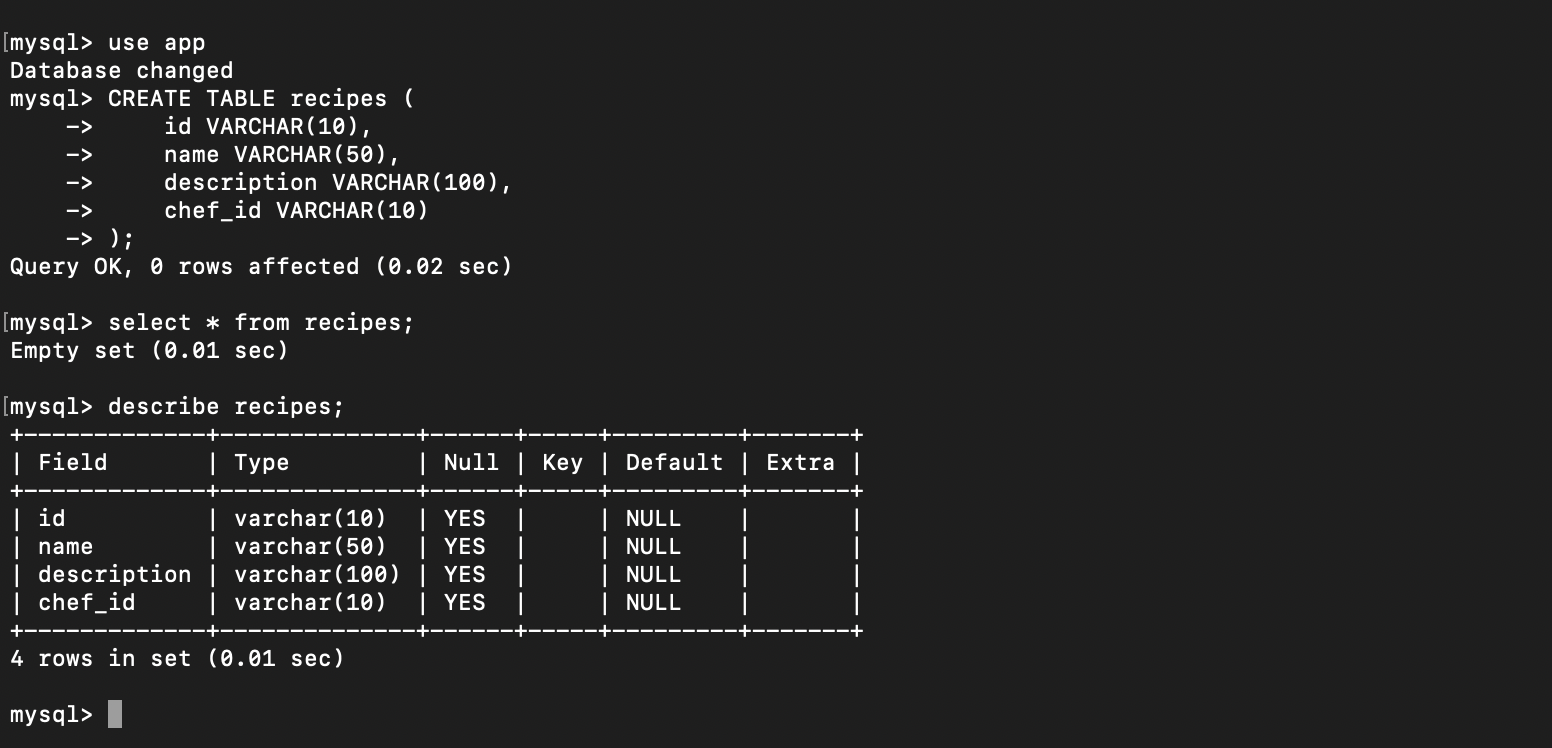
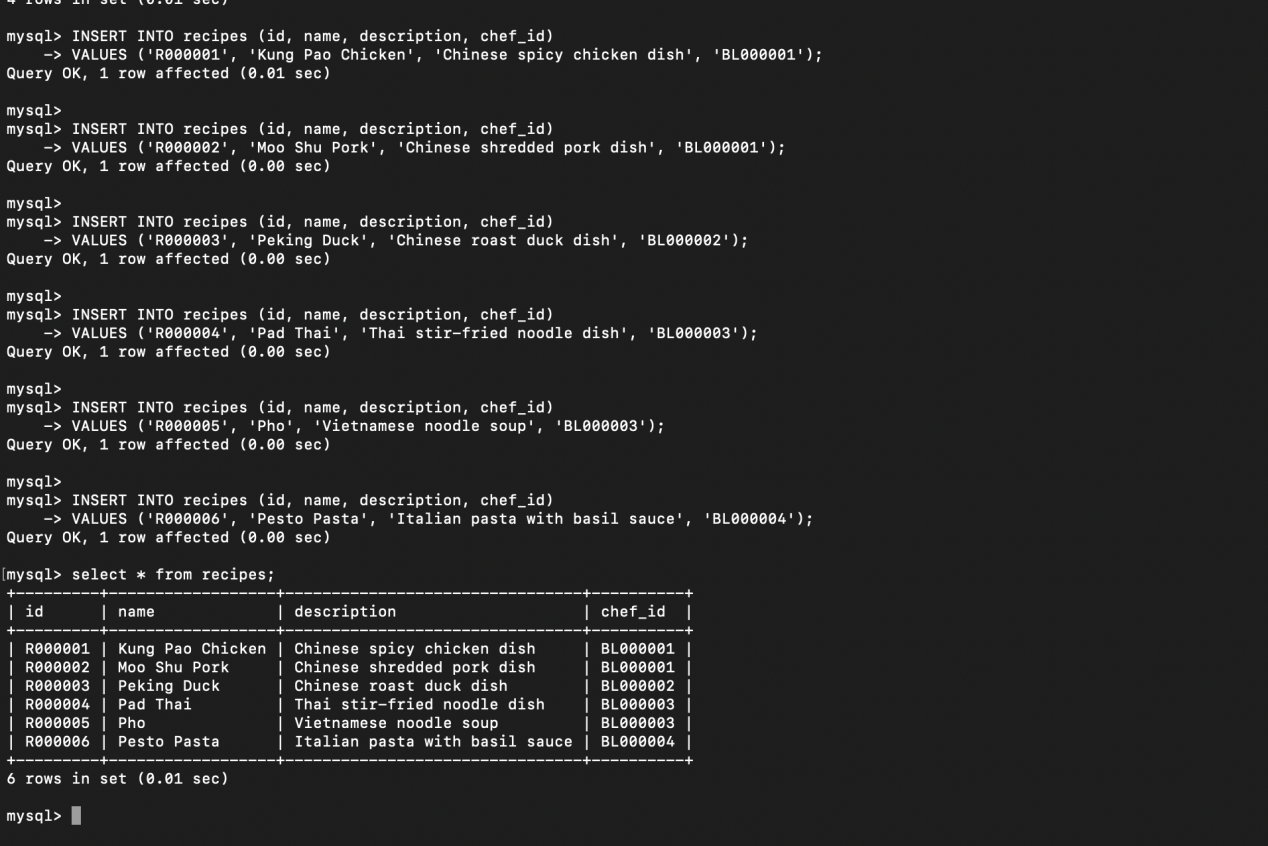
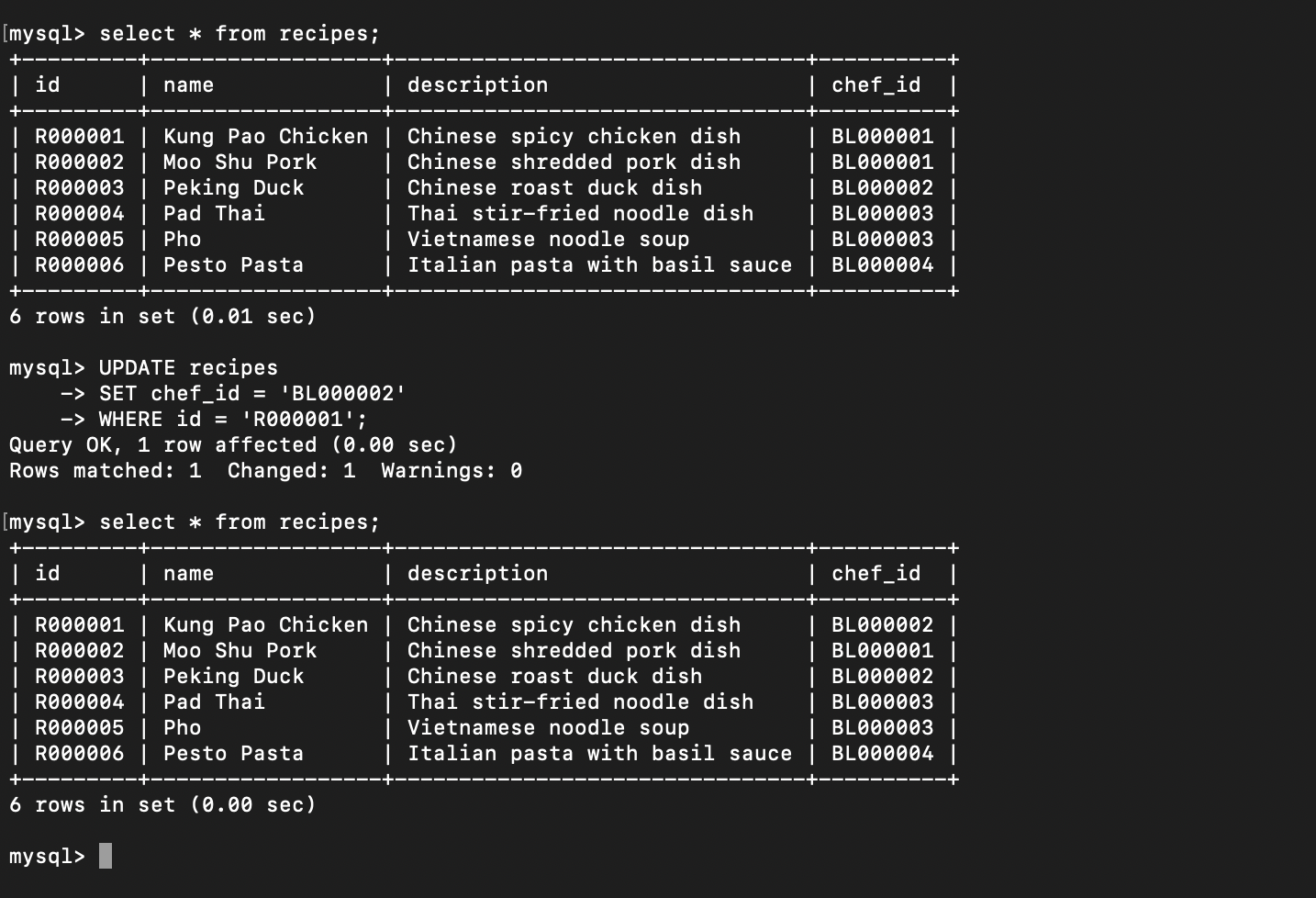
**Week- 4**

**Name:** Ronit Kumar **Reg:** RA2111032010009 **Section:** T2

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

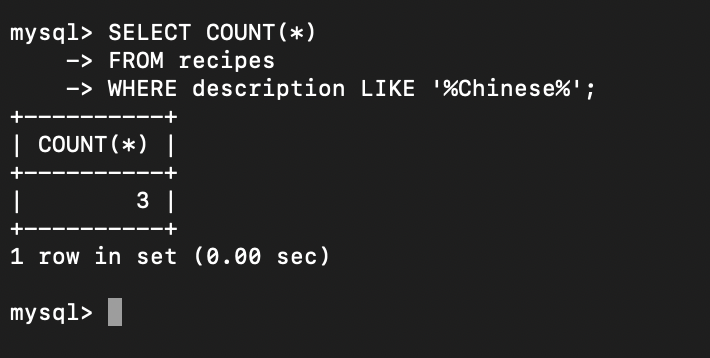
****

****

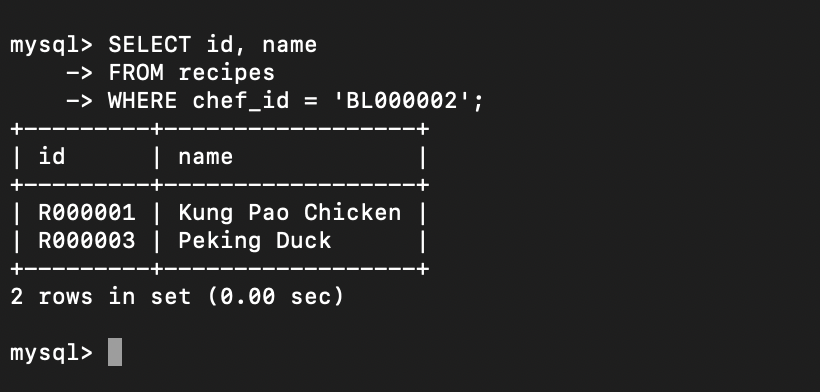
****

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

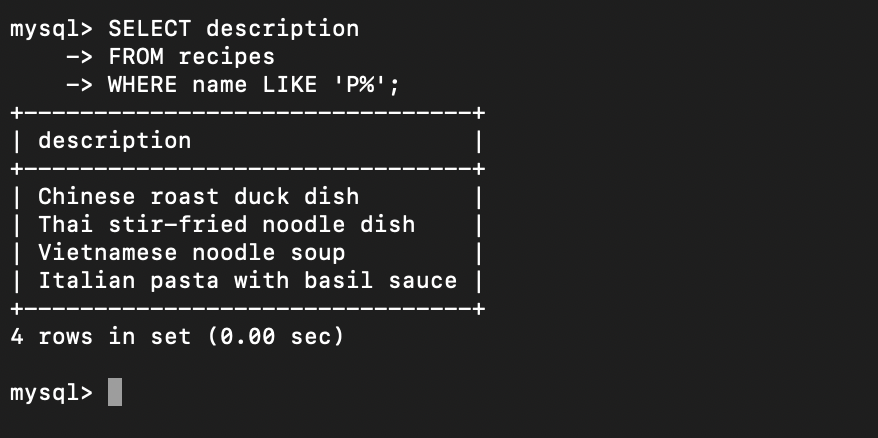
**“Chinese”**

****

**ii) Write a query to display the id, name of the recipes with chef\_id 'BL000002'.**

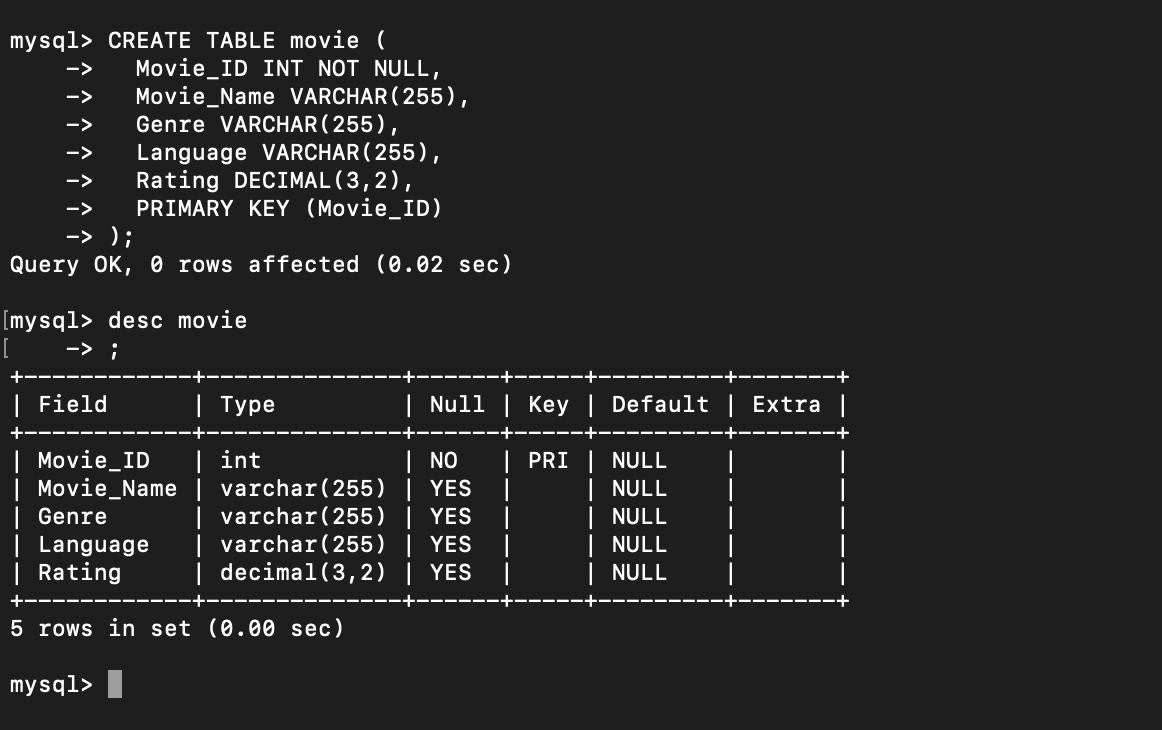
****

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

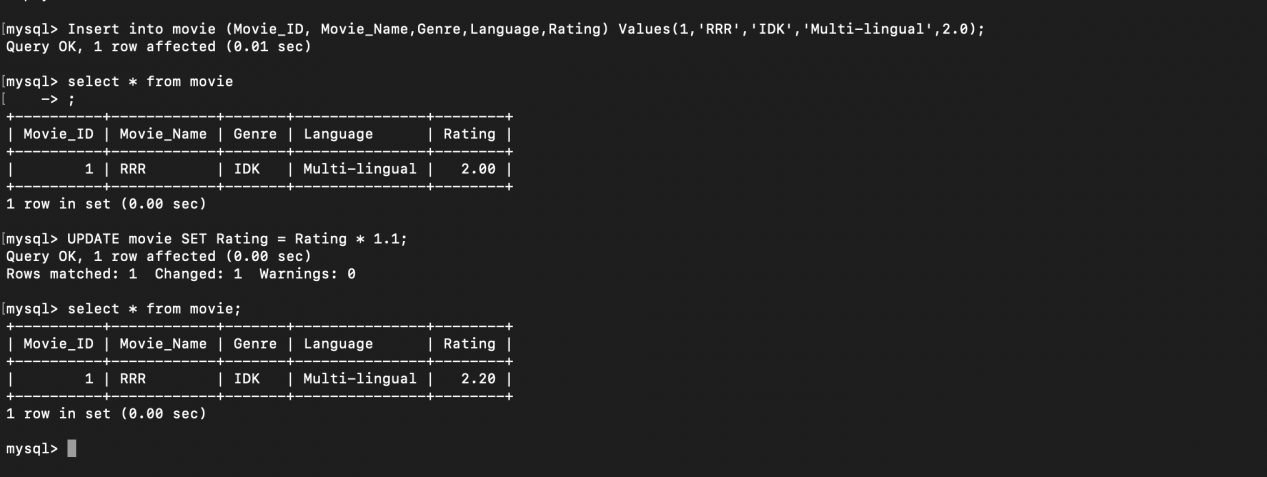
****

**2. Create a table movie of the below structure and assume data types.Movie\_ID,**

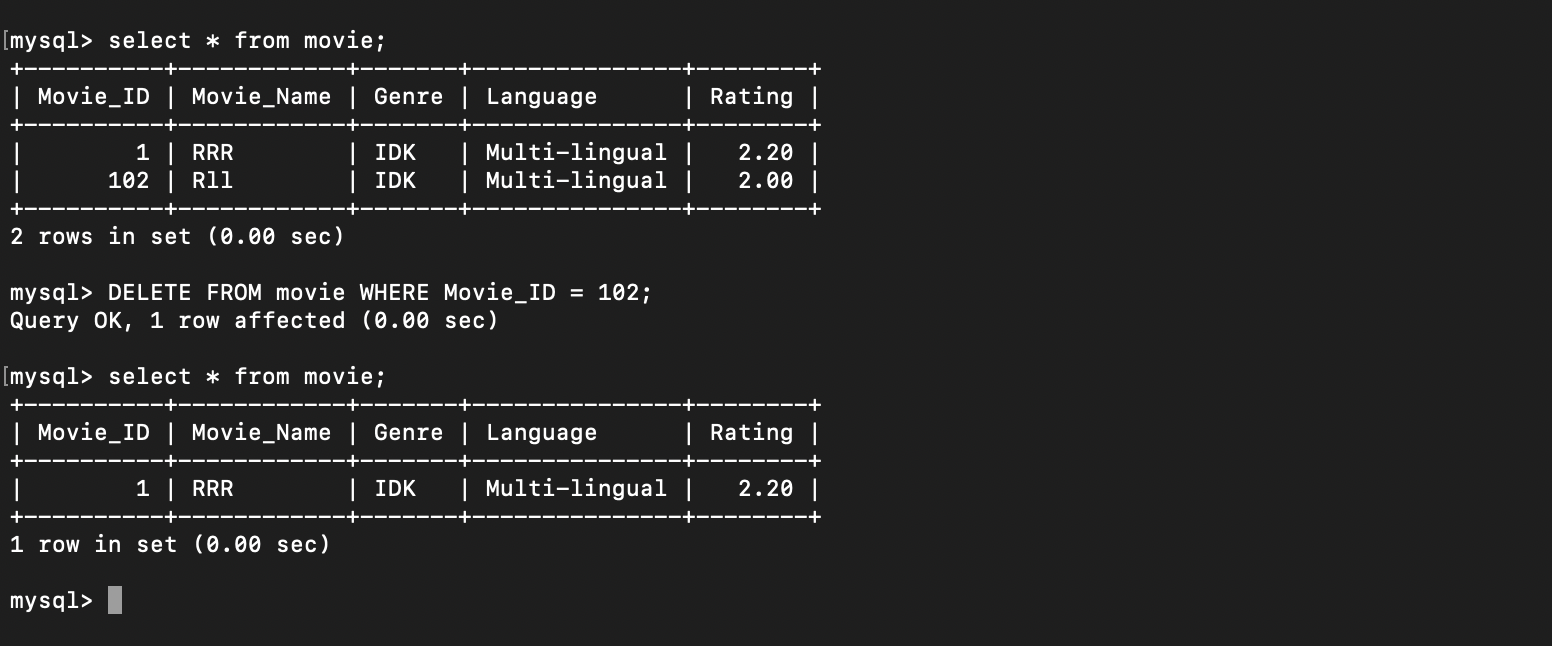
**Movie\_Name, Genre, Language, Rating ,Do the following queries**

****

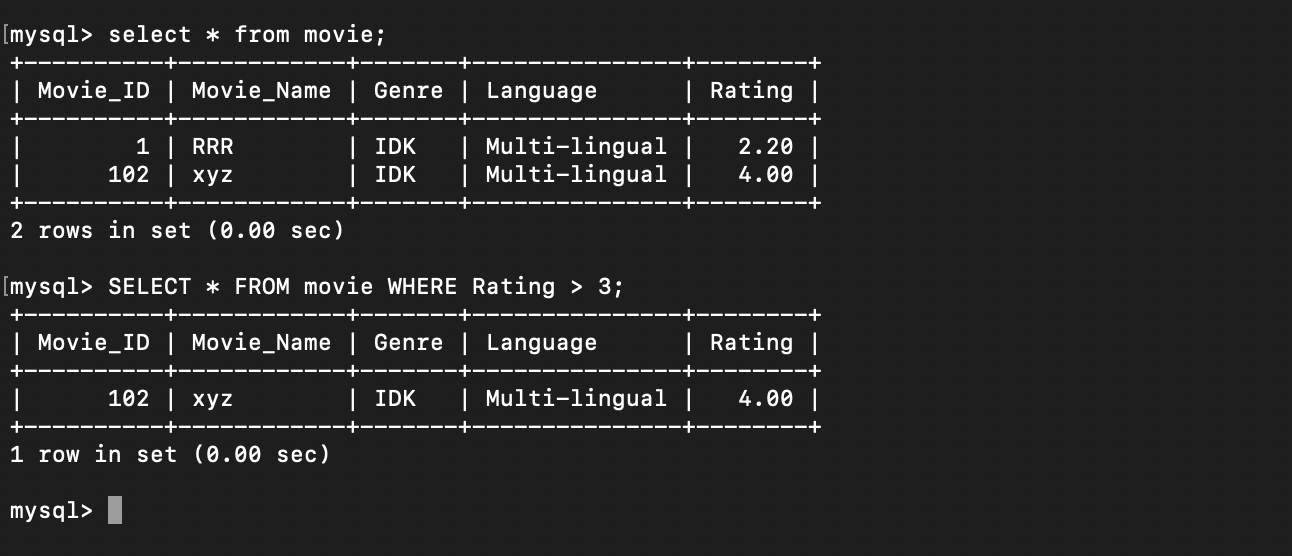
1. **Update the movies rating by 10% and display it**

****

1. **Delete the movies with movie\_id 102**

****

**c. Select movies whose rating is more than 3.**

****

**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()