# Mobilization of Products Via Amazon Logistics

## Milestone: Final Project Report

Group 6

Ratna Manjeera Grandhi
Phani Bhavana Atluri

617-971-6775 (Telephone of Student 1)
513-206-2263 (Telephone of Student 2)

grandhi.r@northeastern.edu
atluri.p@northeastern.edu

**Percentage of Effort Contributed by Student1:_50%_**
**Percentage of Effort Contributed by Student2:_50%_**
**Signature of Student 1: Ratna Manjeera Grandhi**
**Signature of Student 2: Phani Bhavana Atluri**
**Submission Date: 12/10/2022**

## Executive Summary:

As online shopping is booming, retailers are forced to maintain same products at various warehouse locations so based on the customer location, the item can be shipped quickly and easily. With increased number of customers, increase number of addresses, increase number of products, the efficiency of the product reaching the customer has significantly decreased on the months for Amazon. A customer in our proposal is any person who is searching in our catalog of items, interested in an item, places the order for it and pay for it. Logistics who we chose to support is via Amazon as our primary logistics partner but as ancillaries also have all the standard shipping company connections. Depending on the loyalty of the customer, incentives or one time offers can be given to the customer to attract their attention which increases product sales. Our project will model the efficient way to attract the customer into buying the product from our market and have their item reach them fastest and cost-effective way.

## I. Introduction

Focusing on keen customer satisfaction would be our first and foremost priority in building confidence and trust in our company. Those customers spreading the word and social media advertising will be our source of increasing the number of customers, type of products to be sold, and according to the region maintain enough products to mobilize the orders instantly when the orders are placed.

Company will consist of the following departments: customer care, order management, warehouse management, marketing team, IT team. Each department will work to satisfy the main moto of the company to make the customer satisfied with service received and build network. Customer details consisting bit of personal information will be stored which consist of age, gender, shipping address, customer preferences, search history. These sets of data will help the database filter and suggest their interest of products and attract the customer to buy the products and having the items ship to them the fastest way possible to decrease the timeline of having the product move from the warehouse to the customer location.

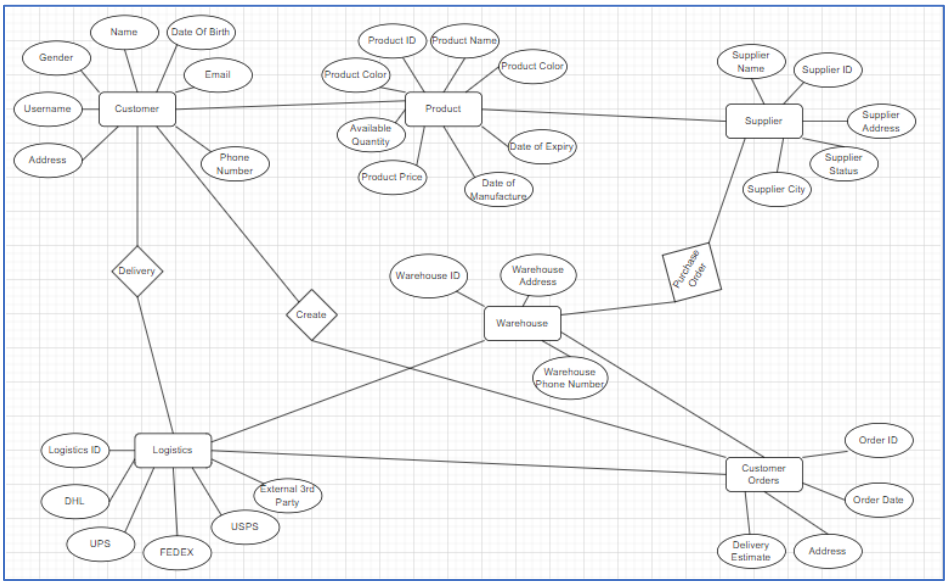
## Project Requirements/Assumptions:

1) Database and servers are to be multi-node and multi-output devices.
2) A customer can create an account and browse but no orders can be created.
3) A customer can influence number of new customers nor have no influence at all.
4) A product can be introduced into our database newly but it’s not necessarily a new product in the market.
5) A customer can order multiple items and not all items can be within the same warehouse.
6) The logistics team can use own or other shipping methodologies to have the product mobilized and reach the customer within a short period.
7) At times due to natural calamities, mobilization of products maybe not be possible from one warehouse and in that case would need to be mobilized from other warehouses which

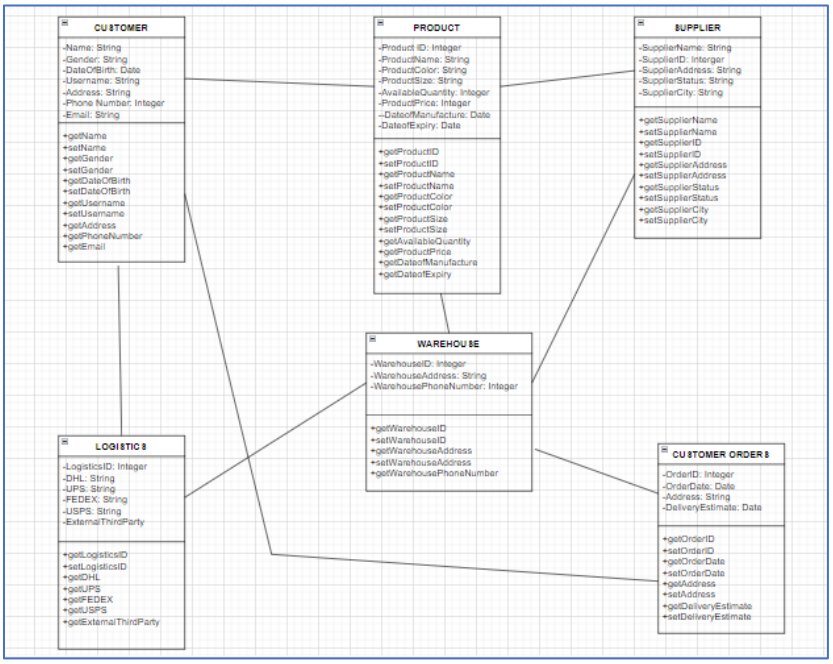
would mean the company would need to digest those additional costs to keep the customers satisfied.

## II. Conceptual Data Modeling

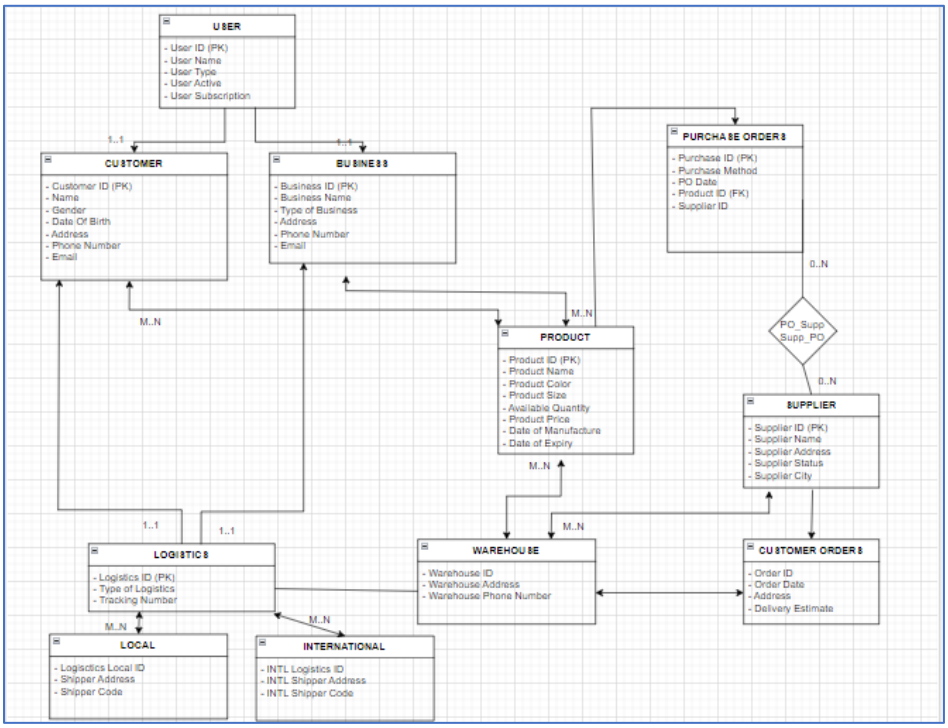
1. EER Diagram




2. UML Diagram



## III. Mapping Conceptual Model to Relational Model




**Primary Key – Bold** **Foreign Key –** *Italicized*

User (**User_ID,** User_name, User_Type, User_Active, User_Subcription)
Customer (**Customer_ID,** *User_ID*, Name, Gender, Dateofbirth, Address, Phone Number, Email)
Business (**Business_ID**, *User_ID*, Business_Name, Type_Of_Business, Address, Phone_Number, Email)
Product (**Product_ID**, Product_name, product_color, product_size, available_quantity, product_price, date_of_Manufacture, date_of_expiry)
Purchase_Order (**Purchase_ID,** Purchase_Method, PO_Date, *Product_ID*,, *Supplier_ID*)
PO_Supp (PO_Supp, Supp_PO)
Supplier (**Supplier_ID,** Supplier_Name, Supplier_Address, Supplier_Status, Supplier_City)
Customer_Order (**Order_ID,** Order_date, address, Delivery_Estimate)
Warehouse (**Warehouse_ID**, Warehouse_address, warehouse_phonenumber)
Logistics (**Logistics_ID**, Type_of_logistics, Tracking_number)
Local (**Logistics_Local_ID,** *Logistics_ID*, Shipper_Address, Shipper_Code)
International (**INTL_Logistics_ID**, *INTL_Shipper*_Address, INTL_Shipper_Code)

## IV. Implementation of Relation Model Via MySQL

**MySQL Implementation:**

The database and schema was created in MySQL workbench and inserted sample data. Following the queries were tested and performed:

**Query 1 (Simple):**

```
select usr.user_id, usr.user_name,
cus.customer_name
from users usr, customers cus
where usr.user_id = cus.user_id
and user_name = 'User1';
```

Result Grid | Filter Rows: Search | Export:

| user_id | user_name | customer_name |
|---|---|---|
| 1234 | User1 | Amazon |

**Query 2 (Nested):**

```
select usr.user_id, usr.user_name
from users usr
where user_id IN (select user_id
                  from  customers
                 where customer_name = 'Walmart');
```

Result Grid | Filter Rows: Sear

| user_id | user_name |
|---|---|
| 12354 | User3 |
| NULL | NULL |

**Query 3 (Correlated):**

```
select business_name, address, city,
state, zip_code
from business
where user_id IN (select user_id
          from users
       where user_id IN (select user_id
       from customers
       where customer_name = 'Amazon'));
```

Result Grid | Filter Rows: Search | Export:

| business_name | address | city | state | zip_code |
|---|---|---|---|---|
| Amazon | 3123 Amazon Way | Seattle | Washington | 12345 |

**Query 4 (GROUP BY):**

```
select product_name, product_size,
COUNT(product_price)
from products
group by product_name, product_size;
```

Result Grid | Filter Rows: Search | Export:

| product_na... | product_si... | COUNT(product_pri... |
|---|---|---|
| Printer | Large | 2 |
| Printer | Small | 3 |
| Printer | E-Large | 2 |

**Query 5 (UNION ALL):**

```
SELECT cus.customer_name, usr.user_name
FROM customers cus
LEFT JOIN   users usr
```

Result Grid | Filter Rows: Searc

| customer_name | user_name |
|---|---|
| Amazon | User1 |
| Samsung | NULL |
| Walmart | User3 |
| Cisco | NULL |
| Amazon | User1 |
| NULL | User4 |
| Walmart | User3 |
| NULL | User5 |
| NULL | User2 |

```
ON cus.user_id = usr.user_id
UNION ALL
SELECT cus.customer_name, usr.user_name
FROM customers cus
RIGHT JOIN   users usr
ON cus.user_id = usr.user_id;
```

**Query 6 (UNION):**

```
SELECT bus.business_name, usr.user_name
FROM business bus
LEFT JOIN   users usr
ON bus.user_id = usr.user_id
UNION
SELECT bus.business_name, usr.user_name
FROM business bus
RIGHT JOIN   users usr
ON bus.user_id = usr.user_id;
```

Result Grid  Filter Rows:

| business_name | user_name |
|---|---|
| Amazon Business | NULL |
| Amazon | User1 |
| Walmart | NULL |
| NULL | User4 |
| NULL | User3 |
| NULL | User5 |
| NULL | User2 |

**Query 7 (MAX):**

```
select product_name, product_color,
product_size
from products
where product_price = (select
                        MAX(product_price)
                  from products
                  where product_size = 'E-Large');
```

Result Grid  Filter Rows: Search

| product_na... | product_col... | product_si... |
|---|---|---|
| Printer | Black | E-Large |

**Query 8 (EXISTS):**

```
select customer_name, user_id,
       dateofbirth, address, city, state, zip_code
from customers cus
where EXISTS (select *
            from users usr,
                business bus
           where usr.user_id = bus.user_id
            and usr.user_id = cus.user_id);
```

Result Grid  Filter Rows: Search  Export:

| customer_name | user_id | dateofbirth | address | city | state | zip_code |
|---|---|---|---|---|---|---|
| Amazon | 1234 | 1964-07-10 | 3123 Amazon Way | Seattle | Washington | 12351 |

**Query 9 (NOT EXISTS):**

```
select customer_name, user_id, dateofbirth,
address, city, state, zip_code
from customers cus
where NOT EXISTS (select *
            from users usr,
```

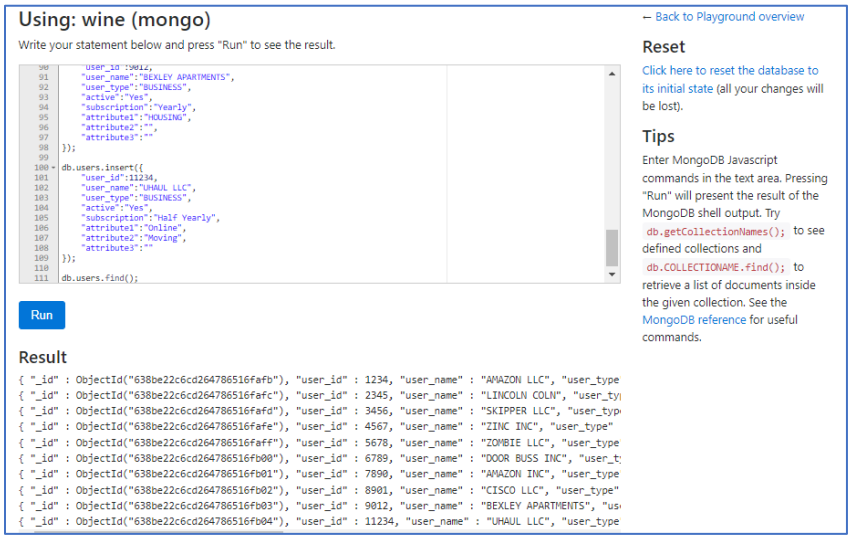
Result Grid  Filter Rows: Search  Export:

| customer_name | user_id | dateofbirth | address | city | state | zip_code |
|---|---|---|---|---|---|---|
| Samsung | 4564 | 1923-10-12 | 3123 Samsung Way | San Jose | California | 91320 |
| Walmart | 12354 | 1922-07-10 | 3123 Walmart Way | Atlanta | Georgia | 12456 |
| Cisco | 4564 | 1914-03-10 | 3123 Cisco Way | San Jose | California | 91320 |

business bus
where usr.user_id = bus.user_id
and usr.user_id = cus.user_id);

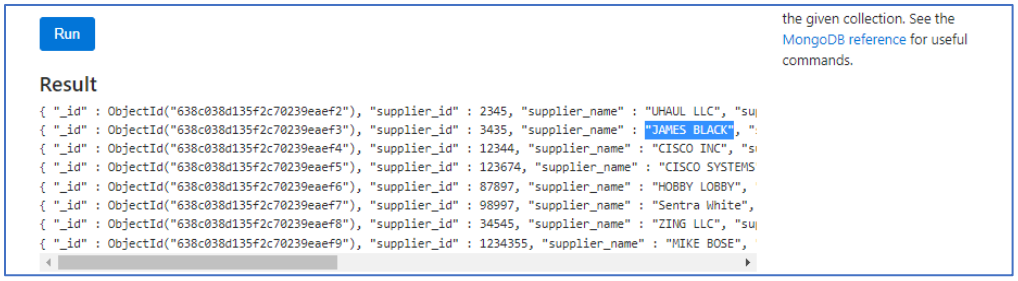
## V. Implementation in NoSQL:

```
db.users.insert({
    "user_id":1234,
    "user_name":"AMAZON LLC",
    "user_type":"BUSINESS",
    "active":"Yes",
    "subscription":"Yearly",
    "attribute1":"Online",
    "attribute2":"",
    "attribute3":""
});
```

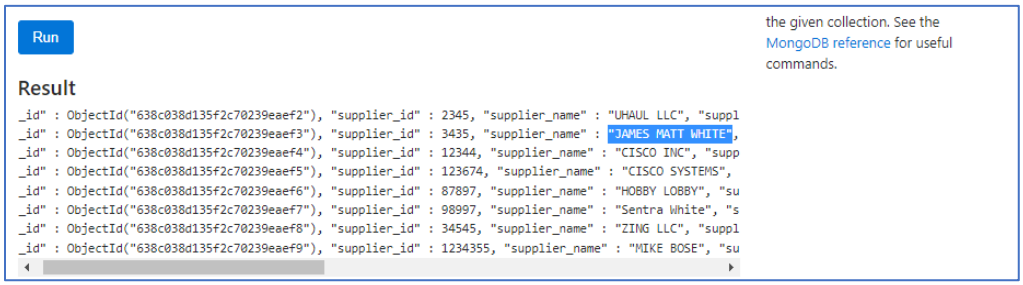



```
db.supplier.update(
   {"supplier_id":3435},
   {$set:
      {"supplier_name": "JAMES MATT WHITE"}
   });
```

## Before:




## After:




## Using a function in NoSQL:

```
db.product.aggregate(
 [{$group:{_id:"$product_size", total:{$sum:"$product_price"}}},
 {$sort:{total:1}}
 ]);
```


## VI. Database Access Via Python

The database is accessed using Python and visualization of analyzed data is shown below. The connection of MySQL to Python is done using mysql.connector, followed by cursor.execute to run and fetchall from query.

Command:
pip install mysql-connector-python

```
In [1]: pip install mysql-connector-python

        Requirement already satisfied: mysql-connector-python in ./opt/anaconda3/lib/python3.9/site-packages (8.0.31)
        Requirement already satisfied: protobuf<=3.20.1,>=3.11.0 in ./opt/anaconda3/lib/python3.9/site-packages (from mysql-c
        onnector-python) (3.20.1)
        Note: you may need to restart the kernel to use updated packages.
```

Command:
import mysql.connector
from mysql.connector import Error
import pandas as pd

```
In [1]: pip install mysql-connector-python

        Requirement already satisfied: mysql-connector-python in ./opt/anaconda3/lib/python3.9/site-packages (8.0.31)
        Requirement already satisfied: protobuf<=3.20.1,>=3.11.0 in ./opt/anaconda3/lib/python3.9/site-packages (from mysql-c
        onnector-python) (3.20.1)
        Note: you may need to restart the kernel to use updated packages.
```

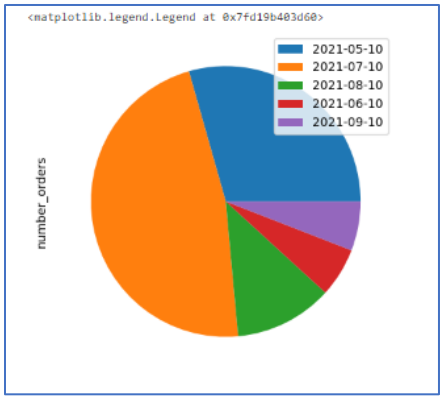
Command:

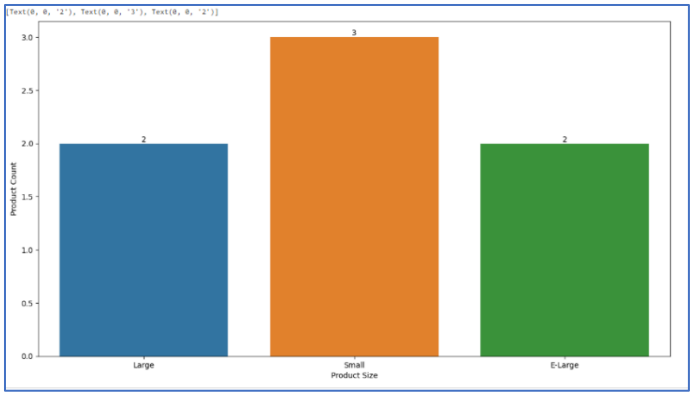
```
connection = mysql.connector.connect(user='root', password='sql@1234',
                                     host='localhost',
                                     database='amaz')

if connection.is_connected():
    db_info=connection.get_server_info()
    print('Connected to MySQL Server Version ',db_info)
    cursor = connection.cursor()
    cursor.execute("select database();")
    record = cursor.fetchone()
    print("Your connected to database: ", record)
```

```
In [41]: connection = mysql.connector.connect(user='root', password='sql@1234',
                                              host='localhost',
                                              database='amaz')

         if connection.is_connected():
               db_info=connection.get_server_info()
               print('Connected to MySQL Server Version ',db_info)
               cursor = connection.cursor()
               cursor.execute("select database();")
               record = cursor.fetchone()
               print("Your connected to database: ", record)

         Connected to MySQL Server Version  8.0.31
         Your connected to database:  ('amaz',)
```

**Graph1: Number of Orders According to Dates**

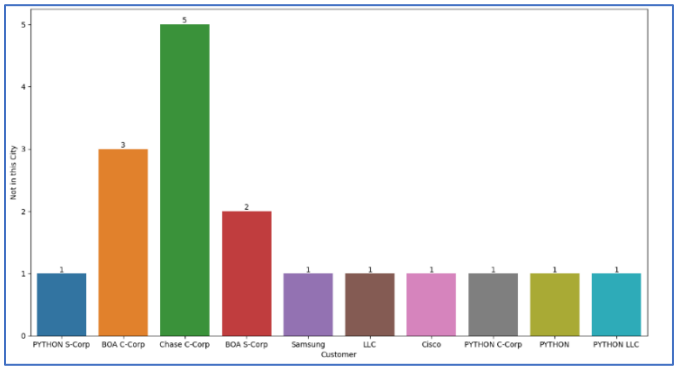
**Graph2: Product Size with Count**

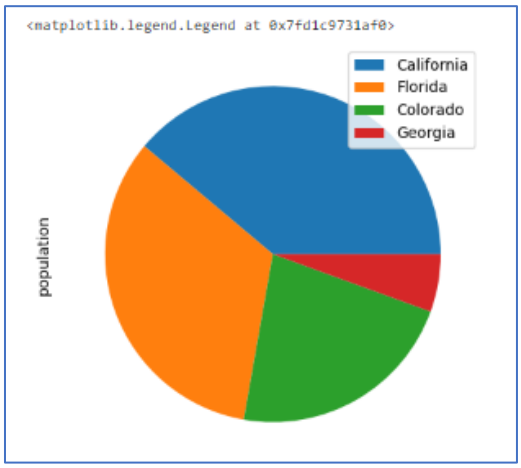



**Graph3: Customer Counts Not in City**

**Group4: Population in Different Cities**




## VII: Summary

The proposal will have a database and multiple servers to withstand multiple requests from various locations and process requests without any hassle or delays. The database will have the capability to store customer interests, age, gender, and search preference so next time the customer browses new products suitable for their interests are presented. The same set of data can be used to suggest a new incoming customer who could possibly have the same set of interests and preferences. The database will store, and record data based on the timeframe as to what type of items are being ordered in which regions so that the respective warehouses can maintain sufficient stock and mobilize the products with less effort, increase efficiency, and cost effective to the customer and internally to the company. Each time a new customer is created in the database, the system will be able to link preference and auto suggest increasing the interest of the customer. Time to time, this data is pulled to analyze the trend to our internal marketing team to possibly propose new features or new departmental additions to increase and grow our company.