

Procedimientos almacenados

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 (numérico entero) o conjuntos de resultados.

```
CREATE PROCEDURE <nombre_procedure> [@param1 <tipo>, ...]  
AS  
-- Sentencias del procedimiento  
GO
```

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

```
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)  
GO
```

```
CREATE PROCEDURE Dice_Hola
AS
PRINT 'Hola';
```

go

```
EXEC Dice_Hola;
```

```
CREATE PROCEDURE Dice_Palabra
@palabra CHAR(30)
AS
PRINT @palabra;
```

GO

```
EXEC Dice_Palabra 'Lo que quiera';
EXEC Dice_Palabra 'Otra cosa';
EXEC Dice_Palabra;
```

```
CREATE PROCEDURE Dice_Palabra2 @palabra CHAR(30)='No indicó nada'
AS
PRINT @palabra;
```

GO

```
EXEC Dice_Palabra2 'Lo que quiera';
EXEC Dice_Palabra2 'Otra cosa';
EXEC Dice_Palabra2;
```

```
create table prueba
(dato1 integer,
 dato2 varchar(30))
```

Insertar datos...

```
select dato1,dato2 from prueba;
```

```
CREATE PROCEDURE VerTabla
AS
    select dato1,dato2 from prueba;
go

exec VerTabla;
```

```
CREATE PROCEDURE VerTabla2 @limite integer
AS
    select dato1,dato2 from prueba
    where dato1>@limite;
go

exec VerTabla2 2;
```

```
CREATE PROCEDURE VerTabla3 @limite integer, @filtro varchar(30)
AS
    select dato1,dato2 from prueba
    where dato1>@limite and dato2 like @filtro
go

exec VerTabla3 1, 'dato%';
exec VerTabla3 @filtro='d%', @limite=2;
```

```
CREATE PROCEDURE VerTabla4 @limite integer, @filtro varchar(30) = '%'
AS
    select dato1,dato2 from prueba
    where dato1>@limite and dato2 like @filtro
go

exec VerTabla4 @limite=2;
```

```
CREATE PROCEDURE ContarRegistros
    @limite integer,
    @filtro varchar(30) = '%',
    @resultado integer output
AS
    select @resultado=(select count(*) from prueba
    where dato1>@limite and dato2 like @filtro)
go

DECLARE @valor AS integer;
EXEC ContarRegistros 1, 'a%', @valor OUTPUT;
PRINT @valor;
go
```

También podemos usar:
Set @resultado=...

Parámetros de entrada

Los parámetros de entrada posibilitan pasar información a un procedimiento.

Para que un procedimiento almacenado admita parámetros de entrada se deben declarar variables como parámetros al crearlo. La sintaxis es:

```
create proc NOMBREPROCEDIMIENTO  
@NOMBREPARAMETRO TIPO =VALORPORDEFEECTO  
as SENTENCIAS;
```

```
create procedure pa_libros_autor  
    @autor varchar(30)  
as  
    select titulo, editorial, precio  
    from libros  
    where autor= @autor;  
go
```


Parámetros de salida

Los procedimientos almacenados pueden devolver información, para ello se emplean parámetros de salida. El valor se retorna a quien realizó la llamada con parámetros de salida. Para que un procedimiento almacenado devuelva un valor se debe declarar una variable con la palabra clave "output" al crear el procedimiento:

```
create procedure NOMBREPROCEDIMIENTO
@PARAMETROENTRADA TIPO =VALORPORDEFECTO,
@PARAMETROSALIDA TIPO=VALORPORDEFECTO output
as
    SENTENCIAS
select @PARAMETROSALIDA=SENTENCIAS;
```

```
create procedure pa_autor_sumaypromedio
@autor varchar(30)='% ',
@suma decimal(6,2) output, @promedio decimal(6,2) output
as
select titulo,editorial,precio from libros where autor like @autor
select @suma=sum(precio) from libros where autor like @autor
select @promedio=avg(precio) from libros where autor like @autor;
go
```