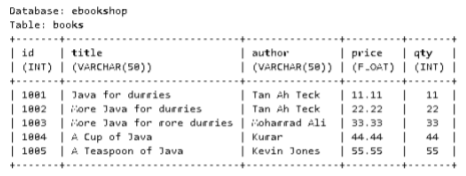
**PRACTICAL-1**

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



**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\_READ\_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();

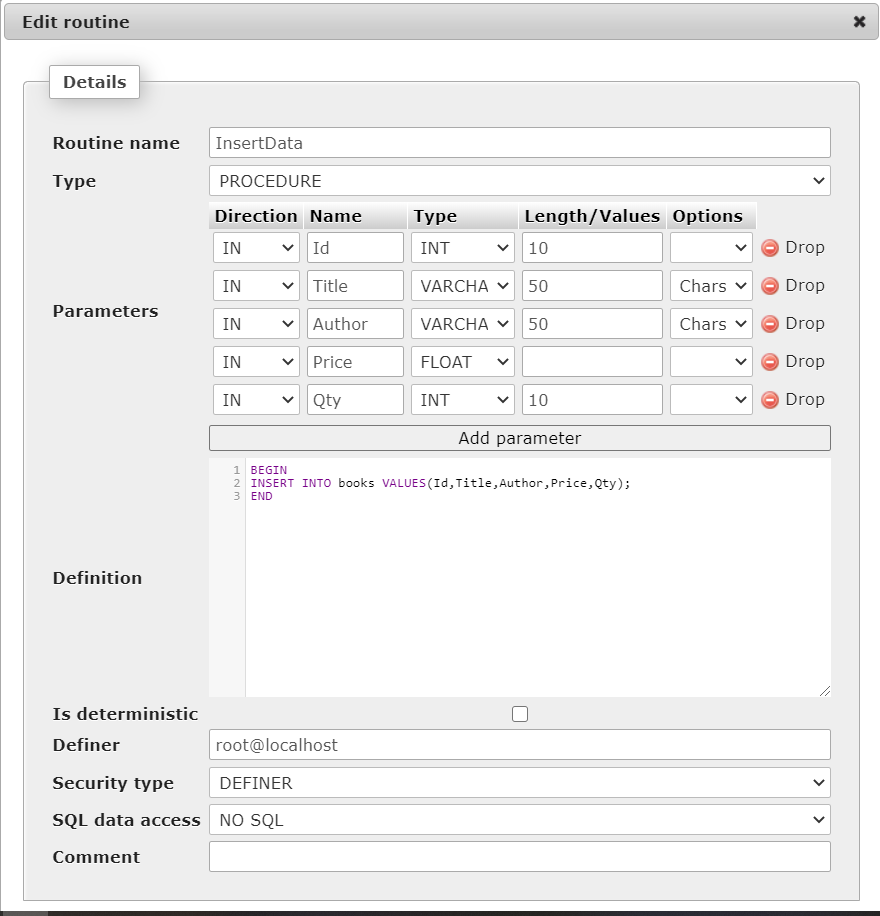
} catch(Exception e){

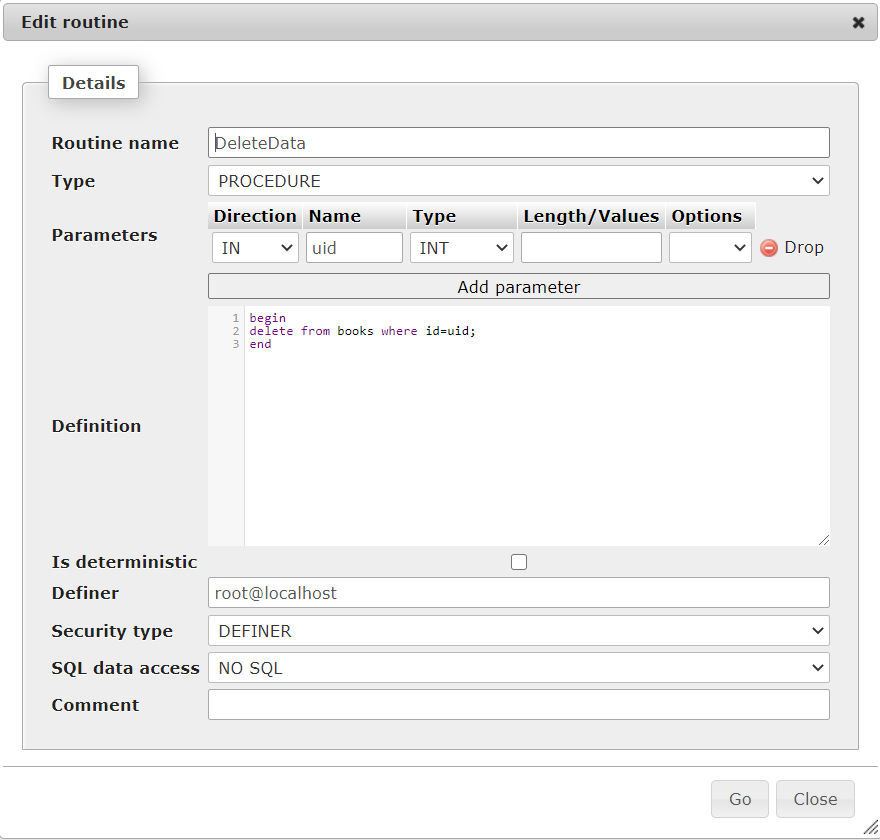
System.out.println("\nError: " + e);

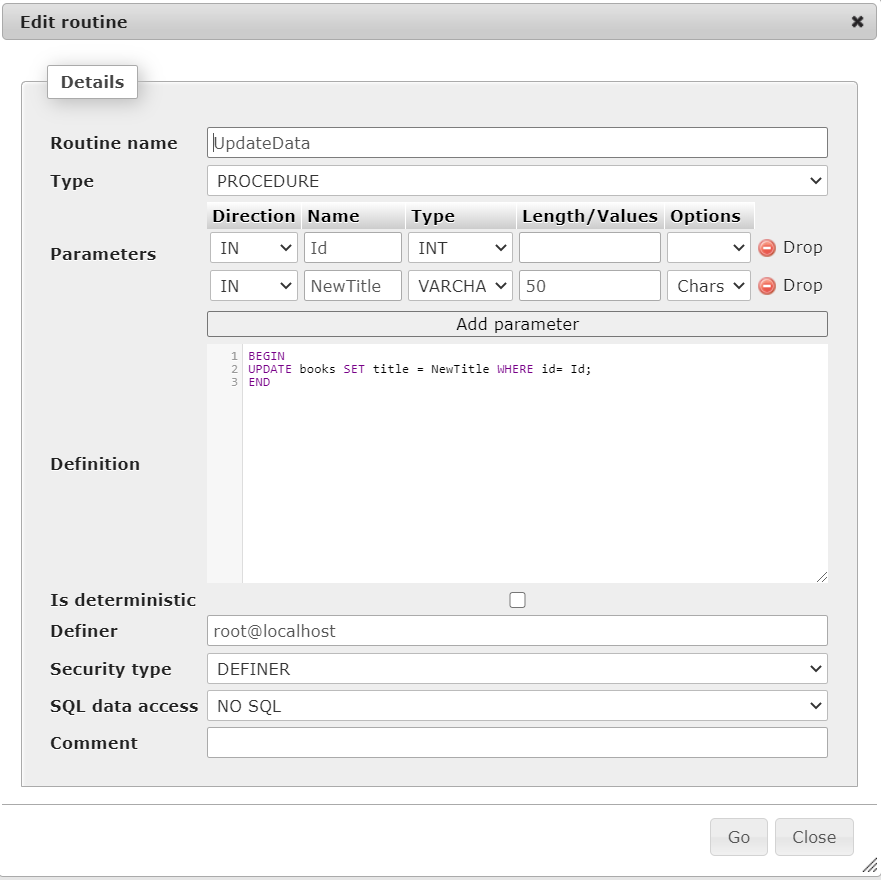
}

}

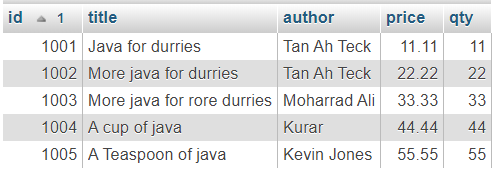
}

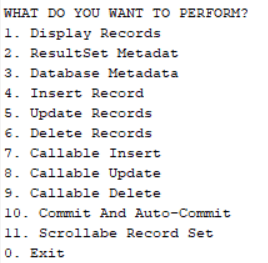




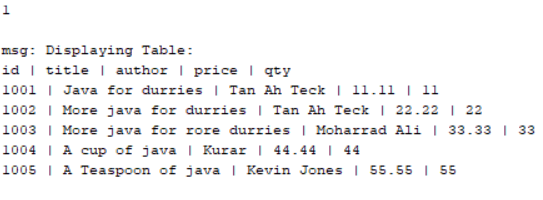


**OUTPUT:**

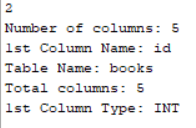




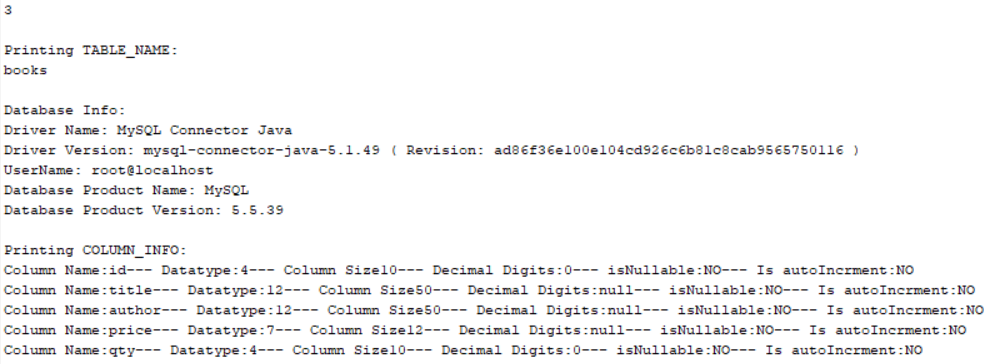
a. Fetch and display records from a table using field index:



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

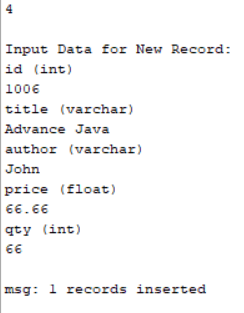
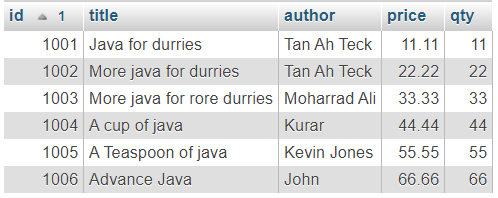


c. Display database properties using Database metadata:

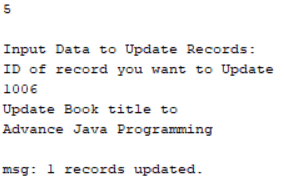


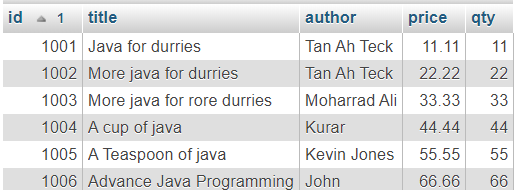
d. Using prepared statement perform insert, update and delete operations:

1)Insert:

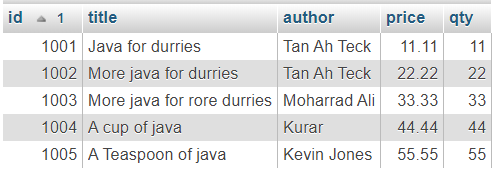
 

2)Update:





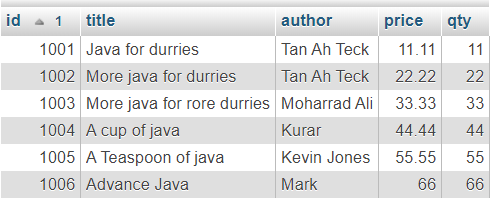
3)Delete:

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

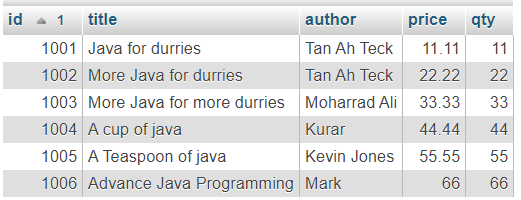
1)Insert:





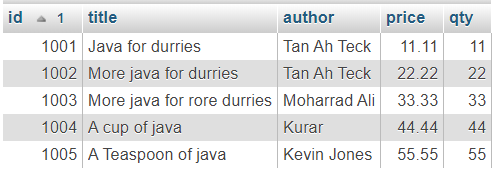
2)Update:



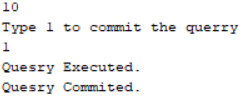


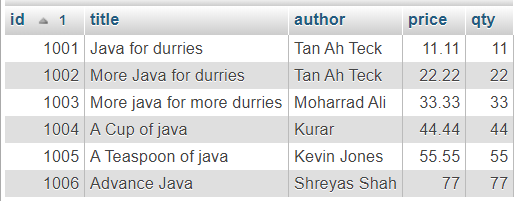
3)Delete:



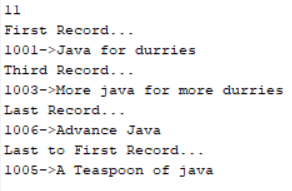


f. Perform commit and set auto commit:





g. Display Scrollable Record Set:



**CONCLUSION:**

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