# ***Task 5: CASE Statements for Conditional Transformation***

- ## Objective

Use SQL CASE statements to transform and categorize data based on specified conditions.

- ## Dataset Setup

Table Creation

```
CREATE TABLE StudentScores (
StudentID INT,
TotalScore INT,
MathScore INT,
ScienceScore INT
);
```

Insert Sample Data

```
(1, 95, 45, 50),
(2, 85, 35, 60),
(3, 75, 40, 30),
(4, 65, 25, 20);
INSERT INTO StudentScores (StudentID, TotalScore, MathScore, ScienceScore) VALUES
```

- ## Tasks to Perform

### *1. Assign Grades Based on Total Scores*

```
SELECT
StudentID,
TotalScore,
CASE
WHEN TotalScore >= 90 THEN 'A'
WHEN TotalScore >= 80 THEN 'B'
WHEN TotalScore >= 70 THEN 'C'
ELSE 'D (Fail)'
END AS Grade
FROM StudentScores
ORDER BY StudentID;
```

- **Expected Output**

StudentID TotalScore Grade
1 95 A
2 85 B
3 75 C
4 65 D (Fail)

### *2. Identify Pass/Fail Status in Each Subject*

```
SELECT
StudentID,
MathScore,
CASE
WHEN MathScore IS NULL THEN 'No Score'
WHEN MathScore >= 40 THEN 'Pass'
ELSE 'Fail'
END AS MathStatus,
ScienceScore,
CASE
WHEN ScienceScore IS NULL THEN 'No Score'
WHEN ScienceScore >= 40 THEN 'Pass'ELSE 'Fail'
END AS ScienceStatus
FROM StudentScores
ORDER BY StudentID;
```

- **Expected Output**

StudentID MathScore MathStatus ScienceScore ScienceStatus
1 45 Pass 50 Pass
2 35 Fail 60 Pass
3 40 Pass 30 Fail
4 25 Fail 20 Fail

## • Validation

- Grades align with total score ranges.
- Pass/Fail correctly reflects the 40-point threshold.

## • Deliverables

- Task5.sql → SQL code
- Task5_output.txt → query outputs
- Task5_grades.csv → exported grades
- Task5_passfail.csv → exported pass/fail statuses
- README.md → documentation (this file)