**EXPERIMENT – 10**

**AIM :**

Demonstrate the concept of Control Structures.

**THEORY :**

In this experiment we will learn about Control Structures. The control flow function evaluates the condition specified in it. The output generated by them can be a true, false, static value or column expression. We can use the control flow functions in the SELECT, WHERE, ORDER BY, and GROUP BY clause.

There are three types of Control Structures:

* **SELECTION –** The selection structure tests a condition, then executes one sequence of statements instead of another, depending on whether the condition is true or false. A condition is any variable or expression that returns a BOOLEAN value (TRUE or FALSE).
* **SEQUENCE –** The sequence-structure simply executes a sequence of statements in the order in which they occur.
* **ITERATION –** The iteration structure executes a sequence of statements repeatedly as long as a condition holds true.

**CONDITIONAL CONTROL: IF AND CASE STATEMENTS**

Often, it is necessary to take alternative actions depending on circumstances. The IF statement lets you execute a sequence of statements conditionally. That is, whether the sequence is executed or not depends on the value of a condition. There are three forms of IF statements: IF-THEN, IF-THEN-ELSE, and IF-THEN-ELSIF. The CASE statement is a compact way to evaluate a single condition and choose between many alternative actions.

**IF-THEN STATEMENT:**

IF condition THEN

sequence\_of\_statements

END IF;

**IF-THEN-ELSE STATEMENT:**

IF condition THEN

sequence\_of\_statements1

ELSE

sequence\_of\_statements2

END IF;

**IF-THEN-ELSIF STATEMENT:**

IF condition1 THEN

sequence\_of\_statements1

ELSIF condition2 THEN

sequence\_of\_statements2

ELSE

sequence\_of\_statements3

END IF;

**CASE STATEMENT:**

CASE grade

WHEN 'A' THEN dbms\_output.put\_line('Excellent');

WHEN 'B' THEN dbms\_output.put\_line('Very Good');

WHEN 'C' THEN dbms\_output.put\_line('Good');

WHEN 'D' THEN dbms\_output.put\_line('Fair');

WHEN 'F' THEN dbms\_output.put\_line('Poor');

ELSE dbms\_output.put\_line('No such grade');

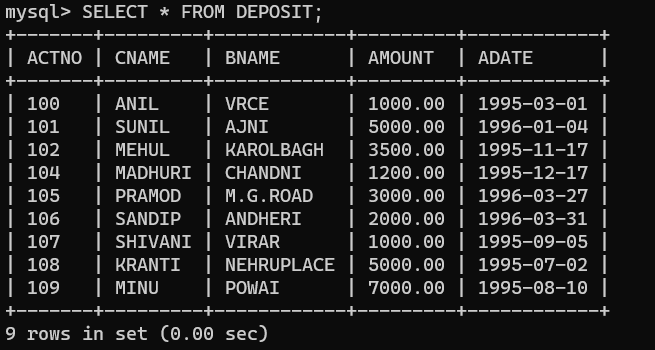
END CASE;

The CASE statement is more readable and more efficient. So, when possible, rewrite lengthy IF-THEN-ELSIF statements as CASE statements.

**PROCEDURE :**

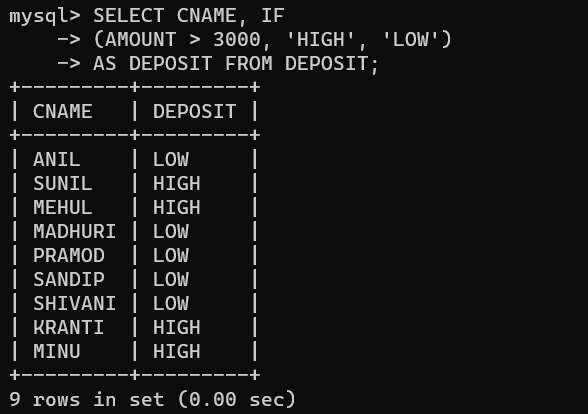
**GIVEN TABLE:**

**DEPOSIT:**

****

**QUERY :** Show name of cutomers and their deposit, if it is higher than 3000 then return high else return low.

**OUTPUT:**

****