

# Práctica 26

## Procedimientos en SQL Server

### Objetivo

*En esta práctica se desarrollan procedimientos mediante el empleo del software **SQL Server**.*

### Introducción

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 y un bloque de código. Los procedimientos almacenados pueden devolver un valor o un conjunto de resultados.

### Equipo necesario

Una computadora con sistema operativo **Windows** que cuente con el software **SQL Server**.

### Metodología

1. Inicie el software **SSMS** y establezca la conexión con el servidor.
2. Oprima el botón **New Query** y copie el siguiente procedimiento, mediante el cual se pretende hacer el registro de nuevos empleados:

```
CREATE OR ALTER PROCEDURE NuevoEmpleado
    @cve int,
    @nom varchar(20),
    @ape varchar(20),
    @suc int,
    @sdo Decimal(10,2),
    @fin datetime
AS
INSERT INTO EMPLEADO
(Cve_Empleado, Nombre, Apellido, Puesto, Sucursal, Sueldo,
Fecha_Ingreso)
```

```
VALUES (@cve, @nom, @ape, @pue, @suc, @sdo, @fin)
```

3. En la lista desplegable que se encuentra en la barra de herramientas seleccione la base de datos **empresa**.
4. Oprima el botón **Execute** y se habrá creado el procedimiento.
5. Obtenga una imagen completa de la pantalla, mostrando el procedimiento completo y sin errores, consérvela como evidencia.
6. Oprima el botón New Query y copie el siguiente script, para ejecutar del procedimiento almacenado anterior, donde se hace una conversión de texto a fecha, (donde 103 = dd/mm/yyyy).

```
DECLARE @fecha_ing DATETIME  
SET @fecha_ing = convert(DATETIME, '18/10/2024', 103)  
EXEC dbo.NuevoEmpleado 123, 'Pedro', 'Herrera', 'Sup03', 180,  
15000, @fecha_ing
```

7. Ejecute el script para insertar los datos del nuevo empleado en la tabla.
8. Si vuelve a ejecutar el mismo script obtendrá un mensaje de error.
9. Obtenga una imagen completa de la pantalla, mostrando el mensaje de error obtenido, consérvela como evidencia.
10. Siempre es deseable que los mensajes de error del procedimiento se controlen dentro de un bloque **TRY CATCH** y formando parte de una transacción, por lo que regrese a la ventana donde escribió el procedimiento y agregue las líneas en rojo, para obtener el siguiente resultado:

```
CREATE OR ALTER PROCEDURE NuevoEmpleado  
    @cve int,  
    @nom varchar(20),  
    @ape varchar(20),  
    @pue varchar(20),  
    @suc int,  
    @sdo Decimal(10,2),  
    @fin datetime  
  
AS  
BEGIN TRY  
    BEGIN TRAN  
    INSERT INTO EMPLEADO  
    (Cve_Empleado, Nombre, Apellido, Puesto, Sucursal, Sueldo,  
    Fecha_Ingreso)  
    VALUES (@cve, @nom, @ape, @pue, @suc, @sdo, @fin)  
    COMMIT  
END TRY  
BEGIN CATCH  
    ROLLBACK  
    PRINT ERROR_MESSAGE()  
END CATCH
```



11. Oprima el botón **Execute** para actualizar el procedimiento.
12. Obtenga una imagen completa de la pantalla, mostrando el procedimiento completo y sin errores, consérvela como evidencia.
13. Oprima el botón **New Query** y haga la inserción de un nuevo empleado mediante el siguiente script:

```
DECLARE @fecha_ing datetime
SET @fecha_ing = convert(datetime, '18/10/2024', 103)
EXEC dbo.NuevoEmpleado 125, 'Rosa', 'Rosales', 'Ven03', 250,
7500, @fecha_ing
```
14. Ejecute el script para insertar los datos del nuevo empleado en la tabla, y verá que el procedimiento funciona exactamente igual.
15. Vuelva a ejecutar el script anterior con los mismos datos, y observe que el procedimiento es capaz de controlar la situación y enviar el mismo mensaje de error, pero sin indicar que falló el procedimiento:
16. Obtenga una imagen completa de la pantalla, mostrando el mensaje de error emitido por el procedimiento, consérvela como evidencia.
17. Cuando se desea que los parámetros de un procedimiento almacenado sean de entrada/salida, es necesario especificarlo a través de la cláusula **OUTPUT**, tanto en la definición del procedimiento como en su ejecución.
18. Ahora, seleccione la base de datos **banco** en la lista desplegable que se encuentra en la barra de herramientas.
19. Oprima el botón **New Query** y convierta la transacción hecha en la práctica 21 (puntos 15 al 21) en un procedimiento almacenado, para registrar movimientos entre dos cuentas, poniendo **CREATE OR ALTER PROCEDURE Traspaso** en vez de **DECLARE**, quite las asignaciones a las variables y agregue la cláusula **AS** a continuación, antes de iniciar la transacción.
20. Oprima el botón **Execute** y se habrá creado el procedimiento.
21. Obtenga una imagen completa de la pantalla, mostrando el procedimiento completo y sin errores, consérvela como evidencia.
22. Oprima el botón New Query y pruebe su procedimiento, haciendo los siguientes traspasos:

```
EXEC Traspaso 1000, '2024003', '2024001'
EXEC Traspaso 2500, '2024002', '2024004'
EXEC Traspaso 1500, '2024001', '2024002'
EXEC Traspaso 1500, '2024004', '2024005'
```
23. Como deberá demostrar, el procedimiento funciona correctamente y las transacciones se realizan con éxito, excepto en el último caso, donde la cuenta receptora no existe.
24. Obtenga una imagen completa de la pantalla, mostrando las cuatro ejecuciones del procedimiento, consérvela como evidencia.

25. Oprima el botón **New Query** y copie el siguiente procedimiento con parámetros de salida.

```
CREATE PROCEDURE ConsultaSaldo @numero varchar(12),@saldo
decimal(10,2) OUTPUT
AS
BEGIN
SELECT @saldo = Saldo FROM CUENTAS WHERE Numero = @numero
END
```

26. Oprima el botón **Execute** y se habrá creado el procedimiento.

27. Obtenga una imagen completa de la pantalla, mostrando el procedimiento completo y sin errores, consérvela como evidencia.

28. Oprima el botón **New Query** y copie el siguiente script, que ilustra la forma en que se debe ejecutar este procedimiento:

```
DECLARE @saldo decimal(10,2)
EXEC ConsultaSaldo '2024002', @saldo OUTPUT
PRINT CONCAT('Saldo: ', @saldo)
```

29. Ejecute el script para consultar el saldo de esa cuenta.

30. Obtenga una imagen completa de la pantalla, mostrando el mensaje emitido por el procedimiento, consérvela como evidencia.

31. Un procedimiento almacenado puede devolver valores enteros a través de la instrucción RETURN, normalmente se utilizan los valores de retorno para determinar si la ejecución del procedimiento ha sido correcta o no.

32. Oprima el botón **New Query** y copie el siguiente procedimiento con salida mediante el RETURN.

```
CREATE OR ALTER PROCEDURE Retiros @numero varchar(20)
AS
BEGIN
IF (SELECT count(*) FROM Movimientos WHERE Numero = @numero AND
Saldo_Nuevo < Saldo_Anterior) > 0
RETURN 1
ELSE
RETURN 0
END
```

33. Oprima el botón **Execute** y se habrá creado el procedimiento.

34. Obtenga una imagen completa de la pantalla, mostrando el procedimiento completo y sin errores, consérvela como evidencia.

35. Oprima el botón **New Query** y copie el siguiente script para ejecutar el procedimiento anterior.

```
DECLARE @rv int
EXEC @rv = Retiros'2024001'
```



```
PRINT @rv  
EXEC @rv = Retiros'2024002'  
PRINT @rv
```

36. Ejecute el script para saber si esas cuentas tienen movimientos de salida, en ambos casos se muestra el valor devuelto de **1**, dado que se han realizado retiros de esas cuentas.
37. Obtenga una imagen completa de la pantalla, mostrando las respuestas emitidas por el procedimiento, consérvela como evidencia.
38. Los procedimientos almacenados pueden devolver uno o varios conjuntos de resultados en una tabla.
39. Oprima el botón **New Query** y copie el siguiente procedimiento, que devuelve un conjunto de resultados en una tabla.

```
CREATE PROCEDURE MovimientosCuenta @numero varchar(12)  
AS  
BEGIN  
SELECT C.Numero, Saldo_Anterior, Saldo_Nuevo, IMPORTE,M.Fecha  
FROM MOVIMIENTOS M INNER JOIN CUENTAS C ON M.Numero = C.Numero  
WHERE C.Numero = @numero  
ORDER BY M.Fecha DESC  
END
```

40. Oprima el botón **Execute** y se habrá creado el procedimiento.
41. Obtenga una imagen completa de la pantalla, mostrando el procedimiento completo y sin errores, consérvela como evidencia.
42. Oprima el botón **New Query** y ejecute este procedimiento de forma directa con EXEC.

```
EXEC MovimientosCuenta '2024001'
```

43. Obtenga una imagen completa de la pantalla, mostrando la respuesta emitida por el procedimiento, consérvela como evidencia.
44. Cierre la aplicación SSMS.
45. Fin de la Práctica.

## Evidencias

El alumno deberá enviar al instructor las evidencias requeridas durante la realización de la práctica.

## Sugerencias didácticas

El instructor deberá orientar a los alumnos que tengan dificultades en la interpretación y la realización de las instrucciones de la práctica, pero sin darles las respuestas.

## Resultados

Se demuestra haber adquirido suficiente conocimiento sobre la creación de procedimientos en SQL Server.

## Bibliografía

- <https://learn.microsoft.com/en-us/sql/ssms/user-assistance-in-sql-server-management-studio?view=sql-server-ver16&viewFallbackFrom=sql-server-ver19>