UNIVERSIDAD PERUANA LOS ANDES

FACULTAD DE INGIENERIA

ESCUELA PROFESIONAL DE INGIENERIA DE SISTEMAS Y COMPUTACION

DESARROLLO DE LIBRO
MODULO 6

DOCENTE:
MG. ING. RAÚL FERNÁNDEZ BEJARANO

NOMBRE:
VARGAS SEDANO BRAYAN YEFERSON

CLASE:
BASE DE DATOS II

CICLO:
V

DICIEMBRE - 2024

SQL (LENGUAJE DE CONSULTA ESTRUCTURADO)

1. Introducción a Transacciones

Las transacciones permiten agrupar operaciones para asegurar la integridad de los datos mediante los principios ACID (Atomicidad, Consistencia, Aislamiento, Durabilidad).
Comandos principales:

- BEGIN TRANSACTION: Inicia una nueva transacción.
- COMMIT: Guarda los cambios.
- ROLLBACK: Revierte los cambios en caso de error.

Ejemplo práctico:

```
-- Transacción para transferir fondos entre cuentas
BEGIN TRANSACTION;
-- Debitar de la cuenta origen
UPDATE Cuentas
SET Saldo = Saldo - 100
WHERE CuentaID = 1;
-- Acreditar a la cuenta destino
UPDATE Cuentas
SET Saldo = Saldo + 100
WHERE CuentaID = 2;
-- Validación de errores
IF @@ERROR = 0
    COMMIT;  -- Confirmar cambios si no hay errores
ELSE
    ROLLBACK;  -- Revertir cambios en caso de error
```

2. Estructuras de Control

Transact-SQL incluye estructuras lógicas para controlar el flujo de ejecución:

- IF...ELSE: Realiza decisiones condicionales.
- WHILE: Crea bucles con una condición.
- GOTO: Permite saltar a una etiqueta específica.

Ejemplo práctico:

```
-- Estructura IF...ELSE
DECLARE @Saldo DECIMAL(10, 2) = 500;
IF @Saldo >= 1000
    PRINT 'Saldo suficiente para inversiones.';
ELSE
    PRINT 'Saldo insuficiente para inversiones.';
-- Bucle WHILE
DECLARE @Contador INT = 1;

WHILE @Contador <= 5
BEGIN
    PRINT 'Iteración: ' + CAST(@Contador AS NVARCHAR(10));
    SET @Contador = @Contador + 1;
END;
-- GOTO para salto a etiqueta
IF @Saldo < 1000
    GOTO Fin;
PRINT 'Procesando transacciones...';
Fin:
PRINT 'Fin del script.';
```

3. Funciones Almacenadas y Cursores

- Procedimientos almacenados: Permiten encapsular lógica reutilizable en scripts.
- Cursores: Se usan para recorrer filas de datos de manera secuencial.

3.1 Procedimientos almacenados

Ejemplo práctico:

```
-- Crear procedimiento almacenado
CREATE PROCEDURE ObtenerSaldo
    @CuentaID INT
AS
BEGIN
    SELECT Saldo
    FROM Cuentas
    WHERE CuentaID = @CuentaID;
END;

-- Ejecutar procedimiento
EXEC ObtenerSaldo @CuentaID = 1;
```

3.2 Uso de Cursores

Ejemplo práctico:

```
-- Usar un cursor para actualizar saldos
DECLARE CursorSaldos CURSOR FOR
SELECT CuentaID, Saldo
FROM Cuentas;

DECLARE @CuentaID INT;
DECLARE @Saldo DECIMAL(10, 2);

OPEN CursorSaldos;
FETCH NEXT FROM CursorSaldos INTO @CuentaID, @Saldo;

WHILE @@FETCH_STATUS = 0
BEGIN
    -- Incrementar saldo en 10%
    UPDATE Cuentas
    SET Saldo = @Saldo * 1.1
    WHERE CuentaID = @CuentaID;

    FETCH NEXT FROM CursorSaldos INTO @CuentaID, @Saldo;
END;

CLOSE CursorSaldos;
DEALLOCATE CursorSaldos;
```