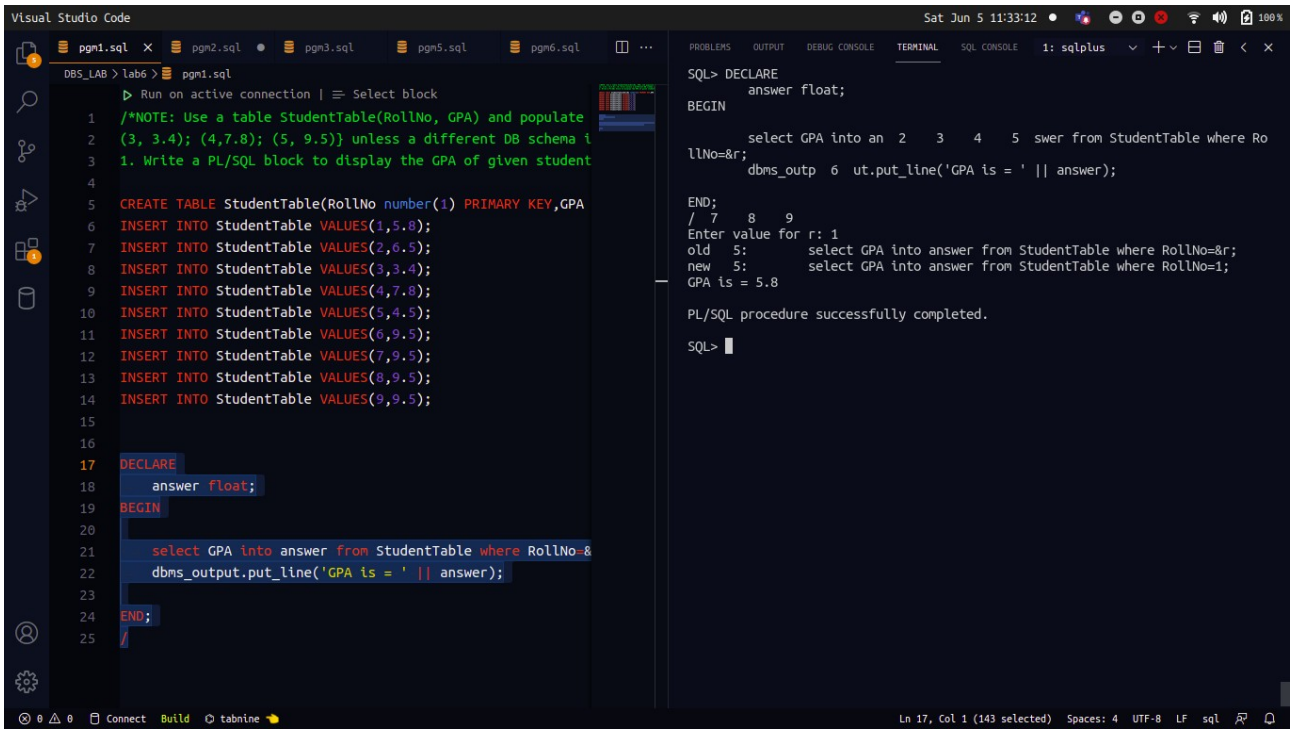
190905514 C SECTION 4th SEMESTER MOHAMMAD TOFIK

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 :

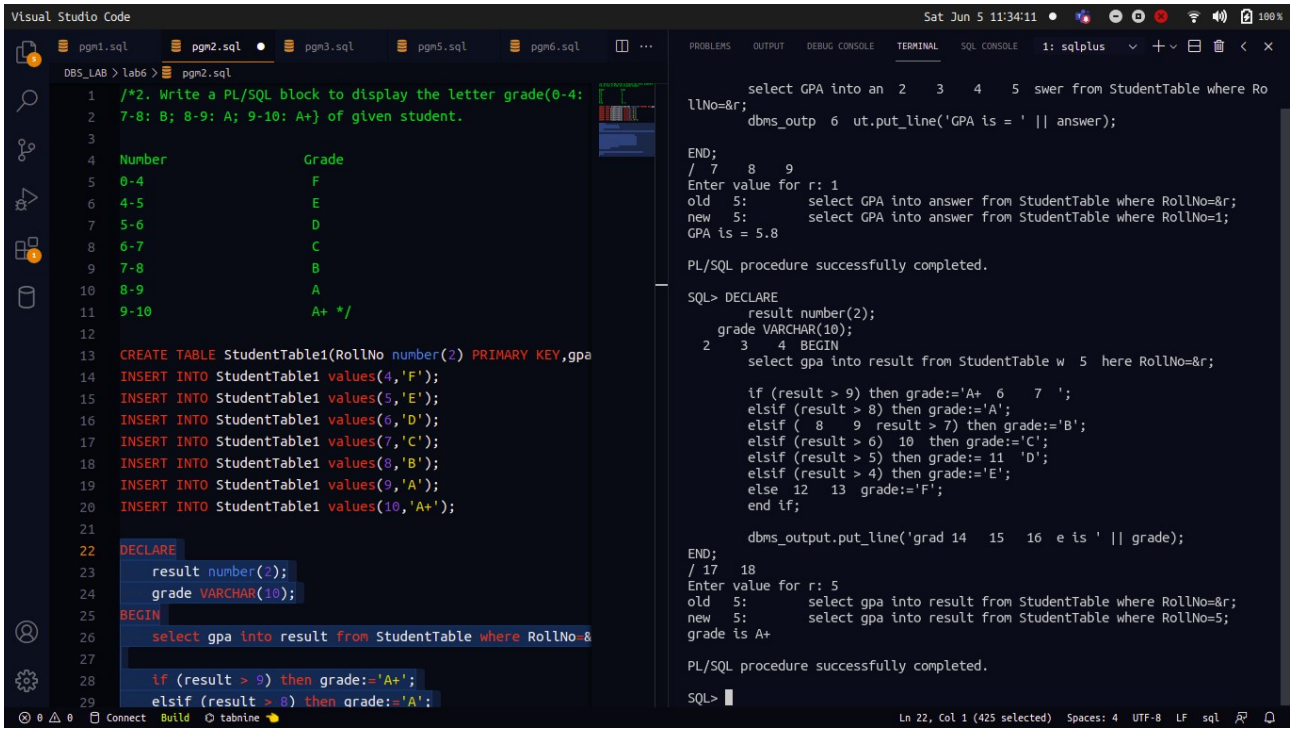



```
/*QUESTION-2. Write a PL/SQL block to display the letter grade(0-4: F; 4-5:
E; 5-6: D; 6-7: C;
7-8: B; 8-9: A; 9-10: A+} of given student.

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 :



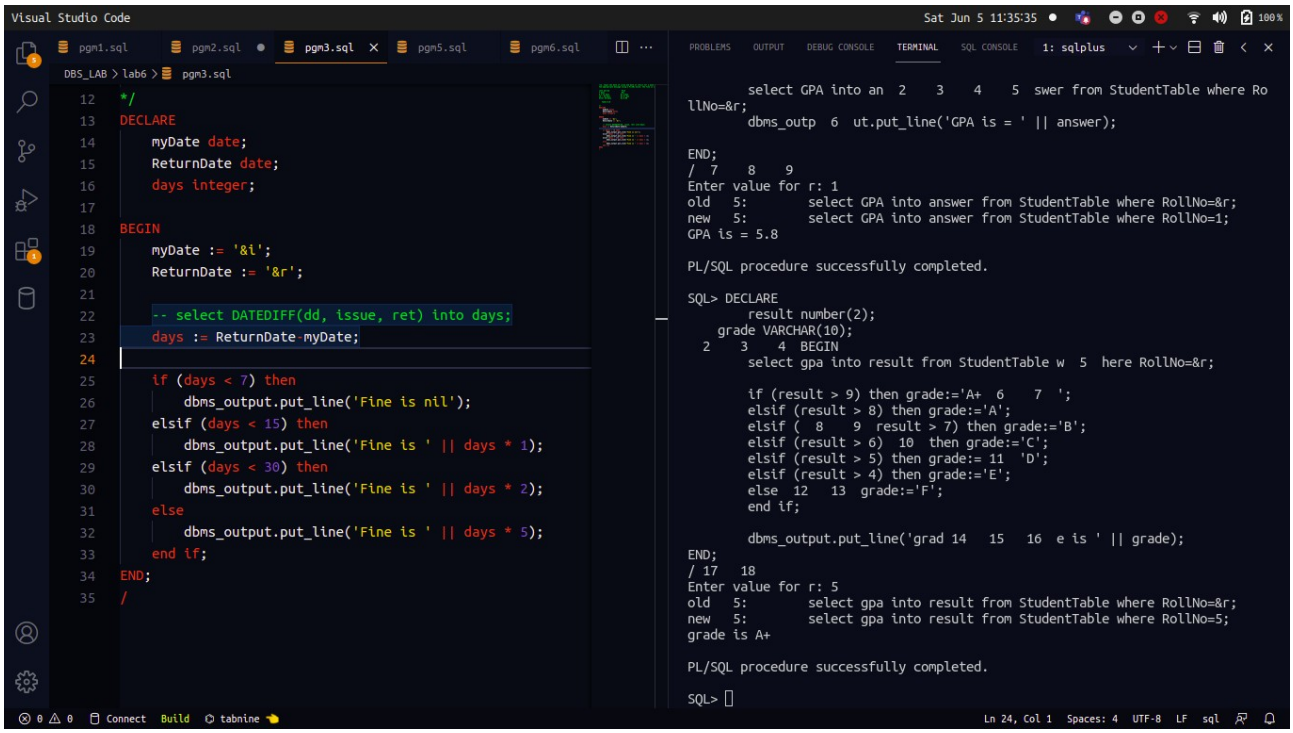
/*QUESTION-3. Input the date of issue and date of return for a book. Calculate and display the fine with
the appropriate message using a PL/SQL block. The fine is charged as per the table 8.1:

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

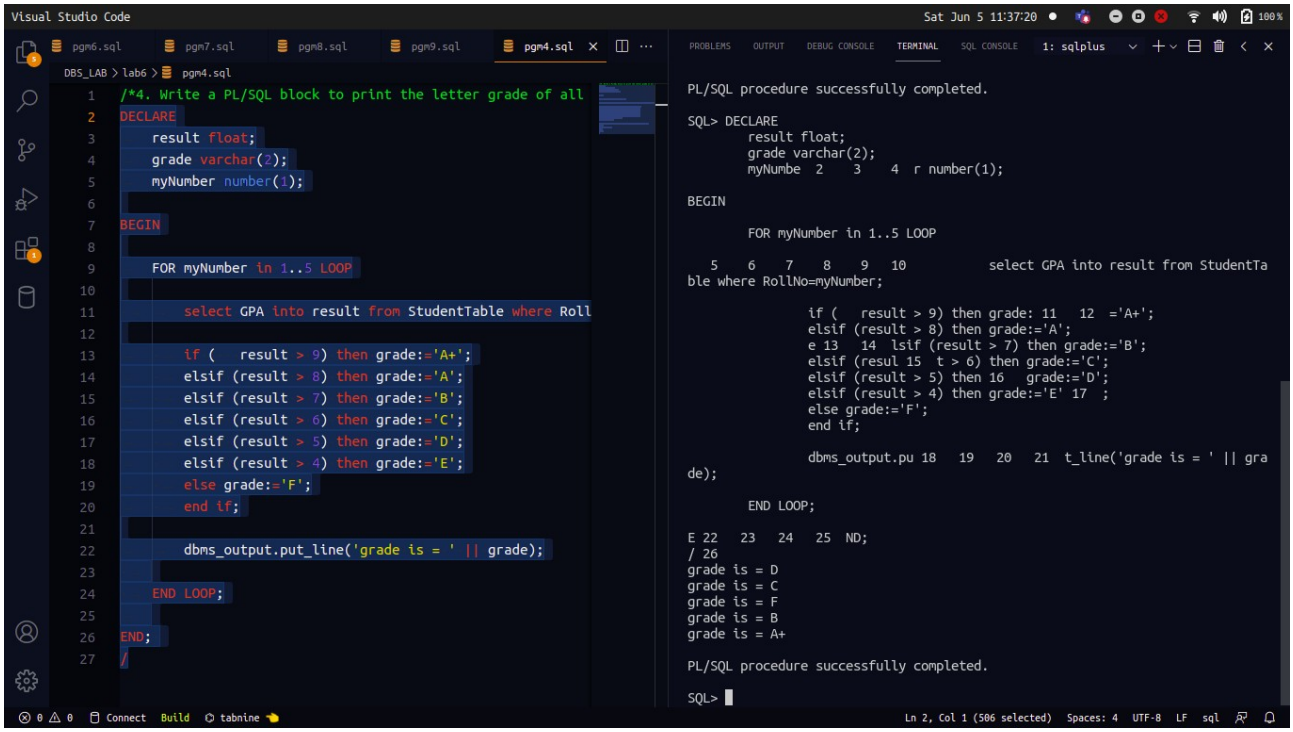
```
*/
DECLARE
myDate date;
ReturnDate date;
days integer;

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 :




```
/*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';
```

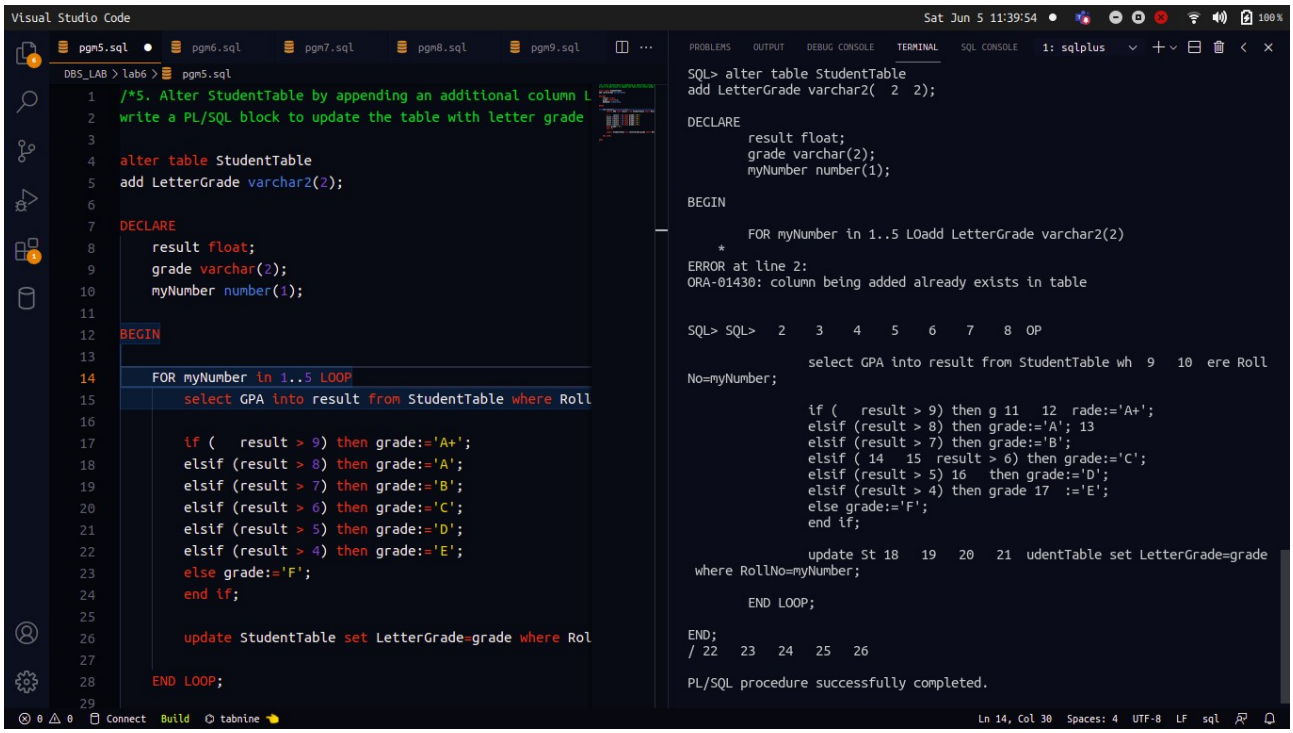
```
elsif (result > 4) then grade:='E';
else grade:='F';
end if;

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

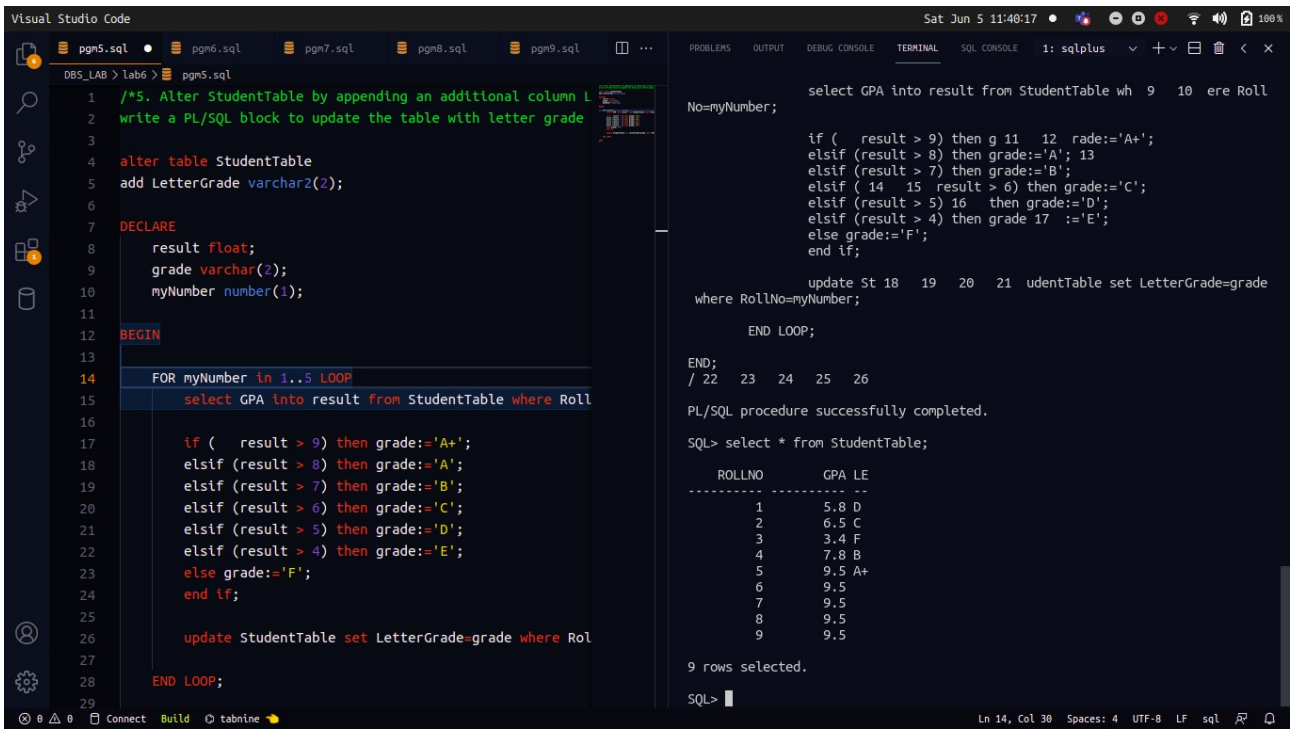



```
/*QUESTION-5. Alter StudentTable by appending an additional column
LetterGrade Varchar2(2). Then
write a PL/SQL block to update the table with letter grade of each
student.*/
alter table StudentTable
add LetterGrade varchar2(2);

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;
/
```

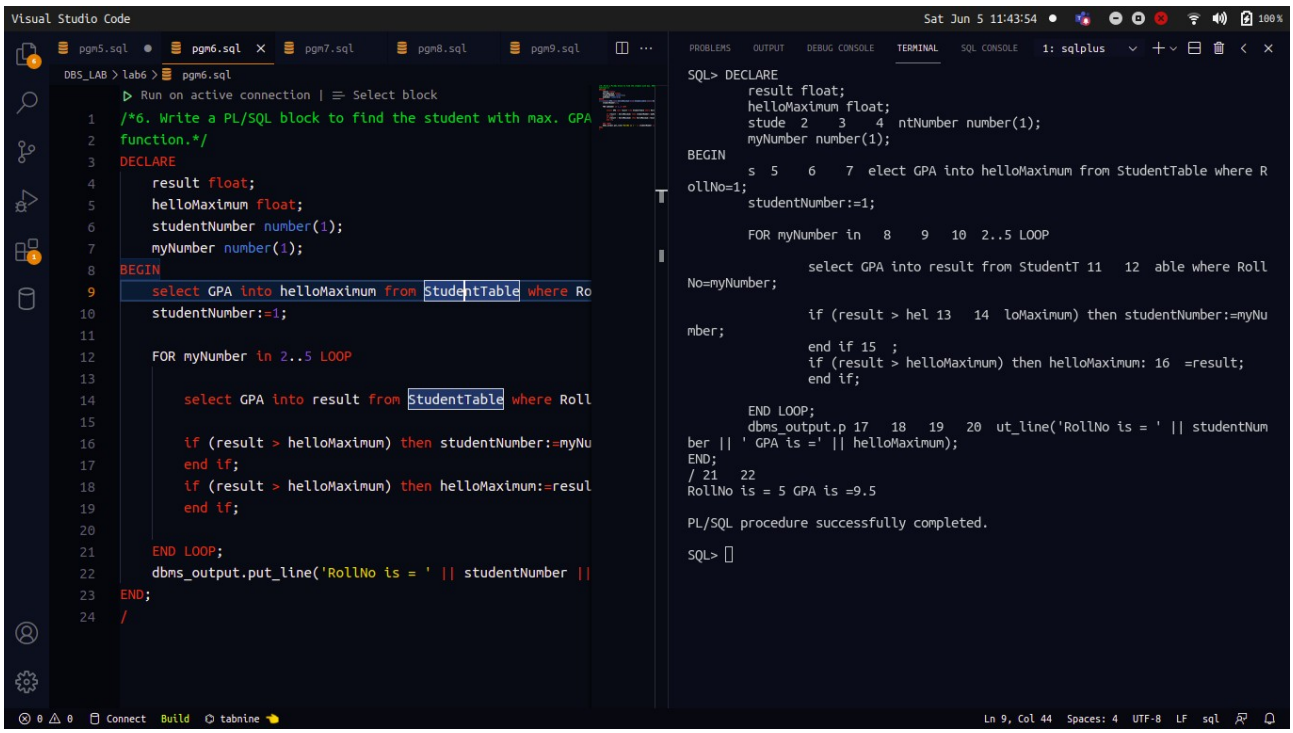
OUTPUT :






```
/*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;
/
```

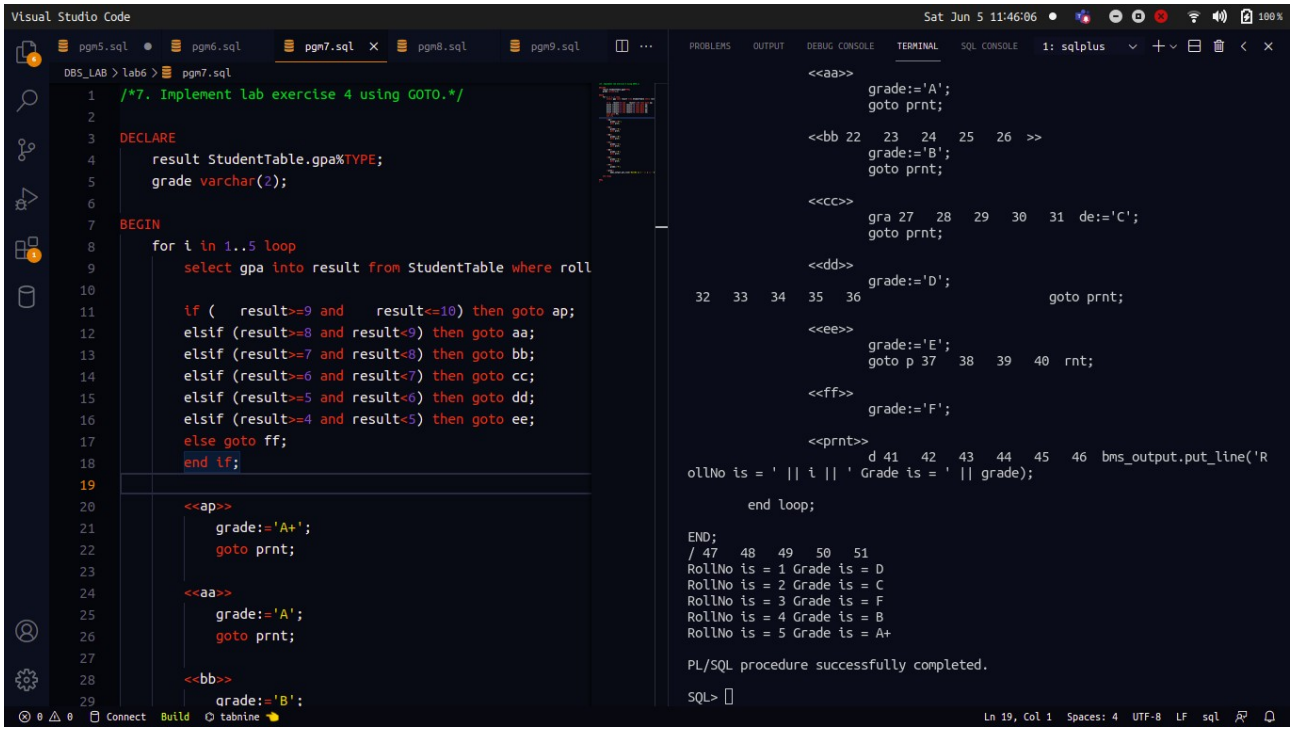
OUTPUT :




```
/*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;
/
```

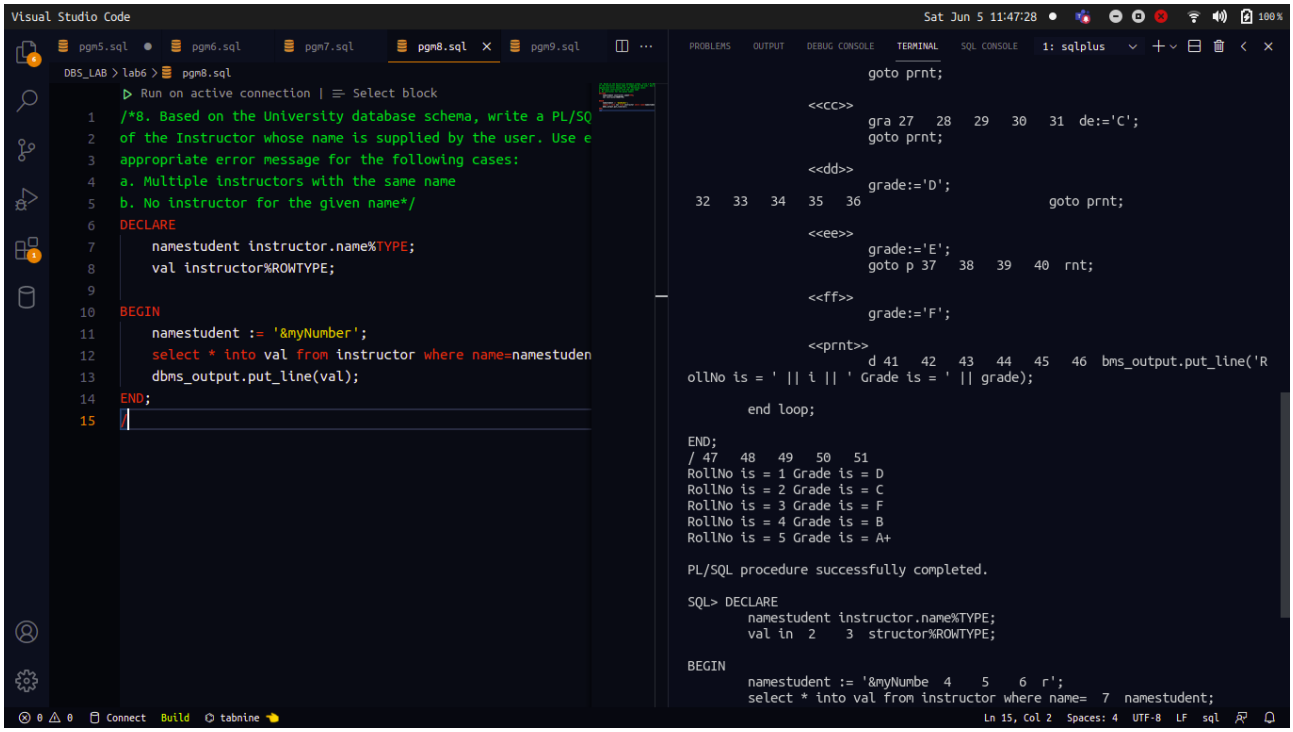
OUTPUT :



/*QUESTION-8. 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:
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 :




```
/*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 :