

```

-- 1. Cambia la intercalación de la base de datos Gestion8 para que el sistema sea
sensible a las mayúsculas y minúsculas.
    1.1. Redacta una consulta para comprobar si el sistema no es sensible a las
mayúsculas.
        SELECT * FROM CLIENTES -- la tabla se llama clientes, no debería importar
    1.2. Redacta y ejecuta una consulta para cambiar la intercalación.
        ALTER DATABASE Gestion8 COLLATE Modern_Spanish_CS_AI
GO
    1.3. Volver a ejecutar la SELECT
    1.4. --Vuelve a dejar la intercalación como estaba.
        ALTER DATABASE Gestion8 COLLATE Modern_Spanish_CI_AI
GO

-- 2. Crea una nueva base de datos Gestion2, los archivos físicos se llamarán
también Gestion2 y se creará en la misma carpeta que la base de datos Gestion8.
CREATE DATABASE Gestion2; -- La crea en la carpeta por defecto de SQL Server.
-- o bien
CREATE DATABASE Gestion2 ON (NAME='Gestion2' , FILENAME='F:\SQL\Gestion2.mdf')
LOG ON (NAME='Gestion2_log' , FILENAME='F:\SQL\Gestion2.ldf')
GO

Nota: Sustituye la ruta F:\SQL\ por la tuya donde tengas Gestion8.

-- 3. En Gestion2, crea las mismas tablas que tenemos en Gestion con las mismas
relaciones entre ellas. Este ejercicio se resuelve con varias sentencias.
USE Gestion2
CREATE TABLE dbo.empleados(
    numemp      INT NOT NULL CONSTRAINT pk_empleados PRIMARY KEY CLUSTERED,
    nombre      CHAR(20) NULL,
    edad        INT NULL,
    oficina     INT NULL,
    titulo       NVARCHAR(20) NULL,
    contrato    DATETIME NULL,
    jefe INT NULL CONSTRAINT FK_empleados_jefe REFERENCES dbo.empleados,
    cuota       MONEY NULL,
    ventas      MONEY NULL
)
CREATE TABLE dbo.oficinas(
    oficina     INT NOT NULL,
    ciudad      NVARCHAR(20) NULL,
    region      NVARCHAR(20) NULL,
    dir         INT NULL,
    objetivo    MONEY NULL,
    ventas      MONEY NULL CONSTRAINT d1 DEFAULT (0),
    CONSTRAINT PK_oficinas PRIMARY KEY (oficina),
    CONSTRAINT FK_oficinas_directores FOREIGN KEY (dir) REFERENCES empleados
)
ALTER TABLE dbo.empleados WITH CHECK ADD CONSTRAINT FK_empleados_oficina
FOREIGN KEY(oficina) REFERENCES dbo.oficinas (oficina)
CREATE TABLE dbo.clientes(
    numclie     INT NOT NULL,
    nombre      NVARCHAR(20) NOT NULL,
    repclie     INT NULL CONSTRAINT FK_clientes_repclie FOREIGN KEY REFERENCES
dbo.empleados,
    limitecredito INT NOT NULL,
    CONSTRAINT PK_clientes PRIMARY KEY (numclie)
)
CREATE TABLE dbo.productos(
    idfab       NVARCHAR(20) NOT NULL,
    idproducto  NVARCHAR(20) NOT NULL,
    descripcion CHAR(20) NOT NULL,
    precio      MONEY NULL,
    existencias INT NOT NULL,
    CONSTRAINT PK_productos PRIMARY KEY (idfab,idproducto))
CREATE TABLE dbo.pedidos(

```

```

codigo      INT IDENTITY NOT NULL,
numpedido   NVARCHAR(10) NOT NULL,
fechapedido DATETIME NOT NULL,
clie        INT NOT NULL,
rep         INT NOT NULL,
fab         NVARCHAR(20) NOT NULL,
producto    NVARCHAR(20) NOT NULL,
cant        INT NULL,
importe     MONEY NULL,
CONSTRAINT PK_pedidos PRIMARY KEY (codigo),
CONSTRAINT FK_pedidos_rep FOREIGN KEY (rep) REFERENCES empleados,
CONSTRAINT FK_pedidos_clie FOREIGN KEY (clie) REFERENCES clientes,
CONSTRAINT FK_pedidos_productos FOREIGN KEY (fab,producto) REFERENCES
productos)
-- La restricción NOT NULL en los campos que son Primary key no es necesaria ya que
si no se indica, el sistema lo asigna de oficio. Las restricciones NULL indicadas
tampoco son necesarias porque es el valor asumido por defecto.

-- 4. La tabla Productos de Gestion2 consideramos que sufre pocas operaciones de
nuevos registros o eliminación de registros y el único campo que sufre
actualizaciones frecuentes es el campo existencias; por otro lado es frecuente
consultar los productos por el campo descripcion (aunque varios productos puedan
tener la misma descripción). ¿Se podría mejorar la base de datos? Si es que sí
indica de qué forma y escribe la instrucción correspondiente; en cualquier caso
justifica la respuesta.
CREATE INDEX IX_descripcion ON Productos (descripcion);

-- 5. ¿Qué se puede hacer para que no puedan existir en la tabla de clientes dos
clientes con el mismo nombre?
CREATE UNIQUE INDEX IX_nombre ON Clientes (nombre)
-- o bien
ALTER TABLE Clientes ADD CONSTRAINT IX_nombre UNIQUE (nombre);

-- 6. En nuestro sistema vamos a efectuar muchas consultas recuperando a la vez los
pedidos y los productos asociados a cada uno de ellos, por eso sería bueno...
Completa la frase y escribe la sentencia SQL correspondiente.
CREATE VIEW pedidos_productos AS (SELECT * FROM pedidos INNER JOIN productos ON
fab=idfab AND producto=idproducto);

-- 7. Elimina los clientes dejando en pedidos el campo cliente a NULL.
-- Este ejercicio se resuelve con varias sentencias y se puede resolver de varias
maneras. Resuélvelo de distintas formas.
ALTER TABLE pedidos ALTER COLUMN clie INT NULL; -- Esto de todas formas
-- Opcion 1
ALTER TABLE pedidos DROP CONSTRAINT FK_pedidos_clie;
ALTER TABLE pedidos ADD CONSTRAINT FK_pedidos_clie FOREIGN KEY (clie) REFERENCES
Clientes ON DELETE SET NULL;
DELETE Clientes;
SELECT * FROM pedidos -- Para comprobar el resultado;
-- Opcion 2
UPDATE pedidos SET clie = NULL;
DELETE Clientes
SELECT * FROM pedidos;

-- 8. Ahora queremos permitir que dos clientes tengan el mismo nombre.
DROP INDEX IX_nombre ON Clientes;
--o bien (según se haya resuelto el apartado 5)
ALTER TABLE Clientes DROP CONSTRAINT IX_nombre;

-- 9. Elimina la base de datos Gestion2, así podrás volver a hacer los ejercicios.
USE Gestion8 -- Para poder borrar Gestion2, esta no puede ser la bd activa
DROP DATABASE Gestion2;
GO

```