**Ayush Goyal**

**190905522 CSE D 62**

**DBS Lab-6 (Week 6) – PL/SQL Basics**

**Use a table StudentTable(RollNo, GPA) and populate the table with {(1, 5.8); (2, 6.5); (3, 3.4); (4,7.8); (5, 9.5)}unless a different DB schema is explicitly specified.**

**CODE:**

set serveroutput on

create table StudentTable(

rollNo number(2),

gpa numeric(4,2));

insert into StudentTable values(1,5.8);

insert into StudentTable values(2,6.5);

insert into StudentTable values(3,3.4);

insert into StudentTable values(4,7.8);

insert into StudentTable values(5,9.5);

**OUTPUT:**

**Text

Description automatically generated**

1. **Write a PL/SQL block to display the GPA of given student.**

**CODE:**

DECLARE

    roll\_number StudentTable.rollNo%TYPE;

    score StudentTable.gpa%TYPE;

BEGIN

    roll\_number:='&r';

    select gpa into score from StudentTable where rollNo=roll\_number;

    dbms\_output.put\_line(score);

END;

/

**OUTPUT:**

**Text

Description automatically generated**

**Usage of IF –THEN:**

1. **Write a PL/SQL block to display the letter grade (0-4: F; 4-5: E; 5-6: D; 6-7: C; 7-8: B; 8-9: A; 9-10: A+} of given student.**

**CODE:**

DECLARE

    roll\_number StudentTable.rollNo%TYPE;

    score StudentTable.gpa%TYPE;

BEGIN

    roll\_number:='&r';

    select gpa into score from StudentTable where rollNo=roll\_number;

IF score between 0 and 4 THEN

    dbms\_output.put\_line('F');

ELSIF score between 4 and 5 then

    dbms\_output.put\_line('E');

ELSIF score between 5 and 6 then

    dbms\_output.put\_line('D');

ELSIF score between 6 and 7 then

    dbms\_output.put\_line('C');

ELSIF score between 7 and 8 then

    dbms\_output.put\_line('B');

ELSIF score between 8 and 9 then

    dbms\_output.put\_line('A');

ELSE

    dbms\_output.put\_line('A+');

END IF;

END;

/

**OUTPUT:**

**Text

Description automatically generated**

1. **Input the date of issue and date of return for a book. Calculate and display the fine with the appropriate message using a PL/SQL block. The fine is charged as per the table 8.1:**

**Table

Description automatically generated**

**CODE:**

DECLARE

    issue\_date date;

    return\_date date;

    diff number;

    fine number;

BEGIN

    issue\_date:= TO\_DATE('&issue\_date','DD-MM-YYYY');

    return\_date:= TO\_DATE('&return\_date','DD-MM-YYYY');

    diff:=TO\_DATE(return\_date, 'DD-MM-YYYY') - TO\_DATE(issue\_date, 'DD-MM-YYYY');

IF diff between 0 and 7 THEN

    fine:=0;

ELSIF diff between 8 and 15 THEN

    fine := (diff-7)\*1;

ELSIF diff between 16 and 30 THEN

    fine := 8 + (diff-15)\*2;

ELSE

    fine := 8 + 30 + (diff-30)\*5;

END IF;

dbms\_output.put\_line('FINE = ' || fine);

END;

/

**OUTPUT:**

**Text

Description automatically generated**

**Simple LOOP:**

1. **Write a PL/SQL block to print the letter grade of all the students (Roll No: 1 -5).**

**CODE:**

DECLARE

    roll\_no StudentTable.rollNo%TYPE;

    score StudentTable.gpa%TYPE;

BEGIN

    roll\_no := 1;

LOOP

IF roll\_no > 5 THEN

EXIT;

END IF;

select gpa into score from StudentTable where rollNo=roll\_no;

IF score between 0 and 4 THEN

    dbms\_output.put\_line('Grade : F');

ELSIF score between 4 and 5 THEN

    dbms\_output.put\_line('Grade : E');

ELSIF score between 5 and 6 THEN

    dbms\_output.put\_line('Grade : D');

ELSIF score between 6 and 7 THEN

    dbms\_output.put\_line('Grade : C');

ELSIF score between 7 and 8 THEN

    dbms\_output.put\_line('Grade : B');

ELSIF score between 8 and 9 THEN

    dbms\_output.put\_line('Grade : A');

ELSE

    dbms\_output.put\_line('Grade : A+');

END IF;

roll\_no := roll\_no + 1;

END LOOP;

END;

/

**OUTPUT:**

**Text

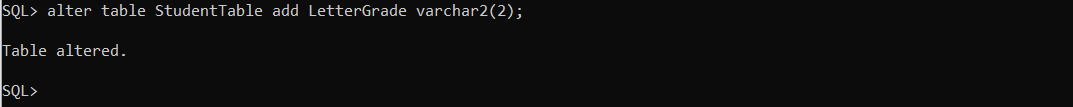
Description automatically generated**

**Usage of WHILE:**

1. **Alter StudentTable by appending an additional column LetterGrade Varchar2(2). Then write a PL/SQL block to update the table with letter grade of each student.**

**CODE:**

alter table StudentTable add LetterGrade varchar2(2);

****

DECLARE

    roll\_no StudentTable.rollNo%TYPE;

    score StudentTable.gpa%TYPE;

BEGIN

    roll\_no := 1;

while(roll\_no<6)

LOOP

IF roll\_no > 5 THEN

    EXIT;

END IF;

select gpa into score from StudentTable where rollNo=roll\_no;

IF score between 0 and 4 THEN

    update StudentTable set LetterGrade='F' where rollNo=roll\_no;

ELSIF score between 4 and 5 THEN

    update StudentTable set LetterGrade='E' where rollNo=roll\_no;

ELSIF score between 5 and 6 THEN

    update StudentTable set LetterGrade='D' where rollNo=roll\_no;

ELSIF score between 6 and 7 THEN

    update StudentTable set LetterGrade='C' where rollNo=roll\_no;

ELSIF score between 7 and 8 THEN

    update StudentTable set LetterGrade='B' where rollNo=roll\_no;

ELSIF score between 8 and 9 THEN

    update StudentTable set LetterGrade='A' where rollNo=roll\_no;

ELSE

    update StudentTable set LetterGrade='A+' where rollNo=roll\_no;

END IF;

roll\_no := roll\_no + 1;

END LOOP;

END;

/

**OUTPUT:**

**Text

Description automatically generated**

**Shape

Description automatically generated with medium confidence**

**Usage of FOR:**

1. **Write a PL/SQL block to find the student with max. GPA without using aggregate function.**

**CODE:**

DECLARE

i number := 1;

    roll\_no StudentTable.rollNo%TYPE;

    score StudentTable.gpa%TYPE;

    highest StudentTable.gpa%TYPE;

BEGIN

    roll\_no := 1;

select gpa into highest from StudentTable where rollNo=roll\_no;

FOR i IN 1..5 LOOP

select gpa into score from StudentTable where rollNo=roll\_no;

IF score>highest THEN

    highest:=score;

END IF;

roll\_no := roll\_no + 1;

END LOOP;

dbms\_output.put\_line('Max grade : ' || highest);

END;

/

**OUTPUT:**

**Text

Description automatically generated**

**Usage of GOTO:**

1. **Implement lab exercise4using GOTO.**

**CODE:**

DECLARE

    g char(2);

    roll\_no StudentTable.rollNo%TYPE;

    score StudentTable.gpa%TYPE;

BEGIN

    roll\_no := 1;

<<loopbegin>>

select gpa into score from StudentTable where rollNo=roll\_no;

IF score between 0 and 4 THEN

    g := 'F';

ELSIF score between 4 and 5 THEN

    g := 'E';

ELSIF score between 5 and 6 THEN

    g := 'D';

ELSIF score between 6 and 7 THEN

    g := 'C';

ELSIF score between 7 and 8 THEN

    g := 'B';

ELSIF score between 8 and 9 THEN

    g := 'A';

ELSE

    g := 'A+';

END IF;

dbms\_output.put\_line('Roll no : '||roll\_no||' Grade : '||g);

roll\_no := roll\_no + 1;

IF roll\_no<6 THEN

    GOTO loopbegin;

END IF;

END;

/

**OUTPUT:**

**Text

Description automatically generated**

**Exception Handling:**

1. **Based on the University database schema, write a PL/SQL block to display the details of the instructor whose name is supplied by the user. Use exceptions to show appropriate error message for the following cases:**
2. **Multiple instructors with the same name**

**CODE:**

DECLARE

    Multiple\_Instructor Exception;

    inst instructor%ROWTYPE;

    inp instructor.name%TYPE;

    n number(10);

BEGIN

    inp := '&name';

select count(id) into n from instructor group by name having name=inp;

IF n>1 THEN

    RAISE Multiple\_Instructor;

ELSE

    select \* into inst from instructor where instructor.name = inp;

    dbms\_output.put\_line(inst.id ||' '|| inst.name ||' '|| inst.dept\_name ||' '|| inst.salary);

END IF;

EXCEPTION

WHEN Multiple\_Instructor THEN

    dbms\_output.put\_line('Duplicate names found!');

END;

/

**OUTPUT:**

**Text

Description automatically generated**

1. **No instructor for the given name**

**CODE:**

DECLARE

    Multiple\_Instructor Exception;

    inst instructor%ROWTYPE;

    inp instructor.name%TYPE;

    n number(10);

BEGIN

inp := '&name';

select count(id) into n from instructor group by name having name=inp;

IF n>1 THEN

    RAISE Multiple\_Instructor;

ELSIF n=1 THEN

    select \* into inst from instructor where instructor.name = inp;

    dbms\_output.put\_line(inst.id ||' '|| inst.name ||' '|| inst.dept\_name ||' '|| inst.salary);

ELSE

    RAISE NO\_DATA\_FOUND;

END IF;

EXCEPTION

WHEN Multiple\_Instructor THEN

    dbms\_output.put\_line('Duplicate names found!');

WHEN NO\_DATA\_FOUND THEN

    dbms\_output.put\_line('Instructor not found!');

END;

/

**OUTPUT:**

**Text

Description automatically generated**

1. **Extend lab exercise5 to validate the GPA value used to find letter grade. If it is outside the range, 0 –10, display an error message, ‘Out of Range’ via an exception handler.**

We need to update someone’s gpa to more than 10 to check for exception.

update StudentTable set gpa=10.7 where rollNo=5;

Graphical user interface

Description automatically generated with low confidence

**CODE:**

DECLARE

    Out\_of\_range Exception;

    roll\_no StudentTable.rollNo%TYPE;

    score StudentTable.gpa%TYPE;

BEGIN

roll\_no := 1;

while(roll\_no<6)

LOOP

IF roll\_no > 5 THEN

    EXIT;

END IF;

select gpa into score from StudentTable where rollNo=roll\_no;

IF score between 0 and 4 THEN

    update StudentTable set LetterGrade='F' where rollNo=roll\_no;

ELSIF score between 4 and 5 THEN

    update StudentTable set LetterGrade='E' where rollNo=roll\_no;

ELSIF score between 5 and 6 THEN

    update StudentTable set LetterGrade='D' where rollNo=roll\_no;

ELSIF score between 6 and 7 THEN

    update StudentTable set LetterGrade='C' where rollNo=roll\_no;

ELSIF score between 7 and 8 THEN

    update StudentTable set LetterGrade='B' where rollNo=roll\_no;

ELSIF score between 8 and 9 THEN

    update StudentTable set LetterGrade='A' where rollNo=roll\_no;

ELSIF score between 9 and 10 THEN

    update StudentTable set LetterGrade='A+' where rollNo=roll\_no;

ELSE

    RAISE Out\_of\_range;

END IF;

roll\_no := roll\_no + 1;

END LOOP;

EXCEPTION

WHEN Out\_of\_range THEN

    dbms\_output.put\_line('GPA is out of range!');

END;

/

**OUTPUT:**

**Text

Description automatically generated**

**THE END**