



BASE DE DATOS AVANZADO I

Unidad 4: Programación Transact SQL

Tema 8: Procedimientos Almacenados - Parte I





Tema 08: Procedimientos Almacenados

4.1. Tema 8: Procedimientos Almacenados

4.1.1. Definición y tipos.

4.1.2. Construcción de procedimientos almacenados.

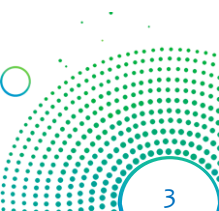
4.1.3. Manejo de parámetros: valores de entrada, valores de retorno.





Capacidades

1. Implementa procedimientos almacenados, funciones, en todos sus tipos, y triggers para garantizar el alto rendimiento de la base de datos, al hacer las transacciones





1. Definición y tipos

¿Qué es un procedimiento almacenado?

- Los procedimientos almacenados son grupos formados por instrucciones SQL y el lenguaje de control de flujo.
- Cuando se ejecuta un procedimiento, se prepara un plan de ejecución para que la subsiguiente ejecución sea muy rápida.





1. Definición y tipos

Procedimiento Almacenado

Los procedimientos almacenados pueden:

- ✓ Incluir parámetros.
- ✓ Llamar a otros procedimientos.
- ✓ Devolver un valor de estado a un procedimiento de llamada o lote para indicar el éxito o el fracaso del mismo y la razón de dicho fallo.
- ✓ Devolver valores de parámetros a un procedimiento de llamada o lote.
- ✓ Ejecutarse en SQL Server remotos.

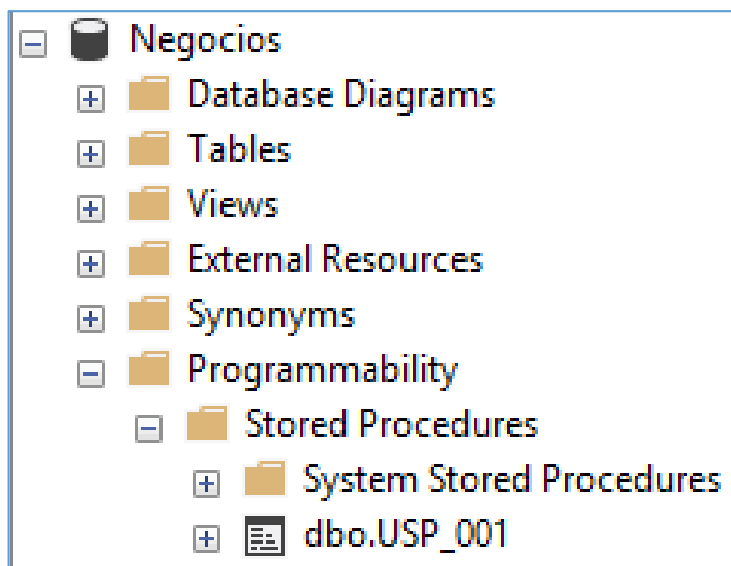




1. Definición y tipos

Tipos de procedimiento almacenado

- Existen 02 tipo de procedimientos almacenados:
 1. Procedimientos Almacenados del sistema (SP...).
 2. Procedimiento Almacenados del usuario (USP...).





1. Definición y tipos

Tipos de procedimiento almacenado

- Los procedimientos almacenados del usuario son módulos o rutinas que encapsulan código para su reutilización. Incluyen parámetros de entrada, devolver resultados y mensajes, invocar instrucciones del Lenguaje de Definición de Datos (DDL) e instrucciones de Lenguaje de Manipulación de Datos (DML), así como devolver parámetros de salida
- Los procedimientos almacenados del sistema son rutinas proporcionadas por el proveedor y orientadas a las actividades de mantenimiento y administración del gestor de base de datos





1. Definición y tipos

- *¿Cómo se visualiza los Procedimiento Almacenados de la Base de datos?*
- Se consulta el catálogo (SYS. PROCEDURES) de procedimientos
- almacenado es en el servidor y la base de datos.

```
/*Listar los procedimientos almacenados*/  
Select * from sys.procedures  
go
```





2. Construcción de procedimientos almacenados

¿Cómo se crea un procedimiento almacenado?

- Los procedimientos almacenados se crean con CREATE PROCEDURE. Para ejecutar un procedimiento almacenado, ya sea un procedimiento del sistema o uno definido por el usuario, use el comando EXECUTE.
- También, puede utilizar el nombre del procedimiento almacenado solo, siempre que sea la primera palabra de una instrucción o lote.





Fase 1: Crea el plan de ejecución de un Procedimiento Almacenado.

- Estructura de la Creación de un procedimiento almacenado.

CREATE PROCEDURE nombreProc

<lista de parámetros>

AS

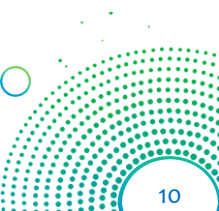
BEGIN

.....

Lista de sentencias T.SQL

.....

END





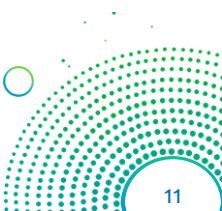
2. Construcción de procedimientos almacenados

Fase 2: Ejecuta el plan de ejecución de un Procedimiento Almacenado.

- Ejecución de un procedimiento almacenado.

EXECUTE nombreProc

go





2. Construcción de procedimientos almacenados

Ejemplo de un Procedimiento Almacenado.

- Crear un SP sin parámetros, que realice una consulta.

```
--Fase 01: Crear el plan
Create Procedure USP_001
As
Begin
    Select * from Ventas.clientes
End
go

--Fase 02: Ejecutar el plan
Execute USP_001
go
```





2. Construcción de procedimientos almacenados

¿Cómo se Modifica un Procedimiento Almacenado?

- Se modifica con ALTER PROCEDURE. Luego se ejecuta el procedimiento almacenado.
- Estructura para modificar un procedimiento almacenado.

ALTER PROCEDURE nombreProc

<lista de parámetros>

AS

BEGIN

.....

Lista de sentencias T.SQL

.....

END





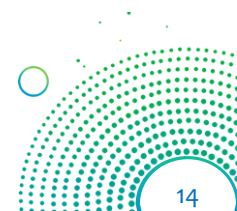
2. Construcción de procedimientos almacenados

Ejemplo de modificar un Procedimiento Almacenado.

- Modificar el procedimiento anterior para incluir parámetro.

```
--Fase 01: Modificar el plan
Alter Procedure USP_001
    @p_idpais char(3) --Parámetro
As
Begin
    Select * from Ventas.clientes
    Where idpais =@p_idpais
End
go

--Fase 02: Ejecutar el plan
Execute USP_001 '003'
go
```






2. Construcción de procedimientos almacenados

Ejemplo de modificar un Procedimiento Almacenado.

- El resultado sería así. Solo visualiza el valor del parámetro.*



Results		Messages			
	IdCliente	NomCliente	DirCliente	idpais	fonoCliente
1	BOTTM	Bottom-Dollar Markets	23 Tsawassen Blvd.	003	(604) 555-3745
2	FISSA	FISSA Fabrica Inter. Salchichas S.A.	C/ Moralarzal, 86	003	(91) 555 55 93
3	FOLKO	Folk och fä HB	Åkergatan 24	003	
4	ISLAT	Island Trading	Garden House\r\nCrowther Way	003	
5	MAISD	Maison Dewey	Rue Joseph-Bens 532	003	(02) 201 24 68
6	OTTIK	Ottiles Käseladen	Mehrheimerstr. 369	003	0221-0765721
7	PERIC	Pericles Comidas clasicas	Calle Dr. Jorge Cash 321	003	(5) 545-3745
8	REGGC	Reggiani Caseifici	Strada Provinciale 124	003	0522-556722



2. Construcción de procedimientos almacenados

¿Cómo se Elimina un Procedimiento Almacenado?

- Se elimina con DROP PROCEDURE.

DROP PROCEDURE nombreProc

GO

```
Drop Procedure USP_021  
go
```




2. Construcción de procedimientos almacenados

A tomar en cuenta

- Se debe tomar en cuenta que un procedimiento almacenado puede pertenecer a un esquema, por lo tanto se debe escribir primero el nombre del esquema, luego el nombre del procedimiento, separado por un punto.

`nombreEsquema.nombreProc`

- Ejemplo: Si el procedimiento está en el esquema Ventas.

`Ventas.USP_001`





3. Manejo de parámetros

Especificar Parámetros

- Un procedimiento almacenado se comunica con el programa que lo llama mediante sus parámetros.
- Los parámetros se pueden utilizar como variables estándar en el TRANSACT-SQL. El procedure también puede devolver valores al programa que lo llama mediante parámetros OUTPUT.
- Un procedimiento almacenado puede tener hasta 2.100 parámetros, cada uno de ellos con un nombre, un tipo de datos, una dirección y un valor predeterminado.

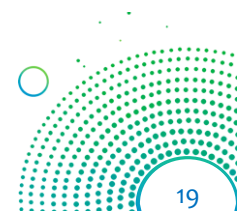




3. Manejo de parámetros

Especificar parámetros

- Especificar un nombre: Deben empezar por un solo carácter @, como una variable estándar de TRANSACT-SQL.
- Especificar una dirección: Puede ser de entrada, o de salida; por defecto es un parámetro de entrada. Para especificar un parámetro de salida, debe indicar la palabra clave OUTPUT en la definición del parámetro del procedimiento almacenado.
- Especificar un valor predeterminado: Puede crearse parámetros especificando un valor predeterminado para los mismos. Al ejecutar el procedure, se utilizará el valor predeterminado si no se ha especificado ningún otro.





3. Manejo de parámetros

Ejemplo con uso de parámetros

- Modificar el SP para dar un valor predefinido al parámetro.

```
/*Modificar el SP para dar un valor predefinido al parámetro*/
--Fase 01: Modificar el plan
Alter Procedure USP_001
    @p_idpais char(3) = '001' --Parámetro con valor definido
As
Begin
    Select * from Ventas.clientes
    Where idpais =@p_idpais
End
go
```




3. Manejo de parámetros

Ejemplo con uso de parámetros

- Ejecutar sin parámetro porque va asumir el valor definido en el procedimiento.

```
--Ejecutar sin parámetro  
Execute USP_001  
go
```



	IdCliente	NomCliente	DirCliente	idpais	fonoCliente
1	ALLLI	Alfredo Kimball	Ca Paujile 123	001	98765567
2	BLAUS	Blauer See Delikatessen	Forsterstr. 57	001	0621-08460
3	BONAP	Bon app	12, rue des Bouchers	001	91.24.45.41
4	CHOPS	Chop-suey Chinese	Hauptstr. 29	001	
5	ERNSH	Ernst Handel	Kirchgasse 6	001	7675-3426
6	FRANK	Frankenversand	Berliner Platz 43	001	089-0877451
7	FRANS	Franchi S.p.A.	Via Monte Bianco 34	001	011-4988261




3. Manejo de parámetros

Ejemplo con uso de parámetros

- Ejecutar con parámetro.

```
--Ejecutar con parámetro  
Execute USP_001 '004'  
go
```



	IdCliente	NomCliente	DirCliente	idpais	fonoCliente
1	AROUT	Around the Hom	120 Hanover Sq.	004	(71) 555-7788
2	COMMI	Comercio Mineiro	Av. dos Lusiadas, 23	004	
3	FAMIA	Familia Arquibaldo	Rua Oros, 92	004	(11) 555-9857
4	GALED	Galeria del gastronomo	Rambla de Cataluña, 23	004	(93) 203 4561
5	LEHMS	Lehmans Marktstand	Magazinweg 7	004	069-0245874
6	OLDWO	Old World Delicatessen	2743 Bering St.	004	(907) 555-2880



3. Manejo de parámetros

Ejemplo con uso de parámetros

- Utilizando parámetros fechas. Pedidos entre 02 fechas.

```
--Fase 01: Crear el plan
Create Procedure USP_002
    @p_Fecha01 Date, @p_Fecha02 Date
As
Begin
    Select * From Ventas.pedidoscabe
    Where FechaPedido Between @p_Fecha01 and @p_Fecha02
End
go

--Fase 02: Ejecutar procedimiento
Execute USP_002 '21/02/98', '05/03/98'
go
```






3. Manejo de parámetros

Ejemplo con uso de parámetros

- Utilizando parámetros fechas. Resultado.



	IdPedido	IdCliente	IdEmpleado	FechaPedido	FechaEntrega
1	10901	HILAA	4	1998-02-23 00:00:00.000	1998-03-23 00:00:00.000
2	10902	FOLKO	1	1998-02-23 00:00:00.000	1998-03-23 00:00:00.000
3	10903	HANAR	3	1998-02-24 00:00:00.000	1998-03-24 00:00:00.000
4	10904	WHITC	3	1998-02-24 00:00:00.000	1998-03-24 00:00:00.000
5	10905	WELLI	9	1998-02-24 00:00:00.000	1998-03-24 00:00:00.000
6	10906	WOLZA	4	1998-02-25 00:00:00.000	1998-03-11 00:00:00.000
7	10907	SPECD	6	1998-02-25 00:00:00.000	1998-03-25 00:00:00.000

3. Manejo de parámetros

Ejemplo con uso de parámetros

- Validar el SP anterior para que la primera fecha parámetro, sea menor a la segunda, de no ser así genere un error de severidad 16 y contrólolo con Try / Catch.

```
--Fase 01: Modificar el plan
Alter Procedure USP_002
    @p_Fecha01 Date, @p_Fecha02 Date
As
Begin
    Begin Try
        If @p_Fecha01 < @p_Fecha02
            Select * From Ventas.pedidoscabe
            Where FechaPedido Between @p_Fecha01 and @p_Fecha02
        Else
            RaisError ('Error... La primer fecha es mayor que la segunda..',16,1)
        End Try
        Begin Catch
            Print Error_message()
        End Catch
    End
go
```




3. Manejo de parámetros

Ejemplo con uso de parámetros

- Ejecutar el procedimiento almacenado colocando la primera fecha mayor a la segunda.

```
--Fase 02: Ejecutar SP  
Execute USP_002 '05/03/98', '21/02/98'  
go
```

 Messages

Error... La primer fecha es mayor que la segunda..



3. Manejo de parámetros

Ejemplo con uso de parámetros

- Ejecutar el procedimiento almacenado colocando la primera fecha menor a la segunda.

```
--Fase 02: Ejecutar SP  
Execute USP_002 '21/02/98', '05/03/98'  
go
```



	IdPedido	IdCliente	IdEmpleado	FechaPedido	FechaEntrega
1	10901	HILAA	4	1998-02-23 00:00:00.000	1998-03-23 00:00:00.000
2	10902	FOLKO	1	1998-02-23 00:00:00.000	1998-03-23 00:00:00.000
3	10903	HANAR	3	1998-02-24 00:00:00.000	1998-03-24 00:00:00.000
4	10904	WHITC	3	1998-02-24 00:00:00.000	1998-03-24 00:00:00.000
5	10905	WELLI	9	1998-02-24 00:00:00.000	1998-03-24 00:00:00.000
6	10906	WOLZA	4	1998-02-25 00:00:00.000	1998-03-11 00:00:00.000



3. Manejo de parámetros

- Ejemplo con uso de parámetros de entrada y salida
- Crear un procedimiento con Parámetro de entrada y de salida que reciba el Id Cliente y devuelva la cantidad de pedidos de dicho cliente.

```
--Fase 01: Crear el plan
Create Procedure USP_003
    @p_idcliente char(10),
    @p_CanPed Smallint OutPut
As
Begin
    Set @p_CanPed = (Select Count(IdPedido)
                    From Ventas.pedidoscabe
                    Where IdCliente = @p_idcliente)
End
go
```



3. Manejo de parámetros

Ejemplo con uso de parámetros de entrada y salida

- Ejecutar el procedimiento con Parámetro de entrada y de salida.

```
--Fase 02: Ejecutar el plan dentro de un Bloque
Begin
    Declare @v_idcliente char(10) = 'ANTON'
    Declare @v_CantidadPed smallint
    Execute USP_003 @p_idcliente = @v_idcliente, @p_canped = @v_CantidadPed OutPut
    Print 'El Cliente: ' + @v_idcliente
    Print 'Ha realizado ' + Cast(@v_CantidadPed As Varchar(15)) + ' pedidos'
End
```

```
Messages
El Cliente: ANTON
Ha realizado 7 pedidos
```





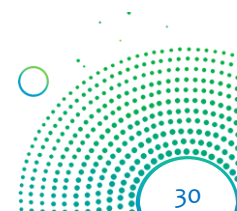
Ejercicio 1

- En la empresa, se pide implementar un procedimiento almacenado que permita registrar nuevos productos.
- Solución:

```
--Fase 01: Crear el plan
Create Procedure USP_IngProducto
@p_IdProducto      int,
@p_NomProducto     varchar(40),
@p_IdProveedor     int,
@p_IdCategoria     int,
@p_CantxUnidad     varchar(20),
@p_PrecioUnidad    smallmoney,
@p_UnidadesEnExistencia smallint,
@p_UnidadesEnPedido smallint
As
Begin
    Insert Compras.productos
    Values
    (@p_IdProducto, @p_NomProducto, @p_IdProveedor, @p_IdCategoria, @p_CantxUnidad,
    @p_PrecioUnidad, @p_UnidadesEnExistencia, @p_UnidadesEnPedido)
End
go
```

- Ahora ejecutar:

```
--Fase 02: Ejecutar el plan
Execute USP_IngProducto '100', 'Pastilla antialer', '1', '3', 'Uni', '2.5', '0', '100'
go
```





Ejercicio 2

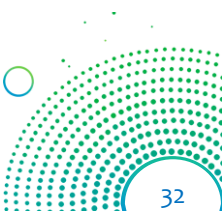
- Modificar el SP para validar la No existencia del Id Producto, así mismo validar la existencia de las llaves foráneas.
- Solución:

```
--Fase 01: Modificar el plan
Alter Procedure USP_IngProducto
@p_IdProducto      int,
@p_NomProducto     varchar(40),
@p_IdProveedor     int,
@p_IdCategoria     int,
@p_CantxUnidad     varchar(20),
@p_PrecioUnidad    smallmoney,
@p_UnidadesEnExistencia smallint,
@p_UnidadesEnPedido smallint
As
Begin
    If Not Exists (Select * from Compras.productos Where IdProducto = @p_IdProducto)
    Begin
        If Exists (Select * from Compras.proveedores Where IdProveedor = @p_IdProveedor) And
            Exists (Select * From Compras.categorias Where IdCategoria = @p_IdCategoria)
            Insert Compras.productos
            Values
            (@p_IdProducto, @p_NomProducto, @p_IdProveedor, @p_IdCategoria, @p_CantxUnidad,
            @p_PrecioUnidad, @p_UnidadesEnExistencia, @p_UnidadesEnPedido)
        Else
            RaisError ('Error.. con las foráneas de proveedor o categoria',10,1)
        End
    else
        RaisError ('Error....IdProducto repetido...Intenta nuevamente.',10,1)
    End
End
go
```




Conclusiones

- Un procedimiento almacenado de SQL Server es un grupo de una o varias instrucciones Transact-SQL o una referencia a un método de Common Runtime Language (CLR) de Microsoft .NET Framework .
- Los procedimientos se asemejan a las construcciones de otros lenguajes de programación.





Bibliografía

Microsoft (2017) Procedimientos almacenados (motor de base de datos). Recuperado de: <https://docs.microsoft.com/es-es/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-2017>



GRACIAS

**SEDE LIMA CENTRO**

Av. Uruguay 514
Cercado – Lima
Teléfono: 419-2900

SEDE INDEPENDENCIA

Av. Carlos Izaguirre 233
Independencia – Lima
Teléfono: 633-5555

SEDE BREÑA

Av. Brasil 714 – 792
(CC La Rambla – Piso 3)
Breña – Lima
Teléfono: 633-5555

SEDE TRUJILLO

Calle Borgoño 361
Trujillo
Teléfono: (044) 60-2000

SEDE SAN JUAN DE LURIGANCHO

Av. Próceres de la Independencia 3023-3043
San Juan de Lurigancho – Lima
Teléfono: 633-5555

SEDE BELLAVISTA

Av. Mariscal Oscar R. Benavides 3866 – 4070
(CC Mall Aventura Plaza)
Bellavista – Callao
Teléfono: 633-5555

SEDE AREQUIPA

Av. Porongoche 500
(CC Mall Aventura Plaza)
Paucarpata - Arequipa
Teléfono: (054) 60-3535