



FACENA – UNNE

Licenciatura en Sistemas de Información

Bases de Datos I

Transacciones y Rollback en SQL Server

Autor: Grupo 6

Año: 2025

1. INTRODUCCIÓN

Una transacción es una unidad lógica de trabajo que agrupa operaciones para que se ejecuten de forma atómica: o bien todas tienen éxito (COMMIT) o ninguna tiene efecto (ROLLBACK). Esto protege la consistencia e integridad de los datos ante errores o fallos.

2. TRANSACCIONES EXPLÍCITAS CON TRY...CATCH

SQL Server permite manejar errores con bloques TRY...CATCH y confirmar o deshacer cambios mediante COMMIT y ROLLBACK. A continuación se analiza un caso exitoso y otro fallido a partir de procedimientos almacenados.

2.1 sp_registrar_socio_suscripcion_pago (caso de éxito)

Este procedimiento inserta un socio, crea su suscripción y registra un pago en una única transacción. Si todo funciona, se confirma con COMMIT.

```
EXEC sp_registrar_socio_suscripcion_pago
@dni=88887777,
@nombre_socio='TransaccionOK',
@apellido_socio='Ejemplo',
@fecha_nac='1995-05-05',
@id_contacto=1,
@id_tipo_suscripcion=1,
@monto=12000,
@id_socio_out=@idSocioOK OUTPUT,
@id_suscripcion_out=@idSusOK OUTPUT,
@id_pago_out=@idPagoOK OUTPUT;
```

2.2 sp_demo_error_tx (caso con rollback)

En este procedimiento se fuerza un error con la operación 1/0 dentro de la transacción. El CATCH ejecuta ROLLBACK para deshacer los cambios.

```
BEGIN TRY
    EXEC sp_demo_error_tx
        @dni=77776666,
        @nombre_socio='TransaccionFALLA',
        @apellido_socio='Rollback',
        @fecha_nac='1999-09-09',
        @id_contacto=1,
        @id_tipo_suscripcion=2;
END TRY
BEGIN CATCH
    PRINT 'ERROR DETECTADO → ROLLBACK ejecutado';
    SELECT ERROR_NUMBER() AS CodigoError, ERROR_MESSAGE() AS MensajeError;
END CATCH;
```

3. PATRONES ÚTILES EN TRANSACCIONES

- Uso de @@TRANCOUNT para detectar si existe una transacción activa y decidir entre BEGIN TRAN o SAVE TRAN (puntos de guardado).
- Confirmación con COMMIT únicamente si el procedimiento abrió la transacción, evitando confirmar transacciones externas por error.
- Manejo de errores con THROW para propagar el problema al llamador luego del ROLLBACK.
- Verificaciones finales que demuestran que los datos del caso fallido no se persisten tras el ROLLBACK.

4. CONCLUSIÓN

Como grupo comprendimos que el control transaccional asegura la atomicidad y consistencia de los cambios. El uso combinado de TRY...CATCH, @@TRANCOUNT, COMMIT/ROLLBACK y SAVEPOINTS permite construir procedimientos robustos ante fallas y mantener la integridad del sistema.