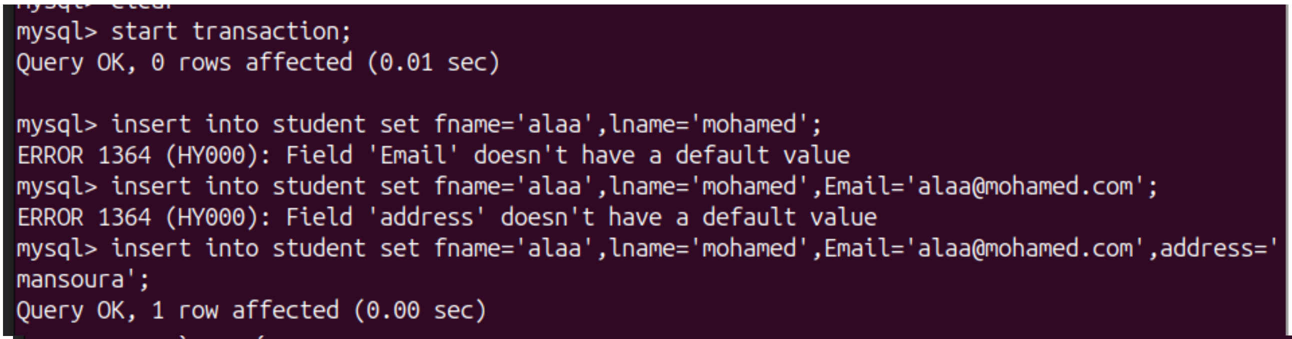
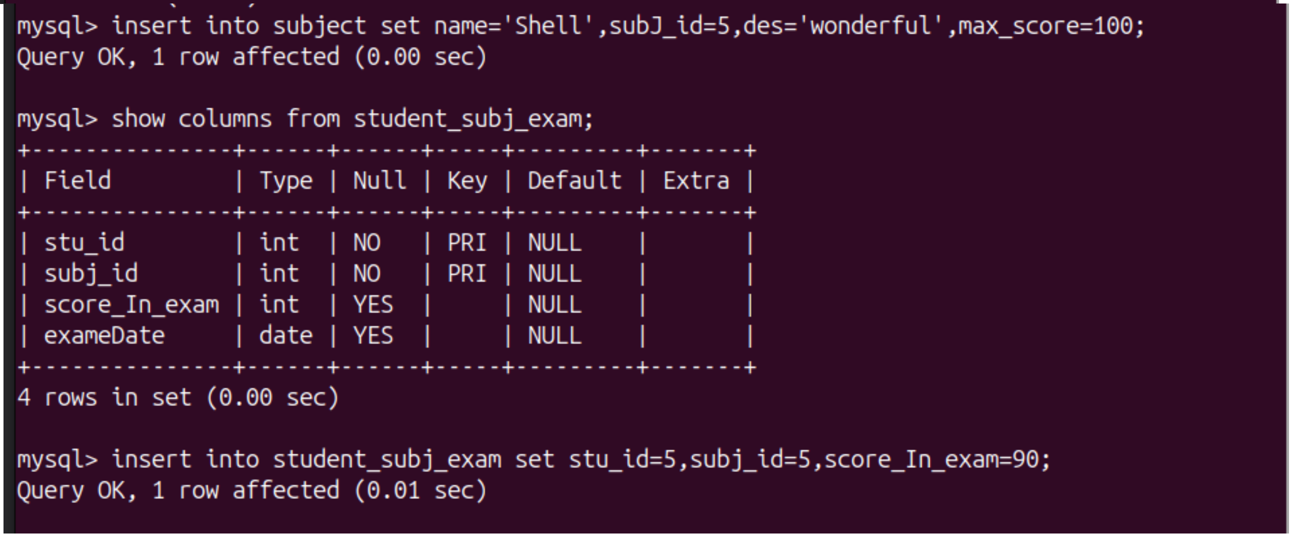
Lap3

1. Insert new student and his score in exam in different subjects as transaction

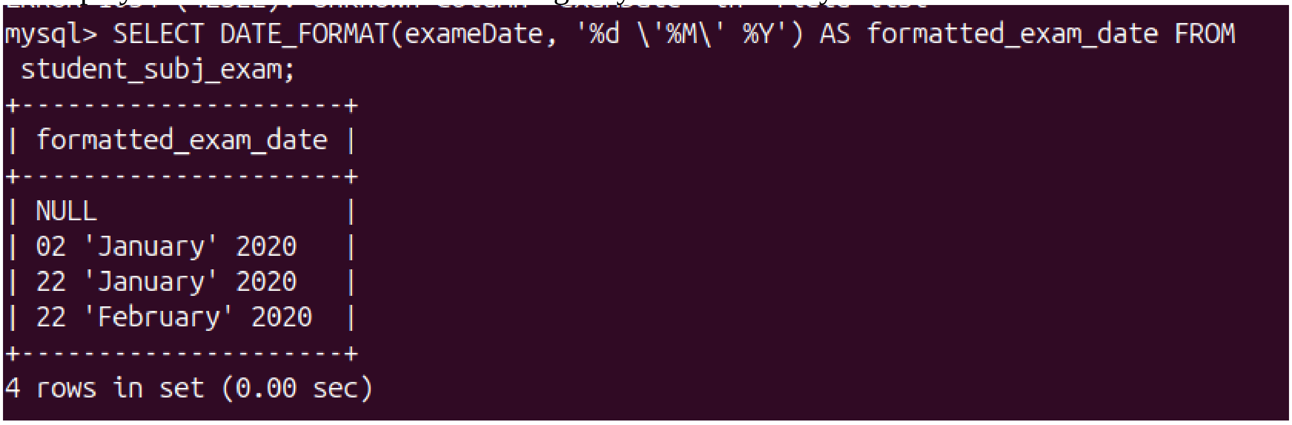


```
mysql> start transaction;
Query OK, 0 rows affected (0.01 sec)

mysql> insert into student set fname='alaa',lname='mohamed';
ERROR 1364 (HY000): Field 'Email' doesn't have a default value
mysql> insert into student set fname='alaa',lname='mohamed',Email='alaa@mohamed.com';
ERROR 1364 (HY000): Field 'address' doesn't have a default value
mysql> insert into student set fname='alaa',lname='mohamed',Email='alaa@mohamed.com',address='
mansoura';
Query OK, 1 row affected (0.00 sec)
```



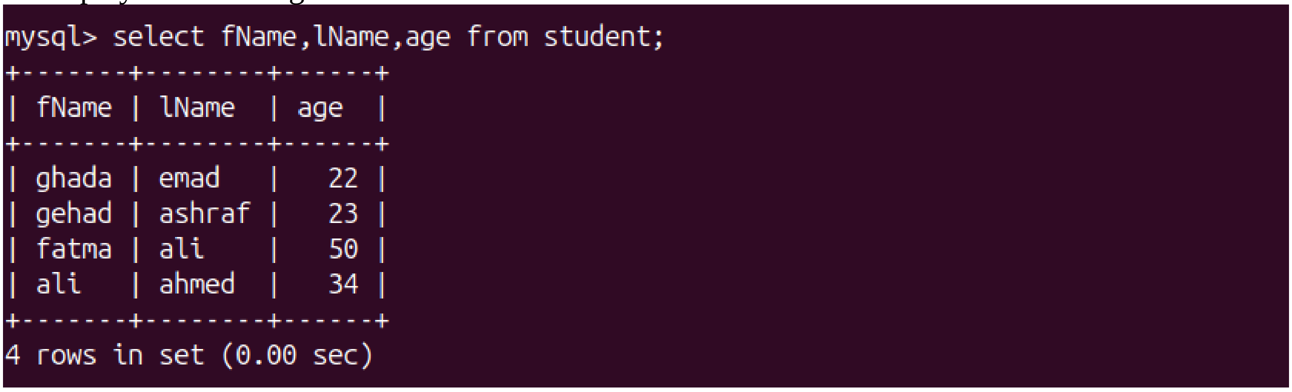
```
mysql> insert into subject set name='Shell',subJ_id=5,des='wonderful',max_score=100;
Query OK, 1 row affected (0.00 sec)

mysql> show columns from student_subj_exam;
+---------------+------+------+-----+---------+-------+
| Field         | Type | Null | Key | Default | Extra |
+---------------+------+------+-----+---------+-------+
| stu_id        | int  | NO   | PRI | NULL    |       |
| subj_id       | int  | NO   | PRI | NULL    |       |
| score_In_exam | int  | YES  |     | NULL    |       |
| exameDate     | date | YES  |     | NULL    |       |
+---------------+------+------+-----+---------+-------+
4 rows in set (0.00 sec)

mysql> insert into student_subj_exam set stu_id=5,subj_id=5,score_In_exam=90;
Query OK, 1 row affected (0.01 sec)
```

2. Display the date of exam as the following: day 'month name' year.



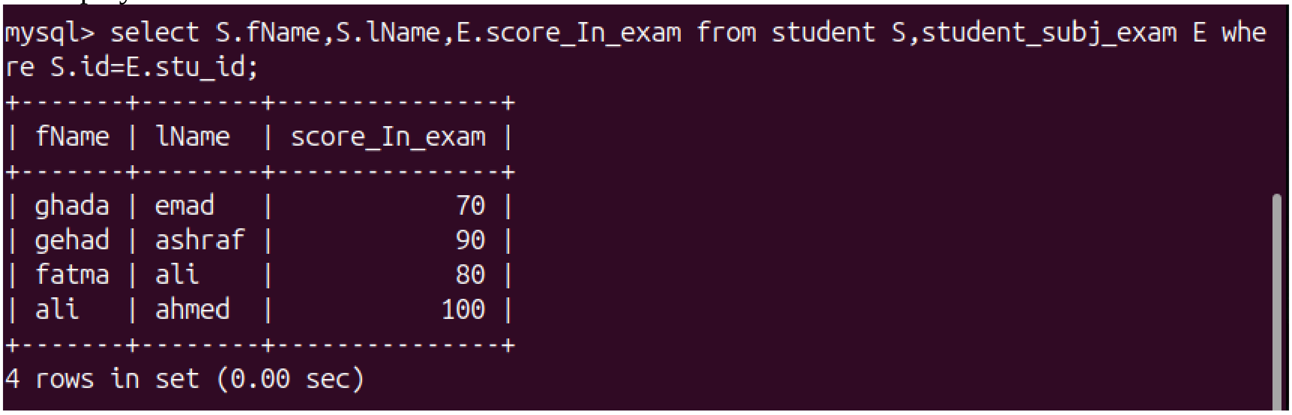
```
mysql> SELECT DATE_FORMAT(exameDate, '%d \'%M\' %Y') AS formatted_exam_date FROM
 student_subj_exam;
+---------------------+
| formatted_exam_date |
+---------------------+
| NULL                |
| 02 'January' 2020   |
| 22 'January' 2020   |
| 22 'February' 2020  |
+---------------------+
4 rows in set (0.00 sec)
```

3. Display name and age of each students



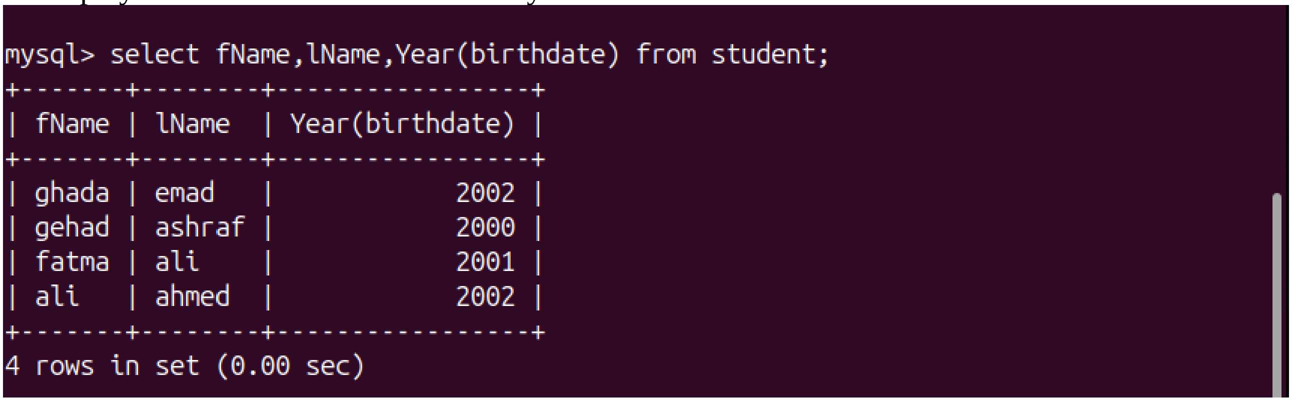
```
mysql> select fName,lName,age from student;
+-------+--------+------+
| fName | lName  | age  |
+-------+--------+------+
| ghada | emad   |   22 |
| gehad | ashraf |   23 |
| fatma | ali    |   50 |
| ali   | ahmed  |   34 |
+-------+--------+------+
4 rows in set (0.00 sec)
```

4. Display the name of students with their Rounded score in each Exam



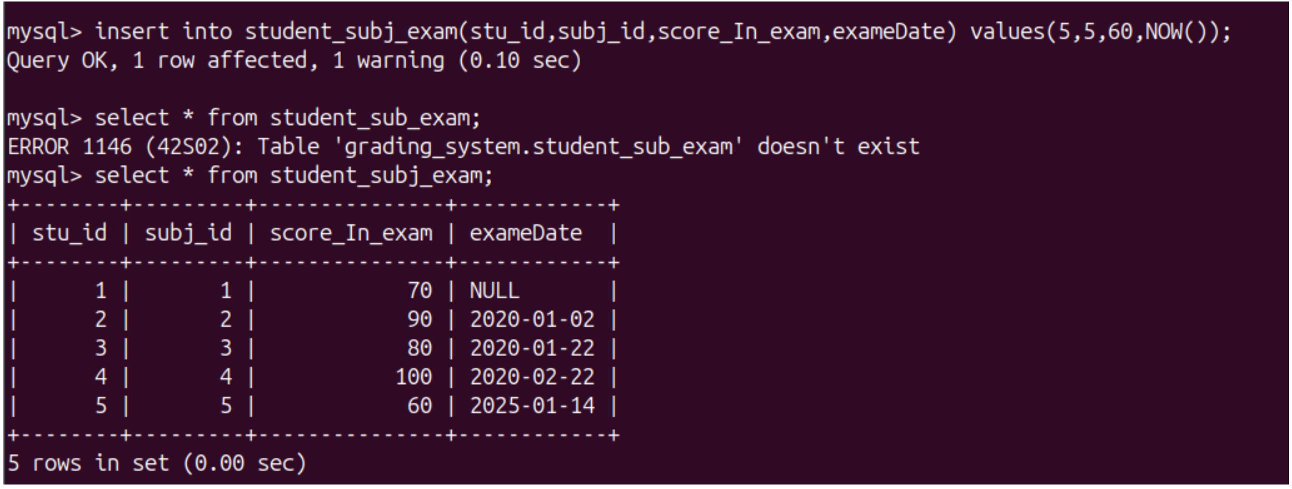
```
mysql> select S.fName,S.lName,E.score_In_exam from student S,student_subj_exam E whe
re S.id=E.stu_id;
+-------+--------+---------------+
| fName | lName  | score_In_exam |
+-------+--------+---------------+
| ghada | emad   |            70 |
| gehad | ashraf |            90 |
| fatma | ali    |            80 |
| ali   | ahmed  |           100 |
+-------+--------+---------------+
4 rows in set (0.00 sec)
```

5. Display the name of students with the year of Birthdate



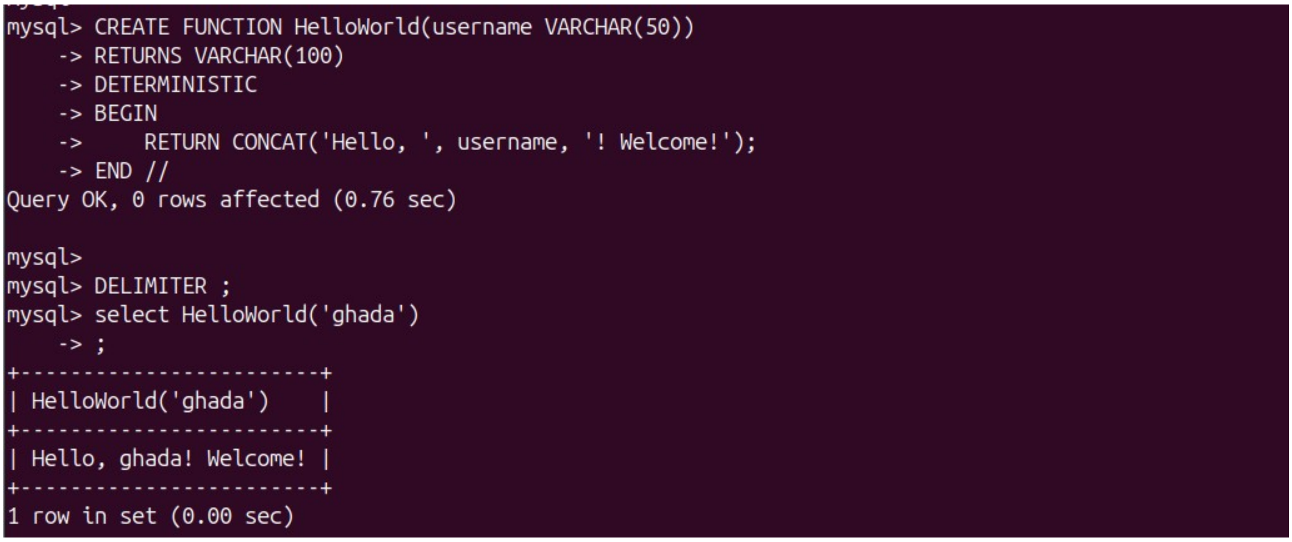
```
mysql> select fName,lName,Year(birthdate) from student;
+-------+--------+-----------------+
| fName | lName  | Year(birthdate) |
+-------+--------+-----------------+
| ghada | emad   |            2002 |
| gehad | ashraf |            2000 |
| fatma | ali    |            2001 |
| ali   | ahmed  |            2002 |
+-------+--------+-----------------+
4 rows in set (0.00 sec)
```

6. Add new exam result, in date column use NOW



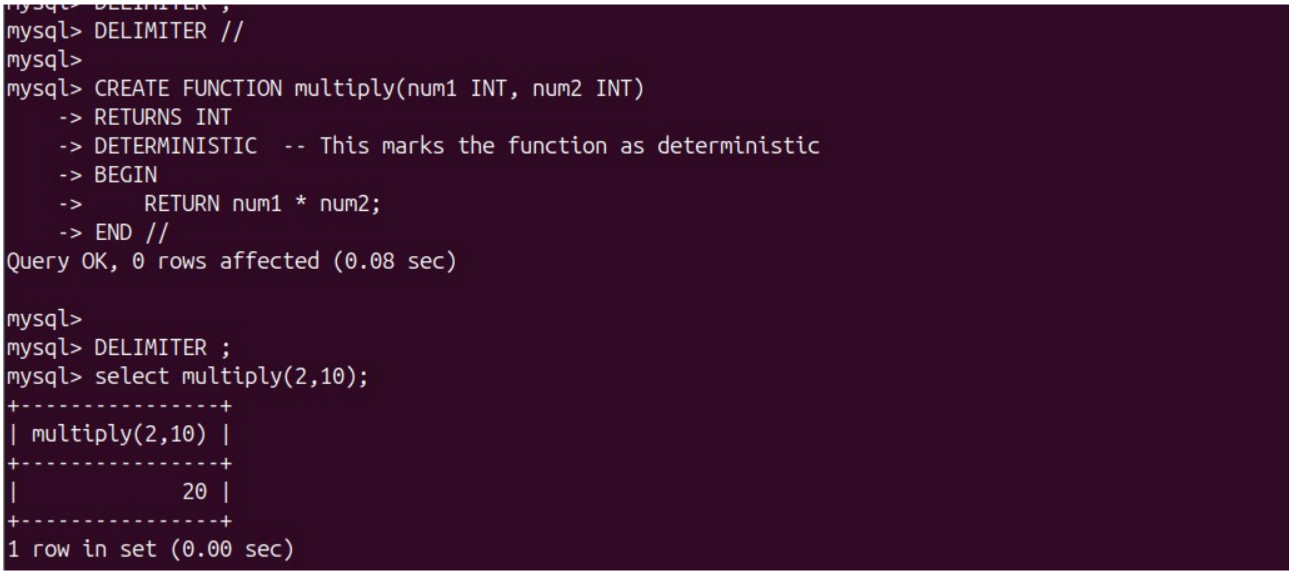
```
mysql> insert into student_subj_exam(stu_id,subj_id,score_In_exam,exameDate) values(5,5,60,NOW());
Query OK, 1 row affected, 1 warning (0.10 sec)

mysql> select * from student_sub_exam;
ERROR 1146 (42S02): Table 'grading_system.student_sub_exam' doesn't exist
mysql> select * from student_subj_exam;
+--------+---------+---------------+------------+
| stu_id | subj_id | score_In_exam | exameDate  |
+--------+---------+---------------+------------+
|      1 |       1 |            70 | NULL       |
|      2 |       2 |            90 | 2020-01-02 |
|      3 |       3 |            80 | 2020-01-22 |
|      4 |       4 |           100 | 2020-02-22 |
|      5 |       5 |            60 | 2025-01-14 |
+--------+---------+---------------+------------+
5 rows in set (0.00 sec)
```

7. Create Hello world function which take username and return welcome
message to user using his name



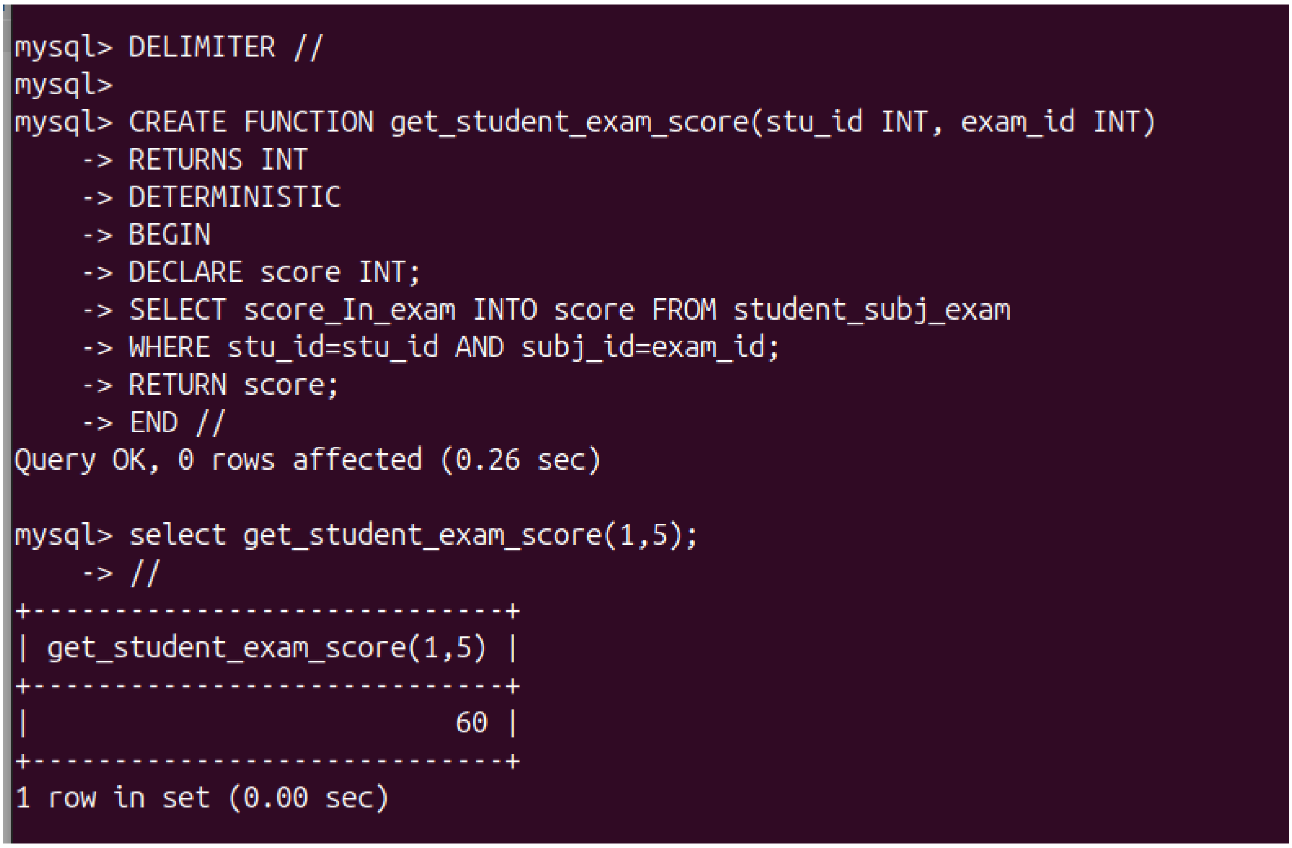
```
mysql> CREATE FUNCTION HelloWorld(username VARCHAR(50))
    -> RETURNS VARCHAR(100)
    -> DETERMINISTIC
    -> BEGIN
    ->     RETURN CONCAT('Hello, ', username, '! Welcome!');
    -> END //
Query OK, 0 rows affected (0.76 sec)

mysql>
mysql> DELIMITER ;
mysql> select HelloWorld('ghada')
    -> ;
+------------------------+
| HelloWorld('ghada')    |
+------------------------+
| Hello, ghada! Welcome! |
+------------------------+
1 row in set (0.00 sec)
```

8. Create multiply function which take two number and return the multiply of them



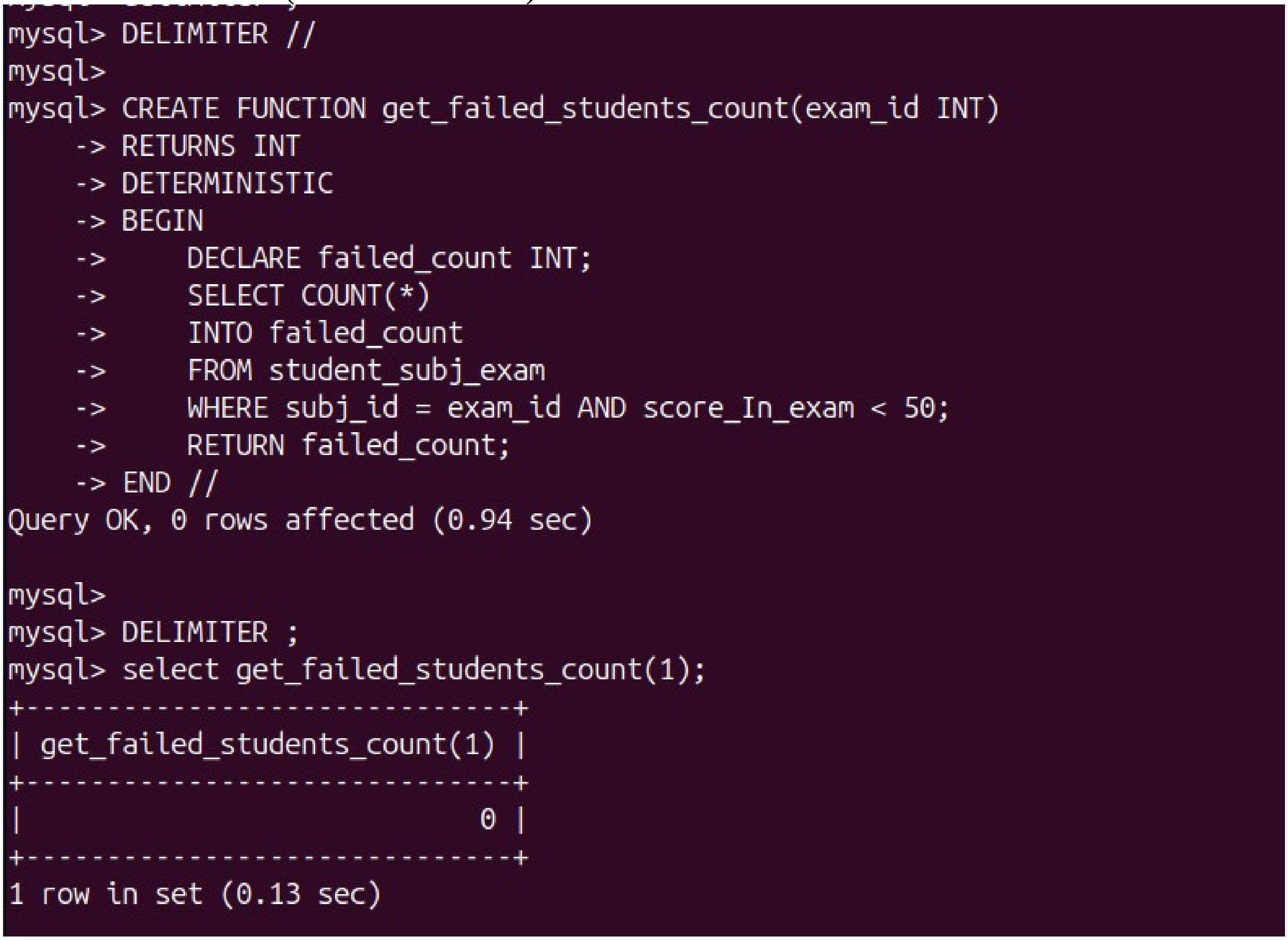
```
mysql> DELIMITER //
mysql>
mysql> CREATE FUNCTION multiply(num1 INT, num2 INT)
    -> RETURNS INT
    -> DETERMINISTIC  -- This marks the function as deterministic
    -> BEGIN
    ->     RETURN num1 * num2;
    -> END //
Query OK, 0 rows affected (0.08 sec)

mysql>
mysql> DELIMITER ;
mysql> select multiply(2,10);
+----------------+
| multiply(2,10) |
+----------------+
|             20 |
+----------------+
1 row in set (0.00 sec)
```

9. Create function which takes student id and Exam id and return score the student in Exam.



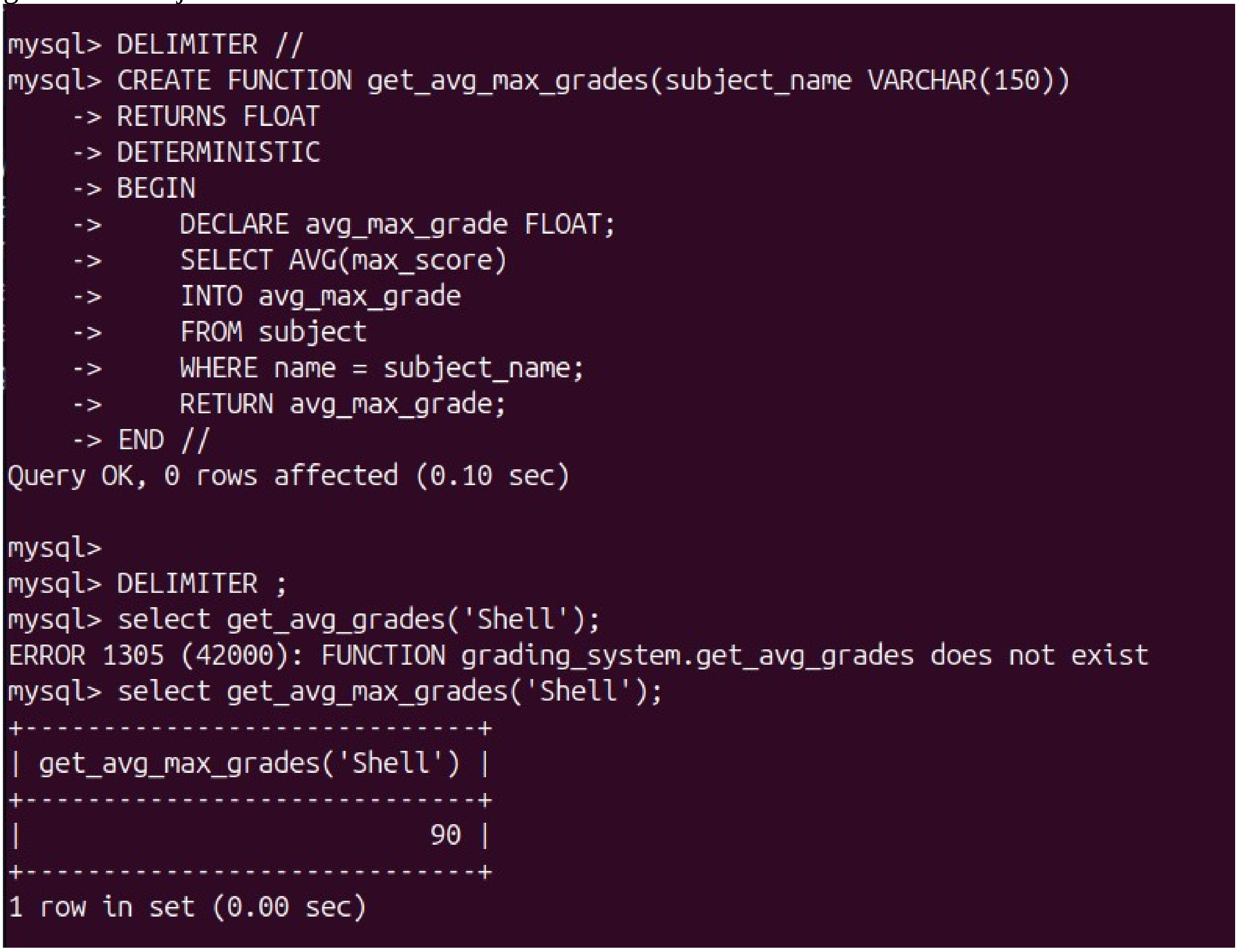
```
mysql> DELIMITER //
mysql>
mysql> CREATE FUNCTION get_student_exam_score(stu_id INT, exam_id INT)
    -> RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    -> DECLARE score INT;
    -> SELECT score_In_exam INTO score FROM student_subj_exam
    -> WHERE stu_id=stu_id AND subj_id=exam_id;
    -> RETURN score;
    -> END //
Query OK, 0 rows affected (0.26 sec)

mysql> select get_student_exam_score(1,5);
    -> //
+-----------------------------+
| get_student_exam_score(1,5) |
+-----------------------------+
|                          60 |
+-----------------------------+
1 row in set (0.00 sec)
```

10. Create function which takes Exam id and return the number of students who failed in a Exam (Score less than 50).



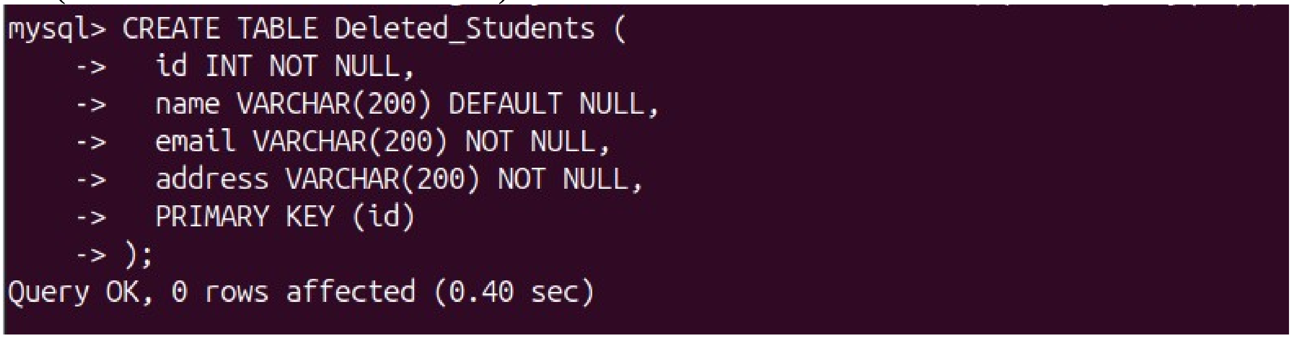
```
mysql> DELIMITER //
mysql>
mysql> CREATE FUNCTION get_failed_students_count(exam_id INT)
    -> RETURNS INT
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE failed_count INT;
    ->     SELECT COUNT(*)
    ->     INTO failed_count
    ->     FROM student_subj_exam
    ->     WHERE subj_id = exam_id AND score_In_exam < 50;
    ->     RETURN failed_count;
    -> END //
Query OK, 0 rows affected (0.94 sec)

mysql>
mysql> DELIMITER ;
mysql> select get_failed_students_count(1);
+------------------------------+
| get_failed_students_count(1) |
+------------------------------+
|                            0 |
+------------------------------+
1 row in set (0.13 sec)
```

11. Create function which take subject name and return the average of max grades for subject



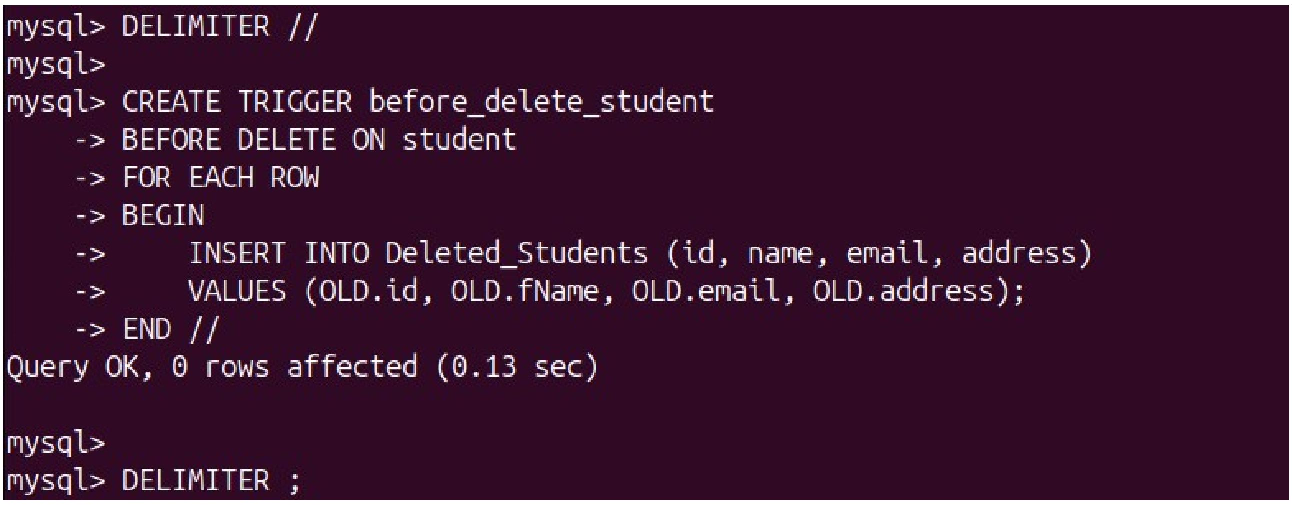
```
mysql> DELIMITER //
mysql> CREATE FUNCTION get_avg_max_grades(subject_name VARCHAR(150))
    -> RETURNS FLOAT
    -> DETERMINISTIC
    -> BEGIN
    ->     DECLARE avg_max_grade FLOAT;
    ->     SELECT AVG(max_score)
    ->     INTO avg_max_grade
    ->     FROM subject
    ->     WHERE name = subject_name;
    ->     RETURN avg_max_grade;
    -> END //
Query OK, 0 rows affected (0.10 sec)

mysql>
mysql> DELIMITER ;
mysql> select get_avg_grades('Shell');
ERROR 1305 (42000): FUNCTION grading_system.get_avg_grades does not exist
mysql> select get_avg_max_grades('Shell');
+-----------------------------+
| get_avg_max_grades('Shell') |
+-----------------------------+
|                          90 |
+-----------------------------+
1 row in set (0.00 sec)
```

12. Create Table called Deleted_Students which will hold the deleted students info(same columns as in student tables)



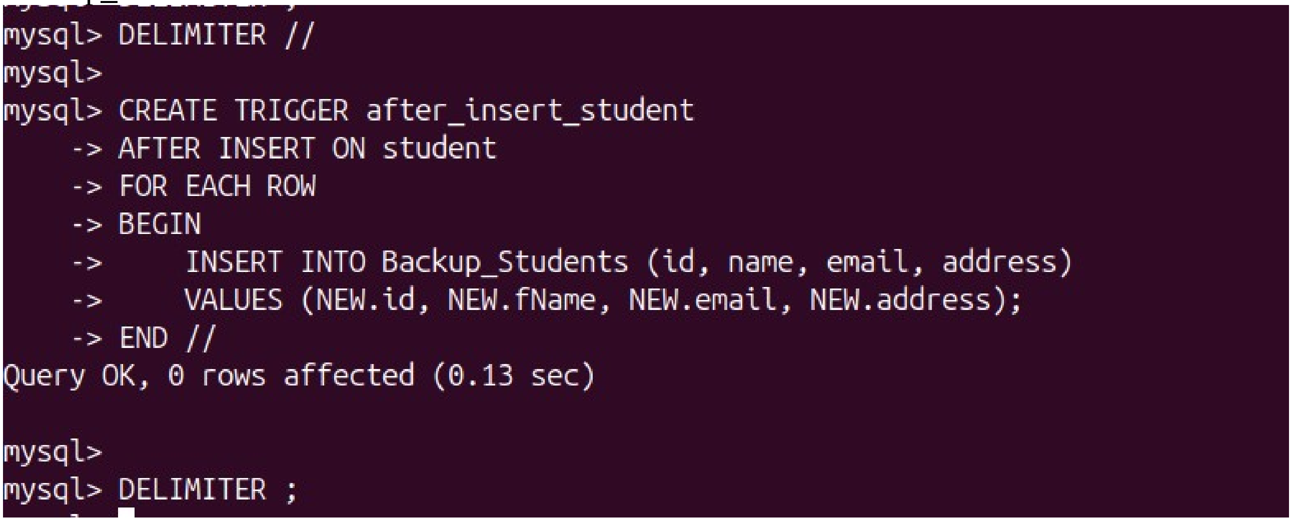
```
mysql> CREATE TABLE Deleted_Students (
    ->   id INT NOT NULL,
    ->   name VARCHAR(200) DEFAULT NULL,
    ->   email VARCHAR(200) NOT NULL,
    ->   address VARCHAR(200) NOT NULL,
    ->   PRIMARY KEY (id)
    -> );
Query OK, 0 rows affected (0.40 sec)
```

13. Create trigger to save the deleted student from Student table to Deleted_Students.



```
mysql> DELIMITER //
mysql>
mysql> CREATE TRIGGER before_delete_student
    -> BEFORE DELETE ON student
    -> FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO Deleted_Students (id, name, email, address)
    ->     VALUES (OLD.id, OLD.fName, OLD.email, OLD.address);
    -> END //
Query OK, 0 rows affected (0.13 sec)

mysql>
mysql> DELIMITER ;
```

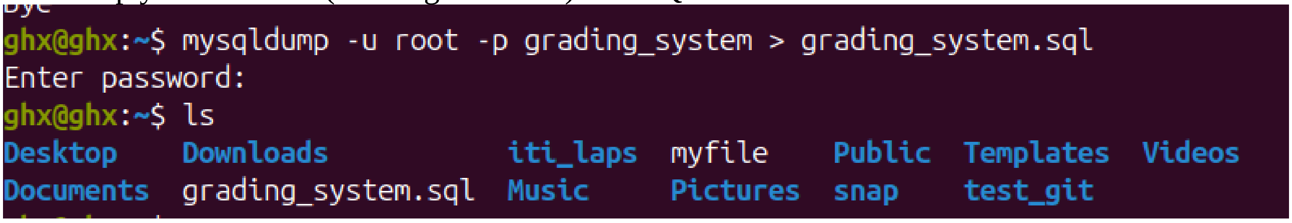
14. Create trigger to save the newly added students to Student table to Backup_Students.



```
mysql> DELIMITER //
mysql>
mysql> CREATE TRIGGER after_insert_student
    -> AFTER INSERT ON student
    -> FOR EACH ROW
    -> BEGIN
    ->     INSERT INTO Backup_Students (id, name, email, address)
    ->     VALUES (NEW.id, NEW.fName, NEW.email, NEW.address);
    -> END //
Query OK, 0 rows affected (0.13 sec)

mysql>
mysql> DELIMITER ;
```

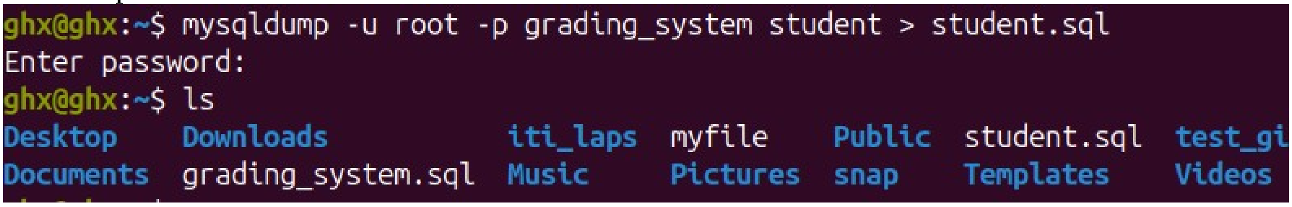
15. (Bouns) Create trigger to keep track the changes of contact info table (add/update rows); it will logs the time of action and description of action to another table.

16. Dump your database (Grading Database) into SQL file.



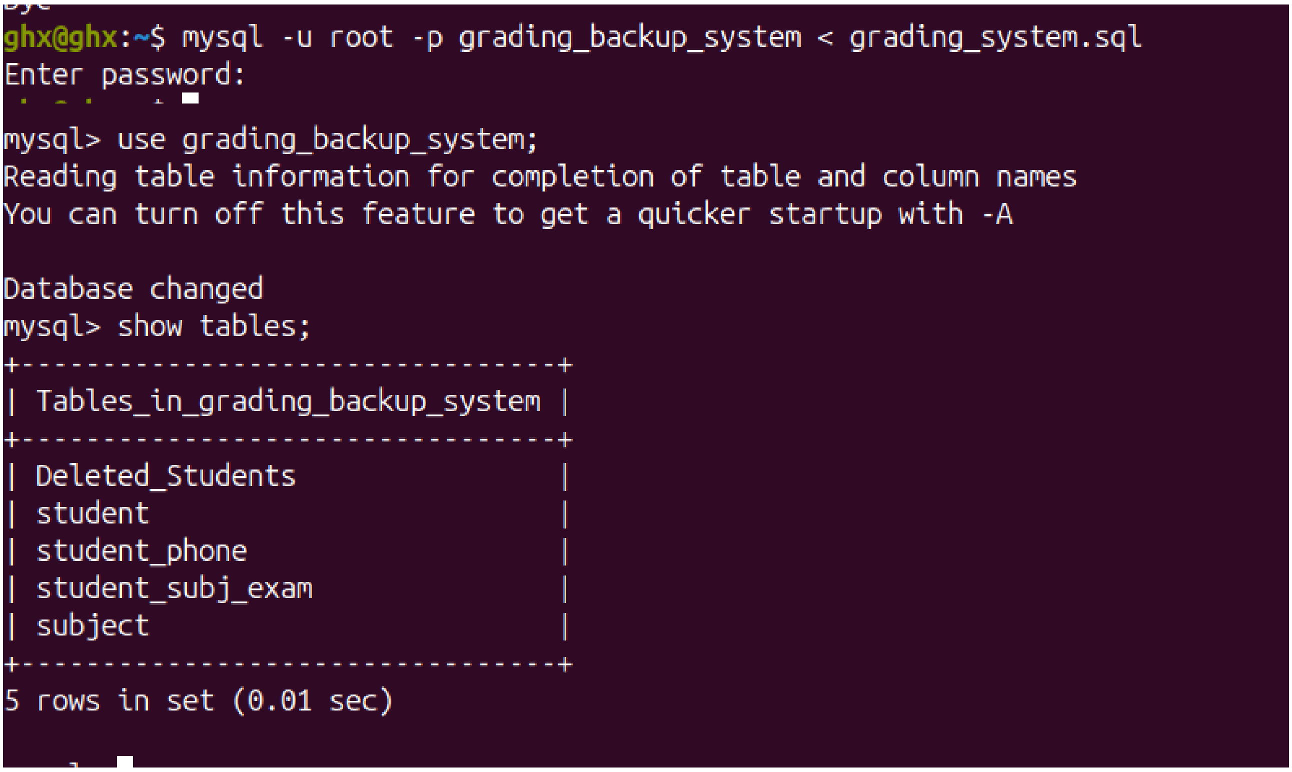
```
ghx@ghx:~$ mysqldump -u root -p grading_system > grading_system.sql
Enter password:
ghx@ghx:~$ ls
Desktop    Downloads           iti_laps  myfile    Public  Templates  Videos
Documents  grading_system.sql  Music     Pictures  snap    test_git
```

17. Dump Students table into file.



```
ghx@ghx:~$ mysqldump -u root -p grading_system student > student.sql
Enter password:
ghx@ghx:~$ ls
Desktop    Downloads           iti_laps  myfile    Public  student.sql  test_gi
Documents  grading_system.sql  Music     Pictures  snap    Templates    Videos
```

18. Import SQL file into your backup database (Grading_Backup Database)



```
Bye
ghx@ghx:~$ mysql -u root -p grading_backup_system < grading_system.sql
Enter password:

mysql> use grading_backup_system;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+---------------------------------+
| Tables_in_grading_backup_system |
+---------------------------------+
| Deleted_Students                |
| student                         |
| student_phone                   |
| student_subj_exam               |
| subject                         |
+---------------------------------+
5 rows in set (0.01 sec)
```