```
/*=============================================================================
* CustApp       : A JDBC APP to purchase a book for a customer.
* EECS3421      : Introduction to Database Management Systems,  Fall 2017
* Project 2     : CustApp.java
* Student Name  : Balakrishnan Lakshmi, Rajkumar
* Student Login : kumarraj
* Student ID    : 213141197
=============================================================================*/

import java.util.*;
import java.util.regex.Matcher;
import java.util.regex.Pattern;
import java.net.*;
import java.text.*;
import java.lang.*;
import java.io.*;
import java.sql.*;

/*=============================================================================
CLASS CustApp
=============================================================================*/

public class CustApp
{
   private Connection conDB;                    // Connection to the database system.
   private String     url="";                   // URL: Which database?
   private String     myCustId="";              // Customer Id.
   private Integer    custID=0;                 // Who are we tallying?
   private String     custName="";              // Name of that customer.
   private String     custCity="";              // City of that customer.
   private String     custUpdate1 = "";         // Update for - New name of the customer.
   private String     custUpdate2 = "";         // Update for -New city of the customer.
   private boolean    cupname = false;          // Boolean new name
   private boolean    cupcity = false;          // Boolean new city
   private ArrayList  <String> cat = new ArrayList<>();   // list of categories in the database
   private ArrayList  <String> titles;          // list of titles under the catop.
   private int        list=0;
   private String     catop = "";               // Category option.
   private String     titleop = "";             // Title option.
   private String     btitle= "";               // Book's title option.
   private int        byear=0;                  // Book's year.
   private String     blanguage = "";           // Book's language.
   private int        bweight=0;                // Book's weight.
   private String     cart= "";                 // Item's in the cart.
   private String     booknum="";               // Book's Number from the list of books.
   private double     minprice=0;               // Minimum price of the book.
   private String     quantity="";              // Quantity of the books to be bought.
   private double     finalprice=0;             // Final price of the book to be bought.
   private String     bclub="";                 // Book's club that offers minimal price.
   Timestamp          ctime;                    // Current time.
   DateFormat         cdate;                    // Current date.
   String             when;                     // When was the book bought - Date & Time.

   Map <Integer,ArrayList<String>> books = new HashMap <Integer,ArrayList<String>> ();




   // Constructor
   public CustApp () {
      // Set up the DB connection.
      try {
         // Register the driver with DriverManager.
         Class.forName("com.ibm.db2.jcc.DB2Driver").newInstance();
      } catch (ClassNotFoundException e) {
         e.printStackTrace();
         System.exit(0);
      } catch (InstantiationException e) {
         e.printStackTrace();
         System.exit(0);
      } catch (IllegalAccessException e) {
         e.printStackTrace();
         System.exit(0);
      }
```

```
        // URL: Which database?
        url = "jdbc:db2:c3421a";

        // Initialize the connection.
        try {
            // Connect with a fall-thru id & password
            conDB = DriverManager.getConnection(url);
        } catch(SQLException e) {
            System.out.print("\nSQL: database connection error.\n");
            System.out.println(e.toString());
            System.exit(0);
        }

        // Let's have autocommit turned off.  No particular reason here.
        try {
            conDB.setAutoCommit(false);
        } catch(SQLException e) {
            System.out.print("\nFailed trying to turn autocommit off.\n");
            e.printStackTrace();
            System.exit(0);
        }


        //First part of the app to get the customer information.
        custInfo();

        //Second part of the app to get the category and book information.
        catnbook();

        //Third part of the app to purchase of the book.
        buybook();

        // Commit.  Okay, here nothing to commit really, but why not...
        try {
            conDB.commit();
        } catch(SQLException e) {
            System.out.print("\nFailed trying to commit.\n");
            e.printStackTrace();
            System.exit(0);
        }
        // Close the connection.
        try {
            conDB.close();
        } catch(SQLException e) {
            System.out.print("\nFailed trying to close the connection.\n");
            e.printStackTrace();
            System.exit(0);
        }

    }

//=============================================================================================================

    public void custInfo()
    {
     // Who are we tallying?
     Pattern idPattern = Pattern.compile("\\d+");
        Scanner input1 = new Scanner(System.in);
        System.out.print("\nEnter the Customer ID: ");
        myCustId = input1 .nextLine();
        Matcher idmatcher = idPattern.matcher(myCustId);

        //Is this custID for real?
        while(!idmatcher.matches() || !find_customer(Integer.parseInt(myCustId)))
        {
         System.out.print("There is no customer #"+myCustId+" in the database. Please try again: ");
            myCustId = input1 .nextLine();
            idmatcher = idPattern.matcher(myCustId);
        }


        // Update the Customer Information
        Scanner input2= new Scanner(System.in);
```

```
    Pattern catPattern = Pattern.compile("^(?:yes|no)$");
      System.out.print("\nWould you like to update the customer information? Enter Yes || No : ");
      custUpdate1 = input1.nextLine().toLowerCase();
      Matcher catmatcher = catPattern.matcher(custUpdate1);

      while(!catmatcher.matches())
      {
       System.out.print("The entered option is invalid. Please try again: ");
          custUpdate1 = input1.nextLine().toLowerCase();
          catmatcher = catPattern.matcher(custUpdate1);
      }

      if (custUpdate1.equals("yes"))
      {
       Pattern updatePattern = Pattern.compile("^(?:name|city|both)$");
       System.out.print("\nPossible ways of updating customer information are:");
       System.out.print("\n1.Name\n2.City\n3.Both");
       System.out.print("\nChoose from one of the above options:");
          custUpdate2 = input2.nextLine().toLowerCase();
          Matcher updatematcher = updatePattern.matcher(custUpdate2);

          while(!updatematcher.matches())
          {
           System.out.print("The entered option is invalid. Please try again: ");
              custUpdate2 = input2.nextLine().toLowerCase();
              updatematcher = updatePattern.matcher(custUpdate2);
          }

       if(custUpdate2.equals("name"))
       {
        cupname=true;
        System.out.print("\nEnter the new name: ");
        custUpdate2 = input2.nextLine();
        update_customer(Integer.parseInt(myCustId),custUpdate2);
        System.out.print("\nName has been successfully updated to "+custUpdate2 +" !!!\n");
       }
       else if(custUpdate2.equals("city"))
       {
        cupcity=true;
        System.out.print("Enter the new city: ");
        custUpdate2 = input2.nextLine();
        update_customer(Integer.parseInt(myCustId),custUpdate2);
        System.out.print("\nCity has been successfully updated to "+custUpdate2 +" !!!\n");
       }
       else if(custUpdate2.equals("both"))
       {
        System.out.print("Enter the new name: ");
        custUpdate1 = input2.nextLine();
        System.out.print("Enter the new city: ");
        custUpdate2 = input2.nextLine();
        update_customer(Integer.parseInt(myCustId),custUpdate1,custUpdate2);
        System.out.print("\nName has been successfully updated to "+custUpdate1 +" !!!");
        System.out.print("\nCity has been successfully updated to "+custUpdate2 +" !!!\n");
       }
       System.out.print("\n-------------------------------------------------------------------------------");
      }
     System.out.print("\n--------------------------------------------------------------------------------\n");

   }

//=============================================================================================================

   public void catnbook()
   {
    Scanner input3 = new Scanner(System.in);
      System.out.print("\nThe categories of books in our database are:\n");
      cat = fetch_categories();
      list=1;
      for(String c: cat)
      {
       System.out.println(list+"."+c);
       list++;
      }

      System.out.print("Choose from one of the above categories to see the book titles: ");
```

```
        catop = input3.nextLine().toLowerCase();
        while(!cat.contains(catop))
        {
        System.out.print("There is no category \""+catop+"\" in the database. Please try again!");
        System.out.print("\nEnter the category:");
        catop = input3.nextLine().toLowerCase();
        }


        System.out.print("\nThe book titles under the category \""+catop+"\" are:\n");
        titles = fetch_titles(catop);
        list=1;
        for(String t: titles)
        {
         System.out.println(list+"."+t);
         list++;
        }
        System.out.print("Choose from one of the above book titles to view its details: ");
        Scanner input4 = new Scanner(System.in);
        titleop = input4.nextLine();
        while(!titles.contains(titleop))
        {
            System.out.print("\nThe book \""+titleop+ "\" under the category \""+catop+"\" doesn't exist. Please try again!\n");
            System.out.print("\n--------------------------------------------------------------------------------");
            catnbook();
        }
     System.out.print("\n--------------------------------------------------------------------------------\n");
    }

//=============================================================================================================

    public void buybook()
    {
     books=find_book(titleop,catop);
        if(books.size()>0)
        {
         System.out.println("\nThe books tilted \""+titleop+ "\" under the category \""+catop+"\" are: ");
         int j = 0;

            while (j <=(books.size()-1))
            {
                btitle = books.get(j).get(0);
                byear = Integer.parseInt(books.get(j).get(1));
                blanguage = books.get(j).get(2);
                bweight = Integer.parseInt((books.get(j).get(3)));
                System.out.println("\nBook No: "+j+1+"\tTitle: "+btitle +"\tYear: "+byear+"\tLanguage: "+blanguage+"\tWeight:
"+bweight);
                j++;
            }
        }

        Pattern buyPattern = Pattern.compile("^(?:yes|no)$");
        Scanner input5 = new Scanner(System.in);
        System.out.print("\nWould you like to buy the book? Enter Yes || No : ");
        cart = input5.nextLine().toLowerCase();
        Matcher buymatcher = buyPattern.matcher(cart);

        while(!buymatcher.matches())
        {
         System.out.print("The entered option is invalid. Please try again: ");
         cart = input5.nextLine().toLowerCase();
         buymatcher = buyPattern.matcher(cart);
        }
       // System.out.print("cid: "+custID+"\ncatop: "+catop+"\nbtitle: "+btitle+"\nbyear: "+byear+"\nlanguage:
"+blanguage+"\nbweight: "+bweight);

        if(cart.equals("yes"))
        {
         Pattern bookNumPattern = Pattern.compile("\\d+");
         System.out.print("\nEnter the book no you wish to buy : ");
         booknum = input5.nextLine();
         Matcher bookNummatcher = bookNumPattern.matcher(booknum);

         while(!bookNummatcher.matches() || !(Integer.parseInt(booknum)>0 && Integer.parseInt(booknum)<=books.size()))
         {
```

```
         System.out.print("Invalid Book No! Please enter a valid number to proceed: ");
         booknum = input5.nextLine();
         bookNummatcher = bookNumPattern.matcher(booknum);
        }

        btitle = books.get(Integer.parseInt(booknum)-1).get(0);
           byear = Integer.parseInt(books.get(Integer.parseInt(booknum)-1).get(1));
           blanguage = books.get(Integer.parseInt(booknum)-1).get(2);
           bweight = Integer.parseInt((books.get(Integer.parseInt(booknum)-1).get(3)));

        minprice = min_price(custID,catop, btitle, byear);
        System.out.print("\nThe minimum price of the book \""+btitle+"\" is : "+minprice);

        Pattern quanPattern = Pattern.compile("\\+?\\d+");
        System.out.print("\nEnter the number of books you wish to buy : ");
        Scanner input6= new Scanner(System.in);
        quantity = input6.nextLine();
        Matcher quanmatcher = quanPattern.matcher(quantity);

        while(!quanmatcher.matches() || Integer.parseInt(quantity)<=0)
        {
         System.out.print("Sorry the cart seems to be empty, Please enter a valid number to proceed: ");
         quantity = input6.nextLine();
         quanmatcher = quanPattern.matcher(quantity);
        }
        finalprice = minprice*Integer.parseInt(quantity);
           //System.out.print("cid: "+custID+"\ncatop: "+catop+"\nbtitle: "+btitle+"\nbyear: "+byear+"\nlanguage:
"+blanguage+"\nbweight: "+bweight+"\nQuantity: "+quantity+"\nClub: "+bclub+"\n");
        insert_purchase(custID, bclub, btitle, byear, Integer.parseInt(quantity));
        System.out.print("\n************************************");
        System.out.print("\n*Transaction Complete!");
        System.out.print("\n*Customer ID : "+custID);
        System.out.print("\n*Club        : "+bclub);
        System.out.print("\n*Title       : "+btitle);
        System.out.print("\n*Year        : "+byear);
        System.out.print("\n*Quantity    : "+quantity);
        System.out.print("\n*Total Price : "+finalprice);
        System.out.print("\n*Time        : "+when);
        System.out.print("\n*Thank you, please visit us again.");
        System.out.print("\n************************************\n");

       }
       else
       {
        System.out.print("\n\n************************************");
          System.out.print("\n*    Thanks for visisting us.      *");
          System.out.print("\n*       Have a nice day.           *");
          System.out.print("\n************************************\n");
       }


   }

//*************************************************************************************************

   public boolean find_customer(int myCustId) {
      String           queryText = "";     // The SQL text.
      PreparedStatement querySt   = null;   // The query handle.
      ResultSet        answers   = null;   // A cursor.

      boolean          inDB      = false;  // Return.

      queryText =
         "SELECT *          "
       + "FROM yrb_customer "
       + "WHERE cid = ?     ";

    // Prepare the query.
      try {
         querySt = conDB.prepareStatement(queryText);
      } catch (SQLException e) {
         System.out.println("SQL#1 failed in prepare");
         System.out.println(e.toString());
         System.exit(0);
      }
```

```
        // Execute the query.
        try {
            querySt.setInt(1, myCustId);
            answers = querySt.executeQuery();
        } catch (SQLException e) {
            System.out.println("SQL#1 failed in execute");
            System.out.println(e.toString());
            System.exit(0);
        }

        // Answer
        try {
            if (answers.next()) {
                inDB = true;
                custID = answers.getInt("cid");
                custName = answers.getString("name");
                custCity = answers.getString("city");
                System.out.println("\nCustomer ID: " + custID + "\tCustomer Name: " +custName +"\tCity: "+custCity);
            } else {
                inDB = false;
                custName = null;
            }
        } catch (SQLException e) {
            System.out.println("SQL#1 failed in cursor.");
            System.out.println(e.toString());
            System.exit(0);
        }

        // Close the cursor.
        try {
            answers.close();
        } catch (SQLException e) {
            System.out.print("SQL#1 failed closing cursor.\n");
            System.out.println(e.toString());
            System.exit(0);
        }

        // We're done with the handle.
        try {
            querySt.close();
        } catch (SQLException e) {
            System.out.print("SQL#1 failed closing the handle.\n");
            System.out.println(e.toString());

            System.exit(0);
        }

        return inDB;
    }

//-----------------------------------------------------------------------------------------------

    public boolean update_customer(int myCustId2, String... custUpdate2) {
        String           queryText = "";     // The SQL text.
        PreparedStatement querySt   = null;   // The query handle.

        boolean          inDB      = false;  // Return.

        if(cupname)
        {
        queryText =
            "UPDATE yrb_customer "
          + "SET name = ?        "
          + "WHERE cid = ?       ";
        }
        else if(cupcity)
        {
         queryText =
                  "UPDATE yrb_customer "
                + "SET city = ?        "
                + "WHERE cid = ?       ";
        }
        else
        {
```

```
        queryText =
                  "UPDATE yrb_customer     "
                + "SET name = ?, city = ?  "
                + "WHERE cid = ?           ";
        }

        // Prepare the query.
        try {
            querySt = conDB.prepareStatement(queryText);
        } catch (SQLException e) {
            System.out.println("SQL#1 failed in prepare");
            System.out.println(e.toString());
            System.exit(0);
        }

        // Execute the query.
        try {
         if(cupname || cupcity)
         {
          querySt.setString(1, custUpdate2[0]);
          querySt.setInt(2, myCustId2);
         }
         else
         {
          querySt.setString(1, custUpdate2[0]);
          querySt.setString(2, custUpdate2[1]);
          querySt.setInt(3, myCustId2);
         }
            querySt.executeUpdate();
            inDB= true;
        } catch (SQLException e) {
            System.out.println("SQL#1 failed in update");
            System.out.println(e.toString());
            System.exit(0);
        }

        // We're done with the handle.
        try {
            querySt.close();
        } catch (SQLException e) {
            System.out.print("SQL#1 failed closing the handle.\n");
            System.out.println(e.toString());

            System.exit(0);
        }

        return inDB;
    }

//-------------------------------------------------------------------------------------------------

    public ArrayList <String> fetch_categories() {
     String            queryText = "";     // The SQL text.
        PreparedStatement querySt   = null;   // The query handle.
        ResultSet         answers   = null;   // A cursor.
        ArrayList <String> cats = new ArrayList < String > ();

        queryText =
            "SELECT *          "
          + "FROM yrb_category ";

        // Prepare the query.
        try {
            querySt = conDB.prepareStatement(queryText);
        } catch (SQLException e) {
            System.out.println("SQL#2 failed in prepare");
            System.out.println(e.toString());
            System.exit(0);
        }

        // Execute the query.
        try {
            answers = querySt.executeQuery();
        } catch (SQLException e) {
            System.out.println("SQL#2 failed in execute");
```

```
            System.out.println(e.toString());
            System.exit(0);
        }

        // Answer
        try {
            for (int i=1; answers.next(); i++)
            {
                String mycategories = answers.getString("cat");
                cats.add(mycategories);
            }
        } catch (SQLException e) {
            System.out.println("SQL#2 failed in cursor.");
            System.out.println(e.toString());
            System.exit(0);
        }

        // Close the cursor.
        try {
            answers.close();
        } catch (SQLException e) {
            System.out.print("SQL#2 failed closing cursor.\n");
            System.out.println(e.toString());
            System.exit(0);
        }

        // We're done with the handle.
        try {
            querySt.close();
        } catch (SQLException e) {
            System.out.print("SQL#2 failed closing the handle.\n");
            System.out.println(e.toString());
            System.exit(0);
        }

        return cats;
    }

//-------------------------------------------------------------------------------------------------

    public ArrayList <String> fetch_titles(String category) {
     String            queryText = "";     // The SQL text.
        PreparedStatement querySt   = null;   // The query handle.
        ResultSet         answers   = null;   // A cursor.
        ArrayList <String> titles = new ArrayList < String > ();

        queryText =
          "Select distinct title " + " from yrb_book where cat = ? "+"and title in (select o.title from " +
                "yrb_offer o " + "where o.club in " + "(Select club " + "from yrb_member " +
                "where cid = ?)) and year in " + "(select o.year " + "from yrb_offer o " +
                "Where o.club in (Select club " + "from yrb_member " + "where cid = ?))";

        // Prepare the query.
        try {
            querySt = conDB.prepareStatement(queryText);
        } catch (SQLException e) {
            System.out.println("SQL#2 failed in prepare");
            System.out.println(e.toString());
            System.exit(0);
        }

        // Execute the query.
        try {
         querySt.setString(1, category);
            querySt.setInt(2, custID);
            querySt.setInt(3, custID);
            answers = querySt.executeQuery();
        } catch (SQLException e) {
            System.out.println("SQL#2 failed in execute");
            System.out.println(e.toString());
            System.exit(0);
        }

        // Answer
        try {
```

```
            for (int i=0; answers.next(); i++)
            {
               String mytitles= answers.getString("title");
               titles.add(mytitles);
            }
         } catch (SQLException e) {
            System.out.println("SQL#2 failed in cursor.");
            System.out.println(e.toString());
            System.exit(0);
         }

         // Close the cursor.
         try {
            answers.close();
         } catch (SQLException e) {
            System.out.print("SQL#2 failed closing cursor.\n");
            System.out.println(e.toString());
            System.exit(0);
         }

         // We're done with the handle.
         try {
            querySt.close();
         } catch (SQLException e) {
            System.out.print("SQL#2 failed closing the handle.\n");
            System.out.println(e.toString());
            System.exit(0);
         }

         return titles;
      }
//-------------------------------------------------------------------------------------------


    public Map <Integer,ArrayList<String>> find_book(String title,String category) {
     String            queryText = "";     // The SQL text.
         PreparedStatement querySt   = null;   // The query handle.
         ResultSet         answers   = null;   // A cursor.
         ArrayList <String> bookinfo = new ArrayList < String > ();
         Map <Integer,ArrayList<String>> bookDetails = new HashMap <Integer,ArrayList<String>> ();
         String  btitles = "";
         Integer bweights;
         String  blanguages;
         Integer byears;

         queryText =
                "SELECT *                  "
              + "FROM yrb_book             "
              + "WHERE title = ? and cat = ?";
         // Prepare the query.
         try {
            querySt = conDB.prepareStatement(queryText);
         } catch (SQLException e) {
            System.out.println("SQL#2 failed in prepare");
            System.out.println(e.toString());
            System.exit(0);
         }

         // Execute the query.
         try {
          querySt.setString(1, title);
          querySt.setString(2, category);
            answers = querySt.executeQuery();
         } catch (SQLException e) {
            System.out.println("SQL#2 failed in execute");
            System.out.println(e.toString());
            System.exit(0);
         }

         // Answer.
         try {
          int i;
            for (i=0; answers.next(); i++)
            {
             bookinfo.clear();
```

```
            btitles = answers.getString("title");bookinfo.add(btitles);
              byears = answers.getInt("year");bookinfo.add(byears.toString());
              blanguages = answers.getString("language");bookinfo.add(blanguages);
              bweights = answers.getInt("weight");bookinfo.add(bweights.toString());
              bookDetails.put(i, bookinfo);
            }
        } catch (SQLException e) {
            System.out.println("SQL#2 failed in cursor.");
            System.out.println(e.toString());
            System.exit(0);
        }

        // Close the cursor.
        try {
            answers.close();
        } catch (SQLException e) {
            System.out.print("SQL#2 failed closing cursor.\n");
            System.out.println(e.toString());
            System.exit(0);
        }

        // We're done with the handle.
        try {
            querySt.close();
        } catch (SQLException e) {
            System.out.print("SQL#2 failed closing the handle.\n");
            System.out.println(e.toString());
            System.exit(0);
        }

        return bookDetails;
    }

//-----------------------------------------------------------------------------------------------

    public double min_price(int cid, String catg, String title, int year) {
        String           queryText = "";     // The SQL text.
        PreparedStatement querySt   = null;   // The query handle.
        ResultSet         answers   = null;   // A cursor.

        String           queryText2 = "";     // The SQL text.
        PreparedStatement querySt2   = null;   // The query handle.
        ResultSet         answers2   = null;   // A cursor.

        double price=0;

        queryText =
          "SELECT min(price) "
              +"FROM yrb_offer "
              +"WHERE title = ? AND year = ? "
              +"and club in (SELECT club FROM yrb_member WHERE cid = ?)";

        queryText2 =
          "SELECT o.club "
            +"FROM yrb_member m, yrb_offer o "
            +"WHERE o.club = m.club and o.year = ? "
                        +"and m.cid = ? and o.title = ? and o.price = ? ";

      // Prepare the query.
        try {
            querySt = conDB.prepareStatement(queryText);
            querySt2 = conDB.prepareStatement(queryText2);
        } catch (SQLException e) {
            System.out.println("SQL#1 failed in prepare");
            System.out.println(e.toString());
            System.exit(0);
        }

        // Execute the query.
        try {
         querySt.setString(1, title);
            querySt.setInt(2, year);
            querySt.setInt(3, cid);
            answers = querySt.executeQuery();
        } catch (SQLException e) {
```

```
            System.out.println("SQL#1 failed in execute");
            System.out.println(e.toString());
            System.exit(0);
        }

      // Answer.
        try {
          if (answers.next())
          {
              price = answers.getDouble(1);
          }
        } catch (SQLException e) {
            System.out.println("SQL#2 failed in cursor.");
            System.out.println(e.toString());
            System.exit(0);
        }


        // Execute the query2.
        try {
        querySt2.setInt(1, byear);
        querySt2.setInt(2, custID);
        querySt2.setString(3, btitle);
        querySt2.setDouble(4, price);
        answers2 = querySt2.executeQuery();
        } catch (SQLException e) {
            System.out.println("SQL#1 failed in execute");
            System.out.println(e.toString());
            System.exit(0);
        }

        // Answer2.
        try {
          if (answers2.next())
          {
              bclub = answers2.getString(1);
          }
        } catch (SQLException e) {
            System.out.println("SQL#2 failed in cursor.");
            System.out.println(e.toString());
            System.exit(0);
        }

        // Close the cursor.
        try {
            answers.close();
            answers2.close();
        } catch (SQLException e) {
            System.out.print("SQL#2 failed closing cursor.\n");
            System.out.println(e.toString());
            System.exit(0);
        }

        // We're done with the handle.
        try {
            querySt.close();
            querySt2.close();
        } catch (SQLException e) {
            System.out.print("SQL#2 failed closing the handle.\n");
            System.out.println(e.toString());
            System.exit(0);
        }

        return price;
    }

//-------------------------------------------------------------------------------------------

    public void insert_purchase(int cid, String club, String title, int year, int qnty) {
     String            queryText = "";     // The SQL text.
        PreparedStatement querySt   = null;   // The query handle.

       ctime = new Timestamp(System.currentTimeMillis());
       cdate = new SimpleDateFormat("yyyy-MM-dd-HH.mm.ss");
       when = cdate.format(ctime);
```

```
        queryText = "Insert into yrb_purchase values (?,?,?,?,?,?) ";

        // Prepare the query.
        try {
            querySt = conDB.prepareStatement(queryText);
        } catch (SQLException e) {
            System.out.println("SQL#1 failed in prepare");
            System.out.println(e.toString());
            System.exit(0);
        }

        // Execute the query.
        try {
            querySt.setInt(1, cid);
            querySt.setString(2, club);
            querySt.setString(3, title);
            querySt.setInt(4, year);
            querySt.setString(5, when);
            querySt.setInt(6, qnty);
            querySt.executeUpdate();
        } catch (SQLException e) {
            System.out.println("SQL#6 failed in update");
            System.out.println(e.toString());
            System.exit(0);
        }

        // We're done with the handle.
        try {
            querySt.close();
        } catch (SQLException e) {
            System.out.print("SQL#6 failed closing the handle.\n");
            System.out.println(e.toString());
            System.exit(0);
        }

    }

//**************************************************************************************************

    public static void main(String[] args)
    {
        CustApp ct = new CustApp();
    }
}
```