FUNCIONES DEFINIDAS POR EL USUARIO

INTRODUCCIÓN

El uso de los procedimientos almacenados no es muy flexible, porque solo pueden ser usados con las instrucciones EXECUTE o INSERT … EXECUTE.
Si un procedimiento almacenado devuelve un solo valor, no puede ser usado dentro de una expresión.
Si un procedimiento almacenado devuelve un conjunto de resultados, no podemos usarlo en la cláusula FROM de una instrucción TRANSACT.
Algunas funciones definidas por el usuario se parecen a las vistas, pero su definición puede incluir más de una instrucción y tener parámetros.
Las funciones definidas por el usuario se pueden llamar igual que los procedimientos almacenados, y una función escalar definida por el usuario se puede usar en cualquier lugar de una instrucción Transact SQL en el que sea válido el uso de una expresión.
Una función definida por el usuario que devuelva una tabla se puede usar en la cláusula FROM de cualquier instrucción del DML.

Las funciones definidas por el usuario comparten muchos de los beneficios de los procedimientos almacenados, pero además, tienen más aplicaciones útiles:

- Usar el conjunto de resultados devuelto por un procedimiento almacenado en la cláusula FROM de una consulta.
- Combinar los resultados de dos procedimientos almacenados sin necesidad de usar tablas temporales para almacenar los resultados intermedios.
- Usar el resultado de un procedimiento almacenado como argumento del operador IN.
- Usar un procedimiento almacenado como subconsulta en la cláusula WHERE.
- Crear una vista que no podría ser resuelta con una única instrucción SELECT.
- Crear una vista con parámetros.
- Ampliar la lista de las funciones predefinidas con funciones financieras de todo tipo.
- Crear nuevas funciones matemáticas para cualquier app científica especial necesaria.

TIPOS DE FUNCIONES DEFINIDAS POR EL USUARIO

Dependiendo del valor de retorno, las funciones definidas por el usuario se pueden dividir en tres grupos:

- Funciones escalares: devuelven un único valor escalar.
- Funciones tabulares: devuelven un conjunto de resultados completo (como una tabla).
- Funciones en línea: son un caso especial de las tabulares pero constan de una sola instrucción Select.

FUNCIONES ESCALARES
Se pueden usar en cualquier lugar donde se admita una expresión. Ejemplo:

- En la cláusula SELECT de una instrucción SELECT, como parte de una expresión o como una columna individual.
- En la cláusula SET de una instrucción UPDATE, como valor para insertar en la columna de la tabla que estamos modificando.

- En la cláusula FROM de cualquier instrucción DML (SELECT, UPDATE, INSERT, DELETE) como una tabla derivada de una sola fila y una sola columna.
- En la cláusula FROM de cualquier instrucción DML, como parte de la condición de combinación en la cláusula ON.
- En las cláusulas WHERE o HAVING de cualquier instrucción DML.
- En las cláusulas GROUP BY, como parte de una condición de agrupamiento.
- En la cláusula ORDER BY de cualquier instrucción, como criterio de ordenación.
- Como valor DEFAULT para una columna.
- Dentro de la definición de una restricción CHECK.
- Dentro de una expresión CASE.
- Dentro de una instrucción PRINT, si la función definida por el usuario devuelve una cadena.
- Como parte de la condición en instrucciones IF o WHILE.
- Como parte de la definición de una columna calculada.
- Como parámetro, al llamar a un procedimiento almacenado u otra función definida por el usuario.
- Como valor de retorno de un procedimiento almacenado, si la función definida por el usuario devuelve un valor entero.
- Como valor de retorno de otra función escalar definida por el usuario.

Las funciones escalares definidas por el usuario se pueden combinar con otras funciones en una expresión, siempre que los tipos de datos sean compatibles con la operación.

Las funciones escalares definidas por el usuario se identifican porque el tipo de datos de su valor de retorno es escalar y porque su definición va encerrada dentro de un bloque BEGIN…END.

Dentro de una función definida por el usuario no es posible modificar tablas directamente. Tampoco se pueden modificar las tablas temporales.

FUNCIONES EN LÍNEA

Una vista se puede usar en cualquier instrucción DML.
Dentro de la vista podemos usar una cláusula WHERE para limitar a filas específicas los resultados de la vista, pero esta restricción es inflexible, ya que forma parte de la definición de la vista.
En la consulta en la que se usa la vista podemos usar otra cláusula WHERE que limite la búsqueda. En este caso, una vez que la definición de la vista se haya combinado con la de la consulta, SQL Server combinará ambas cláusulas WHERE en el plan de consulta final.

Para crear algo parecido con parámetros, SQL Server posee las funciones definidas por el usuario.

Una función en línea definida por el usuario contiene una sola instrucción SELECT pero, a diferencia de las vistas, puede usar varios parámetros para restringir la consulta y da una interfaz de llamada más sencilla que las vistas.

Las funciones en línea definidas por el usuario se pueden identificar porque devuelven una tabla y su definición consta de una sola instrucción SELECT, sin un bloque BEGIN … END.

- Una función en línea se puede usar en cualquier lugar en el que se admitiría una tabla:
- En la cláusula SELECT de una instrucción SELECT, como parte de una subconsulta que devuelva un único valor.
- En la cláusula SET de una instrucción UPDATE, como parte de una subconsulta que devuelva un único valor para una columna de una tabla que queremos modificar.
- En la cláusula FROM de cualquier instrucción DML.
- En la cláusula WHERE o HAVING de cualquier instrucción DML, como parte de una subconsulta que devuelva un único valor para compararlo con una columna o variable.
- En la cláusula WHERE de cualquier instrucción DML, como parte de una subconsulta que comience con EXISTS o con NOT EXISTS.
- En las cláusulas WHERE o HAVING de cualquier instrucción DML, como parte de una subconsulta con los operadores IN o NOT IN, siempre que la subconsulta devuelva una sola columna.

FUNCIONES TABULARES

Las funciones tabulares de varis instrucciones (o funciones tabulares) se parecen a las funciones en línea. De hecho, podemos usarlas en las mismas situaciones, pero no estamos restringidos a definirlas con una única instrucción SELECT.

Una función tabular devuelve un único conjunto de resultados: el contenido de una variable de tabla con un formato predefinido indicado en la declaración de la función.

El resultado de una función tabular es siempre de solo lectura, mientras que el resultado de una función en línea puede serlo o no, según la instrucción SELECT usada para definir la función.

Las funciones tabulares se parecen a procedimientos almacenados, pero devuelven un único conjunto de resultados

En una función tabular se pueden definir parámetros, al igual que en las funciones escalares definidas por el usuario y en los procedimientos almacenados. Sin embargo, en una función definida por el usuario no se puede no se puede crear un parámetro de salida.

Comentarios a la sintaxis de CREATE FUNCTION:

RETURNS @nombretabla TABLE

Definición de cada una de las columnas de la variable tabla @nombretabla, siguiendo las mismas reglas que en las tablas normales.

La instrucción RETURN sola, que envía la variable de la tabla como conjunto de resultados al proceso que efectuó la llamada.

El contenido de una función tabular es similar al obtenido con procedimiento almacenado que use una tabla temporal para almacenar resultados intermedios y como paso final ejecute una consulta de selección sobre esa tabla.

Dentro de funciones escalares o tabulares no se pueden crear tablas; en cambio, se pueden usar variables de tabla. Sin embargo, es buena práctica reemplazar las tablas temporales por variables de procedimientos almacenados y en las secuencias de comandos, siempre que sea posible.

Las funciones tabulares se parecen a las de en línea excepto que se deben declarar de forma explícita las columna de la tabla devuelta. Sin embargo, las funciones tabulares ofrecen mucha más flexibilidad que las funciones en línea, así como los procedimientos almacenados permiten más flexibilidad que las vistas.

A las funciones tabulares se les llama de la misma forma que a las funciones en línea.
Sin embargo, SQL Server ejecuta estas funciones tabulares de una forma absolutamente distinta a las funciones en línea:

- SQL Server combina la definición de las funciones en línea con la definición de la consulta exterior y produce un único plan de ejecución, en el que no se pueden hallar rastros de la función definida por el usuario.
- La primera vez que se ejecuta una función tabular SQL Server la compila y guarda en memoria un plan de ejecución. Cada vez que llamemos a esta función, la consulta que realiza la llamada requerirá la ejecución del plan de ejecución guardado y la función devolverá la variable de tabla, que se usará en la columna exterior.

ELIMINACIÓN DE FUNCIONES DEFINIDAS POR EL USUARIO

La instrucción es DROP FUNCTION.
Antes de eliminar una función definida por el usuario (como con otros objetos) conviene verificar sus relaciones con otros objetos.
Si una función definida por el usuario forma parte de la definición de una restricción no es posible eliminarla. Si se elimina una función definida por el usuario y hay otras funciones, vistas, triggers o procedimientos almacenados que usan aquella función, la próxima vez que se ejecute algunos de estos objetos se producirá un error.

IMPEDIR LA MODIFICACIÓN DE LOS OBJETOS RELACIONADOS: LA OPCIÓN SCHEMABINDING.
Mediante esta opción se puede evitar la modificación de los objetos relacionados con una función definida por el usuario. Si activamos esta opción no podremos usar ninguna instrucción ALTER para modificar la definición de los objetos relacionados, ni usar ninguna instrucción DROP para eliminarlos. Este vínculo desaparece si se elimina la función o si se modifica su definición sin usar la opción SCHEMANBINDING.
Para usar esta opción se deben cumplir las siguientes condiciones:

- Todas las funciones y vistas a las que haga referencia la función también deben estar definidas con SCHEMABINDING.
- Toda referencia a objetos dentro de la función debe usar nombres de dos partes (propietario.objeto).

- Todo objeto al que haga referencia la función debe pertenecer a la misma base de datos que la función.
- El usuario que crea la función (no necesariamente su propietario) debe tener permiso REFERENCES sobre todos los objetos a los que se haga referencia dentro de la función. Se recomienda que sólo ejecuten la instrucción CREATE FUNCTION los miembros de la función db_owner.

FUNCIONES DETERMINISTAS Y NO DETERMINISTAS

Algunas funciones devuelven siempre el mismo valor cuando son llamadas con el mismo conjunto de argumentos. Estas funciones reciben el nombre de deterministas. Esto es importante si queremos crear un índice agrupado o un índice de cualquier tipo sobre una columna calculada, ya que estos índices solo se pueden crear a partir de funciones deterministas.

Una buena parte de las funciones predefinidas son deterministas:

ABS	DATEDIFUNCIÓN	PARSENAME
ACOS	DAY	POWER
ASIN	DEGREES	RADIANS
ATAN	EXP	ROUND
ATN2	FLOBJETOR	SIGN
CEILING	ISNULL	SIN
COALESCE	ISNUMERIC	SQUARE
COS	LOG	SQRT
COT	LOG10	TAN
DATALENGTH	MONTH	YEAR
DATEADD	NULLIF	

Algunas funciones son deterministas o no deterministas, según el modo en que se usen:

- CAST es determinista para todos los tipos de valores, excepto al convertir desde datetime, smalldatetime y de un sql_variant que contenga un valor de fecha, porque el resultado final depende de configuraciones regionales.
- RAND es determinista cuando se indica el valor de la semilla; de lo contrario es no determinista.
- CONVERT es determinista o no determinista en los mismos casos que CAST, excepto cuando se convierten datos datetime o smalldatetime y se indica un estilo, en cuyo caso, el resultado es siempre predecible, y la función, determinista.
- ISDATE es determinista a menos que la usemos con CONVERT y con un estilo predecible.

Una función definida por el usuario es determinista solamente si:

- Todas las funciones (predefinidas o definidas por el usuario) a las que hace referencia la función son deterministas.
- La función está definida con la opción SCHEMABINDING.
- La función no hace referencia a objetos, tales como tablas, vistas o procedimientos almacenados extendidos, que no estén definidos dentro de la propia función.

Dentro de una función definida por el usuario no son válidas las funciones predefinidas que utilicen la hora actual:

CURRENT_TIMESTAMP	GETDATE
GETUTCDATE	IDENTITY
NEWID	TEXTPTR
@@DBTS	@@MAX_CONNECTIONS

Tampoco son válidas las funciones estadísticas del sistema:

@@CONNECTIONS	@@PACK_RECEIVED
@@CPU_BUSY	@@PACK_SENT
Fn_virtualfilestat	@@TIMETICKS
@@IDLE	@@TOTAL_ERRORS
@@IO_BUSY	@@TOTAL_READ
@@PACKET_ERRORS	@@TOTAL_WRITE

MODIFICACIÓN DE LA DEFINICIÓN DE FUNCIONES DEFINIDAS POR EL USUARIO

Se modifica con ALTER FUNCTION. Las características son similares a las de los programas (encriptación, permisos, etc).
Si no se usa la opción SCHEMABINDING al ejecutar esta orden, se perderá el vínculo entre la función y los objetos de los que depende.

USO DE FUNCIONES DEFINIDAS POR EL USUARIO Y LA SEGURIDAD.

La concesión o denegación de permisos sobre las funciones definidas por el usuario dependen del tipo de función:

- Para las funciones escalares: se pueden conceder o denegar permisos de tipo EXECUTE o REFERENCES.
- Para las funciones en línea: se pueden conceder o denegar permisos para SELECT, INSERT, UPDATE, DELETE o REFERENCES.
- Para las funciones tabulares de varias instrucciones se pueden conceder o denegar permisos para SELECT y REFERENCES.

Al igual que en los procedimientos almacenados y en las vistas, si todos los objetos a los que se hace referencia pertenecen al mismo propietario que la función y un usuario trata de usarla, SQL Server solo controlará los permisos para la función.

APLICACIÓN DE FUNCIONES DEFINIDAS POR EL USUARIO.

Las fórmulas de uso frecuente se suelen convertir en funciones escalares. En este caso queda guardado en memoria el plan de ejecución de la función.
Desde una función definida por el usuario es posible llamar a otras, pero solamente hasta 32 niveles de anidamiento.

Se debe tener siempre en cuenta que modificar los objetos subyacentes puede afectar al resultado de una función definida por el usuario, a menos que al crear la función definida por el usuario se utilice la opción SCHEMABINDING.

Una aplicación de una función definida por el usuario es convertir un procedimiento almacenado que solo entrega un parámetro de salida (OUTPUT) en una función escalar. Esta es una forma más natural de utilizarla.

Otra aplicación es la conversión de un procedimiento almacenado que devuelve un único conjunto de resultados en una función tabular.

Otra aplicación es convertir vistas en funciones definidas por el usuario. En este caso el único beneficio es usar parámetros. Sin embargo, cuando sólo usamos una vista para leer datos, nos será provechoso convertirla en una función tabular, ya que en su primera ejecución SQL Server la optimizará y la compilará, con lo que ganaremos en rendimiento respecto de una vista.

También se pueden usar funciones definidas por el usuario en las restricciones. Se puede usar una función escalar en cualquier lugar en el que pueda ubicarse una expresión:

- Restricciones DEFAULT
- Restricciones CHECK
- Objetos DEFAULT
- Objetos RULE
- Una restricción PRIMARY KEY definida en una columna calculada a partir de una función definida por el usuario, siempre que los valores devueltos sean únicos.
- Una restricción UNIQUE definida en una columna calculada a partir de una función definida por el usuario, siempre que los valores devueltos sean únicos.

Por tanto, es posible acceder desde el interior de una restricción a valores pertenecientes a otras tablas, siempre que la restricción use una función definida por el usuario que produzca su resultado a partir de los datos externos.
El único lugar en el que podría ser usada una función tabular o una función en línea es como una subconsulta de una restricción CHECK, pero las CHECK no admiten subconsultas.