## Indian Institute of Information Technology Surat

****

# Lab Report on

# Advanced Database Management (CS 604) Practical

**Submitted by**

### [RAHUL KUMAR SINGH] (UI21CS44)

**Course Faculty**

### Mr. Rishi Sharma

## Department of Computer Science and Engineering

## Indian Institute of Information Technology Surat

## Gujarat-394190, India

**Jan-2024**

## Lab No: 2

**Aim: Write a PL/SQL code block to find total and**

**average of 6 subjects and display the grade.**

**Description:** The provided PL/SQL code block calculates the total and average scores of a student in six subjects and determines the corresponding grade based on a simple grading system.

* The code assumes the existence of a table named student\_scores with columns for student ID and scores in each of the six subjects.
* It fetches the scores for a specified student ID, calculates the total and average, and assigns a grade (A, B, C, D, or F) based on the average score.
* The results, including the student ID, total, average, and grade, are then displayed using the DBMS\_OUTPUT.PUT\_LINE function.

## Source Code:

**With Table:**

-- Table Creation

CREATE TABLE student\_scores (

student\_id NUMBER,

sub1 NUMBER,

sub2 NUMBER,

sub3 NUMBER,

sub4 NUMBER,

sub5 NUMBER,

sub6 NUMBER

);

-- Insertion

INSERT INTO student\_scores VALUES (1, 85, 92, 78, 90, 88, 95);

INSERT INTO student\_scores VALUES (2, 95, 88, 76, 82, 90, 87);

INSERT INTO student\_scores VALUES (3, 85, 98, 66, 72, 60, 67);

INSERT INTO student\_scores VALUES (4, 55, 88, 56, 72, 70, 77);

INSERT INTO student\_scores VALUES (5, 65, 78, 26, 82, 60, 77);

INSERT INTO student\_scores VALUES (6, 45, 68, 16, 52, 60, 57);

INSERT INTO student\_scores VALUES (7, 95, 38, 86, 82, 80, 87);

INSERT INTO student\_scores VALUES (8, 65, 58, 96, 62, 70, 57);

INSERT INTO student\_scores VALUES (9, 75, 98, 66, 42, 10, 87);

INSERT INTO student\_scores VALUES (10, 95, 88, 86, 12, 60, 57);

-- PL/SQL block to calculate total, average, and display grade

DECLARE

v\_student\_id student\_scores.student\_id%TYPE;

v\_sub1 student\_scores.sub1%TYPE;

v\_sub2 student\_scores.sub2%TYPE;

v\_sub3 student\_scores.sub3%TYPE;

v\_sub4 student\_scores.sub4%TYPE;

v\_sub5 student\_scores.sub5%TYPE;

v\_sub6 student\_scores.sub6%TYPE;

v\_total NUMBER;

v\_average NUMBER;

v\_grade VARCHAR2(2);

BEGIN

SELECT student\_id, sub1, sub2, sub3, sub4, sub5, sub6

INTO v\_student\_id, v\_sub1, v\_sub2, v\_sub3, v\_sub4, v\_sub5, v\_sub6

FROM student\_scores

WHERE student\_id = 1;

-- Calculate total and average

v\_total := v\_sub1 + v\_sub2 + v\_sub3 + v\_sub4 + v\_sub5 + v\_sub6;

v\_average := v\_total / 6;

-- Grade Determination

IF v\_average >= 90 THEN

v\_grade := 'A';

ELSIF v\_average >= 80 THEN

v\_grade := 'B';

ELSIF v\_average >= 70 THEN

v\_grade := 'C';

ELSIF v\_average >= 60 THEN

v\_grade := 'D';

ELSE

v\_grade := 'F';

END IF;

-- Results

DBMS\_OUTPUT.PUT\_LINE('Student ID: ' || v\_student\_id);

DBMS\_OUTPUT.PUT\_LINE('Total: ' || v\_total);

DBMS\_OUTPUT.PUT\_LINE('Average: ' || v\_average);

DBMS\_OUTPUT.PUT\_LINE('Grade: ' || v\_grade);

END;

/

**Without\_Table:**

**SET SERVEROUTPUT ON;**

**DECLARE**

**subject1 NUMBER := 85;**

**subject2 NUMBER := 92;**

**subject3 NUMBER := 78;**

**subject4 NUMBER := 90;**

**subject5 NUMBER := 88;**

**subject6 NUMBER := 95;**

**total NUMBER;**

**average NUMBER;**

**grade VARCHAR2(2);**

**BEGIN**

**total := subject1 + subject2 + subject3 + subject4 + subject5 + subject6;**

**average := total / 6;**

**IF average >= 90 THEN**

**grade := 'A';**

**ELSIF average >= 80 THEN**

**grade := 'B';**

**ELSIF average >= 70 THEN**

**grade := 'C';**

**ELSIF average >= 60 THEN**

**grade := 'D';**

**ELSE**

**grade := 'F';**

**END IF;**

**DBMS\_OUTPUT.PUT\_LINE('Total: ' || total);**

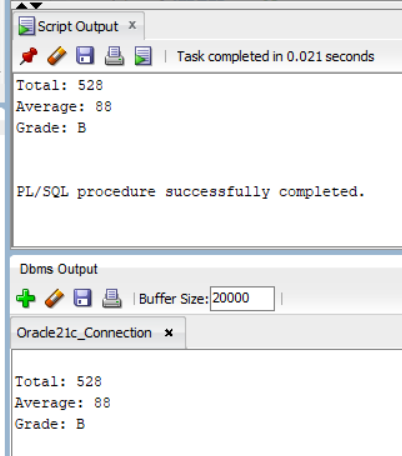
**DBMS\_OUTPUT.PUT\_LINE('Average: ' || average);**

**DBMS\_OUTPUT.PUT\_LINE('Grade: ' || grade);**

**END;**

**/**

## Output:



## Conclusion:

* The code is structured in a modular manner using a PL/SQL block for better understanding.
* The code is designed for execution in interactive environments.
* Utilized DECLARE and BEGIN sections to define variables and execute procedural logic.
* Applied the DBMS\_OUTPUT.PUT\_LINE function for displaying total marks, average marks, and the corresponding grade.