



Procedimientos almacenados en Transact SQL

Un procedimiento es un programa dentro de la base de datos que ejecuta una acción o conjunto de acciones específicas.

Un procedimiento tiene un nombre, un conjunto de parámetros (opcional) y un bloque de código.

En **Transact SQL** los procedimientos almacenados pueden devolver valores (numerico entero) o conjuntos de resultados.

Para crear un procedimiento almacenado debemos emplear la sentencia **CREATE PROCEDURE**.

```
CREATE PROCEDURE <nombre_procedure> [@param1 <tipo>, ...]
```

```
AS
```

```
-- Sentencias del procedure
```

Para modificar un procedimiento almacenado debemos emplear la sentencia **ALTER PROCEDURE**.

```
ALTER PROCEDURE <nombre_procedure> [@param1 <tipo>, ...]
```

```
AS
```

```
-- Sentencias del procedure
```

El siguiente ejemplo muestra un procedimiento almacenado, denominado spu_addCliente que inserta un registro en la tabla "CLIENTES".

.NET

- ▶ [Introducción a.NET](#)
- ▶ [Conceptos generales .NET](#)
- ▶ [C#](#)
- ▶ [Tutorial C#](#)
- ▶ [ASP.NET](#)
- ▶ [ADO.NET](#)
- ▶ [LinQ](#)
- ▶ [WCF](#)

SQL

- ▶ [Tutorial SQL](#)

SQL Server

- ▶ [Transact-SQL](#)
- ▶ [Tutorial de Transact-SQL](#)

ORACLE

- ▶ [Tutorial PL/SQL](#)
- ▶ [PL/SQL](#)

Reina Roja

- ▶ [Humor](#)
- ▶ [Bibliografia](#)
- ▶ [Mapa devjoker.com](#)
- ▶ [Acerca de ...](#)

```

CREATE PROCEDURE spu_addCliente @nombre varchar(100),
                                @apellido1 varchar(100),
                                @apellido2 varchar(100),
                                @nifCif varchar(20),
                                @fxNacimiento datetime

AS

INSERT INTO CLIENTES
(nombre, apellido1, apellido2, nifcif, fxnacimiento) VALUES
(@nombre, @apellido1, @apellido2, @nifCif, @fxNacimiento)

```

Para la ejecución de un procedimiento almacenado debemos utilizar la sentencia **EXEC**. Cuando la ejecución del procedimiento almacenado es la primera instrucción del lote, podemos omitir el uso de **EXEC**.

El siguiente ejemplo muestra la ejecución del procedimiento almacenado anterior.

```

DECLARE @fecha_nacimiento datetime

set @fecha_nacimiento = convert(datetime, '13/05/1975', 103)

EXEC spu_addCliente 'Pedro', 'Herrarte', 'Sanchez',
                    '00000002323', @fecha_nacimiento

```

Siempre es deseable que las instrucciones del procedimiento estén dentro de un bloque **TRY CATCH** y controlados por una transacción.

```

ALTER PROCEDURE spu_addCliente @nombre varchar(100),
                                @apellido1 varchar(100),

```

Comentarios



```

        @apellido2 varchar(100),
        @nifCif varchar(20),
        @fxNacimiento datetime

    AS

    BEGIN TRY

        BEGIN TRAN

        INSERT INTO CLIENTES

        (nombre, apellido1, apellido2, nifcif, fxnacimiento) VALUES

        (@nombre, @apellido1, @apellido2, @nifCif, @fxNacimiento)

        COMMIT

    END TRY

    BEGIN CATCH

        ROLLBACK

        PRINT ERROR_MESSAGE()

    END CATCH

```



Si queremos que los parámetros de un procedimiento almacenado sean de entrada-salida debemos especificarlo a través de la palabra clave **OUTPUT** , tanto en la definición del procedure como en la ejecución.

El siguiente ejemplo muestra la definición de un procedure con parámetros de salida.

```

CREATE PROCEDURE spu_ObtenerSaldoCuenta @numCuenta varchar(20),

        @saldo decimal(10,2) output

    AS

    BEGIN

        SELECT @saldo = SALDO

```

```
FROM CUENTAS
WHERE NUMCUENTA = @numCuenta

END
```

Y para ejecutar este procedure:

```
DECLARE @saldo decimal(10,2)
EXEC spu_ObtenerSaldoCuenta '200700000001', @saldo output
PRINT @saldo
```

Un procedimiento almacenado puede devolver valores numericos enteros a través de la instrucción RETURN. Normalmente debemos utilizar los valores de retorno para determinar si la ejecución del procedimiento ha sido correcta o no. Si queremos obtener valores se recomienda utilizar parámetros de salida o funciones escalares (se verán mas adelante en este tutorial).

El siguiente ejemplo muestra un procedimiento almacenado que devuelve valores.

```
CREATE PROCEDURE spu_EstaEnNumerosRojos @numCuenta varchar(20)
AS
BEGIN
    IF (SELECT SALDO FROM CUENTAS
        WHERE NUMCUENTA = @numCuenta) < 0
    BEGIN
        RETURN 1
    END
    ELSE
```

```
        RETURN 0  
  
    END
```

El siguiente ejemplo muestra como ejecutar el procedure y obtener el valor devuelto.

```
DECLARE @rv int  
  
EXEC @rv = spu_EstaEnNumerosRojos '200700000001'  
  
PRINT @rv
```

Otra característica muy interesante de los procedimientos almacenados en Transact SQL es que pueden devolver uno o varios conjuntos de resultados.

El siguiente ejemplo muestra un procedimiento almacenado que devuelve un conjunto de resultados.

```
CREATE PROCEDURE spu_MovimientosCuenta @numCuenta varchar(20)  
  
AS  
  
BEGIN  
  
    SELECT @numCuenta,  
           SALDO_ANTERIOR,  
           SALDO_POSTERIOR,  
           IMPORTE,  
           FXMOVIMIENTO  
  
    FROM MOVIMIENTOS  
  
    INNER JOIN CUENTAS ON MOVIMIENTOS.IDCUENTA = CUENTAS.IDCUENTA  
  
    WHERE NUMCUENTA = @numCuenta
```

ORDER BY FXMOVIMIENTO DESC

END

La ejecución del procedimiento se realiza normalmente.

EXEC spu_MovimientosCuenta '200700000001'

El resultado de la ejecución ...

NUMCUENTA	SALDO_ANTERIOR	SALDO_POSTERIOR	IMPORTE	FXMOVIMIENTO
2007000000001	50.99	100.99	50.00	2007-08-25 16:18:36.490
2007000000001	0.99	50.99	50.00	2007-08-23 16:20:41.183
2007000000001	50.99	0.99	50.00	2007-08-23 16:16:29.840
2007000000001	0.99	50.99	50.00	2007-08-23 16:14:05.900

[Inicio](#) | [Indice](#)



[Anterior](#)

[Indice](#)

[Siguiente](#)



[Version para imprimir](#)

Procedimientos almacenados en Transact SQL

Pedro Herrarte Sánchez



Pedro Herrarte, es consultor independiente, ofreciendo servicios de consultoría, análisis, desarrollo y formación. Posee mas de diez años de experiencia trabajando para las principales empresas de España. Es especialista en tecnologías .NET, entornos Web (ASP.NET, ASP.NET MVC, jQuery, HTML5), bases de datos (SQL Server y ORACLE) e integración de sistemas. Es experto en desarrollo (C#, VB.Net, T-SQL, PL/SQL, , ASP, CGI , C, Pro*C, Java, Essbase, Vignette, PowerBuilder y Visual Basic ...) y bases de datos (SQL Server y ORACLE). Pedro es MCP y [MAP 2012](#), es fundador, diseñador y programador de www.devjoker.com.



Twitter



Seguir a @PedroHerrarte


📅 Fecha de alta: 02/09/2007 📅 Última actualización: 02/09/2007 👁️ Visitas totales: 394218

Valorar el contenido:





Me gusta

A 2394 personas les gusta esto. [Regístrate](#) para ver qué les gusta a tus amigos.



devjoker.com
2394 Me gusta

 Me gusta esta página

 Compartir

Sé el primero de tus amigos en indicar que te gusta.

Últimas consultas realizadas en los foros

Últimas preguntas sin contestar en los foros de devjoker.com

01/12/2009 **estoy en problemas**

2 5 ★★★★★ ORACLE
1279

24/11/2007 **Crear documento de Word desde Java.**

13 5 ★★★★★ Java
11936

10/10/2007 **Actualizar Datos desde un Archivo TXT**

1 5 ★★★★★ SQL
1792

10/07/2007 **Comparar cadenas en PL/SQL**

12 5 ★★★★★ ORACLE
9248

16/04/2007 **No puedo arrancar servicio LISTENER**

2 5 ★★★★★ ORACLE
2942

Transact SQL C# WCF ASP.NET PL/SQL SQL LINQ
TFS CSS BSD ADO.NET Basic .NET Visual Studio ASP
Foros Windows Linux
Java SQL Server Entity Framework ORACLE RSS



www.devjoker.com | [Siguenos en facebook!](#) | [Acerca de ...](#) | [djk Contratación](#) | [Mapa del site](#)