TOPS TECHNOLOGY

Module 4 – Introduction to DBMS

Presented By :

Nandni Vala

PL/SQL Control Structures

1. What are control structures in PL/SQL? Explain the IF-THEN and LOOP controlstructures.

- control structures in PL/SQL allow the flow of execution to be directed based on certain conditions or repeated for a set number of iterations. The primary control structures are:
- **Conditional Control**: IF-THEN, IF-THEN-ELSE, CASE
- **Iterative Control**: LOOP, FOR LOOP, WHILE LOOP
- **Transfer Control**: GOTO, EXIT, RETURN, CONTINUE
- **IF-THEN Control Structure**
- The IF-THEN statement is used to execute a block of code conditionally. If the condition evaluates to TRUE, the statements inside the block are executed.

- **Syntax**:
- IF condition THEN
- -- statements to execute if condition is TRUE
- END IF;
- **Example**:
- DECLARE
- salary NUMBER := 5000;
- BEGIN
- IF salary > 4000 THEN
- DBMS_OUTPUT.PUT_LINE('Salary is above threshold.');
- END IF;
- END;
- **IF-THEN-ELSE Control Structure**
- The IF-THEN-ELSE statement allows two possible execution paths: one if the condition is true and

- **Syntax**:
- IF condition THEN
- -- statements to execute if condition is TRUE
- ELSE
- -- statements to execute if condition is FALSE
- END IF;
- **Example**:
- DECLARE
- age NUMBER := 20;
- BEGIN
- IF age >= 18 THEN
- DBMS_OUTPUT.PUT_LINE('Eligible to vote.');
- ELSE
- DBMS_OUTPUT.PUT_LINE('Not eligible to vote.');

- **LOOP Control Structure**
- The LOOP is an iterative control structure used to repeatedly execute a block of code as long as a condition is true.
- **Syntax**:
- LOOP
- -- statements to execute repeatedly
- EXIT WHEN condition;
- END LOOP;
- **Example**:
- DECLARE
- counter NUMBER := 1;
- BEGIN
- LOOP
- DBMS_OUTPUT.PUT_LINE('Counter: ' || counter);
- counter := counter + 1;
- EXIT WHEN counter > 5; -- Exit when counter exceeds 5
- END LOOP;
- END;

2. How do control structures in PL/SQL help in writing complex queries?

- Control structures in PL/SQL are essential for writing complex queries and scripts because they allow developers to implement conditional logic, loops, and exception handling.
- **Conditional Statements (IF-THEN-ELSE)**:
- Conditional statements allow you to execute specific parts of the code based on whether a condition is true or false.
- They are useful in queries where you need to make decisions about data manipulation or to control the flow of logic.
- Example:
- IF salary > 5000 THEN
- DBMS_OUTPUT.PUT_LINE('High salary');
- ELSE
- DBMS_OUTPUT.PUT_LINE('Low salary');
- END IF;

- **Loops (FOR, WHILE, and LOOP)**:
- Loops are used to iterate over a set of records or repeat a block of code multiple times.
- This is helpful for queries that need to process multiple rows, aggregate results, or perform operations on each row in a cursor or collection.
- Example (Using a FOR loop):
- FOR i IN 1..10 LOOP
- DBMS_OUTPUT.PUT_LINE('Number: ' || i);
- END LOOP;