

Name → Laxman kumar Vashist

Class → BTech. 2<sup>nd</sup> Year { E }

University RollNo. → 191500429

Class RollNo. → '37'

Subject → Database Management  
Systems Lab

Subject Code → BCSC 0802

Subject Teacher → MRS. Gunjan  
Bharadwaj

**Qus 1** → Write a PL/SQL code block to compute the factorial of a number.

**CODE** →

=====

The screenshot shows an SQL Worksheet interface. At the top right, there are links for Feedback, Help, and a user profile for laxman.vashist\_cs19@gla.ac.in. Below the header, the title 'SQL Worksheet' is displayed. To the right of the title are buttons for Clear, Find, Actions (with a dropdown arrow), Save, and a Run button with a play icon. The main area contains a PL/SQL code block with the following text:

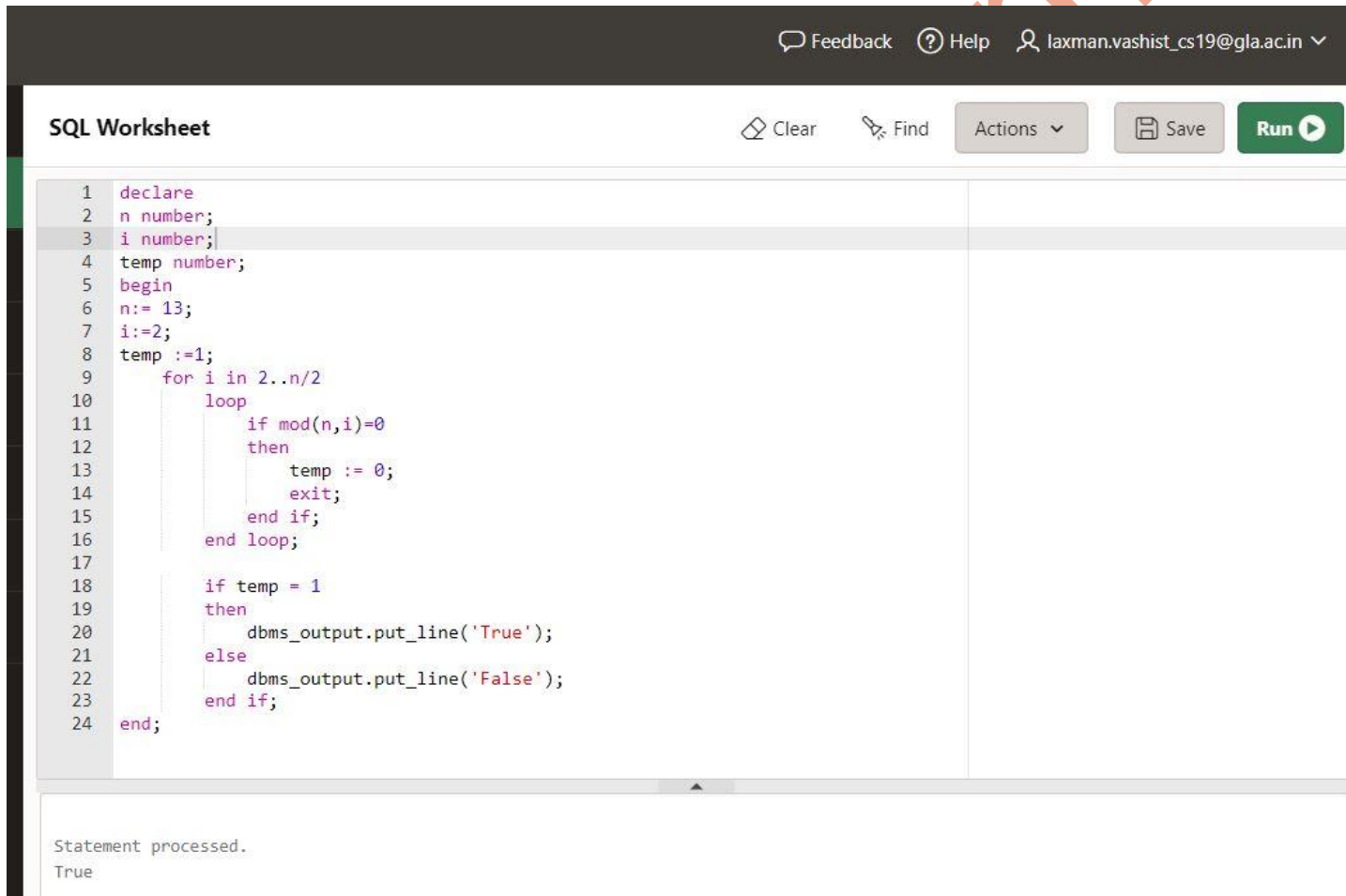
```
1 declare
2 fact number:=1;
3 n number:=5;
4 begin
5 while n>0 loop
6 fact:=n*fact;
7 n:=n-1;
8 end loop;
9 dbms_output.put_line(fact);|
10 end;
```

Below the code editor, the output area shows the message 'Statement processed.' followed by the result '120'.

**Qus 2** ➔ Write a PL/SQL code block to determine whether the number is prime or not.

**CODE** ➔

=====



The screenshot shows an SQL Worksheet interface with a dark header bar containing 'Feedback', 'Help', and a user profile 'laxman.vashist\_cs19@gla.ac.in'. Below the header, the 'SQL Worksheet' title is on the left, and 'Clear', 'Find', 'Actions', 'Save', and 'Run' buttons are on the right. The main area contains a PL/SQL code block with line numbers 1 through 24. The code declares variables 'n', 'i', and 'temp', sets 'n' to 13, and uses a loop to check for divisibility. It outputs 'True' because 13 is a prime number. At the bottom, a status bar indicates 'Statement processed.' and the output 'True'.

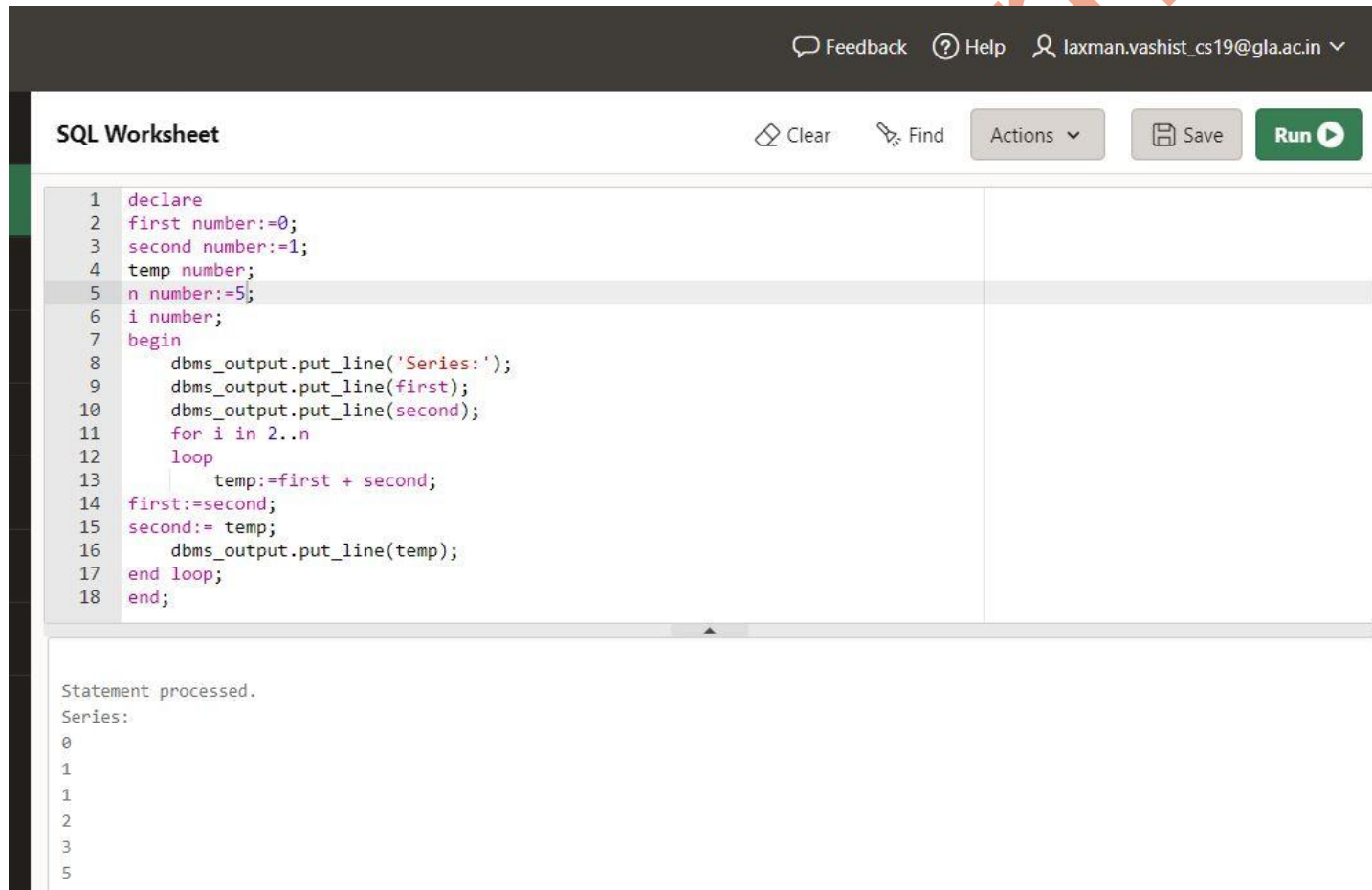
```
1 declare
2 n number;
3 i number;
4 temp number;
5 begin
6 n:= 13;
7 i:=2;
8 temp :=1;
9     for i in 2..n/2
10         loop
11             if mod(n,i)=0
12                 then
13                     temp := 0;
14                     exit;
15                 end if;
16         end loop;
17
18         if temp = 1
19             then
20                 dbms_output.put_line('True');
21             else
22                 dbms_output.put_line('False');
23             end if;
24 end;
```

Statement processed.  
True

**Qus 3** → Write a PL/SQL code block to display n terms of a fibonacci series.

**CODE** →

=====



The screenshot shows an SQL Worksheet interface with a dark header bar containing 'Feedback', 'Help', and a user profile 'laxman.vashist\_cs19@gla.ac.in'. Below the header, the 'SQL Worksheet' title is on the left, and 'Clear', 'Find', 'Actions', 'Save', and 'Run' buttons are on the right. The main area contains a PL/SQL code block with 18 lines of code. The code declares variables for the first and second numbers, a loop counter, and a temporary variable. It then uses a loop to calculate the Fibonacci series up to the specified number of terms. The output area at the bottom shows the message 'Statement processed.' followed by the generated series: 0, 1, 1, 2, 3, 5.

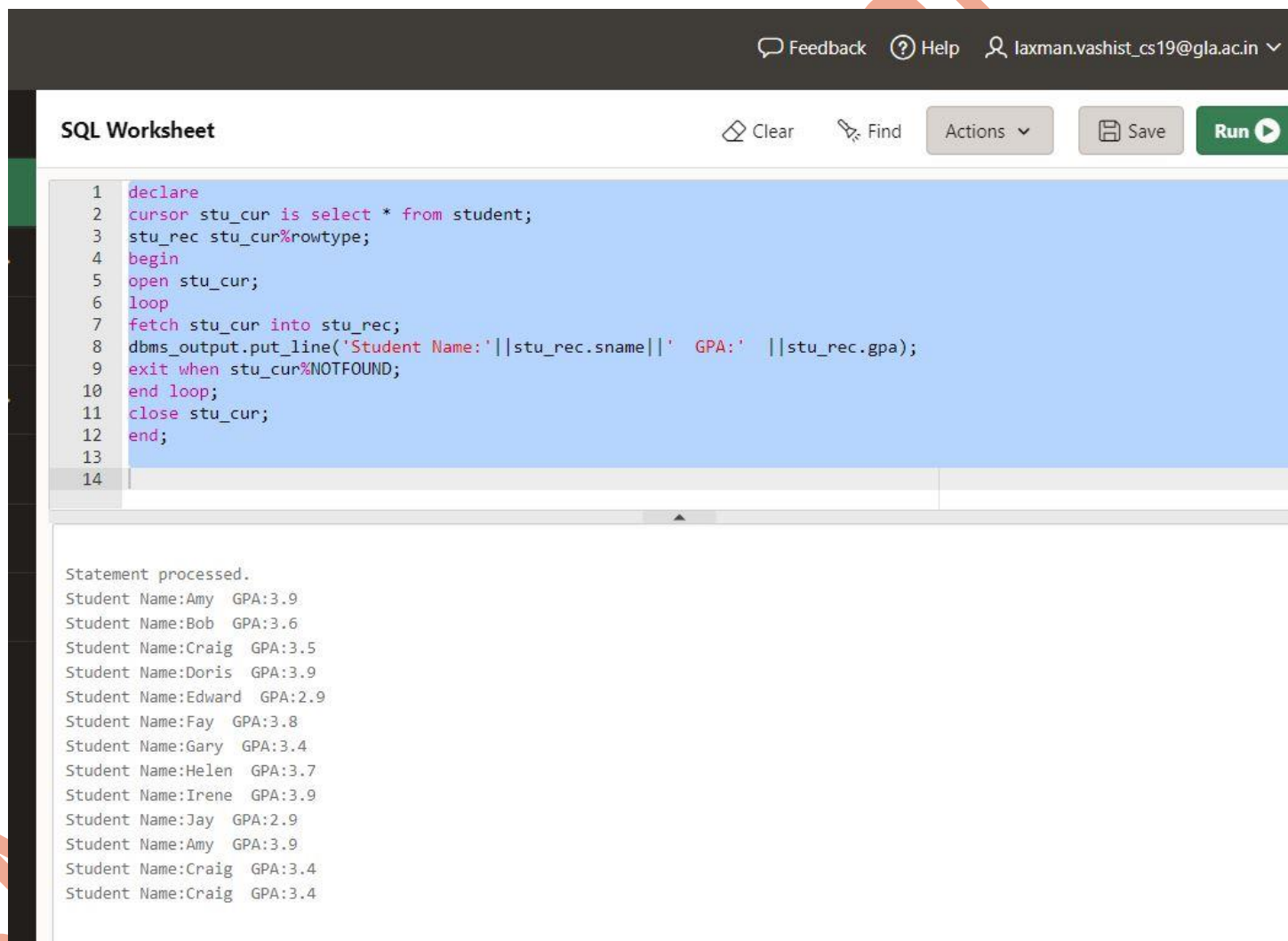
```
1 declare
2 first number:=0;
3 second number:=1;
4 temp number;
5 n number:=5;
6 i number;
7 begin
8     dbms_output.put_line('Series:');
9     dbms_output.put_line(first);
10    dbms_output.put_line(second);
11    for i in 2..n
12    loop
13        temp:=first + second;
14        first:=second;
15        second:= temp;
16        dbms_output.put_line(temp);
17    end loop;
18 end;
```

Statement processed.  
Series:  
0  
1  
1  
2  
3  
5

**Qus 4** → Write a PL/SQL code block to display the names and GPA of students from student table using an explicit cursor.

**CODE** →

=====



The screenshot shows an SQL Worksheet interface with a dark header bar containing 'Feedback', 'Help', and a user profile 'laxman.vashist\_cs19@gla.ac.in'. Below the header, the 'SQL Worksheet' title is on the left, and 'Clear', 'Find', 'Actions', 'Save', and 'Run' buttons are on the right. The main area contains PL/SQL code for an explicit cursor. The output pane at the bottom shows the statement was processed successfully and lists 15 rows of student data.

```
1 declare
2 cursor stu_cur is select * from student;
3 stu_rec stu_cur%rowtype;
4 begin
5 open stu_cur;
6 loop
7 fetch stu_cur into stu_rec;
8 dbms_output.put_line('Student Name: ' || stu_rec.sname || ' GPA: ' || stu_rec.gpa);
9 exit when stu_cur%NOTFOUND;
10 end loop;
11 close stu_cur;
12 end;
```

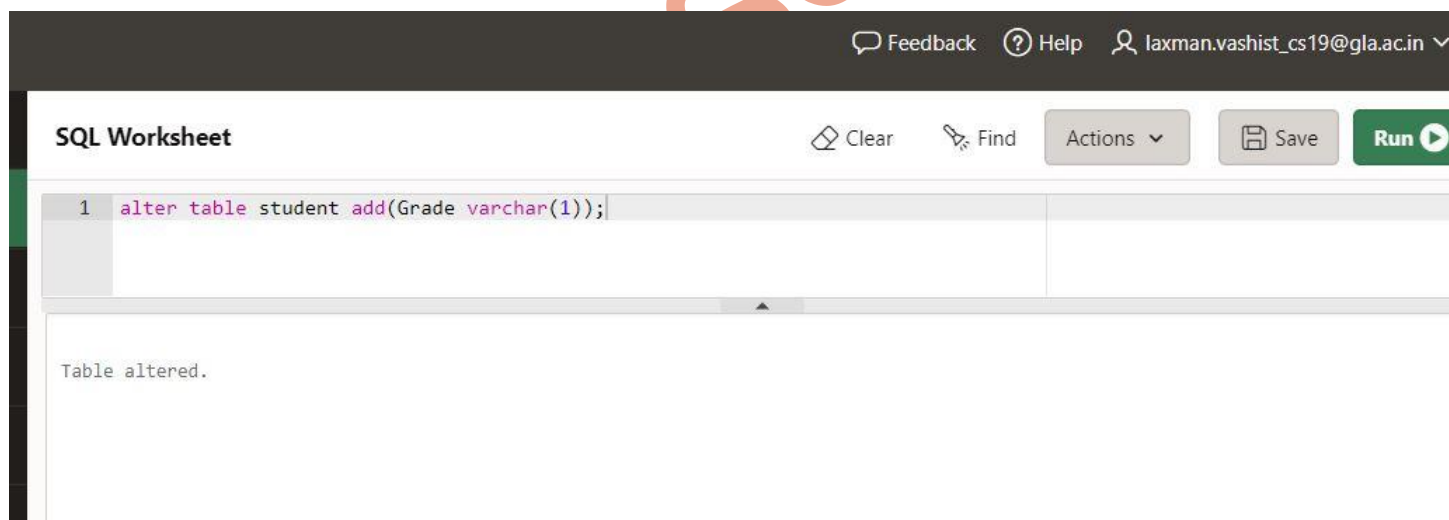
Statement processed.

Student Name:	Amy	GPA:	3.9
Student Name:	Bob	GPA:	3.6
Student Name:	Craig	GPA:	3.5
Student Name:	Doris	GPA:	3.9
Student Name:	Edward	GPA:	2.9
Student Name:	Fay	GPA:	3.8
Student Name:	Gary	GPA:	3.4
Student Name:	Helen	GPA:	3.7
Student Name:	Irene	GPA:	3.9
Student Name:	Jay	GPA:	2.9
Student Name:	Amy	GPA:	3.9
Student Name:	Craig	GPA:	3.4
Student Name:	Craig	GPA:	3.4

**Qus 5** ➔ Write a PL/SQL code block that displays the names, GPA of students along with the grades of students after calculation from student table using an explicit cursor.  
Add a column grade to the student table; update the grades of students to the table after calculation. (The criteria of grade can be considered as grade = A if gpa>3.7; and grade = B, otherwise).

**CODE** ➔

=====



The screenshot shows a web-based SQL Worksheet interface. At the top, there is a navigation bar with links for 'Feedback', 'Help', and a user profile 'laxman.vashist\_cs19@gla.ac.in'. Below this, the 'SQL Worksheet' title is displayed. To the right of the title are buttons for 'Clear', 'Find', 'Actions' (with a dropdown arrow), 'Save', and a green 'Run' button with a play icon. The main text area contains a single line of SQL code: `1 alter table student add(Grade varchar(1));`. Below the code editor, a message box displays the text 'Table altered.'.