

AIM:- To write a programming using SQL procedure
Function sandopen number theory & business scenarios
like

1. Write SQL/SQL block that calculates the average age of players & displays the result.
2. Write SQL/SQL block that inserts a new player record into the player table.
3. To create function that returns the total no. of teams in a particular cricket Board.
4. To write a non-recursive PL/SQL procedure to retrieve even-numbered player ID's registered for any tournament.

Write a SQL block that calculates the average age of players & displays the result

DECLARE

total_age NUMBER = 0;

num_players NUMBER = 0;

avg_age NUMBER = 0;

BEGIN

-- Using a cursor to loop through
FOR Player IN (SELECT Age FROM player)
LOOP
total_age = total_age + Player.Age; --
Summing up all the Ages

Num_players = num_players + 1; --

no. of players

END LOOP;

-- Calculating the average use

IF num_players > 0 THEN

avg_age = total_age / num_players;

END IF;

-- Displaying the result

DBMS_OUTPUT.PUT_LINE('Total Player: ' || num_players);

DBMS_OUTPUT.PUT_LINE('Average Age: ' || avg_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_Player ID VARCHAR(6) = 'Player ID'; --

You can generate a unique Player ID as needed

V_Team ID VARCHAR(6) = 'Team ID'; --

Replace with the 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 }, 'YYYY-MM-DD')

Replace with the actual Date of Birth

V - Playing Role VARCHAR (25) = { playing Role }

{ playing Role }

V - E-mail VARCHAR (40) = { email }

V - Contact - DO NUMBER = { phone }, --

Replace with the actual Contact Number

BEGIN

INSERT INTO PLAYER (Player ID, Team ID, F-Name,

L-Name, Age, Date of Birth, Playing Role, Email, Contact NO)

VALUES (CV - Player ID, V - Team ID, V - F-Name, V - L-Name,

V - Age, V - Date of Birth, V - Playing, V - Email, V - Contact NO);

COMMIT;

DBMS_OUTPUT.PUT_LINE ('Player record is inserted

Successfully :);

EXCEPTION

WHEN errors then

DBMS_OUTPUT.PUT_LINE ('Error: SQL Error');

ROLL BACK;

END;

/

Enter the player ID: 671

Enter the team ID: CC BOB

Enter the fName: aqhu

Enter the VName: sharma

Enter the Age : 23

Enter of D.O.B. 17-07-1999

Enter the playing Role: All Rounder

Enter the contact-no: 9797181815

Player record inserted successfully.

function successfully created

SQL >

Declare

number res;

Begin

res = GET_TOTEL TeamID Board('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 player ID's registered for
any team

CREATE OR REPLACE PROCEDURE

GET_EVEN_NUMBERED_PLAYER_IDS

BEGIN

FOR player_id IN (SELECT player_id FROM
player WHERE (player_id MOD

2 = 0)

LOOP


DBMS_OUTPUT.PUT_LINE ('Even-Numbered ~~Player~~ ID;

player ID: '|| player_id || player_name || player_id;

END LOOP;

END OF Even Numbered player IDs;

/

VEL TECH - CSE	
EX NO. 10	6
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	2
RECORD (5)	5
TOTAL (20)	17
SIGNATURE & DATE	

Result:- Thus the PL/SQL procedures, Function & loops on numbered theory & business scenarios Experimental successfully completed & results got verified