

TRANSACT-SQL DDL y Funciones

```
CREATE DATABASE nombreBasedeDatos
[ ON [ PRIMARY ] <esp_fichero> [ ,...n ] ]
[ LOG ON < esp_fichero > [ ,...n ] ]
[ COLLATE nbintercalacion ] [;]
<esp_fichero> ::=
(
    NAME = nbfichero_logico ,
    FILENAME = 'nbfichero_fisico'
    [ , SIZE = tamaño [ KB | MB | GB | TB ] ]
    [ , MAXSIZE = { max_size [ KB | MB | GB | TB ] | UNLIMITED } ]
    [ , FILEGROWTH = incremento_crecimiento [ KB | MB | GB | TB | % ] ]
)

DROP DATABASE { nbBasedeDatos } [ ,...n ] [;]

ALTER DATABASE nbBasedeDatos
{
    <cambiar_ficheros>
    | MODIFY NAME = nuevo_nbBasedeDatos
    | COLLATE nbintercalación } [;]
<cambiar_ficheros>::=
{
    ADD FILE < esp_fichero > [ ,...n ] [ TO FILEGROUP { nbgrupo | DEFAULT } ]
    | ADD LOG FILE < esp_fichero > [ ,...n ]
    | REMOVE FILE nbfichero
    | MODIFY FILE < esp_fichero > }
<esp_fichero>::=
(
    NAME = nbarchivo
    [ , NEWNAME = nuevo_nbarchivo ] [ , FILENAME = 'nbarchivo_fisico' ]
    [ , SIZE = tamaño [ KB | MB | GB | TB ] ]
    [ , MAXSIZE = { tamaño_máximo [ KB | MB | GB | TB ] | UNLIMITED } ]
    [ , FILEGROWTH = incremento [ KB | MB | GB | TB | % ] ]
    [ , OFFLINE ] )

CREATE TABLE [ nbBaseDatos.nbEsquema.| nbEsquema.]nbTabla
( { <definicion_columna> | < definicion_colCalc > } [ ,...n ]
  [ <restriccion_tabla> ] [ ,...n ] ) [ ; ]
< definicion_columna > ::=
    nombreCol <tipo_dato> [ COLLATE nbIntercalacion ] [ NULL | NOT NULL ]
    [[CONSTRAINT nbRestriccion ] DEFAULT exp_constante
    | IDENTITY [( semilla, incremento )] [NOT FOR REPLICATION]
    | ROWGUIDCOL ]
    [ <restriccion_columna> [ ...n ] ]
< restriccion_columna > ::=
    [ CONSTRAINT nbRestriccion]
    { PRIMARY KEY | UNIQUE } [ CLUSTERED | NONCLUSTERED ]
    |[FOREIGN KEY] REFERENCES [ nbEsquema.] nbTablaPadre [( col_padre )]
        [ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
        [ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
    |CHECK (expresion_validacion)
< restriccion_tabla > ::=
    [ CONSTRAINT nbRestriccion]
    { PRIMARY KEY | UNIQUE } [CLUSTERED | NONCLUSTERED] (nbCol[ASC|DESC][ ,...n ])
    | FOREIGN KEY(nbCol[,...n])REFERENCES nbTablaPadre [(col_padre)[ ,...n ])
        [ ON DELETE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
        [ ON UPDATE { NO ACTION | CASCADE | SET NULL | SET DEFAULT } ]
    | CHECK (expresion_validacion )
< definicion_colCalc > ::=
    nbCol AS expresion [PERSISTED [NOT NULL] [<restriccion_columna> [...n ] ] ]

DROP TABLE [nbBaseDatos.nbEsquema.|nbEsquema.]nbTabla[ ,...n ] [ ; ]
```

```

ALTER TABLE [nbBaseDatos].[nbEsquema].| nbEsquema.]nbTabla
    {ALTER COLUMN nbColumna {<tipo_dato> [NULL|NOT NULL][COLLATE nbIntercalacion]
    | {ADD|DROP} {ROWGUIDCOL | PERSISTED} }
    | [WITH {CHECK|NOCHECK}] ADD
        { <definicion_columna>
        | <definicion_colCalc>
        | <restriccion_tabla>
        } [ ,...n ]
    | DROP
        { [CONSTRAINT] nbRestriccion
        | COLUMN nbColumna
        } [ ,...n ]
    | {CHECK|NOCHECK} CONSTRAINT {ALL|nbRestriccion[ ,...n ]}
    | {ENABLE|DISABLE} TRIGGER {ALL | nbTrigger [ ,...n ] } } [ ; ]
CREATE VIEW [nbEsquema.] nbVista [(columna [,...n ])] AS (sentencia_select) [ ; ]

DROP VIEW [nbEsquema.]nbVista[ ,...n ] [ ; ]

CREATE [UNIQUE] [CLUSTERED|NONCLUSTERED] INDEX nombre_indice
    ON <objeto> (columna [ASC | DESC] [,...n ] ) [ ; ]

<objeto> ::=
    { [nbBaseDatos].[nbEsquema].| nbEsquema.]nbTablaVista }
DROP INDEX <indice> [ ,...n ] [ ; ]
<indice>::=
    { nombreindice ON [nbBaseDatos.nbEsquema.|nbEsquema.]nbTablaVista }

```

Tipos de datos

xml
date **datetime**
smalldatetime
float [(n)]
real
decimal [(p[,s])]

smallmoney
bigint
int
smallint
tinyint
bit
timestamp
uniqueidentifier

nvarchar [(max)]
nchar [(max)]
varchar [(max)]
char [(max)]
varbinary [(n|max)]
binary [(n)]

Intercalaciones

Latin1_General_BIN

Latin1_General_{CS|CI}_{AS|AI}

Modern_Spanish_{CS|CI}_{AS|AI}

Funciones

GETDATE ()	DATEPART (unidad, fecha)	DAY (fecha)
MONTH (fecha)	YEAR (fecha)	DATENAME (unidad, fecha)
SET DATEFIRST número	DATEADD (unidad, número, fecha)	
DATEDIFF (intervalo, fechainicial, fechafinal)	GETUTCDATE ()	
<unidad>::={year yy yyyy} {quarter qq q} {month mm m} {dayofyear dy y} {day dd d} {week ww wk} {weekday dw w} {hour hh} {minute mi n} {second ss s} {millisecond ms}		
DATEFROMPARTS (año, mes, día)		
ASCII (cadena)	CHAR (numero)	NCHAR (numero)
LEN (Cadena)	LTRIM (Cadena)	RTRIM (Cadena)
UNICODE (cadena)	LEFT (cadena, long)	RIGHT (cadena, long)
LOWER (Cadena)	UPPER (Cadena)	QUOTENAME ('cadena'[, 'caracter'])
SPACE (número)	SUBSTRING (cadena, ini, long)	STR (número[, long[, decimales]])
REVERSE (cadena)	REPLICATE (cadena, long)	PATINDEX ('%patrón%', cadena)
STUFF (cadena_origen, ini, long, cadena_sustituta)		
REPLACE (cadena_original, cadena_a_quitar, cadena_a_poner)		
CHARINDEX (cadena_a_buscar, cadena_origen[, ini])		
CONCAT (cadena1, cadena2 [, ...])		
ROUND (expresion_numerica, long[, funcion])		
CAST (expresion AS tipo_dato)		FORMAT (valor, formato[, cultura])
CONVERT (tipo_dato, expresion[, estilo])		
CASE exp_entrada {WHEN valor THEN resultado}[...n][ELSE resultado_sino] END		
CASE {WHEN condicion THEN resultado}[...n][ELSE resultado_sino] END		
IIF (condicion, valor_siverdadero, valor_else)	CHOOSE (indice, valor1, valor2[, ...])	
ISNULL (expresion, valor)	COALESCE (expresion[, ...n])	