Java Database Connectivity (JDBC)

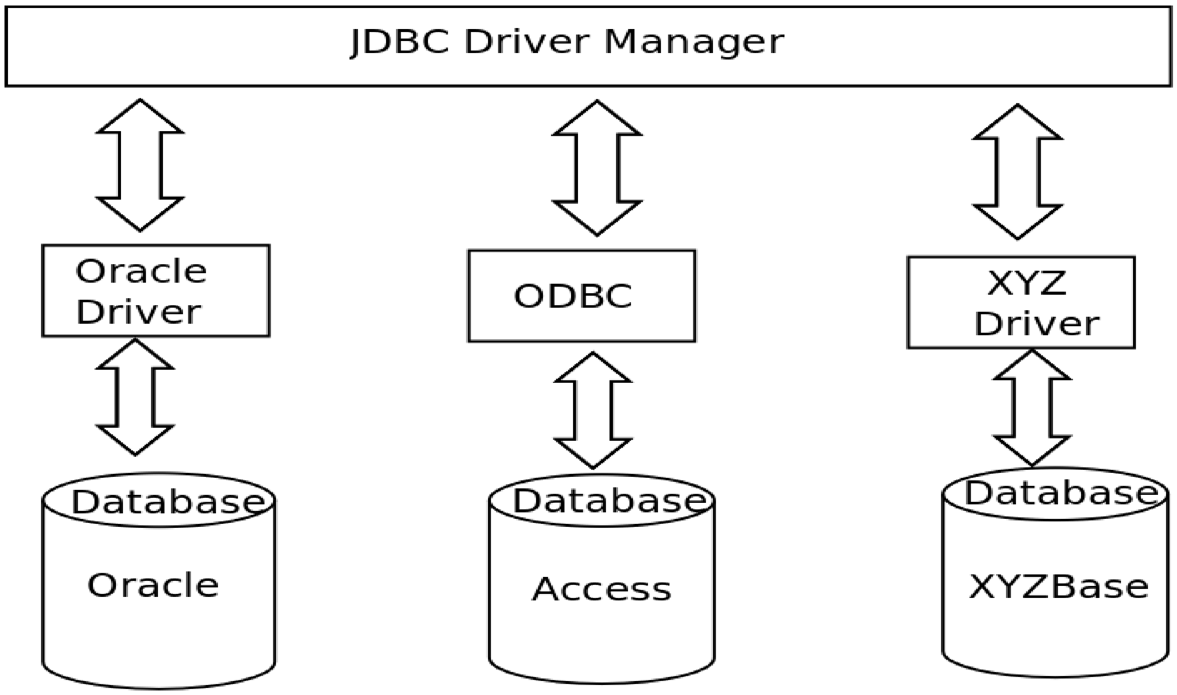
- Database is collection of related data.
- Data is collection of facts and figures which can be processed to produce information.
- Name of a student, age, class and her subjects can be counted as data for recording purposes.
- A database management system stores data, in such a way which is easier to retrieve, manipulate and helps to produce information.

- JDBC is a Java API
- JDBC is used to communicate and interact the query with the database.
- JDBC drivers are used by JDBC API to connect with the database.
- JDBC provides a set of classes and interfaces for connecting to a database, creating and executing SQL statements, and processing the results.

- ***Java.sql*** *p*ackage contains various classes
- Third party vendors implements the ***java.sql.Driver*** interface in their database driver.
- **JDBC includes several key classes and interfaces that are used to interact with a database:**
  - • DriverManager: This class is used to manage a list of database drivers. It can be used to register and deregister drivers, and to establish a connection to a database using a specific driver.
  - • Connection: This interface represents a connection to a database. It can be used to create and execute SQL statements, and to manage the transaction.
  - • Statement: This interface represents a SQL statement that can be executed against a database. It can be used to execute both SELECT and non-SELECT statements.
  - • PreparedStatement: This interface extends Statement and represents a precompiled SQL statement that can be executed multiple times with different parameters.
  - • ResultSet: This interface represents the result of a SELECT statement. It can be used to iterate through the rows of the result set and retrieve the values of the columns.
  - • SQLException: This class represents an exception that is thrown when an error occurs while working with a database.
- DriverManager class is used to manage a list of database drivers. It can be used to register and deregister drivers, and to establish a connection to a database using a specific driver.
- Java Naming and Directory Interface (JNDI) API provides a standard way of accessing different naming and directory services. It allows Java applications to look up and access objects, such as resources, services, and other components,

using a naming or directory service.

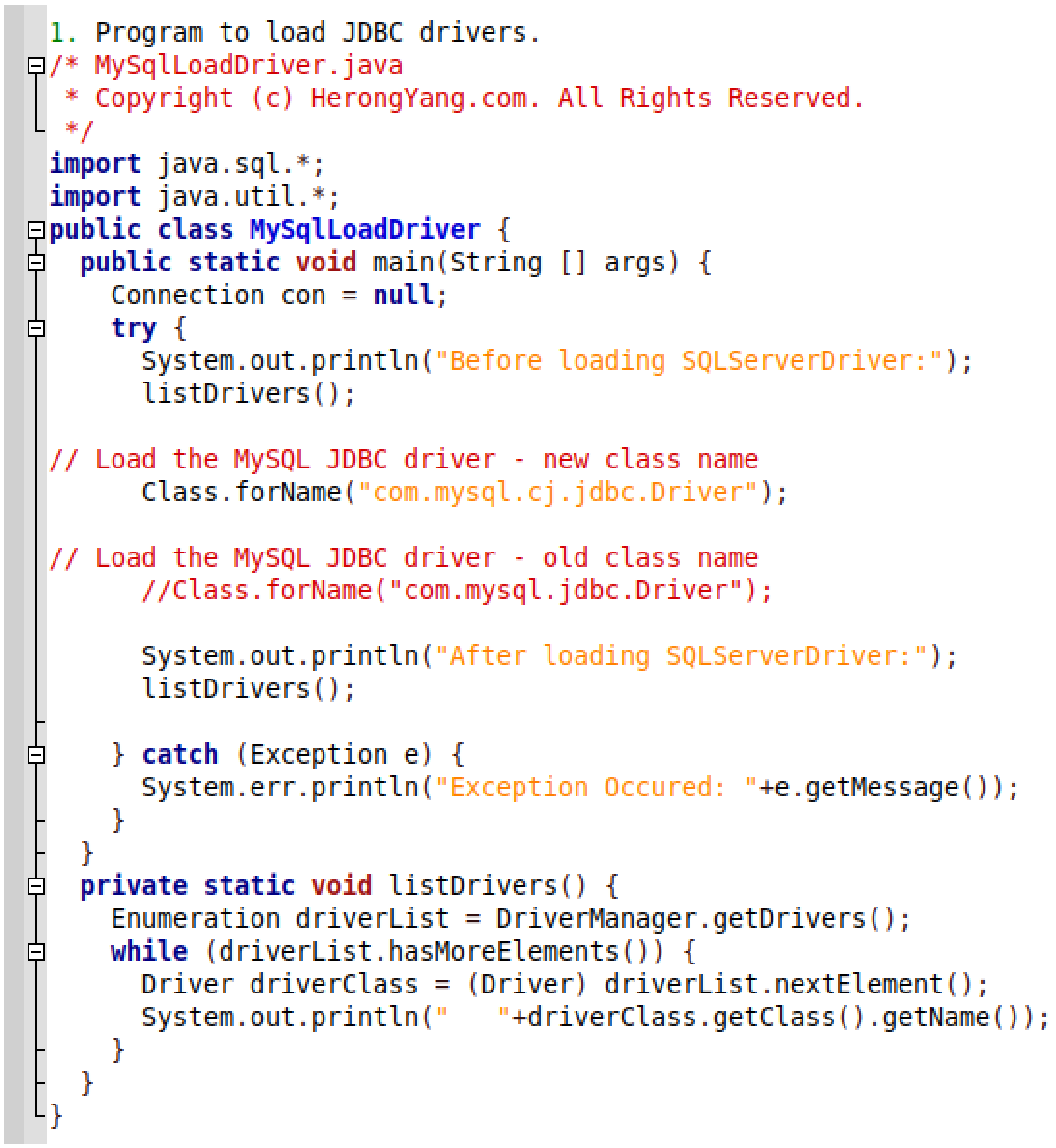
- JDBC Test Suite: The JDBC driver test suite helps us to determine that JDBC drivers will run your program. These tests are not comprehensive or exhaustive, but they do exercise many of the important features in the JDBC API.
- The JDBC Test Suite is a set of test cases and test utilities for testing the compliance and functionality of JDBC drivers. It is designed to be used by JDBC driver vendors and developers to ensure that their drivers are compatible with the JDBC specification and that they function correctly.
- Four types of Java database Drivers:
  - Type 1: JDBC-ODBC Bridge driver
  - Type 2: Native-API driver
  - Type 3: Network-Protocol driver or Intermediate database access server
  - Type 4: Native-Protocol driver



Download the MySQL connector from Official Website.

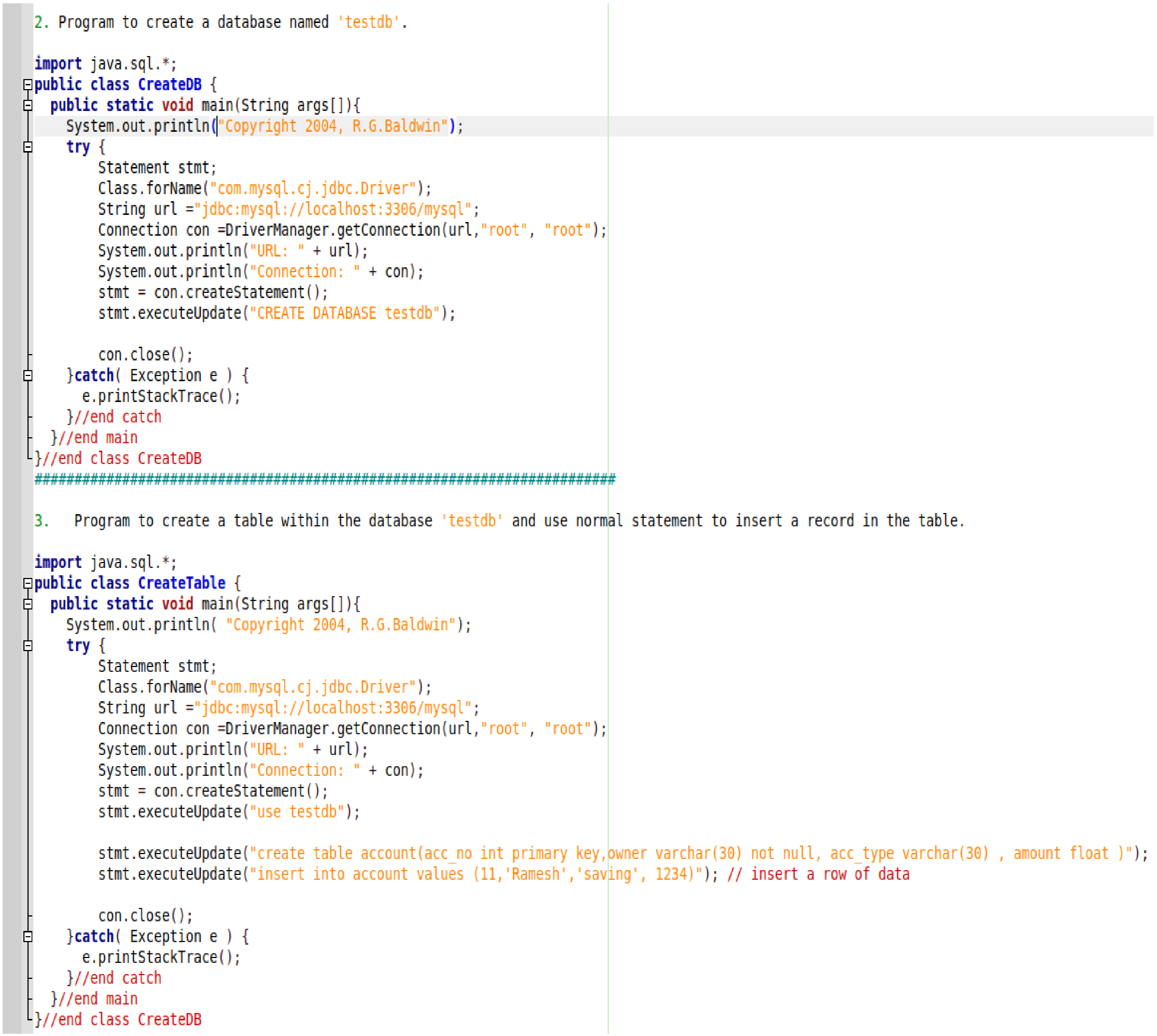
Temporary copy available here:

https://www.dropbox.com/s/s7c4mnfxja6g6vp/mysql-connector-java-8.0.30.jar?dl=0



1. Program to load JDBC drivers.

```java
/* MySqlLoadDriver.java
 * Copyright (c) HerongYang.com. All Rights Reserved.
 */
import java.sql.*;
import java.util.*;
public class MySqlLoadDriver {
  public static void main(String [] args) {
    Connection con = null;
    try {
      System.out.println("Before loading SQLServerDriver:");
      listDrivers();

// Load the MySQL JDBC driver - new class name
      Class.forName("com.mysql.cj.jdbc.Driver");

// Load the MySQL JDBC driver - old class name
      //Class.forName("com.mysql.jdbc.Driver");

      System.out.println("After loading SQLServerDriver:");
      listDrivers();

    } catch (Exception e) {
      System.err.println("Exception Occured: "+e.getMessage());
    }
  }
  private static void listDrivers() {
    Enumeration driverList = DriverManager.getDrivers();
    while (driverList.hasMoreElements()) {
      Driver driverClass = (Driver) driverList.nextElement();
      System.out.println("   "+driverClass.getClass().getName());
    }
  }
}
```



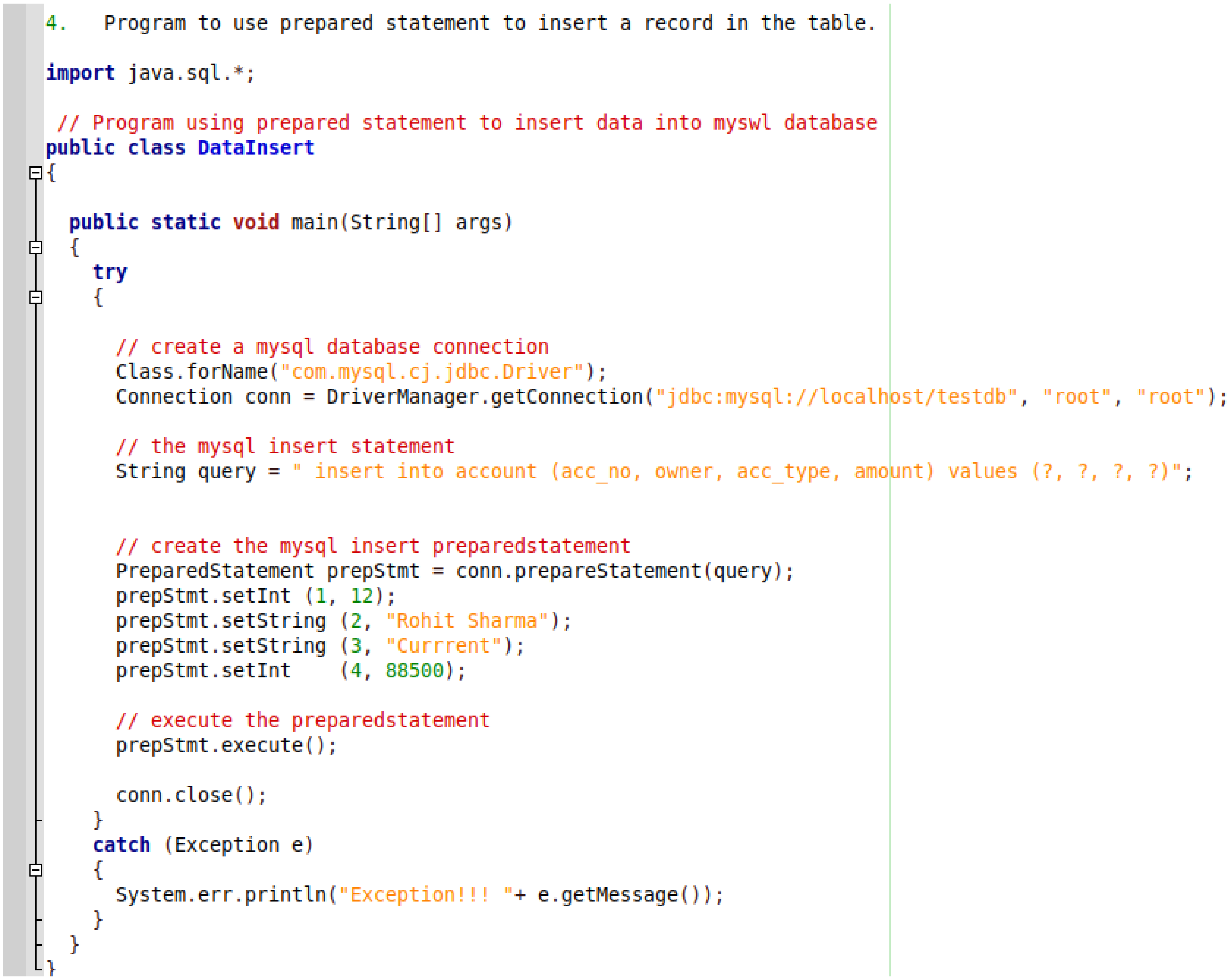
2. Program to create a database named 'testdb'.

```java
import java.sql.*;
public class CreateDB {
  public static void main(String args[]){
    System.out.println("Copyright 2004, R.G.Baldwin");
    try {
        Statement stmt;
        Class.forName("com.mysql.cj.jdbc.Driver");
        String url ="jdbc:mysql://localhost:3306/mysql";
        Connection con =DriverManager.getConnection(url,"root", "root");
        System.out.println("URL: " + url);
        System.out.println("Connection: " + con);
        stmt = con.createStatement();
        stmt.executeUpdate("CREATE DATABASE testdb");

        con.close();
    }catch( Exception e ) {
      e.printStackTrace();
    }//end catch
  }//end main
}//end class CreateDB
```

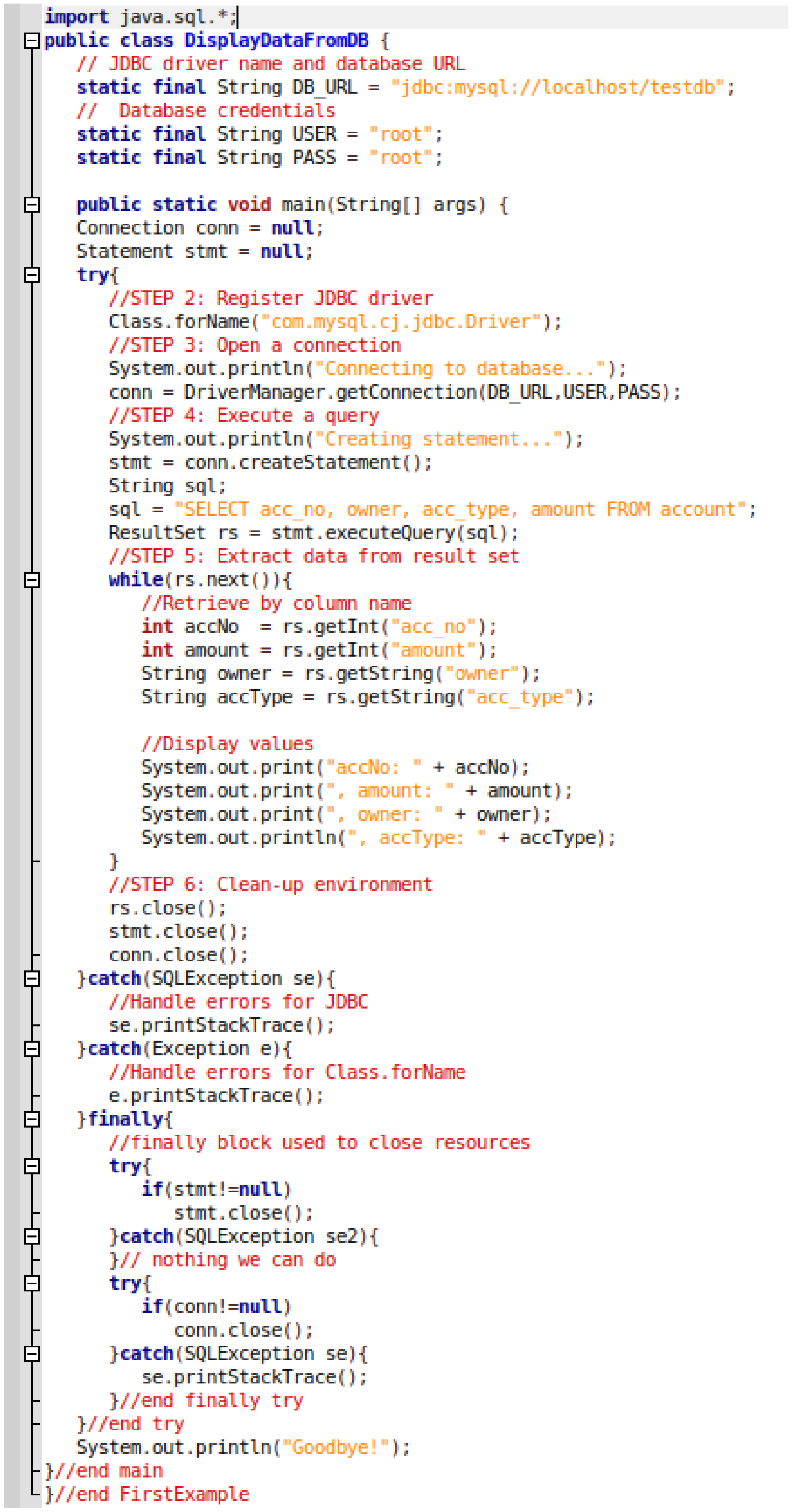
##

3. Program to create a table within the database 'testdb' and use normal statement to insert a record in the table.

```java
import java.sql.*;
public class CreateTable {
  public static void main(String args[]){
    System.out.println( "Copyright 2004, R.G.Baldwin");
    try {
        Statement stmt;
        Class.forName("com.mysql.cj.jdbc.Driver");
        String url ="jdbc:mysql://localhost:3306/mysql";
        Connection con =DriverManager.getConnection(url,"root", "root");
        System.out.println("URL: " + url);
        System.out.println("Connection: " + con);
        stmt = con.createStatement();
        stmt.executeUpdate("use testdb");

        stmt.executeUpdate("create table account(acc_no int primary key,owner varchar(30) not null, acc_type varchar(30) , amount float )");
        stmt.executeUpdate("insert into account values (11,'Ramesh','saving', 1234)"); // insert a row of data

        con.close();
    }catch( Exception e ) {
      e.printStackTrace();
    }//end catch
  }//end main
}//end class CreateDB
```



```
4.   Program to use prepared statement to insert a record in the table.

import java.sql.*;

 // Program using prepared statement to insert data into myswl database
public class DataInsert
{

  public static void main(String[] args)
  {
    try
    {

      // create a mysql database connection
      Class.forName("com.mysql.cj.jdbc.Driver");
      Connection conn = DriverManager.getConnection("jdbc:mysql://localhost/testdb", "root", "root");

      // the mysql insert statement
      String query = " insert into account (acc_no, owner, acc_type, amount) values (?, ?, ?, ?)";


      // create the mysql insert preparedstatement
      PreparedStatement prepStmt = conn.prepareStatement(query);
      prepStmt.setInt (1, 12);
      prepStmt.setString (2, "Rohit Sharma");
      prepStmt.setString (3, "Currrent");
      prepStmt.setInt    (4, 88500);

      // execute the preparedstatement
      prepStmt.execute();

      conn.close();
    }
    catch (Exception e)
    {
      System.err.println("Exception!!! "+ e.getMessage());
    }
  }
}
```

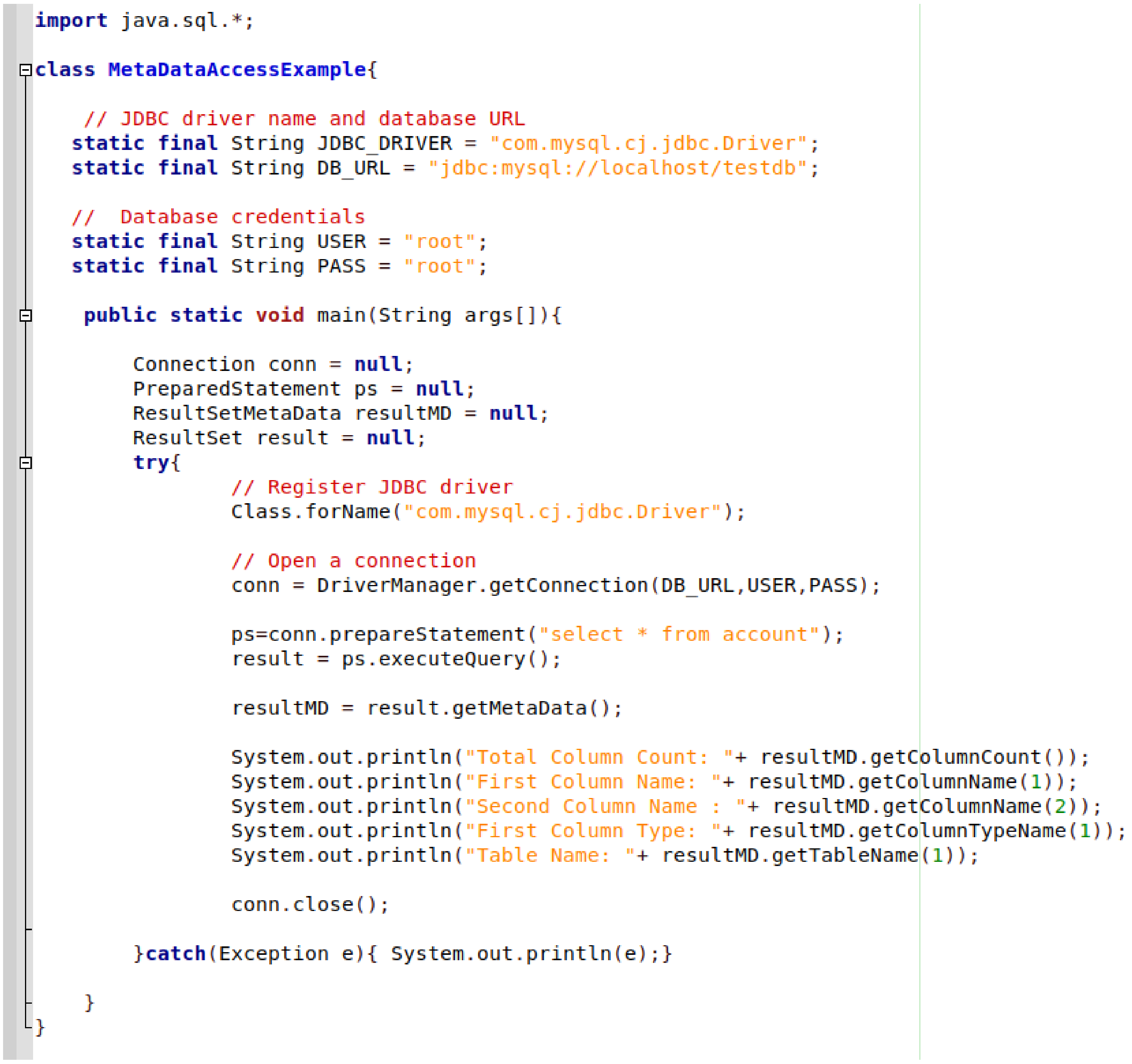


```java
import java.sql.*;
public class DisplayDataFromDB {
   // JDBC driver name and database URL
   static final String DB_URL = "jdbc:mysql://localhost/testdb";
   //  Database credentials
   static final String USER = "root";
   static final String PASS = "root";

   public static void main(String[] args) {
   Connection conn = null;
   Statement stmt = null;
   try{
      //STEP 2: Register JDBC driver
      Class.forName("com.mysql.cj.jdbc.Driver");
      //STEP 3: Open a connection
      System.out.println("Connecting to database...");
      conn = DriverManager.getConnection(DB_URL,USER,PASS);
      //STEP 4: Execute a query
      System.out.println("Creating statement...");
      stmt = conn.createStatement();
      String sql;
      sql = "SELECT acc_no, owner, acc_type, amount FROM account";
      ResultSet rs = stmt.executeQuery(sql);
      //STEP 5: Extract data from result set
      while(rs.next()){
         //Retrieve by column name
         int accNo  = rs.getInt("acc_no");
         int amount = rs.getInt("amount");
         String owner = rs.getString("owner");
         String accType = rs.getString("acc_type");

         //Display values
         System.out.print("accNo: " + accNo);
         System.out.print(", amount: " + amount);
         System.out.print(", owner: " + owner);
         System.out.println(", accType: " + accType);
      }
      //STEP 6: Clean-up environment
      rs.close();
      stmt.close();
      conn.close();
   }catch(SQLException se){
      //Handle errors for JDBC
      se.printStackTrace();
   }catch(Exception e){
      //Handle errors for Class.forName
      e.printStackTrace();
   }finally{
      //finally block used to close resources
      try{
         if(stmt!=null)
            stmt.close();
      }catch(SQLException se2){
      }// nothing we can do
      try{
         if(conn!=null)
            conn.close();
      }catch(SQLException se){
         se.printStackTrace();
      }//end finally try
   }//end try
   System.out.println("Goodbye!");
}//end main
}//end FirstExample
```

6. Program to examining the metadata.

Metadata helps to illustrate the meaning of other data i.e. metadata is a data about other data. In Java, ResultSetMetaData (i.e. java.sql.ResultSetMetaData) interface provides various methods to access metadata from the ResultSet object.

For e.g., to examine metadata of a table such as column names, types and total number of column.



```java
import java.sql.*;

class MetaDataAccessExample{

    // JDBC driver name and database URL
   static final String JDBC_DRIVER = "com.mysql.cj.jdbc.Driver";
   static final String DB_URL = "jdbc:mysql://localhost/testdb";

   //  Database credentials
   static final String USER = "root";
   static final String PASS = "root";

    public static void main(String args[]){

        Connection conn = null;
        PreparedStatement ps = null;
        ResultSetMetaData resultMD = null;
        ResultSet result = null;
        try{
                // Register JDBC driver
                Class.forName("com.mysql.cj.jdbc.Driver");

                // Open a connection
                conn = DriverManager.getConnection(DB_URL,USER,PASS);

                ps=conn.prepareStatement("select * from account");
                result = ps.executeQuery();

                resultMD = result.getMetaData();

                System.out.println("Total Column Count: "+ resultMD.getColumnCount());
                System.out.println("First Column Name: "+ resultMD.getColumnName(1));
                System.out.println("Second Column Name : "+ resultMD.getColumnName(2));
                System.out.println("First Column Type: "+ resultMD.getColumnTypeName(1));
                System.out.println("Table Name: "+ resultMD.getTableName(1));

                conn.close();

        }catch(Exception e){ System.out.println(e);}

    }
}
```