Experiment 2.1

Name: Rohan Jaiswal	**UID:** 21BCS2856
Branch: CSE	**Section/Group:** 608 (B)
Semester: 3rd	**Date of Performance:** 14/09/2022
Subject Name: DBMS	**Subject Code:** 21CSH-214

Aim:

Introduction and implementation of programs using control structures of conditional If-else and case statements.

Objective:

Learning and implementing PL/SQL Conditional Control Statements.

Theory:

Conditional Control:
L/SQL allows the use of an IF statement to control the execution of a block of code.

In PL/SQL, the IF -THEN - ELSIF - ELSE - END IF construct in code blocks allow specifying certain conditions under which a specific block of code should be executed.

1. IF-THEN Statement
Syntax

```
IF condition
THEN
Statement;
END IF;
```

This syntax is used when user needs to execute statements when condition is true.

2. IF-THEN-ELSE Statement
Syntax

```
IF condition
THEN
[Statements to execute when condition is TRUE]
ELSE
[Statements to execute when condition is FALSE]
END IF;
```

This syntax is used to execute one set of statements when condition is TRUE or different set of statements when condition is FALSE.

3. IF-THEN-ELSIF statement
Syntax

```
IF Condition1
THEN
Statements to execute when condition1 is TRUE
ELSIF condition2

THEN

Statements to execute when condition2 is TRUE

END IF;
```

This syntax is used to execute one set of statements when condition1 is TRUE or a different set of statements when condition is FALSE.

4. IF-THEN-ELS-IF-ELSE Statement
Syntax

```
IF condition1
THEN
Statements to execute when condition1 is TRUE
ELSIF condition2
THEN
Statements to execute when condition2 is TRUE
ELSE
Statements to execute when both condition1 and condition2 are FALSE
END IF;
```

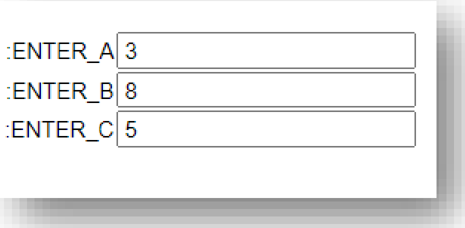
This syntax is used to execute one set of statements if condition1 is TRUE, a different set of statements when condition2 is TRUE or a third set of statements when both condition1 and condition2 are false.

DBMS script and Output:

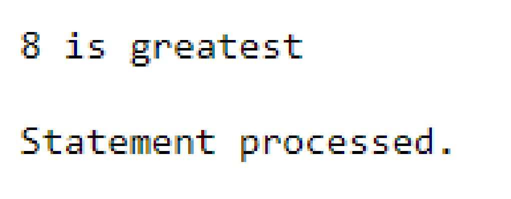
1. Write PL/SQL command to find the greatest of three numbers.

```
DECLARE
        a number;
        b number;
        c number;
BEGIN
        a :=:Enter_a;
        b :=:Enter_b;
        c :=:Enter_c;

        IF a>b then
                IF a>c then
                        dbms_output.put_line(a || ' is greatest');
                ELSE
                        dbms_output.put_line(c || ' is greatest');
                END if;
        ELSE
                IF b>c then
                        dbms_output.put_line(b || ' is greatest');
                ELSE
                        dbms_output.put_line(c || ' is greatest');
                END if;
        END if;
END;
```

Output:



```
:ENTER_A 3
:ENTER_B 8
:ENTER_C 5
```

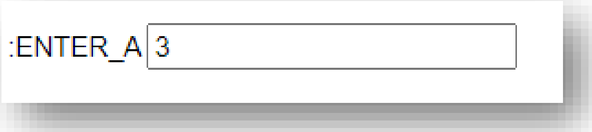


```
8 is greatest

Statement processed.
```

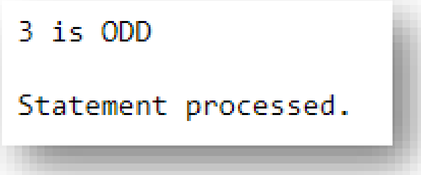
2. Write PL/SQL command to find the number entered by user is odd or even.

```
DECLARE
        a number;
BEGIN
        a :=:Enter_a;

        IF MOD(a,2)=0 THEN
                dbms_output.put_line(a ||' is EVEN');
        ELSE
                dbms_output.put_line(a ||' is ODD');
        END IF;
END;
```

Output:

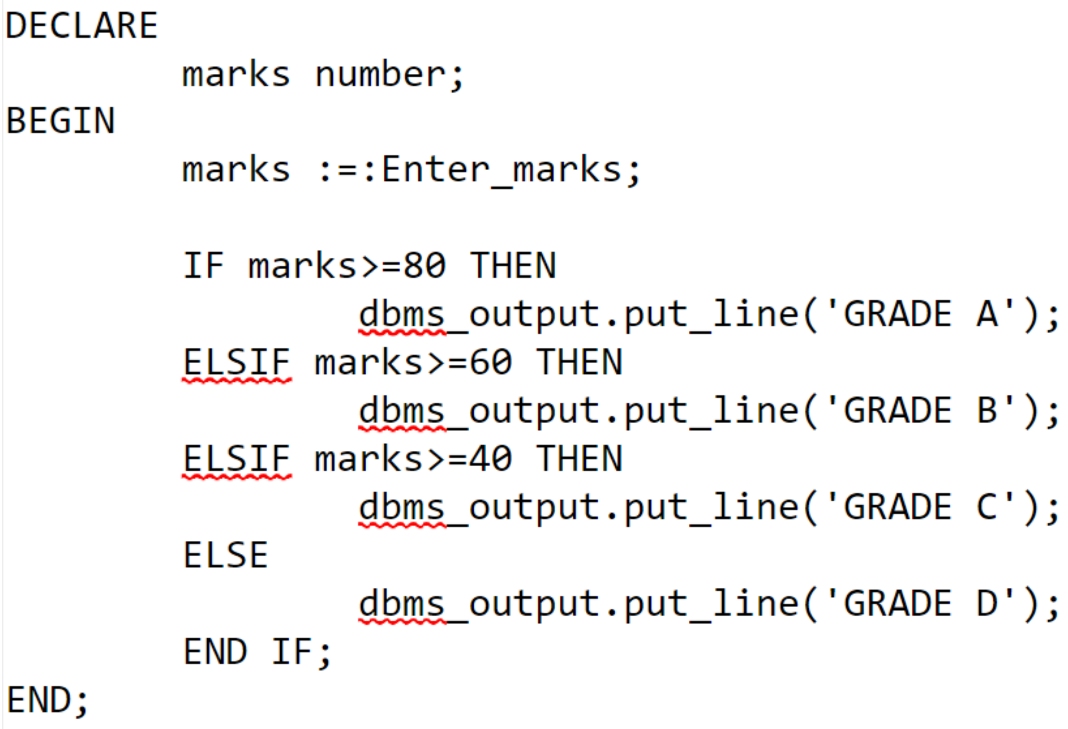


```
:ENTER_A 3
```



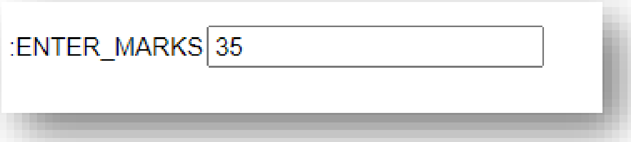
```
3 is ODD

Statement processed.
```

3. Find grade of student from marks entered by user at runtime.

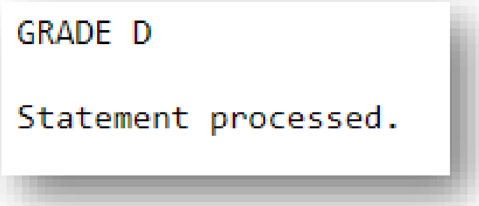


```
DECLARE
        marks number;
BEGIN
        marks :=:Enter_marks;

        IF marks>=80 THEN
                dbms_output.put_line('GRADE A');
        ELSIF marks>=60 THEN
                dbms_output.put_line('GRADE B');
        ELSIF marks>=40 THEN
                dbms_output.put_line('GRADE C');
        ELSE
                dbms_output.put_line('GRADE D');
        END IF;
END;
```

Output:



```
:ENTER_MARKS 35
```



```
GRADE D

Statement processed.
```

4.Create table “ CustID | Cname | Bal | Penality ”

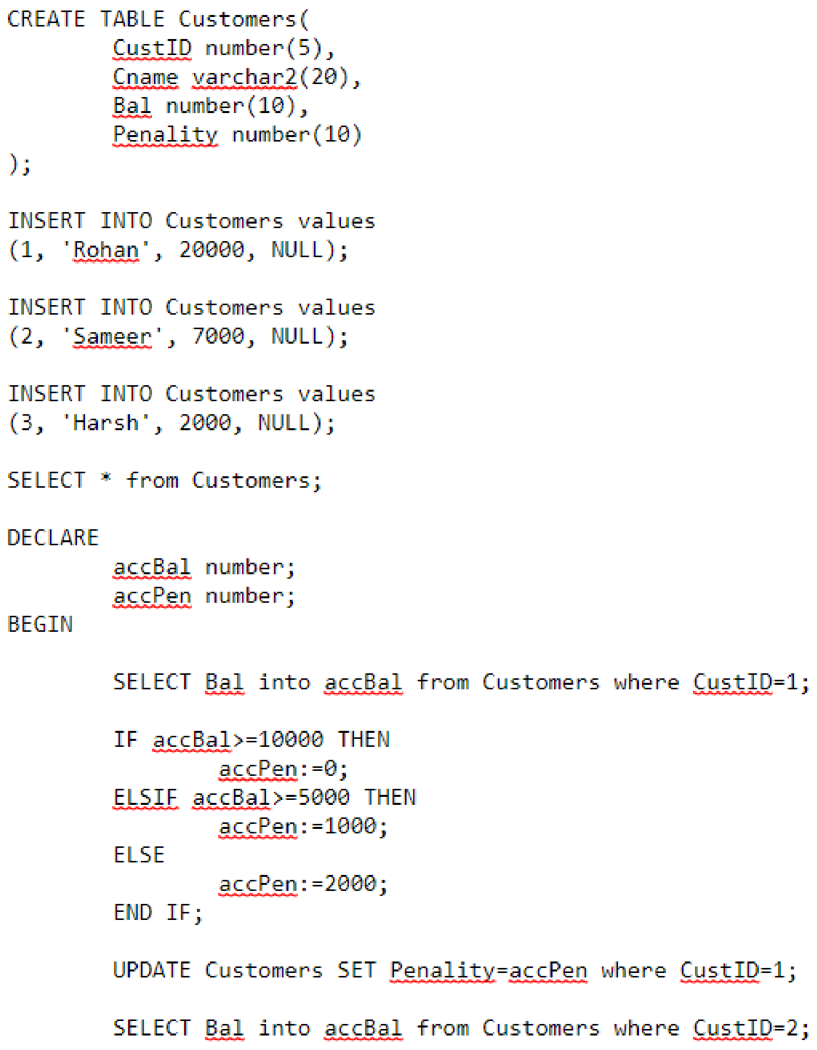
Find & update Penality of given customers

If Bal >=10000 , Penality = 0

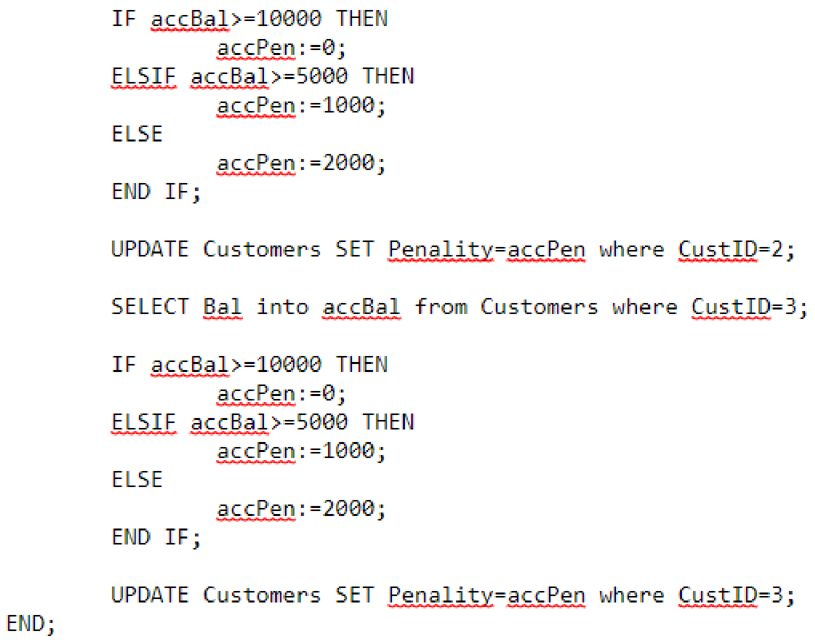
Bal >=5000 , Bal<10000 , Penality =1000

Bal < 5000 , Penality = 2500

Code:



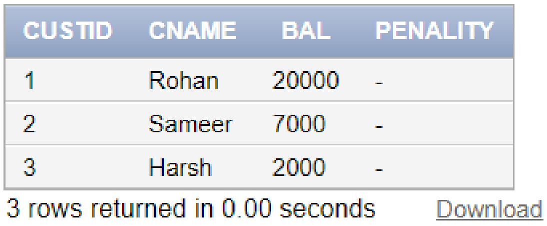
```
CREATE TABLE Customers(
        CustID number(5),
        Cname varchar2(20),
        Bal number(10),
        Penality number(10)
);

INSERT INTO Customers values
(1, 'Rohan', 20000, NULL);

INSERT INTO Customers values
(2, 'Sameer', 7000, NULL);

INSERT INTO Customers values
(3, 'Harsh', 2000, NULL);

SELECT * from Customers;

DECLARE
        accBal number;
        accPen number;
BEGIN

        SELECT Bal into accBal from Customers where CustID=1;

        IF accBal>=10000 THEN
                accPen:=0;
        ELSIF accBal>=5000 THEN
                accPen:=1000;
        ELSE
                accPen:=2000;
        END IF;

        UPDATE Customers SET Penality=accPen where CustID=1;

        SELECT Bal into accBal from Customers where CustID=2;
```



```
        IF accBal>=10000 THEN
                accPen:=0;
        ELSIF accBal>=5000 THEN
                accPen:=1000;
        ELSE
                accPen:=2000;
        END IF;

        UPDATE Customers SET Penality=accPen where CustID=2;

        SELECT Bal into accBal from Customers where CustID=3;

        IF accBal>=10000 THEN
                accPen:=0;
        ELSIF accBal>=5000 THEN
                accPen:=1000;
        ELSE
                accPen:=2000;
        END IF;

        UPDATE Customers SET Penality=accPen where CustID=3;
END;
```

Output:

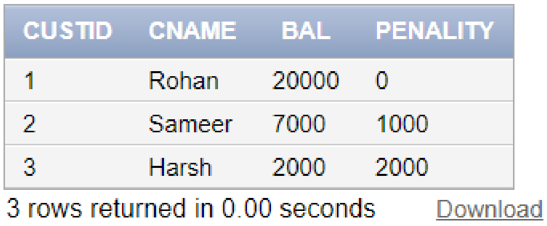
Before Updating:



CUSTID	CNAME	BAL	PENALITY
1	Rohan	20000	-
2	Sameer	7000	-
3	Harsh	2000	-

3 rows returned in 0.00 seconds Download

After Updating:



CUSTID	CNAME	BAL	PENALITY
1	Rohan	20000	0
2	Sameer	7000	1000
3	Harsh	2000	2000

3 rows returned in 0.00 seconds Download

Learning outcomes (What I have learnt):

1. Learned about DBMS languages.
2. I have learned about PL/SQL block Structure.
3. Learn about Four components of Pl/SQL and their function.