

#CrecimientoProfesional
#AprendeConLosPioneros

Online |  Perú



CAPACITACIÓN
PROFESIONAL

SESIÓN IX

Docente: Victor Gutierrez
Data Architect

Agenda

Funciones personalizadas

Objetivo: Manejo de funciones con el objetivo de crear funciones personalizadas.

- Concepto y tipo de funciones.
- Creación, ejecución y eliminación de funciones.

Functions

Creación de funciones

```
-- Transact-SQL Scalar Function Syntax
```

```
CREATE [ OR ALTER ] FUNCTION [ schema_name. ] function_name  
( [ { @parameter_name [ AS ] [ type_schema_name. ] parameter_data_type  
  [ = default ] [ READONLY ] }  
  [ ,...n ]  
  ]  
)  
RETURNS return_data_type  
  [ WITH <function_option> [ ,...n ] ]  
  [ AS ]  
BEGIN  
  function_body  
  RETURN scalar_expression  
END  
[ ; ]
```

```
-- Transact-SQL Inline Table-Valued Function Syntax
```

```
CREATE [ OR ALTER ] FUNCTION [ schema_name. ] function_name  
( [ { @parameter_name [ AS ] [ type_schema_name. ] parameter_data_type  
  [ = default ] [ READONLY ] }  
  [ ,...n ]  
  ]  
)  
RETURNS TABLE  
  [ WITH <function_option> [ ,...n ] ]  
  [ AS ]  
RETURN [ ( ) select_stmt [ ) ]  
[ ; ]
```

Functions

Modificación de una función

- Modifica una función Transact-SQL o CLR existente, creada anteriormente por medio de la ejecución de la instrucción CREATE FUNCTION, sin cambiar los permisos y sin que afecte a ninguna otra función, procedimiento almacenado o desencadenador dependiente.

```
-- Transact-SQL Scalar Function Syntax
ALTER FUNCTION [ schema_name. ] function_name
( [ { @parameter_name [ AS ] [ type_schema_name. ] parameter_data_type
    [ = default ] }
  [ ,...n ]
]
)
RETURNS return_data_type
    [ WITH <function_option> [ ,...n ] ]
    [ AS ]
BEGIN
    function_body
    RETURN scalar_expression
END
[ ; ]
```

Functions

Eliminación de una función

- Quita una o más funciones definidas por el usuario de la base de datos actual. Las funciones definidas por el usuario se crean mediante CREATE FUNCTION y se modifican con ALTER FUNCTION.
- La función DROP admite funciones escalares definidas por el usuario

```
DROP FUNCTION [ IF EXISTS ] { [ schema_name. ] function_name } [ ,...n ]  
[;]
```

```
DROP FUNCTION IF EXISTS sf_get_Cliente;
```

Functions

Ejecución de una función

```
USE [AdventureWorks2016CTP3]
GO

-- Declare a variable to return the results of the function.
DECLARE @ret nvarchar(15);

-- Execute the function while passing a value to the @status parameter
EXEC @ret = dbo.ufnGetSalesOrderStatusText
    @Status = 5;

-- View the returned value. The Execute and Select statements must be executed at the same time
SELECT N'Order Status: ' + @ret;

-- Result:
-- Order Status: Shipped
```

Agenda

Manejo de vistas

Objetivo: Manejo de Vistas para la optimización o distribución de información.

- Creación de vistas y visitas materializadas.
- Eliminación y modificación de vistas.
- Vistas indexadas. Modificar datos mediante una vista.

Vistas

- Crea un objeto cuyo resultado es una tabla virtual cuyo contenido (columnas y filas) tiene como origen un script
- Para darle una capa de presentación al modelo físico de la Base de datos
- Para encapsular el origen de tablas y columnas; de esta manera se permite asegurar el acceso a los datos
- No ocupa espacio, los datos están almacenados en las tablas que esta utiliza

```
-- Syntax for SQL Server and Azure SQL Database

CREATE [ OR ALTER ] VIEW [ schema_name . ] view_name [ (column [ ,...n ] ) ]
[ WITH <view_attribute> [ ,...n ] ]
AS select_statement
[ WITH CHECK OPTION ]
[ ; ]

<view_attribute> ::=
{
    [ ENCRYPTION ]
    [ SCHEMABINDING ]
    [ VIEW_METADATA ]
}
```

Vistas

Create View

No puede incluir en una vista:

- Una cláusula ORDER BY, a menos que también haya una cláusula TOP en la lista de selección de la instrucción SELECT
- La palabra clave INTO
- La cláusula OPTION
- Una referencia a una tabla temporal o a una variable de tabla

ENCRYPTION

Cifra las el contenido de **sys.syscomments** que contienen el texto de la instrucción CREATE VIEW. El uso de WITH ENCRYPTION evita que la vista se publique como parte de la replicación de SQL Server.

Vistas

Alter View

Modifica una vista creada anteriormente. Esto incluye una vista indexada. ALTER VIEW no afecta a los desencadenadores o procedimientos almacenados dependientes y no cambia los permisos.

```
ALTER VIEW [ schema_name . ] view_name [ (
column [ ,...n ] ) ]
[ WITH <view_attribute> [ ,...n ] ]
AS select_statement
[ WITH CHECK OPTION ] [ ; ]
```

```
<view_attribute> ::=
{
    [ ENCRYPTION ]
    [ SCHEMABINDING ]
    [ VIEW_METADATA ]
}
```

Vistas

Drop View

Elimina una o más vistas de la base de datos actual. DROP VIEW se puede ejecutar en vistas indexadas.

```
DROP VIEW [ IF EXISTS ] [ schema_name . ]  
view_name [ ...,n ] [ ; ]
```

Vistas

Create View (Actualizable)

- Es posible modificar los datos de una tabla base subyacente mediante una vista, siempre que se cumplan las siguientes condiciones:
- Cualquier modificación, incluidas las instrucciones UPDATE, INSERT y DELETE, debe hacer referencia a las columnas de una única tabla base.
- Las columnas que se vayan a modificar en la vista deben hacer referencia directa a los datos subyacentes de las columnas de la tabla. Las columnas no se pueden obtener de otra forma, como las siguientes:
- Una función de agregado: AVG, COUNT, SUM, MIN, MAX, GROUPING, STDEV, STDEVP, VAR y VARP.
- Un cálculo. La columna no se puede calcular a partir de una expresión que utilice otras columnas. Las columnas formadas mediante los operadores de conjunto UNION, UNION ALL, CROSSJOIN, EXCEPT e INTERSECT equivalen a un cálculo y tampoco son actualizables.
- Las columnas que se van a modificar no se ven afectadas por las cláusulas GROUP BY, HAVING o DISTINCT.
- No se utiliza TOP con la cláusula WITH CHECK OPTION en la vista.

Vistas

Create View (Indexadas)

El rendimiento de una vista puede ser significativamente peor que el de una tabla, especialmente cuando la vista lee de varias tablas.

Las vistas indexadas tienen aplicado un índice agrupado único. Las vistas indexadas también pueden usar índices no exclusivos y no agrupados, pero solo si existe un índice agrupado único.

Cuando se agrega el primer índice a una vista, se construye usando una estructura de árbol b similar a la de un índice de tabla. El árbol b incluye los datos de las columnas que están presentes en el índice.

Sin embargo, a diferencia de un índice de tabla, todas las columnas que se incluyen en la vista también se agregan a las filas de datos del índice. Esto puede generar grandes aumentos en los requisitos de almacenamiento de una base de datos, pero mejora el rendimiento de la vista.

- Debe crear un primer índice creado de tipo clúster único.
- Si existe un **index cluster** puede crear más índices no clúster.
- El índice clúster único mejora el rendimiento de la consulta
- **WITH SCHEMABINDING**. La razón por la que tiene esta opción activada es porque al crear índices en las vistas, en realidad se almacenan físicamente en la base de datos

System Object	Value
Options	
ANSI NULLs	True
Encrypted	False
Quoted identifier	True
Schema bound	True

Referencias

Create View

<https://docs.microsoft.com/es-es/sql/t-sql/statements/create-view-transact-sql?view=sql-server-ver15>

Modificar datos a travez de una vista

<https://docs.microsoft.com/es-es/sql/relational-databases/views/modify-data-through-a-view?view=sql-server-ver15>

Vistas Indexadas

<https://docs.microsoft.com/es-es/sql/relational-databases/views/create-indexed-views?view=sql-server-ver15>

Referencias

Create Function

<https://docs.microsoft.com/es-es/sql/t-sql/statements/create-function-transact-sql?view=sql-server-ver15>

Alter Function

<https://docs.microsoft.com/es-es/sql/t-sql/statements/alter-function-transact-sql?view=sql-server-ver15>

Drop Function

<https://docs.microsoft.com/es-es/sql/t-sql/statements/drop-function-transact-sql?view=sql-server-ver15>

Exec Function

<https://docs.microsoft.com/es-es/sql/relational-databases/user-defined-functions/execute-user-defined-functions?view=sql-server-ver15>



CAPACITACIÓN
PROFESIONAL