Final Submission Document

# Details about the files submitted

Under the submitted link, you will find two other folders whose names correspond to their content. The first folder should contain the SQL dump asked for, the content directly extracted from the database. To properly view the database queries and triggers, click on the “COSC 3380 | Team 9 : Database.sql” file and click on the ellipse next to the download button. Choose the option to open in a new window. After that, you should be able to view the entire report without any trouble. Furthermore, the "front-end material" folder should hold the README file, allowing you to run the content on your device. All of the folder names should be clear in their content.

# Rundown

## *What type of user roles are in our application and what data can be added, modified, and edited?*

Our website has two user roles: the customer and the admin.

CUSTOMER FUNCTIONALITIES:

The users only have access to the products, and they can add and remove items from their shopping cart afterward they can checkout. They cannot add any tuples to the database and can only remove them from the database when they checkout. Once they check out, the “shoppingCart” entity will lose a tuple.

ADMIN FUNCTIONALITIES:

They have access to the same products the customer has; however, they can add products and process orders. These actions will impact the database, and the “Add Product” tab will add a new tuple to the product entity. Furthermore, the “Order Processing” tab will show you the rundown of all the orders placed. In this tab, you can update the status of the orders, which will notify the user once they log into their account. You can click the order ID to view the user item’s details and receipt.

## *Semantic constraints, which are implemented as triggers*

There are a total of three triggers, and you can see the code for these triggers under the tables "orders," "Products," and "shoppingCart."

Under the "orders" table, we have "updateOrderItem." The "updateOrderItem" helps relocate the data from the shopping cart to the "order\_item" table to help store the data once a new order is inserted, as the system erases the data from the shopping cart after the user is done checking out.

In the "products" table, there are triggers "notEnoughInvent" and "restockInventory" meant to manage the inventory notifications. "notEnoughInventory" will limit how much the customer can purchase, and the "restockInventory" will notify the system to increase the stock by ten when the product quantity is less than five.

In the "shoppingCart" table, there are "notEnoughInvent" and "removeFromCart"; the first one works closely similar to the one in the products table. Furthermore, the "removeFromCart" will ensure that whichever item the customer puts to zero is removed from the shopping cart.

## *Types of queries/reports available in your application.*

There are two types of reports available under the admin account. You can access the order and supplier reports under the “Admin Reports” dropdown menu in the admin’s profile. These reports are automatically updated as they pull information from tables in our database. Therefore you can test the functionality by processing orders and seeing the results in the two reports below.

Report 1 (Total revenue between two date parameters):

This report will retrieve data from our orders, users, and visitors table to show a report to the admin of the total amount our website has earned based on the range provided. The ‘from’ and ‘to’ dates are used in our query to access and deliver data from each table to our front-end view, where the admin can get an idea of how much total revenue has accumulated within their chosen time period. This report uses three tables and three queries to show the order’s ID, date, and amount, the customer’s name, and the total sum of money earned according to the time period given.

Report 2 (Total quantity sold for a supplier):

This report provides our website’s admin with an analysis to see how each supplier performs in our store. Using the “order\_item,” “suppliers,” and “products” tables, we can display distinct orders occurring for each respective supplier. In the report view, we chose to show product ID, quantity sold, and order ID from our tables to demonstrate the different transactions occurring to calculate the total amount pledged for the supplier. This second report also utilized two queries that used the supplier ID as input to manipulate the data shown in the report. In the top right of the page, we have also decided to show the total sum of the products sold by the supplier based on whichever supplier the admin chooses to inspect. Using this information, the admin can better understand which suppliers are performing better or worse.

# Web Link; Users

Link: <https://group9poss.azurewebsites.net/>

\* Please note that you will have to log in with one of the following so you can navigate the website with little to no problem. \*

CUSTOMER CREDENTIALS

Username: Dave

Password: asdf

ADMIN CREDENTIALS

Username: admin

Password: COSC3380

Possible orders to cancel:

\* This is an option to see how the “statusUpdate” trigger works \*

Order Id User Id

237 14

226 43

224 40

Corresponding account information:

237

Username: rob

Password: yes123

226

Username: John Doe

Password: asdzxc

224

Username: RandomJoe1

Password: 123456