**NOTAS TP GDD**

------------------------------------------------------------------

rows count

>SELECT COUNT(\*) FROM [GD2C2014].[gd\_esquema].Maestra

**548755**

------------------------------------------------------------------

información de espacio en disco para la tabla Vendor y sus índices.

> USE GD2C2014;

GO

EXEC sp\_spaceused N'gd\_esquema.Maestra';

GO

**name rows reserved data index\_size unused**

**1 Maestra 548755 627336 KB 627208 KB 8 KB 120 KB**

|  |
| --- |
| **declare** @nombre **varchar**(50)-- declare declara una variable  -- @nombre es el identificador de la  -- variable de tipo varchar **set** @nombre = 'www.devjoker.com' -- El signo = es un operador  -- www.devjoker.com es un literal **print** @Nombre -- Imprime por pantalla el valor de @nombre.   -- No diferencia mayúsculas ni minúsculas |

Scripts y lotes.

    Un script de **Transact SQL** es un conjunto de sentencias de **Transact SQL** en formato de texto plano que se ejecutan en un servidor de **SQL Server**.

    Un script está compuesto por uno o varios lotes. Un lote delimita el alcance de las variables y sentencias del script. Dentro de un mismo script se diferencian los diferentes lotes a través de las instrucción **GO.**

|  |
| --- |
| -- Este es el primer lote del script  **SELECT** \* **FROM** COMENTARIOS  **GO** -- GO es el separador de lotes  -- Este es el segundo lote del script **SELECT** getdate**()** -- getdate() es una función integrada que devuelve  -- la fecha  **GO**  Indica a las herramientas de Microsoft® SQL Server™ el final de un lote de instrucciones Transact-SQL.  En ocasiones es necesario separar las sentencias en varios lotes, porque **Transact SQL** no permite la ejecución de ciertos comandos en el mismo lote, si bien normalmente también se utilizan los lotes para realizar separaciones lógicas dentro del script.  "go" es un signo de finalización de un lote de sentencias. No es una sentencia, es un comando. El lote de sentencias está compuesto por todas las sentencias antes de "go" o todas las sentencias entre dos "go".  Las sentencias no deben ocupar la misma linea en la que está "go".  Habrá notado que no se puede ejecutar un procedimiento almacenado luego de otras sentencias a menos que se incluya "execute" (o "exec").  Por ejemplo, si tipeamos:  **select \*from empleados;**  **sp\_helpconstraint empleados;**  muestra un mensaje de error porque no puede procesar ambas sentencias como un solo lote. Para que no ocurra debemos tipear:  **select \*from empleados;**  **exec sp\_helpconstraint empleados;**  o separar los lotes con "go":  **select \*from empleados;**  **go**  **sp\_helpconstraint empleados;**  Las siguientes sentencias no pueden ejecutarse en el mismo lote: create rule, create default,create view, create procedure, create trigger. Cada una de ellas necesita ejecutarse separándolas con "go". Por ejemplo:  **create table....**  **go**  **create rule...**  **go**  Recuerde que si coloca "go" no debe incluir el "punto y coma" (;) al finalizar una instrucción  **USE AdventureWorks2012;**  **GO**  **DECLARE @NmbrPeople int**  **SELECT @NmbrPeople = COUNT(\*)**  **FROM Person.Person;**  **PRINT 'The number of people as of ' +**  **CAST(GETDATE() AS char(20)) + ' is ' +**  **CAST(@NmbrPeople AS char (10));**  **GO**  **DESPUES DE EJECUTAR…**  > USE [GD2C2014]  GO  CREATE SCHEMA [LOS\_DESCONOCIDOS] AUTHORIZATION [gd]  GO  **EJECUTANDO…**  > SELECT \*  FROM INFORMATION\_SCHEMA.SCHEMATA WHERE SCHEMA\_NAME = 'LOS\_DESCONOCIDOS';  **SE OBTIENE…**  **CATALOG\_NAME, SCHEMA\_NAME, SCHEMA\_OWNER, DEFAULT\_CHARACTER\_SET\_CATALOG, DEFAULT\_CHARACTER\_SET\_SCHEMA, DEFAULT\_CHARACTER\_SET\_NAME**  **GD2C2014, LOS\_DESCONOCIDOS, gd, NULL, NULL, iso\_1** |