

Práctica A: Creación de funciones definidas por el usuario

Objetivos

Después de realizar esta práctica, el alumno será capaz de:

- Crear funciones escalares definidas por el usuario.
- Crear funciones definidas por el usuario con valores de tabla de varias instrucciones.
- Crear funciones definidas por el usuario con valores de tabla en línea.

Requisitos previos

- El archivo lab072.zip contiene los archivos de comandos necesarios para la realización de la práctica, así como las correspondientes soluciones.
- Para la realización de las prácticas es necesario ejecutar el script Restore.cmd del directorio Restore.

Ejercicio 1

Creación de una función escalar definida por el usuario

Los productos que vende su compañía están sujetos a tipos impositivos diferentes que se basan en las categorías de productos. Algunos productos como las bebidas tienen impuestos altos y otros productos como los condimentos no tienen impuestos. Decide crear una función definida por el usuario para encapsular la lógica de impuestos. Esta función definida por el usuario le ofrece un punto central para administrar los tipos impositivos y reducir el número de ubicaciones en las que debe repetir y mantener la lógica.

En este ejercicio, creará y ejecutará una secuencia de comandos que crea una función escalar definida por el usuario. Una vez creada, la probará para asegurarse de que funciona.

✍ Para crear una función escalar definida por el usuario

En este procedimiento, va a ejecutar un archivo de comandos para crear un desencadenador.

1. Abra el Analizador de consultas de SQL Server y, si así se le pide, conéctese con el servidor local con autenticación de Microsoft Windows®.
2. En la lista **BD**, haga clic en **ClassNorthwind**.
3. Abra, examine y ejecute TaxRate.sql.

Esta secuencia de comandos muestra una instrucción CASE que calcula un tipo impositivo para cada producto según la categoría del producto.

4. Cree una función escalar definida por el usuario denominada **fn_TaxRate** para encapsular la instrucción CASE. Acepte un parámetro como **@ProdID int** y un tipo de datos de retorno **numeric(5,4)**.

El archivo Soluciones\fn_TaxRate.sql es la secuencia de comandos completa que contiene esta función.

✍ Para probar la función

En este procedimiento, va a probar la función que acaba de crear; para ello, seleccionará columnas de la tabla **Products**.

- Ejecute la siguiente instrucción SELECT para seleccionar **ProductName**, **UnitPrice**, **CategoryID**, **TaxRate** y un valor calculado **PriceWithTax** de la tabla **Products**:

```
SELECT ProductName, UnitPrice, CategoryID,  
       ClassNorthwind.dbo.fn_TaxRate(ProductID) AS TaxRate,  
       UnitPrice * ClassNorthwind.dbo.fn_TaxRate(ProductID)  
       AS PriceWithTax  
FROM Products
```

La columna **TaxRate** debe contener valores de 1.00, 1.05 ó 1.10 para cada producto.

La columna **PriceWithTax** debe contener **UnitPrice** multiplicado por **TaxRate**.

Ejercicio 2

Creación de una función definida por el usuario con valores de tabla de varias instrucciones

En este ejercicio, creará una función definida por el usuario con valores de tabla de varias instrucciones que consulta la tabla **Employees** de la base de datos **ClassNorthwind** y muestra todos los subordinados directos e indirectos.

✍ Para crear una función definida por el usuario con valores de tabla de varias instrucciones

La tabla **Employees** de la base de datos **ClassNorthwind** contiene una columna denominada **ReportsTo** que contiene el número de Id. de empleado del jefe a cuyas órdenes está cada empleado.

En este ejercicio, creará una función definida por el usuario que toma el número de **EmployeeID** de un jefe como parámetro y se repite en toda la tabla **Employee**, con lo que se obtienen los empleados que están a las órdenes de ese jefe designado en todos los niveles.

- Abra, revise y ejecute `fn_FindReports.sql`.

Esta secuencia de comandos crea una función denominada **fn_FindReports** que devuelve una tabla de empleados subordinados.

✍ Para probar la función

En este procedimiento, va a probar la función que acaba de crear; para ello, seleccionará columnas de la función **fn_FindReports**.

1. Ejecute la siguiente instrucción **SELECT** para seleccionar las columnas **EmployeeID**, **Name**, **Title** y **MgrEmployeeID** de la función **fn_FindReports**.

```
SELECT EmployeeID, [Name], Title, MgrEmployeeID
FROM dbo.fn_FindReports(5)
```

Esta instrucción **SELECT** devuelve los empleados que están a las órdenes de Steven Buchanan (**EmployeeID** 5).

2. Ejecute la siguiente instrucción **SELECT** para seleccionar las columnas **EmployeeID**, **Name**, **Title** y **MgrEmployeeID** de la función **fn_FindReports**.

```
SELECT EmployeeID, Name, Title, MgrEmployeeID
FROM dbo.fn_FindReports(2)
```

Esta instrucción devuelve una tabla que contiene los nombres de los empleados que están a las órdenes de Andrew Fuller (**EmployeeID** 2).

El archivo Soluciones\Call_fn_FindReports.sql es la secuencia de comandos completa que contiene esta instrucción.

Ejercicio 3

Creación de una función definida por el usuario con valores de tabla en línea

En este ejercicio, creará una función definida por el usuario con valores de tabla en línea como alternativa a una vista. Esta función, denominada **fn_LargeFreight**, aceptará una cantidad en dólares como parámetro y devolverá pedidos con gastos de envío que tengan un importe superior a dicha cantidad en dólares.

✍ Para crear una función definida por el usuario con valores de tabla en línea

La tabla **Orders** de la base de datos **ClassNorthwind** contiene una columna denominada **Freight** que contiene la cantidad en dólares cobrada por los gastos de envío de cada pedido. Desea combinar esa tabla con la tabla **Shippers** para devolver la información de pedido y envío. Además, desea filtrar el conjunto de resultados para mostrar sólo aquellos pedidos que tienen gastos de envío cuantiosos.

En este procedimiento, creará una función definida por el usuario con valores de tabla en línea que sirva como vista con parámetros.

- Cree una función definida por el usuario con valores de tabla en línea denominada **fn_LargeFreight** que acepte un parámetro denominado **@FreightAmt** del tipo de datos **money** y devuelva el resultado de la siguiente instrucción SELECT:

```
SELECT S.ShipperID, S.CompanyName,
       O.OrderID, O.ShippedDate, O.Freight
FROM Shippers AS S JOIN Orders AS O
     ON S.ShipperID = O.ShipVia
WHERE O.Freight > @FreightAmt
```

El archivo Soluciones\fn_LargeFreight.sql es la secuencia de comandos completa que contiene esta función.

✍ Para probar la función

En este procedimiento, va a probar la función que acaba de crear; para ello, seleccionará columnas de la función **fn_LargeFreight**.

- Ejecute la siguiente instrucción SELECT para seleccionar filas de la función **fn_LargeFreight** cuyas ventas superen los 600 dólares.

```
SELECT * FROM fn_LargeFreight(600)
```

El archivo Soluciones\Call_fn_LargeFreight.sql es la secuencia de comandos completa que contiene esta instrucción.