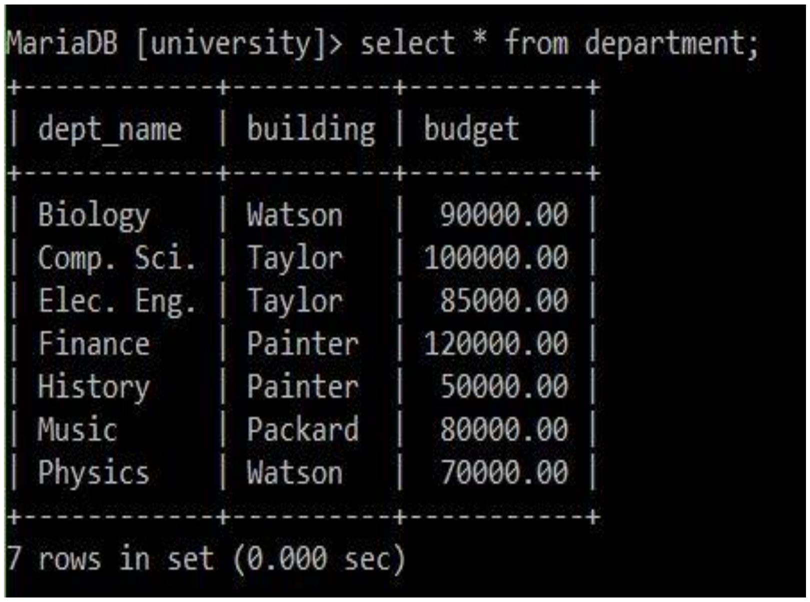
ASSIGNMENT -6 **ABHICHAL RAJPUT**

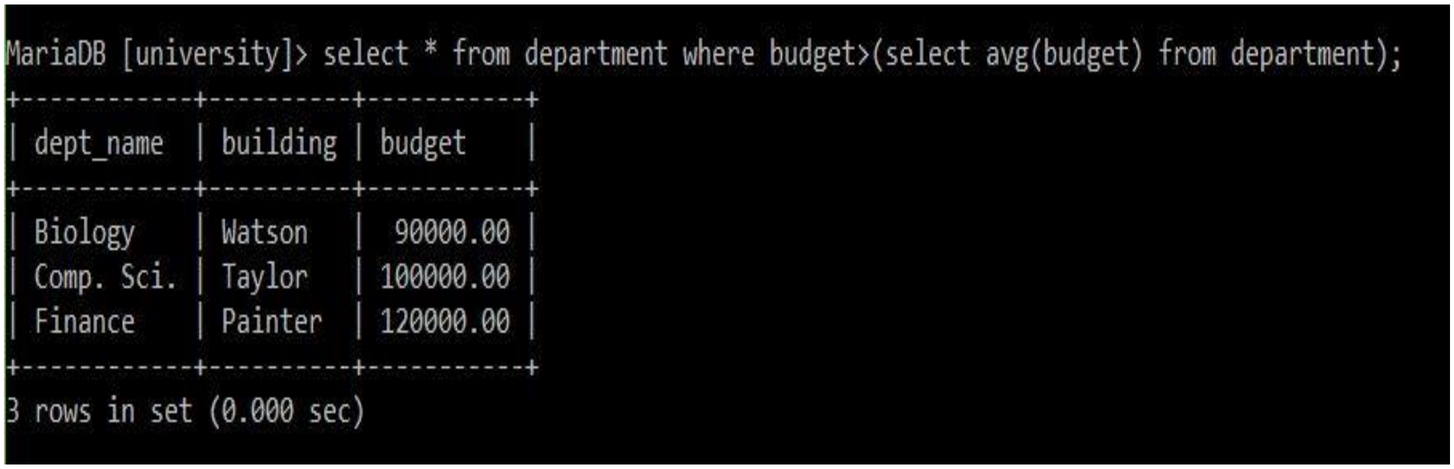
Q1. show the details of the departments which have budgets more than the average budget across all departments. First show it without defining any function, then show it by defning a function avg_budget that return average budget across all departments.

Step-1:- showing the Budgets for every department



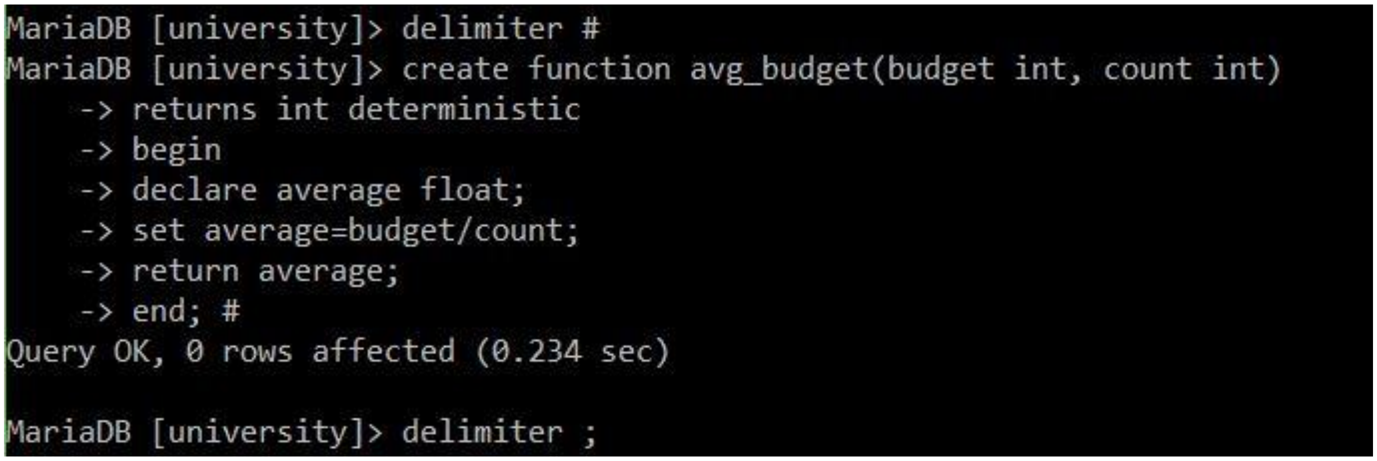
```
MariaDB [university]> select * from department;
+------------+----------+-----------+
| dept_name  | building | budget    |
+------------+----------+-----------+
| Biology    | Watson   |  90000.00 |
| Comp. Sci. | Taylor   | 100000.00 |
| Elec. Eng. | Taylor   |  85000.00 |
| Finance    | Painter  | 120000.00 |
| History    | Painter  |  50000.00 |
| Music      | Packard  |  80000.00 |
| Physics    | Watson   |  70000.00 |
+------------+----------+-----------+
7 rows in set (0.000 sec)
```

Step2:- showing the departments having budget more than average budget across all departments without defining any function.



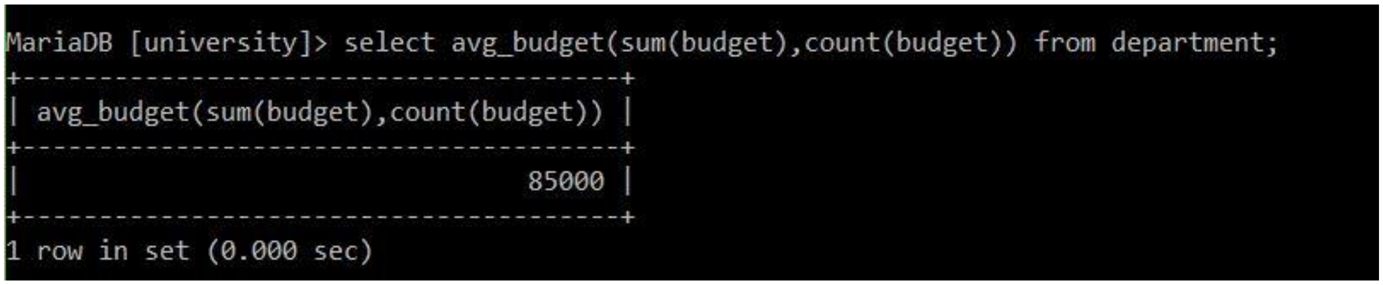
```
MariaDB [university]> select * from department where budget>(select avg(budget) from department);
+------------+----------+-----------+
| dept_name  | building | budget    |
+------------+----------+-----------+
| Biology    | Watson   |  90000.00 |
| Comp. Sci. | Taylor   | 100000.00 |
| Finance    | Painter  | 120000.00 |
+------------+----------+-----------+
3 rows in set (0.000 sec)
```

Step 3:- Creating a user defined function (avg_budget) for calculating the average of all the departments.



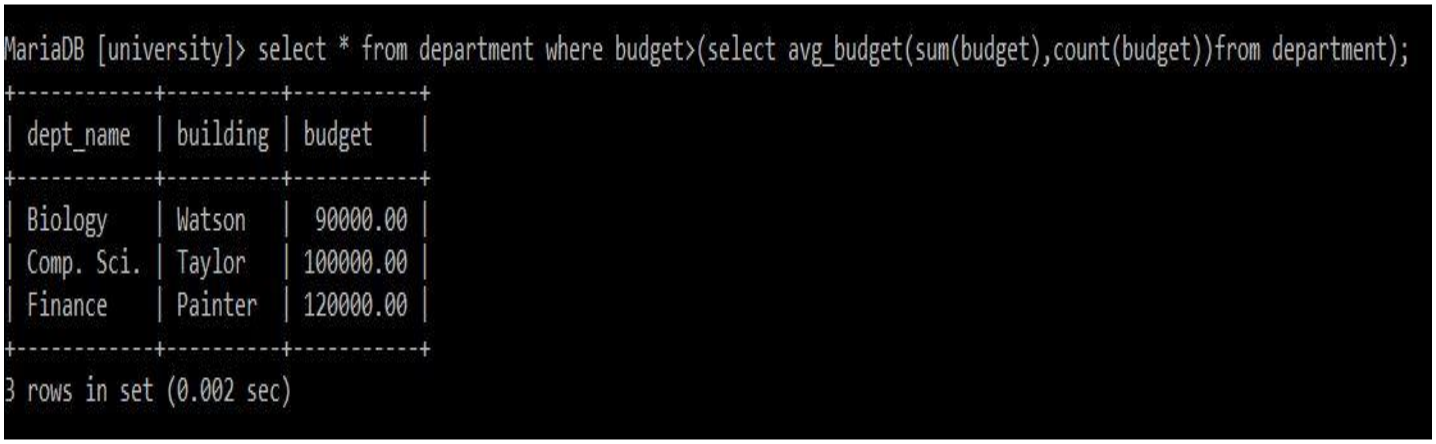
```
MariaDB [university]> delimiter #
MariaDB [university]> create function avg_budget(budget int, count int)
    -> returns int deterministic
    -> begin
    -> declare average float;
    -> set average=budget/count;
    -> return average;
    -> end; #
Query OK, 0 rows affected (0.234 sec)

MariaDB [university]> delimiter ;
```

Step 4:- Showing the average budget using the function defined above to show it's working correctly.



```
MariaDB [university]> select avg_budget(sum(budget),count(budget)) from department;
+---------------------------------------+
| avg_budget(sum(budget),count(budget)) |
+---------------------------------------+
|                                 85000 |
+---------------------------------------+
1 row in set (0.000 sec)
```

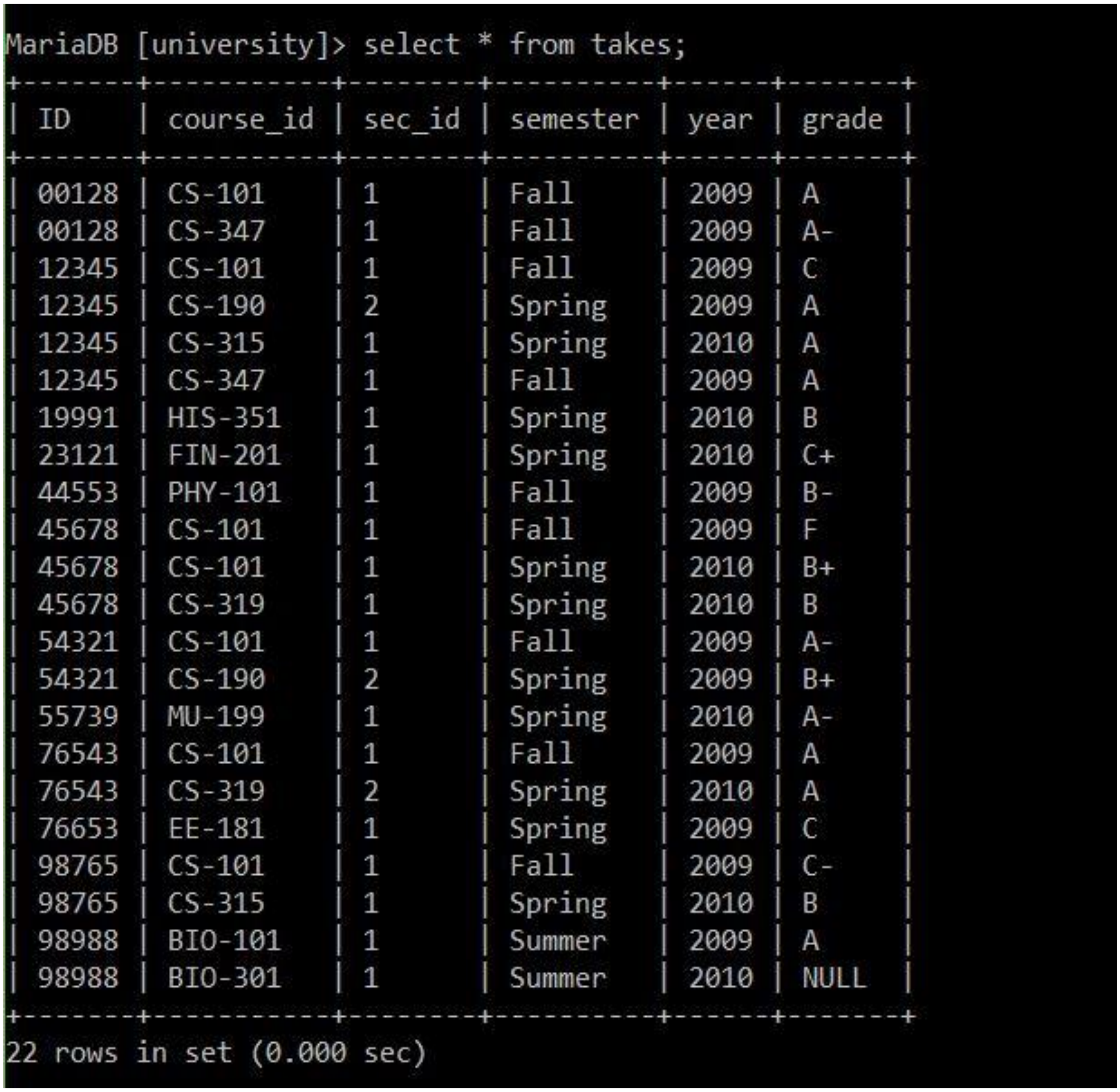
Step 5:- Finding all the departments having budget greater than the average budget across all departments using the user defined avg_budget function created above.



```
MariaDB [university]> select * from department where budget>(select avg_budget(sum(budget),count(budget))from department);
+------------+----------+-----------+
| dept_name  | building | budget    |
+------------+----------+-----------+
| Biology    | Watson   |  90000.00 |
| Comp. Sci. | Taylor   | 100000.00 |
| Finance    | Painter  | 120000.00 |
+------------+----------+-----------+
3 rows in set (0.002 sec)
```

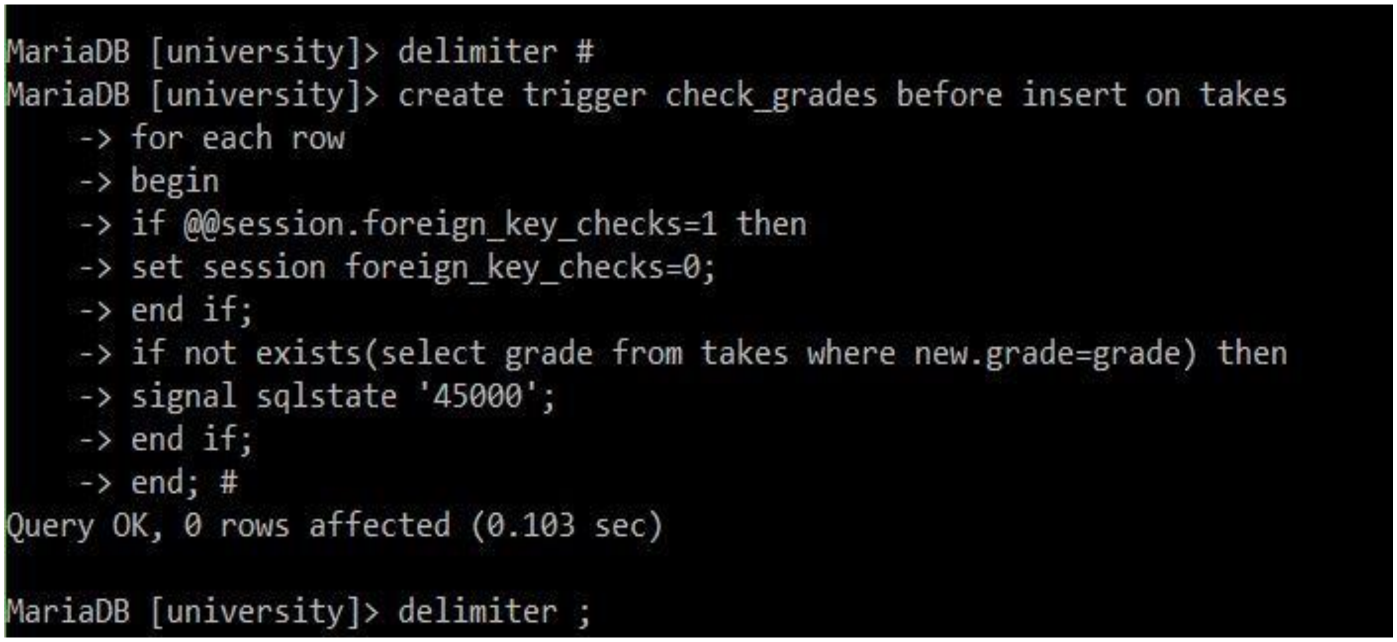
Q2. Create a trigger that will not allow to enter any record into the takes table with a grade that is not used before in any record in the takes table.

Step 1:- Showing the takes table initially i.e.(before any operation)



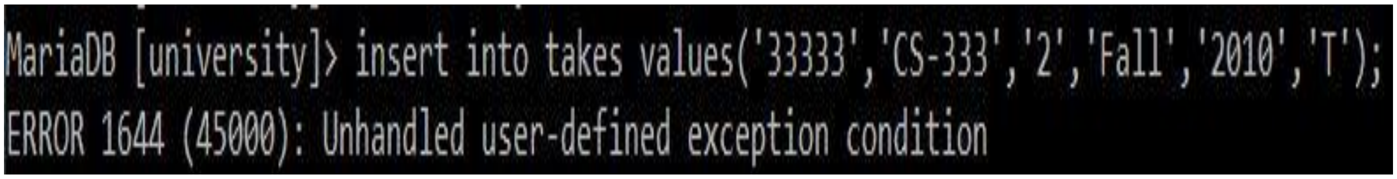
```
MariaDB [university]> select * from takes;
+-------+-----------+--------+----------+------+-------+
| ID    | course_id | sec_id | semester | year | grade |
+-------+-----------+--------+----------+------+-------+
| 00128 | CS-101    | 1      | Fall     | 2009 | A     |
| 00128 | CS-347    | 1      | Fall     | 2009 | A-    |
| 12345 | CS-101    | 1      | Fall     | 2009 | C     |
| 12345 | CS-190    | 2      | Spring   | 2009 | A     |
| 12345 | CS-315    | 1      | Spring   | 2010 | A     |
| 12345 | CS-347    | 1      | Fall     | 2009 | A     |
| 19991 | HIS-351   | 1      | Spring   | 2010 | B     |
| 23121 | FIN-201   | 1      | Spring   | 2010 | C+    |
| 44553 | PHY-101   | 1      | Fall     | 2009 | B-    |
| 45678 | CS-101    | 1      | Fall     | 2009 | F     |
| 45678 | CS-101    | 1      | Spring   | 2010 | B+    |
| 45678 | CS-319    | 1      | Spring   | 2010 | B     |
| 54321 | CS-101    | 1      | Fall     | 2009 | A-    |
| 54321 | CS-190    | 2      | Spring   | 2009 | B+    |
| 55739 | MU-199    | 1      | Spring   | 2010 | A-    |
| 76543 | CS-101    | 1      | Fall     | 2009 | A     |
| 76543 | CS-319    | 2      | Spring   | 2010 | A     |
| 76653 | EE-181    | 1      | Spring   | 2009 | C     |
| 98765 | CS-101    | 1      | Fall     | 2009 | C-    |
| 98765 | CS-315    | 1      | Spring   | 2010 | B     |
| 98988 | BIO-101   | 1      | Summer   | 2009 | A     |
| 98988 | BIO-301   | 1      | Summer   | 2010 | NULL  |
+-------+-----------+--------+----------+------+-------+
22 rows in set (0.000 sec)
```

Step 2:- Creating a Trigger which envoke before insertion of values. If grade is present in the takes table then values will be inserted else an error will be produced by the query.



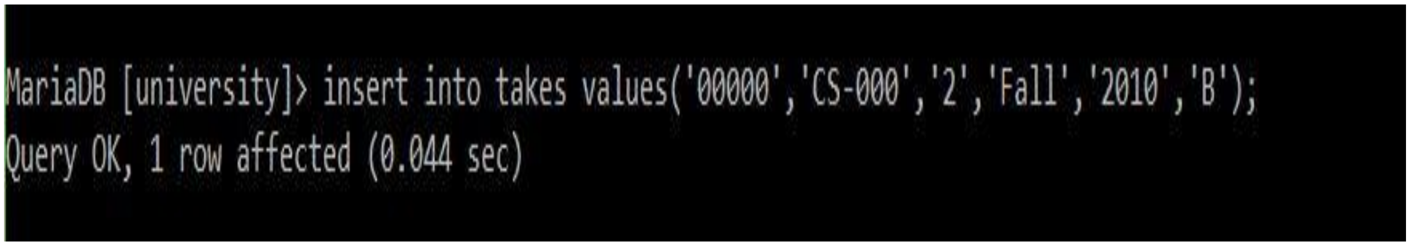
```
MariaDB [university]> delimiter #
MariaDB [university]> create trigger check_grades before insert on takes
    -> for each row
    -> begin
    -> if @@session.foreign_key_checks=1 then
    -> set session foreign_key_checks=0;
    -> end if;
    -> if not exists(select grade from takes where new.grade=grade) then
    -> signal sqlstate '45000';
    -> end if;
    -> end; #
Query OK, 0 rows affected (0.103 sec)

MariaDB [university]> delimiter ;
```

Step 3:- Trying to insert a row in takes with grade='T' which is not in takes table. Hence, an error is produced after execution of query



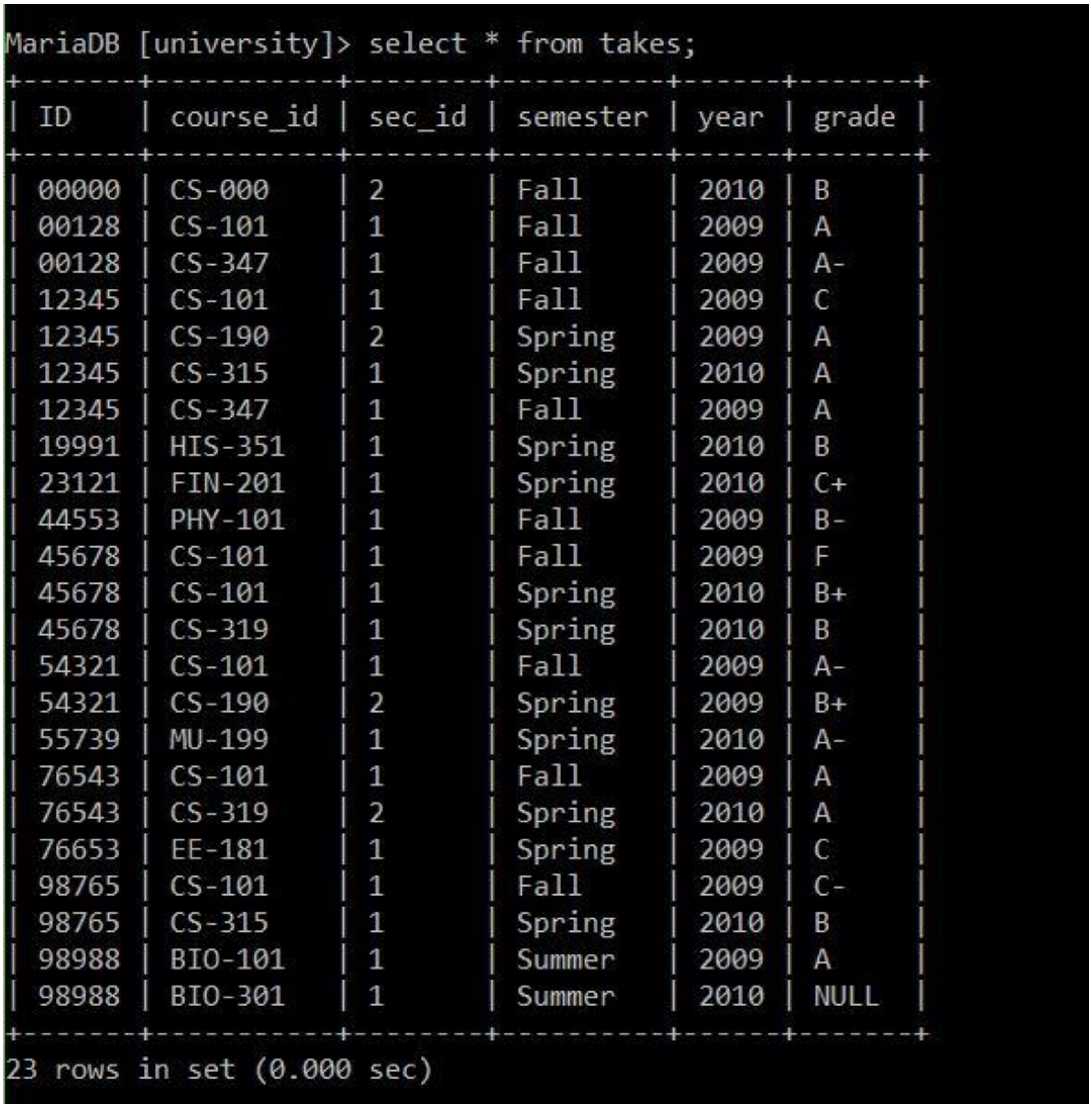
```
MariaDB [university]> insert into takes values('33333','CS-333','2','Fall','2010','T');
ERROR 1644 (45000): Unhandled user-defined exception condition
```

Step 4:- Trying to insert a row in takes with grade='B' which is present takes table. Hence, inserted into the table.



```
MariaDB [university]> insert into takes values('00000','CS-000','2','Fall','2010','B');
Query OK, 1 row affected (0.044 sec)
```

Step 5:- Showing the after table , the row with grade ‘B’ and ID=00000 is inserted into the table.

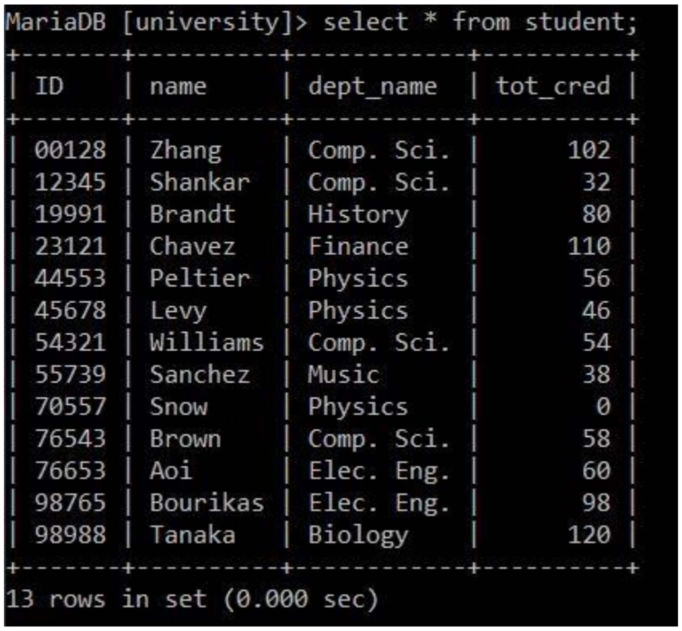


```
MariaDB [university]> select * from takes;
+-------+-----------+--------+----------+------+-------+
| ID    | course_id | sec_id | semester | year | grade |
+-------+-----------+--------+----------+------+-------+
| 00000 | CS-000    | 2      | Fall     | 2010 | B     |
| 00128 | CS-101    | 1      | Fall     | 2009 | A     |
| 00128 | CS-347    | 1      | Fall     | 2009 | A-    |
| 12345 | CS-101    | 1      | Fall     | 2009 | C     |
| 12345 | CS-190    | 2      | Spring   | 2009 | A     |
| 12345 | CS-315    | 1      | Spring   | 2010 | A     |
| 12345 | CS-347    | 1      | Fall     | 2009 | A     |
| 19991 | HIS-351   | 1      | Spring   | 2010 | B     |
| 23121 | FIN-201   | 1      | Spring   | 2010 | C+    |
| 44553 | PHY-101   | 1      | Fall     | 2009 | B-    |
| 45678 | CS-101    | 1      | Fall     | 2009 | F     |
| 45678 | CS-101    | 1      | Spring   | 2010 | B+    |
| 45678 | CS-319    | 1      | Spring   | 2010 | B     |
| 54321 | CS-101    | 1      | Fall     | 2009 | A-    |
| 54321 | CS-190    | 2      | Spring   | 2009 | B+    |
| 55739 | MU-199    | 1      | Spring   | 2010 | A-    |
| 76543 | CS-101    | 1      | Fall     | 2009 | A     |
| 76543 | CS-319    | 2      | Spring   | 2010 | A     |
| 76653 | EE-181    | 1      | Spring   | 2009 | C     |
| 98765 | CS-101    | 1      | Fall     | 2009 | C-    |
| 98765 | CS-315    | 1      | Spring   | 2010 | B     |
| 98988 | BIO-101   | 1      | Summer   | 2009 | A     |
| 98988 | BIO-301   | 1      | Summer   | 2010 | NULL  |
+-------+-----------+--------+----------+------+-------+
23 rows in set (0.000 sec)
```

ASSIGNMENT -6 **ABHICHAL RAJPUT**

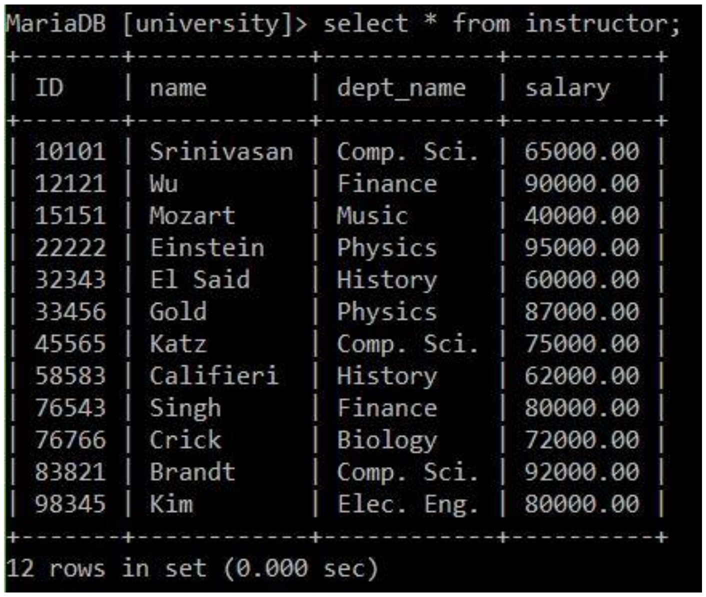
Q3. Create a view to show the students names and their advisors names.

Step 1:- Showing the student table.



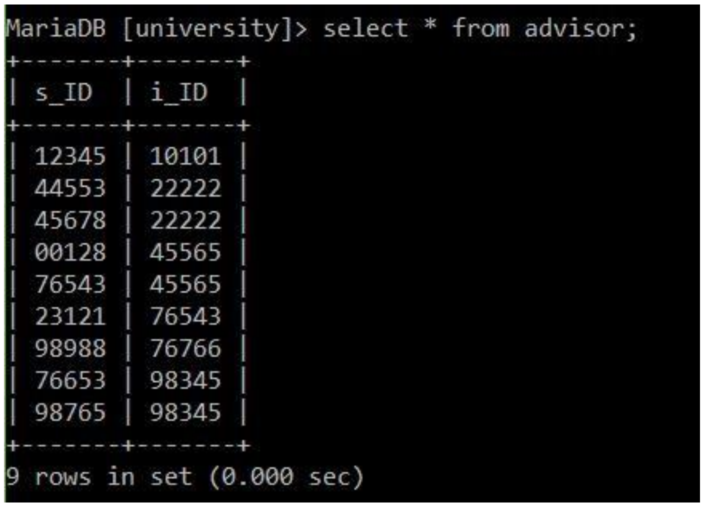
```
MariaDB [university]> select * from student;
+-------+----------+------------+----------+
| ID    | name     | dept_name  | tot_cred |
+-------+----------+------------+----------+
| 00128 | Zhang    | Comp. Sci. |      102 |
| 12345 | Shankar  | Comp. Sci. |       32 |
| 19991 | Brandt   | History    |       80 |
| 23121 | Chavez   | Finance    |      110 |
| 44553 | Peltier  | Physics    |       56 |
| 45678 | Levy     | Physics    |       46 |
| 54321 | Williams | Comp. Sci. |       54 |
| 55739 | Sanchez  | Music      |       38 |
| 70557 | Snow     | Physics    |        0 |
| 76543 | Brown    | Comp. Sci. |       58 |
| 76653 | Aoi      | Elec. Eng. |       60 |
| 98765 | Bourikas | Elec. Eng. |       98 |
| 98988 | Tanaka   | Biology    |      120 |
+-------+----------+------------+----------+
13 rows in set (0.000 sec)
```

Step 2:- Showing the Instructor table.



```
MariaDB [university]> select * from instructor;
+-------+------------+------------+----------+
| ID    | name       | dept_name  | salary   |
+-------+------------+------------+----------+
| 10101 | Srinivasan | Comp. Sci. | 65000.00 |
| 12121 | Wu         | Finance    | 90000.00 |
| 15151 | Mozart     | Music      | 40000.00 |
| 22222 | Einstein   | Physics    | 95000.00 |
| 32343 | El Said    | History    | 60000.00 |
| 33456 | Gold       | Physics    | 87000.00 |
| 45565 | Katz       | Comp. Sci. | 75000.00 |
| 58583 | Califieri  | History    | 62000.00 |
| 76543 | Singh      | Finance    | 80000.00 |
| 76766 | Crick      | Biology    | 72000.00 |
| 83821 | Brandt     | Comp. Sci. | 92000.00 |
| 98345 | Kim        | Elec. Eng. | 80000.00 |
+-------+------------+------------+----------+
12 rows in set (0.000 sec)
```

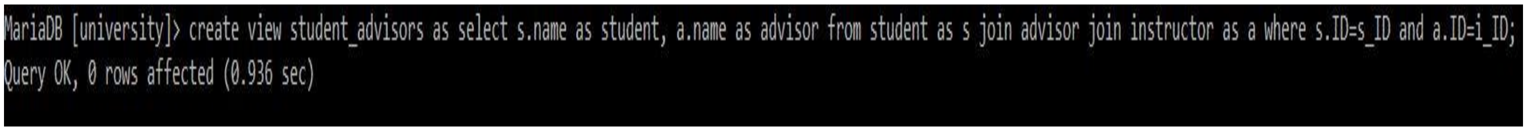
Step 3:- Showing the advisor table which tells us which instructor is advisor of which student.



```
MariaDB [university]> select * from advisor;
+-------+-------+
| s_ID  | i_ID  |
+-------+-------+
| 12345 | 10101 |
| 44553 | 22222 |
| 45678 | 22222 |
| 00128 | 45565 |
| 76543 | 45565 |
| 23121 | 76543 |
| 98988 | 76766 |
| 76653 | 98345 |
| 98765 | 98345 |
+-------+-------+
9 rows in set (0.000 sec)
```

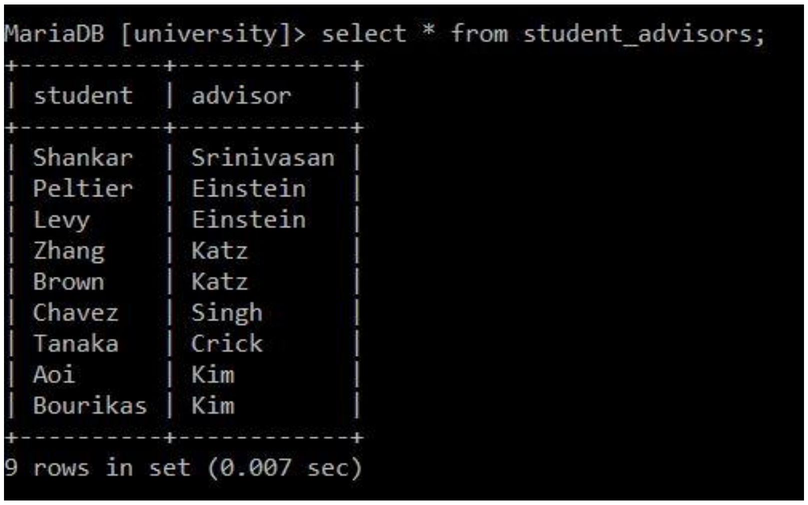
Step 4:- Creating a view (student_advisors) and Selecting the name from student and instructor from join of student, advisor and instructor to get the names of Students and their corresponding advisors.

Query is:- **create view student_advisors as select s.name as student, a.name as advisor from student as s join advisor join instructor as a where s.ID=s_ID and a.ID=i_ID;**



```
MariaDB [university]> create view student_advisors as select s.name as student, a.name as advisor from student as s join advisor join instructor as a where s.ID=s_ID and a.ID=i_ID;
Query OK, 0 rows affected (0.936 sec)
```

Step 5:- Showing the Student names and Advisor names from the view created before i.e. student_advisors.



```
MariaDB [university]> select * from student_advisors;
+----------+------------+
| student  | advisor    |
+----------+------------+
| Shankar  | Srinivasan |
| Peltier  | Einstein   |
| Levy     | Einstein   |
| Zhang    | Katz       |
| Brown    | Katz       |
| Chavez   | Singh      |
| Tanaka   | Crick      |
| Aoi      | Kim        |
| Bourikas | Kim        |
+----------+------------+
9 rows in set (0.007 sec)
```