***PAG 18 DE DEFINICION DE DATOS.PDF***

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

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

para borrar:

DELETE FROM tu\_tabla

WHERE id='34'

SET IDENTITY\_INSERT dbo.herramientas ON

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

este asigna el valor a la columna ID, por ejemplo aqui yo quiero que el serrucho este en la posicion 1.

***pag 110 libro→ MODIFICACIÓN DE UNA BD***

**crea**r una base de datos de tamaño 5mb; máximo tamaño de 10 mb y un incremento de 10%

pag 107 que no distingue de mayor y minúsculas, collate=spanish, y que entienda los caracteres acentuados y que se guarde en una carpeta c:\

despues en modo grafico de nombre: → base de datos2

CREATE DATABASE BASEDEDATOS

ON (NAME=N'BASEDEDATOS',

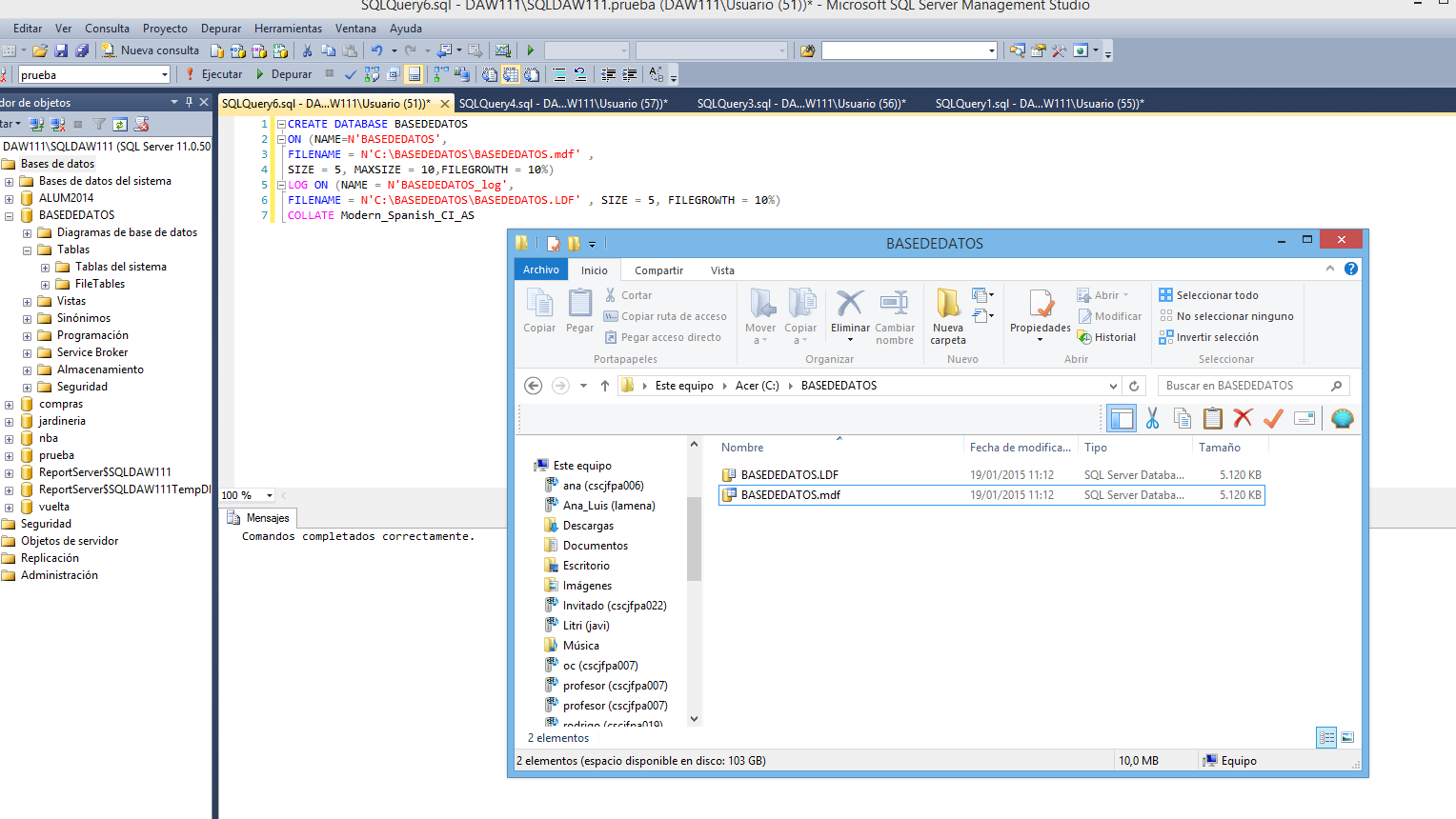
FILENAME = N'C:\BASEDEDATOS\BASEDEDATOS.mdf' ,

SIZE = 5, MAXSIZE = 10,FILEGROWTH = 10%)

LOG ON (NAME = N'BASEDEDATOS\_log',

FILENAME = N'C:\BASEDEDATOS\BASEDEDATOS.LDF' , SIZE = 5, FILEGROWTH = 10%)

COLLATE Modern\_Spanish\_CI\_AS



**dia 20/01/2015**

como comprobar la creación de las carpetas de la base de datos por código:

IF EXISTS(SELECT \* FROM SYS.DATABASES WHERE NAME 'BASEDEDATOS'

BEGIN

DROP DATABASE BASEDEDATOS;

CREATE DATABASE BASEDEDATOS

ON (NAME=N'BASEDEDATOS',

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

SIZE = 5, MAXSIZE = 10,FILEGROWTH = 10%)

LOG ON (NAME = N'BASEDEDATOS\_log',

FILENAME = N'C:\BASEDEDATOS\BASEDEDATOS.LDF' , SIZE = 5, FILEGROWTH = 10%)

COLLATE Modern\_Spanish\_CI\_AS

Cambiar el nombre de la base de datos y la colación a COLLATE latin1\_spanish\_ci:

ALTER DATABASE BASEDEDATOS COLLATE Traditional\_spanish\_ci\_ai;

ALTER DATABASE BASEDEDATOS

MODIFY NAME= PRUEBA2;

GO no es obligatorio ponerlo y las dos cosas de collate y esta de cambio de nombre no se puede hacer a la vez!!!

para borrar:

DROP DATABASE nombre\_de\_la\_basededatos

**CREACIÓN DE TABLAS**

CREATE TABLE nombre\_tabla

( Descripción\_Columna1 [, ... , Descripción\_ColumnaN) ) [, Grupo\_Restricciones\_de\_Tabla ])

ejemplo:

create table usuarios (

nombre varchar(30),

clave varchar(10)

);

Para ver las tablas existentes creadas por los usuarios en una base de datos usamos el procedimiento almacenado "sp\_tables @table\_owner='dbo';":

USE JARDINERIA

GO

sp\_tables @table\_owner='dbo';

Para ver la estructura de una tabla usamos el procedimiento almacenado "sp\_columns" junto al nombre de la tabla:

sp\_columns usuarios;

ejemplo en NBA para la tabla equipos:

USE NBA

sp\_columns equipos;

Creamos una tabla llamada "usuarios" y entre paréntesis definimos los campos y sus tipos:

create table usuarios (

nombre varchar(30),

clave varchar(10)

);

**RESTRICCIONES**

CREATE TABLE provincias(

codigo TINYINT PRIMARY KEY,

nombre VARCHAR(25) NOT NULL);

ahora lo mismo pero a nivel de tabla:

CREATE TABLE provincias(

codigo TINYINT ,

nombre VARCHAR(25) NOT NULL,

CONSTRAINT PKprovincias.codigo este es un nombre que lo inventamos nosotros

PRIMARY KEY (el campo que va ha ser primary key))

**DIA 21/01/2015**

Si no le asignamos un nombre el sistema le asigna uno, lo único que tenemos que buscarlo!!

CREATE TABLE provincias(

codigo TINYINT,

nombre VARCHAR(25) NOT NULL

PRIMARY KEY (codigo)

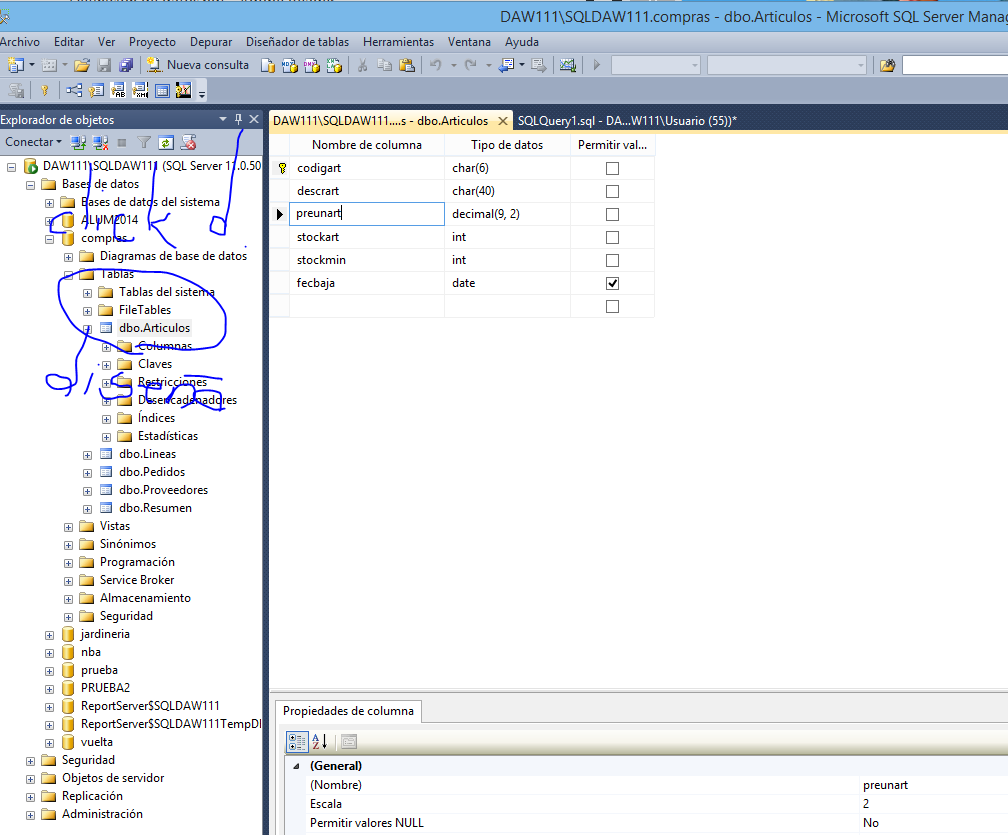
);

El procedimiento almacenado del sistema "sp\_helpconstraint" junto al nombre de la tabla, nos muestra información acerca de las restricciones de dicha tabla.

**sp\_helpconstraint libros;**

**Ejemplo: sp\_helpconstraint Articulos;**

Ahora verlo en modo grafico con botton derecho sobre la tabla→ diseño



hasta página 39 en definición de datos.pdf!

**CAMPOS CALCULADOS**

Un campo calculado es un campo que no se almacena físicamente en la tabla. SQL Server emplea una fórmula que detalla el usuario al definir dicho campo para calcular el valor según otros campos de la misma tabla. pag 41

*desde pag 42 mirar los ejemplos!*

**ALTER TABLE**

Para modificar el diseño de una tabla que ya existe en la base de datos.

*NUEVO CAMPO*:

alter table NOMBRETABLA

add NOMBRENUEVOCAMPO DEFINICION; pag 48

*Para eliminar campos:*

alter table NOMBRETABLA

drop column NOMBRECAMPO; pag 48

crear una tabla(en prueba) que se llama condiciones que va a tener estos campos:DNI, NOMBRE, EDAD, CURSO!

con los siquentes requisitos, el DNI no puede ser nulo y es clave primaria, la edad tiene que ser 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.

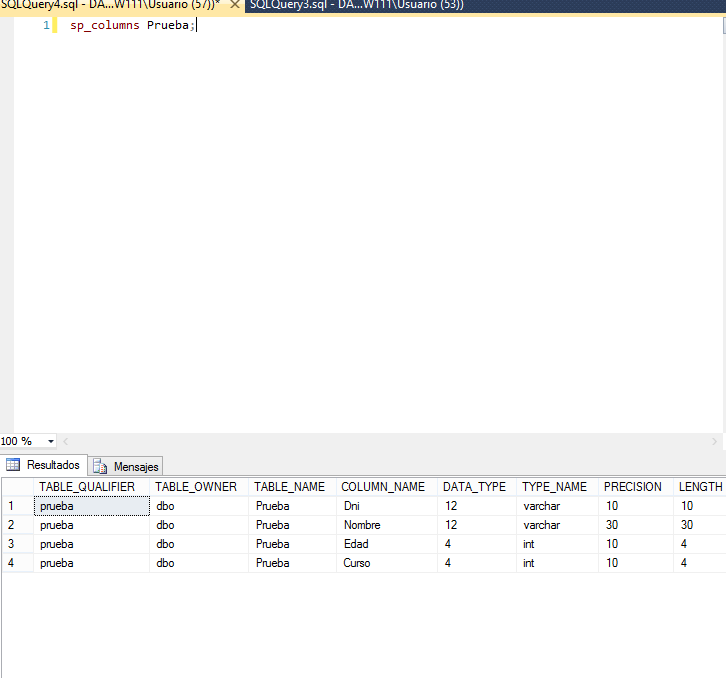
CREATE TABLE Prueba (

Dni VARCHAR(10) NOT NULL PRIMARY KEY,

Nombre VARCHAR(30) CHECK (Nombre=UPPER(Nombre)),

Edad INT CHECK(Edad BETWEEN 5 AND 10),

Curso INTEGER CHECK(Curso BETWEEN 1 AND 3))



CREATE DATABASE banco

ON (NAME=N'banco',

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

SIZE = 5, MAXSIZE = 10,FILEGROWTH = 10%)

LOG ON (NAME = N'BASEDEDATOS\_log',

FILENAME = N'C:\banco\banco.LDF' , SIZE = 5, FILEGROWTH = 10%)

COLLATE Modern\_Spanish\_CI\_AS

CONSTRAINT f\_dni FOREIGN KEY (dni) REFERENCES dni(cliente)

CONSTRAINT f\_cuenta FOREIGN KEY (cod\_cuenta) REFERENCES cod\_cuenta(cuenta)

**DIA 26/01/2015**

Usando la tabla condiciones realizar los siguientes ejercicios:

1. Añadir dos columnas: SEXO (no admite valores nulos) (CHAR(1)) e IMPORTE (TINYINT).

USE prueba;

ALTER TABLE dbo.condiciones ADD SEXO CHAR(1) NOT NULL, IMPORTE TINYINT ;

2. Modificar SEXO a VARCHAR(12) e IMPORTE a INT

ALTER TABLE dbo.condiciones ALTER COLUMN SEXO varchar(12) NULL ;

ALTER TABLE dbo.condiciones ALTER COLUMN IMPORTE INT ;

3. Establecer que el sexo sólo pueda tomar los valores HOMBRE o MUJER

USE prueba;

ALTER TABLE dbo.condiciones ADD CHECK (SEXO LIKE 'HOMBRE' OR SEXO LIKE 'MUJER');

o:

alter table condiciones

ADD

CONSTRAINT ck\_sexo CHECK(sexo in (‘hombre’,’mujer’));

CK\_SEXO ES EL nombre que le asignamos nosotros.

4. Eliminamos las columnas SEXO e IMPORTE

antes:

alter table condiciones DROP constraint ck\_sexo si no hago esto antes no me deja!

USE prueba;

ALTER TABLE dbo.condiciones

DROP COLUMN SEXO, IMPORTE;

5. Eliminamos la restricción de clave primaria de la columna DNI

sp\_helpconstraint condiciones; este comando para ver como lo define y que restricciónes tiene!

ALTER TABLE dbo.condiciones

DROP CONSTRAINT PK\_\_condicio\_\_C035B8DC9434614F;

GO

6. Añadimos la restricción de clave primaria a la columna dni

ALTER TABLE dbo.condiciones

ADD CONSTRAINT PK\_\_condicio\_\_C035B8DC9434614F PRIMARY KEY ( DNI)

GO

7. Se añade la restricción UNIQUE con nombre índice a la columna nombre, se añade la columna codig de tipo TINYINT y se añade la restricción de clave ajena a dicha columnas que referencia al campo código de la tabla provincias .

ALTER TABLE dbo.condiciones

ADD codig TINYINT FOREIGN KEY references herramientas(ID);

ALTER TABLE dbo.condiciones

ADD CONSTRAINT indice UNIQUE (Nombre);

ALTER TABLE condiciones ADD

CONSTRAINT u\_nombre unique (nombre);

ALTER TABLE condiciones ADD

codig tinyint;

ALTER TABLE condiciones ADD

CONSTRAINT fk\_condiciones\_provincias

FOREIGN KEY (codig)

REFERENCES provincias(codigo);

8. Borrar la restricción UNIQUE y la de clave ajena

ALTER TABLE condiciones

DROP CONSTRAINT CK\_\_condicion\_\_Nombr\_\_36B12243

GO

ALTER TABLE condiciones DROP fk\_condiciones\_provincias este tambien

ALTER TABLE condiciones este no hace falta

DROP CONSTRAINT indice

GO

9. Cambiamos el nombre de la tabla a tablanueva;

EXEC sp\_rename 'condiciones', 'tablanueva';

SELECT \* FROM SYS.key\_constraints devuelve las restricciones de PK que tengo en todas las tablas.

**29/01/2015**

UPDATE njuglakers SET

nombre=UPPER(nombre)

USE nba

GO

CREATE VIEW njuglakers as select a.nombre from jugadores //incopmleto

sobre nba, crea 2 vistas; la primera con los siquentes campos: codigodeljugador, nombredlejugador, nombredel equipo y la altura

una segunda: codigodeljugador, nombredelequipo y el nombredeljugador

Insertar 2 nuevos jugadores en estas vistas!

USE nba

GO

CREATE VIEW n1 AS SELECT codigo, nombre, nombre\_equipo, altura FROM jugadores;

GO

CREATE VIEW n2 AS SELECT codigo, nombre, nombre\_equipo FROM jugadores;

GO

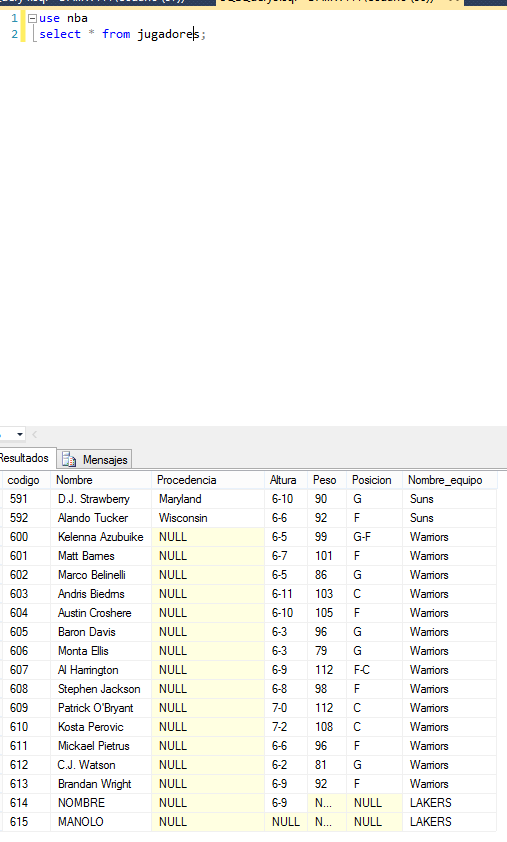
INSERT INTO n1 VALUES(614,'NOMBRE','LAKERS','6-9')

INSERT INTO n2 VALUES(615,'MANOLO','LAKERS')

para borrar:

USE nba

DROP VIEW n1;



Sacar: nombre del jugador, el codigo del jugador, y los puntos por partido de cada jugador:

USE nba

GO

CREATE VIEW n3 AS SELECT nombre, codigo, estadisticas.Puntos\_por\_partido FROM jugadores, estadisticas

where jugadores.codigo=estadisticas.jugador;

actualizar todos los nombres a minusculas de esta view:

USE nba

UPDATE n3 SET

nombre=LCASE(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 avgptos as SELECT b.Nombre, a.Asistencias\_por\_partido, 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

DE la bd de jardineria sacarme una lista que informe por el nombre del cliente del total de cada uno de sus pr¡edidos:

CREATEVIEWFacturaas

selectd.CodigoPedido,c.NombreCliente,sum(d.Cantidad\*PrecioUnidad)asTotalPedido

fromClientesc,Pedidosp,DetallePedidosd

wherec.CodigoCliente=p.CodigoCliente

andp.CodigoPedido=d.CodigoPedido

groupbyc.NombreCliente,d.CodigoPedido;

y�s�gk?

Gabriela

USE jardineria

GO

CREATE VIEW pedazoconsulta AS SELECT Clientes.NombreCliente, Pedidos.CodigoCliente, tabla.suma, tabla.CodigoPedido FROM Pedidos, (SELECT pedidos.codigopedido, SUM(PrecioUnidad\*Cantidad) AS suma FROM DetallePedidos, Pedidos WHERE Pedidos.CodigoPedido=DetallePedidos.CodigoPedido GROUP BY pedidos.codigopedido) AS tabla, Clientes WHERE tabla.CodigoPedido= Pedidos.CodigoPedido AND Pedidos.CodigoCliente=Clientes.CodigoCliente

David

**TRANSACCIONES**

CREATE TABLE test (camp int NOT NULL primary key); esta esta en la base Prueba

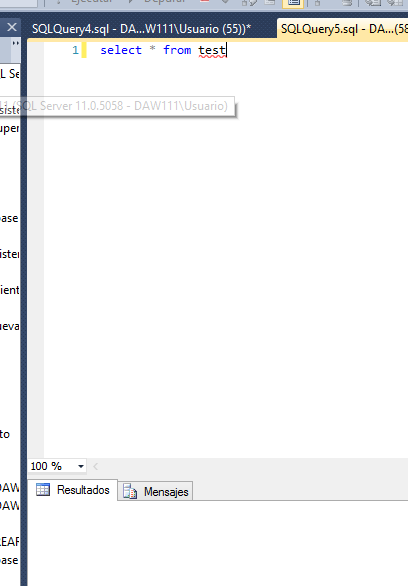
insert into test values(1),(2),(3);

CREATE TABLE test (camp int NOT NULL primary key);

INSERT INTO test values(1),(2),(3);

begin transaction

INSERT INTO test values(4);

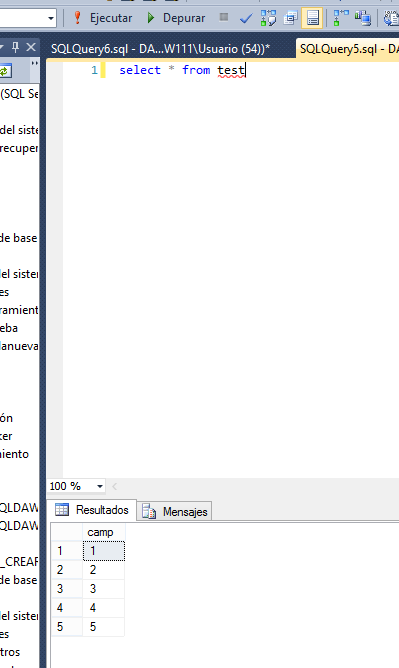
se esta ejecutando al infinito porque no hay commit

begin tran

INSERT INTO test values(5);

INSERT INTO test values(4);

COMMIT;



**USE prueba;**

**BEGIN TRY**

**begin tran**

**INSERT INTO TEST VALUES(7);**

**INSERT INTO TEST VALUES(4);**

**COMMIT**

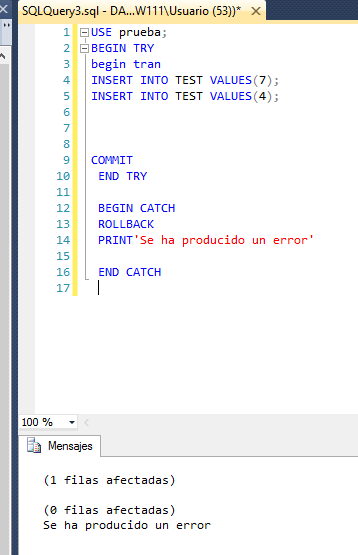
**END TRY**

**BEGIN CATCH**

**ROLLBACK**

**PRINT'Se ha producido un error'**

**END CATCH**



USE prueba;

SET XACT\_ABORT ON

begin tran

INSERT INTO TEST VALUES(7);

INSERT INTO TEST VALUES(4);

COMMIT

si se produce un error

en tiempo de ejecución, SQL Server deshará completamente todas las transacciones

abiertas.

DBCC USEROPTIONS para saber en que nivel de aislamiento esta la base de datos

**MIRAR EL DOCUMENTO WORD: Ejemplos\_ver\_aislamientos.docx**