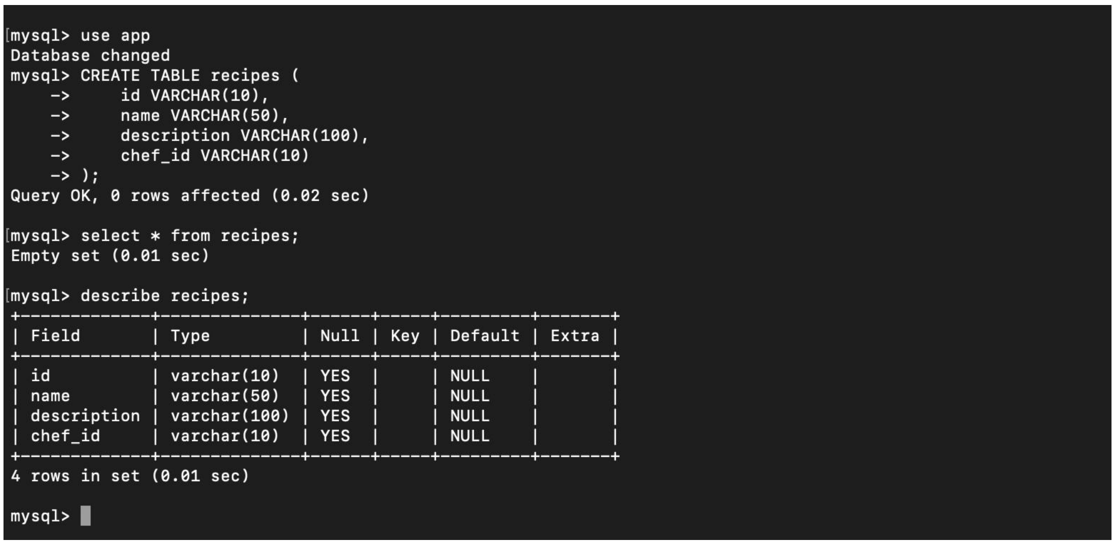
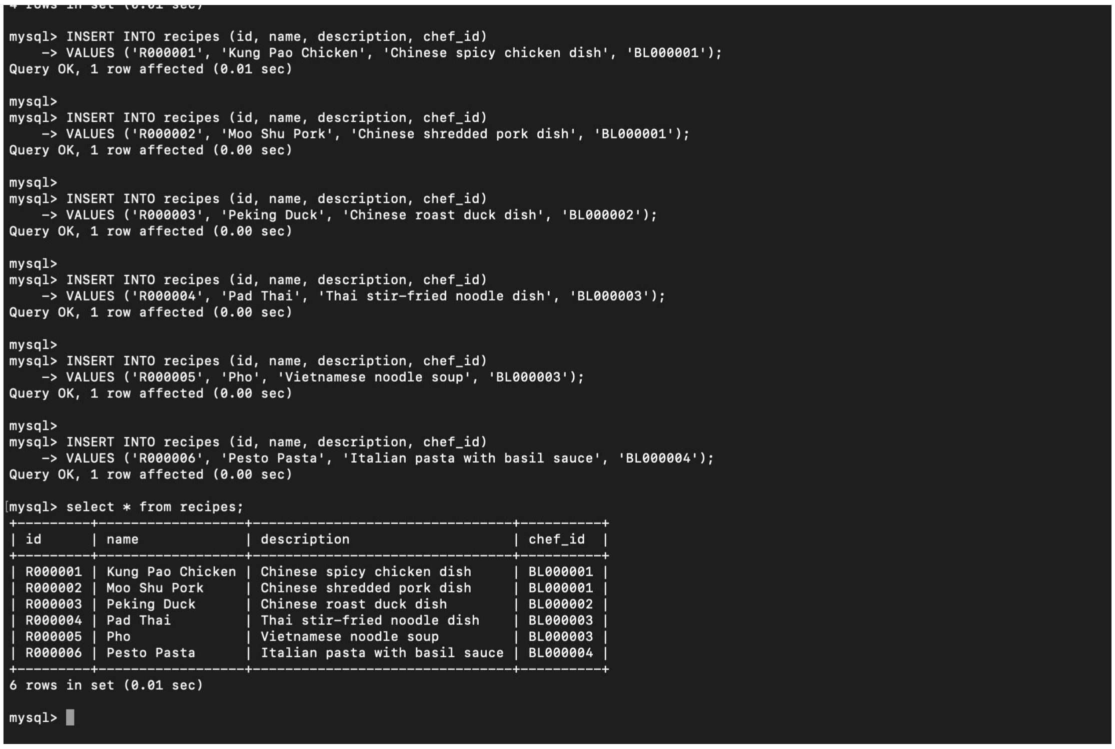
Week- 4

Name: Ronit Kumar Reg: RA2111032010009 Section: T2

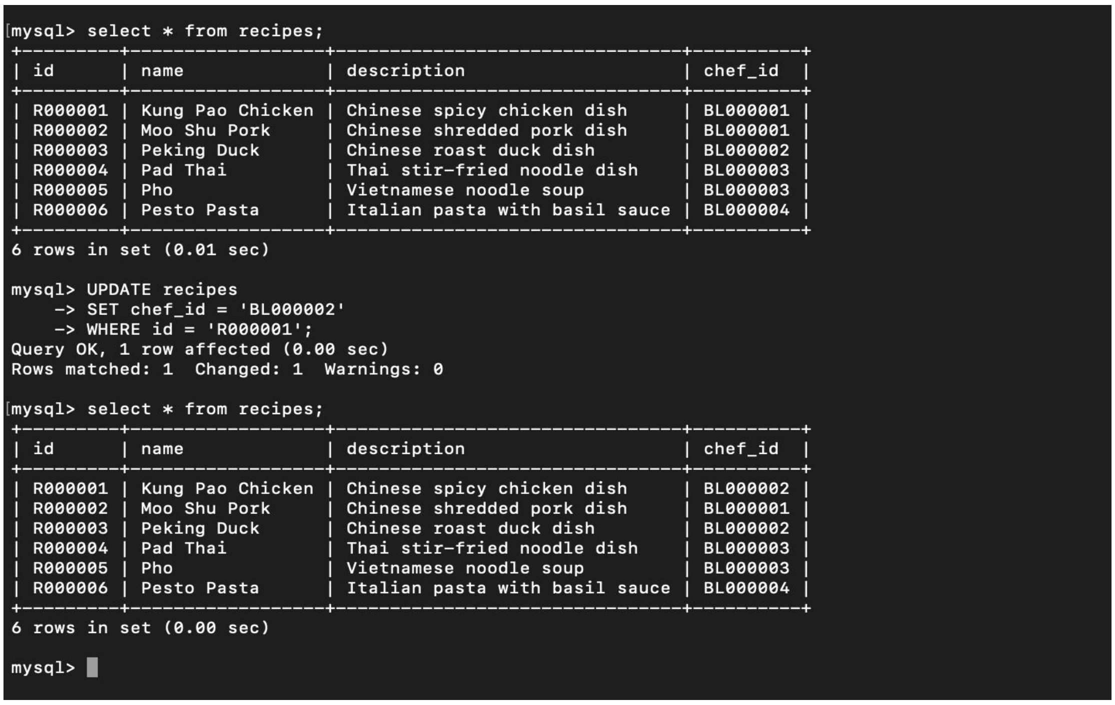
1. Create the below table and execute the insert, update and the below select statements.



```
mysql> use app
Database changed
mysql> CREATE TABLE recipes (
    ->     id VARCHAR(10),
    ->     name VARCHAR(50),
    ->     description VARCHAR(100),
    ->     chef_id VARCHAR(10)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> select * from recipes;
Empty set (0.01 sec)

mysql> describe recipes;
+-------------+--------------+------+-----+---------+-------+
| Field       | Type         | Null | Key | Default | Extra |
+-------------+--------------+------+-----+---------+-------+
| id          | varchar(10)  | YES  |     | NULL    |       |
| name        | varchar(50)  | YES  |     | NULL    |       |
| description | varchar(100) | YES  |     | NULL    |       |
| chef_id     | varchar(10)  | YES  |     | NULL    |       |
+-------------+--------------+------+-----+---------+-------+
4 rows in set (0.01 sec)

mysql> 
```

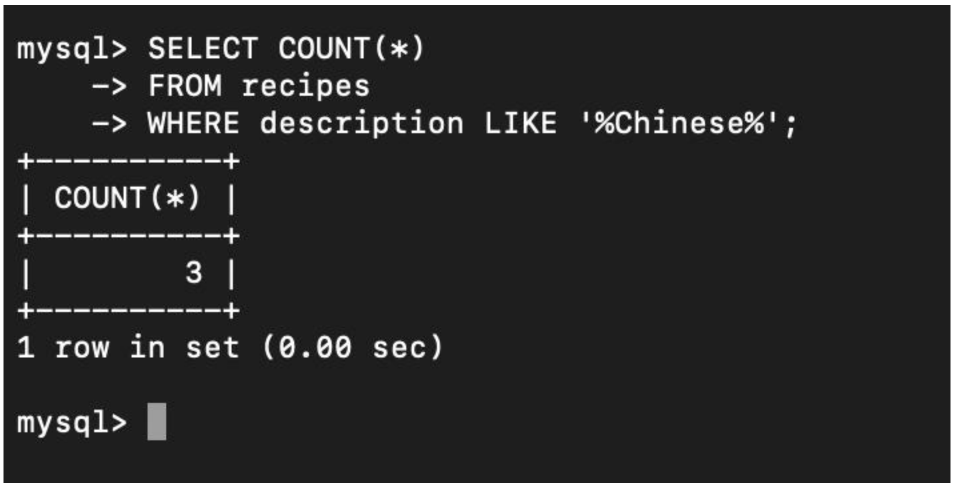


```
4 rows in set (0.01 sec)

mysql> INSERT INTO recipes (id, name, description, chef_id)
    -> VALUES ('R000001', 'Kung Pao Chicken', 'Chinese spicy chicken dish', 'BL000001');
Query OK, 1 row affected (0.01 sec)

mysql>
mysql> INSERT INTO recipes (id, name, description, chef_id)
    -> VALUES ('R000002', 'Moo Shu Pork', 'Chinese shredded pork dish', 'BL000001');
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO recipes (id, name, description, chef_id)
    -> VALUES ('R000003', 'Peking Duck', 'Chinese roast duck dish', 'BL000002');
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO recipes (id, name, description, chef_id)
    -> VALUES ('R000004', 'Pad Thai', 'Thai stir-fried noodle dish', 'BL000003');
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO recipes (id, name, description, chef_id)
    -> VALUES ('R000005', 'Pho', 'Vietnamese noodle soup', 'BL000003');
Query OK, 1 row affected (0.00 sec)

mysql>
mysql> INSERT INTO recipes (id, name, description, chef_id)
    -> VALUES ('R000006', 'Pesto Pasta', 'Italian pasta with basil sauce', 'BL000004');
Query OK, 1 row affected (0.00 sec)

mysql> select * from recipes;
+---------+------------------+--------------------------------+----------+
| id      | name             | description                    | chef_id  |
+---------+------------------+--------------------------------+----------+
| R000001 | Kung Pao Chicken | Chinese spicy chicken dish     | BL000001 |
| R000002 | Moo Shu Pork     | Chinese shredded pork dish     | BL000001 |
| R000003 | Peking Duck      | Chinese roast duck dish        | BL000002 |
| R000004 | Pad Thai         | Thai stir-fried noodle dish    | BL000003 |
| R000005 | Pho              | Vietnamese noodle soup         | BL000003 |
| R000006 | Pesto Pasta      | Italian pasta with basil sauce | BL000004 |
+---------+------------------+--------------------------------+----------+
6 rows in set (0.01 sec)

mysql> 
```



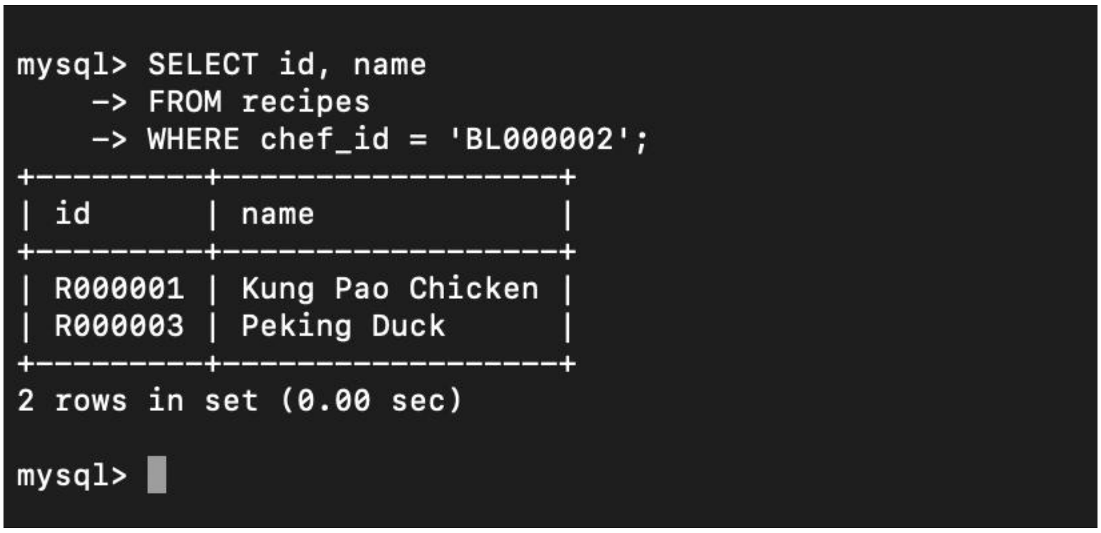
```
mysql> select * from recipes;
+---------+-----------------+--------------------------------+----------+
| id      | name            | description                    | chef_id  |
+---------+-----------------+--------------------------------+----------+
| R000001 | Kung Pao Chicken | Chinese spicy chicken dish     | BL000001 |
| R000002 | Moo Shu Pork    | Chinese shredded pork dish     | BL000001 |
| R000003 | Peking Duck     | Chinese roast duck dish        | BL000002 |
| R000004 | Pad Thai        | Thai stir-fried noodle dish    | BL000003 |
| R000005 | Pho             | Vietnamese noodle soup         | BL000003 |
| R000006 | Pesto Pasta     | Italian pasta with basil sauce | BL000004 |
+---------+-----------------+--------------------------------+----------+
6 rows in set (0.01 sec)

mysql> UPDATE recipes
    -> SET chef_id = 'BL000002'
    -> WHERE id = 'R000001';
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from recipes;
+---------+-----------------+--------------------------------+----------+
| id      | name            | description                    | chef_id  |
+---------+-----------------+--------------------------------+----------+
| R000001 | Kung Pao Chicken | Chinese spicy chicken dish     | BL000002 |
| R000002 | Moo Shu Pork    | Chinese shredded pork dish     | BL000001 |
| R000003 | Peking Duck     | Chinese roast duck dish        | BL000002 |
| R000004 | Pad Thai        | Thai stir-fried noodle dish    | BL000003 |
| R000005 | Pho             | Vietnamese noodle soup         | BL000003 |
| R000006 | Pesto Pasta     | Italian pasta with basil sauce | BL000004 |
+---------+-----------------+--------------------------------+----------+
6 rows in set (0.00 sec)

mysql> 
```

i) Write a query to display the total number of recipes available with the description "Chinese"



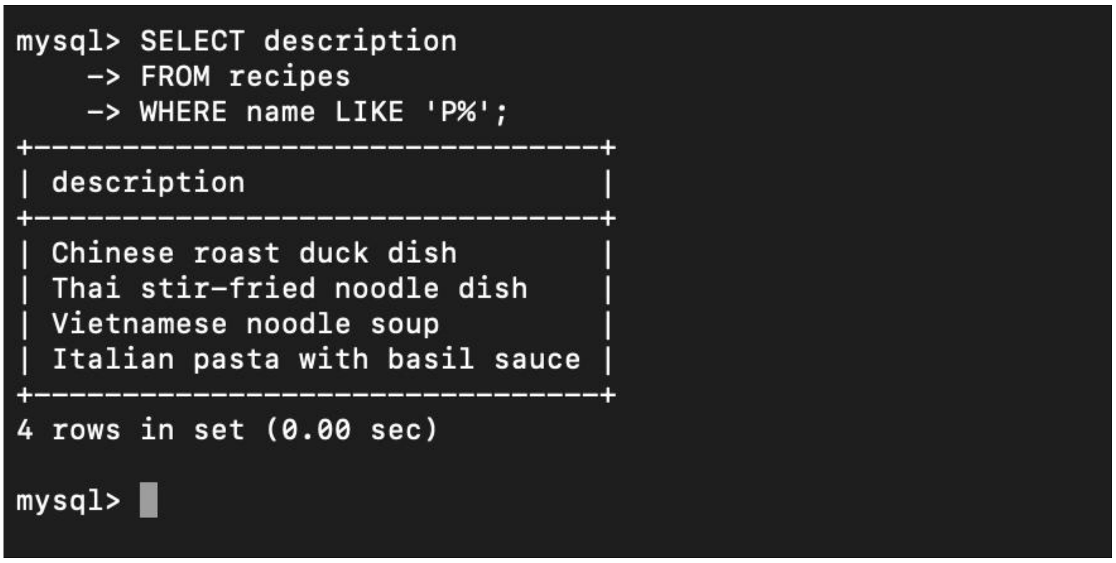
```
mysql> SELECT COUNT(*)
    -> FROM recipes
    -> WHERE description LIKE '%Chinese%';
+----------+
| COUNT(*) |
+----------+
|        3 |
+----------+
1 row in set (0.00 sec)

mysql> 
```

ii) Write a query to display the id, name of the recipes with chef_id 'BL000002'.



```
mysql> SELECT id, name
    -> FROM recipes
    -> WHERE chef_id = 'BL000002';
+---------+-----------------+
| id      | name            |
+---------+-----------------+
| R000001 | Kung Pao Chicken |
| R000003 | Peking Duck     |
+---------+-----------------+
2 rows in set (0.00 sec)

mysql> 
```

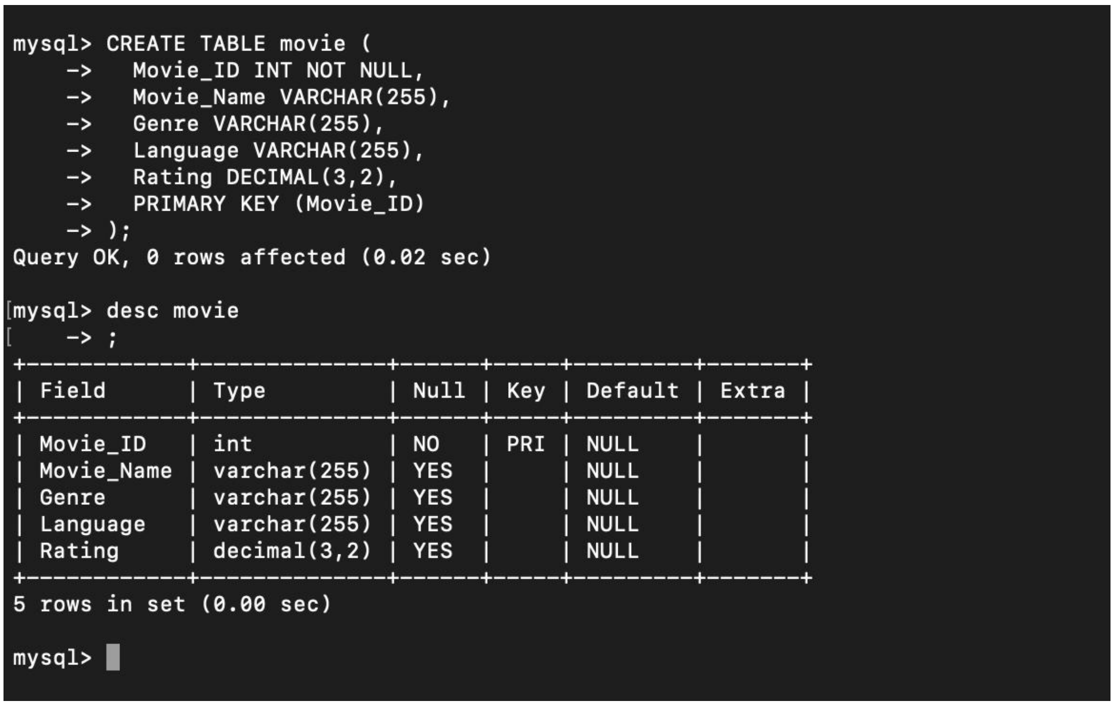
iii) Write a query to display the description of the recipes whose name begins with 'P'.



```
mysql> SELECT description
    -> FROM recipes
    -> WHERE name LIKE 'P%';
+-------------------------------+
| description                   |
+-------------------------------+
| Chinese roast duck dish       |
| Thai stir-fried noodle dish   |
| Vietnamese noodle soup        |
| Italian pasta with basil sauce |
+-------------------------------+
4 rows in set (0.00 sec)

mysql> 
```

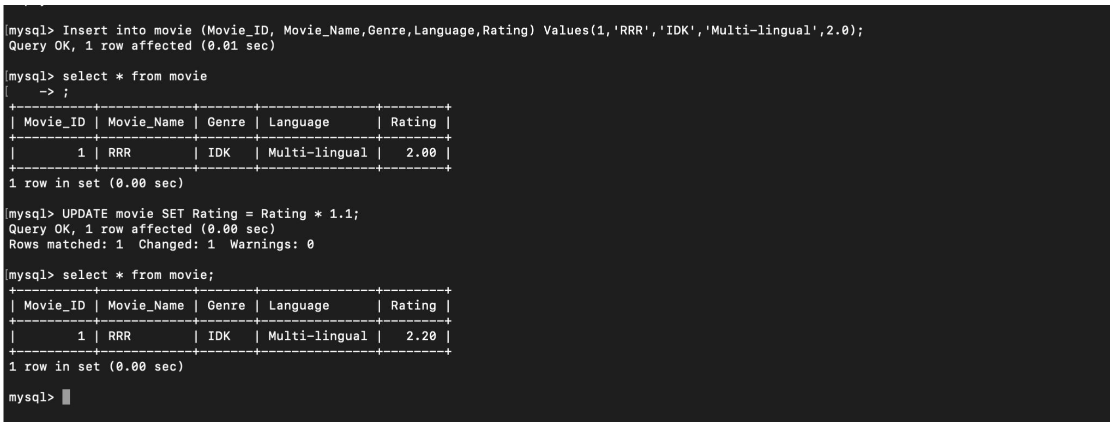
2. Create a table movie of the below structure and assume data types.Movie_ID,

Movie_Name, Genre, Language, Rating ,Do the following queries



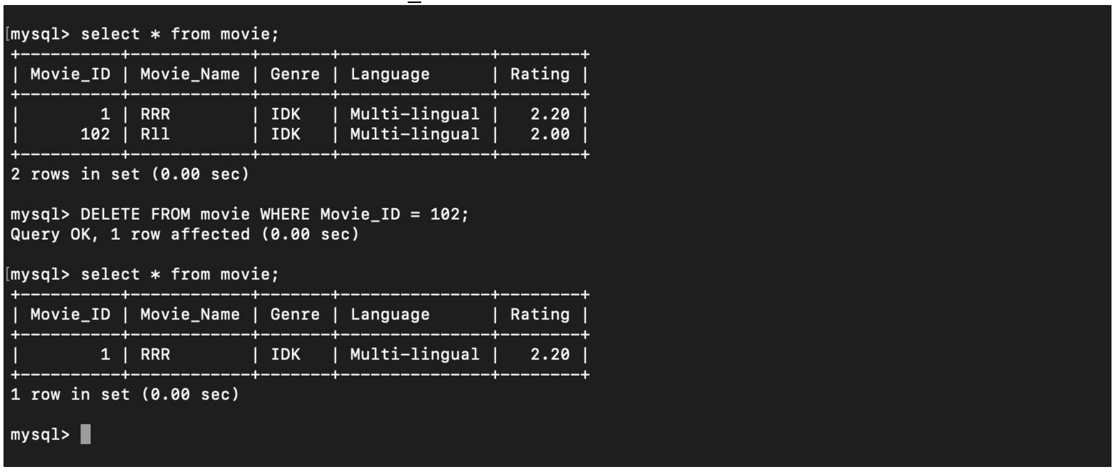
```
mysql> CREATE TABLE movie (
    ->   Movie_ID INT NOT NULL,
    ->   Movie_Name VARCHAR(255),
    ->   Genre VARCHAR(255),
    ->   Language VARCHAR(255),
    ->   Rating DECIMAL(3,2),
    ->   PRIMARY KEY (Movie_ID)
    -> );
Query OK, 0 rows affected (0.02 sec)

mysql> desc movie
    -> ;
+------------+--------------+------+-----+---------+-------+
| Field      | Type         | Null | Key | Default | Extra |
+------------+--------------+------+-----+---------+-------+
| Movie_ID   | int          | NO   | PRI | NULL    |       |
| Movie_Name | varchar(255) | YES  |     | NULL    |       |
| Genre      | varchar(255) | YES  |     | NULL    |       |
| Language   | varchar(255) | YES  |     | NULL    |       |
| Rating     | decimal(3,2) | YES  |     | NULL    |       |
+------------+--------------+------+-----+---------+-------+
5 rows in set (0.00 sec)

mysql> 
```

a. Update the movies rating by 10% and display it



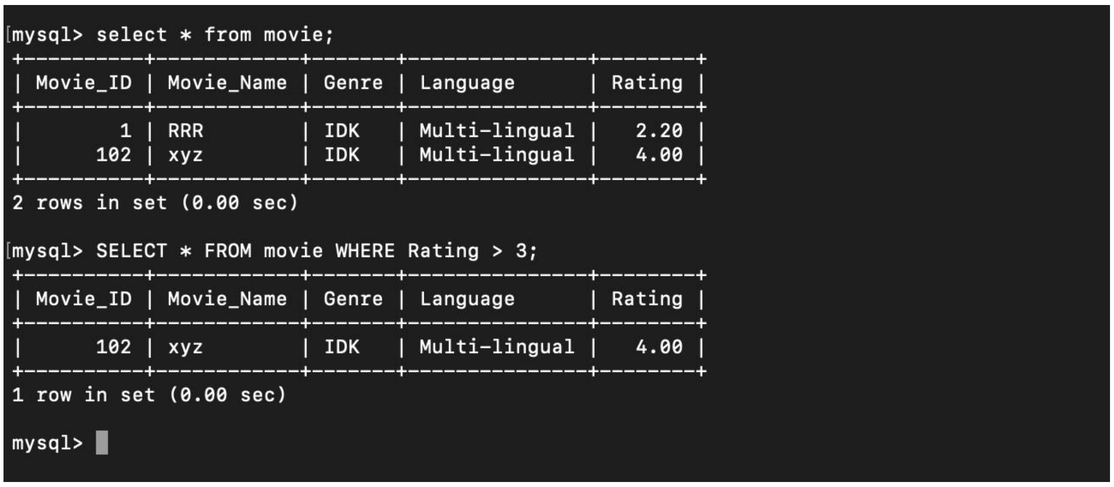
```
mysql> Insert into movie (Movie_ID, Movie_Name,Genre,Language,Rating) Values(1,'RRR','IDK','Multi-lingual',2.0);
Query OK, 1 row affected (0.01 sec)

mysql> select * from movie
    -> ;
+----------+------------+-------+---------------+--------+
| Movie_ID | Movie_Name | Genre | Language      | Rating |
+----------+------------+-------+---------------+--------+
|        1 | RRR        | IDK   | Multi-lingual |   2.00 |
+----------+------------+-------+---------------+--------+
1 row in set (0.00 sec)

mysql> UPDATE movie SET Rating = Rating * 1.1;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from movie;
+----------+------------+-------+---------------+--------+
| Movie_ID | Movie_Name | Genre | Language      | Rating |
+----------+------------+-------+---------------+--------+
|        1 | RRR        | IDK   | Multi-lingual |   2.20 |
+----------+------------+-------+---------------+--------+
1 row in set (0.00 sec)

mysql> 
```

b. Delete the movies with movie_id 102



```
mysql> select * from movie;
+----------+------------+-------+---------------+--------+
| Movie_ID | Movie_Name | Genre | Language      | Rating |
+----------+------------+-------+---------------+--------+
|        1 | RRR        | IDK   | Multi-lingual |   2.20 |
|      102 | R11        | IDK   | Multi-lingual |   2.00 |
+----------+------------+-------+---------------+--------+
2 rows in set (0.00 sec)

mysql> DELETE FROM movie WHERE Movie_ID = 102;
Query OK, 1 row affected (0.00 sec)

mysql> select * from movie;
+----------+------------+-------+---------------+--------+
| Movie_ID | Movie_Name | Genre | Language      | Rating |
+----------+------------+-------+---------------+--------+
|        1 | RRR        | IDK   | Multi-lingual |   2.20 |
+----------+------------+-------+---------------+--------+
1 row in set (0.00 sec)

mysql> 
```

c. Select movies whose rating is more than 3.



```
[mysql> select * from movie;
+----------+------------+-------+---------------+--------+
| Movie_ID | Movie_Name | Genre | Language      | Rating |
+----------+------------+-------+---------------+--------+
|        1 | RRR        | IDK   | Multi-lingual |   2.20 |
|      102 | xyz        | IDK   | Multi-lingual |   4.00 |
+----------+------------+-------+---------------+--------+
2 rows in set (0.00 sec)

[mysql> SELECT * FROM movie WHERE Rating > 3;
+----------+------------+-------+---------------+--------+
| Movie_ID | Movie_Name | Genre | Language      | Rating |
+----------+------------+-------+---------------+--------+
|      102 | xyz        | IDK   | Multi-lingual |   4.00 |
+----------+------------+-------+---------------+--------+
1 row in set (0.00 sec)

mysql> 
```

3. Create a course database with the following fields Product(ID, Prod_name,

Supplier_id,Unit_price,Package,OrderID),OrderItem(ID,Order_id,Product_id,Unit_price, Quantity) using Foreign key
a. Display the total quantity of every product in the stock
b. Sort the Unit_price based on the supplier_id
c. Display the Product_name along with order_id and supplier_id

```
CREATE TABLE OrderItem (
ID INT PRIMARY KEY,
Order_id INT,
Product_id INT,
Unit_price DECIMAL(10,2),
Quantity INT,
FOREIGN KEY (Order_id) REFERENCES "Order"(Order_id),
FOREIGN KEY (Product_id) REFERENCES Product(ID)
);

CREATE TABLE Product (
ID INT PRIMARY KEY,
Prod_name VARCHAR(255),
Supplier_id INT,
Unit_price DECIMAL(10,2),
Package VARCHAR(50),
OrderID INT,
FOREIGN KEY (Supplier_id) REFERENCES Supplier(ID),
FOREIGN KEY (OrderID) REFERENCES OrderItem(Order_id)
);
INSERT INTO Product (ID, Prod_name, Supplier_id, Unit_price, Package, OrderID)
VALUES (1, 'TABLE', 1001, 500.57, 'PACK1', 'ORD1'),
(2, 'CHAIR', 2002, 250.60, 'PACK2', 'ORD2'),
(3, 'BLACKBOARD', 1003, 1000.99, 'PACK3', 'ORD3');
INSERT INTO OrderItem (ID, Order_id, Product_id, Unit_price, Quantity)
VALUES (1, 'ORD1', 1, 500.57, 20),
(2, 'ORD2', 2, 250.60, 10),
(3, 'ORD3', 3, 1000.99, 35);
```

4. Write a SQL lite3 statement to create a table named as job including columns job_id,job_title,Min-salary,Max_salary.job_id column does not contain any duplicate value at the time of insertion

```
import sqlite3
conn = sqlite.3.connect('app.db')
print "opened db succesfully";
conn.execute('''
CREATE TABLE job (
  job_id INTEGER PRIMARY KEY,
  job_title TEXT NOT NULL,
  Min_salary REAL,
  Max_salary REAL,
  UNIQUE(job_id)
);
''')
print "Table created succesfully";
conn.close()
```

5. Write a SQL lite3 statement to create a table names as job_history including
columns
employee_id, start_date, end_date, job_id and department_id and make sure that, the
employee_id column does not contain any duplicate value at the time of insertion and
the
foreign key column job_id contain only those values which are exists in the jobs table.

```
import sqlite3
conn = sqlite.3.connect('app.db')
print "opened db succesfully";
conn.execute('''
CREATE TABLE job_history (
  employee_id INTEGER NOT NULL,
  start_date TEXT NOT NULL,
  end_date TEXT NOT NULL,
  job_id TEXT NOT NULL,
  department_id INTEGER NOT NULL,
  PRIMARY KEY (employee_id, start_date),
  FOREIGN KEY (job_id) REFERENCES jobs(job_id)
);
''')
print "Table created succesfully";
conn.close()
```