

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 : S1L4

DBMS TASK - 6 REPORT

Title: Procedures, Functions And Loops

Submitted by:

VTU NO	REGISTER NUMBER	STUDENT NAME
VTU28953	24UECS0493	CH. KISHORE

TASK b: 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: ' || ceil(avg_age));
END;
```

Output:

Total Players: 14

Total Age: 342

Average Age: 24

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 DateofBirth

v_PlayingRole VARCHAR(25) := '&PlayingRole';

v_email VARCHAR(40) := '&email';

v_contact_no NUMBER := ☎ -- 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, DateofBirth, PlayingRole, email, contact_no, batting, bowling)

VALUES (v_PlayerID, v_TeamID, v_FName, v_LName, v_Age, v_DateofBirth, v_PlayingRole, v_email, v_contact_no, v_batting, v_bowling);

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.
PL/SQL procedure successfully completed.

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

```

CREATE OR REPLACE FUNCTION GetTotalTeamsInBoard(BoardID1 VARCHAR2) RETURN
NUMBER IS
    v_TotalTeams NUMBER := 0;
BEGIN
    SELECT COUNT(*) INTO v_TotalTeams FROM Team WHERE BoardID = BoardID1;
    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
```

```
Res number;
```

```
Begin
```

```
res:= GetTotalTeamsInBoard('B1001');
```

```
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 MOD(TO_NUMBER(PlayerID),  
2) = 0)
```

```
        LOOP
```

```
            DBMS_OUTPUT.PUT_LINE('Even-Numbered PlayerID: ' || player_rec.PlayerID);
```

```
        END LOOP;
```

```
END GetEvenNumberedPlayerIDs;
```

```
/
```

```
SQL> EXEC GetEvenNumberedPlayerIDs;
```

```
Even-Numbered PlayerID: 102
```

```
Even-Numbered PlayerID: 12
```

```
Even-Numbered PlayerID: 18
```

```
Even-Numbered PlayerID: 2
```

```
Even-Numbered PlayerID: 62
```

```
Even-Numbered PlayerID: 66
```

```
Even-Numbered PlayerID: 676
```

PL/SQL procedure successfully completed.

Result:

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