

Informe: Transact-SQL y Control de Transacciones

Introducción a Transact-SQL (T-SQL)

Transact-SQL (T-SQL) es una extensión del lenguaje SQL estándar, desarrollada por Microsoft para trabajar con SQL Server. T-SQL incluye características adicionales como control de flujo, manejo de errores, variables y procedimientos almacenados, permitiendo una mayor flexibilidad y control sobre las operaciones en bases de datos. Una de sus principales fortalezas es el manejo de transacciones, lo que asegura la integridad y consistencia de los datos.

1. Transacciones y TCL (Transaction Control Language)

Las transacciones en T-SQL son bloques de operaciones que se ejecutan como una unidad indivisible. Si alguna operación dentro de la transacción falla, todas las operaciones deben revertirse para garantizar que la base de datos permanezca en un estado coherente. Los comandos TCL controlan estas transacciones:

- **BEGIN TRANSACTION:** Inicia una transacción.
- **COMMIT:** Confirma los cambios realizados por la transacción.
- **ROLLBACK:** Revierte los cambios realizados en caso de error.

Ejemplo:

```
sql
Copiar código
BEGIN TRANSACTION;
UPDATE Clientes SET Saldo = Saldo - 100 WHERE ID = 1;
IF @@ERROR <> 0
    ROLLBACK;
ELSE
    COMMIT;
```

2. Elementos Auxiliares en T-SQL

2.1 Comando USE

Permite cambiar el contexto de la base de datos con la que se está trabajando.

```
sql
Copiar código
USE MiBaseDeDatos;
```

2.2 Variables

Las variables almacenan valores temporales que pueden ser utilizados durante la ejecución de una consulta.

```
sql
Copiar código
DECLARE @NombreCliente VARCHAR(50);
SET @NombreCliente = 'Juan';
PRINT @NombreCliente;
```

2.3 GO

Marca el final de un bloque de comandos. Es necesario para ejecutar múltiples instrucciones de forma secuencial.

```
sql
Copiar código
USE MiBaseDeDatos;
GO
SELECT * FROM Clientes;
GO
```

2.4 EXECUTE

Ejecuta procedimientos almacenados o comandos dinámicos.

```
sql
Copiar código
EXEC MiProcedimientoAlmacenado;
```

2.5 PRINT

Muestra mensajes o resultados en la consola.

```
sql
Copiar código
PRINT 'Operación completada con éxito';
```

3. Control de Flujo en T-SQL

IF...ELSE

Permite ejecutar diferentes bloques de código según una condición.

```
sql
Copiar código
IF @Saldo > 0
    PRINT 'Saldo positivo';
ELSE
    PRINT 'Saldo insuficiente';
```

CASE

Es similar a un **IF**, pero se utiliza dentro de una consulta.

```
sql
Copiar código
SELECT Nombre,
       CASE
         WHEN Saldo > 0 THEN 'Activo'
         ELSE 'Inactivo'
       END AS Estado
FROM Clientes;
```

WHILE

Crea bucles que se ejecutan mientras una condición sea verdadera.

```
sql
Copiar código
DECLARE @Contador INT = 1;
WHILE @Contador <= 5
BEGIN
  PRINT @Contador;
  SET @Contador = @Contador + 1;
END;
```

GOTO

Salta a una etiqueta específica en el código.

```
sql
Copiar código
GOTO Fin;
PRINT 'Este mensaje no se mostrará';
Fin:
PRINT 'Fin del script';
```

4. Procedimientos Almacenados y Cursores

4.1 Procedimientos Almacenados

Son conjuntos de instrucciones SQL que se almacenan en la base de datos y se pueden ejecutar repetidamente.

```
sql
Copiar código
CREATE PROCEDURE MiProcedimiento
AS
BEGIN
  SELECT * FROM Clientes;
END;
```

4.2 Cursores

Permiten recorrer registros uno a uno. Se utilizan cuando es necesario procesar cada fila individualmente.

- **Declaración de Cursor:**

```
sql
Copiar código
DECLARE MiCursor CURSOR FOR SELECT Nombre FROM Clientes;
OPEN MiCursor;
FETCH NEXT FROM MiCursor INTO @Nombre;
```

- **Funciones del Cursor:**
 - **FIRST:** Mueve el cursor al primer registro.
 - **LAST:** Mueve el cursor al último registro.
 - **NEXT:** Mueve el cursor al siguiente registro.
 - **PRIOR:** Mueve el cursor al registro anterior.
 - **RELATIVE:** Se mueve un número específico de filas.
 - **ABSOLUTE:** Mueve el cursor a una posición específica.

Tipos de Cursores

- **Estático:** No permite modificaciones mientras se recorre.
- **Dinámico:** Refleja los cambios en los datos mientras se recorre.
- **De sólo lectura:** No permite actualizaciones.
- **De desplazamiento hacia adelante:** Solo se puede recorrer en una dirección.

5. Manejo de Errores y Excepciones

TRY/CATCH

Permite capturar errores y manejar excepciones de manera controlada.

```
sql
Copiar código
BEGIN TRY
    INSERT INTO Clientes (ID, Nombre) VALUES (1, 'Juan');
END TRY
BEGIN CATCH
    PRINT 'Ocurrió un error';
END CATCH;
```

6. Triggers (Desencadenadores)

Desencadenadores DML

Se ejecutan automáticamente en respuesta a eventos **INSERT**, **UPDATE** o **DELETE** en una tabla.

```
sql
Copiar código
CREATE TRIGGER trg_AfterInsert ON Clientes
AFTER INSERT
AS
BEGIN
    PRINT 'Nuevo cliente agregado';
END;
```

Desencadenadores DDL

Se activan con cambios en la estructura de la base de datos, como la creación o eliminación de tablas.

```
sql
Copiar código
CREATE TRIGGER trg_AfterCreate ON DATABASE
FOR CREATE_TABLE
AS
BEGIN
    PRINT 'Se creó una nueva tabla';
END;
```

Desencadenadores Logon

Se ejecutan cuando un usuario inicia sesión en el servidor SQL.

```
sql
Copiar código
CREATE TRIGGER trg_Logon
ON ALL SERVER
FOR LOGON
AS
BEGIN
    PRINT 'Nuevo inicio de sesión';
END;
```

Conclusión

Transact-SQL amplía las capacidades del SQL estándar, proporcionando herramientas avanzadas para el control de flujo, el manejo de errores y la automatización de tareas. Las transacciones aseguran la consistencia de los datos, mientras que los procedimientos almacenados y los triggers permiten optimizar el rendimiento y la seguridad. El conocimiento de estos elementos es fundamental para gestionar eficientemente bases de datos y desarrollar aplicaciones robustas en entornos empresariales. T-SQL, con su rica funcionalidad, es una herramienta esencial para cualquier profesional que trabaje con bases de datos en SQL Server.