TASK 6: Procedures, Function and Loops

Aim: To write a programming using PL/SQL Procedures, Functions and Loops on Number theory and business scenarios.

## 1. PL/SQL block to calculate the average age of students and display the result

DECLARE  
 total\_age NUMBER := 0;  
 num\_students NUMBER := 0;  
 avg\_age NUMBER := 0;  
BEGIN  
 -- Using a cursor to loop through all students  
 FOR student\_rec IN (SELECT Age FROM Student) LOOP  
 total\_age := total\_age + student\_rec.Age; -- Summing up the ages  
 num\_students := num\_students + 1; -- Counting the number of students  
 END LOOP;  
  
 -- Calculating the average age  
 IF num\_students > 0 THEN  
 avg\_age := total\_age / num\_students;  
 END IF;  
  
 -- Displaying the result  
 DBMS\_OUTPUT.PUT\_LINE('Total Students: ' || num\_students);  
 DBMS\_OUTPUT.PUT\_LINE('Total Age: ' || total\_age);  
 DBMS\_OUTPUT.PUT\_LINE('Average Age: ' || avg\_age);  
END;

|  |  |  |
| --- | --- | --- |
| Total Students | Total Age | Average Age |
| 14 | 342 | 24.42 |

## 2. PL/SQL block to insert a new student record into the Student table

DECLARE  
 v\_StudentID VARCHAR(6) := '&StudentID'; -- You can generate a unique StudentID as needed  
 v\_DeptID VARCHAR(6) := '&DeptID'; -- Replace with the actual DeptID  
 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\_Email VARCHAR(40) := '&email';  
 v\_Contact\_No NUMBER := &phone; -- Replace with the actual contact number  
BEGIN  
 INSERT INTO Student (StudentID, DeptID, FName, LName, Age, DateofBirth, Email, Contact\_No)  
 VALUES (v\_StudentID, v\_DeptID, v\_FName, v\_LName, v\_Age, v\_DateofBirth, v\_Email, v\_Contact\_No);  
 COMMIT;  
 DBMS\_OUTPUT.PUT\_LINE('Student record inserted successfully.');  
EXCEPTION  
 WHEN OTHERS THEN  
 DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);  
 ROLLBACK;  
END;  
/

Example input:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| StudentID | DeptID | FName | LName | Age | DateofBirth | Email | Contact\_No |
| 201 | CUB01 | Rahul | Sharma | 23 | 1999-07-17 | rahulsharma@gmail.com | 9797181815 |

Student record inserted successfully.

## 3. Function to return total number of departments in a particular University

CREATE OR REPLACE FUNCTION GetTotalDepartmentsInUniversity(UnivID VARCHAR2) RETURN NUMBER IS  
 v\_TotalDepartments NUMBER := 0;  
BEGIN  
 SELECT COUNT(\*) INTO v\_TotalDepartments FROM Department WHERE UnivID = UnivID;  
 RETURN v\_TotalDepartments;  
EXCEPTION  
 WHEN NO\_DATA\_FOUND THEN  
 RETURN 0; -- If university doesn't exist or has no departments  
 WHEN OTHERS THEN  
 RETURN -1; -- Indicate error  
END GetTotalDepartmentsInUniversity;  
/

Usage:

DECLARE  
num\_res NUMBER;  
BEGIN  
 num\_res := GetTotalDepartmentsInUniversity('UID01');  
 DBMS\_OUTPUT.PUT\_LINE('No. of departments: ' || num\_res);  
END;  
/

No. of departments: 2

## 4. Non-recursive PL/SQL procedure to retrieve even-numbered StudentIDs registered for any exam

CREATE OR REPLACE PROCEDURE GetEvenNumberedStudentIDs IS  
BEGIN  
 FOR student\_rec IN (SELECT StudentID FROM Student WHERE TO\_NUMBER(StudentID) MOD 2 = 0) LOOP  
 DBMS\_OUTPUT.PUT\_LINE('Even-Numbered StudentID: ' || student\_rec.StudentID);  
 END LOOP;  
END GetEvenNumberedStudentIDs;  
/

Result:

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