CU

CHANDIGARH
UNIVERSITY

**NAME:** Manan Verma **SEMESTER:** 4th

**SECTION:** 24_AIT_KRG-G1 **UID:** 24BAI70008

**SUBJECT CODE:** 24CSH-298

**FACULTY'S NAME:** Mr .SHALABH BHATIA

# Experiment 4

## AIM:

To design and implement PL/SQL programs utilizing conditional control statements such as IF–ELSE, ELSIF, ELSIF ladder, and CASE constructs in order to control the flow of execution based on logical conditions and to analyze decision-making capabilities in PL/SQL blocks.

## Software Requirements:

- **Database Management System:**
  - PostgreSQL
- **Database Administration Tool:**
  - pgAdmin

### Objective:

Implement control structures in PL/SQL (IF-ELSE, ELSE-IF, ELSE-IF LADDER, CASE STATEMENTS in PL-SQL BLOCK)

## Problem Statement

Develop and execute PL/SQL programs that demonstrate the use of **conditional control statements**. The programs should employ **IF–ELSE**, **ELSIF**, **ELSIF ladder**, and **CASE** statements to evaluate given conditions and control the flow of execution accordingly, thereby illustrating decision-making capabilities in PL/SQL blocks.

## Table Creation:

```
CREATE TABLE student_lab (
student_id SERIAL PRIMARY KEY,
student_name VARCHAR(50),
marks INTEGER,
day_no INTEGER
);

INSERT INTO student_lab (student_name, marks, day_no) VALUES
('Manan', 95, 1),
('Aarav', 78, 3),
('Simran', 62, 5),
('Kunal', 45, 7),
('Neha', -5, 9);
```

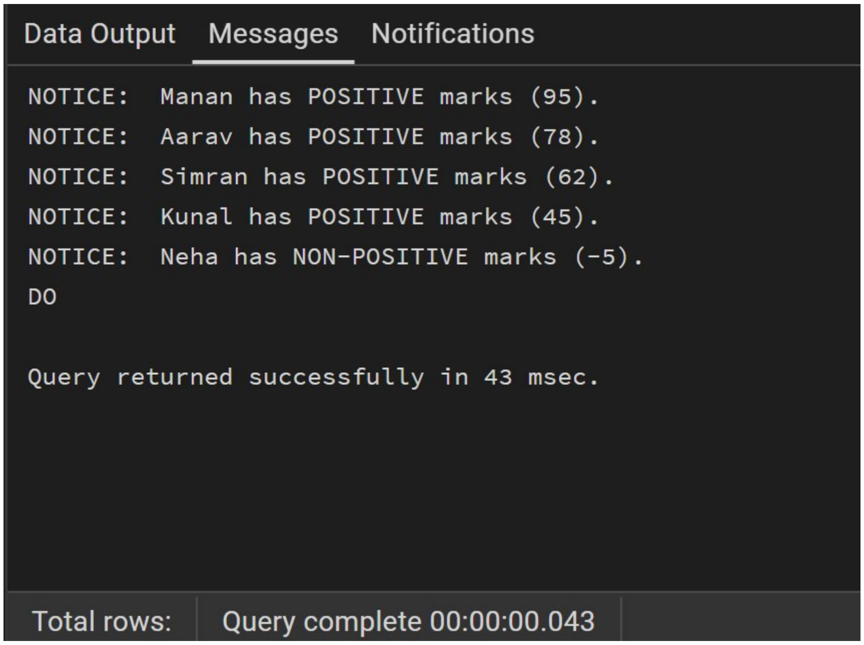
## 1. Problem Statement – IF–ELSE Statement

Write a PL/SQL program to check whether a given number is positive or non-positive using the **IF–ELSE** conditional control statement and display an appropriate message.

## Code:

```
DO $$
DECLARE
   r RECORD;
BEGIN
   FOR r IN SELECT * FROM student_lab LOOP
      IF r.marks > 0 THEN
         RAISE NOTICE '% has POSITIVE marks (%).', r.student_name, r.marks;
      ELSE
         RAISE NOTICE '% has NON-POSITIVE marks (%).', r.student_name, r.marks;
      END IF;
   END LOOP;
END $$;
```

## Output:



```
NOTICE:  Manan has POSITIVE marks (95).
NOTICE:  Aarav has POSITIVE marks (78).
NOTICE:  Simran has POSITIVE marks (62).
NOTICE:  Kunal has POSITIVE marks (45).
NOTICE:  Neha has NON-POSITIVE marks (-5).
DO

Query returned successfully in 43 msec.
```

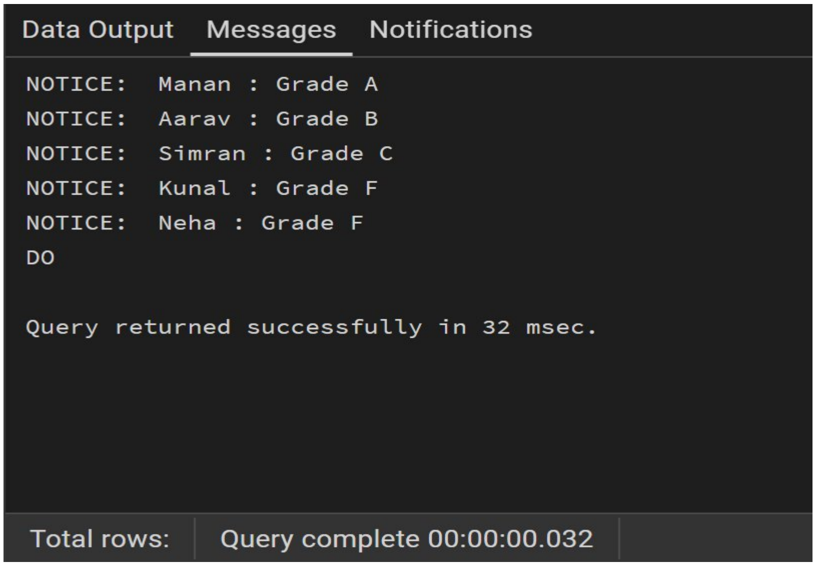
## 2. Problem Statement – IF–ELSIF–ELSE Statement

Write a PL/SQL program to evaluate the grade of a student based on the obtained marks using the **IF–ELSIF–ELSE** statement and display the corresponding grade.

## **Code**:

```
DO $$
DECLARE
   r RECORD;
BEGIN
   FOR r IN SELECT * FROM student_lab LOOP
      IF r.marks >= 90 THEN
         RAISE NOTICE '% : Grade A', r.student_name;
      ELSIF r.marks >= 75 THEN
         RAISE NOTICE '% : Grade B', r.student_name;
      ELSIF r.marks >= 60 THEN
         RAISE NOTICE '% : Grade C', r.student_name;
      ELSE
         RAISE NOTICE '% : Grade F', r.student_name;
      END IF;
   END LOOP;
END $$;
```

## **Output**:



```
NOTICE:  Manan : Grade A
NOTICE:  Aarav : Grade B
NOTICE:  Simran : Grade C
NOTICE:  Kunal : Grade F
NOTICE:  Neha : Grade F
DO

Query returned successfully in 32 msec.
```

Total rows: Query complete 00:00:00.032

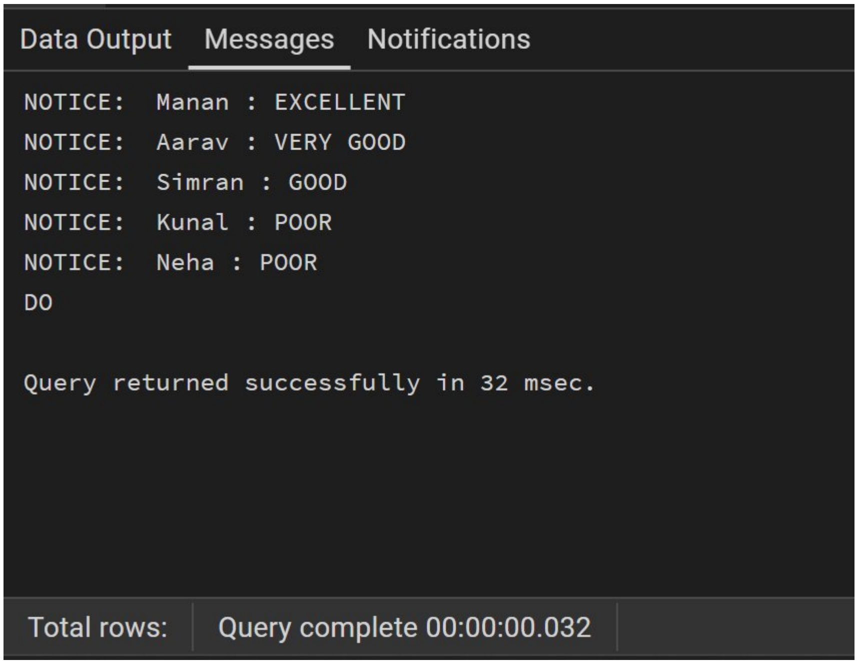
## 3. Problem Statement – ELSIF Ladder

Write a PL/SQL program to determine the performance status of a student based on marks using an **ELSIF ladder** and display the appropriate result.

## Code:

```
DO $$
DECLARE
   r RECORD;
BEGIN
   FOR r IN SELECT * FROM student_lab LOOP
      IF r.marks >= 90 THEN
         RAISE NOTICE '% : EXCELLENT', r.student_name;
      ELSIF r.marks >= 75 THEN
         RAISE NOTICE '% : VERY GOOD', r.student_name;
      ELSIF r.marks >= 60 THEN
         RAISE NOTICE '% : GOOD', r.student_name;
      ELSIF r.marks >= 50 THEN
         RAISE NOTICE '% : AVERAGE', r.student_name;
      ELSE
         RAISE NOTICE '% : POOR', r.student_name;
      END IF;
   END LOOP;
END $$;
```

## Output:



```
NOTICE:  Manan : EXCELLENT
NOTICE:  Aarav : VERY GOOD
NOTICE:  Simran : GOOD
NOTICE:  Kunal : POOR
NOTICE:  Neha : POOR
DO

Query returned successfully in 32 msec.
```

# 4. Problem Statement – CASE Statement

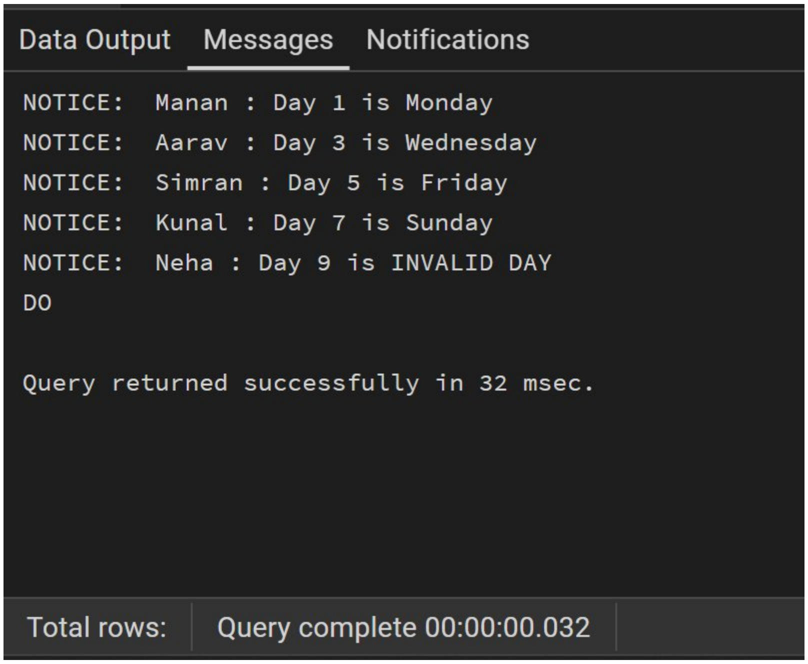
## Code:

```
DO $$
DECLARE
   r RECORD;
   day_name TEXT;
BEGIN
   FOR r IN SELECT * FROM student_lab LOOP
      day_name := CASE r.day_no
         WHEN 1 THEN 'Monday'
         WHEN 2 THEN 'Tuesday'
         WHEN 3 THEN 'Wednesday'
         WHEN 4 THEN 'Thursday'
         WHEN 5 THEN 'Friday'
         WHEN 6 THEN 'Saturday'
         WHEN 7 THEN 'Sunday'
```

```
            ELSE 'INVALID DAY'
          END;

          RAISE NOTICE '% : Day % is %', r.student_name, r.day_no, day_name;
        END LOOP;
      END $$;
```

## Output:



```
NOTICE:  Manan : Day 1 is Monday
NOTICE:  Aarav : Day 3 is Wednesday
NOTICE:  Simran : Day 5 is Friday
NOTICE:  Kunal : Day 7 is Sunday
NOTICE:  Neha : Day 9 is INVALID DAY
DO

Query returned successfully in 32 msec.
```

## Learning Outcomes:

After completing this experiment, the student is able to:
- Understand conditional control statements in PL/pgSQL.
- Implement IF–ELSE and IF–ELSIF–ELSE statements.
- Apply ELSIF ladder for evaluating multiple conditions.
- Use CASE statements for selection-based execution.
- Control the flow of execution in PL/pgSQL blocks.
- Execute and analyze procedural SQL programs in PostgreSQL.