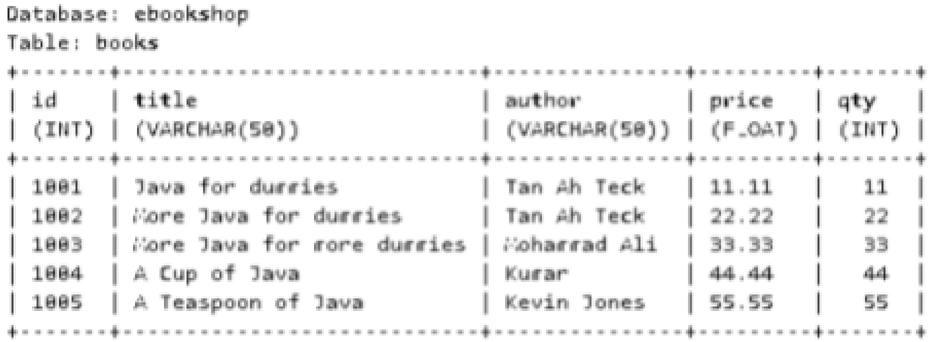
PRACTICAL-1

AIM: [Create following table using mysql and perform following task.]



```
Database: ebookshop
Table: books
+-------+----------------------------+----------------+----------+-------+
| id    | title                      | author         | price    | qty   |
| (INT) | (VARCHAR(50))              | (VARCHAR(50))  | (FLOAT)  | (INT) |
+-------+----------------------------+----------------+----------+-------+
| 1001  | Java for dummies           | Tan Ah Teck    | 11.11    |   11  |
| 1002  | More Java for dummies      | Tan Ah Teck    | 22.22    |   22  |
| 1003  | More Java for more dummies | Mohammad Ali   | 33.33    |   33  |
| 1004  | A Cup of Java              | Kumar          | 44.44    |   44  |
| 1005  | A Teaspoon of Java         | Kevin Jones    | 55.55    |   55  |
+-------+----------------------------+----------------+----------+-------+
```

a. Fetch and display records from a table using field index

b. Fetch and display records from a table using Result set metadata.

c. Display database properties using Database metadata

d. Using prepared statement perform insert, update and delete operations.

e. Perform insert, update and delete using callable statement.

f. Perform commit and set auto commit.

g. Display Scrollable Record Set

PROGRAM:

```
import java.sql.*;
import java.util.*;

public class first {
    public static void main(String[] args) throws Exception {
        int menu = 0;
        Scanner sr = new Scanner(System.in);

        try{
            Class.forName("com.mysql.jdbc.Driver");
        //MAKING CONNECTION TO DB
        Connection con = 
DriverManager.getConnection("jdbc:mysql://localhost:3306/ebookshop", "root", "");


        do{


          System.out.println("\nWHAT DO YOU WANT TO PERFORM?");
          System.out.println("1. Display Records");
          System.out.println("2. ResultSet Metadat");
          System.out.println("3. Database Metadata");
          System.out.println("4. Insert Record");
          System.out.println("5. Update Records");
          System.out.println("6. Delete Records");
          System.out.println("7. Callable Insert");
          System.out.println("8. Callable Update");
          System.out.println("9. Callable Delete");
          System.out.println("10. Commit And Auto-Commit");
          System.out.println("11. Scrollabe Record Set");
          System.out.println("0. Exit");
          menu = sr.nextInt();


          switch(menu){
            case 1:
              //DISPLAYING RECORDS
              Statement stmt = con.createStatement();
              ResultSet rs = stmt.executeQuery("select * from books");
              System.out.println("\nmsg: Displaying Table:");
              while(rs.next()){
```

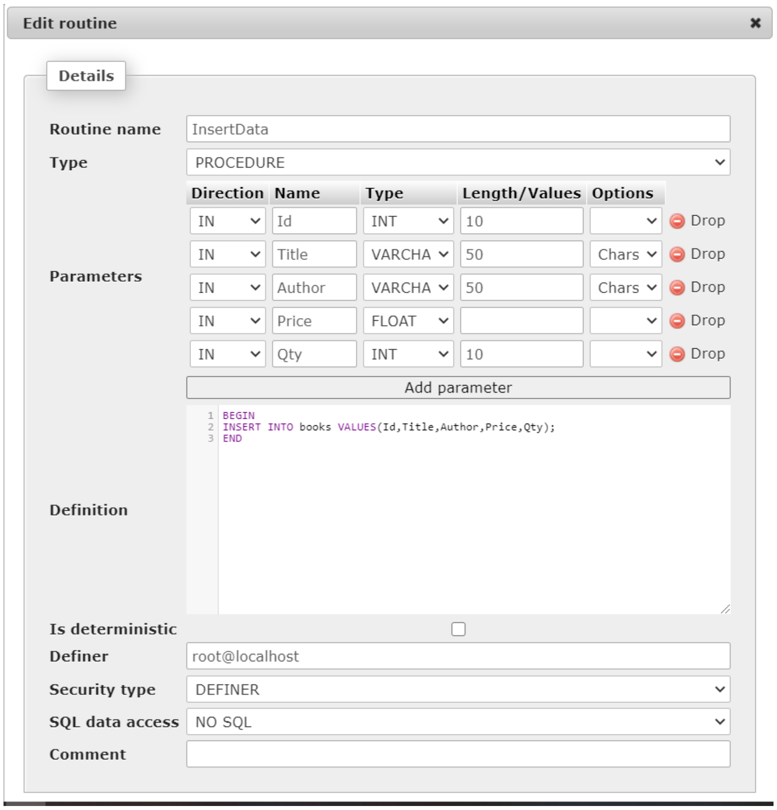
```
                        System.out.println(rs.getInt(1) + " | " + rs.getString(2) + " | " + rs.getString(3) +
" | " + rs.getFloat(4) + " | " + rs.getInt(5));

                    }

                break;


                case 2:

                    Statement stmtm = con.createStatement();

                      //Retrieving the data

                      ResultSet rsm = stmtm.executeQuery("select * from books");

                      ResultSetMetaData rsMetaData = rsm.getMetaData();

                      //Number of columns

                      System.out.println("Number of columns: "+rsMetaData.getColumnCount());

                      //Column name

                      System.out.println("1st Column Name: "+rsMetaData.getColumnName(1));

                      //Name of Table

                      System.out.println("Table Name: "+rsMetaData.getTableName(1));

                      //Columns of Table

                      System.out.println("Total columns: "+rsMetaData.getColumnCount());

                      //Type of 1st column

                      System.out.println("1st Column Type: "+rsMetaData.getColumnTypeName(1));

                      break;


                case 3:

                    //DATABASE METADATA

                    DatabaseMetaData databaseMetaData = con.getMetaData();

                    //Print TABLE_TYPE "TABLE"

                    ResultSet resultSet = databaseMetaData.getTables(null, null, null, new
String[]{"TABLE"});

                    System.out.println("\nPrinting TABLE_NAME:");
                    while(resultSet.next()){
                        System.out.println(resultSet.getString("TABLE_NAME"));
                    }
                    System.out.println("\nDatabase Info: ");
                    System.out.println("Driver Name: "+databaseMetaData.getDriverName());
                    System.out.println("Driver Version: "+databaseMetaData.getDriverVersion());
                    System.out.println("UserName: "+databaseMetaData.getUserName());
                    System.out.println("Database Product Name: "+databaseMetaData.getDatabaseProductName());
                    System.out.println("Database Product Version: "+databaseMetaData.getDatabaseProductVersion());


                    ResultSet columns = databaseMetaData.getColumns(null,null, "books", null);
                    System.out.println("\nPrinting COLUMN_INFO:");
                    while(columns.next())
                    {
                        String columnName = columns.getString("COLUMN_NAME");
                        String datatype = columns.getString("DATA_TYPE");
                        String columnsize = columns.getString("COLUMN_SIZE");
                        String decimaldigits = columns.getString("DECIMAL_DIGITS");
                        String isNullable = columns.getString("IS_NULLABLE");
                        String is_autoIncrment = columns.getString("IS_AUTOINCREMENT");
                        //Printing results
                        System.out.println("Column Name:" +columnName + "--- Datatype:" + datatype + "--- Column Size" + columnsize + "--- Decimal Digits:" + decimaldigits + "--- isNullable:" + isNullable + "--- Is autoIncrment:" + is_autoIncrment);
                    }
                break;
```

```
          case 4:
             //INSERTING RECORDS

             int id, qty;
             float price;
             String title, author;

             System.out.println("\nInput Data for New Record:");
             System.out.println("id (int)");
             id = sr.nextInt();
             sr.nextLine();
             System.out.println("title (varchar)");
             title = sr.nextLine();
             System.out.println("author (varchar)");
             author = sr.nextLine();
             System.out.println("price (float)");
             price = sr.nextFloat();
             System.out.println("qty (int)");
             qty = sr.nextInt();

             PreparedStatement pstmt = con.prepareStatement("insert into books
values(?,?,?,?,?)");
             pstmt.setInt(1, id);
             pstmt.setString(2, title);
             pstmt.setString(3, author);
             pstmt.setFloat(4, price);
             pstmt.setInt(5, qty);
```

```
            int i = pstmt.executeUpdate();
            System.out.println("\nmsg: " + i + " records inserted\n");
        break;

        case 5:
            //UPDATING RECORDS
            System.out.println("\nInput Data to Update Records:");
            System.out.println("ID of record you want to Update");
            int id2 = sr.nextInt();
            sr.nextLine();
            System.out.println("Update Book title to ");
            String title2 = sr.nextLine();
            PreparedStatement ustmt = con.prepareStatement("UPDATE books SET title = ? WHERE id = ?");

            ustmt.setString(1, title2);
            ustmt.setInt(2, id2);
            int rowAffected = ustmt.executeUpdate();

            System.out.println("\nmsg: "+rowAffected + " records updated.\n");
        break;

        case 6:
            //DELETING RECORDS
            System.out.println("\nInput Data to Delete Records:");
            System.out.println("ID of record you want to Delete");
            int id3 = sr.nextInt();
            PreparedStatement dstmt = con.prepareStatement("delete from books where id=?");
```

```
     dstmt.setInt(1, id3);
     int rowDeleted = dstmt.executeUpdate();

     System.out.println("\nmsg: "+rowDeleted + " records deleted.\n");
 break;

case 7:
     //CALLABLE INSERTING RECORDS
     CallableStatement stmti= con.prepareCall("{call InsertData(?,?,?,?,?)}");
     stmti.setInt(1, 1006);
     stmti.setString(2, "Advance Java");
     stmti.setString(3, "Shreyas Shah");
     stmti.setFloat(4, 77);
     stmti.setInt(5, 77);
     stmti.execute();
     System.out.println("Successeful Inserted");
    break;

 case 8:
     //CALLABLE UPDATING RECORDS
     CallableStatement stmtu= con.prepareCall("{call UpdateData(?,?)}");
     stmtu.setInt(1, 1007);
     stmtu.setString(2, "Advance Java Programming");
     stmtu.execute();
     System.out.println("Successfully Updated");
     break;
```

```
        case 9:
            //CALLABLE DELETING RECORDS
            CallableStatement stmtd= con.prepareCall("{call DeleteData(?)}");
            stmtd.setInt(1, 1006);
            stmtd.execute();
            System.out.println("Successfully Deleted");
            break;

        case 10:
//COMMIT AND AUTO-COMMIT
            con.setAutoCommit(false);


            System.out.println("Type 1 to commit the querry");
            short flag=sr.nextShort();
            if (flag==1){
              PreparedStatement comstnt = con.prepareStatement("insert into books values(?,?,?,?,?)");
              comstnt.setInt(1, 1006);
            comstnt.setString(2, "Advance Java");
            comstnt.setString(3, "Shreyas Shah");
            comstnt.setFloat(4, 77);
            comstnt.setInt(5, 77);
            comstnt.executeUpdate();
            System.out.println("Quesry Executed.");
            con.commit();
            System.out.println("Quesry Commited.");
            }
             else{
              System.out.println("Quesry Rollbacked.");
              con.rollback();
             }
            break;



        case 11:
            //SCROLLABLE RECORD TYPE
            Statement
str=con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,ResultSet.CONCUR_REA
D_ONLY);
            ResultSet rsr = str.executeQuery("select * from books");

            rsr.first();
            System.out.println("First Record...");
            System.out.println(rsr.getInt(1) + "->"+ rsr.getString(2));
            rsr.absolute(3);
            System.out.println("Third Record...");
            System.out.println(rsr.getInt(1) + "->"+ rsr.getString(2));
            rsr.last();
            System.out.println("Last Record...");
            System.out.println(rsr.getInt(1) + "->"+ rsr.getString(2));
            rsr.previous();
            //rsr.relative(-1);
            System.out.println("Last to First Record...");
            System.out.println(rsr.getInt(1) + "->"+ rsr.getString(2));
            break;

        case 0:
                  System.out.println("\nExiting...\n");
              break;

              default:
                  System.out.println("\nWrong Input!\n");
              break;
        }
    } while(menu != 0);

    //CLOSING CONNECTION TO DB
    con.close();

} catch(Exception e){
    System.out.println("\nError: " + e);
}
}
}
```



Routine name: InsertData

Type: PROCEDURE

Parameters

Direction	Name	Type	Length/Values	Options	
IN	Id	INT	10		Drop
IN	Title	VARCHA	50	Chars	Drop
IN	Author	VARCHA	50	Chars	Drop
IN	Price	FLOAT			Drop
IN	Qty	INT	10		Drop

Definition

```
BEGIN
INSERT INTO books VALUES(Id,Title,Author,Price,Qty);
END
```

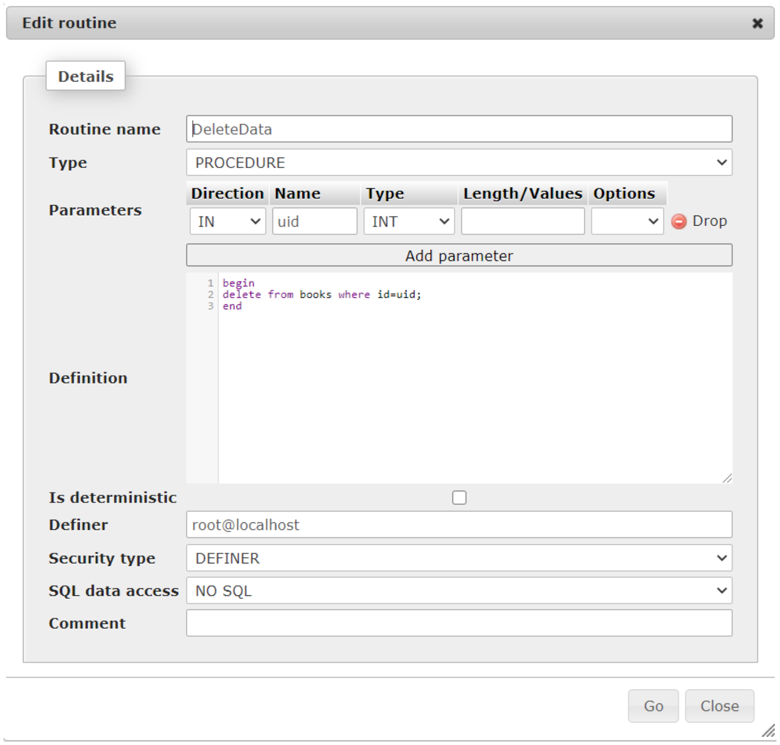
Is deterministic ☐

Definer: root@localhost

Security type: DEFINER

SQL data access: NO SQL

Comment:



Routine name: DeleteData

Type: PROCEDURE

Parameters

Direction	Name	Type	Length/Values	Options	
IN	uid	INT			Drop

Definition

```
begin
delete from books where id=uid;
end
```

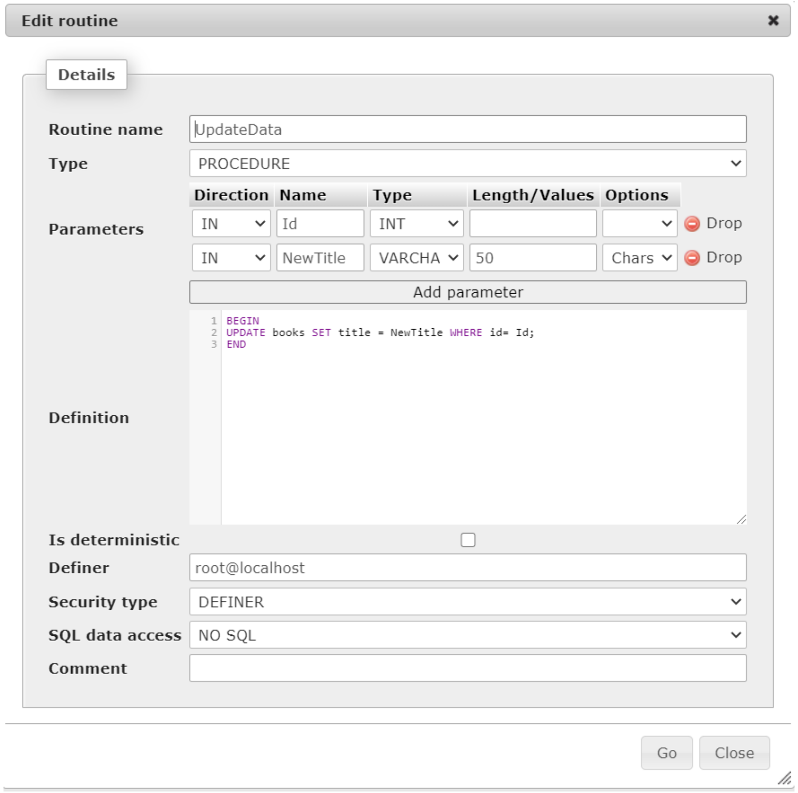
Is deterministic ☐

Definer: root@localhost

Security type: DEFINER

SQL data access: NO SQL

Comment:



Routine name: UpdateData

Type: PROCEDURE

Parameters

Direction	Name	Type	Length/Values	Options
IN	Id	INT		
IN	NewTitle	VARCHA	50	Chars

Definition

```
BEGIN
UPDATE books SET title = NewTitle WHERE id= Id;
END
```

Is deterministic: □

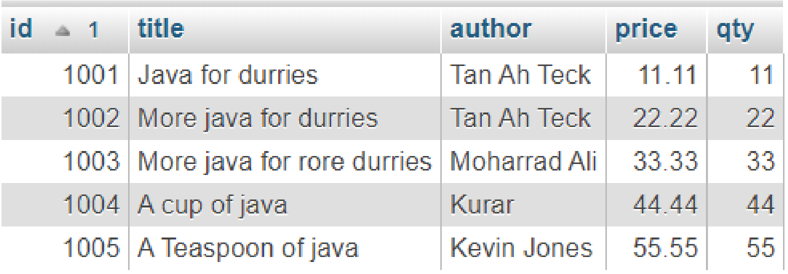
Definer: root@localhost

Security type: DEFINER

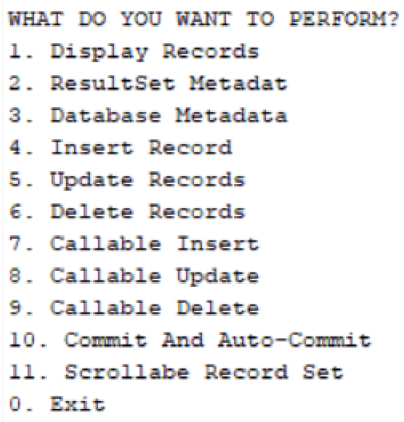
SQL data access: NO SQL

Comment:

OUTPUT:

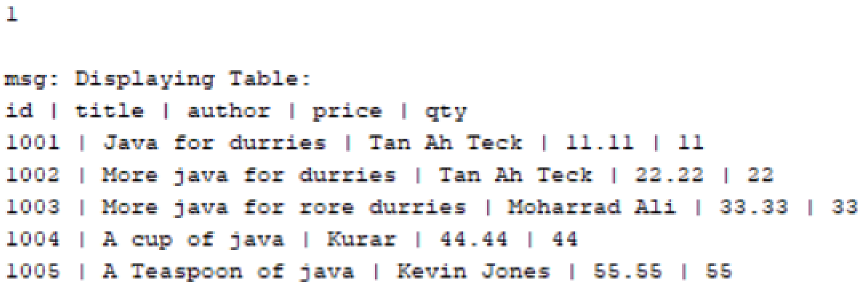


id ▲ 1	title	author	price	qty
1001	Java for durries	Tan Ah Teck	11.11	11
1002	More java for durries	Tan Ah Teck	22.22	22
1003	More java for rore durries	Moharrad Ali	33.33	33
1004	A cup of java	Kurar	44.44	44
1005	A Teaspoon of java	Kevin Jones	55.55	55



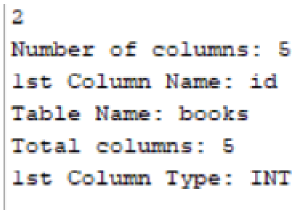
```
WHAT DO YOU WANT TO PERFORM?
1. Display Records
2. ResultSet Metadat
3. Database Metadata
4. Insert Record
5. Update Records
6. Delete Records
7. Callable Insert
8. Callable Update
9. Callable Delete
10. Commit And Auto-Commit
11. Scrollabe Record Set
0. Exit
```

a. Fetch and display records from a table using field index:



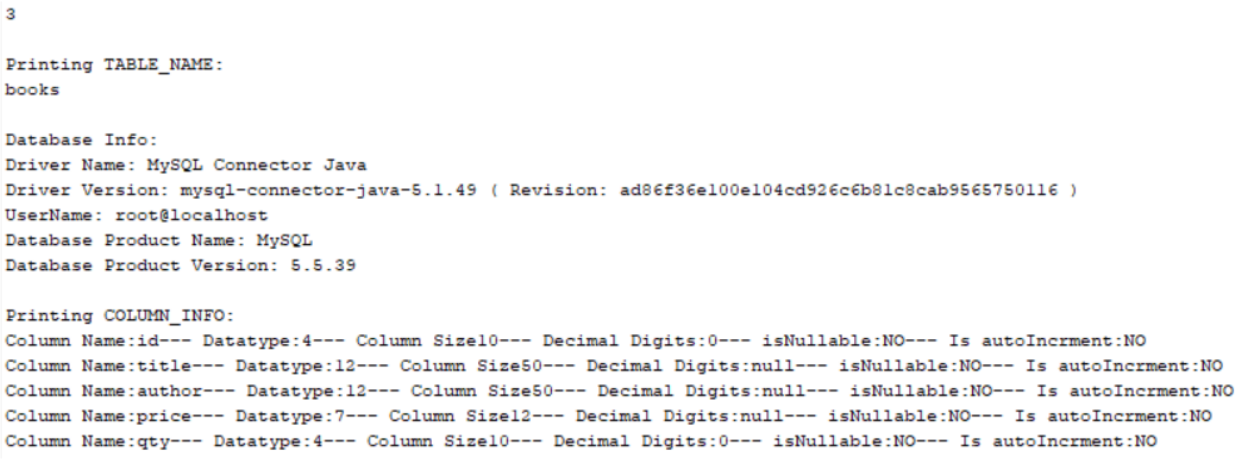
```
1

msg: Displaying Table:
id | title | author | price | qty
1001 | Java for durries | Tan Ah Teck | 11.11 | 11
1002 | More java for durries | Tan Ah Teck | 22.22 | 22
1003 | More java for rore durries | Moharrad Ali | 33.33 | 33
1004 | A cup of java | Kurar | 44.44 | 44
1005 | A Teaspoon of java | Kevin Jones | 55.55 | 55
```

b. Fetch and display records from a table using Result set metadata:



```
2
Number of columns: 5
1st Column Name: id
Table Name: books
Total columns: 5
1st Column Type: INT
```

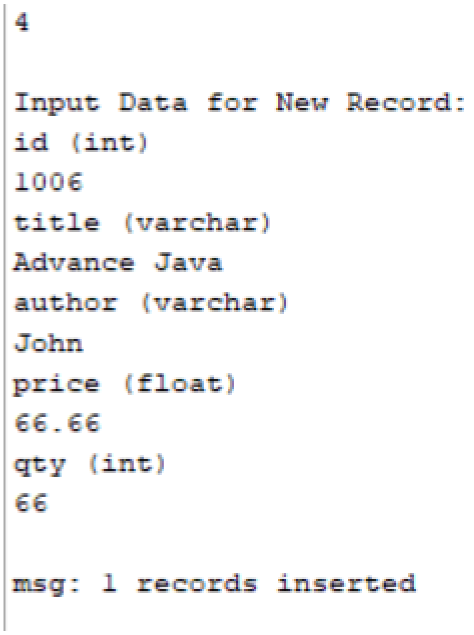
c. Display database properties using Database metadata:



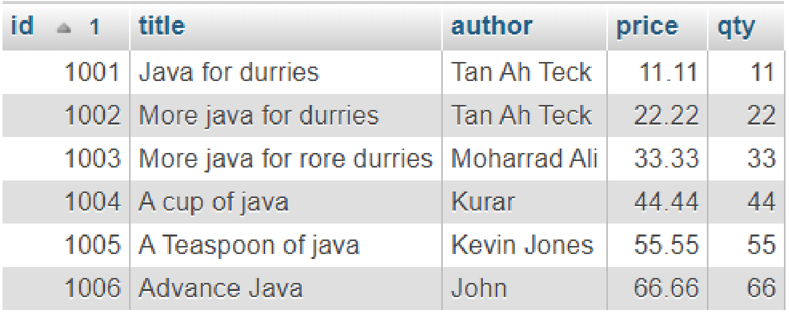
```
3

Printing TABLE_NAME:
books

Database Info:
Driver Name: MySQL Connector Java
Driver Version: mysql-connector-java-5.1.49 ( Revision: ad86f36e100e104cd926c6b81c8cab9565750116 )
UserName: root@localhost
Database Product Name: MySQL
Database Product Version: 5.5.39

Printing COLUMN_INFO:
Column Name:id--- Datatype:4--- Column Size10--- Decimal Digits:0--- isNullable:NO--- Is autoIncrment:NO
Column Name:title--- Datatype:12--- Column Size50--- Decimal Digits:null--- isNullable:NO--- Is autoIncrment:NO
Column Name:author--- Datatype:12--- Column Size50--- Decimal Digits:null--- isNullable:NO--- Is autoIncrment:NO
Column Name:price--- Datatype:7--- Column Size12--- Decimal Digits:null--- isNullable:NO--- Is autoIncrment:NO
Column Name:qty--- Datatype:4--- Column Size10--- Decimal Digits:0--- isNullable:NO--- Is autoIncrment:NO
```

d. Using prepared statement perform insert, update and delete operations:

1)Insert:

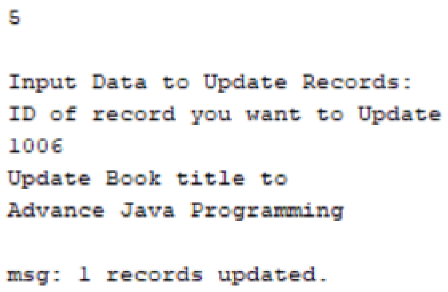


```
4

Input Data for New Record:
id (int)
1006
title (varchar)
Advance Java
author (varchar)
John
price (float)
66.66
qty (int)
66

msg: 1 records inserted
```

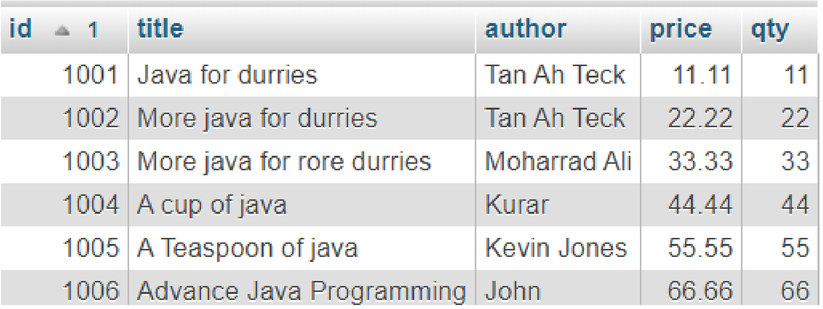


id ▲ 1	title	author	price	qty
1001	Java for durries	Tan Ah Teck	11.11	11
1002	More java for durries	Tan Ah Teck	22.22	22
1003	More java for rore durries	Moharrad Ali	33.33	33
1004	A cup of java	Kurar	44.44	44
1005	A Teaspoon of java	Kevin Jones	55.55	55
1006	Advance Java	John	66.66	66

2)Update:

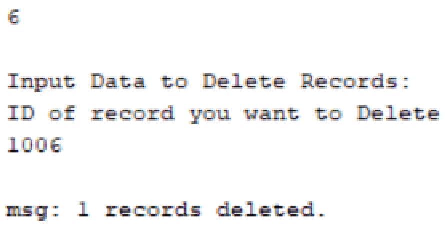


```
5

Input Data to Update Records:
ID of record you want to Update
1006
Update Book title to
Advance Java Programming

msg: 1 records updated.
```

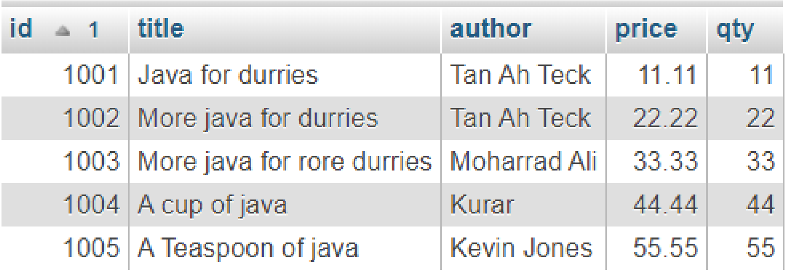


id ▲ 1	title	author	price	qty
1001	Java for durries	Tan Ah Teck	11.11	11
1002	More java for durries	Tan Ah Teck	22.22	22
1003	More java for rore durries	Moharrad Ali	33.33	33
1004	A cup of java	Kurar	44.44	44
1005	A Teaspoon of java	Kevin Jones	55.55	55
1006	Advance Java Programming	John	66.66	66

3)Delete:



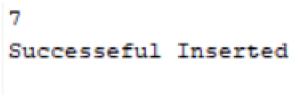
```
6

Input Data to Delete Records:
ID of record you want to Delete
1006

msg: 1 records deleted.
```



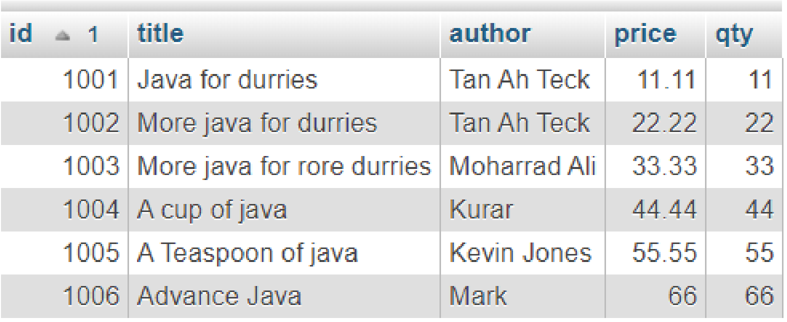
id ▲ 1	title	author	price	qty
1001	Java for durries	Tan Ah Teck	11.11	11
1002	More java for durries	Tan Ah Teck	22.22	22
1003	More java for rore durries	Moharrad Ali	33.33	33
1004	A cup of java	Kurar	44.44	44
1005	A Teaspoon of java	Kevin Jones	55.55	55

e. Perform insert, update and delete using callable statement:

1)Insert:

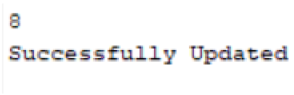


```
7
Successeful Inserted
```

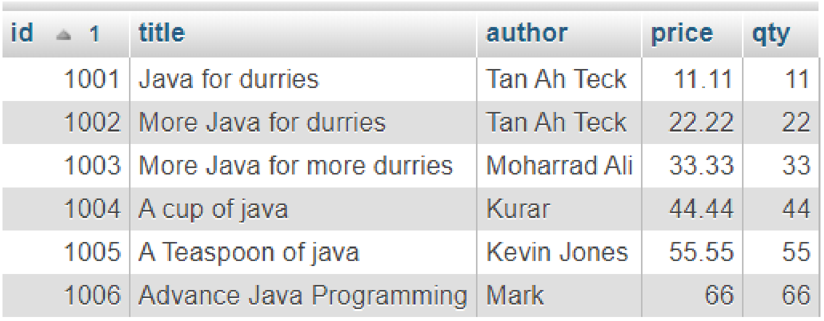


id ▲ 1	title	author	price	qty
1001	Java for durries	Tan Ah Teck	11.11	11
1002	More java for durries	Tan Ah Teck	22.22	22
1003	More java for rore durries	Moharrad Ali	33.33	33
1004	A cup of java	Kurar	44.44	44
1005	A Teaspoon of java	Kevin Jones	55.55	55
1006	Advance Java	Mark	66	66

2)Update:

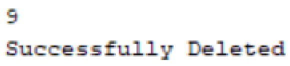


```
8
Successfully Updated
```

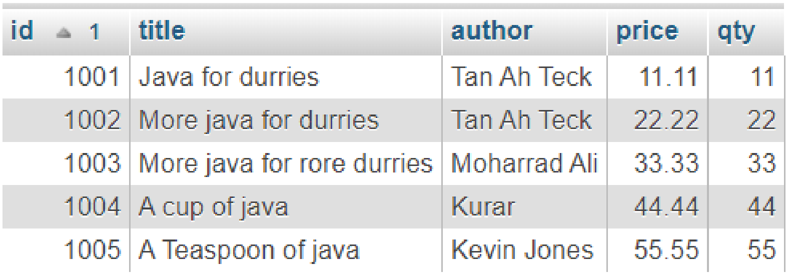


id ▲ 1	title	author	price	qty
1001	Java for durries	Tan Ah Teck	11.11	11
1002	More Java for durries	Tan Ah Teck	22.22	22
1003	More Java for more durries	Moharrad Ali	33.33	33
1004	A cup of java	Kurar	44.44	44
1005	A Teaspoon of java	Kevin Jones	55.55	55
1006	Advance Java Programming	Mark	66	66

3)Delete:

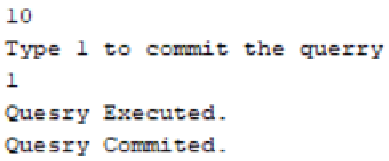


```
9
Successfully Deleted
```

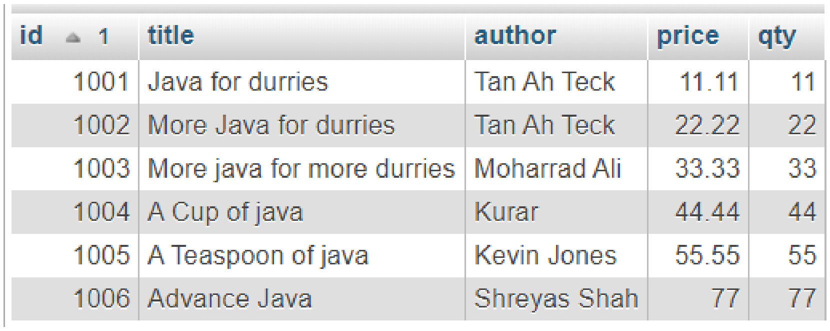


id ▲ 1	title	author	price	qty
1001	Java for durries	Tan Ah Teck	11.11	11
1002	More java for durries	Tan Ah Teck	22.22	22
1003	More java for rore durries	Moharrad Ali	33.33	33
1004	A cup of java	Kurar	44.44	44
1005	A Teaspoon of java	Kevin Jones	55.55	55

f. Perform commit and set auto commit:

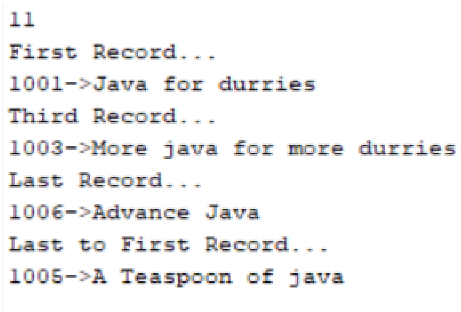


```
10
Type 1 to commit the querry
1
Quesry Executed.
Quesry Commited.
```



id ▲ 1	title	author	price	qty
1001	Java for durries	Tan Ah Teck	11.11	11
1002	More Java for durries	Tan Ah Teck	22.22	22
1003	More java for more durries	Moharrad Ali	33.33	33
1004	A Cup of java	Kurar	44.44	44
1005	A Teaspoon of java	Kevin Jones	55.55	55
1006	Advance Java	Shreyas Shah	77	77

g. Display Scrollable Record Set:



```
11
First Record...
1001->Java for durries
Third Record...
1003->More java for more durries
Last Record...
1006->Advance Java
Last to First Record...
1005->A Teaspoon of java
```

CONCLUSION:

We studied about JDBC and how to connect MySQL with in and perform basic tasks on records of the table.