KK14203

OBJECT-ORIENTED PROGRAMMING

SEMESTER 2 SESSION 2019/2020

SECTION 1

INDIVIDUAL PROJECT:

ORDERING RECORD (CAFE)

PREPARED FOR

DR. SAMRY @ MOHD SHAMRIE SAININ

PREPARED BY

NAME	ELLY MAZLIN BINTI RAHIM
MATRIC NO.	BI19110025
DATE OF SUBMISSION	6th AUGUST 2020

JAVA CODE

```
import java.awt.Toolkit;
import java.awt.event.WindowEvent;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.logging.Level;
import java.util.logging.Logger;
import javax.swing.JOptionPane;
import javax.swing.table.DefaultTableModel;

public class Ordering_record extends javax.swing.JFrame {
    public Ordering_record() {
        initComponents();
    }

    DefaultTableModel model;
    String item;
    int price;
    Connection connect;
    PreparedStatement pstatement;
    PreparedStatement pstatement2;

    public void exit(){
    WindowEvent winClosingEvent = new WindowEvent(this, WindowEvent.WINDOW_CLOSING);
    Toolkit.getDefaultToolkit().getSystemEventQueue().postEvent(winClosingEvent);
    }

    @SuppressWarnings("unchecked")
    private void initComponents() {

        jLabel1 = new javax.swing.JLabel();
        jLabel2 = new javax.swing.JLabel();
```

```
cbxAmericano = new javax.swing.JCheckBox();
cbxLatte = new javax.swing.JCheckBox();
cbxCappucino = new javax.swing.JCheckBox();
cbxEspresso = new javax.swing.JCheckBox();
cbxMacchiato = new javax.swing.JCheckBox();
cbxMocha = new javax.swing.JCheckBox();
jScrollPane1 = new javax.swing.JScrollPane();
jTable1 = new javax.swing.JTable();
jLabel3 = new javax.swing.JLabel();
txtquantity = new javax.swing.JSpinner();
btnAdd = new javax.swing.JButton();
btnRemove = new javax.swing.JButton();
btnCalculate = new javax.swing.JButton();
jLabel4 = new javax.swing.JLabel();
jLabel5 = new javax.swing.JLabel();
jLabel6 = new javax.swing.JLabel();
txtTotal_Price = new javax.swing.JTextField();
txtPayment = new javax.swing.JTextField();
txtBalance = new javax.swing.JTextField();
jMenuBar1 = new javax.swing.JMenuBar();
menuExit = new javax.swing.JMenu();
jMenuItem1 = new javax.swing.JMenuItem();
jLabel1.setText("jLabel1");
setDefaultCloseOperation(javax.swing.WindowConstants.EXIT_ON_CLOSE);
jLabel2.setFont(new java.awt.Font("Tahoma", 0, 14)); // NOI18N
jLabel2.setText("SUNSHINE CAFE MENU");
cbxAmericano.setText("Americano");
cbxLatte.setText("Latte");
cbxCappucino.setText("Cappucino");
cbxEspresso.setText("Espresso");
cbxMacchiato.setText("Macchiato");
cbxMocha.setText("Mocha");

jTable1.setModel(new javax.swing.table.DefaultTableModel(
    new Object [][] {
    },
    new String [] {
        "Item", "Quantity", "Price", "Total "
    }
));
jScrollPane1.setViewportView(jTable1);
jLabel3.setText("Quantity");
btnAdd.setText("Add");
btnAdd.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnAddActionPerformed(evt);
    }
});

btnRemove.setText("Remove");
btnRemove.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnRemoveActionPerformed(evt);
    }
});

btnCalculate.setText("Calculate");
btnCalculate.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        btnCalculateActionPerformed(evt);
    }
});

jLabel4.setText("Total Price");
jLabel5.setText("Payment");
jLabel6.setText("Balance");

txtPayment.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        txtPaymentActionPerformed(evt);
    }
});

menuExit.setText("File");
```

```
        jMenuItem1.setAccelerator(javax.swing.KeyStroke.getKeyStroke(java.awt.event.KeyEvent.VK_C,
java.awt.event.InputEvent.ALT_MASK | java.awt.event.InputEvent.CTRL_MASK));
        jMenuItem1.setText("Exit");
        jMenuItem1.addActionListener(new java.awt.event.ActionListener() {
            public void actionPerformed(java.awt.event.ActionEvent evt) {
                jMenuItem1ActionPerformed(evt);
            }
        });
        menuExit.add(jMenuItem1);
        jMenuBar1.add(menuExit);
        setJMenuBar(jMenuBar1);

        javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());
        getContentPane().setLayout(layout);
        layout.setHorizontalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(239, 239, 239)
                        .addComponent(jLabel2))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(52, 52, 52)
                        .addComponent(cbxAmericano)
                        .addGap(17, 17, 17)
                        .addComponent(cbxLatte)
                        .addGap(18, 18, 18)
                        .addComponent(cbxCappucino)
                        .addGap(18, 18, 18)
                        .addComponent(cbxEspresso)
                        .addGap(18, 18, 18)
                        .addComponent(cbxMacchiato)
                        .addGap(18, 18, 18)
                        .addComponent(cbxMocha)))
                .addGap(0, 0, Short.MAX_VALUE))
            .addGroup(layout.createSequentialGroup()
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING)
                    .addGroup(layout.createSequentialGroup()
                        .addGap(22, 22, 22)
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)
                            .addComponent(btnCalculate, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addComponent(btnAdd, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
                            .addGroup(layout.createSequentialGroup()
                                .addComponent(jLabel3)
                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)
                                .addComponent(txtquantity, javax.swing.GroupLayout.PREFERRED_SIZE, 71,
javax.swing.GroupLayout.PREFERRED_SIZE))
                            .addComponent(btnRemove, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
                        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 33,
Short.MAX_VALUE)
                        .addComponent(jScrollPane1, javax.swing.GroupLayout.PREFERRED_SIZE, 424,
javax.swing.GroupLayout.PREFERRED_SIZE))
                    .addGroup(layout.createSequentialGroup()
                        .addGap(106, 106, 106)
                        .addComponent(txtTotal_Price, javax.swing.GroupLayout.PREFERRED_SIZE, 123,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(txtPayment, javax.swing.GroupLayout.PREFERRED_SIZE, 123,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(18, 18, 18)
                        .addComponent(txtBalance, javax.swing.GroupLayout.PREFERRED_SIZE, 123,
javax.swing.GroupLayout.PREFERRED_SIZE)
                        .addGap(0, 0, Short.MAX_VALUE)))
                .addGap(27, 27, 27))
            .addGroup(layout.createSequentialGroup()
                .addGap(144, 144, 144)
                .addComponent(jLabel4)
                .addGap(82, 82, 82)
                .addComponent(jLabel5)
                .addGap(99, 99, 99)
                .addComponent(jLabel6)
                .addContainerGap(javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        );
        layout.setVerticalGroup(
            layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(layout.createSequentialGroup()
                .addGap(23, 23, 23)
                .addComponent(jLabel2)
                .addGap(28, 28, 28)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(cbxAmericano)
                    .addComponent(cbxLatte)
                    .addComponent(cbxCappucino)
                    .addComponent(cbxEspresso)
                    .addComponent(cbxMacchiato)
                    .addComponent(cbxMocha))
                .addGap(28, 28, 28)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .addGroup(layout.createSequentialGroup()
                        .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                            .addComponent(jLabel3)
                            .addComponent(txtquantity, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                        .addGap(18, 18, 18)
                        .addComponent(btnAdd)
                        .addGap(18, 18, 18)
                        .addComponent(btnRemove)
                        .addGap(18, 18, 18)
                        .addComponent(btnCalculate))
                    .addComponent(jScrollPane1, javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.PREFERRED_SIZE, 143, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addGap(36, 36, 36)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(jLabel4)
                    .addComponent(jLabel5)
                    .addComponent(jLabel6))
                .addGap(18, 18, 18)
                .addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)
                    .addComponent(txtTotal_Price, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                    .addComponent(txtBalance, javax.swing.GroupLayout.PREFERRED_SIZE, javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
```

```
                    .addComponent(txtPayment,                                  javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE))
                .addContainerGap(26, Short.MAX_VALUE))
        );

        pack();
    }

    public void sales(){
        int id = 0;
        try {
            Class.forName("com.mysql.jdbc.Driver");
            connect = DriverManager.getConnection("jdbc:mysql://localhost/sunshinecafe", "root", "");

            String total_price = txtTotal_Price.getText();
            String balance = txtBalance.getText();
            String payment = txtPayment.getText();

            String query = "insert into sales(total_price, payment, balance)values(?, ?, ?)";
            pstatement = connect.prepareStatement(query, Statement.RETURN_GENERATED_KEYS);
            pstatement.setString(1, total_price);
            pstatement.setString(2, payment);
            pstatement.setString(3, balance);
            pstatement.executeUpdate();
            ResultSet rs = pstatement.getGeneratedKeys();

            if(rs.next()){
                id = rs.getInt(1);
            }
            int row = jTable1.getRowCount();

            String    query2    =    "insert    into    order_record(order_record_id,    item_name,    price,    quantity,
total_price)values(?, ?, ?, ?, ?)";
            pstatement2 = connect.prepareStatement(query2);

            String item_name = "";
            int price;
            int quantity;
            int total = 0;
            for(int a = 0; a < jTable1.getRowCount(); a++){
                item_name = (String)jTable1.getValueAt(a, 0);
                price = (int)jTable1.getValueAt(a, 2);
                quantity = (int)jTable1.getValueAt(a, 1);
                total = (int)jTable1.getValueAt(a, 3);

                pstatement2.setInt(1, id);
                pstatement2.setString(2, item_name);
                pstatement2.setInt(3, price);
                pstatement2.setInt(4, quantity);
                pstatement2.setInt(5, total);
                pstatement2.executeUpdate();
            }
          JOptionPane.showMessageDialog(this, "Calculation Done");
        } catch (ClassNotFoundException ex) {
            Logger.getLogger(Ordering_record.class.getName()).log(Level.SEVERE, null, ex);
        } catch (SQLException ex) {
            Logger.getLogger(Ordering_record.class.getName()).log(Level.SEVERE, null, ex);
        }
    }
    private void btnRemoveActionPerformed(java.awt.event.ActionEvent evt) {

        model.removeRow(jTable1.getSelectedRow());
        int Total_Price = 0;
        for(int a = 0; a < jTable1.getRowCount(); a++){
            Total_Price = Total_Price + Integer.parseInt(jTable1.getValueAt(a, 3).toString());
        }
        txtTotal_Price.setText(Integer.toString(Total_Price));
    }
    private void txtPaymentActionPerformed(java.awt.event.ActionEvent evt) {
    }
    private void btnAddActionPerformed(java.awt.event.ActionEvent evt) {
        if(cbxAmericano.isSelected() == true){
            item = "Americano";
            price = 4;
        }else if(cbxLatte.isSelected() == true){
            item = "Latte";
            price = 6;
        }else if(cbxCappucino.isSelected() == true){
            item = "Cappucino";
            price = 4;
        }else if(cbxEspresso.isSelected() == true){
            item = "Espresso";
            price = 5;
        }else if(cbxMacchiato.isSelected() == true){
            item = "Macchiato";
            price = 5;
        }else if(cbxMocha.isSelected() == true){
            item = "Mocha";
            price = 4;
        }

        int quantity = Integer.parseInt(txtquantity.getValue().toString());
        int total = quantity * price;
        model = (DefaultTableModel)jTable1.getModel();

        model.addRow(new Object[]{
            item,
            quantity,
            price,
            total
        });

        int Total_Price = 0;
        for(int a = 0; a < jTable1.getRowCount(); a++){
            Total_Price = Total_Price + Integer.parseInt(jTable1.getValueAt(a, 3).toString());
        }
        txtTotal_Price.setText(Integer.toString(Total_Price));
    }

    private void btnCalculateActionPerformed(java.awt.event.ActionEvent evt) {
        int total = Integer.parseInt(txtTotal_Price.getText());
        int payment = Integer.parseInt(txtPayment.getText());
```

```
        int balance = payment - total;
        txtBalance.setText(String.valueOf(balance));
        sales();
    }

    private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
        exit();
    }

    public static void main(String args[]) {
        try {
            for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {
                if ("Nimbus".equals(info.getName())) {
                    javax.swing.UIManager.setLookAndFeel(info.getClassName());
                    break;
                }
            }
        } catch (ClassNotFoundException ex) {
            java.util.logging.Logger.getLogger(Ordering_record.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (InstantiationException ex) {
            java.util.logging.Logger.getLogger(Ordering_record.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (IllegalAccessException ex) {
            java.util.logging.Logger.getLogger(Ordering_record.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        } catch (javax.swing.UnsupportedLookAndFeelException ex) {
            java.util.logging.Logger.getLogger(Ordering_record.class.getName()).log(java.util.logging.Level.SEVERE,
null, ex);
        }
        java.awt.EventQueue.invokeLater(new Runnable() {
            public void run() {
                new Ordering_record().setVisible(true);
            }
        });
    }

    private javax.swing.JButton btnAdd;
    private javax.swing.JButton btnCalculate;
    private javax.swing.JButton btnRemove;
    private javax.swing.JCheckBox cbxAmericano;
    private javax.swing.JCheckBox cbxCappucino;
    private javax.swing.JCheckBox cbxEspresso;
    private javax.swing.JCheckBox cbxLatte;
    private javax.swing.JCheckBox cbxMacchiato;
    private javax.swing.JCheckBox cbxMocha;
    private javax.swing.JLabel jLabel1;
    private javax.swing.JLabel jLabel2;
    private javax.swing.JLabel jLabel3;
    private javax.swing.JLabel jLabel4;
    private javax.swing.JLabel jLabel5;
    private javax.swing.JLabel jLabel6;
    private javax.swing.JMenuBar jMenuBar1;
    private javax.swing.JMenuItem jMenuItem1;
    private javax.swing.JScrollPane jScrollPane1;
    private javax.swing.JTable jTable1;
    private javax.swing.JMenu menuExit;
    private javax.swing.JTextField txtBalance;
    private javax.swing.JTextField txtPayment;
    private javax.swing.JTextField txtTotal_Price;
    private javax.swing.JSpinner txtquantity;

}
```

OBJECT ORIENTED CONCEPT IMPLEMENTATION

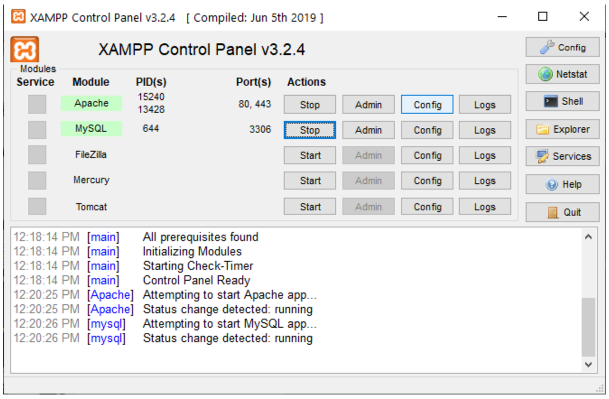
The concept of Object-oriented that was implemented in this project is abstraction, encapsulation, object, class and many more. The abstraction is like the objects, classes and variables that was declare and form in this project. The class is representing the set of properties or methods that are common to all objects of one type. There are many objects that was formed in this project as well as the method. Other than that, for the Encapsulation that implemented in this project, there are fields within a class private was formed, then providing access to them via public methods.

READ AND WRITE IMPLEMENTATION

When using this project, the data will be saved into the database. For this project, the order record will be saved into the browser database. While doing this, I need to create an object of the class and specify the filename or the directory name. The filename in this project is "jdbc:mysql://localhost/sunshinecafe".

USER MANUAL

The database of this project is using the browser. The link of this database is http://localhost/phpmyadmin/db_structure.php?db=sunshinecafe . But, before you run this project, you must download the XAMPP Control Panel so that you can save the order that you make into the database and after that change the setting as below.



XAMPP Control Panel v3.2.4 [Compiled: Jun 5th 2019]

XAMPP Control Panel v3.2.4

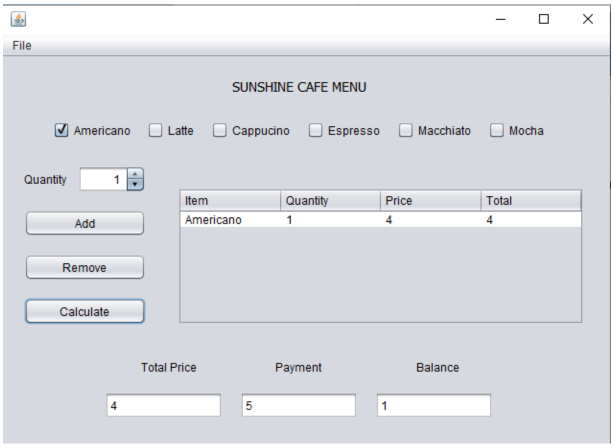
Service	Module	PID(s)	Port(s)	Actions			
□	Apache	15240 13428	80, 443	Stop	Admin	Config	Logs
□	MySQL	644	3306	Stop	Admin	Config	Logs
□	FileZilla			Start	Admin	Config	Logs
□	Mercury			Start	Admin	Config	Logs
□	Tomcat			Start	Admin	Config	Logs

```
12:18:14 PM  [main]    All prerequisites found
12:18:14 PM  [main]    Initializing Modules
12:18:14 PM  [main]    Starting Check-Timer
12:18:14 PM  [main]    Control Panel Ready
12:20:25 PM  [Apache]  Attempting to start Apache app...
12:20:25 PM  [Apache]  Status change detected: running
12:20:26 PM  [mysql]   Attempting to start MySQL app...
12:20:26 PM  [mysql]   Status change detected: running
```

Figure 1: Setting of the XAMPP Control Panel.

The procedure of using this system is as following:

1. Firstly, choose the drinks that you want.
2. Secondly, choose how many drinks that you want to buy and click **Add** button.
3. Thirdly, select order that you want to remove and click **Remove** button.
4. Then, when you are done with your order, the overall price that you should pay will show up at the **Total Price** section.
5. Next, put how much you will pay at the **Payment** section and the balance will show up at the **Balance** section.
6. The record of your order will be saved into the database.



SUNSHINE CAFE MENU

☑ Americano □ Latte □ Cappucino □ Espresso □ Macchiato □ Mocha

Quantity 1

Item	Quantity	Price	Total
Americano	1	4	4

Total Price: 4

Payment: 5

Balance: 1

Figure 2: How the system works.

The figure below shows the database of the recorded order. There are 2 tables name **order_record** and **sales**. **sales** table will save the data of the total price, payment and the balance. While the **order_record** will save the data of the item name, price, quantity and the total price of the specific drink.

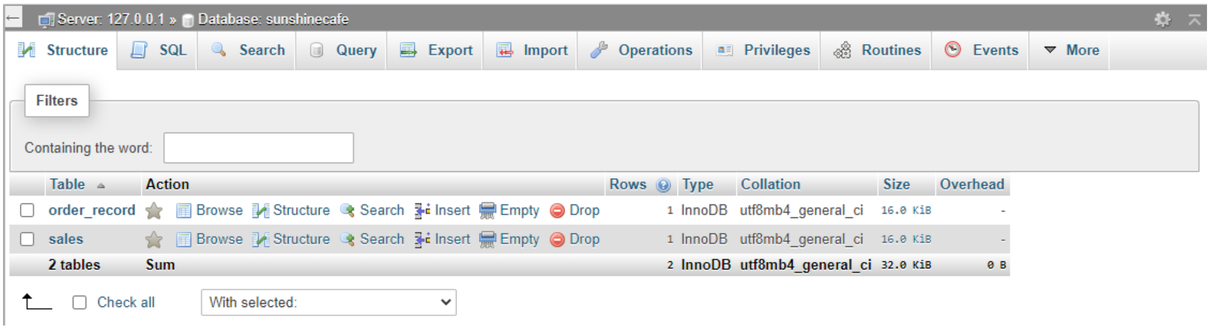
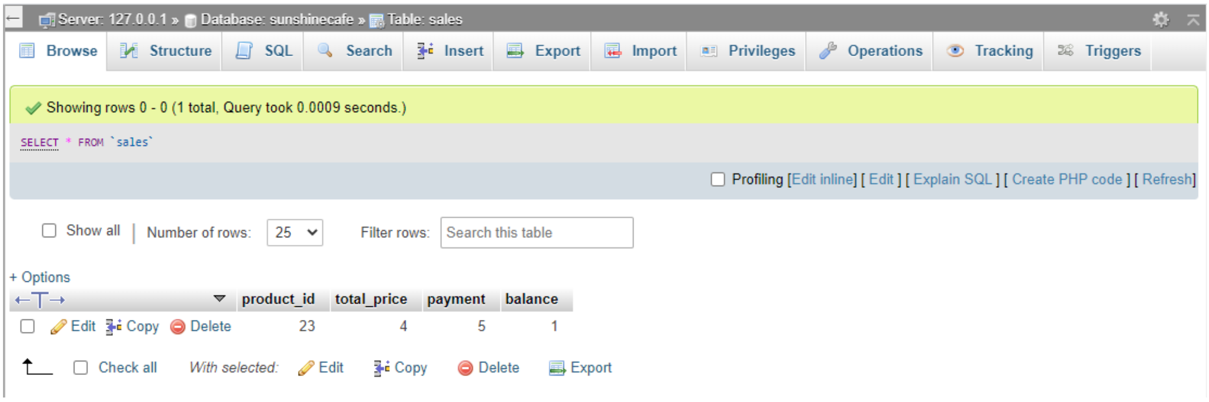


	Table	Action	Rows	Type	Collation	Size	Overhead
□	order_record	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
□	sales	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	16.0 KiB	-
	2 tables	Sum	2	InnoDB	utf8mb4_general_ci	32.0 KiB	0 B


Server: 127.0.0.1 » Database: sunshinecafe » Table: sales
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers

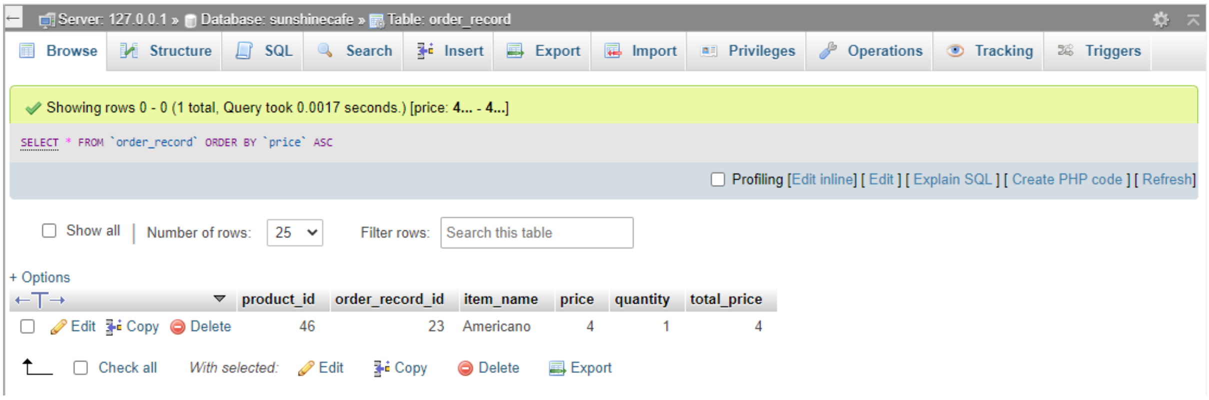
✓ Showing rows 0 - 0 (1 total, Query took 0.0009 seconds.)

```
SELECT * FROM `sales`
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25
Filter rows: Search this table
+ Options

	product_id	total_price	payment	balance
□ Edit Copy Delete	23	4	5	1

□ Check all
With selected: Edit Copy Delete Export


Server: 127.0.0.1 » Database: sunshinecafe » Table: order_record
Browse
Structure
SQL
Search
Insert
Export
Import
Privileges
Operations
Tracking
Triggers

✓ Showing rows 0 - 0 (1 total, Query took 0.0017 seconds.) [price: 4... - 4...]

```
SELECT * FROM `order_record` ORDER BY `price` ASC
```

□ Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]
□ Show all | Number of rows: 25
Filter rows: Search this table
+ Options

	product_id	order_record_id	item_name	price	quantity	total_price
□ Edit Copy Delete	46	23	Americano	4	1	4

□ Check all
With selected: Edit Copy Delete Export

Figure 3: The database of the recorded order.