**Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology**

 **(Deemed to be University Estd. u/s 3 of UGC Act, 1956)**

**School of Computing**

**B.Tech. – Computer Science and Engineering**

**VTR UGE2021- (CBCS)**

Academic Year: 2025–2026

SUMMER SEMESTER - SS2526

Course Code : 10211CS207

Course Name : Database Management Systems

Slot No : S2L3

DBMS TASK - 6 REPORT

Title:Procedures, Function and Loops

Submitted by:

|  |  |  |
| --- | --- | --- |
| **VTUNO** | **REGISTER NUMBER** | **STUDENT NAME** |
| VTU27916 | 24UECS1462 | J.SAI VINEETH KUMAR |

Procedures, Function 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 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 teams in a particular Cricket Board.
4. To write a non-recursive PL/SQL procedure to retrieve even-numbered PlayerIDs 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 Player) 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: 24.42

**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(6) := '&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', 'YYYY-MM-DD'); -- Replace with the actual DateofBirth

v\_PlayingRole VARCHAR(25) := '&PlayingRole';

v\_email VARCHAR(40) := '&email';

v\_contact\_no NUMBER := &phone; -- Replace with the actual contact number

BEGIN

INSERT INTO Player (PlayerID, TeamID, FName, LName, Age, DateofBirth, PlayingRole, email, contact\_no)

VALUES (v\_PlayerID, v\_TeamID, v\_FName, v\_LName, v\_Age, v\_DateofBirth, v\_PlayingRole, v\_email, v\_contact\_no);

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Player record inserted successfully.');

EXCEPTION

WHEN OTHERS THEN

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

ROLLBACK;

END;

/

Enter the PlayerID: 676

Enter the TeamID: CCB01

Enter the FName: Rahul

Enter the LName: Sharma

Enter the Age: 23

Enter the DateofBirth: 17-07-1999

Enter the PlayingRole: AllRounder

Enter the email: rahulsharma@gmail.com

Enter the Contact\_no: 9797181815

Player record inserted successfully.

**To create a function that returns the total number of teams in a particular Cricket Board.**

CREATE OR REPLACE FUNCTION GetTotalTeamsInBoard(BoardID VARCHAR2) RETURN NUMBER IS

v\_TotalTeams NUMBER := 0;

BEGIN

SELECT COUNT(\*) INTO v\_TotalTeams FROM Team WHERE BoardID = BoardID;

RETURN v\_TotalTeams;

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 GetTotalTeamsInBoard;

/

**Function successfully created.**

**SQL>**

Declare

**number res;**

**Begin  
res:=** GetTotalTeamsInBoard(‘BID01’);

DBMS\_OUTPUT.PUT\_LINE(‘No of teams: ‘||res);

END;

/

No of teams: 2

**To write a non-recursive PL/SQL procedure to retrieve even-numbered PlayerIDs registered for any tournament.**

CREATE OR REPLACE PROCEDURE GetEvenNumberedPlayerIDs IS

BEGIN

FOR player\_rec IN ( SELECT PlayerID FROM Player WHERE TO\_NUMBER(PlayerID) MOD 2 = 0)

LOOP

DBMS\_OUTPUT.PUT\_LINE('Even-Numbered PlayerID: ' || player\_rec.PlayerID);

END LOOP;

END GetEvenNumberedPlayerIDs;

/

**Result:**

Thus the PL/SQL Procedures, Functions and loops on Number theory and business scenarios experiment was successfully completed and results are verified.