Ejercicio 1: Creación de las tablas para una tienda virtual.

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.	

```
-- Crea la tabla Familia
-------------------------------------------------------------------------------
CREATE TABLE FAMILIA(
Codfamilia number(3) NOT NULL,
Denofamilia varchar2(50) unique not null,
CONSTRAINT pk_codfamilia PRIMARY KEY (Codfamilia));
-------------------------------------------------------------------------------
-- crea la tabla Producto
CREATE TABLE PRODUCTO(
Codproducto number(5) NOT NULL,
Denoproducto varchar2(20) not null,
Descripcion varchar2(100),
PrecioBase number(8,2) not null,
PorcReposicion number(3),
UnidadesMinimas number(4) not null,
UnidadesActuales number(4) not null,
Codfamilia number(3) not null,
CONSTRAINT pk_codproducto PRIMARY KEY(Codproducto),
CONSTRAINT chk_preciobase check (PrecioBase >0),
CONSTRAINT fk_codfamilia FOREIGN KEY (Codfamilia) REFERENCES FAMILIA(Codfamilia),
CONSTRAINT chk_PorcReposicion CHECK (PorcReposicion >0),
CONSTRAINT chk_UnidadesMinimas CHECK (UnidadesMinimas>0),
CONSTRAINT chk_UnidadesActuales CHECK (UnidadesActuales >0));

-- crea la tabla TIENDA
-----------------------------------------------
CREATE TABLE TIENDA(
Codtienda number(3) NOT NULL,
Denotienda varchar2(20) NOT NULL,
Telefono varchar2(11),
CodigoPostal VARCHAR2(5) NOT NULL,
Provincia varchar2(10) NOT NULL,
CONSTRAINT Pk codtienda PRIMARY KEY (Codtienda));
-------------------------------------------------------------
-- crea la tabla STOCK
CREATE TABLE STOCK(
Codtienda number(3) NOT NULL,
Codproducto number(5) NOT NULL,
Unidades number(6) NOT NULL,
CONSTRAINT chk Unidades  CHECK (Unidades >=0),
CONSTRAINT FK_codtienda FOREIGN KEY (Codtienda) REFERENCES tienda(Codtienda),
CONSTRAINT FK_codproducto FOREIGN KEY (codproducto) REFERENCES producto(codproducto),
CONSTRAINT PK_stock PRIMARY KEY (Codtienda,Codproducto));
```

Ejercicio 2: Hacer unas modificaciones en las tablas.

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.
- •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.
- •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.
- •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..

B) Renombra la tabla STOCK por PRODXTIENDAS.

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

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

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

```sql
-- Añadir a la tabla STOCK
-- Una columna de tipo fecha llamada FechaUltimaEntrada que por defecto tome el
valor de la fecha actual.
ALTER TABLE STOCK  ADD FechaUltimaEntrada DATE DEFAULT  SYSDATE;
-- 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.
ALTER TABLE STOCK ADD Beneficio NUMBER(1) CONSTRAINT CHK_BENE_STOCK CHECK(Beneficio
in (1,2,3,4,5));
--En la tabla PRODUCTO
-- Eliminar de la tabla producto la columna Descripción.
ALTER TABLE producto DROP COLUMN descripcion;
-- Añadir una columna llamada perecedero que únicamente acepte los valores: S o N.
ALTER TABLE producto ADD perecedero char CONSTRAINT CHK_PERE_STOCK CHECK(perecedero
IN('S','N'));
-- Modificar el tamaño de la columna Denoproducto a 50.
ALTER TABLE PRODUCTO MODIFY Denoproducto VARCHAR2(50);
-- 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.
ALTER TABLE familia ADD IVA NUMBER(2) CONSTRAINT CHK_IVA_FAMI CHECK(IVA IN(21,10,4));
-- 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..
ALTER TABLE TIENDA ADD CONSTRAINT UNI_TIEN_COD UNIQUE(CodigoPostal);
-- Renombra la tabla STOCK  por PRODXTIENDAS.
RENAME  STOCK  TO PRODXTIENDAS;
--Elimina la tabla FAMILIA  y constraints relacionadas.
DROP TABLE FAMILIA  CASCADE CONSTRAINTS;
-- Crea un usuario llamado C##INVITADO siguiendo los pasos de la unidad 1 y dale
todos los privilegios sobre la tabla PRODUCTO.
CONNECT sys as sysdba
CREATE USER C##INVITADO IDENTIFIED BY BD02;
GRANT ALL ON c##prueba.PRODUCTO TO C##INVITADO;
--  Retira los permisos de modificar la estructura de la tabla y borrar contenido de
la tabla PRODUCTO al usuario anterior.
REVOKE ALTER, DELETE ON c##prueba.PRODUCTO FROM C##INVITADO;
```

Ejercicio 3: Obtener el diagrama del modelo entidad relación.

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.
En este enlace tienes los pasos a seguir.