada.
```

D) Crea un usuario llamado C##INVITADO siguiendo los pasos de la unidad 1 y dale todos los privilegios sobre la tabla PRODUCTO.

```
SQL> CREATE USER C##INVITADO IDENTIFIED BY INVITADO;

Usuario creado.
```

```
SQL> GRANT ALL PRIVILEGES ON PRODUCTO TO C##INVITADO;

Concesi¾n terminada correctamente.
```

E) Retira los permisos de modificar la estructura de la tabla y borrar contenido de la tabla PRODUCTO al usuario anterior.

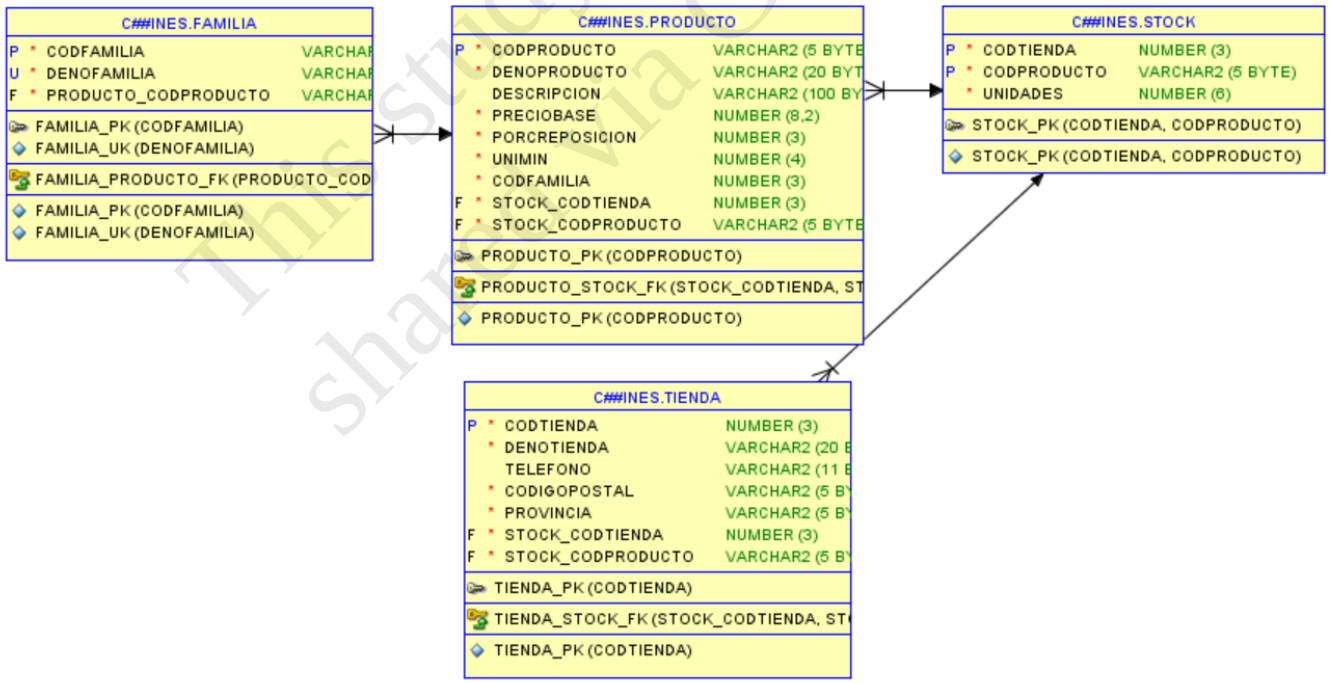
```
SQL> REVOKE ALTER, UPDATE ON PRODUCTO FROM C##INVITADO;

Revocaci¾n terminada correctamente.
```

3. EJERCICIO:

SQLDeveloper permite obtener el diagrama del modelo entidad relación a partir de las tablas ya creadas con la información contenida en el Diccionario de Datos. Una vez tengas realizados los ejercicios 1 y 2 genera el diagrama entidad relación y expórtalo en formato PNG.

Crea Tienda:



Modifica Tienda: