Tema 5

Nombrar Objetos

Tipos de Datos

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.

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

-- Para reutilizarlo debo establecer

SET IDENTITY\_INSERT dbo.herramientas ON

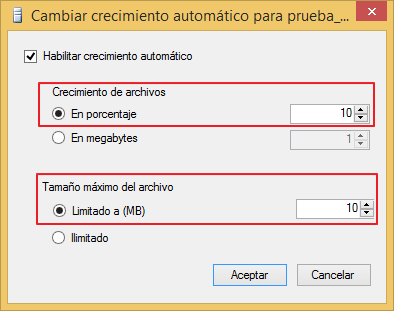
INSERT INTO dbo.herramientas (ID, Nombre) VALUES (1, 'Serrucho');

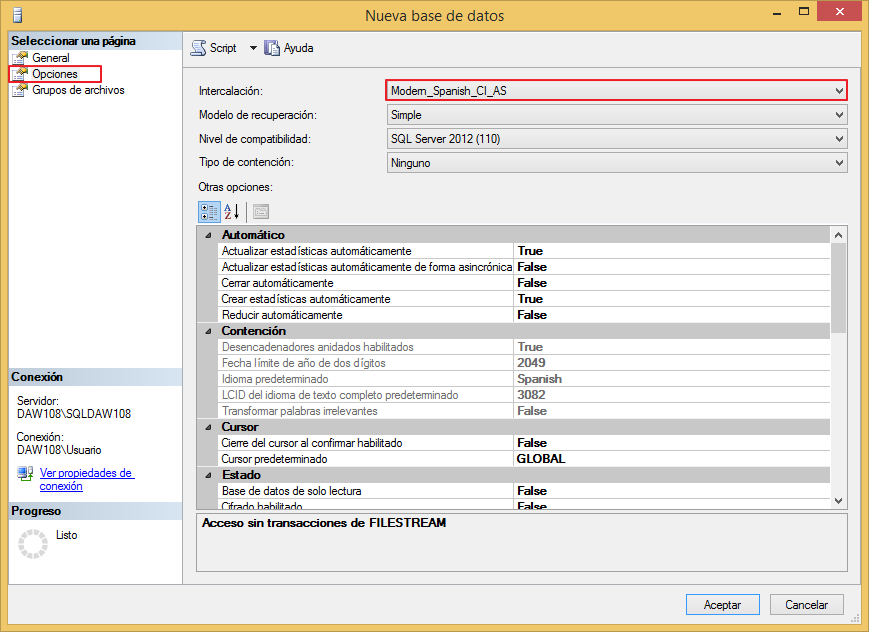
--Crear un BD llamada pruebaBD del tamaño para el fichero de 5M, el máximo tamaño de 10M

-- y un crecimiento del 10%, colación Modern\_Spanish, que no sea CASE SENSITIVE y que

-- entienda los caracteres acentuados, bajo C:

De modo gráfico primero crearemos la carpeta donde alojaremos nuestro dos archivos .mdf y .ldf





Modo comandos:

CREATE DATABASE pruebaBBDD

ON

(NAME=pruebaBBDD\_dat,

FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\pruebaBBDD.mdf',

SIZE=5MB,

MAXSIZE=10MB,

FILEGROWTH= 10%)

LOG ON

(NAME=pruebaBBDD\_log,

FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\pruebaBBDD\_log.ldf',

SIZE = 1,

FILEGROWTH = 10%)

COLLATE Modern\_Spanish\_CI\_AS

Microsoft SQL hace que sea un poco más difícil eliminar una tabla si ya existe. Para lograr esto, necesitarás usar una frase If (condicional sí) y la función EXISTS (existe): IF EXISTS(SELECT 1 FROM sys.objects WHERE OBJECT\_ID = OBJECT\_ID(N'Drop\_Me') AND type = (N'U')) DROP TABLE Drop\_Me (por sus términos en inglés, si existe(seleccionar 1 desde sys.objetcs donde object\_id = object\_id(N'Drop\_Me") y escribe = (N'U')) eliminar tabla Drop\_Me.

IF EXISTS (SELECT \* FROM sys.databases

WHERE name='pruebaBD')

BEGIN

DROP DATABASE pruebaBD;

END

GO

CREATE DATABASE pruebaBBDD

ON

(NAME=pruebaBBDD\_dat,

FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\pruebaBBDD.mdf',

SIZE=5MB,

MAXSIZE=10MB,

FILEGROWTH= 10%)

LOG ON

(NAME=pruebaBBDD\_log,

FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL11.MSSQLSERVER\MSSQL\DATA\pruebaBBDD\_log.ldf',

SIZE = 1,

FILEGROWTH = 10%)

COLLATE Modern\_Spanish\_CI\_AS

--Cambiar el nombre y la colación, prueba a pueba3

USE master;

GO

ALTER DATABASE pruebaBD

MODIFY NAME=pruebaBD2;

USE master;

GO

ALTER DATABASE pruebaBD2

COLLATE Traditional\_Spanish\_CS\_AS;

--Mediante este comando podemos ver qué tipos de datos tiene formada la BBDD.

sp\_columns equipos;

-- Ejemplo diferentes formas para crear una tabla

CREATE TABLE provincias (

codigo TINYINT PRIMARY KEY,

nombre VARCHAR(25) NOT NULL -- Esta restricción solo la podemos hacer a nivel de columna

);

-- A nivel de tabla

CREATE TABLE provincias(

codigo TINYINT,

nombre VARCHAR(25) NOT NULL,

CONSTRAINT PKprovincias -- nombre que le pongo a la restricción

PRIMARY KEY (codigo) -- esta es la clave primaria

);

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Comprobar que las restricciones de la tabla Compras si coinciden con los apuntes de la profesora (pdf pág. 46).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

/\* Crear una tabla condiciones con los siguientes campos y restrcciones:

DNI --> NOT NULL PK

Nombre --> UPPERCASE

Edad --> BETWEEN 5 AND 20

Curso --> CHECK 1,2,3. NOT NULL

\*/

Use prueba;

CREATE TABLE condiciones (

nombre varchar(30),

dni varchar(10) NOT NULL PRIMARY KEY,

edad int,

curso varchar(20) NOT NULL

)

ALTER TABLE condiciones

ADD CONSTRAINT chk\_edad CHECK (edad BETWEEN 5 AND 20),

ADD CONSTRAINT chk\_curso CHECK (curso LIKE '[1-3]')

ADD CONSTRAINT chk\_nombre CHECK nombre UPPER;

Otra forma de hacerlo (probar ambas)

ALTER TABLE condiciones

CONSTRAINT chk\_edad CHECK (edad BETWEEN 5 AND 20),

CONSTRAINT chk\_curso CHECK (curso LIKE '[1-3]')

CONSTRAINT chk\_nombre CHECK (nombre=UPPER(nombre));

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Vistas

Otra forma de crear vistas:

CREATE VIEW NOMBRE\_VISTA (columna1,columnna2...) as consulta

Ejemplo del WITH CHECK OPTION

Use prueba;

CREATE TABLE cocineros (

nombre varchar(20)

edad integer,

pais varchar(20)

);

INSERT INTO cocineros

values ('pepe', 23,'España'),

('Juan',33, 'Francia'),

('Manolo', 56, 'Francia');

CREATE VIEW franceses as

SELECT nombre, pais FROM cocineros

WHERE pais='Francia' WITH CHECK OPTION

INSERT INTO franceses

VALUES ('AMria','España')

--Crear una vista con el nombre de los jugadores y el nombre de los

--equipos de los lakers y establecer con WITH CHECK OPTION

Use nba;

go

CREATE VIEW ejuglakers AS

SELECT Nombre, Nombre\_equipo

FROM jugadores

WHERE Nombre\_equipo LIKE 'Lakers' WITH CHECK OPTION

--Crear una vista sobre esta que solo me saque los nombres de los jugadores

CREATE VIEW nomejuglakers AS

SELECT Nombre FROM ejuglakers

--Consulta que modifique el nombre de los jugadores poniéndolos en mayúsculas a través de la última vista que hemos creado

UPDATE nomejuglakers SET Nombre=UPPER(Nombre);

--Vemos que al ejecutar esta actualizacion sobre la vista nomejuglakers, ha puesto en mayúsculas los nombres de los jugadores. Pero también lo ha hecho en la vista ejuglakers y en la tabla jugadores.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

--Sobre nba, crea dos vistas. La 1ra con los sig campos:

--cod\_jug, nom\_jug, nom\_equipo, altura.

--2da.vista: cod\_jug, nom\_equipo, nom\_jug.

CREATE VIEW vista\_jug1 AS

SELECT codigo, Nombre, Nombre\_equipo, Altura

FROM jugadores

CREATE VIEW vista\_jug2 AS

SELECT codigo, Nombre\_equipo, Nombre

FROM jugadores

--insertar dos nuevos jugadores en estas vistas y decir

-- si se puede o no razonando el por qué.

INSERT INTO vista\_jug1 VALUES (800,'Messi','Suns','6-8');

(1 filas afectadas)

INSERT INTO vista\_jug2 VALUES (805,'Suns','Ronaldo');

(1 filas afectadas)

Si se puede ya que no hay restricciones

-- Vista nom\_jug, cod\_jug y los puntos por partido de cada

jugador

Use nba;

GO

CREATE VIEW v\_puntos AS

SELECT j.Nombre, j.codigo, e.Puntos\_por\_partido

FROM jugadores j, estadisticas e

WHERE j.codigo=e.jugador;

-- Cambiar a minúsculas en nombre de los jugadores en la vista v\_puntos

UPDATE v\_puntos 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

Use nba;

GO

CREATE VIEW avg\_jug2 AS

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

FROM estadisticas a, jugadores b

WHERE Puntos\_por\_partido>(SELECT AVG(Puntos\_por\_partido)

FROM estadisticas) AND b.codigo=a.jugador;

Este esta mal comprobar por que

CREATE VIEW avg\_jug AS

SELECT \*

FROM estadisticas

WHERE Puntos\_por\_partido>(SELECT AVG(Puntos\_por\_partido)

FROM estadisticas)

SELECT CodigoPedido, SUM(PrecioUnidad \* Cantidad) AS Total\_x\_pedido FROM DetallePedidos

GROUP BY CodigoPedido

SELECT \* FROM DetallePedidos

SELECT \* FROM Clientes

SELECT a.CodigoPedido, SUM(PrecioUnidad \* Cantidad) AS Total\_x\_pedido

FROM DetallePedidos a,

GROUP BY CodigoPedido