

## Introducción al Entorno/Teoría y Práctica sobre el Lenguaje SQL

MS-SQLSERVER es un motor o entorno para tratar Base de Datos Relacionales (SGBDR o RDBMS en Inglés).

Reglas principales del Modelo Relacional:

Unicidad.

Identificador

Clave Primaria.

Clave Secundaria

Integridad Diferencial

Clave Extranjera o Foráneas (Foreign key)

Base de Datos ComercialTB.

Tabla Clientes:

NUMEROCLI	NOMBRE	PROVINCIA
15	Sánchez S.A.	Sevilla
20	Mrt. Gonz	Madrid
35	Contr Alme, S.A.	Sevilla
138	Contr Solm, S.A.	Santander

Tabla Pedidos:

NUMEROPDO	FECHA	NUMEROCLI	ESTADO
1210	07/08/2017	15	E
1230	07/07/2017	35	E
1301	05/04/2017	15	P
1280	01/03/2017	20	EP
1150	18/02/2017	15	E
1250	02/01/2017	35	P

Tabla Clientes:

NUMEROCLI: Será la Clave Primaria.

PROVINCIA : Identificador clave Secundaria Clientes

Tabla Pedidos:

NUMEROPDO : Identificador clave Primaria Pedidos.

NUMEROCLI : Clave Extranjera de Pedidos que hace referencia a NUMEROCLI de Clientes.

Administrar una Base de Datos.(Desde el Entorno, o desde Transact SQL T-SQL).

Gestionar el Espacio de Almacenamiento.

Configurar la Base de Datos.

Gestionar los Objetos de la Base de Datos.

Traducción de las restricciones de análisis.  
Gestionar la seguridad de acceso.  
Copias de Seguridad.

---

\* Estructura del Lenguaje SQL.

DDL(Data Definition Language).  
DML(Data Manipulation Language).  
Órdenes de Control de Transacciones.  
Órdenes de Control de Sesión.  
Órdenes de Control de Sistema.

-----  
SubLenguaje DDL.Creación, Manipulación de Objetos y Derechos y/o  
Permisos en Base de Datos.  
-----

CREATE (Creación de Objetos).  
DROP (Borrado de Objetos).  
ALTER (Alteración o Modificación de Objetos Existentes).  
EXECUTE (Ejecución de Código).  
GRANT (asignación de Permisos).  
REVOKE (Revocación de Permisos).

SubLenguaje DML.Permite la Manipulación de los Datos Contenidos en  
Una Tabla.(Borrar,Insertar,Modificar,Etc.).  
-----

SELECT (Consulta y Selección de Elementos)  
INSERT (Insertar nuevos datos o Elementos).  
DELETE (Borrado de Elementos).  
UPDATE (Modificado de Elementos).

Crear una Base de Datos: Sintaxis.

CREATE DATABASE nombrebase [ON [PRIMARY]  
[[[ NAME=nombreLogico, ]  
FILENAME='nombrefisico'  
[, SIZE=tamaño]  
[,MAXSIZE= { tamañomaximo | UNLIMITED } ]  
[, FILEGROWTH = valorincrementobloques de 64KB.] ) [,...]] ' 1 megabyte archivo  
datos 10% arch. traza  
[ LOG ON { archivo } ]  
[COLLATE nombreclasificacion]  
[ FOR ATTACH | FOR ATTACH\_REBUILD\_LOG ]

---

```

CREATE DATABASE TutorialDB
ON PRIMARY(
    name=TutorialDB_data,
    filename='D:\CursoSqlServer2017\Data\TutorialDB.mdf',
    size=6MB,
    maxsize=15MB,
    filegrowth=1MB)
LOG ON(
    name=gescom_log,
    filename='D:\CursoSqlServer2017\Data\TutorialDB_log.ldf',
    size=2MB,
    maxsize=2MB,
    filegrowth=0MB
)
COLLATE modern_spanish_ci_as;

```

---

**Name** nombre lógico del archivo.

**FILENAME** Ubicación y nombre fijo del Archivo.

**SIZE** Tamaño inicial del Archivo.

**MAXSIZE** Tamaño Máximo del Archivo. Defecto Mega Bytes.

**UNLIMITED** Sin Tamaño Máximo. El Tamaño máximo es la capacidad física del Disco Duro

**FILEGROWTH** Indica paso de Incremento para el Tamaño del Archivo. Si el valor no se especifica el incremento es de 1 MegaByte para archivos de Datos y el 10% sobre los archivos de trazas.

**LOG ON** Ubicación del Archivo de Trazas de las transacciones. Archivo que se utiliza para registrar todas las operaciones realizadas sobre la Base de Datos. Se utiliza para recuperar datos en caso de error.

**COLLATE** Indica la Clasificación de la Base de Datos.

**FOR ATTACH** Permite crear una Base de Datos usando Archivos ya creados. La compatibilidad de los archivos debe ser de un 90%.

**FOR ATTACH\_REBUILD\_LOG**

Crear una Base de Datos adjuntándole los archivos mdf ndf los archivos de traza los log se crean vacíos.

---

**Ficheros .mdf** Los MDF es donde están toda la información de la base. Contiene las filas, columnas, campos y datos creados por una aplicación o usuario. La creación de una columna de base de datos, las modificaciones e información de la creación de los registros, todo está almacenado en este archivo para el uso las aplicaciones y de búsquedas.

**Ficheros .ldf.** Son la archivos de registro de transacciones.

El propósito primario de un archivo LDF es proveer el concepto ACID – Atomicity (Atomicidad), Consistency (Consistencia), Isolation (Aislamiento) y Durability (Durabilidad).

**Atomicidad:** Si una parte de la transacción falla, la transacción entera falla, y el estado de la base de datos se queda sin cambios.

**Consistencia:** Cualquier transacción lleva la base de datos de un estado válido a otro.

**Aislamiento:** La ejecución de transacciones concurrentes lleva la base de datos a un estado como si las transacciones fueran ejecutadas en serie, una por una.

**Durabilidad:** Una vez aplicada, la transacción permanece así, incluso en caso de errores, pérdida de energía, o bloqueos.

Este tipo de archivos poseen toda la información necesaria para recomponer una Base de Datos dañada.

## PRIMARY

Especifica que la lista de elementos **filespec** asociada define el archivo principal. El primer archivo especificado en la entrada **filespec** del grupo de archivos principal se convierte en el archivo principal.

---

## Modificar Tamaño.

Sintaxis:

```
ALTER DATABASE nombre
MODIFY FILE
(NAME=nombrelogico
[, SIZE=TAMAÑO]
[,MAXSIZE=TAMAÑO]
[,FILEGROWTH=valorincremento])
```

---

```
ALTER DATABASE TutorialDB
MODIFY FILE (
name=TutorialDB_data,
size=30MB);
```

**ALTER DATABASE {NOMBREBASE | CURRENT} MODIFY NAME=NUEVONOMBREBASE**

La palabra clave **CURRENT** le dice a alter que se ejecute en la Base de Datos que se ejecuta la instrucción

---

## Estado de la Base de Datos

Sintaxis:

**ALTER DATABASE { NombreBase | CURRENT} SET opción.**

**ONLINE** permite hacer visible la Base de Datos

**OFFLINE** Permite hacer inaccesible la Base de Datos. La base de datos se para y se cierra correctamente.

**EMERGENCY** La base de Datos sólo es accesible en modo lectura, el registro de traza esta desactivado y sólo se permite el acceso a los administradores.

#### Acceso

**SINGLE\_USER** Acceso limitado a un único usuario.

**RESTRICTED\_USER** Solo se pueden conectar a la base de datos los miembros con Roles db\_owner, dbcreator o sysadmin.

**MULTI\_USER** Este es el modo predeterminado. Permite acceder a los usuarios con permisos suficientes.

#### Operaciones Posibles

**READ\_ONLY** La base de datos solo está disponible en modo lectura.

**READ\_WRITE** La Base de datos sólo está disponible en modo lectura/escritura.

#### Gestión de Transacciones

**ROLLBACK AFTER nombre** La anulación de la transacción es efectiva después de unos segundos de espera.

**ROLLBACK IMMEDIATE** La anulación de la transacción es inmediata.

---

#### **Eliminar una Base de Datos.**

**DROP DATABASE nombreBasededatos**

**Los archivos físicos también se eliminan.**

---

#### **Tipos de Datos:**

Caracteres:

**char[(n)]** Cadena de caracteres de longitud fija de n caracteres como máximo de 1 a 8.000.

**varchar(n | max)** Cadena de caracteres longitud variable de n caracteres como máximo. 1 a 8.000

**nchar[(n)]** Cadena de caracteres Unicode como máximo 4.000

**nvarchar(n | max)** Cadena de caracteres Unicode como máximo 4.000.

Numéricos:

**decimal [(p,d)]** Numérico exacto con precisión p(número total de cifras) con d cifras a la derecha de la coma.

P= 1 a 38 18 por defecto

Ejemplo: decimal(8,3) 99999,999 a -99999,999

**numeric [(p[,d])]** Idéntico o muy parecido al decimal.

**int** Número entero. -2/31 (-2.147783.648) y +(2.147.483.647)

**tinyint** Número entero positivo entre 0 y 255.

**float[(n)]** Numérico aproximado de n cifras con n entre 1 y 53.

**real** Idéntico a float(24)

**money** Numérico formato moneda desde -922.337.203.685.477,5.808 y +922.337.203.685.477,5.807 (8Bytes).

Fechas y Hora:

**datetime** Permite almacenar una fecha y una hora. Desde 1.753 hasta 9.999

**smalldatetime** Almacenar Fecha y Hora 1.900 al 2.079 .

**datetime2** Fecha y Hora desde 01/01/0001 a 31/12/9.999

**datetimeoffset** Fecha y Hora 01/01/0001 al 31/12/9.999 la información horaria se almacena en formato UTC. Y la diferencia horaria se conserva para encontrar la hora local indicada inicialmente.

**date** Fecha 01/01/0001 al 31/12/9.999

**time** Hora dato Positivo inferior a 24:00

---

Sub-lenguaje DML. (Sentencia SQL de Manipulación de Tablas.)

**CREATE TABLE**  
**SELECT \* O [CAMPOS....]**  
**INSERT INTO**  
**DROP TABLE**  
**ALTER TABLE**

---

USE TutorialDB

CREATE TABLE TABLA02  
    (coddepar INT NOT NULL,  
      codiarea INT NOT NULL,  
      nemplead INT NOT NULL,  
      salarioe decimal(10,2) NOT NULL,  
      CONSTRAINT pk\_ekjemplo3 PRIMARY KEY (coddepar,codiarea)  
    );

USE TutorialDB

CREATE TABLE Tabla03  
    (empno INT CONSTRAINT Itabla03 PRIMARY KEY,  
      nombre NVARCHAR(15),  
      profesion NVARCHAR(19),  
      fechaalta DATE,  
      salario DECIMAL(10,2),  
      num\_segsocial INT CONSTRAINT Itabla031 UNIQUE  
    );

USE TutorialDB

CREATE TABLE Tabla04  
    (empno INT CONSTRAINT Itabla04 PRIMARY KEY,  
      nombre NVARCHAR(15),  
      profesion NVARCHAR(19),  
      fechaalta DATE,  
      salario DECIMAL(10,2),  
      num\_segsocial INT CONSTRAINT Itabla041 UNIQUE NONCLUSTERED  
    );

---

INSERT

-- Abrir Base de Datos Tutorial DB

USE TutorialDB

-- Crear nueva Tabla llamada 'Customers' con Chema 'dbo'

-- La tabla si existe la borramos

GO

IF OBJECT\_ID('dbo.Tabla01', 'U') IS NOT NULL

-- Insertamos datos en la Tabla

INSERT INTO dbo.Tabla01

([CustomerId],[Nombre],[Lugar],[Email],[sueldo])

VALUES

( 80, N'Juan Jose', N'Albacete', N'No tiene',100.20),

( 200, N'Katia', N'Rusia', N'katia@adventure-works.com',1000.45),

( 300, N'Francisco Garcia', N'Salamanca', N'fjose@adventure-works.com',2000.56),

( 400, N'Joan Vilanova', N'Barcelona', N'jvilanova@adventure-works.com',3000.54)

GO

-- Visualizamos todos los registros de la Tabla 'Customers'

SELECT \* FROM dbo.Tabla01;

---