

PRÁCTICA 1.-CREACIÓN DE TABLAS SQL

Ejercicio 1.- Crear las siguientes tablas de acuerdo con las restricciones que se mencionan:

FABRICANTES			
COLUMNA	TAMAÑO	TIPO	OBLIG.
COD_FABRICANTE	3	Entero	Si
NOMBRE	15	Texto-Variable	
PAIS	15	Texto-Variable	

La clave primaria es COD_FABRICANTE.
Las columnas NOMBRE y PAIS han de almacenarse en mayúscula.

```
/* FABRICANTES */
CREATE TABLE FABRICANTES(
    COD_FABRICANTE NUMBER(3) CONSTRAINT PK_FA PRIMARY KEY,
    NOMBRE VARCHAR2(15) CONSTRAINT CK_NO CHECK(NOMBRE=UPPER(NOMBRE)),
    PAIS VARCHAR2(15) CONSTRAINT CK_PA CHECK(PAIS=UPPER(PAIS))
);
Number (3) es lo mismo que NUMBER (3,0).
```

Hay que tener en cuenta que redondea el valor introducido. Ejemplo: 123,89 se guarda 124

ARTICULOS			
COLUMNA	TAMAÑO	TIPO	OBLIG.
ARTICULO	20	Texto-Variable	Si
COD_FABRICANTE	3	Entero	Si
PESO	3	Entero	Si
CATEGORIA	20	Texto-Variable	Si
PRECIO_VENTA	4	Entero	
PRECIO_COSTO	4	Entero	
EXISTENCIAS	5	Entero	

La clave primaria está formada por las columnas: ARTICULO, COD_FABRICANTE, PESO y CATEGORIA.
COD_FABRICANTE es clave ajena que referencia a la tabla FABRICANTES.

```
/* ARTÍCULOS */
CREATE TABLE ARTICULOS(
    ARTICULO VARCHAR2(20),
    COD_FABRICANTE NUMBER(3),
    PESO NUMBER(3),
    CATEGORIA VARCHAR2(20),
    PRECIO_VENTA NUMBER (4),
    PRECIO_COSTO NUMBER (4),
    EXISTENCIAS NUMBER (5),
    CONSTRAINT PK_ART PRIMARY KEY (ARTICULO, COD_FABRICANTE, PESO, CATEGORIA),
    CONSTRAINT FK_AR_FA FOREIGN KEY (COD_FABRICANTE) REFERENCES FABRICANTES
);
```

TIENDAS			
COLUMNA	TAMAÑO	TIPO	OBLIG.
NIF	10	Texto	Si
NOMBRE	10	Texto-Variante	
DIRECCIÓN	20	Texto-Variante	
POBLACIÓN	20	Texto-Variante	
PROVINCIA	20	Texto-Variante	
CODPOSTAL	5	Entero	
La clave primaria es NIF.			
PROVINCIA ha de almacenarse en mayúscula.			

```

CREATE TABLE TIENDAS(
    NIF CHAR(10) CONSTRAINT PK_TI PRIMARY KEY,
    NOMBRE VARCHAR2(10),
    DIRECCIÓN VARCHAR2(20),
    POBLACIÓN VARCHAR2(20),
    PROVINCIA VARCHAR2(20)CONSTRAINT CK_PRO CHECK(PROVINCIA=UPPER(PROVINCIA)),
    CODPOSTAL NUMBER(5)
);

```

PEDIDOS			
COLUMNA	TAMAÑO	TIPO	OBLIG.
NIF	10	Texto	Si
ARTICULO	20	Texto-Variante	Si
COD_FABRICANTE	3	Entero	Si
PESO	3	Entero	Si
CATEGORIA	20	Texto-Variante	Si
FECHA_PEDIDO		Fecha	Si
UNIDADES_PEDIDAS	4	Entero	
La clave primaria está formada por las columnas: NIF, ARTICULO, COD_FABRICANTE, PESO, CATEGORIA y FECHA_PEDIDO.			
COD_FABRICANTE es clave ajena que referencia a la tabla FABRICANTES.			
UNIDADES_PEDIDAS debe ser positivo.			
CATEGORIA ha de ser 'Primera', 'Segunda' o 'Tercera'.			
Las columnas ARTICULO, COD_FABRICANTE, PESO y CATEGORIA son clave ajena y referencian a la tabla ARTICULOS. Realizar un borrado en cascada.			
NIF es clave ajena y referencia a la tabla TIENDAS.			

```

CREATE TABLE PEDIDOS(
    NIF CHAR(10) CONSTRAINT FK_PE_TI REFERENCES TIENDAS,
    ARTICULO VARCHAR2(20),
    COD_FABRICANTE NUMBER(3) CONSTRAINT FK_PE_FA REFERENCES FABRICANTES,
    PESO NUMBER(3),
    CATEGORIA VARCHAR2(10),
    FECHA_PEDIDO DATE,
    UNIDADES_PEDIDAS NUMBER(4) CONSTRAINT CK_PEUP CHECK (UNIDADES_PEDIDAS>=0),
    CONSTRAINT PK_PED PRIMARY KEY(NIF,ARTICULO, COD_FABRICANTE, PESO,
        CATEGORIA, FECHA_PEDIDO),
    CONSTRAINT CK_CATPE CHECK(CATEGORIA IN('Primera','Segunda','Tercera')),
    CONSTRAINT FK_PE_AR
        FOREIGN KEY (ARTICULO, COD_FABRICANTE, PESO, CATEGORIA)
        REFERENCES ARTICULOS ON DELETE CASCADE);

```

VENTAS			
COLUMNA	OBLIG.	TIPO	OBLIG.
NIF	10	Texto	Si
ARTICULO	20	Texto-Variable	Si
COD_FABRICANTE	3	Entero	Si
PESO	3	Entero	Si
CATEGORIA	10	Texto-Variable	Si
FECHA_VENTA		DATE	Si
UNIDADES_VENDIDAS	4	Entero	
 La clave primaria está formada por las columnas: NIF, ARTICULO, COD_FABRICANTE, PESO, CATEGORIA y FECHA_VENTA. COD_FABRICANTE es clave ajena que referencia a la tabla FABRICANTES. UNIDADES_VENDIDAS debe ser positivo. CATEGORIA ha de ser 'Primera', 'Segunda' o 'Tercera'. Las columnas ARTICULO, COD_FABRICANTE, PESO y CATEGORIA son clave ajena y referencian a la tabla ARTICULOS. Realizar un borrado en cascada. NIF es clave ajena y referencia a la tabla TIENDAS.			

```

CREATE TABLE VENTAS(
    NIF CHAR(10) CONSTRAINT FK_VE_TI REFERENCES TIENDAS,
    ARTICULO VARCHAR2(20),
    COD_FABRICANTE NUMBER(3) CONSTRAINT FK_VE_FA REFERENCES FABRICANTES,
    PESO NUMBER(3),
    CATEGORIA VARCHAR2(10)
        CONSTRAINT CK_CATVE CHECK(CATEGORIA IN('Primera', 'Segunda', 'Tercera')),
    FECHA_VENTA DATE,
    UNIDADES_VENDIDAS NUMBER(4) CONSTRAINT CK_VEUV CHECK (UNIDADES_VENDIDAS>0),
    CONSTRAINT PK_VEN PRIMARY KEY
        (NIF,ARTICULO, COD_FABRICANTE, PESO, CATEGORIA, FECHA_VENTA),
    CONSTRAINT FK_VEAR FOREIGN KEY(ARTICULO, COD_FABRICANTE, PESO, CATEGORIA)
        REFERENCES ARTICULOS ON DELETE CASCADE
);

```

Ejercicio 2.- Realiza las siguientes modificaciones en las tablas anteriores:

- 1) Añade las siguientes restricciones a la tabla Artículo:
 - PRECIO_VENTA, PRECIO_COSTO y PESO han de ser positivos.
 - CATEGORIA sólo puede tomar los valores ‘Primera’, ‘Segunda’ o ‘Tercera’.

```
ALTER TABLE ARTICULOS
```

```
ADD CONSTRAINT CK_VENTA CHECK (PRECIO_VENTA>0)
```

```
ADD CONSTRAINT CK_COSTO CHECK (PRECIO_COSTO>0)
```

```
ADD CONSTRAINT CK_PES CHECK (PESO>0)
```

```
ADD CONSTRAINT CK_AR_CA CHECK (CATEGORIA
```

```
IN('Primera','Segunda','Tercera'))
```

- 1) Añadir una restricción a la tabla TIENDAS para que el NOMBRE de la tienda sea de tipo título. Función INITCAP()

```
/* Sin dar nombre a las restricciones:  
ALTER TABLE TIENDAS ADD CHECK (NOMBRE=INITCAP(NOMBRE)) */
```

```
/* Dando nombre a las restricciones */  
ALTER TABLE TIENDAS ADD CONSTRAINT CK_NOMTITU  
CHECK (NOMBRE=INITCAP(NOMBRE));
```

- 2) Visualizar las constraints definidas para las tablas anteriores.

```
SELECT TABLE_NAME, CONSTRAINT_NAME, CONSTRAINT_TYPE  
FROM USER_CONSTRAINTS  
WHERE TABLE_NAME IN  
( 'ARTICULOS', 'TIENDAS', 'FABRICANTES', 'PEDIDOS', 'VENTAS' );
```

- 3) Modificar las columnas de las tablas PEDIDOS y VENTAS para que las UNIDADES_VENDIDAS y las UNIDADES_PEDIDAS puedan almacenar cantidades numéricas de 6 dígitos.

```
ALTER TABLE PEDIDOS MODIFY (UNIDADES_PEDIDAS NUMBER(6));  
ALTER TABLE VENTAS MODIFY (UNIDADES_VENDIDAS NUMBER(6));
```

- 4) Impedir que se den de alta más tiendas en la provincia de TOLEDO. (! = 'TOLEDO')

```
ALTER TABLE TIENDAS ADD CONSTRAINT CK_TIENTO  
CHECK (PROVINCIA != 'TOLEDO');
```

- 5) Añadir a las tablas PEDIDOS y VENTAS una nueva columna para que almacenen el PVP del artículo.

```
ALTER TABLE PEDIDOS ADD (PVP NUMBER(4));  
ALTER TABLE VENTAS ADD (PVP NUMBER(4));
```

- 6) Borra la tabla FABRICANTES

```
DROP TABLE FABRICANTES CASCADE CONSTRAINTS;
```

7) Elimina todos los datos de la tabla pedido
`TRUNCATE TABLE PEDIDOS;`