**TRANSACT SQL**

**TEMA 3: TRANSACCIONES**

**Transacciones en SQL Server**

Una transacción es un conjunto de instrucciones u operaciones de manipulación de datos que se ejecutan como un único bloque o en una misma unidad de trabajo. Esto quiere decir que si falla una operación fallan todas. El ejemplo más claro son las transacciones bancarias.

SQL server trabaja por defecto con Transacciones de confirmación automática, es decir, cada instrucción individual es una transacción y se confirma automáticamente. Las inserciones, borrados y actualizaciones (insert, delete,update) internamente generan una transacción. Esto quiere decir que por ejemplo al borrar un registro se produjera un error, no se borraría ninguno.

**TRANSACCIONES IMPLÍCITAS Y EXPLÍCITAS**

**Transacciones Implícitas**

Una nueva transacción que se inicia automáticamente cuando se ejecuta una instrucción que realiza modificaciones en los datos de la BD, pero cada transacción se completa explícitamente con una instrucción COMMIT o **ROLLBACK TRANSACTION.**

**Activar las transacciones implícitas**

**SET IMPLICT\_TRASACTION ON**

**Desactivar las transacciones implícitas**

**SET IMPLICT\_TRASACTION OFF**

**Transacciones Explícitas**

Cada transacción se inicia explícitamente con la instrucción **BEGIN TRANSACTION** y se termina explícitamente con una instrucción **COMMIT** o **ROLLBACK TRANSACTION.**

Inicio de una transacción:

|  |  |
| --- | --- |
|  | BEGIN TRAN[SACTION][NombreTransaccion] |

Validación de transacción:

|  |  |
| --- | --- |
|  | COMMIT TRAN[SACTION] [NombreTransaccion] |

Rechazar o anular la transacción:

|  |  |
| --- | --- |
|  | ROLLBACK TRAN[SACTION] [NombreTransaccion] |

Para verificar el número de transacciones activas se utiliza la variable del sistema **@@TRANCOUNT.**

**Estructura de un script con transacciones y commit y rollback**

BEGIN TRY

BEGIN TRAN T1

S1

S2

…

SN

COMMIT TRAN T1

END TRY

BEGIN CATCH

ROLLBACK TRAN

….

END CATCH

**Transacciones Anidadas**

Podemos anidar transacciones. Cuando anidamos varias transacciones la instrucción COMMIT afectará a la última transacción abierta, pero ROLLBACK afectará a todas las transacciones abiertas.

**Puntos de control o recuperación SAVEPOINT**

Los puntos de control o recuperación (SavePoint) permiten manejar las transacciones por pasos, pudiendo hacer rollbacks hasta un punto marcado por el savepoint y no por toda la transacción.

Declaración de punto de control:

|  |  |
| --- | --- |
|  | SAVE TRAN[SACTION] [NombrePuntoControl] |

Anulación de Transacción o punto de control:

|  |  |
| --- | --- |
|  | ROLLBACK TRAN[SACTION] [NombreTransaccion | NombrePuntoControl] |

Ejemplo de estructura

|  |  |
| --- | --- |
|  | BEGIN TRAN Modificación  UPDATE Tabla SET ....  UPDATE Tabla1 SET ....  SAVE TRAN a  UPDATE Tabla2 SET ....  UPDATE Tabla3 SET ....  ROLLBACK TRAN a    ROLLBACK TRAN Modificación  COMMIT TRAN Modificación |
|  |  |

El primer Rollback guardaría las que están en el Begin y el punto de salvamento, deshaciendo las demás órdenes.

Ejemplo de transacciones usando @@ERROR, con esto se almacena el número del error que se ha producido en última instancia. Se actualiza en cada instrucción, por eso después de cada sentencia se debe comprobar el valor de la variable para ver si tiene error. Lo que se hace es crear variables en cada instrucción almacenando el valor de @@ERROR en cada una de ellas. El valor de las variables debe ser entero.