

LAB EXCERSIZE :

/*NOTE: 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.

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

```
CREATE TABLE StudentTable(RollNo number(1) PRIMARY KEY,GPA number(2,1));
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(5,4.5);
INSERT INTO StudentTable VALUES(6,9.5);
INSERT INTO StudentTable VALUES(7,9.5);
INSERT INTO StudentTable VALUES(8,9.5);
INSERT INTO StudentTable VALUES(9,9.5);
```

DECLARE

answer float;

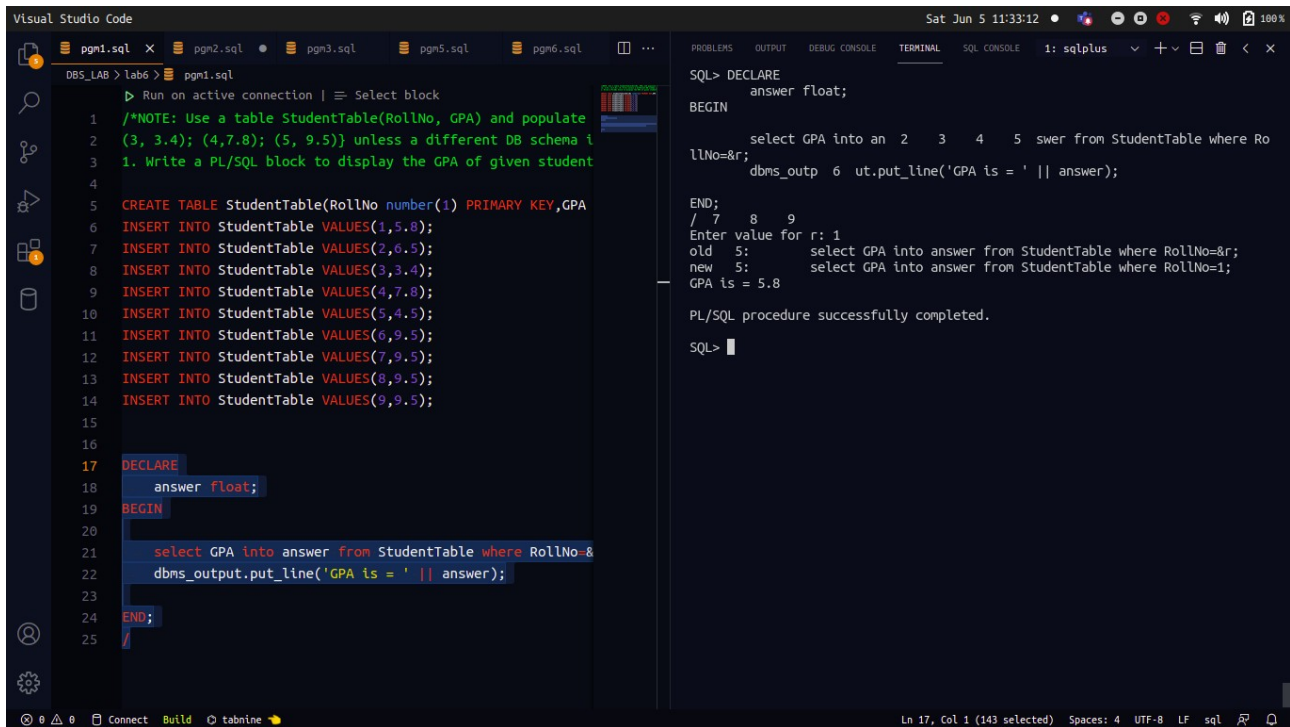
BEGIN

```
select GPA into answer from StudentTable where RollNo=&r;
dbms_output.put_line('GPA is = ' || answer);
```

END;

/

OUTPUT :



The screenshot shows the Visual Studio Code interface. The editor on the left contains a PL/SQL script for creating a table, inserting data, and a PL/SQL block to query the GPA. The terminal on the right shows the execution of this script, including the table creation, data insertion, and the successful execution of the PL/SQL block which outputs 'GPA is = 5.8'.

```
DBS_LAB > Tab6 > pgm1.sql
Run on active connection | Select block
/*NOTE: Use a table StudentTable(RollNo, GPA) and populate
(3, 3.4); (4,7.8); (5, 9.5)} unless a different DB schema i
1. Write a PL/SQL block to display the GPA of given student
4
5 CREATE TABLE StudentTable(RollNo number(1) PRIMARY KEY,GPA
6 INSERT INTO StudentTable VALUES(1,5.8);
7 INSERT INTO StudentTable VALUES(2,6.5);
8 INSERT INTO StudentTable VALUES(3,3.4);
9 INSERT INTO StudentTable VALUES(4,7.8);
10 INSERT INTO StudentTable VALUES(5,4.5);
11 INSERT INTO StudentTable VALUES(6,9.5);
12 INSERT INTO StudentTable VALUES(7,9.5);
13 INSERT INTO StudentTable VALUES(8,9.5);
14 INSERT INTO StudentTable VALUES(9,9.5);
15
16
17 DECLARE
18     answer float;
19 BEGIN
20
21     select GPA into answer from StudentTable where RollNo=8
22     dbms_output.put_line('GPA is = ' || answer);
23
24 END;
25

SQL> DECLARE
      answer float;
BEGIN
      select GPA into an 2 3 4 5 swer from StudentTable where Ro
      llNo=&r;
      dbms_outp 6 ut.put_line('GPA is = ' || answer);
END;
/ 7 8 9
Enter value for r: 1
old 5:      select GPA into answer from StudentTable where RollNo=&r;
new 5:      select GPA into answer from StudentTable where RollNo=1;
GPA is = 5.8
PL/SQL procedure successfully completed.
SQL>
```

/*QUESTION-2. 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.

Number Grade

0-4 F
4-5 E
5-6 D
6-7 C
7-8 B
8-9 A
9-10 A+ */

```
CREATE TABLE StudentTable1(RollNo number(2) PRIMARY KEY,gpa VARCHAR(10));
INSERT INTO StudentTable1 values(4,'F');
INSERT INTO StudentTable1 values(5,'E');
INSERT INTO StudentTable1 values(6,'D');
INSERT INTO StudentTable1 values(7,'C');
INSERT INTO StudentTable1 values(8,'B');
INSERT INTO StudentTable1 values(9,'A');
INSERT INTO StudentTable1 values(10,'A+');
```

DECLARE

```

result number(2);
grade VARCHAR(10);
BEGIN
select gpa into result from StudentTable where RollNo=&r;

if (result > 9) then grade:='A+';
elsif (result > 8) then grade:='A';
elsif (result > 7) then grade:='B';
elsif (result > 6) then grade:='C';
elsif (result > 5) then grade:='D';
elsif (result > 4) then grade:='E';
else grade:='F';
end if;

dbms_output.put_line('grade is ' || grade);
END;
/

```

OUTPUT :

The screenshot shows the Visual Studio Code editor with a PL/SQL script in the left pane and its execution output in the right pane (Terminal).

Left Pane (Code):

```

1  /*2. Write a PL/SQL block to display the letter grade(0-4:
2  7-8: B; 8-9: A; 9-10: A+) of given student.
3
4  Number          Grade
5  0-4             F
6  4-5             E
7  5-6             D
8  6-7             C
9  7-8             B
10 8-9             A
11 9-10            A+ */
12
13 CREATE TABLE StudentTable1(RollNo number(2) PRIMARY KEY,gpa
14 INSERT INTO StudentTable1 values(4,'F');
15 INSERT INTO StudentTable1 values(5,'E');
16 INSERT INTO StudentTable1 values(6,'D');
17 INSERT INTO StudentTable1 values(7,'C');
18 INSERT INTO StudentTable1 values(8,'B');
19 INSERT INTO StudentTable1 values(9,'A');
20 INSERT INTO StudentTable1 values(10,'A+');
21
22 DECLARE
23     result number(2);
24     grade VARCHAR(10);
25 BEGIN
26     select gpa into result from StudentTable where RollNo=&
27
28     if (result > 9) then grade:='A+';
29     elsif (result > 8) then grade:='A';

```

Right Pane (Terminal):

```

select GPA into an 2 3 4 5 swer from StudentTable where Ro
llNo=&r;
dbms_outp 6 ut.put_line('GPA is = ' || answer);

END;
/ 7 8 9
Enter value for r: 1
old 5: select GPA into answer from StudentTable where RollNo=&r;
new 5: select GPA into answer from StudentTable where RollNo=1;
GPA is = 5.8

PL/SQL procedure successfully completed.

SQL> DECLARE
result number(2);
grade VARCHAR(10);
2 3 4 BEGIN
select gpa into result from StudentTable w 5 here RollNo=&r;

if (result > 9) then grade:='A+ 6 7 ' ;
elsif (result > 8) then grade:='A';
elsif ( 8 9 result > 7) then grade:='B';
elsif (result > 6) 10 then grade:='C';
elsif (result > 5) then grade:= 11 'D';
elsif (result > 4) then grade:='E';
else 12 13 grade:='F';
end if;

dbms_output.put_line('grad 14 15 16 e is ' || grade);

END;
/ 17 18
Enter value for r: 5
old 5: select gpa into result from StudentTable where RollNo=&r;
new 5: select gpa into result from StudentTable where RollNo=5;
grade is A+

PL/SQL procedure successfully completed.

SQL>

```

/*QUESTION-3. 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:

Late period	Fine
7 days	NIL
8 - 15 days	Rs.1/day
16 - 30 days	Rs.2/ day
After 30 days	Rs.5.00

Table

```

*/
DECLARE
myDate date;
ReturnDate date;
days integer;

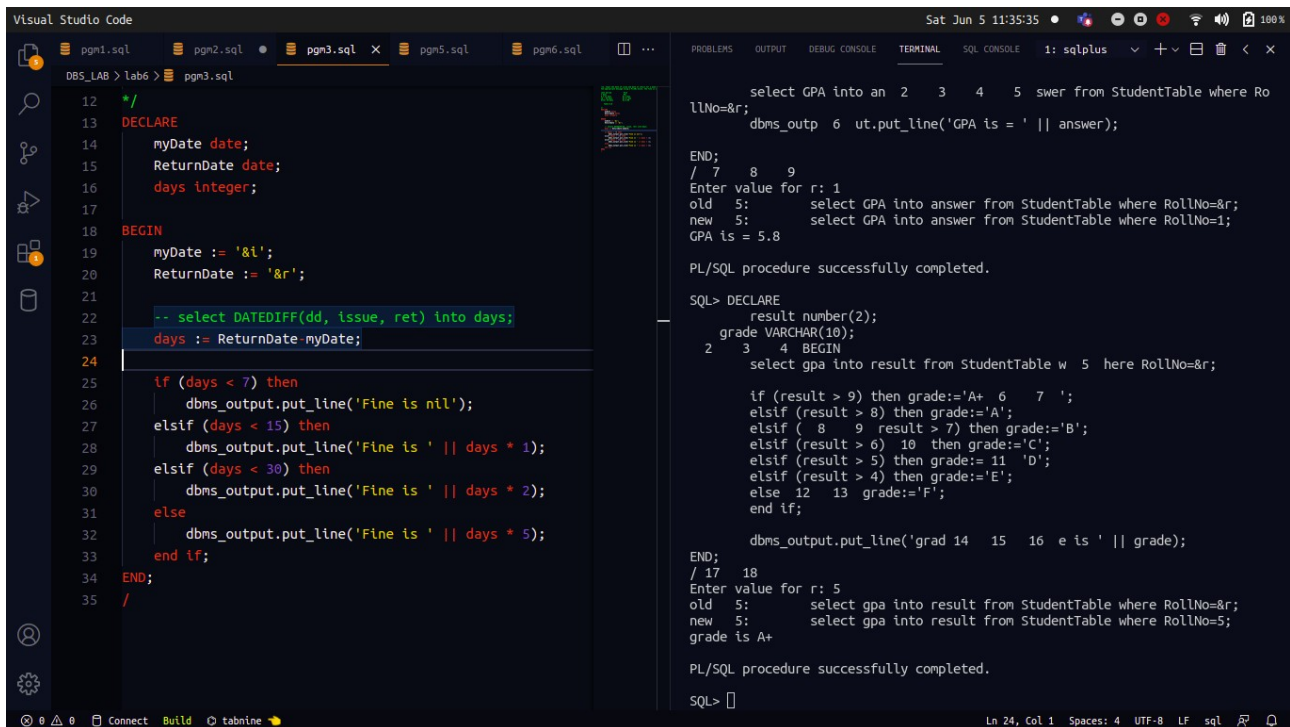
BEGIN
myDate := '&i';
ReturnDate := '&r';

-- select DATEDIFF(dd, issue, ret) into days;
days := ReturnDate-myDate;

if (days < 7) then
dbms_output.put_line('Fine is nil');
elsif (days < 15) then
dbms_output.put_line('Fine is ' || days * 1);
elsif (days < 30) then
dbms_output.put_line('Fine is ' || days * 2);
else
dbms_output.put_line('Fine is ' || days * 5);
end if;
END;
/

```

OUTPUT :



The screenshot shows the Visual Studio Code editor with a file named `pgm3.sql` open. The code is a PL/SQL procedure that calculates the number of days between two dates and prints a message based on the result. The terminal on the right shows the execution of the procedure, which successfully completed and printed the message: "Fine is 5".

```
12  */
13  DECLARE
14      myDate date;
15      ReturnDate date;
16      days integer;
17
18  BEGIN
19      myDate := '&l';
20      ReturnDate := '&r';
21
22      -- select DATEDIFF(dd, issue, ret) into days;
23      days := ReturnDate-myDate;
24
25      if (days < 7) then
26          dbms_output.put_line('Fine is nil');
27      elsif (days < 15) then
28          dbms_output.put_line('Fine is ' || days * 1);
29      elsif (days < 30) then
30          dbms_output.put_line('Fine is ' || days * 2);
31      else
32          dbms_output.put_line('Fine is ' || days * 5);
33      end if;
34  END;
35  /
```

Terminal Output:

```
select GPA into an 2 3 4 5 swer from StudentTable where Ro
llNo=&r;
dbms_outp 6 ut.put_line('GPA is = ' || answer);

END;
/ 7 8 9
Enter value for r: 1
old 5: select GPA into answer from StudentTable where RollNo=&r;
new 5: select GPA into answer from StudentTable where RollNo=1;
GPA is = 5.8

PL/SQL procedure successfully completed.

SQL> DECLARE
    result number(2);
    grade VARCHAR(10);
2 3 4 BEGIN
    select gpa into result from StudentTable w 5 here RollNo=&r;

    if (result > 9) then grade:='A+ 6 7 ' ;
    elsif (result > 8) then grade:='A';
    elsif ( 8 9 result > 7) then grade:='B';
    elsif (result > 6) 10 then grade:='C';
    elsif (result > 5) then grade:= 11 'D';
    elsif (result > 4) then grade:='E';
    else 12 13 grade:='F';
    end if;

    dbms_output.put_line('grad 14 15 16 e is ' || grade);

END;
/ 17 18
Enter value for r: 5
old 5: select gpa into result from StudentTable where RollNo=&r;
new 5: select gpa into result from StudentTable where RollNo=5;
grade is A+

PL/SQL procedure successfully completed.

SQL>
```

QUESTION-4. Write a PL/SQL block to print the letter grade of all the students(RollNo: 1 - 5).*/

DECLARE

result float;

grade varchar(2);

myNumber number(1);

BEGIN

FOR myNumber in 1..5 LOOP

select GPA into result from StudentTable where RollNo=myNumber;

if (result > 9) then grade:='A+';

elsif (result > 8) then grade:='A';

elsif (result > 7) then grade:='B';

elsif (result > 6) then grade:='C';

elsif (result > 5) then grade:='D';

```

elsif (result > 4) then grade:='E';
else grade:='F';
end if;

dbms_output.put_line('grade is = ' || grade);
END LOOP;
END;
/

```

The screenshot shows the Visual Studio Code editor with a file named `pgm4.sql` open. The script contains a PL/SQL block that declares variables `result` (float), `grade` (varchar(2)), and `myNumber` (number(1)). It then enters a `FOR` loop from 1 to 5, where it selects the GPA into the `result` variable from the `StudentTable` based on the `RollNo` (which is `myNumber`). Inside the loop, it uses an `IF` statement to assign a letter grade based on the GPA value. After the loop, it prints the grade for each number using `dbms_output.put_line`.

The SQL console on the right shows the execution of the script, displaying the output: `grade is = D`, `grade is = C`, `grade is = F`, `grade is = B`, and `grade is = A+`.

QUESTION-5. 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.*/

```

alter table StudentTable
add LetterGrade varchar2(2);

```

```

DECLARE
result float;
grade varchar(2);
myNumber number(1);

```

```

BEGIN

```

```

FOR myNumber in 1..5 LOOP
select GPA into result from StudentTable where RollNo=myNumber;

```

```

if ( result > 9) then grade:='A+';
elsif (result > 8) then grade:='A';
elsif (result > 7) then grade:='B';
elsif (result > 6) then grade:='C';
elsif (result > 5) then grade:='D';
elsif (result > 4) then grade:='E';
else grade:='F';
end if;
update StudentTable set LetterGrade=grade where RollNo=myNumber;
END LOOP;
END;
/

```

OUTPUT :

The screenshot shows the Visual Studio Code interface with a PL/SQL script in the editor and its execution output in the terminal.

Editor Content (pgm5.sql):

```

1  /*5. Alter StudentTable by appending an additional column L
2  write a PL/SQL block to update the table with letter grade
3
4  alter table StudentTable
5  add LetterGrade varchar2(2);
6
7  DECLARE
8      result float;
9      grade varchar(2);
10     myNumber number(1);
11
12 BEGIN
13
14     FOR myNumber in 1..5 LOOP
15         select GPA into result from StudentTable where Roll
16
17         if ( result > 9) then grade:='A+';
18         elsif (result > 8) then grade:='A';
19         elsif (result > 7) then grade:='B';
20         elsif (result > 6) then grade:='C';
21         elsif (result > 5) then grade:='D';
22         elsif (result > 4) then grade:='E';
23         else grade:='F';
24         end if;
25
26         update StudentTable set LetterGrade=grade where Rol
27
28     END LOOP;
29

```

Terminal Output:

```

SQL> alter table StudentTable
add LetterGrade varchar2( 2 2);

DECLARE

result float;
grade varchar(2);
myNumber number(1);

BEGIN

    FOR myNumber in 1..5 Loop add LetterGrade varchar2(2)

*
ERROR at line 2:
ORA-01430: column being added already exists in table

SQL> SQL> 2 3 4 5 6 7 8 OP

select GPA into result from StudentTable wh 9 10 ere Roll

No=myNumber;

if ( result > 9) then g 11 12 rade:='A+';
elsif (result > 8) then grade:='A'; 13
elsif (result > 7) then grade:='B';
elsif ( 14 15 result > 6) then grade:='C';
elsif (result > 5) 16 then grade:='D';
elsif (result > 4) then grade 17 := 'E';
else grade:='F';
end if;

update St 18 19 20 21 udentTable set LetterGrade=grade

where RollNo=myNumber;

END LOOP;

END;
/ 22 23 24 25 26

PL/SQL procedure successfully completed.

```



```

1  /*5. Alter StudentTable by appending an additional column L
2  write a PL/SQL block to update the table with letter grade
3
4  alter table StudentTable
5  add LetterGrade varchar2(2);
6
7  DECLARE
8      result float;
9      grade varchar(2);
10     myNumber number(1);
11
12 BEGIN
13
14     FOR myNumber in 1..5 LOOP
15         select GPA into result from StudentTable where Roll
16
17         if ( result > 9) then grade:='A+';
18         elsif (result > 8) then grade:='A';
19         elsif (result > 7) then grade:='B';
20         elsif (result > 6) then grade:='C';
21         elsif (result > 5) then grade:='D';
22         elsif (result > 4) then grade:='E';
23         else grade:='F';
24         end if;
25
26         update StudentTable set LetterGrade=grade where Rol
27
28     END LOOP;
29
*/

```

```

No=myNumber;
select GPA into result from StudentTable wh 9 10 ere Roll

if ( result > 9) then g 11 12 rade:='A+';
elsif (result > 8) then grade:='A'; 13
elsif (result > 7) then grade:='B';
elsif ( 14 15 result > 6) then grade:='C';
elsif (result > 5) 16 then grade:='D';
elsif (result > 4) then grade 17 := 'E';
else grade:='F';
end if;

update St 18 19 20 21 udentTable set LetterGrade=grade
where RollNo=myNumber;

END LOOP;

END;
/ 22 23 24 25 26

PL/SQL procedure successfully completed.

SQL> select * from StudentTable;

ROLLNO      GPA LE
-----
1          5.8 D
2          6.5 C
3          3.4 F
4          7.8 B
5          9.5 A+
6          9.5
7          9.5
8          9.5
9          9.5

9 rows selected.

SQL>

```

/*QUESTION-6. Write a PL/SQL block to find the student with max. GPA without using aggregate function.*/

```

DECLARE
result float;
helloMaximum float;
studentNumber number(1);
myNumber number(1);
BEGIN
select GPA into helloMaximum from StudentTable where RollNo=1;
studentNumber:=1;

FOR myNumber in 2..5 LOOP

select GPA into result from StudentTable where RollNo=myNumber;
if (result > helloMaximum) then studentNumber:=myNumber;
end if;
if (result > helloMaximum) then helloMaximum:=result;
end if;
END LOOP;
dbms_output.put_line('RollNo is = ' || studentNumber || ' GPA is = ' ||
helloMaximum);
END;
/

```

OUTPUT :

The screenshot shows the Visual Studio Code interface. On the left, the editor displays a PL/SQL script for finding the student with the maximum GPA. The script includes a comment, a function declaration, a loop, and a select statement. On the right, the SQL console shows the execution of the script, with the output indicating that the procedure was successfully completed.

```

1  /*6. Write a PL/SQL block to find the student with max. GPA
2  function.*/
3  DECLARE
4      result float;
5      helloMaximum float;
6      studentNumber number(1);
7      myNumber number(1);
8  BEGIN
9      select GPA into helloMaximum from StudentTable where RollNo=1;
10     studentNumber:=1;
11
12     FOR myNumber in 2..5 LOOP
13
14         select GPA into result from StudentTable where Roll
15
16         if (result > helloMaximum) then studentNumber:=myNu
17         end if;
18         if (result > helloMaximum) then helloMaximum:=resul
19         end if;
20
21     END LOOP;
22     dbms_output.put_line('RollNo is = ' || studentNumber ||
23
24     /

```

```

SQL> DECLARE
result float;
helloMaximum float;
stude 2 3 4 ntNumber number(1);
myNumber number(1);
BEGIN
s 5 6 7 elect GPA into helloMaximum from StudentTable where R
ollNo=1;
studentNumber:=1;
FOR myNumber in 8 9 10 2..5 LOOP
select GPA into result from StudentT 11 12 able where Roll
No=myNumber;
if (result > hel 13 14 loMaximum) then studentNumber:=myNu
mber;
end if 15 ;
if (result > helloMaximum) then helloMaximum: 16 =result;
end if;
END LOOP;
dbms_output.p 17 18 19 20 ut_line('RollNo is = ' || studentNum
ber || ' GPA is = ' || helloMaximum);
END;
/ 21 22
RollNo is = 5 GPA is =9.5
PL/SQL procedure successfully completed.
SQL>

```

/*QUESTION-7. Implement lab exercise 4 using GOTO.*/

DECLARE

result StudentTable.gpa%TYPE;
grade varchar(2);

BEGIN

for i in 1..5 loop

select gpa into result from StudentTable where rollno = i;

if (result>=9 and result<=10) then goto ap;
elsif (result>=8 and result<9) then goto aa;
elsif (result>=7 and result<8) then goto bb;
elsif (result>=6 and result<7) then goto cc;
elsif (result>=5 and result<6) then goto dd;
elsif (result>=4 and result<5) then goto ee;
else goto ff;
end if;

<<ap>>

grade:='A+';

goto prnt;

<<aa>>

grade:='A';

goto prnt;

```

<<bb>>
grade:='B';
goto prnt;
<<cc>>
grade:='C';
goto prnt;
<<dd>>
grade:='D';
goto prnt;
<<ee>>
grade:='E';
goto prnt;
<<ff>>
grade:='F';
<<prnt>>
dbms_output.put_line('RollNo is = ' || i || ' Grade is = ' || grade);

end loop;

END;
/

```

OUTPUT :

The screenshot shows the Visual Studio Code editor with a PL/SQL script in the left pane and its output in the right pane. The script is a PL/SQL block that implements a loop to calculate grades based on a result value. The output shows the execution of the script, displaying the roll number and grade for each iteration.

```

Visual Studio Code
DBS_LAB > lab6 > pgm7.sql
1  /*7. Implement lab exercise 4 using GOTO.*/
2
3  DECLARE
4      result StudentTable.gpa%TYPE;
5      grade varchar(2);
6
7  BEGIN
8      for i in 1..5 loop
9          select gpa into result from StudentTable where roll
10
11             if ( result>=9 and result<=10) then goto ap;
12             elsif (result>=8 and result<9) then goto aa;
13             elsif (result>=7 and result<8) then goto bb;
14             elsif (result>=6 and result<7) then goto cc;
15             elsif (result>=5 and result<6) then goto dd;
16             elsif (result>=4 and result<5) then goto ee;
17             else goto ff;
18             end if;
19
20             <<ap>>
21                 grade:='A+';
22                 goto prnt;
23
24             <<aa>>
25                 grade:='A';
26                 goto prnt;
27
28             <<bb>>
29                 grade:='B';

```

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL SQL CONSOLE 1: sqlplus
<<aa>>
    grade:='A';
    goto prnt;

<<bb 22 23 24 25 26 >>
    grade:='B';
    goto prnt;

<<cc>>
    gra 27 28 29 30 31 de:='C';
    goto prnt;

<<dd>>
    grade:='D';
    goto prnt;

<<ee>>
    grade:='E';
    goto p 37 38 39 40 rnt;

<<ff>>
    grade:='F';

<<prnt>>
    d 41 42 43 44 45 46 bms_output.put_line('R
ollNo is = ' || i || ' Grade is = ' || grade);

end loop;

END;
/ 47 48 49 50 51
RollNo is = 1 Grade is = D
RollNo is = 2 Grade is = C
RollNo is = 3 Grade is = F
RollNo is = 4 Grade is = B
RollNo is = 5 Grade is = A+

PL/SQL procedure successfully completed.

SQL>

```

/*QUESTION-8. Based on the University database schema, write a PL/SQL block to display the details

appropriate error message for the following cases:

- a. Multiple instructors with the same name
- b. No instructor for the given name*/

DECLARE

```
namestudent instructor.name%TYPE;  
val instructor%ROWTYPE;
```

BEGIN

```
namestudent := '&myNumber';
select * into val from instructor where name=namestudent;
dbms_output.put_line(val);
END;
/
```

OUTPUT :

Visual Studio Code

pgm5.sql

pgm6.sql

pgm7.sql

pgm8.sql

pgm9.sql

DBS_LAB > lab6 > pgm8.sql

Run on active connection | Select block

```
1  /*8. Based on the University database schema, write a PL/SQL
2  of the Instructor whose name is supplied by the user. Use e
3  appropriate error message for the following cases:
4  a. Multiple instructors with the same name
5  b. No instructor for the given name*/
6  DECLARE
7      namestudent instructor.name%TYPE;
8      val instructor%ROWTYPE;
9
10 BEGIN
11     namestudent := '&myNumber';
12     select * into val from instructor where name=namestuden
13     dbms_output.put_line(val);
14 END;
15
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

SQL CONSOLE

1: sqlplus

goto print;

<<cc>>

gra 27 28 29 30 31 de:='C';

goto print;

<<dd>>

grade:='D';

32 33 34 35 36

goto print;

<<ee>>

grade:='E';

goto p 37 38 39 40 rnt;

<<ff>>

grade:='F';

<<print>>

d 41 42 43 44 45 46 bms_output.put_line('R

ollNo is = ' || i || ' Grade is = ' || grade);

end loop;

END;

/ 47 48 49 50 51

RollNo is = 1 Grade is = D

RollNo is = 2 Grade is = C

RollNo is = 3 Grade is = F

RollNo is = 4 Grade is = B

RollNo is = 5 Grade is = A+

PL/SQL procedure successfully completed.

SQL> DECLARE

namestudent instructor.name%TYPE;

val in 2 3 structor%ROWTYPE;

BEGIN

namestudent := '&myNumbe 4 5 6 r';

select * into val from instructor where name= 7 namestudent;

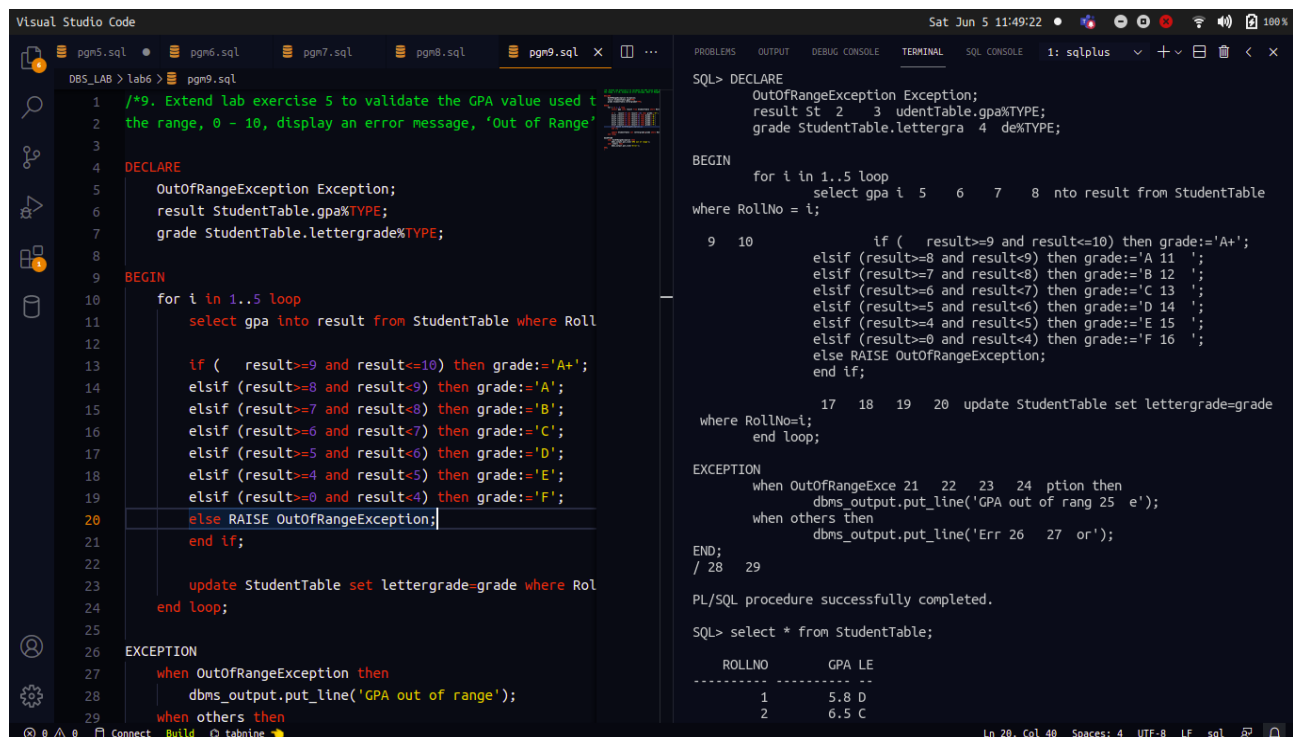
Ln 15, Col 2 Spaces: 4 UTF-8 LF sql 52

```
/*QUESTION-9. Extend lab exercise 5 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.*/
```

DECLARE

```
OutOfRangeException Exception;
result StudentTable.gpa%TYPE;
grade StudentTable.lettergrade%TYPE;
BEGIN
for i in 1..5 loop
select gpa into result from StudentTable where RollNo = i;
if ( result>=9 and result<=10) then grade:='A+';
elsif (result>=8 and result<9) then grade:='A';
elsif (result>=7 and result<8) then grade:='B';
elsif (result>=6 and result<7) then grade:='C';
elsif (result>=5 and result<6) then grade:='D';
elsif (result>=4 and result<5) then grade:='E';
elsif (result>=0 and result<4) then grade:='F';
else RAISE OutOfRangeException;
end if;
update StudentTable set lettergrade=grade where RollNo=i;
end loop;
EXCEPTION
when OutOfRangeException then
dbms_output.put_line('GPA out of range');
when others then
dbms_output.put_line('Error');
END;
/
```

OUTPUT :



The screenshot shows the Visual Studio Code interface with a SQL script in the editor and its execution output in the terminal. The SQL script defines an exception, loops through GPA values, assigns letter grades, and handles errors. The terminal output shows the successful completion of the PL/SQL procedure and a query result table.

```
SQL> DECLARE
  OutOfRangeException Exception;
  result StudentTable.gpa%TYPE;
  grade StudentTable.lettergrade%TYPE;
BEGIN
  for i in 1..5 loop
    select gpa into result from StudentTable where RollNo = i;
    if ( result>=9 and result<=10) then grade:='A+';
    elsif (result>=8 and result<9) then grade:='A';
    elsif (result>=7 and result<8) then grade:='B';
    elsif (result>=6 and result<7) then grade:='C';
    elsif (result>=5 and result<6) then grade:='D';
    elsif (result>=4 and result<5) then grade:='E';
    elsif (result>=0 and result<4) then grade:='F';
    else RAISE OutOfRangeException;
    end if;
    update StudentTable set lettergrade=grade where RollNo=i;
  end loop;
EXCEPTION
  when OutOfRangeException then
    dbms_output.put_line('GPA out of range');
  when others then
    dbms_output.put_line('Error');
END;
/
```

PL/SQL procedure successfully completed.

```
SQL> select * from StudentTable;
```

ROLLNO	GPA	LE
1	5.8	D
2	6.5	C