

reporte practica 26

evidencias instruccion 5

```
CREATE OR ALTER PROCEDURE NuevoEmpleado
    @cve int,
    @nom varchar(20),
    @ape varchar(20),
    @suc int,
    @pue varchar(50),
    @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)
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-30T07:19:02.4324355-06:00

evidencias instruccion 9

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

100 %

Messages

Msg 2627, Level 14, State 1, Procedure dbo.NuevoEmpleado, Line 10 [Batch Start Line 0]
Violation of PRIMARY KEY constraint 'PK_Empleado'. Cannot insert duplicate key in object 'dbo.Empleado'.
The statement has been terminated.

Completion time: 2025-04-30T07:25:14.0719875-06:00

evidencias instruccion 12

```
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 TRAN
END TRY
BEGIN CATCH
    ROLLBACK TRAN
    PRINT ERROR_MESSAGE()
END CATCH
```

100 %

Messages Live Query Statistics

Commands completed successfully.

Completion time: 2025-04-30T07:27:30.2013739-06:00

evidencia instruccion 16

```
DECLARE @fecha_ing datetime
SET @fecha_ing = convert(datetime, '18/10/2024', 103)
EXEC dbo.NuevoEmpleado 125, 'Rosa', 'Rosales', 'Ven03', 250,
    7500, @fecha_ing
```

100 %

Messages

(0 rows affected)

Violation of PRIMARY KEY constraint 'PK_Empleado'. Cannot insert duplicate key in object 'dbo.i

Completion time: 2025-04-30T07:28:39.8420738-06:00

evidencias instruccion 21

```
CREATE OR ALTER PROCEDURE Traspaso (@CuentaOrigen INT, @CuentaDestino INT, @importe INT)
AS
BEGIN
    BEGIN TRANSACTION
    BEGIN TRY
        -- Actualizar saldo de la cuenta de origen
        UPDATE CUENTAS SET Saldo = Saldo - @importe WHERE Numero = @CuentaOrigen
        IF @@ROWCOUNT = 0
        BEGIN
            RAISERROR('No se encontró la cuenta de origen.', 16, 1);
            RETURN;
        END;

        -- Verificar saldo suficiente en la cuenta de origen
        IF (SELECT Saldo FROM CUENTAS WHERE Numero = @CuentaOrigen) < @importe
        BEGIN
            RAISERROR('Saldo insuficiente en la cuenta de origen.', 16, 1);
            RETURN;
        END;

        -- Transferir el importe a la cuenta de destino
        UPDATE CUENTAS SET Saldo = Saldo + @importe WHERE Numero = @CuentaDestino
        IF @@ROWCOUNT = 0
        BEGIN
            RAISERROR('No se encontró la cuenta de destino.', 16, 1);
            RETURN;
        END;

        COMMIT TRANSACTION
    END TRY
    CATCH
        ROLLBACK TRANSACTION
        RAISERROR('Error al realizar el traspaso.', 16, 1);
    END CATCH
END
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-30T07:37:57.3062342-06:00

error instruccion 24

```
EXEC Traspaso '2024003', '2024001', 1000
EXEC Traspaso '2024002', '2024004', 1000
EXEC Traspaso '2024001', '2024002', 1000
EXEC Traspaso '2024004', '2024005', 1000
```

100 %

Messages

(1 row affected)
Se ha producido un error!
Saldo insuficiente en la cuenta de origen.

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)

(1 row affected)
Se ha producido un error!
Saldo insuficiente en la cuenta de origen.

(1 row affected)

(1 row affected)

(0 rows affected)
Se ha producido un error!
No se encontró la cuenta de destino.

Completion time: 2025-04-30T07:47:39.8282650-06:00

evidencias instruccion 27

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

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-30T07:50:06.4910489-06:00

evidencias instruccion 30

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

100 %

Messages

Saldo: 3000.00

Completion time: 2025-04-30T07:51:01.0157049-06:00

evidencia instruccion 34

```
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
```

100 %

Messages

Commands completed successfully.

Completion time: 2025-04-30T07:52:27.9938773-06:00

evidencia instruccion 37

```
DECLARE @rv int
EXEC @rv = Retiros '2024001'
PRINT @rv
EXEC @rv = Retiros '2024002'
PRINT @rv
```

100 %

Messages

1
0

Completion time: 2025-04-30T07:53:56.4337523-06:00

evidencia instruccion 41

```
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
```

00 %

Messages

Commands completed successfully.

Completion time: 2025-04-30T07:55:08.5801737-06:00

evidencias instruccion 43

```
EXEC MovimientosCuenta '2024001'
```

100 %

Results Messages

	Numero	Saldo_Anterior	Saldo_Nuevo	IMPORTE	Fecha
1	2024001	2000.00	500.00	1500.00	2025-04-13
2	2024001	500.00	-1000.00	1500.00	2025-04-13