In **MS SQL**, you use loops to repeat a set of actions until a condition is met. The most common loop is a **WHILE** loop.

Syntax:

```
WHILE (condition)
BEGIN
   -- Actions to perform
END
```

Loops (WHILE Loop):

- **Purpose:** A loop, like a `WHILE` loop, is used to **repeatedly execute** a set of actions until a condition is met.
- **Usage:** Loops are useful when you need to perform repetitive tasks, such as updating records or running a set of queries multiple times.

Vikas_8460716214

Example:

```
DECLARE @Counter INT = 1;
WHILE @Counter <= 5
BEGIN
   PRINT @Counter;  -- Print the current value
   SET @Counter = @Counter + 1;  -- Increment the counter
END
```

The loop runs 5 times, printing numbers from 1 to 5.

LOOPS are for repeating actions multiple times until a condition changes.

Using a **loop with a CASE statement** in SQL is often used for processing conditional logic in iterations. Here's a **simple example**:

Task:

Assign grades (A, B, C) based on scores in a temporary table using a loop.

Code Example:

```
-- Create a temporary table with sample scores
CREATE TABLE #StudentScores (
   StudentID INT,
   Score INT,
   Grade CHAR(1) NULL
);
-----------------------------------------
-- Insert sample data
INSERT INTO #StudentScores (StudentID, Score)
VALUES (1, 85), (2, 70), (3, 55), (4, 92), (5, 40);
-------------------------------------------------------------------
-- Declare a variable to loop through the table
DECLARE @StudentID INT = 1;

-- Loop through each student and assign a grade
WHILE EXISTS (SELECT 1 FROM #StudentScores WHERE Grade IS NULL) -- Run until all grades are assigned
BEGIN
   -- Update the grade based on the score using CASE
   UPDATE #StudentScores
   SET Grade = CASE
               WHEN Score >= 80 THEN 'A'
               WHEN Score >= 60 THEN 'B'
               ELSE 'C'
            END
   WHERE StudentID = @StudentID;

   -- Move to the next student
   SET @StudentID = @StudentID + 1;
END
```

Vikas_8460716214

```
-- Check the updated data

SELECT * FROM #StudentScores;


-- Drop the temporary table

DROP TABLE #StudentScores;
```

Explanation:

1. A temporary table `#StudentScores` is created with student IDs and scores.
2. A **WHILE** loop checks if there are rows where the `Grade` column is still `NULL`.
3. Inside the loop, the **CASE** statement assigns grades based on conditions:
    - `A` for scores ≥ 80.
    - `B` for scores ≥ 60 but < 80.
    - `C` for scores < 60.
4. The loop increments `@StudentID` to process the next student until all rows have grades.

Vikas_8460716214