**Sushmit Partakke**
**23070521156**

**Write a BASIC LOOP to print numbers from 1 to 10.**

```
SET SERVEROUTPUT ON;
DECLARE
  i     NUMBER := 1;
BEGIN
  LOOP
    DBMS_OUTPUT.PUT_LINE('NUMBER: ' ||  i);
    i := i + 1;
    EXIT WHEN i > 10;
  END LOOP;
END;
/
```

```
NUMBER: 1
NUMBER: 2
NUMBER: 3
NUMBER: 4
NUMBER: 5
NUMBER: 6
NUMBER: 7
NUMBER: 8
NUMBER: 9
NUMBER: 10
```

**Modify the WHILE LOOP to print even numbers from 2 to 10.**

```
SET SERVEROUTPUT ON;
DECLARE
  i     NUMBER := 2;
BEGIN
  LOOP
    DBMS_OUTPUT.PUT_LINE('EVEN NUMBER: ' ||  i);
    i := i + 2;
    EXIT WHEN i > 10;
  END LOOP;
END;
/
```

Output:

```
EVEN NUMBER: 2
EVEN NUMBER: 4
EVEN NUMBER: 6
EVEN NUMBER: 8
EVEN NUMBER: 10
```

**Write a FOR LOOP to print the square of numbers from 1 to 5.**

```
SET SERVEROUTPUT ON;
DECLARE
  i NUMBER ;
BEGIN
  FOR i in 1..5 LOOP
    DBMS_OUTPUT.PUT_LINE('SQUARE NUMBER:  ' || i*i);
    END LOOP;
END;
/
```

Output:

```
SQUARE NUMBER:  1
SQUARE NUMBER:  4
SQUARE NUMBER:  9
SQUARE NUMBER:  16
SQUARE NUMBER:  25
```

**Create a REVERSE FOR LOOP that prints numbers from 10 to 1.**

```
SET SERVEROUTPUT ON;
DECLARE
  i NUMBER ;
BEGIN
  FOR i in REVERSE 1..10 LOOP
    DBMS_OUTPUT.PUT_LINE('NUMBER:  ' || i);
    END LOOP;
END;
/
```

Output:

```
NUMBER:  10
NUMBER:  9
NUMBER:  8
NUMBER:  7
NUMBER:  6
NUMBER:  5
NUMBER:  4
NUMBER:  3
NUMBER:  2
NUMBER:  1
```

**Write a loop that calculates the sum of numbers from 1 to 5.**

```
SET SERVEROUTPUT ON;
DECLARE
 i    NUMBER := 1;
BEGIN
 LOOP
  i := i + 1;
  EXIT WHEN i > 5;
 END LOOP;
 DBMS_OUTPUT.PUT_LINE('NUMBER: ' ||  i);
END;
/
```

```
SET SERVEROUTPUT ON;
DECLARE
  i     NUMBER := 1;
BEGIN
  LOOP
    i := i + 1;
    EXIT WHEN i > 5;
  END LOOP;
  DBMS_OUTPUT.PUT_LINE('NUMBER: ' ||  i);
END;
/
```

## 1. Write a LOOP to insert 5 new departments into a departments table.

```
DECLARE
   i NUMBER := 1;
   dept_id NUMBER := 60; -- Starting department ID
BEGIN
   LOOP
      INSERT INTO departments (department_id, department_name, location)
      VALUES (dept_id, 'Department ' || i, 'Location ' || i);

      dept_id := dept_id + 10;
      i := i + 1;

      EXIT WHEN i > 5;
   END LOOP;
   COMMIT;
   DBMS_OUTPUT.PUT_LINE('5 new departments inserted.');
END;
/
```

## 2.Modify the WHILE LOOP to increase salaries until they reach 10,000

```
DECLARE
   CURSOR emp_cursor IS
      SELECT employee_id, salary
      FROM employees
      WHERE salary < 10000;
```

```
    emp_rec emp_cursor%ROWTYPE;
BEGIN
    OPEN emp_cursor;
    LOOP
        FETCH emp_cursor INTO emp_rec;
        EXIT WHEN emp_cursor%NOTFOUND;

        WHILE emp_rec.salary < 10000 LOOP
            emp_rec.salary := emp_rec.salary * 1.1;  -- Increase by 10%
            UPDATE employees SET salary = emp_rec.salary WHERE employee_id = emp_rec.employee_id;
        END LOOP;
    END LOOP;
    CLOSE emp_cursor;
    COMMIT;
    DBMS_OUTPUT.PUT_LINE('Salaries updated.');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
        ROLLBACK;
END;
/
```

## 3. Write a FOR LOOP to display employee details for IDs 1 to 5.

```
BEGIN
    FOR i IN 1..5 LOOP
        DECLARE
            emp_rec employees%ROWTYPE;
        BEGIN
            SELECT * INTO emp_rec FROM employees WHERE employee_id = i;
            DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp_rec.employee_id || ', Name: ' ||
emp_rec.first_name || ' ' || emp_rec.last_name || ', Salary: ' || emp_rec.salary);
        EXCEPTION
            WHEN NO_DATA_FOUND THEN
                DBMS_OUTPUT.PUT_LINE('Employee with ID ' || i || ' not found.');
        END;
    END LOOP;
END;
/
```

## 4. Create a cursor-based LOOP that prints employee names and salaries.

```
DECLARE
    CURSOR emp_cursor IS
        SELECT first_name, last_name, salary
        FROM employees;
    emp_rec emp_cursor%ROWTYPE;
BEGIN
    OPEN emp_cursor;
    LOOP
        FETCH emp_cursor INTO emp_rec;
        EXIT WHEN emp_cursor%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Name: ' || emp_rec.first_name || ' ' || emp_rec.last_name || ', Salary: ' ||
emp_rec.salary);
```

```
    END LOOP;
    CLOSE emp_cursor;
END;
/
```

**5. Write a loop that calculates the total salary of all employees.**

```
DECLARE
    total_salary NUMBER := 0;
    CURSOR emp_cursor IS
        SELECT salary FROM employees;
    emp_salary NUMBER;
BEGIN
    OPEN emp_cursor;
    LOOP
        FETCH emp_cursor INTO emp_salary;
        EXIT WHEN emp_cursor%NOTFOUND;
        total_salary := total_salary + emp_salary;
    END LOOP;
    CLOSE emp_cursor;
    DBMS_OUTPUT.PUT_LINE('Total Salary of all employees: ' || total_salary);
END;
/
```

```
5 new departments inserted.
Salaries updated.
Employee ID: 1, Name: John Doe, Salary: 10629.37
Employee ID: 2, Name: Jane Smith, Salary: 10610.77
Employee ID: 3, Name: Robert Jones, Salary: 10356.83
Employee ID: 4, Name: Michael Brown, Salary: 10717.95
Employee ID: 5, Name: Linda Davis, Salary: 10042.97
Name: John Doe, Salary: 10629.37
Name: Jane Smith, Salary: 10610.77
Name: Robert Jones, Salary: 10356.83
Name: Michael Brown, Salary: 10717.95
Name: Linda Davis, Salary: 10042.97
Name: David Wilson, Salary: 10248.7
Total Salary of all employees: 62606.59
```