

# CS 166 Project Report

## Group Information

Group 62

Jackson Marolt, jmaro005

Roberto Martinez Garibay, rmart221

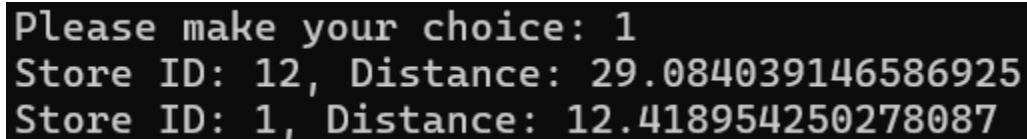
## Implementation Description

Include high-level description of your implementation (1 paragraph max)

We implemented an Amazon storefront like database where Users could shop at stores and managers could change their products around and order new products whenever they needed. There are also admin users that get access to changing any possible values, which include user profiles and store/product information. To implement this, we used Java and SQL. The code was primarily Java with a library that could execute SQL queries.

Include screenshots and/or code snippets for each query. In addition, explain how you implemented each query and what purpose it fulfills. (no more than 1-2 sentences per query)

### Query 1(Stores within 30 miles)



```
Please make your choice: 1
Store ID: 12, Distance: 29.084039146586925
Store ID: 1, Distance: 12.418954250278087
```

To implement this query, we had to use the calculate distance method, which is just the distance formula. We would use a query that would get all of the stores longitudes and latitudes and compare them to the users long/lat to get the distance and return those stores that were less than 30 miles away.

### Query 2(View Store's Product List)

```

Please make your choice: 2
Enter Amazon Store ID: 5
productname      numberofunits  priceperunit
7up              46           3
Pepsi            69           4
Lemonade         100          8
Brisk            24           3
Orange Juice     48           6
Donuts           42           7
Pudding          41           3
Ice Cream        95           6
Hot and Sour Soup 94           5
Egg              25           3

```

To implement this query, we basically just selected all of the products within a store and filtered them by store ID and just printed the results.

### Query 3(Place Order)

```

Please make your choice: 3
Enter Amazon Store ID: 12
Enter Product Name: Egg
Enter Amount: 5

```

NOTE\* We needed to find stores within 30 miles of us and know which products were in the store which required us to do queries 1 & 2 first! To implement this query, we first found stores within 30 miles, then we got the product name and the amount and put them in the orders table. Also ensured that they couldn't order more than what was available.

### Query 4(View 5 recent orders)

```

Please make your choice: 4
ordernumber      customerid  storeid  productname  unitsordered  ordertime
502      1      12      Egg          5      2024-03-17 22:54:04
452      1      12      Pepsi        43     2016-09-10 19:16:00
409      1      13      Hot and Sour Soup 20     2016-09-10 18:47:00
269      1      16      Hot and Sour Soup 5       2016-09-10 16:53:00
5         1      4       Orange Juice 20     2016-09-10 13:02:00

```

NOTE\* This query returned the five most recent orders for that person in general(NOT STORE SPECIFIC). To implement this query, we just grabbed all of the orders where the customer ID was the same as the user and we limited the output to 5.

### Query 5(Update Product)

```

Please make your choice: 5
Which store's products would you like to update?
4
Which product would you like to update?
Egg
Update amount or price? (amount/price)
price
Enter new price: 6

```

To implement this query, we had to ensure that the user was able to update the product in the first place, by either being manager at that store or an admin overall. Then, we had to find what that manager/admin wanted to update about that store and update it and add information to productUpdates table.

#### Query 6(5 recent product updates)

```

Please make your choice: 6
Which store's updates would you like to see? (Enter StoreID)
3

```

updatenumber	managerid	storeid	productname	updatedon
51	10	3	Egg	2024-03-17 22:38:17
3	10	3	Lemonade	2016-09-10 13:02:00

To implement this query, we got all of the rows in the product updates table and limited it to 5 and the store ID had to match the one passed in.

#### Query 7(5 Popular items)

```

Please make your choice: 7
Which store's 5 most popular products would you like to see? (Enter StoreID)
6

```

productname
Egg
Ice Cream
Orange Juice
7up
Donuts

To implement this query, we had to order the product based on the total amount of units they sold and return the top 5.

#### Query 8(5 Popular customers)

```

Which store's 5 most popular customers would you like to see? (Enter StoreID)
14
name      customerid
Bernard.Lubowitz      48
Dedric                93
Viola.Tillman         60
Joyce.Roob            63
Christa               42

```

To implement this query, we had to get the amount of times the user ordered from that store and return the five most.

### Query 9(Product supply request)

```

Please make your choice: 9
Which store would you like to request products for? (Enter StoreID)
3
Input Product Name:
Egg
Enter number of units needed:
100
Enter warehouse ID:
4

```

To implement this query, we had to check if the user was a manager. We then had to check which store they wanted to get their supply request to. Then they needed to ask how many units they needed and from which warehouse. Then, the products would be delivered to the store and the tables updated(ProductUpdates & Product).

### Query 10(10 most recent store orders)

```

Please make your choice: 10
Which store's 10 most recent orders would you like to see? (Enter StoreID)
3

```

orderid	customerid	storeid	productname	unitsordered	ordertime
461	83	3	7up	24	2016-09-10 19:24:00
454	64	3	Brisk	43	2016-09-10 19:19:00
445	3	3	Orange Juice	11	2016-09-10 19:11:00
426	28	3	Donuts	10	2016-09-10 18:57:00
424	24	3	Brisk	20	2016-09-10 18:54:00
418	12	3	Ice Cream	28	2016-09-10 18:50:00
415	35	3	Orange Juice	44	2016-09-10 18:49:00
406	82	3	Donuts	29	2016-09-10 18:43:00
394	89	3	Brisk	25	2016-09-10 18:35:00
371	16	3	7up	35	2016-09-10 18:19:00

To implement this query, we got the orders for the store we wanted and we ordered them by order number. Now normally you would order them by time, however, with how we implemented them, the order number only goes up. We then printed the 10 most recent from that store.

If you did any extra credit, provide screenshots and/or code snippets. Explain how you implemented the extra credit. (triggers/stored procedures, performance tuning, etc)

```
Please make your choice: 2
Enter Amazon Store ID: my name is jackson
ID must be a 32-bit Integer.
```

Some input validation. Basically, it would verify that a user's input was correct or not and tell the user what was wrong with the input.

```
CREATE OR REPLACE LANGUAGE plpgsql;
CREATE OR REPLACE FUNCTION orderNumberFunc()
RETURNS "trigger" AS
$BODY$
DECLARE
    numRows INTEGER;
BEGIN
    SELECT COUNT(*) INTO numRows FROM Orders;
    NEW.orderNumber := numRows + 1;
    RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql VOLATILE;

CREATE TRIGGER orderNumberTrigger BEFORE INSERT
ON Orders FOR EACH ROW
EXECUTE PROCEDURE orderNumberFunc();
```

Triggers. We implemented a few triggers that would create the order number/update number/request number for us by getting the total count of all the orders/updates/requests and adding one to it. This saved us a lot of time by not having to do some queries and was just a nice feature to implement.

### Problems/Findings

Include problems/findings you encountered while working on the project (1-2 paragraphs max)

This project came with many problems. The first problem was just getting the program to run in the first place. The other problem and easily the biggest was planning out a project like this. When the project starts to get big like this, especially when the extra credits are included, you need to have a plan or else the code is just going to be a jumbled mess. This was our biggest problem, we kind of did the required stuff without

much plan and it worked but when it came time for extra credit, it became harder to implement. I found this project to be very useful in developing SQL skills and planning skills. I also found that Java and SQL work very well together.

### **Contributions**

Include descriptions of what each member worked on (1 paragraph max)

Jackson Marolt: I worked on implementing the stores within 30 miles query, viewing store products query, place orders, checking the 5 most recent orders, updating products if you are a manager or admin, 10 most recent store orders, triggers, input validation, and project report.

Roberto Martinez Garibay: View 5 recent product updates, 5 most popular products, 5 most popular customers, product supply request