Task 6: Procedures, Functions and Loops

Aim: To write a programming using PL/SQL procedures, functions and loops on Number theory and business scenarios like

1. write a PL/SQL block that calculates the average age of players and displays the result.
2. write PL/SQL block that insert a new player record into the player table.
3. To create a function that returns the total number of teams in a particular cricket Board.
4. To write a non-recursive PL/SQL procedure to retrieve even-numbered players IDs registered for any tournament.

write a PL/SQL block that calculates the average age of players and displays the result.

```
DECLARE
    total_age NUMBER:=0
    num_players NUMBER:=0;
    avg_age NUMBER:=0;
BEGIN
-using a cursor to loop through all players
FOR player_rec IN (SELECT Age FROM players) LOOP
total_age:=total_age+player_rec.Age;-- summing
up the ages
num_players:=Num_players+1;- counting the
number of players
    END LOOP;
-calculating the average age
    IF num_players>0 THEN
    avg_age:= total_age/num_players;
    END IF;
    -- Displaying the result
DBMS_output.put_LINE('Total players:'||num_
                                    players);
DBMS_output.put_LINE('Total Age:'||total_age);
DBMS_output.put_LINE('Average Age:'||
                                    (avg_age));
    END;
```

output:

```
Total players:14
Total Age : 342
Average Age:24
```

Write a PL/SQL block that calculates the average age of players and displays the result.

```
DECLARE
  total_age NUMBER := 0;
  num_players NUMBER := 0;
  avg_age NUMBER := 0;
BEGIN
  -- Using a cursor to loop through all players
  FOR player_record IN (SELECT age FROM players)
  LOOP
    total_age := total_age + player_record.age; -- summing up the ages
    num_players := num_players + 1; -- counting the number of players
  END LOOP;
  -- calculating the average age
  IF num_players > 0 THEN
    avg_age := total_age / num_players;
  END IF;
  -- Displaying the result
  DBMS_OUTPUT.PUT_LINE('Total players: ' || num_players);
```

Write a PL/SQL block that inserts a new player record into the player table.

```
DECLARE
  v_playerID VARCHAR(6) := '&playerid'; -- you can generate a unique playerID as needed
  v_teamID VARCHAR(10) := '&teamID'; -- Replace with the actual teamID
  v_fname VARCHAR(30) := '&fname';
  v_lname VARCHAR(30) := '&lname';
  v_age NUMBER(5,2) := &age;
  v_DateOfBirth DATE := TO_DATE('&DOB', 'DD-MON-YYYY'); -- Replace with the actual Date of Birth
  v_playingRole VARCHAR(25) := '&playingRole';
  v_email VARCHAR(40) := '&email';
  v_contact_no NUMBER := &phone; -- Replace with the actual contact number
  v_batting varchar(10) := '&batting';
  v_bowling varchar(10) := '&bowling';
BEGIN
```

INSERT INTO player (playerID, teamID, FNAME, LNAME, age, Date of Birth, playing Role, email, contact-no, batting, bowling)
VALUES (i-playerID, v-teamID, v-FNAME, v-LNAME, v-age, v-Date of Birth, v-playing Role, v-email, v-contact-no, v-batting, v-bowling)
COMMIT;
DBMS-output.PUT.LINE ('player record inserted sucessfully.');
EXCEPTION
WHEN OTHERS THEN
DBMS-output.PUT-LINE ('Error:' || SQLERRM);
ROLL BACK;
END;
/
Enter the player ID: 676
Enter the team ID: CC01
Enter the FName: Rahul
Enter the LName: sharma
Enter the age: 23
Enter the Date of Brith: 17-07-1999
Enter the playing Role: All Rounder
Enter the email: rahulsharma@gmail.com
Enter the contact-no: 9747181815
Player record inserted successfully.
pl/sql procedure successfully completed

To create a functions that returns the total number of teams in a particular cricket Board.
CREATE OR REPLACE FUNCTION Get total teams Board (Board ID VARCHAR2)
RETURN NUMBER IS
v-total teams NUMBER := 0;
BEGIN
SELECT COUNT (*) INTO v-total teams FROM team WHERE Board ID = BoardID;
RETURN v-total teams;
EXCEPTION

WHEN NO_DATA_FOUND THEN
-- Handle the case when the board doesn't exist or has no teams
RETURN 0;
WHEN OTHERS THEN
-- Handle other exceptions as needed
RETURN -1; -- Return a negative value to indicate an error
END Get total teams on Board;
/
function successfully created.
SQL>
Declare
Res number;
Begin
res := Get total teams On Board('BDDO1');
DMBS_output.put_line('No of teams: '||res);
END;
/
No of teams:2

To write a non-recursive PL/SQL procedure to retrieve registered for any tournament

CREATE OR REPLACE PROCEDURE Get Even Numbered Players IDs IS BEGIN
FOR player_rec IN (SELECT PlayerID FROM Players WHERE
MOD(TO_NUMBER(PlayerID),2) = 0)
LOOP
DBMS_output.put_line('even-numbered Player ID :'||Player_rec.PlayerID);
END LOOP;
END Get Even Numbered players IDs;
/
SQL> EXEC Get Even Numbered players IDs;

Even - Numbered player ID: 102
Even - Numbered player ID: 12
even - number player ID: 18
Even - Number player ID: 2
even - Number player ID: 82
even - Number player ID: 66
Even - Number player ID: 676
PL/SQL procedure successfully completed

VEL TECH - CSE	
EX NO.	
ERFORMANCE (5)	
ULT AND ANALYSIS (3)	
VA VOCE (3)	
TAL (15)	
N WITH DATE	

Result: Thus PL/SQL procedures, functions and loops on Numbers, theory and business scenarios experiment was successfully completed and results are verified.