

Name → Laxman kumar Vashist

Class → BTech. 2nd Year { E }

University RollNo. → 191500429

Class RollNo. → '37'

Subject → Database Management
Systems Lab

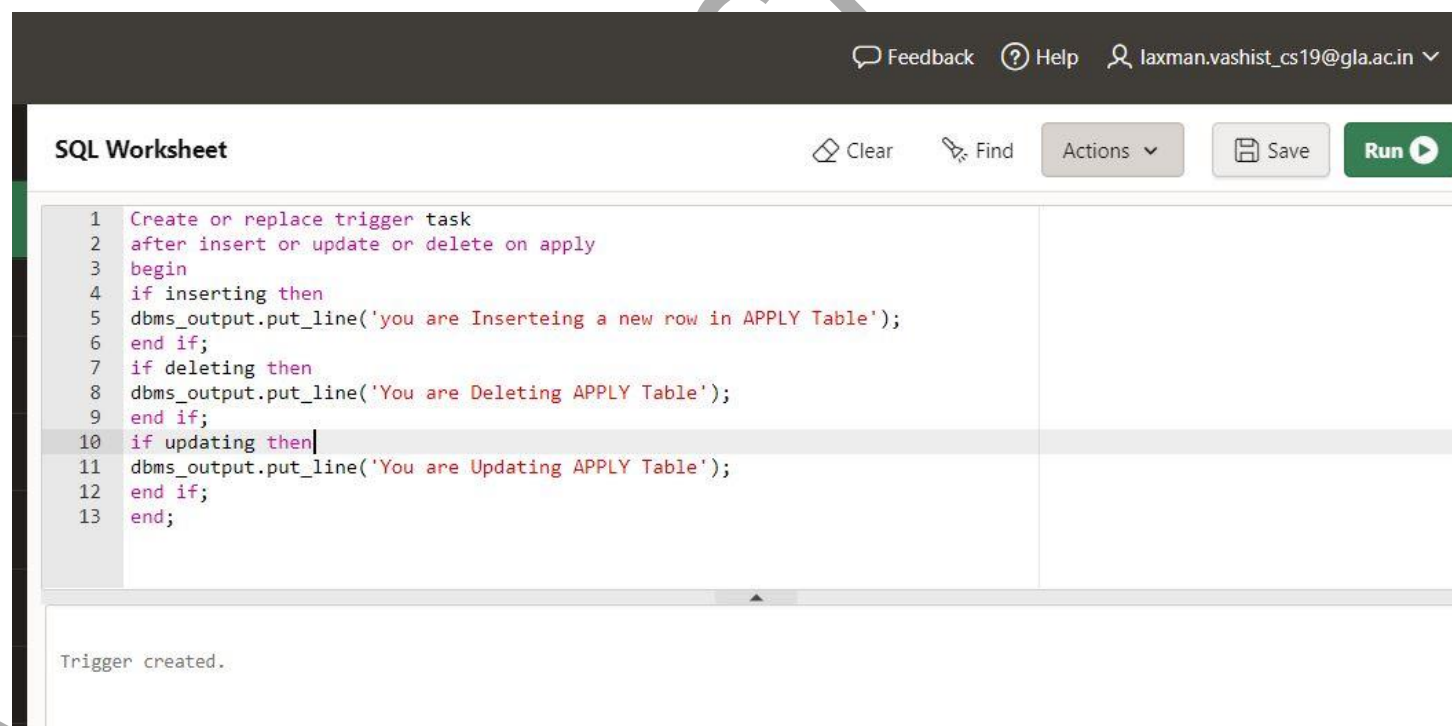
Subject Code → BCSC 0802

Subject Teacher → MRS. Gunjan
Bharadwaj

Qus 1 → Create trigger TASK that will display which DML operation we are performing on Apply Table i.e. if we are running Update on Apply then display 'You are Updating APPLY Table', on deletion display 'You are Deleting APPLY Table', on inserting 'You are Inserting a new row in APPLY Table'. Hint: create trigger TASK on all three DML statements use IF inserting

CODE →

=====



The screenshot shows an SQL Worksheet interface. At the top, there is a header bar with '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 SQL script for creating a trigger named 'TASK' on the 'APPLY' table. The script uses 'IF' statements to handle 'inserting', 'deleting', and 'updating' operations, each displaying a message via 'dbms_output.put_line'. The script is as follows:

```
1 Create or replace trigger task
2 after insert or update or delete on apply
3 begin
4 if inserting then
5 dbms_output.put_line('you are Inserteing a new row in APPLY Table');
6 end if;
7 if deleting then
8 dbms_output.put_line('You are Deleting APPLY Table');
9 end if;
10 if updating then
11 dbms_output.put_line('You are Updating APPLY Table');
12 end if;
13 end;
```

Below the script, the output area shows the message 'Trigger created.'

Qus 2 → Create a trigger that will delete the student data from apply table upon deletion from student table. Hint: We want if you delete any student from STUDENT Table then applications with same SID in APPLY Table will be deleted. i.e. if we run query delete from student where sid=123 then all application of sid 123 should be deleted from APPLY Table.

CODE →

=====

Feedback ? Help laxman.vashist_cs19@gla.ac.in

SQL Worksheet Clear Find Actions Save Run

```
1 create or replace trigger stu_replica
2 after delete on student
3 for each row
4 begin
5 delete from apply where sid=:old.sid;
6 end;
7 Select * from apply;
8 select * from student;
9
10
11
12 delete from student where sid=123;
```

SID	CNAME	MAJOR	DECISION
123	Stanford	CS	Y
123	Stanford	EE	N
123	Berkeley	CS	Y
123	Cornell	EE	Y

Qus 3 → Create a trigger that will insert all deleted rows from table student to table backup student. Hint use BEFORE DELETE

CODE →

=====

Feedback Help laxman.vashist_cs19@gla.ac.in

SQL Worksheet Clear Find Actions Save Run

```
1 create table backup_ student(sid int,sName varchar2(10), GPA real,sizeHS int);
2 create or replace trigger asqw
3 before delete on student
4 for each row
5 begin
6 insert into backup_student values(:new.sid,:new.sName,:new.GPA,:new.sizeHS);
7 end;
8 select * from backup_student;
9 delete student where did=123;
```

ORA-00922: missing or invalid option

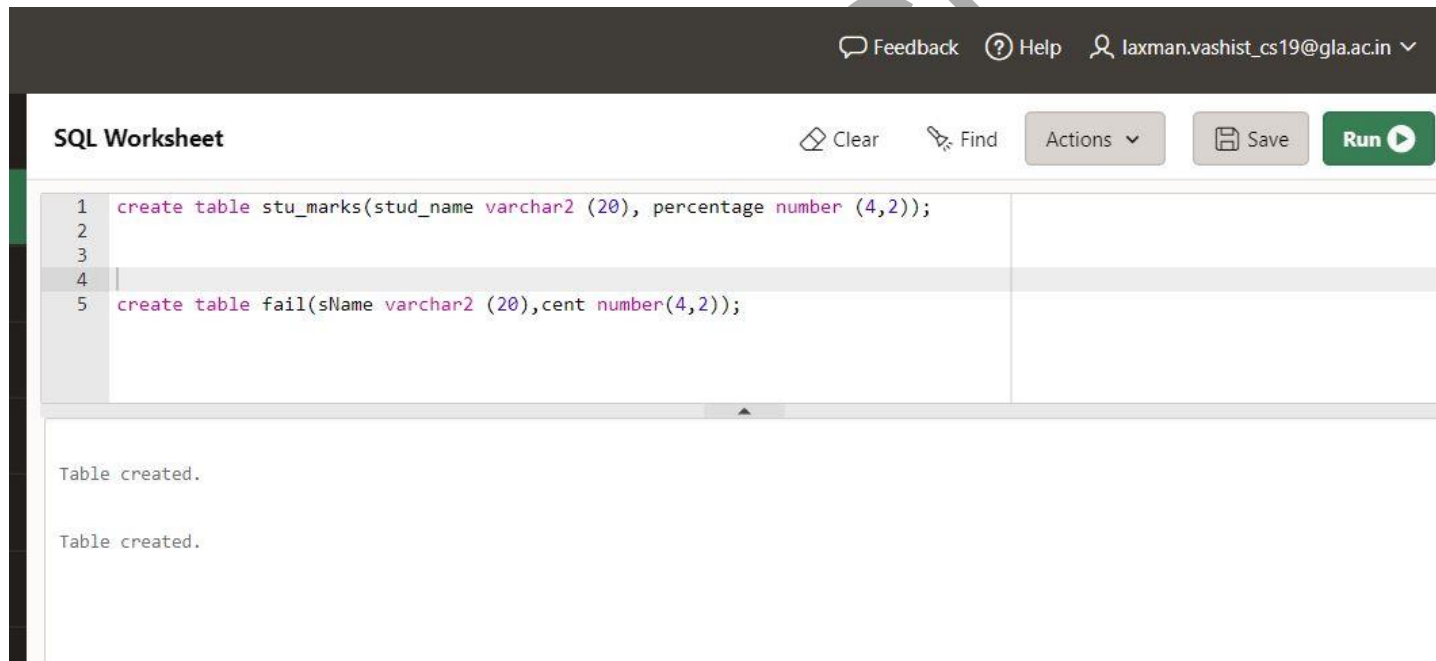
Errors: TRIGGER ASQW
Line/Col: 4/1 PLS-00103: Encountered the symbol "SELECT"

Qus 4 → Create a table STU_MARKS as given below:

A.... Create table FAIL having two columns STUD_NAME and cent for percentage.

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 title 'SQL Worksheet' is on the left, and 'Clear', 'Find', 'Actions', 'Save', and 'Run' buttons are on the right. The main area contains two SQL statements: 'create table stu_marks(stud_name varchar2 (20), percentage number (4,2));' and 'create table fail(sName varchar2 (20),cent number(4,2));'. The output area at the bottom shows 'Table created.' for both statements.

```
1 create table stu_marks(stud_name varchar2 (20), percentage number (4,2));
2
3
4
5 create table fail(sName varchar2 (20),cent number(4,2));
```

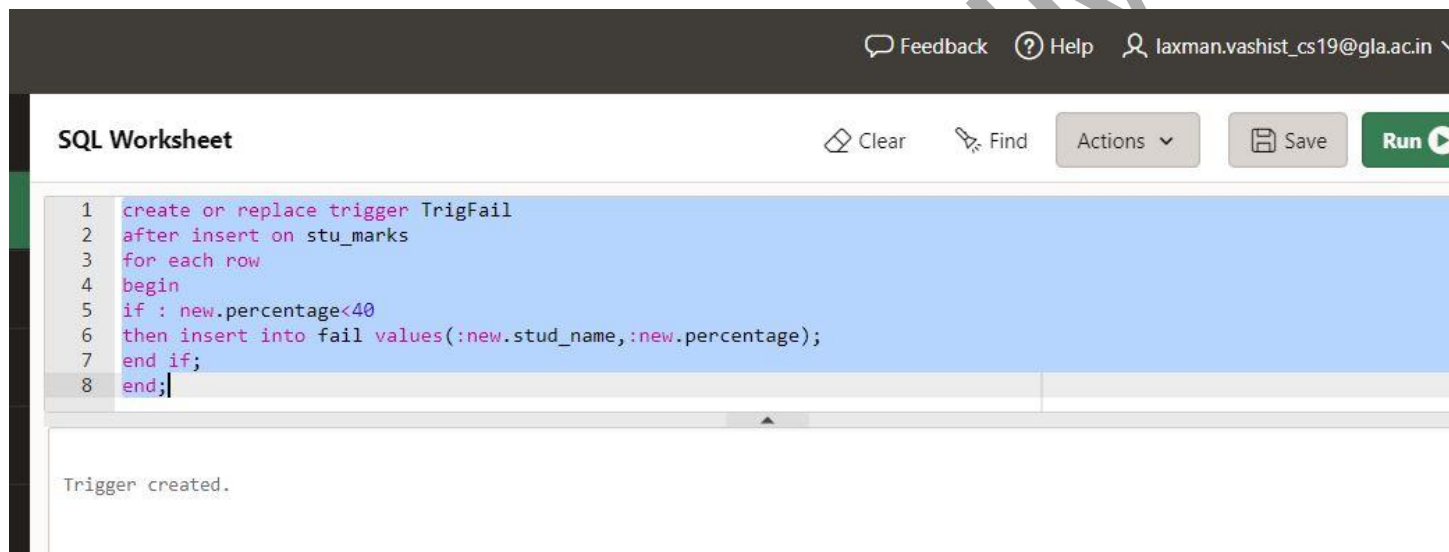
Table created.

Table created.

B>>> Create Trigger TrigFail such that If percentage is less than 40 it will copy name and percentage to FAIL table.

CODE →

=====



The screenshot shows an SQL Worksheet interface. At the top, there is a header bar with 'Feedback', 'Help', and a user profile 'laxman.vashist_cs19@gla.ac.in'. Below the header, the worksheet title is 'SQL Worksheet'. To the right of the title are buttons for 'Clear', 'Find', 'Actions', 'Save', and a green 'Run' button. The main area contains an SQL query:

```
1 create or replace trigger TrigFail
2 after insert on stu_marks
3 for each row
4 begin
5 if : new.percentage<40
6 then insert into fail values(:new.stud_name,:new.percentage);
7 end if;
8 end;
```

 Below the query, the execution result is displayed: 'Trigger created.'

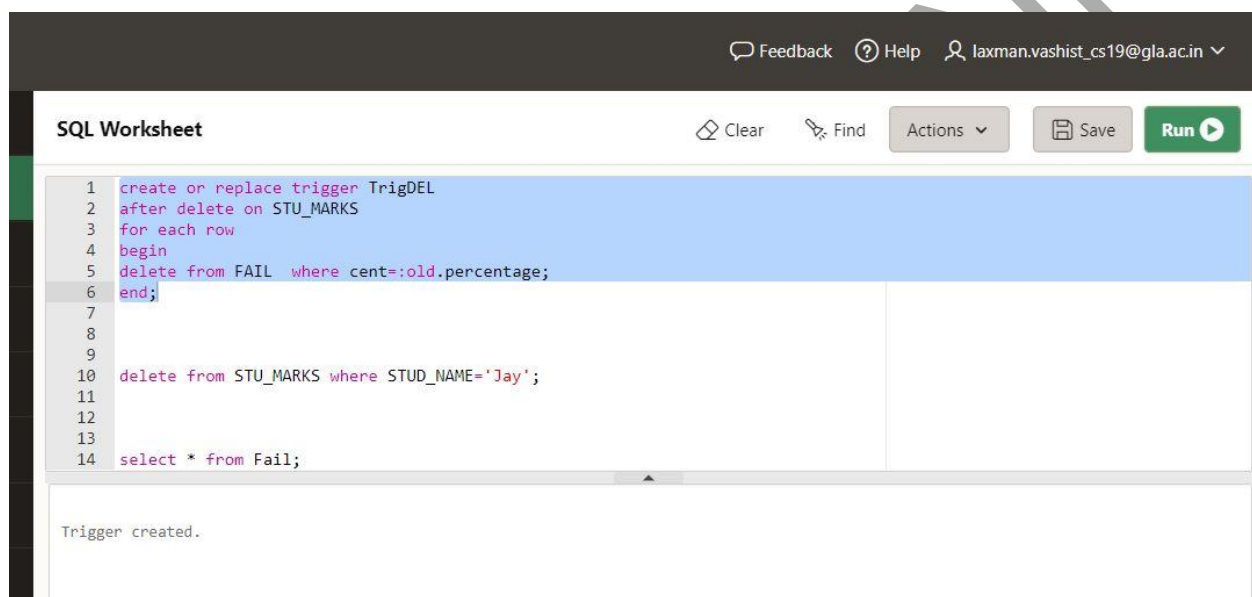
```
1 create or replace trigger TrigFail
2 after insert on stu_marks
3 for each row
4 begin
5 if : new.percentage<40
6 then insert into fail values(:new.stud_name,:new.percentage);
7 end if;
8 end;
```

Trigger created.

C>>> Create Trigger TrigDEL When deletion is performed in Stu_marks table then check whether the students name exists in FAIL table or not, if exists it will delete corresponding record from the table FAIL as well.

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 SQL script with line numbers 1 through 14. The script defines a trigger 'TrigDEL' that fires after a delete on the 'STU_MARKS' table. It checks if the student's name exists in the 'FAIL' table and deletes the record if it does. After the script, the output shows 'Trigger created.'

```
1 create or replace trigger TrigDEL
2 after delete on STU_MARKS
3 for each row
4 begin
5 delete from FAIL where cent=:old.percentage;
6 end;
7
8
9
10 delete from STU_MARKS where STUD_NAME='Jay';
11
12
13
14 select * from Fail;
```

Trigger created.