CREAR OBJETOS BASES DE DATOS

Creamos prueba1

Pag18

CREATE DATABASE Prueba;

CREATE TABLE dbo.herramientas(

ID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

Nombre VARCHAR(40) NOT NULL

)

-- insertamos valores

INSERT INTO dbo.herramientas (Nombre ) VALUES ('Martillo') INSERT INTO dbo.herramientas (Nombre ) VALUES ('Taladro')

-- si borramos, Martillo, se pierde el ID 1. Para reutilizarlo debo establecer

SET IDENTITY\_INSERT dbo.herramientas ON

INSERT INTO dbo.herramientas (ID, Nombre) VALUES (1, 'Serrucho');

CREAR BASE DE DATOS PRUEBA 1 Y INSERTAR

use Prueba1;

CREATE TABLE dbo.herramientas(

ID INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

Nombre VARCHAR(40) NOT NULL

)

-- insertamos valores

INSERT INTO dbo.herramientas (Nombre ) VALUES ('Martillo') INSERT INTO dbo.herramientas (Nombre ) VALUES ('Taladro')

-- si borramos, Martillo, se pierde el ID 1. Para reutilizarlo debo establecer

SET IDENTITY\_INSERT dbo.herramientas ON

INSERT INTO dbo.herramientas (ID, Nombre) VALUES (1, 'Serrucho');

BORRAR

*-- si borramos, Martillo, se pierde el ID 1. Para reutilizarlo debo establecer*

use Prueba1;

DELETE FROM dbo.herramientas WHERE Nombre ='Martillo';

use Prueba1;

SET IDENTITY\_INSERT dbo.herramientas ON

INSERT INTO dbo.herramientas (ID, Nombre) VALUES (1, 'Serrucho');

BORRAR OBJETOS.

Crear prueba base de datos (1x comandos y otra de modo grafico)

TAMAÑO 5 MB

MAXIMO 10 MB

Crecimiento del 10%

FICHERO DATOS

Fichero transaciones

Tamaño max 1 mb

Crecimiento del 10%

Modern\_Spanish\_

No distinga mayúscula de minúsculas

Y que tenga los caracteres acentuados

Guardar carpeta bajo c:/

modo por comandos (crea una carpeta en c que se llame PruebaBaseDatos1)

CREATE DATABASE [PruebaBaseDatos1]

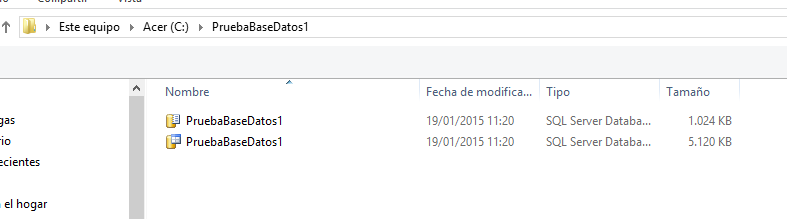
ON (NAME = N'PruebaBaseDatos1',

FILENAME = N'C:\PruebaBaseDatos1\PruebaBaseDatos1.mdf' ,

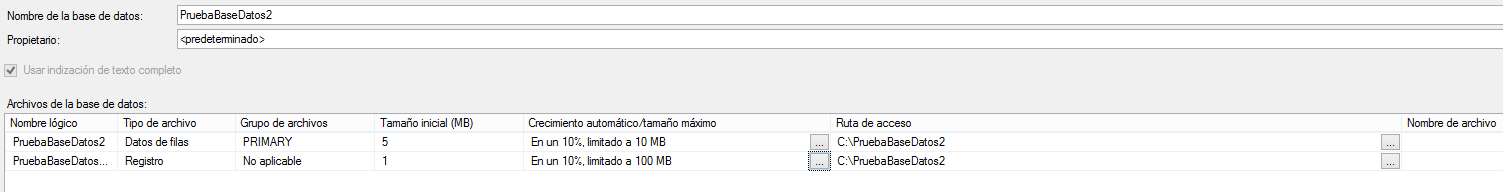
SIZE = 5, MAXSIZE = 10000,FILEGROWTH = 10%) LOG ON (NAME = N'PruebaBaseDatos1\_log',

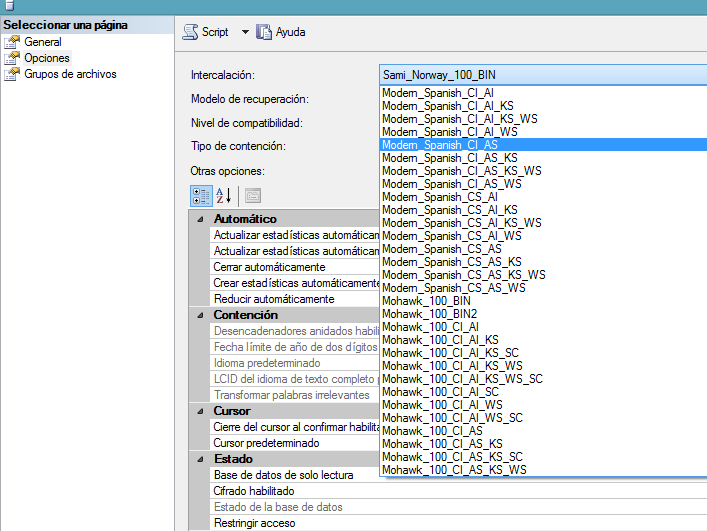
FILENAME = N'C:\PruebaBaseDatos1\PruebaBaseDatos1.LDF' , SIZE = 1, FILEGROWTH = 10%)

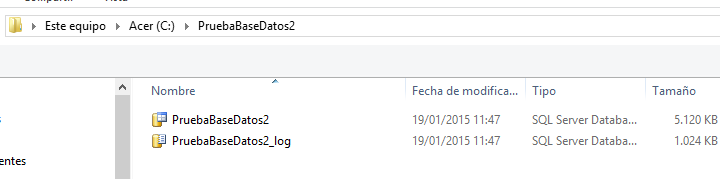
COLLATE Modern\_Spanish\_CI\_AS

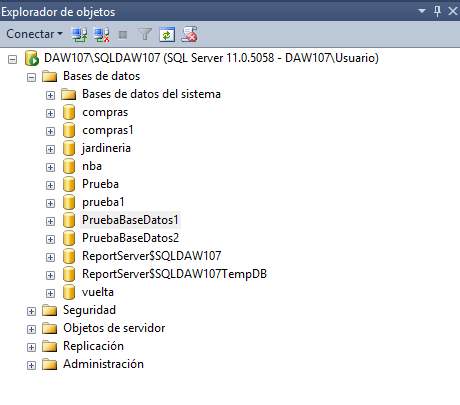


modo grafico (crea una carpeta en c que se llame PruebaBaseDatos2)









if exists(select \* from sysdatabases where name = [PruebaBaseDatos2])

DROP DATABASE [PruebaBaseDatos2] ;

FORMA CORRECTA

if exists(select \* from sysdatabases where name = 'PruebaBaseDatos1')

DROP DATABASE [PruebaBaseDatos1] ;

select \* from sysdatabases ;

OTRO COMANDO:

CREATE DATABASE PruebaBaseDatos1

ON (NAME = 'PruebaBaseDatos1',

FILENAME = 'C:\PruebaBaseDatos1\PruebaBaseDatos1.mdf' ,

SIZE = 5, MAXSIZE = 10000,FILEGROWTH = 10%) LOG ON (NAME = 'PruebaBaseDatos1\_log',

FILENAME = 'C:\PruebaBaseDatos1\PruebaBaseDatos1.LDF' , SIZE = 1, FILEGROWTH = 10%)

COLLATE Modern\_Spanish\_CI\_AS

OTRO COMANDO:

CREATE DATABASE PruebaBaseDatos1

ON (NAME = 'PruebaBaseDatos1',

FILENAME = 'C:\PruebaBaseDatos1\PruebaBaseDatos1.mdf' ,

SIZE = 5MB, MAXSIZE = 10MB,FILEGROWTH = 10%) LOG ON (NAME = 'PruebaBaseDatos1\_log',

FILENAME = 'C:\PruebaBaseDatos1\PruebaBaseDatos1.LDF' , SIZE = 1, FILEGROWTH = 10%)

COLLATE Modern\_Spanish\_CI\_AS

Para modificar la estructura de la base de datos se usa el ALTER

\*Cambiar el nombre y la colación de prueba de base de datos 1. Psg 110. ……..lanzas desde fuera la consulta…no puedes hacerlo desde dentro.

--ALTER DATABASE PruebaBaseDatos1 COLLATE latin1\_general\_ci;--

ALTER DATABASE PruebaBaseDatos1

MODIFY Name=PruebaBaseDato4 ;

**\*BORRAR BASE DATOS 2 DIRECTAMENTE**

DROP DATABASE PruebaBaseDatos2;

CREAR TABLAS

HAY QUIE PONER EN ORDEN LAS TABLAS PORQUE LAS RESTRICCIONES PODRIAN NO SER CUMPLIDAS.

EJEMPLO SI CREO CLIENTES CUENTA, LA TABLA PERTENECE NO PODRIA CREARSE PRIMERO CON LAS CLAVES AJENAS CREADAS PRIMERO

**CREATE DATABASE: ejemplo**

USE nba;

GO

sp\_columns equipos;

**\*RESTRICCIONES donde irian**

create table usuarios (

nombre varchar(30), XXXXXXXXXX

clave varchar(10)

);

XXXXXXXXXXXX

**\*UN EJEMPLO DE RESTRICCIONES.**

create table provincias (

CODIGO TINYINT PRIMARY KEY,

Nombre VARCHAR (10) NOT NULL

)

CREATE TABLE provincias (

CODIGO TINYINT PRIMARY KEY,

Nombre VARCHAR (10) NOT NULL

);

**\*MISMA RESTRICCION A NIVEL DE TABLA**

create table provincias (

CODIGO TINYINT,

Nombre VARCHAR (10)NOT NULL

)

CONSTRAINT PK-PROVINCIAS

PRIMARY KEY (CODIGO)

;

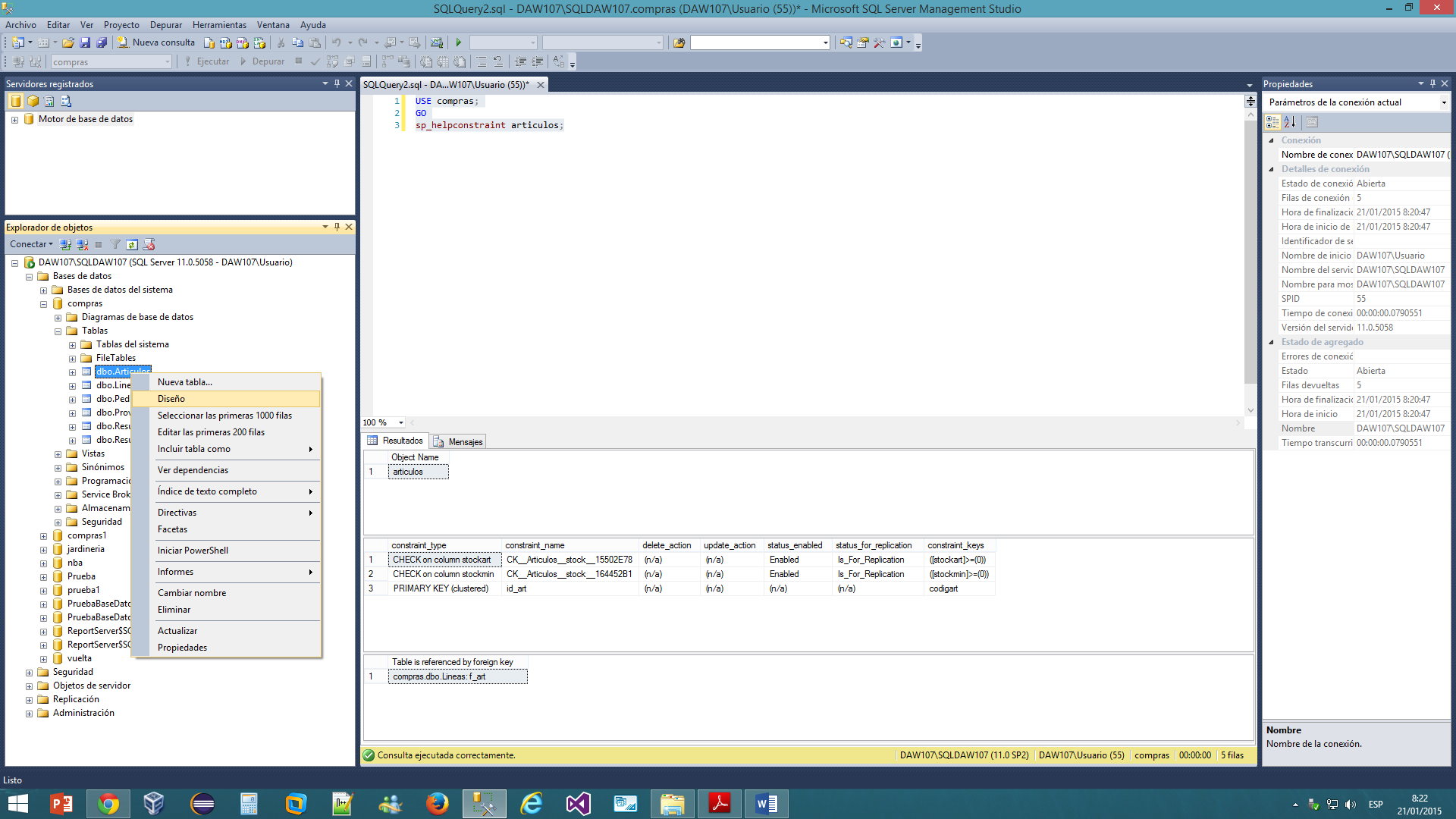
**\*restrinccion para base datos compras**

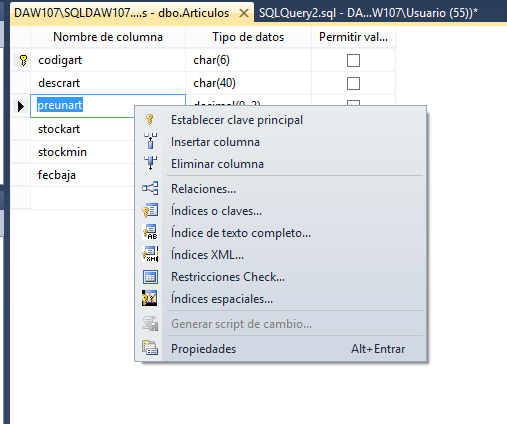
USE compras;

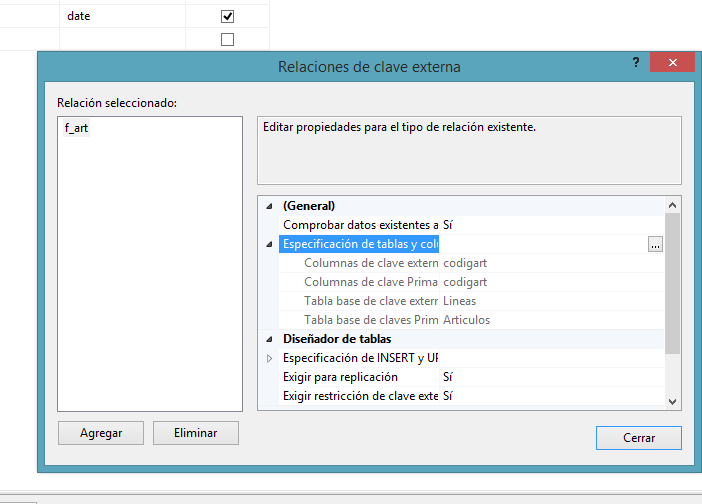
GO

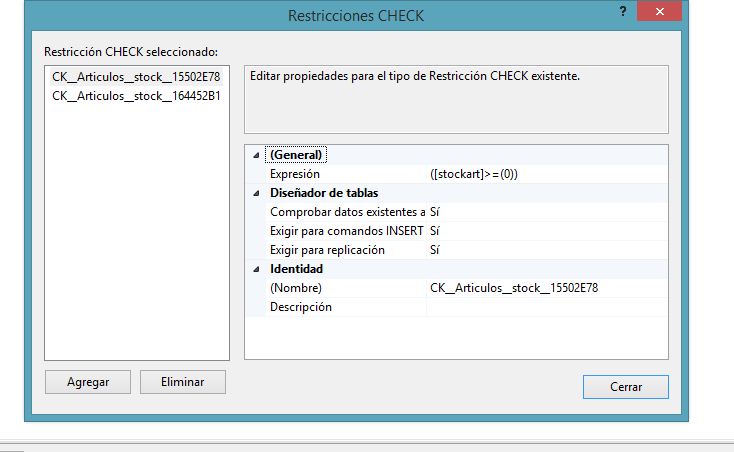
sp\_helpconstraint articulos;

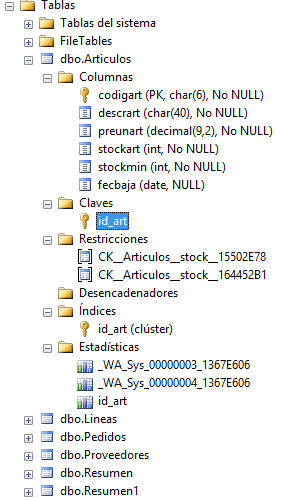
o nivel grafico











\*EJERCICIO PAG 42 CREAR VISTA IGUAL EN BASE DATOS COMPRAS. EJEMPLO PROVEEDORES. VER TABLA Y CON QUE ESTA HECHA.

CREATE TABLE Proveedores1 (

codigpro CHAR(4) NOT NULL CONSTRAINT id\_pro PRIMARY KEY, cifpro CHAR(12) NOT NULL CONSTRAINT u\_cif UNIQUE, nombrpro CHAR(30) NOT NULL,

direcpro CHAR(30) NOT NULL,

cpostpro CHAR(5) NOT NULL CHECK (cpostpro like '[0-9][0-9][0-9][0-9][0-9]'),

localpro CHAR(20) NOT NULL, telefpro CHAR(17) NOT NULL, faxpro CHAR(17),

emailpro CHAR(25),

procepro CHAR(5) NOT NULL CHECK (procepro in ('UE', 'No UE')))

\*CREAR UNA TABLA SOBRE UNA BASE DATOS PRUEBA ,QUE SE LLAME CONDICIONES QUE VA TENER LOS SIGUIENTES CAMPOS:

-DNI

-NOMBRE

-EDAD

-CURSO

CON LOS SIGUIENTES REQUISITOS:

\*EL DNI NO PUEDE SER NULO Y ES CLAVE PRIMARIA.

\*LA EDAD HA DE ESTAR COMPRENDIDA ENTRE 5 Y 20 AÑOS.

\*EL NOMBRE DEBE INTRODUCIRSE EN MAYUSCULAS.

\*EL CURSO SOLO PUEDE ALMACENAR LOS VALORES : 1, 2 o 3. Y NO PUEDE SER NULO.

USE Prueba

GO

CREATE TABLE Condiciones (

Dni CHAR(10) NOT NULL CONSTRAINT Dni PRIMARY KEY,

Edad CHAR (10) NOT NULL,CHECK (Edad like '[5-20]'),

Nombre VARCHAR (60) NOT NULL ,

Curso INT NOT NULL, CHECK (Curso like '[1-3]')

);

DEFINIR CON ALTER Y ANIVEL DE TABALA O COLUMNA

BETWEEN EDAD

CHK\_ MINUSCULA

BORRAR UNA TABLA DE PRUEBA1

USE Prueba1;

GO

DROP TABLE herramientas;

Crear vistas

USE compras;

go

CREATE VIEW dbo.EncabezadoPedido

AS

SELECT dbo.Pedidos.numped,

dbo.Pedidos.fechaped,

dbo.Pedidos.codigpro,

dbo.Pedidos.ivaped,

dbo.Pedidos.fentrped,

dbo.Proveedores.nombrpro,

dbo.Proveedores.direcpro,

dbo.Proveedores.cpostpro,

dbo.Proveedores.localpro,

dbo.Proveedores.telefpro,

dbo.Proveedores.faxpro,

dbo.Proveedores.procepro,

dbo.Proveedores.emailpro,

dbo.Proveedores.cifpro

FROM dbo.Proveedores INNER JOIN

dbo.Pedidos ON dbo.Proveedores.codigpro = dbo.Pedidos.codigpro

luego

USE compras;

SELECT \* FROM EncabezadoPedido;

USE Prueba;

CREATE TABLE cocineros (

nombre VARCHAR(20),

edad INT,

pais VARCHAR (20)

);

insert into cocineros

VALUES

('Pepe',20,'Francia'),

('Juan',20,'Francia'),

('Paco',20,'España'),

('Leron',20,'Francia'),

('Francois',20,'España'),

('Jon',20,'Francia'),

('Pepillo',20,'Francia')

USE Prueba;

SELECT \* FROM cocineros;

CREATE VIEW franceses as SELECT nombre ,pais FROM cocineros

WHERE pais= 'Francia' WITH CHECK OPTION;

USE Prueba;

SELECT \* FROM franceses;

USE Prueba;

insert into cocineros

VALUES

('Maria',20,'España')

USE Prueba;

insert into franceses

VALUES

('Maria',20,'España')

ver

DROP VIEW franceses

**\*CREAR UNA VISTA CON EL NOMBRE Y EL NOMBRE DEL EQUIPO DE LOS JUGAORES DE LOS LAKERS Y CREARLA CON UNA WITH CHECK OPTION**

USE nba;

GO

CREATE VIEW jugadores\_lakers AS

SELECT nombre , Nombre\_equipo

FROM jugadores

WHERE Nombre\_equipo LIKE'Lakers' WITH CHECK OPTION;

**\*crear otra vista sobre esta vista que saque solo los nombres de los jugadores**

USE nba;

GO

CREATE VIEW vista\_jugadores\_lakers AS SELECT nombre FROM jugadores\_lakers;

**\*HACER OTRA MODIFICANDO LOS NOMBRES DE LOS JUAGADORES EN MAYUSCULAS A TRAVES DE LA ULTIMA VISTA QUE HEMOS CREADO**

USE nba;

GO

UPDATE vista\_jugadores\_lakers SET nombre=UPPER(nombre);

USE nba;

select \* from vista\_jugadores\_lakers;

**\*SOBRE LA BASE DE DATOS NBA. CREAR DOS VISTAS, LA PRIMERA CON LOS SIGUIENTES CAMPOS: CODIGO DEL JUGADOR, NOMBRE DEL JUGADOR, NOMBRE DEL EQUIPO Y LA ALTURA.**

**LA SEGUNDA VISTA QUE SOLO PONGA EL CODIGO DEL JUGADOR, EL NOMBRE DEL EQUIPO Y EL NOMBRE DEL JUGADOR.**

**INSERTAR DOS JUGADORES EN ESTA VISTA . DICIENDO SI SE PUEDE O NO, RAZONANDO LA RESPUESTA.**

USE nba;

GO

CREATE VIEW primeraVista\_jugadores AS

SELECT codigo, nombre , Nombre\_equipo , Altura

FROM jugadores;

USE nba;

GO

CREATE VIEW segundaVista\_jugadores AS

SELECT codigo, nombre , Nombre\_equipo

FROM jugadores;

USE nba;

GO

INSERT INTO primeraVista\_jugadores

VALUES

(5008,'Hector Garcia','Timberwolves','6-9');

INSERT INTO segundaVista\_jugadores

VALUES

(5009,'Hector Garcia','Timberwolves');

Para verlos en las tablas teclear:

USE nba;

select \* from segundaVista\_jugadores

order by codigo desc

para verlos en general y los campos generados nulo

USE nba;

SELECT \* FROM jugadores

order by codigo desc;

**\*Sacar una vista con el nombre del jugador, el código y los puntos por partido de cada jugador**

USE nba;

GO

CREATE VIEW terceraVista\_jugadores AS

SELECT jugadores.codigo,jugadores. nombre , Estadisticas.Puntos\_por\_partido

FROM jugadores

INNER JOIN estadisticas ON jugadores.codigo = Estadisticas.jugador;

**\*podría en la terceravista , actualizar los jugadores con sus nombres en minsculas**

USE nba;

GO

UPDATE terceraVista\_jugadores SET nombre=LOWER(nombre);

**\*Crear una vista de aquellos jugadores cuyos puntos por partido son mayores que la media de puntos para todos los partidos.**

--Crear una vista de aquellos jugadores cuyos puntos por partido son mayores que la media de puntos para todos los partidos.--

USE nba;

GO

CREATE VIEW cuartaVista\_jugadores AS

SELECT jugadores. nombre , Estadisticas.Puntos\_por\_partido

FROM jugadores

INNER JOIN estadisticas ON jugadores.codigo = Estadisticas.jugador AND

Estadisticas.Puntos\_por\_partido > (SELECT AVG(Puntos\_por\_partido) FROM estadisticas);

**\*Comprobar**

USE nba;

SELECT \* FROM cuartaVista\_jugadores;

**\*Otra forma**

--Crear una vista de aquellos jugadores cuyos puntos por partido son mayores que la media de puntos para todos los partidos.--

USE nba;

GO

CREATE VIEW quintaVista\_jugadores AS

SELECT b.nombre , a.Puntos\_por\_partido, a.jugador

FROM estadisticas a, jugadores b

WHERE Puntos\_por\_partido >= (SELECT AVG(Puntos\_por\_partido) FROM estadisticas) and b.codigo = a.jugador;

**\*Comprobar**

USE nba;

SELECT \* FROM quintaVista\_jugadores;

**\*De la base de datos de jardinería sacar una lista que informe por nombre del cliente del total de cada uno de sus pedidos**

--\*De la base de datos de jardinería sacar una lista que informe por nombre del cliente del total de cada uno de sus pedidos--

USE jardineria;

GO

CREATE VIEW primeraVista\_ClientesPedidos AS

SELECT Clientes.NombreCliente ,DetallePedidos.CodigoPedido, SUM(CANTIDAD\*PrecioUnidad)

AS Total

FROM Clientes

INNER JOIN Pedidos ON Clientes.CodigoCliente = Pedidos.CodigoCliente

INNER JOIN DetallePedidos ON DetallePedidos.CodigoPedido = Pedidos.CodigoPedido

GROUP BY Clientes.NombreCliente,DetallePedidos.CodigoPedido;

USE jardineria;

GO

SELECT \* FROM primeraVista\_ClientesPedidos;

USE jardineria;

GO

DROP VIEW primeraVista\_ClientesPedidos;