

LAB 6**LAB EXERCISE**

1)

```
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
```

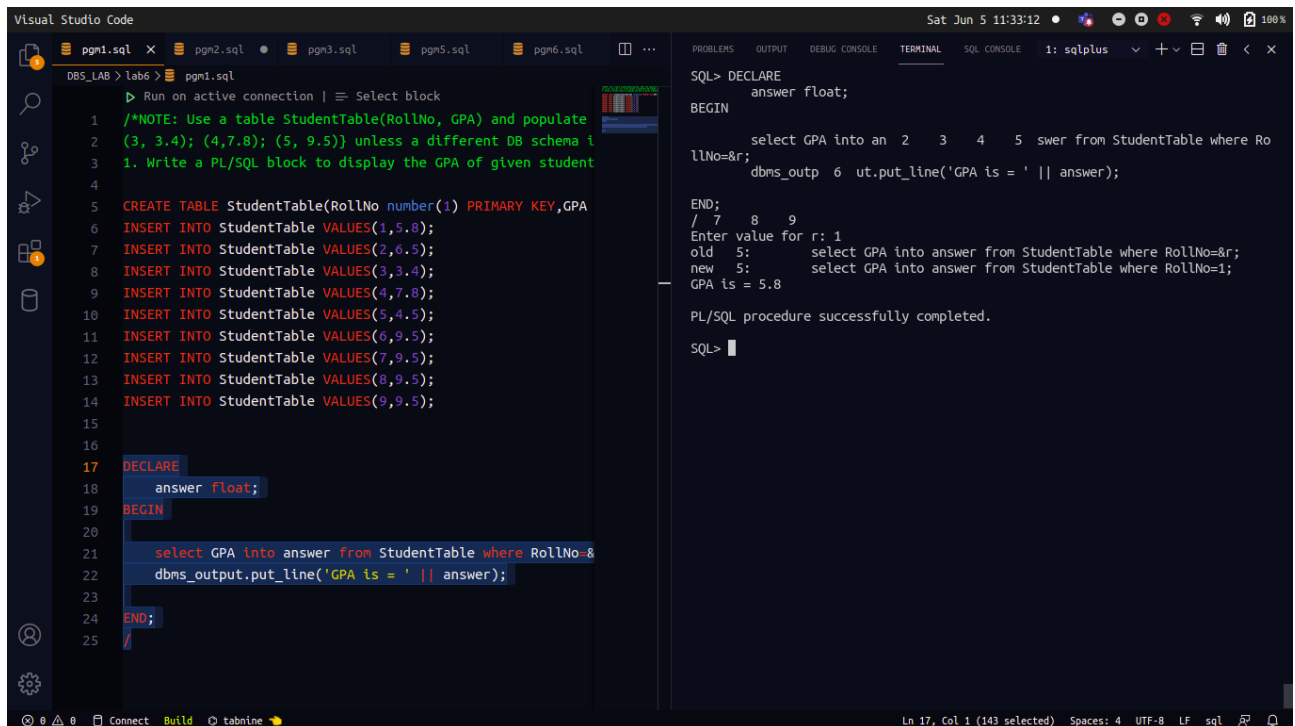
```
select GPA into answer from StudentTable where RollNo=&r;
```

```
dbms_output.put_line('GPA is = ' || answer);
```

```
END;
```

```
/
```

Output:



The screenshot shows the Visual Studio Code interface with a PL/SQL script in the editor and its execution output in the terminal. The script creates a table, inserts data, and declares a procedure to display GPA for a given student. The terminal output shows the execution of the script, including the creation of the table, insertion of data, and the execution of the procedure, which displays the GPA for a given student.

```
Visual Studio Code
pgm1.sql x pgm2.sql pgm3.sql pgm5.sql pgm6.sql
DB5_LAB > lab6 > pgm1.sql
Run on active connection | Select block
1 /*NOTE: Use a table StudentTable(RollNo, GPA) and populate
2 (3, 3.4); (4,7.8); (5, 9.5)} unless a different DB schema i
3 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=&r;
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>
```

2)

```
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 a Visual Studio Code editor with a PL/SQL script in the left pane and its execution output in the right pane (SQL console).

Left Pane (pgm2.sql):

```

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=8
27
28     if (result > 9) then grade:='A+';
29     elsif (result > 8) then grade:='A';

```

Right Pane (SQL console):

```

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>

```

3)

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 interface with a PL/SQL script in the editor and its execution output in the SQL console.

Editor (pgm3.sql):

```

12  */
13  DECLARE
14      myDate date;
15      ReturnDate date;
16      days integer;
17
18  BEGIN
19      myDate := '&i';
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  /

```

SQL Console:

```

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>

```

4)

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;

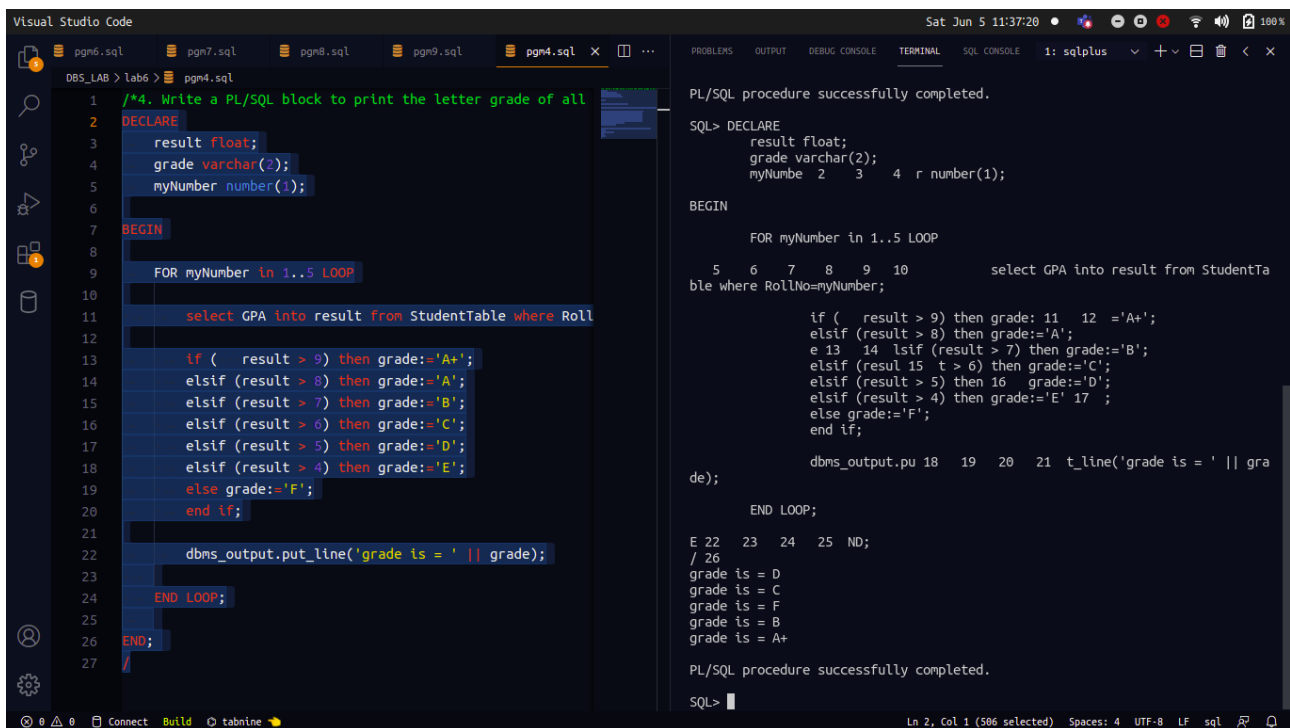
dbms_output.put_line('grade is ' || grade);

END LOOP;

END;

/

Output:



The screenshot shows the Visual Studio Code interface with a PL/SQL script in the editor and its execution output in the terminal. The script is a PL/SQL procedure that calculates letter grades based on GPA values. The terminal output shows the successful completion of the procedure and the resulting grades for various GPA values.

```
1  /*4. Write a PL/SQL block to print the letter grade of all
2  DECLARE
3      result float;
4      grade varchar(2);
5      myNumber number(1);
6
7  BEGIN
8
9      FOR myNumber in 1..5 LOOP
10
11         select GPA into result from StudentTable where Roll
12
13         if ( result > 9) then grade:='A+';
14         elsif (result > 8) then grade:='A';
15         elsif (result > 7) then grade:='B';
16         elsif (result > 6) then grade:='C';
17         elsif (result > 5) then grade:='D';
18         elsif (result > 4) then grade:='E';
19         else grade:='F';
20         end if;
21
22         dbms_output.put_line('grade is = ' || grade);
23
24     END LOOP;
25
26 END;
```

PL/SQL procedure successfully completed.

SQL> DECLARE
result float;
grade varchar(2);
myNumber 2 3 4 r number(1);

BEGIN

FOR myNumber in 1..5 LOOP

5 6 7 8 9 10 select GPA into result from StudentTable where RollNo=myNumber;

if (result > 9) then grade:='A+';
elsif (result > 8) then grade:='A';
e 13 14 l sif (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;

E 22 23 24 25 ND;
/ 26
grade is = D
grade is = C
grade is = F
grade is = B
grade is = A+

PL/SQL procedure successfully completed.

SQL>

5)

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

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.

```

```

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

```

```

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.

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>

```

6)

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;

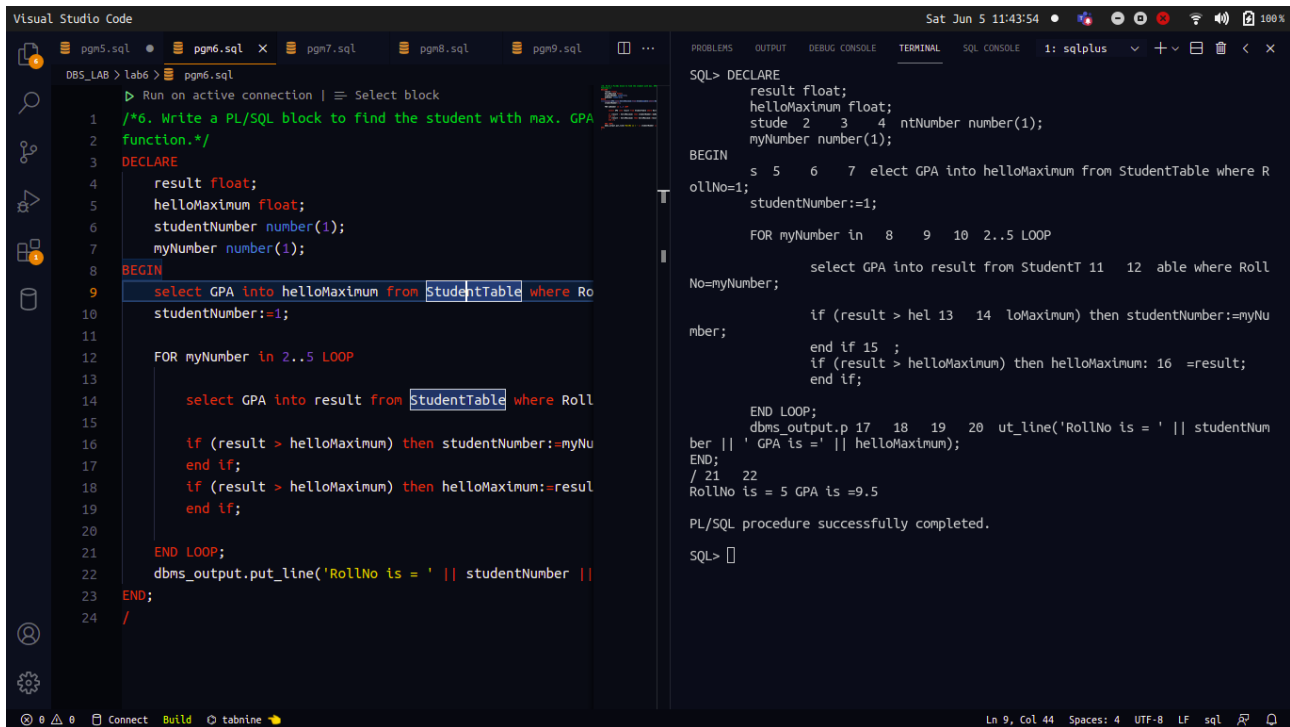
END LOOP;

dbms_output.put_line('RollNo is = ' || studentNumber || ' GPA is = ' || helloMaximum);

END;

/

Output:



The screenshot shows the Visual Studio Code interface with a PL/SQL procedure defined in a file named `pgm6.sql`. The procedure is designed to find the student with the maximum GPA. It uses a loop to iterate through GPA values from 2 to 5, selecting the corresponding student's roll number and GPA from the `StudentTable`. The output is displayed in the SQL console, showing the execution of the procedure and the resulting roll number and GPA.

```
SQL> 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;
        if (result > helloMaximum) then helloMaximum:=result;
        end if;
    END LOOP;
    dbms_output.put_line('RollNo is = ' || studentNumber || ' GPA is = ' || helloMaximum);
END;
/
```

PL/SQL procedure successfully completed.

RollNo is = 5 GPA is =9.5

7)

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;

```
elseif (result>=5 and result<6) then goto dd;  
elseif (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>>
```

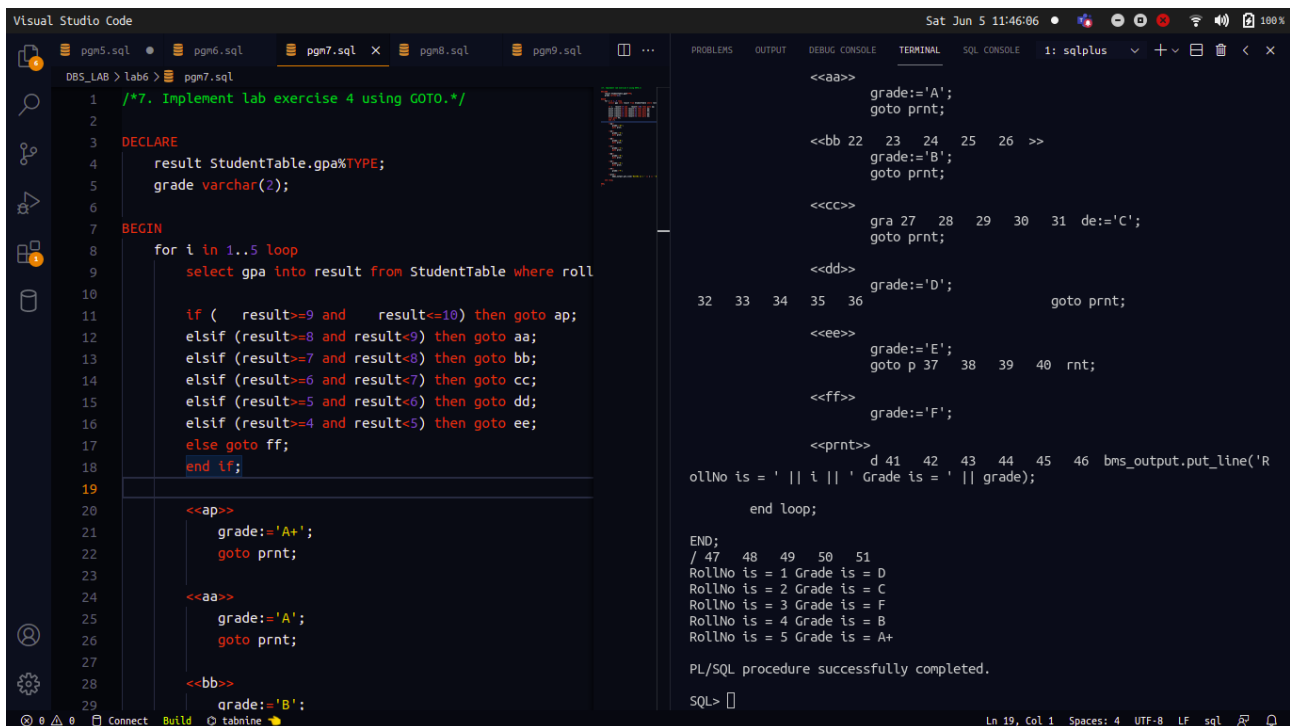
```
dbms_output.put_line('RollNo is = ' || i || ' Grade is = ' || grade);
```

```
end loop;
```

END;

/

Output:



```
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 print;
23
24         <<aa>>
25             grade:='A';
26             goto print;
27
28         <<bb>>
29             grade:='B';
```

```
<<aa>>
grade:='A';
goto print;

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

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

<<dd>>
32 33 34 35 36 grade:='D';
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>
```

8)

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 interface showing a PL/SQL procedure in a file named `pgm8.sql`. The procedure is designed to insert student records into a table based on instructor information.

The code in `pgm8.sql` is as follows:

```
DBS_LAB > lab6 > pgm8.sql
Run on active connection | Select block

/*8. Based on the University database schema, write a PL/SQL
of the Instructor whose name is supplied by the user. Use an
appropriate error message for the following cases:
a. Multiple instructors with the same name
b. No instructor for the given name*/
DECLARE
    name student.instructor.name%TYPE;
    val instructor%ROWTYPE;
BEGIN
    name := 'myNumber';
    select * into val from instructor where name = name;
    dbms_output.put_line(val);
END;
```

The output window on the right shows the successful execution of the procedure, displaying the number of rows inserted for each instructor:

```
PL/SQL procedure successfully completed.

SQL> DECLARE
    name student.instructor.name%TYPE;
    val in 2 3 instructor%ROWTYPE;
BEGIN
    name := 'myNumber';
    select * into val from instructor where name = 'myNumber';
    dbms_output.put_line('RollNo is ' || val.rollno || ' Grade is ' || val.grade);
end loop;
```

9)

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+';
```

```
elseif (result>=8 and result<9) then grade:='A';
```

```
elseif (result>=7 and result<8) then grade:='B';
```

```
elseif (result>=6 and result<7) then grade:='C';
```

```
elseif (result>=5 and result<6) then grade:='D';
```

```
elseif (result>=4 and result<5) then grade:='E';
```

```
elseif (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 PL/SQL procedure in the editor and its output in the terminal. The procedure is designed to validate GPA values and assign letter grades. It includes an exception handler for 'OutOfRangeException' and a general 'others' handler. The output shows the procedure completed successfully and a query result for the StudentTable.

```
Visual Studio Code
DBS_LAB > lab6 > pgm9.sql
1  /*9. Extend lab exercise 5 to validate the GPA value used t
2  the range, 0 - 10, display an error message, 'Out of Range'
3
4  DECLARE
5      OutOfRangeException Exception;
6      result StudentTable.gpa%TYPE;
7      grade StudentTable.lettergrade%TYPE;
8
9  BEGIN
10     for i in 1..5 loop
11         select gpa into result from StudentTable where Roll
12
13         if ( result>=9 and result<=10) then grade:='A+';
14         elsif (result>=8 and result<9) then grade:='A';
15         elsif (result>=7 and result<8) then grade:='B';
16         elsif (result>=6 and result<7) then grade:='C';
17         elsif (result>=5 and result<6) then grade:='D';
18         elsif (result>=4 and result<5) then grade:='E';
19         elsif (result>=0 and result<4) then grade:='F';
20         else RAISE OutOfRangeException;
21     end if;
22
23     update StudentTable set lettergrade=grade where Rol
24 end loop;
25
26 EXCEPTION
27     when OutOfRangeException then
28         dbms_output.put_line('GPA out of range');
29     when others then
30         dbms_output.put_line('Err 26 27 or');
31
32 END;
33 /
34
35 PL/SQL procedure successfully completed.
36
37 SQL> select * from StudentTable;
38
39      ROLLNO      GPA LE
40      -----
41      1          5.8 D
42      2          6.5 C
```