OODS CA3

CUSTOMER INVOICE MANAGEMENT SYSTEM

Liam Durkan

C00264405

Description

A customer invoice management system is designed to handle the data relating to products and customers, using this data invoices can be created and stored to keep track of payments and products sold. The system allows products to be entered into a database where system users can edit, delete and update product information. Customers can be added into the database where their information is stored and used to create invoices for purchases.

The system I have developed uses MYSQL workbench that has been installed onto my machine. Using the mySQL external library and a Java Database Connector queries can be sent to the database and preform Create, Update, Retrieve and Delete Functions.
A GUI was developed using java Swing to build an interface that the user can interact with.

Requirements

Requirements for this project;

- Create Three or more tables
    - Provide inner join over these tables
- Front End GUI
    - Providing create, update, return and delete operations on the database
    - Demonstrate a variety of swing components such as JTextboxes & Buttons
- Provide error handling in the code

Database Tables

Customer Table

customer_id	firstName	lastName	phone	email	street	county	postcode
29	Liam	Durkan	0987654321	liam@email.com	3 Town Lane	Meath	V24 GF65
30	Kyle	Byen	3425364782	kyle@email.com	4 Mill Court	Sligo	D38 HK97
31	Sarah	Smyth	0982536874	sarah34@email.com	Ringlestown	Dublin	D35 HG66
32	Joan	Byrne	0263873648	joan@email.com	New Road	Carlow	J87 HG22
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Column	Datatype	PK	NN	UQ	B...	UN	ZF	AI	G	Default / Expression
customer_id	INT	☑	☑	☐	☐	☐	☐	☑	☐	
firstName	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
lastName	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
phone	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
email	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
street	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
county	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
postcode	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
<click to edit>		☐	☐	☐	☐	☐	☐	☐	☐	

The customer Table assigns each customer with an individual customer_id, no two customers will ever have the same customer_id and each customer_id is unique within the table.
The customer table stores the customer name, phone number, email, street, address, county and postcode.
When entering the phone number, the java program will first check that the phone number is 10 digits long to allow insertion into the table.

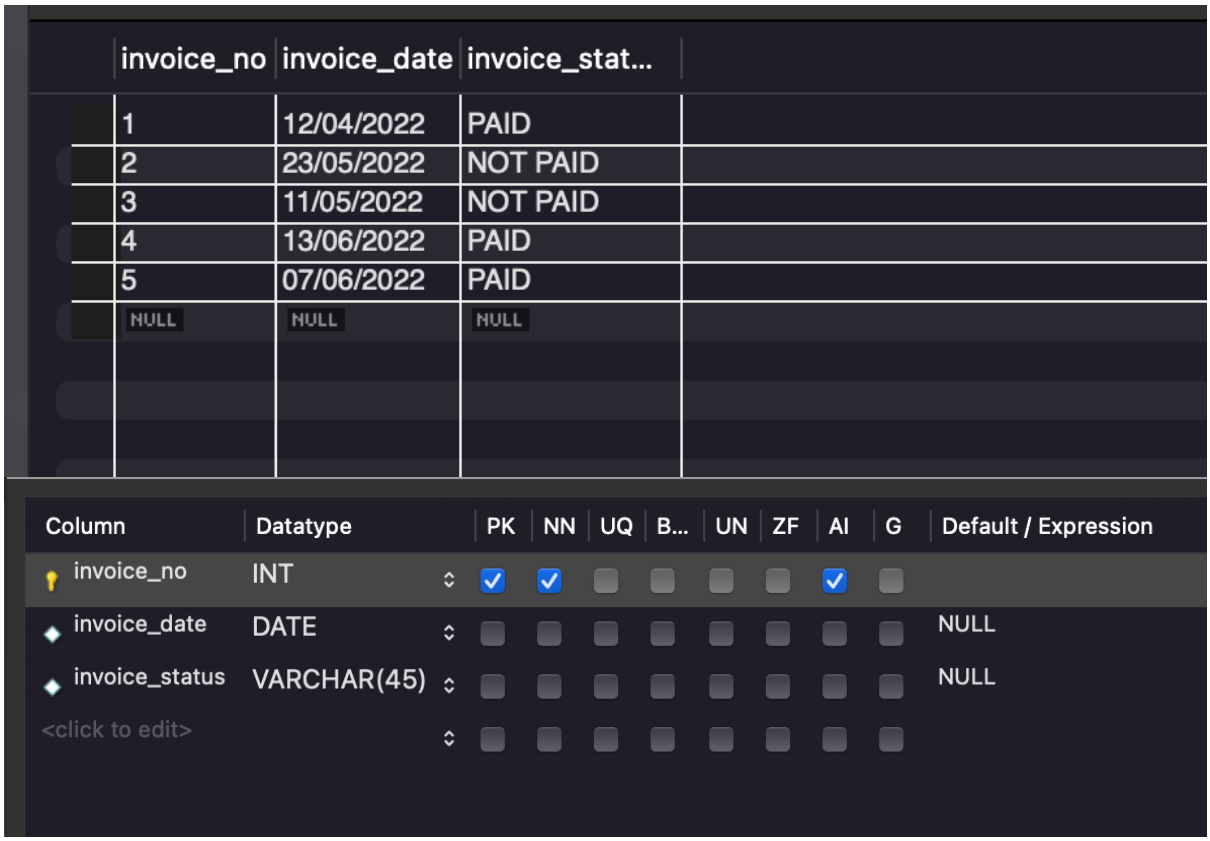
Product Table

productId	name	catagory	description	costPrice	sellPrice
8	Hammer	Tools	Hammer 12" Black	2.99	23.99
9	Screwdriver	Tools	Phillips head orange	3.45	11.99
10	Drill	Power Tools	12 bit set battery drill	27.76	74.99
11	Safety Vest	Saftey	High Vis Yellow Medium	1.25	12.99
12	Safety Vest	Saftey	High Vis Yellow Large	1.30	12.99
NULL	NULL	NULL	NULL	NULL	NULL

Column	Datatype	PK	NN	UQ	B...	UN	ZF	AI	G	Default / Expression
customer_id	INT	☑	☑	☐	☐	☐	☐	☑	☐	
firstName	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
lastName	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
phone	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
email	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
street	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
county	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
postcode	VARCHAR(45)	☐	☑	☐	☐	☐	☐	☐	☐	
<click to edit>		☐	☐	☐	☐	☐	☐	☐	☐	

The product table assigns each product with an individual productId, no two customers will ever have the same productId and each productid is unique within the table.
The product table stores the product name, category of product, a brief description about the product, the cost and selling price for the product.

Invoice Table

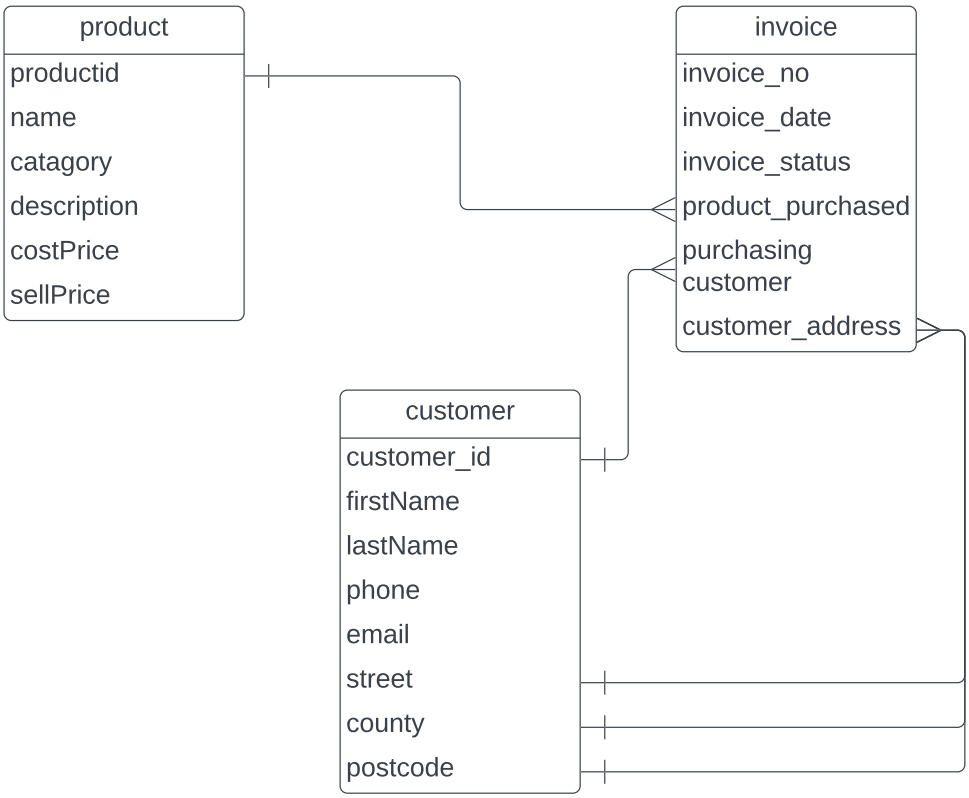


invoice_no	invoice_date	invoice_stat...
1	12/04/2022	PAID
2	23/05/2022	NOT PAID
3	11/05/2022	NOT PAID
4	13/06/2022	PAID
5	07/06/2022	PAID
NULL	NULL	NULL

Column	Datatype	PK	NN	UQ	B...	UN	ZF	AI	G	Default / Expression
invoice_no	INT	☑	☑	☐	☐	☐	☐	☑	☐	
invoice_date	DATE	☐	☐	☐	☐	☐	☐	☐	☐	NULL
invoice_status	VARCHAR(45)	☐	☐	☐	☐	☐	☐	☐	☐	NULL
<click to edit>		☐	☐	☐	☐	☐	☐	☐	☐	

The invoice table assigns each invoice a primary key when created.
The invoice table stores the invoice date and invoice status.

ER Diagram


product
productid
name
catagory
description
costPrice
sellPrice
invoice
invoice_no
invoice_date
invoice_status
product_purchased
purchasing customer
customer_address
customer
customer_id
firstName
lastName
phone
email
street
county
postcode

Code Snippets

1. Register Product JButton action listener

module-info.java | Driver.java | *product.java | customer.java | invoice.java

```
123
124         //Register button MYSQL Statement
125         register.addActionListener(new ActionListener() {
126             public void actionPerformed(ActionEvent e) {
127                 //create strings with data from JTextboxes
128                 String Name = name.getText();
129                 String Catagory = catagory.getText();
130                 String Description = description.getText();
131                 String costPrice = costprice.getText();
132                 String sellPrice = sellprice.getText();
133 
134                 //connection to database
135                 try {
136                     Connection connection = DriverManager.getConnection(jdbcURL, username, password);
137 
138                     //if the connection is successful
139                     if(connection != null) {
140 
141                         //java sql statement to insert into table
142                         String sql = "INSERT INTO product (name,catagory,description,costPrice,sellPrice)"
143                                 + " VALUES (?,?,?,?,?)";
144 
145                         PreparedStatement statement = connection.prepareStatement(sql);
146                         statement.setString(1,  Name);
147                         statement.setString(2,  Catagory);
148                         statement.setString(3,  Description);
149                         statement.setString(4,  costPrice);
150                         statement.setString(5,  sellPrice);
151 
152                         int rows = statement.executeUpdate();
153 
154                         //upon successful insertion into database
155                         if(rows > 0) {
156                             System.out.println("A new Product has been inserted sucessfully.");
157                         }
158                     connection.close();//close connection to database
159                     }
160 
161                 }//catch block if connection to database is not successful
162                 catch (SQLException ex) {
163                     ex.printStackTrace();
164                 }
165             }
166 
167         });
168 
```

The code above is an action listener that inserts data into the table.
When the register button is pressed strings are collected from the JTextfields.
A try block is used to connect to the database, upon successful connection the SQL statement will insert into the database using the INSERT query. If successful the system will print a successful string into the console.
If the database connection cannot successfully be the printstacktrace will show the erroe in the console.

2. Creating the JFrame in the Main Class

```
//Create the JFrame
JFrame frame = new JFrame("Customer Invoice Managment System");//Tile at top of Frame
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);//Exit application on close click
frame.setSize(1000, 600);//Set the size of JFrame

//Creating the panel and adding components
JPanel navPanel = new JPanel();

//Title at top of page
JLabel lblTitle = new JLabel("NAVAGATION MENU");//Create JLable For Title
lblTitle.setBounds(460, 40, 325, 20);//Set Location on JFrame
lblTitle.setFont(new Font("Serif", Font.PLAIN, 22));//Set font size and type
navPanel.add(lblTitle);//add JLable to panel

//Add functionality buttons that call individual classes
JButton customer = new JButton("Customer");//create customer button
navPanel.add(customer);//add customer button to panel
JButton product = new JButton("Product");//create product button
navPanel.add(product);//add product button to panel
JButton invoice = new JButton("Invoice");//create invoice button
navPanel.add(invoice);//add invoice button to panel

//Adding Components to the frame
```

3. Selection on a row in JTable and populating into JTextFields on click.

```
//when row select button is pressed
rowSel.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {

     DefaultTableModel model = (DefaultTableModel)table.getModel();
    int row = table.getSelectedRow();

    //
    name.setText(model.getValueAt(row, 0).toString());
    catagory.setText(model.getValueAt(row, 1).toString());
    description.setText(model.getValueAt(row, 2).toString());
    costprice.setText(model.getValueAt(row, 3).toString());
    sellprice.setText(model.getValueAt(row, 4).toString());

    }
});
//when update button is pressed
```