JDBC Assignment

Makeup Store

Natalia Palej

A00279259

Year 4

Software Design with Artificial Intelligence for Cloud Computing

Contents

[Introduction 2](#_Toc182434378)

[GUI 2](#_Toc182434379)

[Project 4](#_Toc182434380)

[Triggers 4](#_Toc182434381)

[Stored Procedures 5](#_Toc182434382)

[Views 5](#_Toc182434383)

[Conclusion 6](#_Toc182434384)

[References 6](#_Toc182434385)

# Introduction

Makeup Store Project is designed to manage customer orders. The makeupdb database consists of 4 tables, where each is pre-populated with sample data:

* customers (11 columns)
* products (11 columns)
* orders (6 columns)
* order\_details (5 columns)

The application divides into two sets of functionalities – one for customer and one for admin. These features are accessible through a user-friendly Java GUI interface (GUI.java class). The main functionalities include:

|  |  |
| --- | --- |
| **Customer** | **Admin** |
| Register | Login |
| Login | Add New Product |
| Order Products | View Products |
| View All Orders | View Specific Product |
| View Specific Order | Update Product |
| Update Account | Delete Product |
| Delete Account | View Sales with Export Function |
| Exit | View Order Invoice with Export Function |

# GUI

A screenshot of a computer registration

Description automatically generatedA screenshot of a computer login

Description automatically generatedGUI is straightforward and easy to use. Once GUI.java is ran, login screen pops up, where user can either Login or Register.

If Register is selected, user will be prompted to enter few personal details. All the fields are pre-populated with placeholders to guide the user on the inputs.

After successful login, user is brought to main Makeup Store view, where they can browse through different products. Products display can be sorted by selecting various filters in the drop-down menu. User can adjust product quantity by selecting custom component of quantity editor that will display when table row is highlighted.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedUser can also explore different options like “View Orders” to see all orders they’ve made along with details of specific order. “Update Account” to change personal details, “Delete Account” to remove account or “Exit” to close the application.

A screenshot of a computer

Description automatically generated

Admin view varies a lot from customer’s view. Admin can add, update, view or delete product.

Admin can also view and export invoice and sales.

A screenshot of a computer screen

Description automatically generatedA screenshot of a computer

Description automatically generated

# Project

## Triggers

**tr\_calculate\_total\_item\_cost\_before\_insert** – this trigger calculates the total\_item\_cost and inserts the result into order\_details table. It automates the calculation for price \* quantity, making the process efficient and accurate.

A screenshot of a computer code

Description automatically generated

**tr\_calculate\_order\_amount\_after\_insert** – this trigger calculates the tax\_amount and total\_order\_amount. It takes the total\_item\_cost from previous trigger and applies the tax amount for the entire order, it also outputs total\_order\_amount which includes sum of all product quantities and tax\_amount.

## Stored Procedures

A screenshot of a computer program

Description automatically generated

**sp\_create\_order** – this procedure creates new order for a customer. It accepts customer ID as input and returns newly created order ID. If the transaction is successful, it commits it and confirm the order ID. In case of an error, it rollbacks and displays failure message.

**sp\_add\_product\_to\_order** – this procedure allows to add multiple products to single order. It takes the order\_id, product\_code and quantity as input. First, it checks if there’s enough stock. If so, it adds the product to the order\_details table and reduces stock in products table.

**sp\_add\_new\_product** – this procedure adds new product to the makeupdb. It is called within saveProductToDatabase method in AdminScreen class.

**sp\_update\_product** – this procedure updates existing product details based on user input. It is called within updateProductDetails method in AdminScreen class.

## Views

**customer\_orders\_view** – this view gives detailed information for each order. It joins customer order and order\_details tables. It is useful for accessing complete customer orders.

**customer\_details\_view** – this view is designed to isolate customer details. It is intended to streamline retrieving customer data (like name, address, contact details) for invoices.

**sales\_by\_category\_view**, **sales\_by\_product\_view**, **sales\_by\_brand\_view** – these views show sales data grouped by category/product/brand. It provides total quantity sold and sales amount. It is useful for analysing sales performance across orders.

# Conclusion

Makeup Store project is an efficient management application tailored for beauty shop. It allows for easy management of products, orders and customer data.

Throughout the project, I have learned how to design and implement stored procedures, views and triggers to simplify Java code. I also learnt how to handle essential tasks directly in MySQL, which optimizes performance and reduces redundancy in code. One of the biggest challenges was understanding the purpose and logic behind each procedure and trigger.

This project taught me how to efficiently integrate Java and MySQL, it helped me understand how to link GUI with database operations and confirmed that dealing with GUI in Java isn’t my favourite area.

Overall, the application design runs efficiently, it provides smooth and user-friendly layout and handles database accurately.

# Charts

# References

1. OpenAI, ChatGPT. Sample data for database tables generated using ChatGPT.
2. Boots Ireland. Product prices taken from <https://www.boots.ie>
3. Beauty Bay. Product prices taken from <https://www.beautybay.com>
4. Tutorials Point. “Java Swing Layouts.” <https://www.tutorialspoint.com/swing/swing_layout>
5. GeeksforGeeks. “Java Swing JTable.” <https://www.geeksforgeeks.org/java-swing-jtable/>
6. Javatpoint. “Java Swing Tutorial.” <https://www.javatpoint.com/java-swing>