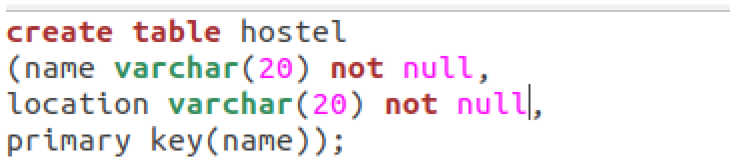
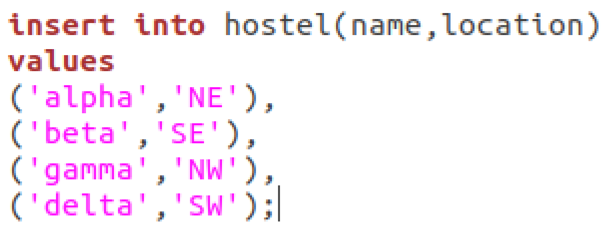
Lab assignment-5
Surendra Baskey(111601027)

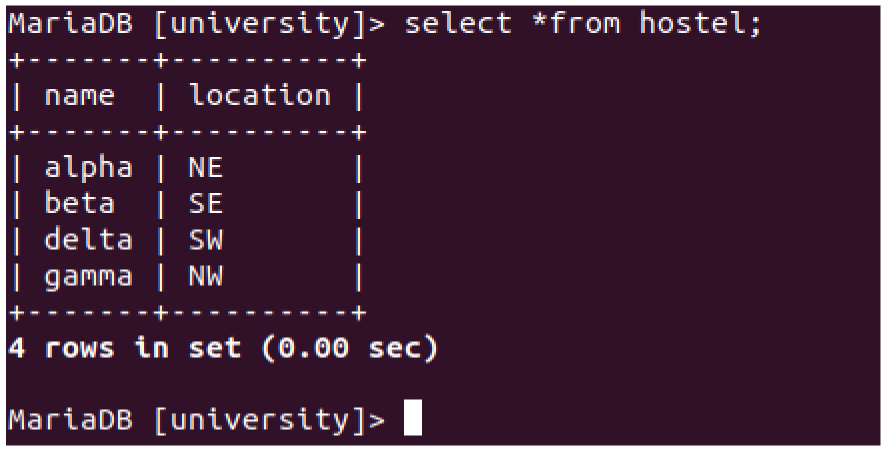
1. Create a table hostel with attributes (name, location). Insert values (‘alpha’,’NE’), (‘beta’,’SE’),(‘gamma’,’NW’),(‘delta’,’SW’). Create a table student_hostel (student_id, hostel_name, room_number). student_id refers to id in student table, hostel_name refers to name in hostel table. room _number is alpha-numeric type. Possible values are A1 to A47, B1 to B47. Write a trigger so that, whenever a new student is added to student table the student gets a room allocated. Rooms are allocated in sequence: hostel names as given before and room numbers as given above.



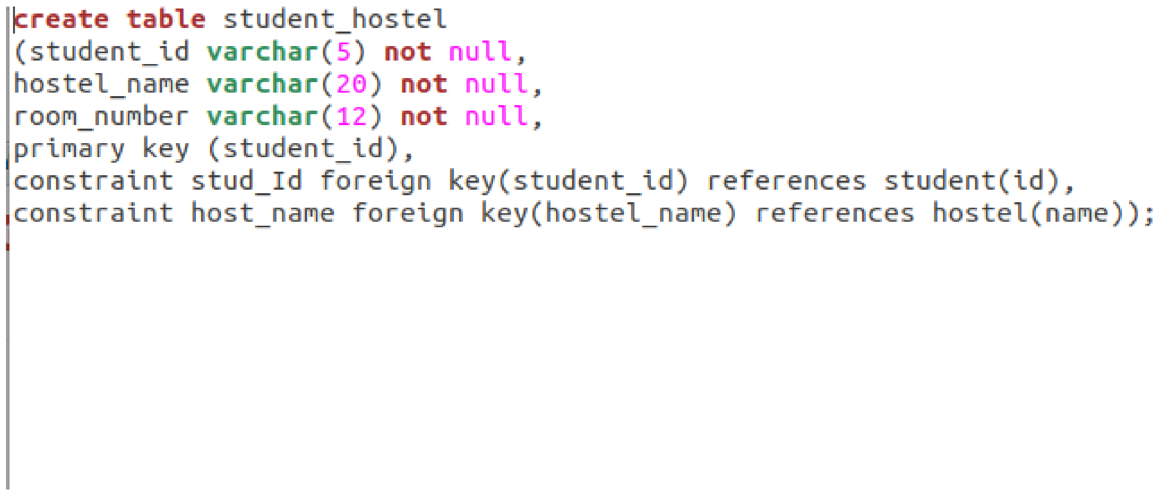
```
create table hostel
(name varchar(20) not null,
location varchar(20) not null,
primary key(name));
```



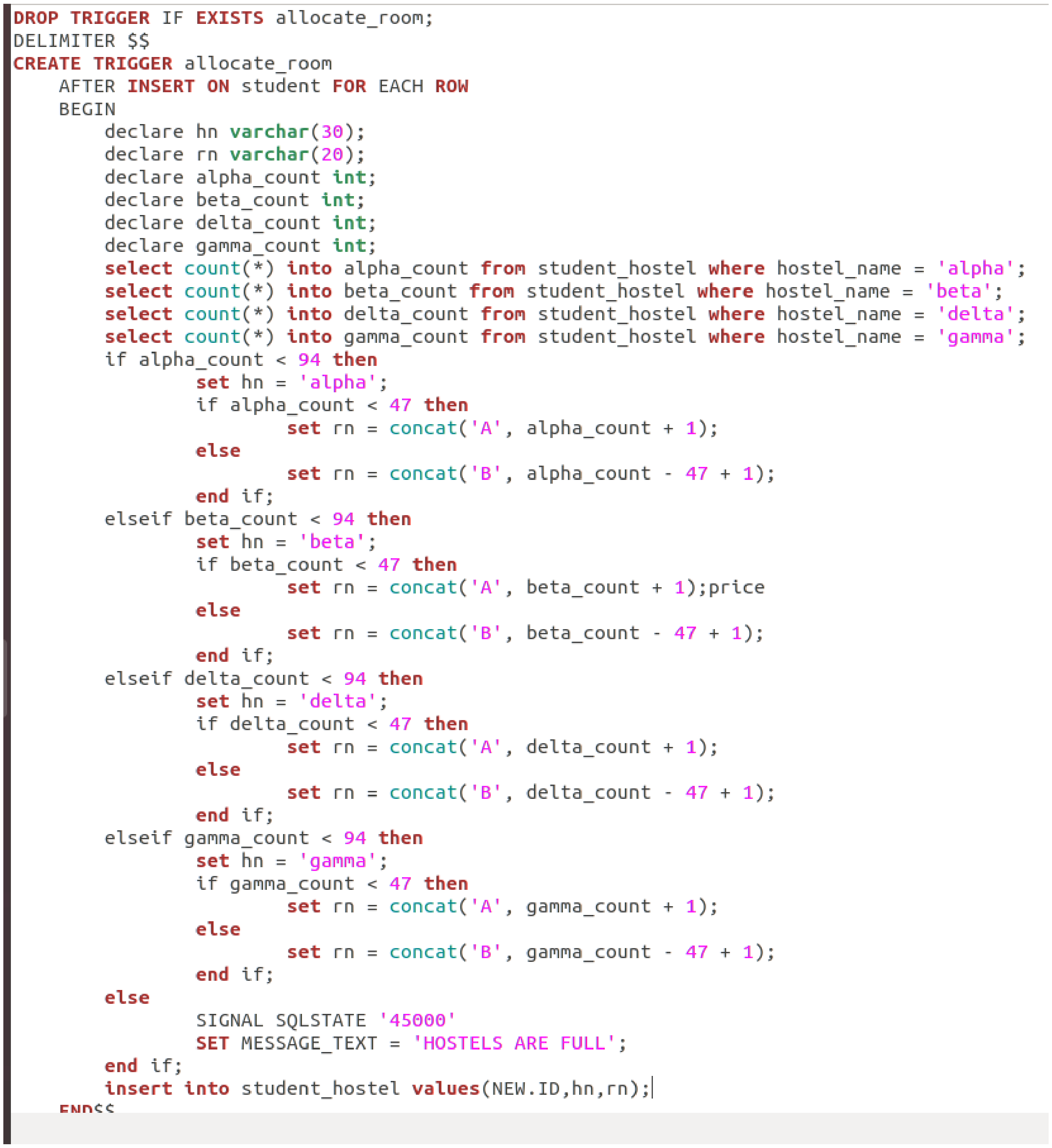
```
insert into hostel(name,location)
values
('alpha','NE'),
('beta','SE'),
('gamma','NW'),
('delta','SW');
```



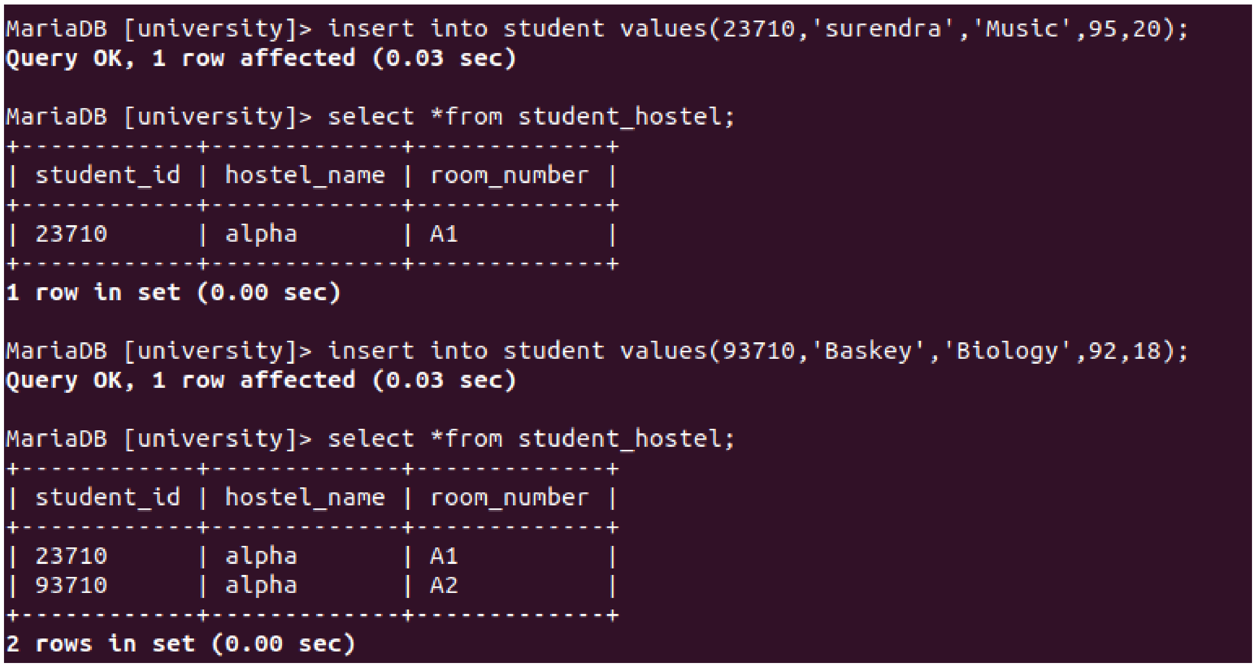
```
MariaDB [university]> select *from hostel;
+-------+----------+
| name  | location |
+-------+----------+
| alpha | NE       |
| beta  | SE       |
| delta | SW       |
| gamma | NW       |
+-------+----------+
4 rows in set (0.00 sec)

MariaDB [university]> 
```



```
create table student_hostel
(student_id varchar(5) not null,
hostel_name varchar(20) not null,
room_number varchar(12) not null,
primary key (student_id),
constraint stud_Id foreign key(student_id) references student(id),
constraint host_name foreign key(hostel_name) references hostel(name));
```

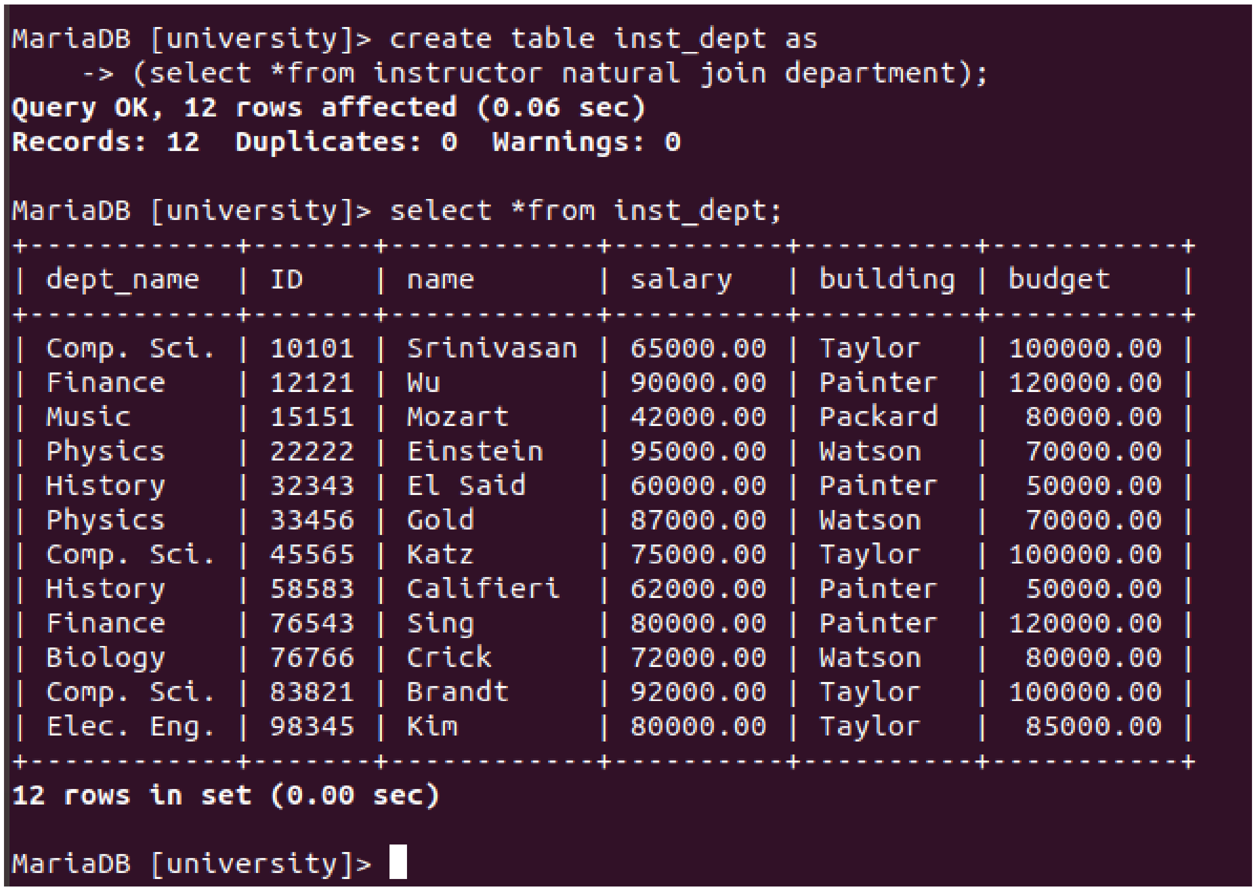


```
DROP TRIGGER IF EXISTS allocate_room;
DELIMITER $$
CREATE TRIGGER allocate_room
    AFTER INSERT ON student FOR EACH ROW
    BEGIN
        declare hn varchar(30);
        declare rn varchar(20);
        declare alpha_count int;
        declare beta_count int;
        declare delta_count int;
        declare gamma_count int;
        select count(*) into alpha_count from student_hostel where hostel_name = 'alpha';
        select count(*) into beta_count from student_hostel where hostel_name = 'beta';
        select count(*) into delta_count from student_hostel where hostel_name = 'delta';
        select count(*) into gamma_count from student_hostel where hostel_name = 'gamma';
        if alpha_count < 94 then
                set hn = 'alpha';
                if alpha_count < 47 then
                        set rn = concat('A', alpha_count + 1);
                else
                        set rn = concat('B', alpha_count - 47 + 1);
                end if;
        elseif beta_count < 94 then
                set hn = 'beta';
                if beta_count < 47 then
                        set rn = concat('A', beta_count + 1);price
                else
                        set rn = concat('B', beta_count - 47 + 1);
                end if;
        elseif delta_count < 94 then
                set hn = 'delta';
                if delta_count < 47 then
                        set rn = concat('A', delta_count + 1);
                else
                        set rn = concat('B', delta_count - 47 + 1);
                end if;
        elseif gamma_count < 94 then
                set hn = 'gamma';
                if gamma_count < 47 then
                        set rn = concat('A', gamma_count + 1);
                else
                        set rn = concat('B', gamma_count - 47 + 1);
                end if;
        else
                SIGNAL SQLSTATE '45000'
                SET MESSAGE_TEXT = 'HOSTELS ARE FULL';
        end if;
        insert into student_hostel values(NEW.ID,hn,rn);
    END$$
```



```
MariaDB [university]> insert into student values(23710,'surendra','Music',95,20);
Query OK, 1 row affected (0.03 sec)

MariaDB [university]> select *from student_hostel;
+------------+------------+-------------+
| student_id | hostel_name | room_number |
+------------+------------+-------------+
| 23710      | alpha      | A1          |
+------------+------------+-------------+
1 row in set (0.00 sec)

MariaDB [university]> insert into student values(93710,'Baskey','Biology',92,18);
Query OK, 1 row affected (0.03 sec)

MariaDB [university]> select *from student_hostel;
+------------+------------+-------------+
| student_id | hostel_name | room_number |
+------------+------------+-------------+
| 23710      | alpha      | A1          |
| 93710      | alpha      | A2          |
+------------+------------+-------------+
2 rows in set (0.00 sec)
```

2. Create a table inst_dept by joining two existing table instructor and department.

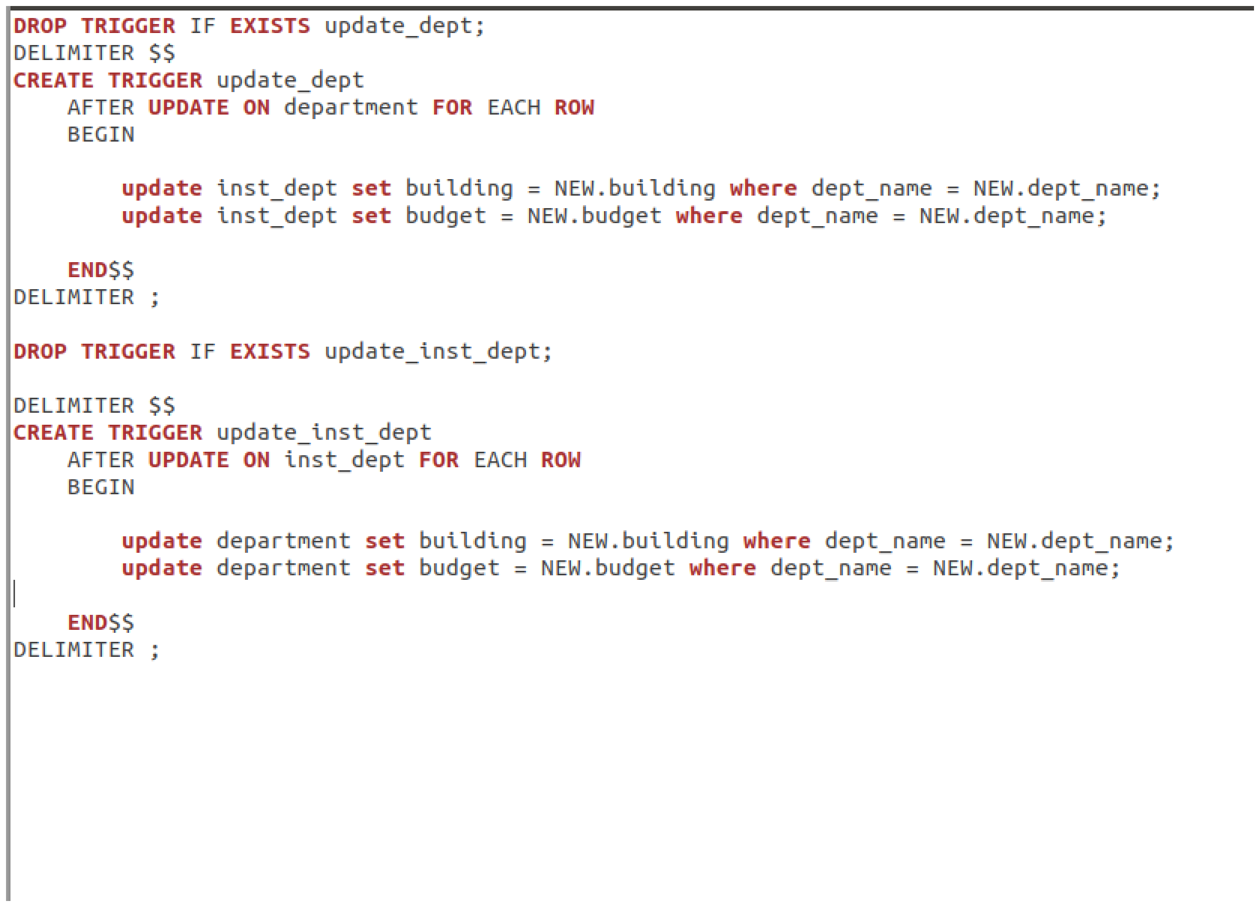


```
MariaDB [university]> create table inst_dept as
    -> (select *from instructor natural join department);
Query OK, 12 rows affected (0.06 sec)
Records: 12  Duplicates: 0  Warnings: 0

MariaDB [university]> select *from inst_dept;
+------------+-------+------------+----------+----------+-----------+
| dept_name  | ID    | name       | salary   | building | budget    |
+------------+-------+------------+----------+----------+-----------+
| Comp. Sci. | 10101 | Srinivasan | 65000.00 | Taylor   | 100000.00 |
| Finance    | 12121 | Wu         | 90000.00 | Painter  | 120000.00 |
| Music      | 15151 | Mozart     | 42000.00 | Packard  |  80000.00 |
| Physics    | 22222 | Einstein   | 95000.00 | Watson   |  70000.00 |
| History    | 32343 | El Said    | 60000.00 | Painter  |  50000.00 |
| Physics    | 33456 | Gold       | 87000.00 | Watson   |  70000.00 |
| Comp. Sci. | 45565 | Katz       | 75000.00 | Taylor   | 100000.00 |
| History    | 58583 | Califieri  | 62000.00 | Painter  |  50000.00 |
| Finance    | 76543 | Sing       | 80000.00 | Painter  | 120000.00 |
| Biology    | 76766 | Crick      | 72000.00 | Watson   |  80000.00 |
| Comp. Sci. | 83821 | Brandt     | 92000.00 | Taylor   | 100000.00 |
| Elec. Eng. | 98345 | Kim        | 80000.00 | Taylor   |  85000.00 |
+------------+-------+------------+----------+----------+-----------+
12 rows in set (0.00 sec)

MariaDB [university]>
```

3. Write a trigger for stopping any update anomalies on budget and building in your

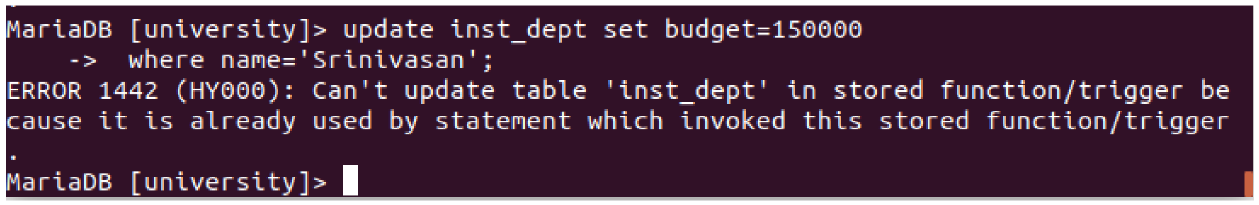
Database.

a. User can update budget and/or building of any individual tuple in inst_dept table or department table. Any update on budget or building in any tuple in any table should automatically reflects on budget or building of corresponding department for all redundant occurrence of it (even in other tables).



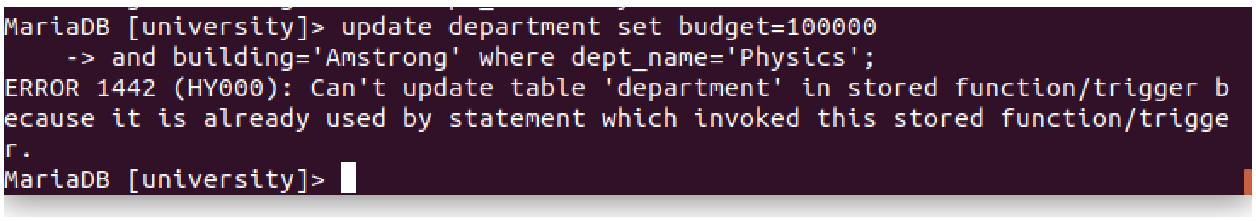
```
DROP TRIGGER IF EXISTS update_dept;
DELIMITER $$
CREATE TRIGGER update_dept
    AFTER UPDATE ON department FOR EACH ROW
    BEGIN

        update inst_dept set building = NEW.building where dept_name = NEW.dept_name;
        update inst_dept set budget = NEW.budget where dept_name = NEW.dept_name;

    END$$
DELIMITER ;

DROP TRIGGER IF EXISTS update_inst_dept;

DELIMITER $$
CREATE TRIGGER update_inst_dept
    AFTER UPDATE ON inst_dept FOR EACH ROW
    BEGIN

        update department set building = NEW.building where dept_name = NEW.dept_name;
        update department set budget = NEW.budget where dept_name = NEW.dept_name;

    END$$
DELIMITER ;
```

b. Validate your trigger by executing following DML
i.
Update budget to ‘1,50,000’ of instructor whose name is “Srinivasan” in inst_dept table.



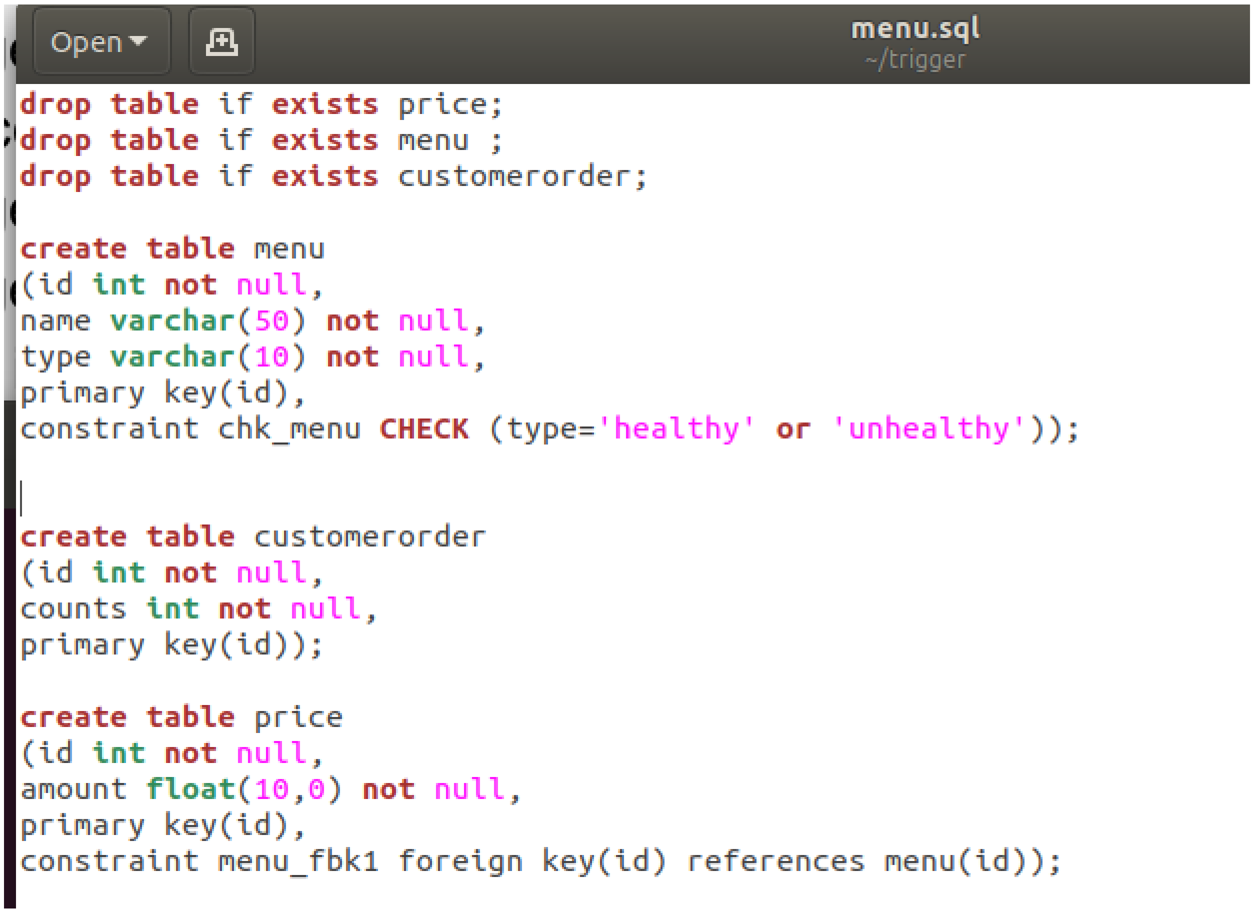
```
MariaDB [university]> update inst_dept set budget=150000
    ->  where name='Srinivasan';
ERROR 1442 (HY000): Can't update table 'inst_dept' in stored function/trigger be
cause it is already used by statement which invoked this stored function/trigger
.
MariaDB [university]> 
```

ii.
Update budget and building of “Physics” department to ‘1,00,000’ and ‘Amstrong’B. Canteen database



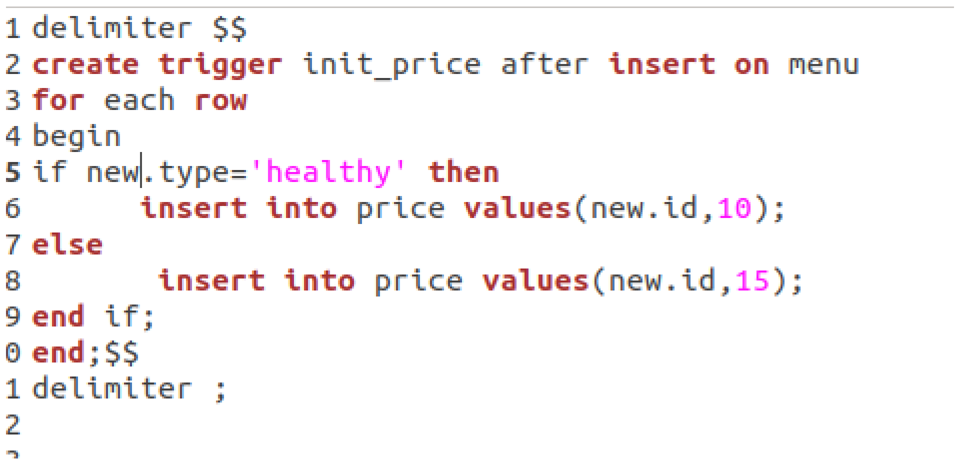
```
MariaDB [university]> update department set budget=100000
    -> and building='Amstrong' where dept_name='Physics';
ERROR 1442 (HY000): Can't update table 'department' in stored function/trigger b
ecause it is already used by statement which invoked this stored function/trigge
r.
MariaDB [university]> 
```

1. Create a database canteen . Create a table menu with attributes id int not null , and name
varchar(50) not null, type that can take value between ‘ healthy’ , and ‘ unhealthy’ . Create another table customerorder with attribute id not null , and count int not null . Create a table price that will contain id of a dish in the menu and amount float .

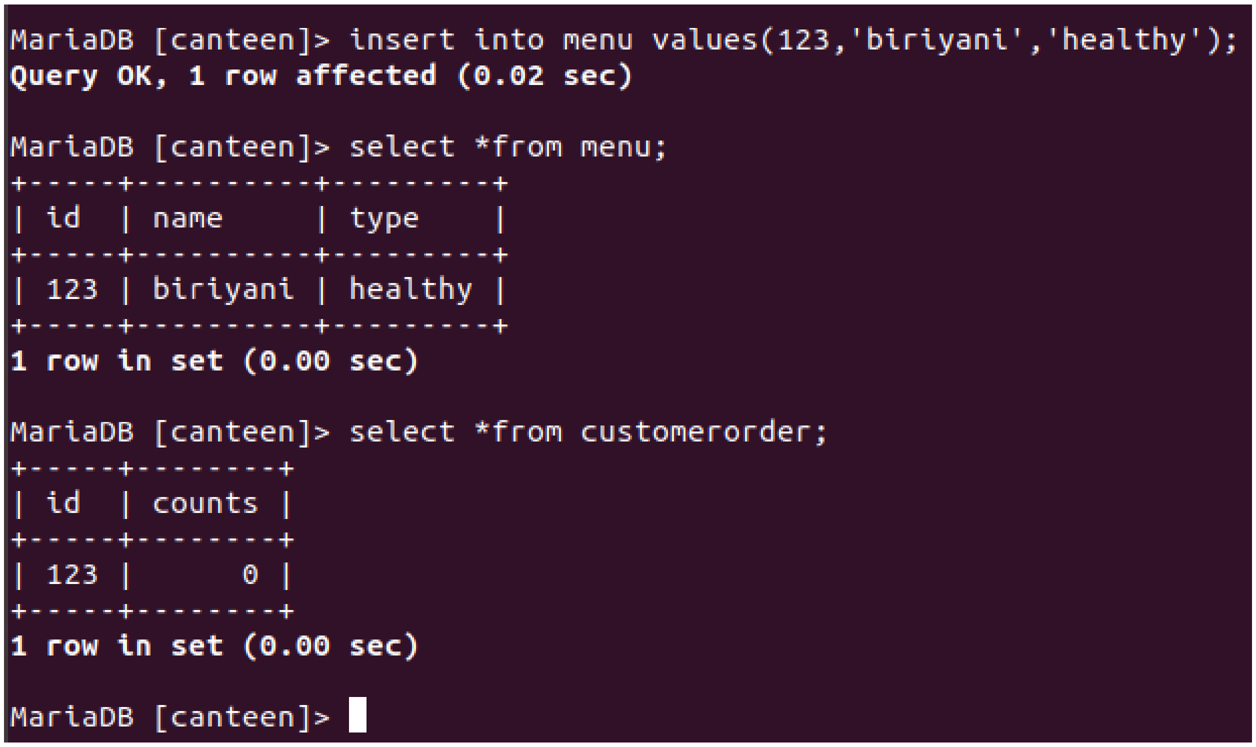


```
drop table if exists price;
drop table if exists menu ;
drop table if exists customerorder;

create table menu
(id int not null,
name varchar(50) not null,
type varchar(10) not null,
primary key(id),
constraint chk_menu CHECK (type='healthy' or 'unhealthy'));


create table customerorder
(id int not null,
counts int not null,
primary key(id));

create table price
(id int not null,
amount float(10,0) not null,
primary key(id),
constraint menu_fbk1 foreign key(id) references menu(id));
```

2. Create a trigger init t hat will initiate the count to zero for each entry in the menu table. Insert one record to menu table, and show how init works.

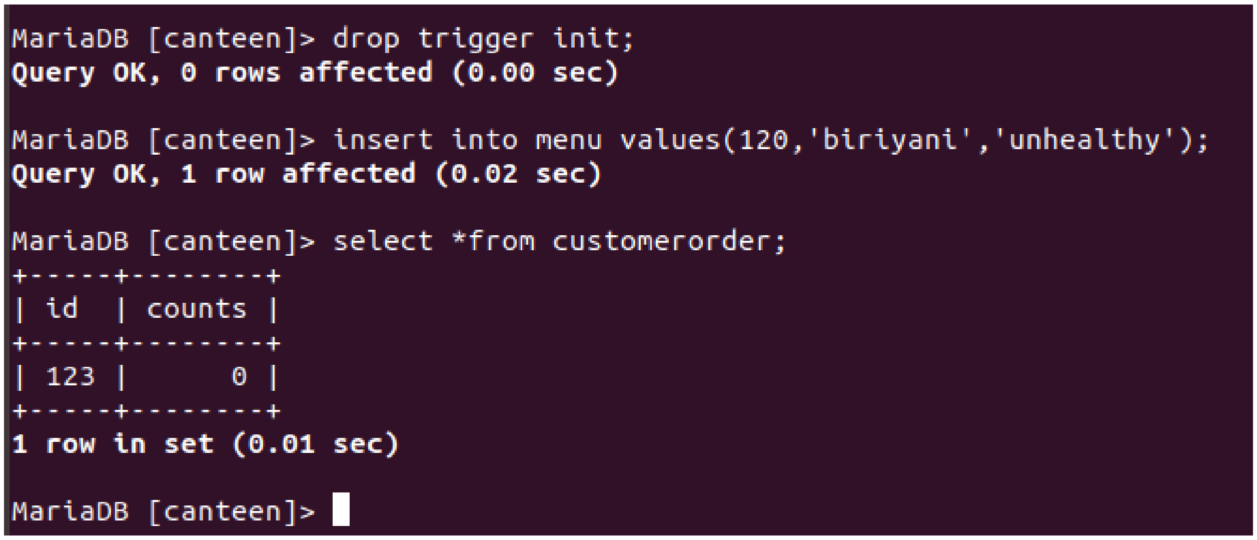


```
delimiter $$
create trigger init_price after insert on menu
for each row
begin
if new.type='healthy' then
      insert into price values(new.id,10);
else
       insert into price values(new.id,15);
end if;
end;$$
delimiter ;
```



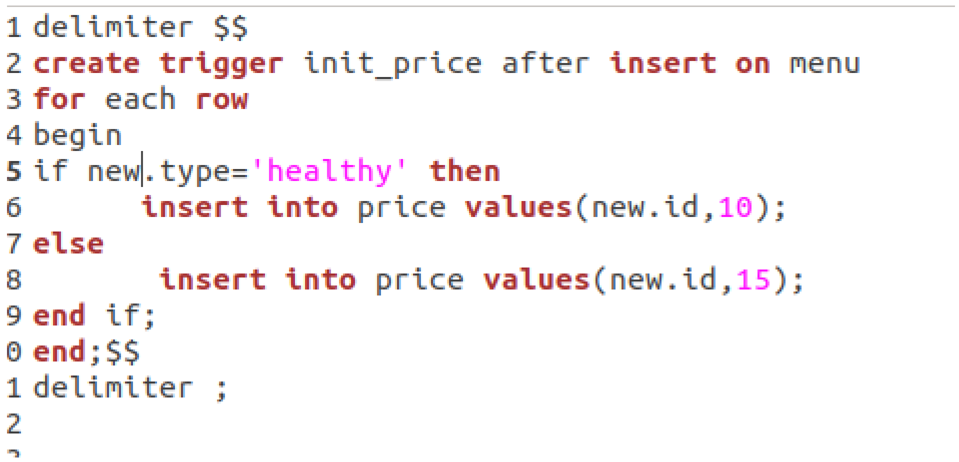
```
MariaDB [canteen]> insert into menu values(123,'biriyani','healthy');
Query OK, 1 row affected (0.02 sec)

MariaDB [canteen]> select *from menu;
+-----+----------+---------+
| id  | name     | type    |
+-----+----------+---------+
| 123 | biriyani | healthy |
+-----+----------+---------+
1 row in set (0.00 sec)

MariaDB [canteen]> select *from customerorder;
+-----+--------+
| id  | counts |
+-----+--------+
| 123 |      0 |
+-----+--------+
1 row in set (0.00 sec)

MariaDB [canteen]>
```

3. Drop the trigger init and show the effect.

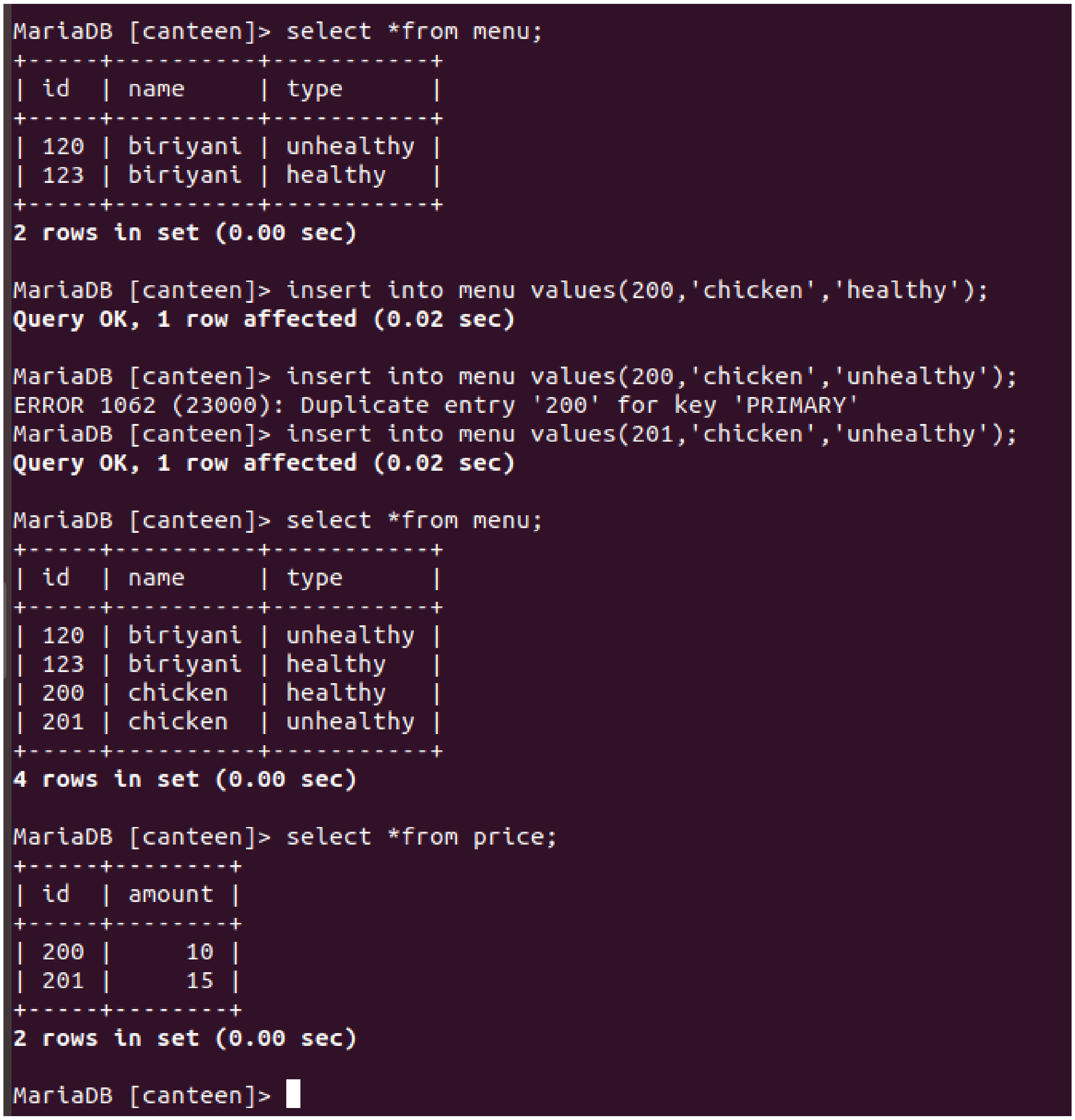


```
MariaDB [canteen]> drop trigger init;
Query OK, 0 rows affected (0.00 sec)

MariaDB [canteen]> insert into menu values(120,'biriyani','unhealthy');
Query OK, 1 row affected (0.02 sec)

MariaDB [canteen]> select *from customerorder;
+-----+--------+
| id  | counts |
+-----+--------+
| 123 |      0 |
+-----+--------+
1 row in set (0.01 sec)

MariaDB [canteen]>
```

4. Create a trigger init_price that will check the type of the dish in the menu, and will automatically initialize a price in the correct table. For healthy food price is 10, and for unhealthy foods price is 15.

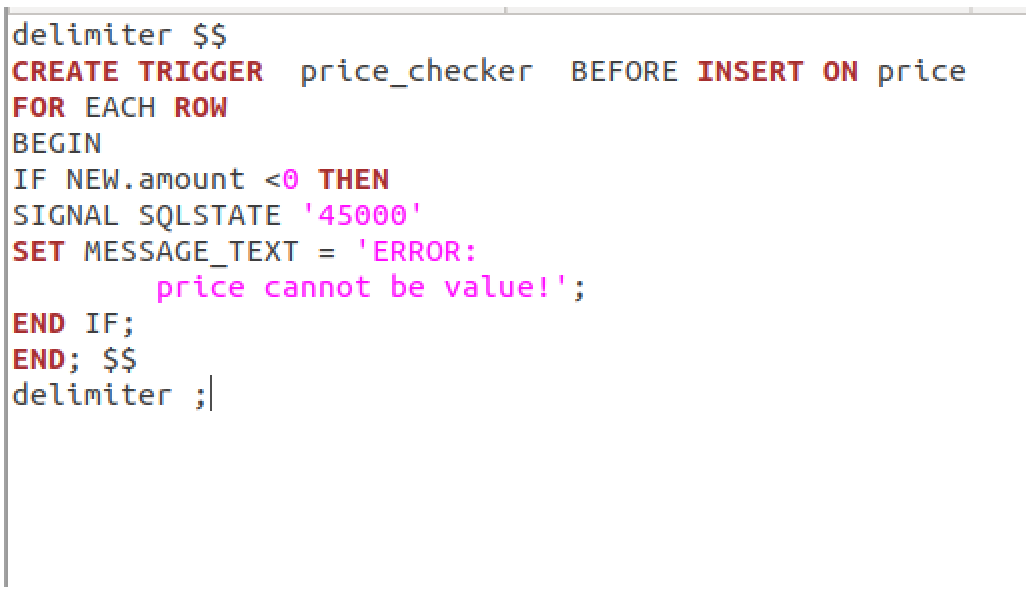


```
delimiter $$
create trigger init_price after insert on menu
for each row
begin
if new.type='healthy' then
      insert into price values(new.id,10);
else
       insert into price values(new.id,15);
end if;
end;$$
delimiter ;

```

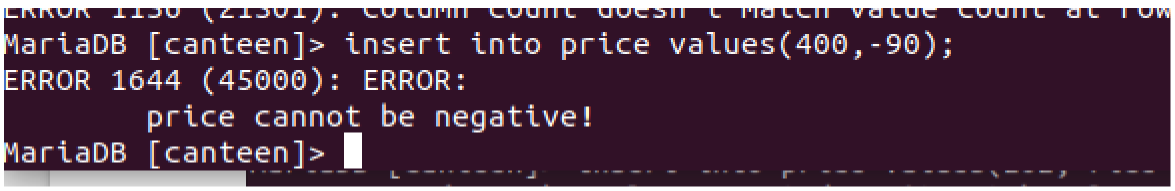


```
MariaDB [canteen]> select *from menu;
+-----+----------+-----------+
| id  | name     | type      |
+-----+----------+-----------+
| 120 | biriyani | unhealthy |
| 123 | biriyani | healthy   |
+-----+----------+-----------+
2 rows in set (0.00 sec)

MariaDB [canteen]> insert into menu values(200,'chicken','healthy');
Query OK, 1 row affected (0.02 sec)

MariaDB [canteen]> insert into menu values(200,'chicken','unhealthy');
ERROR 1062 (23000): Duplicate entry '200' for key 'PRIMARY'
MariaDB [canteen]> insert into menu values(201,'chicken','unhealthy');
Query OK, 1 row affected (0.02 sec)

MariaDB [canteen]> select *from menu;
+-----+----------+-----------+
| id  | name     | type      |
+-----+----------+-----------+
| 120 | biriyani | unhealthy |
| 123 | biriyani | healthy   |
| 200 | chicken  | healthy   |
| 201 | chicken  | unhealthy |
+-----+----------+-----------+
4 rows in set (0.00 sec)

MariaDB [canteen]> select *from price;
+-----+--------+
| id  | amount |
+-----+--------+
| 200 |     10 |
| 201 |     15 |
+-----+--------+
2 rows in set (0.00 sec)

MariaDB [canteen]> 
```

5. Create trigger price_checker that will raise error, and display message ‘price can not be negative’ if accidentally someone tries to enter a negative price for some dish.

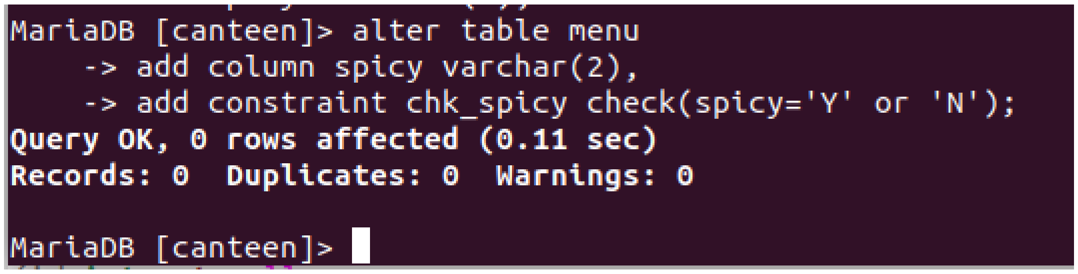


```
delimiter $$
CREATE TRIGGER  price_checker  BEFORE INSERT ON price
FOR EACH ROW
BEGIN
IF NEW.amount <0 THEN
SIGNAL SQLSTATE '45000'
SET MESSAGE_TEXT = 'ERROR:
        price cannot be value!';
END IF;
END; $$
delimiter ;
```

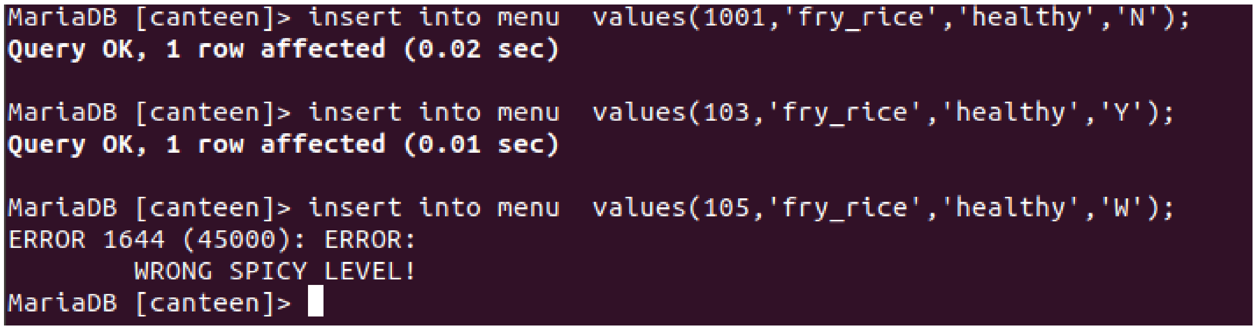


```
ERROR 1136 (21S01): Column count doesn't match value count at row
MariaDB [canteen]> insert into price values(400,-90);
ERROR 1644 (45000): ERROR:
        price cannot be negative!
MariaDB [canteen]>
```

6. Modify the table menu to add an attribute spicy which can take value either ‘Y’ or ‘ N’ . Create the attribute spicy as varchar but write a trigger s picy_checker to check the validity. If invalid then raise an error and throw a message



```
MariaDB [canteen]> alter table menu
    -> add column spicy varchar(2),
    -> add constraint chk_spicy check(spicy='Y' or 'N');
Query OK, 0 rows affected (0.11 sec)
Records: 0  Duplicates: 0  Warnings: 0

MariaDB [canteen]>
```



```
MariaDB [canteen]> insert into menu  values(1001,'fry_rice','healthy','N');
Query OK, 1 row affected (0.02 sec)

MariaDB [canteen]> insert into menu  values(103,'fry_rice','healthy','Y');
Query OK, 1 row affected (0.01 sec)

MariaDB [canteen]> insert into menu  values(105,'fry_rice','healthy','W');
ERROR 1644 (45000): ERROR:
        WRONG SPICY LEVEL!
MariaDB [canteen]>
```