Implementación de funciones definidas por el usuario

Contenido

Notas para el instructor

Este módulo se describe la implementación de las funciones definidas por el usuario. Explica los tres tipos de funciones definidas por el usuario y la sintaxis general para crearlas y modificarlas, y proporciona un ejemplo de cada tipo.

Después de completar este módulo, el alumno será capaz de:

- Describir los tres tipos de funciones definidas por el usuario.
- Crear y modificar funciones definidas por el usuario.
- Crear cada uno de los tres tipos de funciones definidas por el usuario.

Introducción

Objetivo del tema
Proporcionar una introducción a los temas y objetivos del módulo.

Explicación previa
En este módulo aprenderá a crear y utilizar funciones definidas por el usuario.

- ¿Qué es una función definida por el usuario?
- Definición de funciones definidas por el usuario
- Ejemplos de funciones definidas por el usuario

Este módulo proporciona una introducción a las funciones definidas por el usuario. Explica por qué y cómo utilizarlas, y la sintaxis para crearlas.

Después de completar este módulo, el alumno será capaz de:

- Describir los tres tipos de funciones definidas por el usuario.
- Crear y modificar funciones definidas por el usuario.
- Crear cada uno de los tres tipos de funciones definidas por el usuario.

¿Qué es una función definida por el usuario?

Objetivo del tema
Presentar el concepto de funciones definidas por el usuario y exponer las ventajas de su uso.

Explicación previa
Hay tres tipos de funciones definidas por el usuario.

- Funciones escalares
  - Similar a una función integrada
- Funciones con valores de tabla de varias instrucciones
  - Contenido como un procedimiento almacenado
  - Se hace referencia como una vista
- Funciones con valores de tabla en línea
  - Similar a una vista con parámetros
  - Devuelve una tabla como el resultado de una instrucción SELECT única

Con Microsoft® SQL Server™ 2000, puede diseñar sus propias funciones para complementar y ampliar las funciones (integradas) suministradas por el sistema.

Una función definida por el usuario toma cero o más parámetros de entrada y devuelve un valor *escalar* o una tabla. Los parámetros de entrada pueden ser de cualquier tipo de datos, salvo **timestamp**, **cursor** o **table**. Las funciones definidas por el usuario no admiten parámetros de salida.

Sugerencia
Esta página ofrece una introducción a los tres tipos de funciones definidas por el usuario. Indique a los alumnos que en los temas siguientes se tratarán las diferencias entre los tipos de funciones.

SQL Server 2000 admite tres tipos de funciones definidas por el usuario:

Funciones escalares

Una función escalar es similar a una función integrada.

Funciones con valores de tabla de varias instrucciones

Una función con valores de tabla de varias instrucciones devuelve una tabla creada por una o varias instrucciones Transact-SQL y es similar a un procedimiento almacenado. A diferencia de los procedimientos almacenados, se puede hacer referencia a una función con valores de tabla de varias instrucciones en la cláusula FROM de una instrucción SELECT como si se tratara de una vista.

Funciones con valores de tabla en línea

Una función con valores de tabla en línea devuelve una tabla que es el resultado de una sola instrucción SELECT. Es similar a una vista, pero ofrece una mayor flexibilidad que las vistas en el uso de parámetros y amplía las características de las vistas indizadas.

Definición de funciones definidas por el usuario

Objetivo del tema
Presentar los temas de esta sección.

Explicación previa
Esta sección trata la creación, modificación y eliminación de una función definida por el usuario. También trata los permisos.

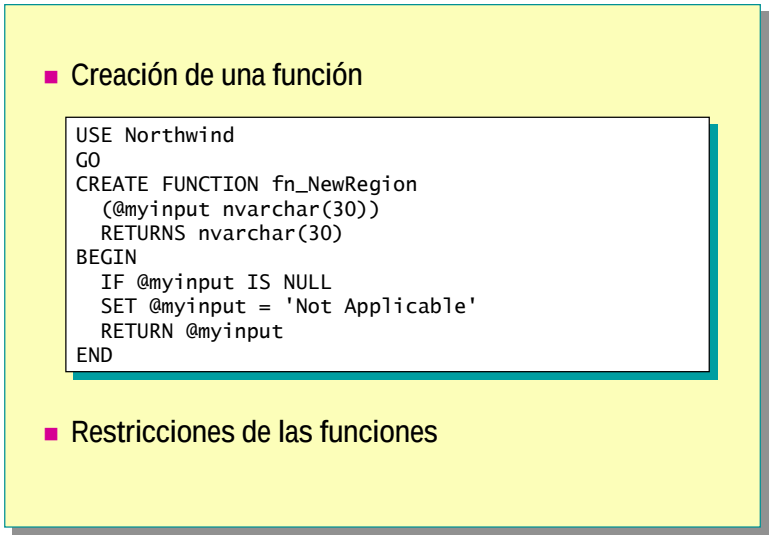
- Creación de una función definida por el usuario
- Creación de una función con enlace a esquema
- Establecimiento de permisos para funciones definidas por el usuario
- Modificación y eliminación de funciones definidas por el usuario

Esta sección trata la creación, modificación y eliminación de una función definida por el usuario. También trata los permisos.

Creación de una función definida por el usuario

Objetivo del tema
Describir la instrucción general CREATE FUNCTION.

Explicación previa
Una función definida por el usuario se crea de forma muy similar a una vista o un procedimiento almacenado.



Sugerencia
Pregunte a los alumnos qué tipo de función definida por el usuario se usa en el ejemplo.

El ejemplo es una función escalar definida por el usuario, trate de hacer una explicación genérica y aplicable a todas las funciones definidas por el usuario.

Una función definida por el usuario se crea de forma muy similar a una vista o un procedimiento almacenado.

Creación de una función

Las funciones definidas por el usuario se crean mediante la instrucción CREATE FUNCTION. Cada nombre descriptivo de una función definida por el usuario (nombreBaseDeDatos.nombrePropietario.nombreFunción) debe ser único. La instrucción especifica los parámetros de entrada con sus tipos de datos, las instrucciones de procesamiento y el valor devuelto con cada tipo de dato.

Sintaxis

```
CREATE FUNCTION [ nombrePropietario. ] nombreFunción
  ( [ { @nombreParámetro tipoDatosParámetroEscalar [ = predeterminado ] } [
  ,...n ] ] )
  RETURNS tipoDatosDevoluciónEscalar
  [ WITH < opciónFunción > [,...n] ]
  [ AS ]
  BEGIN
  cuerpoFunción
  RETURN expresiónEscalar
  END
```

Ejemplo

En este ejemplo se crea una función definida por el usuario para reemplazar un valor NULL por las palabras “Not Applicable”.

```
USE Northwind
GO

CREATE FUNCTION fn_NewRegion
  (@myinput nvarchar(30))
  RETURNS nvarchar(30)
BEGIN
  IF @myinput IS NULL
  SET @myinput = 'Not Applicable'
  RETURN @myinput
END
```

Al hacer referencia a una función escalar definida por el usuario, especifique el propietario y el nombre de la función en una sintaxis de dos partes.

```
SELECT LastName, City, dbo.fn_NewRegion(Region) AS Region,
   Country
FROM dbo.Employees
```

Resultado

Lastname	City	Region	Country
Davolio	Seattle	WA	USA
Fuller	Tacoma	WA	USA
Leverling	Kirkland	WA	USA
Peacock	Redmond	WA	USA
Buchanan	London	Not Applicable	UK
Suyama	London	Not Applicable	UK
King	London	Not Applicable	UK
Callahan	Seattle	WA	USA
Dodsworth	London	Not Applicable	UK

Restricciones de las funciones

Las funciones *no deterministas* son funciones como GETDATE() que pueden devolver diferentes valores cada vez que se invocan con el mismo conjunto de valores de entrada. No se pueden utilizar funciones no deterministas integradas en el texto de funciones definidas por el usuario. Las siguientes funciones integradas son no deterministas.

@@ERROR	FORMATMESSAGE	IDENTITY	USER_NAME
@@IDENTITY	GETANSINULL	NEWID	@@ERROR
@@ROWCOUNT	GETDATE	PERMISSIONS	@@IDENTITY
@@TRANCOUNT	GetUTCDate	SESSION_USER	@@ROWCOUNT
APP_NAME	HOST_ID	STATS_DATE	@@TRANCOUNT
CURRENT_TIMESTAMP	HOST_NAME	SYSTEM_USER	
CURRENT_USER	IDENT_INCR	TEXTPTR	
DATENAME	IDENT_SEED	TEXTVALID	

Creación de una función con enlace a esquema

Objetivo del tema
Describir el propósito y las restricciones del enlace a esquema.

Explicación previa
El enlace a esquema se puede utilizar para enlazar la función con los objetos de base de datos a los que hace referencia.

- Todas las funciones definidas por el usuario y las vistas a las que la función hace referencia también están enlazadas a esquema
- No se utiliza un nombre de dos partes para los objetos a los que hace referencia
- La función y los objetos se encuentran todos en la misma base de datos
- Tiene permiso de referencia en los objetos requeridos

El enlace a esquema se puede utilizar para enlazar la función con los objetos de base de datos a los que hace referencia. Si se crea una función con la opción SCHEMABINDING, los objetos de base de datos a los que la función hace referencia no se pueden modificar (mediante la instrucción ALTER) o quitar (mediante la instrucción DROP).

Una función se puede enlazar a esquema sólo si se cumplen las siguientes condiciones:

- Todas las funciones definidas por el usuario y las vistas a las que la función hace referencia también están enlazadas a esquema.
- No se utiliza un nombre de dos partes en el formato **propietario.nombreObjeto** para los objetos a los que la función hace referencia.
- La función y los objetos a los que hace referencia pertenecen a la misma base de datos.
- El usuario que ejecutó la instrucción CREATE FUNCTION tiene el permiso REFERENCE sobre todos los objetos de la base de datos a los que la función hace referencia.

Establecimiento de permisos para funciones definidas por el usuario

Objetivo del tema
Describir la importancia de establecer permisos para utilizar las funciones definidas por el usuario.

Explicación previa
El modelo de seguridad de las funciones definidas por el usuario es similar al de las vistas.

- Necesita permiso para CREATE FUNCTION
- Necesita permiso para EXECUTE
- Necesita permiso para REFERENCE en las tablas, vistas o funciones citadas
- Debe ser propietario de la función para utilizar la instrucción CREATE o ALTER TABLE

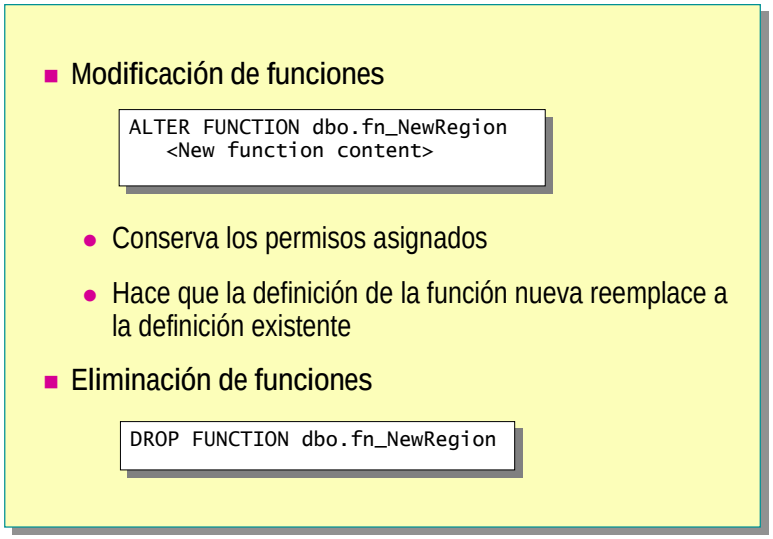
Los requisitos en cuanto a permisos para las funciones definidas por el usuario son similares a los de otros objetos de base de datos.

- Debe tener el permiso CREATE FUNCTION para crear, modificar o quitar funciones definidas por el usuario.
- Para que los usuarios distintos del propietario puedan utilizar una función en una instrucción Transact-SQL, se les debe conceder el permiso EXECUTE sobre la función.
- Si la función está enlazada a esquema, debe tener el permiso REFERENCE sobre las tablas, vistas y funciones a las que la función hace referencia. Los permisos REFERENCE se pueden conceder mediante la instrucción GRANT para las vistas y funciones definidas por el usuario, así como las tablas.
- Si una instrucción CREATE TABLE o ALTER TABLE hace referencia a una función definida por el usuario en una restricción CHECK, cláusula DEFAULT o columna calculada, el propietario de la tabla debe ser también el propietario de la función.

Modificación y eliminación de funciones definidas por el usuario

Objetivo del tema
Describir las instrucciones ALTER FUNCTION y DROP FUNCTION.

Explicación previa
Las funciones definidas por el usuario se modifican mediante la instrucción ALTER FUNCTION.



Las funciones definidas por el usuario se pueden modificar mediante la instrucción ALTER FUNCTION.

La ventaja de modificar una función en lugar de eliminarla y volver a crearla es la misma que para las vistas y los procedimientos. Los permisos sobre la función se mantienen y se aplican inmediatamente a la función revisada.

Modificación de funciones

Las funciones definidas por el usuario se modifican mediante la instrucción ALTER FUNCTION.

Ejemplo

Este ejemplo muestra cómo se modifica una función.

```
ALTER FUNCTION dbo.fn_NewRegion
   <New function content >
```

Eliminación de funciones

Las funciones definidas por el usuario se eliminan mediante la instrucción DROP FUNCTION.

Ejemplo

Este ejemplo muestra cómo se elimina una función.

```
DROP FUNCTION dbo.fn_NewRegion
```

◆ Ejemplos de funciones definidas por el usuario

Objetivo del tema
Presentar los temas de esta sección.

Explicación previa
En esta sección se describen los tres tipos de funciones definidas por el usuario.

- Uso de una función escalar definida por el usuario
- Ejemplo de una función escalar definida por el usuario
- Uso de una función con valores de tabla de varias instrucciones
- Ejemplo de una función con valores de tabla de varias instrucciones
- Uso de una función con valores de tabla en línea
- Ejemplo de una función con valores de tabla en línea

En esta sección se describen los tres tipos de funciones definidas por el usuario. Se describe su propósito y se ofrecen ejemplos de la sintaxis que se puede utilizar para crearlas e invocarlas.

Uso de una función escalar definida por el usuario

Objetivo del tema
Describir cómo funciona una función escalar.

Explicación previa
Una función escalar definida por el usuario es similar a una función integrada.

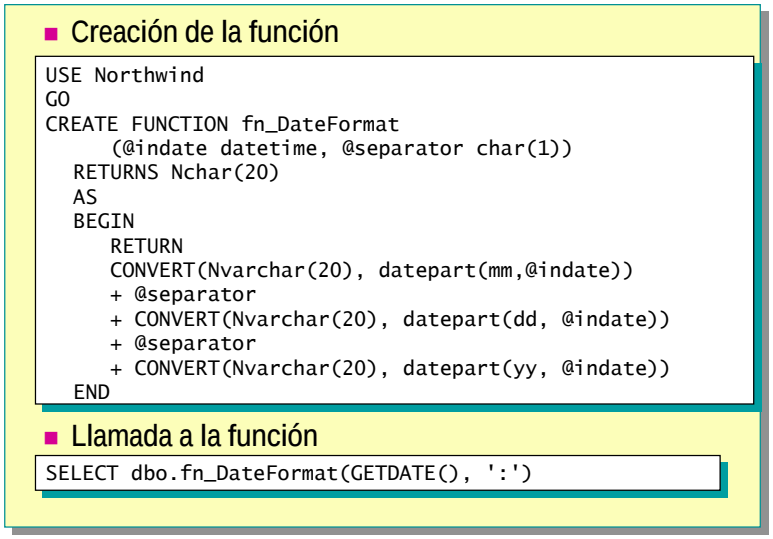
- La cláusula RETURNS especifica el tipo de datos
- La función se define en un bloque BEGIN y END
- El tipo de devolución puede ser cualquier tipo de datos, excepto text, ntext, image, cursor o timestamp

Una función escalar devuelve un solo valor de datos del tipo definido en una cláusula RETURNS. El cuerpo de la función, definido en un bloque BEGIN…END, contiene el conjunto de instrucciones Transact-SQL que devuelven el valor. El tipo de devolución puede ser cualquier tipo de datos, excepto **text**, **ntext**, **image**, **cursor** o **timestamp**.

Ejemplo de una función escalar definida por el usuario

Objetivo del tema
Proporcionar un ejemplo de una función escalar definida por el usuario para su discusión en clase.

Explicación previa
A continuación se ofrece un ejemplo de una función escalar definida por el usuario.



Una función escalar definida por el usuario es similar a una función integrada. Después de crearla, se puede volver a utilizar.

Ejemplo

Este ejemplo crea una función definida por el usuario que recibe separadores de fecha y columna como variables y da formato a la fecha como una cadena de caracteres.

```
USE Northwind
GO
CREATE FUNCTION fn_DateFormat
   (@indate datetime, @separator char(1))
RETURNS Nchar(20)
AS
BEGIN
   RETURN
   CONVERT(Nvarchar(20), datepart(mm,@indate))
   + @separator
   + CONVERT(Nvarchar(20), datepart(dd, @indate))
   + @separator
   + CONVERT(Nvarchar(20), datepart(yy, @indate))
END
```

Una función escalar definida por el usuario se puede invocar de la misma forma que una función integrada.

```
SELECT dbo.fn_DateFormat(GETDATE(), ':')
```

Sugerencia
Este ejemplo muestra cómo se puede utilizar una función no determinista como GETDATE() al llamar a una función definida por el usuario, incluso aunque no se pueda utilizar en una función definida por el usuario.

Uso de una función con valores de tabla de varias instrucciones

Objetivo del tema
Describir cómo funciona una función con valores de tabla de varias instrucciones.

Explicación previa
Una función con valores de tabla de varias instrucciones es una combinación de una vista y un procedimiento almacenado.

- BEGIN y END contienen múltiples instrucciones
- La cláusula RETURNS especifica el tipo de datos de la tabla
- La cláusula RETURNS da nombre y define la tabla

Una función con valores de tabla de varias instrucciones es una combinación de una vista y un procedimiento almacenado. Se pueden utilizar funciones definidas por el usuario que devuelvan una tabla para reemplazar procedimientos almacenados o vistas.

Una función con valores de tabla (al igual que un procedimiento almacenado) puede utilizar lógica compleja y múltiples instrucciones Transact-SQL para crear una tabla. De la misma forma que se utiliza una vista, se puede utilizar una función con valores de tabla en la cláusula FROM de una instrucción Transact-SQL.

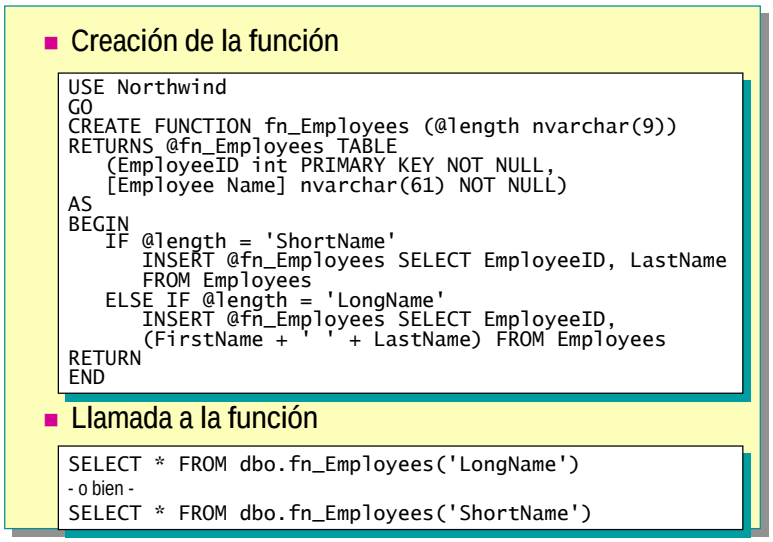
Cuando utilice una función con valores de tabla de varias instrucciones, tenga en cuenta los hechos siguientes:

- BEGIN y END delimitan el cuerpo de la función.
- La cláusula RETURNS especifica **table** como el tipo de datos devuelto.
- La cláusula RETURNS define un nombre para la tabla y su formato. El ámbito del nombre de la variable de retorno es local a la función.

Ejemplo de una función con valores de tabla de varias instrucciones

Objetivo del tema
Proporcionar un ejemplo de una función con valores de tabla de varias instrucciones para su explicación en clase.

Explicación previa
A continuación se ofrece un ejemplo de una función con valores de tabla de varias instrucciones.



Puede crear funciones mediante muchas instrucciones que realizan operaciones complejas.

Ejemplo

Este ejemplo crea una función con valores de tabla de varias instrucciones que devuelve el apellido o el nombre y los apellidos de un empleado, dependiendo del parámetro que se proporcione.

```
USE Northwind
GO

CREATE FUNCTION fn_Employees
   (@length nvarchar(9))
RETURNS @fn_Employees TABLE
   (EmployeeID int PRIMARY KEY NOT NULL,
   [Employee Name] Nvarchar(61) NOT NULL)
AS
BEGIN
   IF @length = 'ShortName'
      INSERT @fn_Employees SELECT EmployeeID, LastName
      FROM Employees
   ELSE IF @length = 'LongName'
      INSERT @fn_Employees SELECT EmployeeID,
      (FirstName + ' ' + LastName) FROM Employees
RETURN
END
```

Puede llamar a la función en lugar de una tabla o vista.

```
SELECT * FROM dbo.fn_Employees('LongName')
```

- o bien -

```
SELECT * FROM dbo.fn_Employees('ShortName')
```

Uso de una función con valores de tabla en línea

Objetivo del tema
Describir cómo funciona una función con valores de tabla en línea.

Explicación previa
Una función con valores de tabla en línea sólo puede contener una instrucción SELECT.

- El contenido de la función es una instrucción SELECT
- No utilice BEGIN y END
- RETURN especifica table como el tipo de datos
- El formato se define por el conjunto de resultados

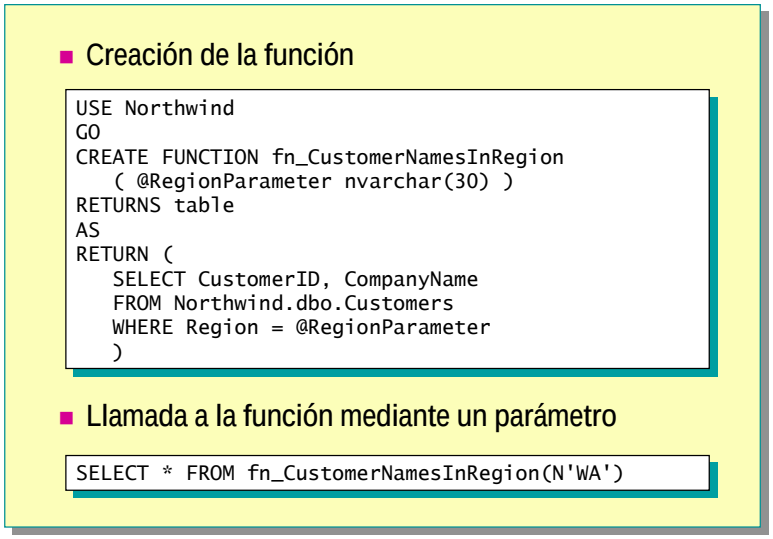
Las funciones en línea definidas por el usuario devuelven una tabla y se hace referencia a ellas en la cláusula FROM, al igual que una vista. Cuando utilice una función en línea definida por el usuario, tenga en cuenta los hechos y directrices siguientes:

- La cláusula RETURN contiene una única instrucción SELECT entre paréntesis. El conjunto de resultados de la instrucción SELECT constituye la tabla que devuelve la función. La instrucción SELECT que se utiliza en una función en línea está sujeta a las mismas restricciones que las instrucciones SELECT que se utilizan en las vistas.
- BEGIN y END no delimitan el cuerpo de la función.
- RETURN especifica **table** como el tipo de datos devuelto.
- No necesita definir el formato de una variable de retorno, ya que lo establece el formato del conjunto de resultados de la instrucción SELECT en la cláusula RETURN.

Ejemplo de una función con valores de tabla en línea

Objetivo del tema
Proporcionar un ejemplo de una función con valores de tabla en línea para su explicación en clase.

Explicación previa
A continuación se ofrece un ejemplo de una función con valores de tabla en línea.



Sugerencia
Resalte que no se puede crear una vista como:
CREATE VIEW CustView AS
SELECT <campos> FROM Customers WHERE Region = @RegionParameter

Las funciones en línea se pueden utilizar para obtener la funcionalidad de las vistas con parámetros.

Al crear una vista no se puede incluir en ella un parámetro proporcionado por el usuario. Esto se suele resolver proporcionando una cláusula WHERE al llamar a la vista. Sin embargo, esto puede requerir la creación de una cadena para ejecución dinámica, lo cual puede aumentar la complejidad de la aplicación. La funcionalidad de una vista con parámetros se puede obtener mediante una función con valores de tabla en línea.

Ejemplo

Este ejemplo crea una función con valores de tabla en línea que toma un valor de región como parámetro.

```
USE Northwind
GO

CREATE FUNCTION fn_CustomerNamesInRegion
   ( @RegionParameter nvarchar(30) )
RETURNS table
AS
RETURN (
   SELECT CustomerID, CompanyName
   FROM Northwind.dbo.Customers
   WHERE Region = @RegionParameter
   )
```

Para llamar a la función, proporcione el nombre de la función como la cláusula FROM y proporcione un valor de región como parámetro.

```
SELECT * FROM fn_CustomerNamesInRegion(N'WA')
```

Sugerencia Las funciones en línea pueden aumentar notablemente el rendimiento cuando se utilizan con vistas indizadas. SQL Server realiza operaciones complejas de agregación y combinación cuando se crea el índice. Las consultas posteriores pueden utilizar una función en línea con un parámetro para filtrar filas del conjunto de resultados simplificado almacenado.