PROCEDIMIENTOS ALMACENADOS

Al crear una aplicación con Microsoft SQL Server, el lenguaje de programación Transact-SQL es la principal interfaz de programación entre las aplicaciones y la base de datos SQL Server. Cuando utilice programas Transact-SQL, dispone de dos métodos para almacenar y ejecutar los programas. Puede almacenar localmente los programas y crear aplicaciones que envíen los comandos a SQL Server y procesen los resultados, o bien almacenar los programas como procedimientos almacenados en SQL Server y crear aplicaciones que ejecuten los procedimientos almacenados y procesen los resultados.

Los procedimientos almacenados de SQL Server son similares a los procedimientos de otros lenguajes de programación en el sentido de que pueden:

- Aceptar parámetros de entrada y devolver varios valores en forma de parámetros de salida al lote o al procedimiento que realiza la llamada.

- Contener instrucciones de programación que realicen operaciones en la base de datos, incluidas las llamadas a otros procedimientos.

- Devolver un valor de estado a un lote o a un procedimiento que realiza una llamada para indicar si la operación se ha realizado correctamente o ha habido un error (y el motivo del mismo).

Puede utilizar la instrucción EXECUTE de Transact-SQL para ejecutar un procedimiento almacenado. Los procedimientos almacenados difieren de las funciones en que no devuelven valores en lugar de sus nombres ni pueden utilizarse directamente en una expresión.

Utilizar procedimientos almacenados en SQL Server en vez de programas Transact-SQL almacenados localmente en equipos clientes presenta las siguientes ventajas:

- Permiten una programación modular.

Puede crear el procedimiento una vez, almacenarlo en la base de datos y llamarlo desde el programa tantas veces como desee. Un especialista en programación de bases de datos puede crear procedimientos almacenados, que luego será posible modificar independientemente del código fuente del programa.

- Permiten una ejecución más rápida.

En los casos en que la operación requiere una gran cantidad de código Transact-SQL o se realiza repetidas veces, los procedimientos almacenados pueden ser más rápidos que los lotes de código Transact-SQL. Los procedimientos son analizados y optimizados en el momento de su creación, y es posible utilizar una versión del procedimiento que se encuentra en la memoria después de haberlo ejecutado una primera vez. Las instrucciones de Transact-SQL que se envían varias veces desde el cliente cada vez que deben ejecutarse tienen que ser compiladas y optimizadas siempre que SQL Server las ejecuta.

- Pueden reducir el tráfico de red.

Una operación que necesite centenares de líneas de código Transact-SQL puede realizarse mediante una sola instrucción que ejecute el código en un procedimiento, en vez de enviar cientos de líneas de código por la red.

- Pueden utilizarse como mecanismo de seguridad.

Es posible conceder permisos a los usuarios para ejecutar un procedimiento almacenado, incluso si no cuentan con permiso para ejecutar directamente las instrucciones del procedimiento.

Para crear un procedimiento almacenado de SQL Server se utiliza la instrucción CREATE PROCEDURE de Transact-SQL; puede utilizarse la instrucción ALTER PROCEDURE para modificar la instrucción y DROP PROCEDURE para eliminar el procedimiento almacenado. La definición del procedimiento almacenado contiene dos componentes principales: la especificación del nombre del procedimiento y sus parámetros, y el cuerpo del procedimiento que contiene las instrucciones de Transact-SQL que pueden realizar las operaciones del procedimiento.

Tipos de procedimientos almacenados

Los procedimientos almacenados pueden dividirse en procedimientos almacenados locales, almacenados en el servidor local, o remotos, almacenados en otro servidor. También pueden clasificarse en procedimientos almacenados definidos por el usuario y en procedimientos almacenados del sistema (system stored procedure), que son procedimientos previamente creados y que forman parte del servidor.

Un procedimiento almacenado puede contener hasta 255 parámetros y un código de retorno.

Nombre y tipo de datos de los parámetros:

Los parámetros deben tener un nombre único y comenzar con el símbolo @ (arroba). Al asignarle un nombre al parámetro también se le debe definir su tipo de dato.

Tipo de parámetro

Todos los parámetros creados se consideran como de entrada, es decir, que reciben datos del programa que invocó al procedimiento almacenado. Adicionando la palabra OUTPUT a la definición del parámetro, el procedimiento puede retornar el valor actual del parámetro al programa que lo invocó.

Códigos de retorno

Los procedimientos almacenados pueden retornar un valor de tipo entero, llamado RETURN CODE, para indicar si su ejecución fue exitosa. Esto se puede realizar creando un parámetro de salida con la opcion OUTPUT y retornar al programa que los ejecutó con la ayuda del comando RETURN. Los siguientes son los códigos con su respectivo valor:

Valor	Significado
0	Procedimiento almacenado completado con éxito
1	Parámetro de entrada no especificado
2	Contenido del parámetro inválido

Creación:

Sintaxis:

CREATE PROCEDURE <nombre_procedure>
[@lista_parámetro_entrada]

[@lista_parámetros_salida OUTPUT]
AS
<conjunto_instrucciones>

Modificación:

Sintaxis:

ALTER PROCEDURE <nombre_procedure>
[@lista_parámetro_entrada]
[@lista_parámetros_salida OUTPUT]
AS
<conjunto_instrucciones>

Eliminación:

DROP PROCEDURE <nombre_procedure>

Ejecución:
Sintaxis:

EXEC|UTE <nombre_procedimiento_almacenado> [lista_datos_entrada]
[,<variable1_salida_proc_almacenado>=<variable_receptora1 OUTPUT,
<variableN_salida_proc_almacenado>=<variable_receptoraN OUTPUT]

Ejemplos:

```
CREATE PROCEDURE MostVehiculo AS
select * from vehiculo

CREATE PROCEDURE ListaProp1
@Mcedula varchar (12) = '70000000'
 AS
if @mcedula is null
begin
  print "Error:  parámetro no hallado"
  return
end
select * from propietario where cedula = @mcedula

CREATE PROCEDURE ListaProp
@Mcedula varchar (12) = Null
 AS
if @mcedula is null
begin
  print "Error:  parámetro no hallado"
  return
end
select * from propietario where cedula = @mcedula

CREATE PROCEDURE BorrVehiculo1
@MnroPlaca varchar(6)
AS
delete from vehiculo where nroplaca = @Mnroplaca
delete from copiavehiculo where nroplaca = @Mnroplaca
```

```
CREATE PROCEDURE MostMarca
@mnroplaca varchar(6),
@mmarca varchar (20) output
As
select @mmarca = marca from vehiculo where nroplaca = @mnroplaca
```

Ejecución:

```
DECLARE @mimarca2 varchar (20)
exec mostmarca 'tis456', @mmarca = @mimarca2 OUTPUT
select @mimarca2
print 'La marca es: ' + @mimarca2
```

PRÁCTICA DE PROCEDIMIENTOS ALMACENADOS

Para este taller tenga en cuenta la base de datos ACADEMICO.
Se sugiere que los nombres de los procedimientos almacenados tengan los dos últimos dígitos de su Pc).

Crear procedimientos almacenados para:

1. Listar los alumnos que perdieron SQL
2. Listar los alumnos que perdieron, al menos, una asignatura.
3. Retornar e imprimir la cantidad de alumnos que perdieron cualquier asignatura, indicando el código de la asignatura
4. Retornar e imprimir la cantidad que pueden habilitar cualquier asignatura, indicando el nombre de la asignatura.
5. **Devolver e imprimir los nombres y apellidos de los alumnos que tienen la mejor nota en cualquier asignatura, indicando el nombre de ésta**.
6. Mostrar la cantidad de alumnos de cualquier programa, indicando el nombre del programa.
7. Listar los alumnos que pueden habilitar, por los datos: Apellidos, Nombres, Nombre de la asignatura y definitiva.
8. Listar las notas de cualquier alumno, por los datos: Apellidos, Nombres, Nombre de la asignatura y definitiva.
9. Retornar e imprimir el promedio de los alumnos de cualquier asignatura, indicando el nombre de la asignatura.
10. **Devolver e imprimir la cantidad de asignaturas que debe habilitar un alumno cualquiera, indicando su identificación. Se debe verificar que la cantidad de asignaturas no supere a 2**.
11. Agregar un nuevo programa
12. Cambiar la definitiva de cualquier alumno, indicando el código, en cualquier asignatura, indicando el nombre de la asignatura.
13. Eliminar cualquier programa, indicando el nombre del programa (se sugiere el programa anexado en el numeral 11.)