EXPERIMENT 7

**AIM**- To create PL/SQL programs to implement various types of control structure.

**FACILITIES REQUIRED**

|  |  |  |
| --- | --- | --- |
| **Serial No.** | **Facilities required** | **Quantity** |
| 1 | System | 1 |
| 2 | Operating System | Windows |
| 3 | Front End |  |
| 4 | Backend | Oracle Apex |

**PL/SQL Syntax:** PL/SQL can also process data using flow of statements. The flow of control statements are classified into the following categories.

• Conditional control –Branching

• Iterative control – looping

• Sequential control – Selection

**BRANCHING in PL/SQL:** Sequence of statements can be executed on satisfying certain condition. If statements are being used and different forms of if are:

1. Simple IF                2. If then else               3. Else if                     4. Nested if

**SELECTION IN PL/SQL (Sequential Controls)**

1. Simple case            2. Searched case

**ITERATIONS IN PL/SQL** :Sequence of statements can be executed any number of times using loop construct. It is broadly classified into:

1.Simple Loop              2. For Loop                  3. While Loop

**SIMPLE IF:**

**Syntax:**

IF condition THEN

statement1;

statement2;

END IF;

**IF-THEN-ELSE STATEMENT:**

**Syntax:**

IF condition THEN

 statement1;

ELSE

statement2;

END IF;

**ELSIF STATEMENTS**:

**Syntax:**

 IF condition1 THEN

statement1;

ELSIF condition2 THEN

statement2;

ELSIF condition3 THEN

statement3;

ELSE

statement;

END IF;

**NESTED IF:**

**Syntax:**

IF condition THEN

statement1;

ELSE

IF condition THEN

statement2;

ELSE

Statement3;

 END IF;

 END IF;

 ELSE

statement3;

END IF;

**SELECTION IN PL/SQL (Sequential Controls**)

**SIMPLE CASE**

**Syntax:**

CASE SELECTOR

WHEN Expr1 THEN statement1;

 WHEN Expr2 THEN statement2;

:

 ELSE

Statement n;

END CASE;

**SEARCHED CASE:**

**Syntax:**

CASE

WHEN searchcondition1 THEN statement1;

WHEN searchcondition2 THEN statement2;

 ::

ELSE

statement;

END CASE;

**ITERATIONS IN PL/SQL**

**SIMPLE LOOP**

**Syntax:**

LOOP

statement1;

EXIT [ WHEN Condition];

 END LOOP;

**Example:**

Declare

 A number:=10;

Begin

Loop a := a+25;

exit when a=250;

end loop;

dbms\_output.put\_line(to\_char(a));

end;

**WHILE LOOP**

**Syntax**

WHILE condition LOOP

statement1;

statement2;

END LOOP;

**Example:**

Declare

i number:=0;

 j number:=0;

begin

while i<=100 Loop

 j := j+i;

 i := i+2;

end loop;

dbms\_output.put\_line(‘the value of j is’ ||j);

end;/

**FOR LOOP**

**Syntax:**

FOR counter IN [REVERSE]

LowerBound..UpperBound

LOOP

statement1;

statement2;

END LOOP;

**Example:**

Begin

For I in 1..2

Loop

Update emp set field = value where condition;

End loop;

End;/

**Queries**

**Q1: write a pl/sql program to swap two numbers**

**Ans:** SQL> DECLARE

    a NUMBER(10) := :P1\_A;

    b NUMBER(10) := :P1\_B;

    c NUMBER(10);

BEGIN

    DBMS\_OUTPUT.PUT\_LINE('THE PREV VALUES OF A AND B WERE:');

    DBMS\_OUTPUT.PUT\_LINE('A = ' || a);

    DBMS\_OUTPUT.PUT\_LINE('B = ' || b);

c := a;

a := b;

b := c;

    DBMS\_OUTPUT.PUT\_LINE('THE VALUES OF A AND B AFTER SWAPPING:');

    DBMS\_OUTPUT.PUT\_LINE('A = ' || a);

    DBMS\_OUTPUT.PUT\_LINE('B = ' || b);

END;

**Q2: Write a pl/sql program to find the largest of three numbers**

**Ans:**SQL> DECLARE

a NUMBER := :P1\_Number1;

b NUMBER := :P2\_Number2;

c NUMBER := :P3\_Number3;

BEGIN

IF a > b THEN

IF a > c THEN

DBMS\_OUTPUT.PUT\_LINE('Biggest is: ' || TO\_CHAR(a));

ELSE

DBMS\_OUTPUT.PUT\_LINE('Biggest is: ' || TO\_CHAR(c));

END IF;

ELSIF b > c THEN

DBMS\_OUTPUT.PUT\_LINE('Biggest is: ' || TO\_CHAR(b));

ELSE

DBMS\_OUTPUT.PUT\_LINE('Biggest is: ' || TO\_CHAR(c));

END IF;

END;

/

**Q3: write a pl/sql program to find the total and average of 6 subjects and display the grade**

**Ans:**SQL>DECLARE

    java NUMBER(10) := :P1\_JAVA;

    dbms NUMBER(10) := :P1\_DBMS;

    co NUMBER(10) := :P1\_CO;

    se NUMBER(10) := :P1\_SE;

    es NUMBER(10) := :P1\_ES;

    ppl NUMBER(10) := :P1\_PPL;

    total NUMBER(10);

    per NUMBER(10);

BEGIN

    total := java + dbms + co + se + es + ppl;

    per := (total / 600) \* 100;

    IF java < 50 OR dbms < 50 OR co < 50 OR se < 50 OR es < 50 OR ppl < 50 THEN

        DBMS\_OUTPUT.PUT\_LINE('FAIL');

    ELSE

        IF per > 75 THEN

            DBMS\_OUTPUT.PUT\_LINE('GRADE A');

        ELSIF per > 65 THEN

            DBMS\_OUTPUT.PUT\_LINE('GRADE B');

        ELSIF per > 55 THEN

            DBMS\_OUTPUT.PUT\_LINE('GRADE C');

        ELSE

            DBMS\_OUTPUT.PUT\_LINE('GRADE D');

        END IF;

        DBMS\_OUTPUT.PUT\_LINE('TOTAL IS: ' || total);

        DBMS\_OUTPUT.PUT\_LINE('PERCENTAGE IS: ' || per);

    END IF;

END;

**Q4: Write a pl/sql program to find the sum of digits in a given number**

**Ans:** SQL>DECLARE

  a NUMBER := :P1\_A;

  d NUMBER := 0;

              sum1 NUMBER := 0;

BEGIN

    WHILE a > 0 LOOP

d := MOD(a, 10);

        sum1 := sum1 + d;

a := TRUNC(a / 10);

    END LOOP;

    DBMS\_OUTPUT.PUT\_LINE('Sum of digits is: ' || sum1);

END;

**Q5: write a pl/sql program to display the number in reverse order**

**Ans:**SQL>DECLARE

    a NUMBER := :P1\_A;

    rev NUMBER := 0;

    d NUMBER;

BEGIN

    WHILE a > 0 LOOP

d := MOD(a, 10);

rev := (rev \* 10) + d;

a := TRUNC(a / 10);

    END LOOP;

    DBMS\_OUTPUT.PUT\_LINE('Reversed number is: ' || rev);

END;

**Q6: Write a PL / SQL program to check whether the given number is prime or not**

**Ans:**SQL>DECLARE

    a NUMBER := :P1\_A;

    c NUMBER := 0;

    i NUMBER;

BEGIN

    FOR i IN 1..a LOOP

        IF MOD(a, i) = 0 THEN

c := c + 1;

        END IF;

    END LOOP;

    IF c = 2 THEN

        DBMS\_OUTPUT.PUT\_LINE(a || ' is a prime number');

    ELSE

        DBMS\_OUTPUT.PUT\_LINE(a || ' is not a prime number');

    END IF;

END;

**Q7: Write a PL/SQL program to find the factorial of a given number**

**Ans:**SQL>DECLARE

    n NUMBER :=      :P1\_N;

    f NUMBER := 1;

    i NUMBER;

BEGIN

    FOR i IN 1..n LOOP

f := f \* i;

    END LOOP;

    DBMS\_OUTPUT.PUT\_LINE('The factorial is ' || f);

END;

**Q8: write a pl/sql code block to calculate the area of a circle for a value of radius varying from 3 to 7. Store the radius and the corresponding values of calculated area in an empty table named areas, consisting of two columns radius & area**

**Ans:SQL>**create table AREA(

    Radius number(10),

    Area number(6,2)

)

**SQL>**DECLARE

   pi CONSTANT NUMBER(5,2) := 3.14;

   radius NUMBER(5) := 3;

   area NUMBER(6,2);

BEGIN

   WHILE radius < 7 LOOP

      area := pi \* POWER(radius, 2);

INSERT INTO AREA (Radius, Area) VALUES (radius, area);

DBMS\_OUTPUT.PUT\_LINE('Inserted -> Radius: ' || radius || ', Area: ' || area);

radius := radius + 1;

   END LOOP;

 COMMIT;

   DBMS\_OUTPUT.PUT\_LINE('All records inserted successfully.');

END;

/

**SQL>**select\* from area

**Q9: write a PL/SQL code block that will accept an account number from the user, check if the users balance is less than minimum balance, only then deduct rs.100/- from the balance. This process is fired on the acct table.**

**Ans:SQL>**create table acct(name varchar2(10),cur\_bal number(10),acctno number(6,2));

**SQL>** insert into acct(acctno ,name ,cur\_bal )values(777,'Sirius',10000);

insert into acct(acctno ,name ,cur\_bal )values(765,'John',1000);

 insert into acct(acctno ,name ,cur\_bal )values(855,'Sam',500);

 insert into acct(acctno ,name ,cur\_bal )values(353,'Peter',800);

**SQL>**select\* from acct

**SQL>** DECLARE

mano NUMBER(5) := :P1\_AccountNo;

mcb NUMBER(6,2);

minibal CONSTANT NUMBER(7,2) := 1000.00;

fine CONSTANT NUMBER(6,2) := 100.00;

BEGIN

SELECT cur\_bal INTO mcb FROM acct WHERE acctno = mano;

IF mcb < minibal THEN

-- Deduct fine

UPDATE acct

SET cur\_bal = cur\_bal - fine

WHERE acctno = mano;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Fine of Rs.' || fine || ' deducted for Account No: ' || mano);

DBMS\_OUTPUT.PUT\_LINE('Updated Balance: ' || (mcb - fine));

ELSE

DBMS\_OUTPUT.PUT\_LINE('No fine applied. Balance is sufficient.');

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Account number not found.');

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END;

/

**SQL>**select\* from acct