

26-8-25
Task 6: procedure, function and loops

Aim: TO write a programming using PL/SQL. Procedures, Functions and loops on Number theory and business scenarios like

1. write PL/SQL block that calculates average age of players and display the result
2. write a 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 particular cricket board.
4. TO write non-recursive PL/SQL procedure retrieves even numbered player ID's registered for tournament.

write 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 * from Player) loop
        total_age := total_age + player_rec.Age; ... summing up ages
        num_players := num_players + 1; ... counting the number of players
    end loop; ... finished
    ... 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 new players record into the player table.

DECLARE

V...player_ID VARCHAR(8) := '&playerid'; -- you can generate unique player ID as needed.

V...TeamID VARCHAR(10) := '&TEAMID'; Replace with actual Team ID

V...FName VARCHAR(30) := '&Fname';

V...LName VARCHAR(30) := '&Lname';

V...Age NUMBER(5,2) := &age;

V...Date of Birth DATE := TO_DATE('&DOB', 'DD-Mon-YYYY');

Replace with actual date of Birth

V...PlayingRole VARCHAR(25) := '&PlayingRole';

V...Email VARCHAR(40) := '&email';

V...Contact_no NUMBER := ☎ -- Replace with 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, PlayingRole, Email, Contact_no, batting, bowling)

VALUES(v...PlayerID, v...TeamID, v...FName, v...LName, v...Age,
v...Date of Birth, v...PlayingRole, v...Email, v...ContactNo,
v...Batting, v...Bowling);

COMMIT;

DBMS-OUTPUT.PUT-LINE('Player record inserted successfully');

EXCEPTION

when others then

DBMS-OUTPUT.PUT-LINE('Error : '||SQLERRM);

ROLL BACK;

END;

/ Enter the player ID: 676

Enter the teamID: CCB01

Enter the FName: Rahul

Enter the LName: Sharma

Enter the Age: 23

Enter the Date of Birth: 17-07-1999

Enter the Playing Role: All Rounder

Enter Email: rahul.sharma@gmail.com

Enter the Contact-No: 9797181815

player record inserted successfully.
PL/SQL procedure successfully completed.

2) Create function returns total number of teams in
a particular Cricket Board.

CREATE OR REPLACE FUNCTION Get_TotalTeamsInBoard(Board
ID IN VARCHAR2) RETURN NUMBER AS

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 board does not raise 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 In Board;

/

PUNCTION successfully created.

SQL>

Declare

Res number;

Begin

res := Get total teams in Board ('B100');

DBMS-OUTPUT.PUT-LINE('no of teams,' || res);

END;

/

NO. of teams:2

To write a non-recursive PL/SQL Procedure retrieve even-numbered Player IDs registered for any tournament.

CREATE OR REPLACE PROCEDURE GetEven Numbered Player IDs IS

BEGIN FOR player_rec IN (SELECT .player_ID FROM player WHERE MOD(TD - NUMBER(player_ID), 2) = 0)

LOOP

DBMS-OUTPUT.PUT-LINE('Even-numbered player ID: ' ||
player_rec.player_ID);

END LOOP;

END GetEvenNumbered Player IDs;

/

SQL EXES Get Even Numbered playerIDs;

Even-numbered player ID: 102
Even-numbered player ID: 12
Even-numbered player ID: 18
Even-numbered player ID: 2
Even-numbered player ID: 62
Even-numbered player ID: 66
Even-numbered player ID: 676

PL/SQL procedure successfully completed.

View: (TIN239.WIN) 11:49 AM 27/2/2023

EX NO.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
PERFORMANCE (5)																				
RESULT AND ANALYSIS (5)																				
VIVA VOCE (5)																				
RECORD IS AT 100% OF TOTAL																				
TOTAL (20)																				
SIGN WITH DATE																				

RESULT: Thus pl/sql procedure, functions and loops on number theory and business scenarios experiments were successfully completed and results are verified.

(Shiva)