

Student's Full Name: John Kwakye Anokye
Course Title: INFO 579
Term name and year: Spring 2023
Submission Week: Week 16 Final Project Report
Instructor's Name: **Rahman Nayem**
Date of Submission: 5/7/2023

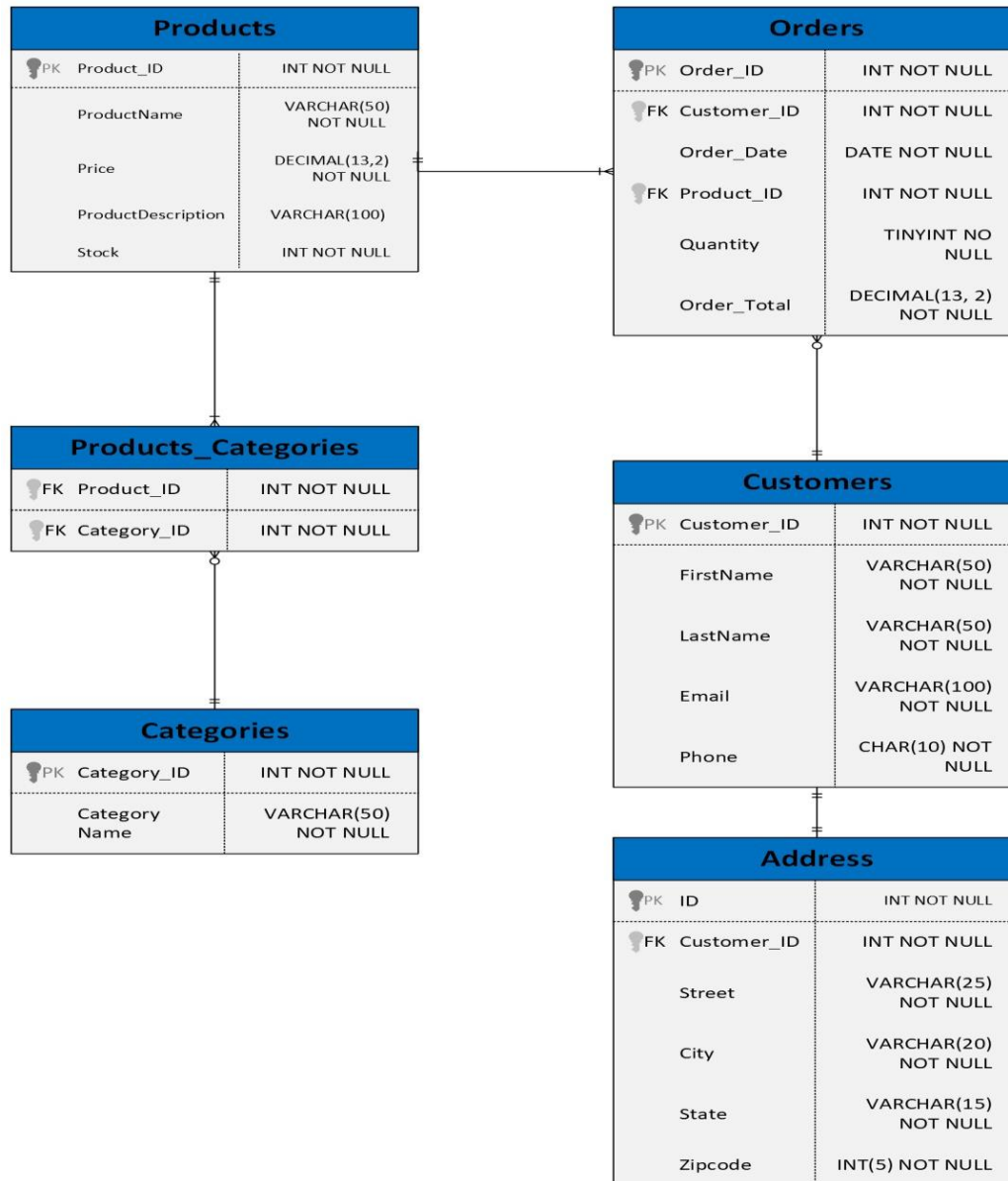
Executive Summary

The advent of the internet has revolutionized the way humanity interacts and conducts business, bringing people closer together and creating new opportunities for entrepreneurs. Particularly since the onset of the global pandemic in 2020, there has been a significant shift towards online purchasing, with individuals increasingly relying on various digital platforms to fulfill their needs. Capitalizing on this trend, I had the privilege of contributing to a project focused on establishing a comprehensive database for a client's thriving online grocery store business. At the core of this venture is the objective to provide customers with a seamless and expedient means of procuring high-quality household goods without the necessity of leaving the comfort of their homes. By building an efficient online platform, our aim was to facilitate a convenient shopping experience for customers, all the while ensuring their trust in receiving exceptional customer service. In order to achieve this, we diligently gathered and organized a wealth of data regarding both customers and inventory.

The data we collected plays a pivotal role in personalizing the shopping experience for each individual customer, tailoring recommendations and promotions to their preferences. Additionally, we implemented a rewards system to acknowledge and incentivize customer loyalty. By leveraging the power of the data we gathered, we were also able to monitor inventory levels, ensuring popular items were promptly restocked to meet demand. Moreover, this comprehensive database serves as a valuable resource for identifying trends, spotting opportunities for improvement in various aspects of operations, and conducting thorough business intelligence studies using tools such as PowerBI. Through a combination of online and offline data collection channels, we amassed a robust dataset that now resides in a MySQL database, serving as the backbone of the business's data management system. With this sophisticated infrastructure in place, our client is poised to thrive in the dynamic landscape of online grocery retail, armed with actionable insights to enhance customer satisfaction, streamline operations, and consistently deliver exceptional service.

5. Develop the physical model based on the Logical Model

ONLINE GROCERY STORE PHYSICAL MODEL



6. Create tables using a database system. Insert data into the database tables.

The screenshot displays a database management interface with a left-hand sidebar and a main right-hand pane. The sidebar, titled 'SCHEMAS', shows a tree view of the 'onlinestore' database, including tables like 'address', 'category', 'customers', 'products', and 'products_categories'. The 'products' table is selected, and its structure is shown in a bottom panel. The main pane displays a SQL script with line numbers 1 through 34. The script includes comments for creating the 'products', 'products_categories', 'customers', and 'address' tables, along with their respective column definitions and foreign key constraints. The 'products' table has columns: Product_ID (int PK), ProductName (varchar), Price (decimal), ProductDescription (varchar), and Stock (int). The 'products_categories' table has columns: Product_ID (int), Category_ID (int), and foreign keys to 'category' and 'products'. The 'customers' table has columns: Customer_ID (int PK), FirstName (varchar), LastName (varchar), Email (varchar), and Phone (char). The 'address' table has columns: ID (int PK), Customer_ID (int), Street (varchar), City (varchar), State (varchar), and Zipcode (int), with a foreign key to 'customers'.

Table: products

Columns:

Column Name	Data Type
Product_ID	int PK
ProductName	varchar(50)
Price	decimal(13,2)
ProductDescription	varchar(100)
Stock	int(7)

```
1 • USE onlinestore;
2   -- creating products table
3 • CREATE TABLE products (
4     Product_ID INT(5) NOT NULL PRIMARY KEY,
5     ProductName VARCHAR(50) NOT NULL,
6     Price DECIMAL(13,2) NOT NULL,
7     ProductDescription VARCHAR(100),
8     Stock INT(7) NOT NULL
9   );
10  -- creating products_categories table
11 • CREATE TABLE products_categories (
12     Product_ID INT(5) NOT NULL,
13     Category_ID INT(5) NOT NULL,
14     FOREIGN KEY (Category_ID) REFERENCES category(Category_ID),
15     FOREIGN KEY (Product_ID) REFERENCES products(Product_ID)
16   );
17  -- creating customers table
18 • CREATE TABLE customers(
19     Customer_ID INT NOT NULL PRIMARY KEY,
20     FirstName VARCHAR(50) NOT NULL,
21     LastName VARCHAR (50) NOT NULL,
22     Email VARCHAR(100) NOT NULL,
23     Phone CHAR(12) NOT NULL
24   );
25  -- creating address table
26 • CREATE TABLE address(
27     ID INT NOT NULL PRIMARY KEY,
28     Customer_ID INT NOT NULL,
29     Street VARCHAR(35) NOT NULL,
30     City VARCHAR(20) NOT NULL,
31     State VARCHAR(15) NOT NULL,
32     Zipcode INT(5) NOT NULL,
33     FOREIGN KEY (Customer_ID) REFERENCES customers(Customer_ID)
34   );
```

6.1. Example of insertion of data into the orders table

1

•

use onlinestore;

2

3

•

INSERT INTO orders (Order_ID, Customer_ID, Order_Date, Product_ID, Quantity, Order_Total)

4

VALUES

5

(1000999 , 14, '2023-05-07', 20, 8, 39.92);

Output

Action Output

#	Time	Action	Message
✓ 1	12:41:56	use onlinestore	0 row(s) affected
✓ 2	12:41:56	select * from orders limit 5	5 row(s) returned
✓ 3	12:46:40	use onlinestore	0 row(s) affected
✗ 4	12:46:40	INSERT INTO orders (Order_ID, Customer_ID, Order_Date, Product_ID, Quantity, Order_Total) VALUES (1000999 , 14, 2023-05-07, 20, 8, 39.92)	Error Code: 1292. Incorrect date value: '2011' for column 'Order_Date' at row 1
✓ 5	12:47:03	use onlinestore	0 row(s) affected
✓ 6	12:47:03	INSERT INTO orders (Order_ID, Customer_ID, Order_Date, Product_ID, Quantity, Order_Total) VALUES (1000999 , 14, '2023-05-07', 20, 8, 39.92)	1 row(s) affected

7. Create a variety of SQL queries to retrieve data from one or many tables.

a. Retrieving all data from orders table

iCHEMAS

Filter objects

- onlinestore
 - Tables
 - address
 - category
 - customers
 - orders
 - products
 - products_categories
 - Views
 - Stored Procedures
 - Functions
- sakila
- sys
- worker
- world

1 • `SELECT * FROM onlinestore.orders;`

Limit to 500 rows

Result Grid

	Order_ID	Customer_ID	Order_Date	Product_ID	Quantity	Order_Total
▶	1000000	14	2023-04-01	7	5	7.45
	1000001	11	2023-04-01	16	10	39.90
	1000002	23	2023-04-01	4	3	62.97
	1000003	27	2023-04-01	14	4	45.96
	1000004	26	2023-04-01	13	6	179.94
	1000005	13	2023-04-01	9	3	19.47
	1000006	15	2023-04-01	13	1	29.99
	1000007	29	2023-04-01	3	6	65.94
	1000008	20	2023-04-01	19	5	4.95
	1000009	28	2023-04-01	1	8	23.92
	1000010	31	2023-04-01	9	3	19.47
	1000011	23	2023-04-01	9	4	25.96
	1000012	23	2023-04-01	8	8	59.92
	1000013	13	2023-04-01	12	1	6.99
	1000014	15	2023-04-01	19	9	8.91

Administration Schemas

b. Query to determine the category each product belongs to

The screenshot shows a database management interface with a left-hand sidebar for 'SCHEMAS' and a main workspace for SQL queries and results.

SCHEMAS Sidebar:

- Filter objects
- onlinestore
 - Tables
 - address
 - category
 - customers
 - orders
 - products
 - products_categories
 - Views
 - Stored Procedures
 - Functions
- sakila
- sys
- worker
- world

SQL Query Editor:

```
1 # Query to determine the category each product belongs to
2 • use onlinestore;
3
4 • SELECT
5     p.Product_ID, p.ProductName, c.CategoryName
6 FROM products as p
7 INNER JOIN products_categories as pc
8     ON p.Product_ID = pc.Product_ID
9 INNER JOIN category as c
10    ON pc.Category_ID = c.Category_ID
11 ORDER BY Product_ID ASC;
```

Result Grid:

	Product_ID	ProductName	CategoryName
▶	1	Apple	Snacks
	1	Apple	Fresh Produce
	2	Milk	Fresh Produce
	2	Milk	Dairy
	2	Milk	Canned and Packaged
	3	Chicken Breast	Frozen Foods
	3	Chicken Breast	Meat and Poultry
	4	Shrimp	Frozen Foods
	4	Shrimp	Seafood
	5	Whole Wheat Bread	Bakery
	6	Orange Juice	Beverages
	6	Orange Juice	Fresh Produce
	6	Orange Juice	Canned and Packaged
	7	Canned Tomatoes	Canned and Packaged
	8	Frozen Mixed Vege...	Canned and Packaged
	9	Potato Chips	Canned and Packaged
	9	Potato Chips	Snacks
	10	Crackers	Snacks
	10	Crackers	Canned and Packaged
	11	Pork	Frozen Foods
	11	Pork	Meat and Poultry
	12	Soda	Canned and Packaged
	12	Soda	Beverages
	13	Beef	Frozen Foods
	13	Beef	Meat and Poultry
	14	Crab	Frozen Foods
	14	Crab	Seafood

Table Information:

Table: **products_categories**

Columns:

- Product_ID int
- Category_ID int

c. Query to determine the products that belong to Seafood, Frozen Foods and Meat and Poultry categories.

The screenshot shows a database management tool interface. On the left, the 'SCHEMAS' pane displays a tree view of the 'onlinestore' database, with tables like 'address', 'category', 'customers', 'orders', 'products', and 'products_categories' listed. The 'products_categories' table is selected. The main pane shows a SQL query:

```
1 # Query to determine the products that belongs to Seafood, Frozen Foods and Meat and Poultry
2 • use onlinestore;
3
4 • SELECT
5   p.Product_ID, p.ProductName, c.CategoryName
6 FROM products as p
7 INNER JOIN products_categories as pc
8 ON p.Product_ID = pc.Product_ID
9 INNER JOIN category as c
10 ON pc.Category_ID = c.Category_ID
11 WHERE c.CategoryName IN ('Seafood', 'Frozen Foods', 'Meat and Poultry')
12 ORDER BY Product_ID ASC;
```

Below the query, the 'Result Grid' shows the results of the query. The grid has columns for 'Product_ID', 'ProductName', and 'CategoryName'. The results are as follows:

Product_ID	ProductName	CategoryName
3	Chicken Breast	Frozen Foods
3	Chicken Breast	Meat and Poultry
4	Shrimp	Frozen Foods
4	Shrimp	Seafood
11	Pork	Frozen Foods
11	Pork	Meat and Poultry
13	Beef	Frozen Foods
13	Beef	Meat and Poultry
14	Crab	Frozen Foods
14	Crab	Seafood

d. Query to determine what product categories does the largest orders belongs to.

The screenshot shows a database management interface with a left-hand sidebar containing a tree view of database objects. The 'onlinestore' database is selected, and the 'Tables' folder is expanded, showing tables like 'address', 'category', 'customers', 'orders', 'products', and 'products_categories'. The main area displays a SQL query in a text editor, and below it, a 'Result Grid' showing the query's output.

SQL Query:

```
1 # What product categories does our largest orders belong to?
2 use onlinestore;
3
4 SELECT
5 p.Product_ID, p.ProductName, c.CategoryName, o.Order_ID, o.Quantity, o.Order_Total
6 FROM orders as o
7 INNER JOIN products as p
8 ON o.Product_ID = p.Product_ID
9 INNER JOIN products_categories as pc
10 ON p.Product_ID = pc.Product_ID
11 INNER JOIN category as c
12 ON pc.Category_ID = c.Category_ID
13 WHERE c.CategoryName = 'Frozen Foods' AND c.CategoryName NOT IN ('Meat and Poultry')
14 ORDER BY Order_Total DESC
15 LIMIT 10;
```

Result Grid:

	Product_ID	ProductName	CategoryName	Order_ID	Quantity	Order_Total
▶	13	Beef	Frozen Foods	1000168	10	299.90
	13	Beef	Frozen Foods	1000160	9	269.91
	13	Beef	Frozen Foods	1000884	9	269.91
	11	Pork	Frozen Foods	1000331	10	259.90
	11	Pork	Frozen Foods	1000812	10	259.90
	11	Pork	Frozen Foods	1000755	10	259.90
	13	Beef	Frozen Foods	1000140	8	239.92
	13	Beef	Frozen Foods	1000177	8	239.92
	13	Beef	Frozen Foods	1000886	8	239.92
	11	Pork	Frozen Foods	1000173	9	233.91

e. Query to determine the top 3 popular products that are frequently ordered by residents living in San Antonio

The screenshot shows a database query editor with a schema tree on the left and a SQL query in the main area. The schema tree shows a database named 'onlinestore' with tables: address, category, customers, orders, products, and products_categories. The SQL query is as follows:

```
1 # What are the top 3 popular products that are frequently ordered by residents living in San Antonio
2 • use onlinestore;
3 • WITH CTE1 AS (
4     SELECT
5         p.ProductName, o.Order_Total, a.City
6     FROM address as a
7     INNER JOIN customers as c
8     ON a.Customer_ID = c.Customer_ID
9     INNER JOIN orders as o
10    ON c.Customer_ID = o.Customer_ID
11    INNER JOIN products as p
12    ON o.Product_ID = p.Product_ID
13    WHERE a.City = 'San Antonio'
14    ORDER BY o.Order_Total DESC)
15 SELECT DISTINCT ProductName, Order_Total, City from CTE1 ORDER BY Order_Total DESC LIMIT 3;
```

The result grid shows the following data:

ProductName	Order_Total	City
Pork	233.91	San Antonio
Shrimp	209.90	San Antonio
Pork	207.92	San Antonio

f. What is the top 5 selling products?

The screenshot shows a database query editor with a schema tree on the left and a SQL query in the main area. The schema tree shows a database named 'onlinestore' with tables: address, category, customers, orders, products, and products_categories. The SQL query is as follows:

```
1 # What is our top 5 best selling product
2 • USE onlinestore;
3 • SELECT p.Product_ID, p.ProductName, SUM(o.Order_Total) as OrderTotal
4     FROM products as p
5     INNER JOIN orders as o
6     ON o.Product_ID = p.Product_ID
7     GROUP BY p.Product_ID, p.ProductName
8     ORDER BY OrderTotal DESC
9     LIMIT 5;
```

The result grid shows the following data:

Product_ID	ProductName	OrderTotal
4	Shrimp	7199.57
11	Pork	6237.60
13	Beef	5368.21
14	Crab	2849.52
3	Chicken Breast	2384.83

g. What is the least 5 selling product?

The screenshot shows a database management interface with a left-hand sidebar for 'SCHEMAS' and a main area for SQL queries and results.

SCHEMAS Sidebar:

- Filter objects
- onlinestore
 - Tables
 - address
 - category
 - customers
 - orders
 - products
 - products_categories
 - Views
 - Stored Procedures
 - Functions
- sakila
- sys
- worker
- world

SQL Query Editor:

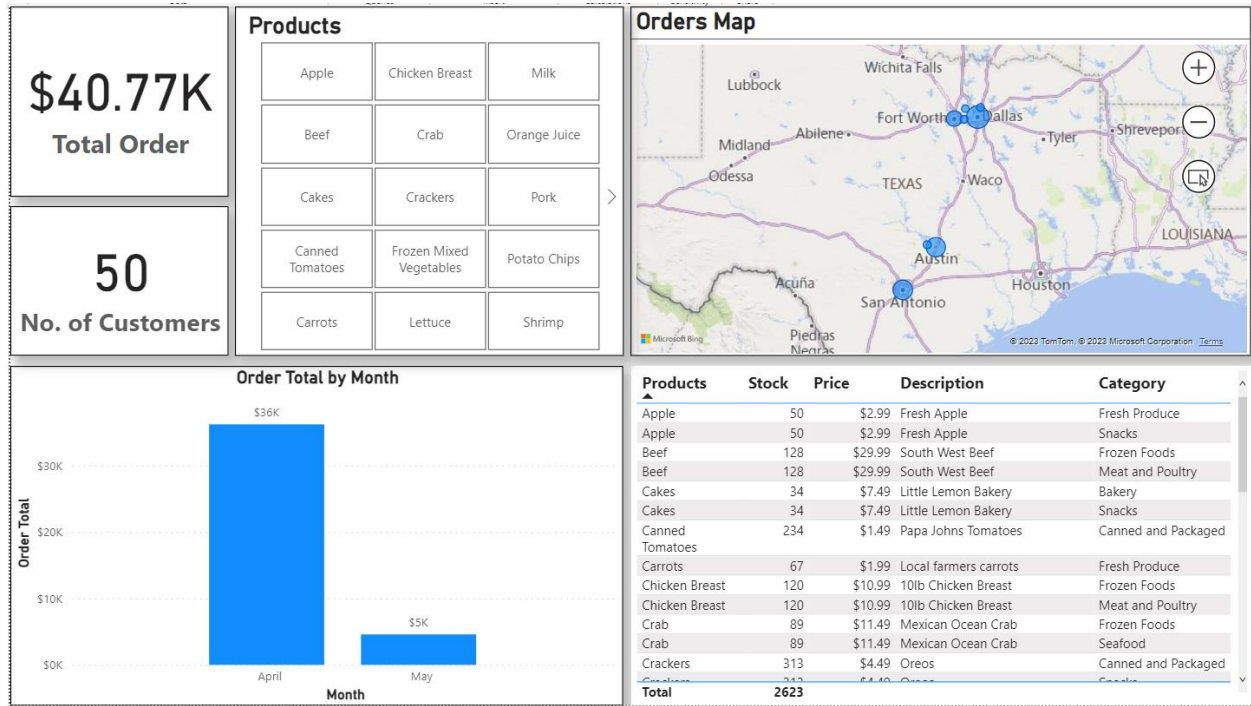
```
1 # What is our least 5 selling product
2 • USE onlinestore;
3 • SELECT p.Product_ID, p.ProductName, SUM(o.Order_Total) as OrderTotal
4 FROM products as p
5 INNER JOIN orders as o
6 ON o.Product_ID = p.Product_ID
7 GROUP BY p.Product_ID, p.ProductName
8 ORDER BY OrderTotal asc
9 LIMIT 5;
```

Result Grid:

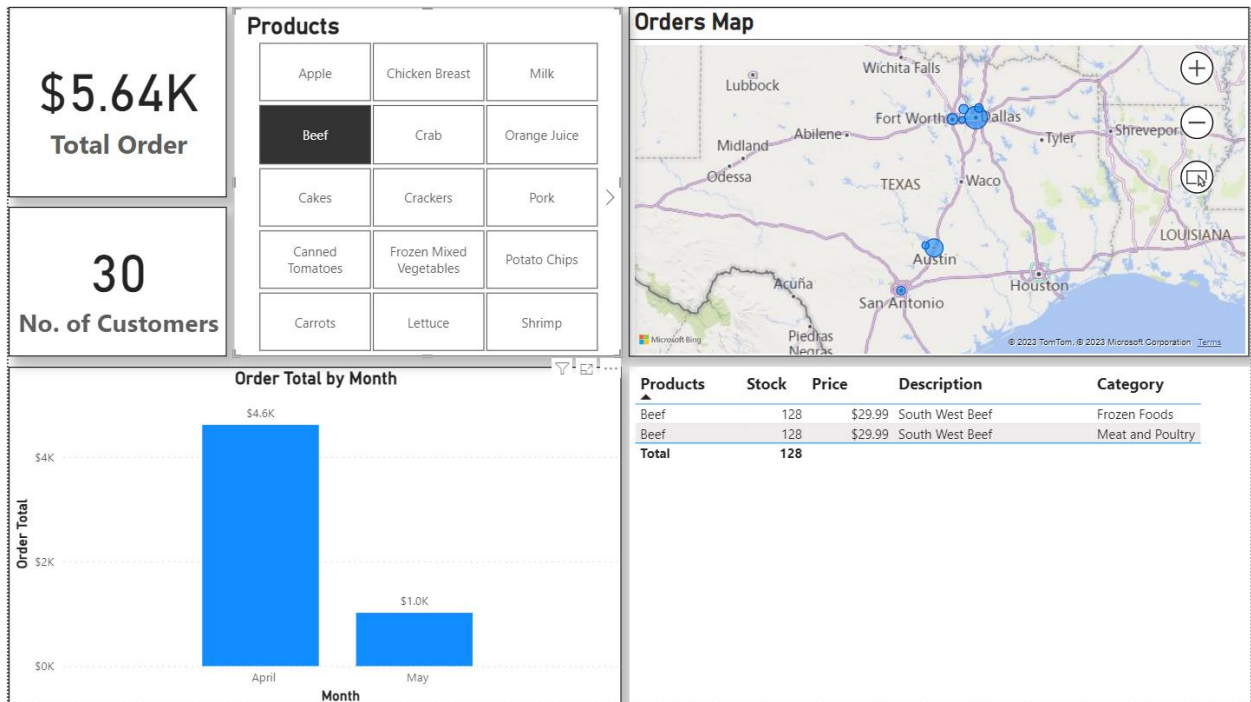
	Product_ID	ProductName	OrderTotal
▶	19	Lettuce	217.80
	7	Canned Tomatoes	371.01
	18	Carrots	469.64
	17	Yogurt	717.60
	20	Water	888.22

EXTRAS

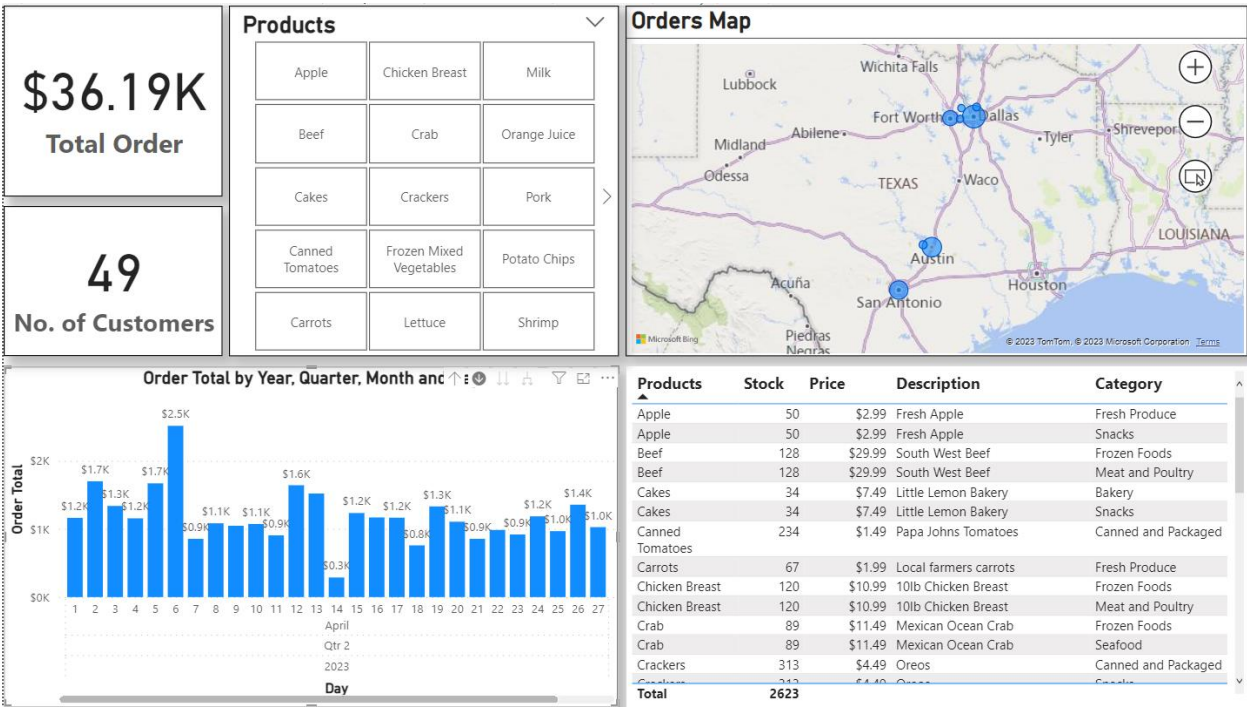
1. Power BI Dashboard with different visualizations showing Products slicer, Total Order generated so far, number of customers and column chart of order per month.



2. Dashboard showing number of sales made by Beef product and number of customers who have purchased beef so far



3. Visualization of daily orders made in the month of April



4. Visualization of orders made by customers living in San Antonio

