Task -6 **PL/SQL Procedures, functions, Loops**

Aim:

To implement PL/SQL Procedures, Functions and loops on Number theory and business scenarios.

Procedure: PL/SQL is a combination of SQL along with the procedural features of programming languages.

Syntax:

```
DECLARE
   <declarations section>
BEGIN
   <executable command(s)>
EXCEPTION
   <exception handling>
END;
```

Program:

```
DECLARE
   message  varchar2(20):= 'booking closed';
BEGIN
   dbms_output.put_line(message);
END;
```

Questions:

1. **Write a PL/SQL block that calculates the average age of players and displays the result.**
2. **Write a PL/SQL block that inserts a new player record into the Player table.**
3. **To create a function that returns the total number of matches played by a specific player.**
4. **To write a non-recursive PL/SQL procedure to retrieve even-numbered PlayerIDs registered for any tournament. - Give me queries with answers**

Q1 .PL/SQL block – Average age of players

```
SET SERVEROUTPUT ON;

   DECLARE
       v_avg_age NUMBER(5,2);
   BEGIN
       SELECT AVG(Age)
       INTO v_avg_age
       FROM Player;

        DBMS_OUTPUT.PUT_LINE('Average Age of Players = ' || v_avg_age);
   END;
   /
```

Q2. PL/SQL block – Insert a new player record

```
DECLARE
BEGIN
   INSERT INTO Player (PlayerID, PlayerName, Age)
   VALUES (105, 'Rohit Sharma', 36);
   DBMS_OUTPUT.PUT_LINE('New Player inserted successfully.');
END;
/
```

Q3. Function – Total number of matches by a specific player

```
CREATE OR REPLACE FUNCTION TotalMatchesByPlayer(p_playerID IN
NUMBER)
RETURN NUMBER
IS
   v_total NUMBER;
BEGIN
   SELECT COUNT(*)
   INTO v_total
   FROM Match
   WHERE PlayerID = p_playerID;
   RETURN v_total;
END;
/
```

Q4. Non-recursive Procedure – Retrieve even-numbered PlayerIDs

```
CREATE OR REPLACE PROCEDURE GetEvenPlayerIDs IS
   CURSOR c1 IS
      SELECT PlayerID, PlayerName
      FROM Player
      WHERE MOD(PlayerID,2) = 0;   -- Even IDs
BEGIN
   FOR rec IN c1 LOOP
```

```
        DBMS_OUTPUT.PUT_LINE('PlayerID: ' || rec.PlayerID || '  Name: ' ||
    rec.PlayerName);
      END LOOP;
    END;
    /
```

Q5 -Using IF-THEN statement

```
SQL>set server output on;
SQL>declare
b number;
c number;
begin
 c:=&temp;
 select mark into b from student where s_id=c;
 if b>50 then
dbms_output.put_line('PASS');
 elsif
 dbms_output.put_line('FAIL');
  end if;
 end;
 /
```

Q6: Sample programs- create table 1

```
Sql>SET SERVEROUTPUT ON;
Sql>CREATE TABLE employees (emp_id   NUMBER PRIMARY KEY,   emp_name
VARCHAR2(100),  age     NUMBER,   salary   NUMBER,   doj     DATE
);
```

-- insert sample rows

```
INSERT INTO employees VALUES (1, 'Alice', 30, 50000, DATE '2015-06-01');
INSERT INTO employees VALUES (2, 'Bob',   35, 60000, DATE '2012-03-15');
INSERT INTO employees VALUES (3, 'Carol', 28, 45000, DATE '2019-11-22');
INSERT INTO employees VALUES (4, 'Dave',  42, 80000, DATE '2010-01-05');
```

COMMIT;

Q7) Function — average age

Code 2

```
Sql> CREATE OR REPLACE FUNCTION get_avg_age RETURN NUMBER IS
  v_avg_age NUMBER;
BEGIN
  SELECT AVG(age) INTO v_avg_age FROM employees;
  RETURN ROUND(v_avg_age, 2);
EXCEPTION
  WHEN NO_DATA_FOUND THEN
    RETURN NULL;                                    -- no rows
  WHEN OTHERS THEN
    RAISE;                                          -- propagate unexpected errors
END get_avg_age;
/
```

Call it

```
Sql> DECLARE
  v_avg NUMBER;
BEGIN
  v_avg := get_avg_age;
  DBMS_OUTPUT.PUT_LINE('Average age = ' || NVL(TO_CHAR(v_avg), 'NULL'));
END;
/
```

Expected output

java

Copy code

Average age = 33.75

(Depending on the data; above with sample rows.)

Q8) Simple numeric FOR loop (anonymous block)

```
Sql>
BEGIN
```

```
  FOR i IN 1..5 LOOP
    DBMS_OUTPUT.PUT_LINE('i = ' || i);
  END LOOP;
END;
/
```

Q9) Cursor FOR loop — iterate rows (recommended style)

```
Sql>
CREATE OR REPLACE PROCEDURE list_emp_names IS
BEGIN
  FOR r IN (SELECT emp_id, emp_name, salary FROM employees ORDER BY emp_id)
LOOP
    DBMS_OUTPUT.PUT_LINE(r.emp_id || ' - ' || r.emp_name || ' : ' || r.salary);
  END LOOP;
END list_emp_names;
/
```

Call

```
Sql>
Copy code
BEGIN
  list_emp_names;
END;
/
```