DATA MANAGEMENT: PMM GROCERY SUPERMARKET CASE STUDY

**TASK 1**

Entities and Attributes:

1. Branch:
   * BranchID (Primary Key)
   * BranchName
   * Address
   * City
2. Staff:
   * StaffID (Primary Key)
   * FirstName
   * LastName
   * Position
   * BranchID (Foreign Key)
3. Product:
   * ProductID (Primary Key)
   * ProductName
   * Category
   * Price
   * Ingredients
   * AllergyAdvice
   * Lifestyle
   * SizeVolume
   * NetWeight
   * DirectionForUse
   * NutritionInfo
   * CountryOfOrigin
   * StorageInstruction
   * Manufacturer
4. Customer:
   * CustomerID (Primary Key)
   * FirstName
   * LastName
   * Email
   * Address
   * City
5. Order:
   * OrderID (Primary Key)
   * OrderDate
   * CustomerID (Foreign Key)
   * BranchID (Foreign Key)
6. OrderDetails:
   * OrderDetailID (Primary Key)
   * OrderID (Foreign Key)
   * ProductID (Foreign Key)
   * Quantity
   * TotalPrice

Relationships:

* Each branch has multiple staff members
* Each branch handles multiple orders
* Each customer can place multiple orders
* Each order can contain multiple products (OrderDetails)

Implementation:

**CREATE** **TABLE** Branch **(**

BranchID SERIAL **PRIMARY** **KEY,**

BranchName VARCHAR**(**100**),**

Address VARCHAR**(**255**),**

City VARCHAR**(**100**)**

**);**

**CREATE** **TABLE** Staff **(**

StaffID SERIAL **PRIMARY** **KEY,**

FirstName VARCHAR**(**100**),**

LastName VARCHAR**(**100**),**

Position VARCHAR**(**100**),**

BranchID INT**,**

**FOREIGN** **KEY** **(**BranchID**)** **REFERENCES** Branch**(**BranchID**)**

**);**

**CREATE** **TABLE** Product **(**

ProductID SERIAL **PRIMARY** **KEY,**

ProductName VARCHAR**(**100**),**

Category VARCHAR**(**100**),**

Price DECIMAL**(**10**,** 2**),**

Ingredients TEXT**,**

AllergyAdvice TEXT**,**

Lifestyle VARCHAR**(**100**),**

SizeVolume VARCHAR**(**100**),**

NetWeight VARCHAR**(**100**),**

DirectionForUse TEXT**,**

NutritionInfo TEXT**,**

CountryOfOrigin VARCHAR**(**100**),**

StorageInstruction TEXT**,**

Manufacturer VARCHAR**(**100**)**

**);**

**CREATE** **TABLE** Customer **(**

CustomerID SERIAL **PRIMARY** **KEY,**

FirstName VARCHAR**(**100**),**

LastName VARCHAR**(**100**),**

Email VARCHAR**(**100**),**

Address VARCHAR**(**255**),**

City VARCHAR**(**100**)**

**);**

**CREATE** **TABLE** Orders **(**

OrderID SERIAL **PRIMARY** **KEY,**

OrderDate DATE**,**

CustomerID INT**,**

BranchID INT**,**

**FOREIGN** **KEY** **(**CustomerID**)** **REFERENCES** Customer**(**CustomerID**),**

**FOREIGN** **KEY** **(**BranchID**)** **REFERENCES** Branch**(**BranchID**)**

**);**

**CREATE** **TABLE** OrderDetails **(**

OrderDetailID SERIAL **PRIMARY** **KEY,**

OrderID INT**,**

ProductID INT**,**

Quantity INT**,**

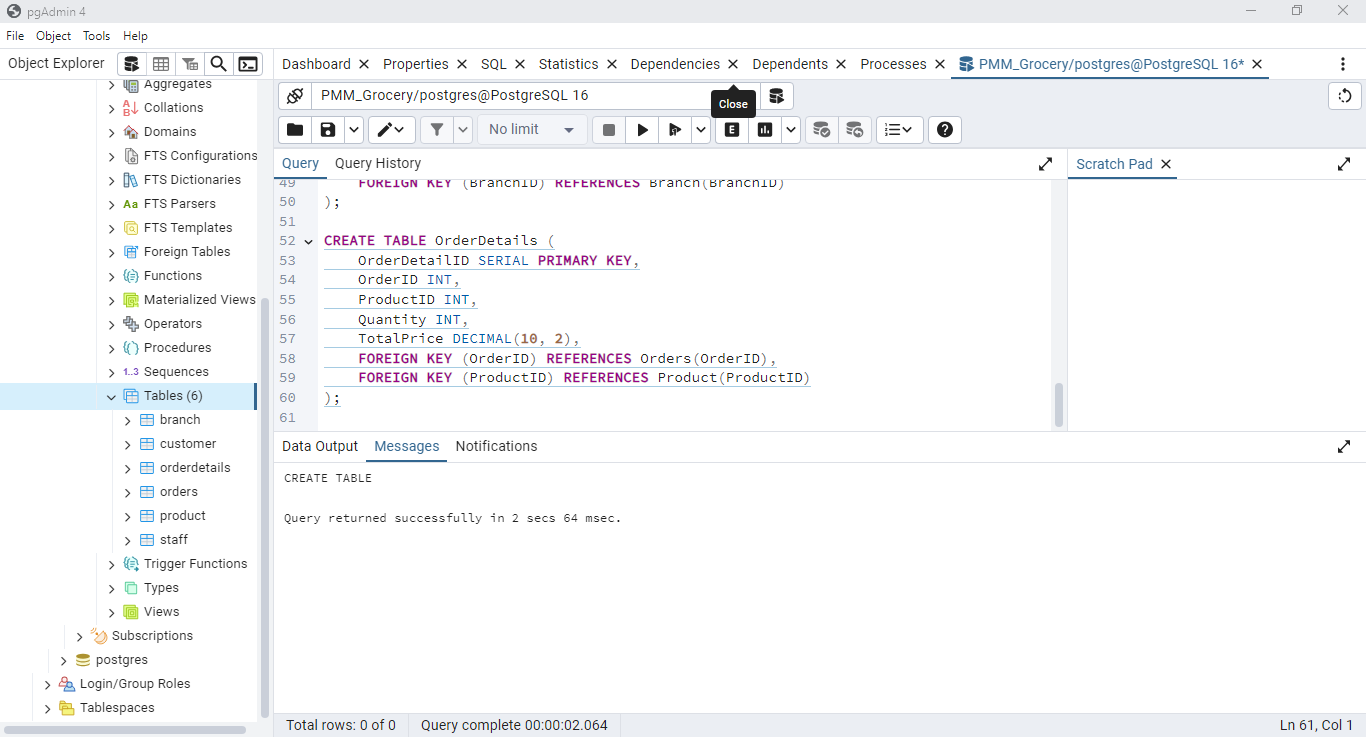
TotalPrice DECIMAL**(**10**,** 2**),**

**FOREIGN** **KEY** **(**OrderID**)** **REFERENCES** Orders**(**OrderID**),**

**FOREIGN** **KEY** **(**ProductID**)** **REFERENCES** Product**(**ProductID**)**

**);**

Screenshot of query running:



Populating the table:

-- Sample Data for Branch

**INSERT** **INTO** Branch **(**BranchName**,** Address**,** City**)** **VALUES**

**(**'Waterlooville'**,** '123 Main St'**,** 'Waterlooville'**),**

**(**'Fareham'**,** '456 Elm St'**,** 'Fareham'**),**

**(**'Gosport'**,** '789 Oak St'**,** 'Gosport'**),**

**(**'Havant'**,** '101 Pine St'**,** 'Havant'**),**

**(**'Chichester'**,** '202 Maple St'**,** 'Chichester'**),**

**(**'Portsmouth'**,** '303 Birch St'**,** 'Portsmouth'**);**

-- Sample Data for Staff

**INSERT** **INTO** Staff **(**FirstName**,** LastName**,** Position**,** BranchID**)** **VALUES**

**(**'John'**,** 'Doe'**,** 'Manager'**,** 1**),**

**(**'Jane'**,** 'Smith'**,** 'Cashier'**,** 1**),**

**(**'Emily'**,** 'Davis'**,** 'Stock Clerk'**,** 2**),**

**(**'Michael'**,** 'Brown'**,** 'Manager'**,** 3**),**

**(**'Sarah'**,** 'Wilson'**,** 'Cashier'**,** 3**);**

-- Sample Data for Product

**INSERT** **INTO** Product **(**ProductName**,** Category**,** Price**,** Ingredients**,** AllergyAdvice**,** Lifestyle**,** SizeVolume**,** NetWeight**,** DirectionForUse**,** NutritionInfo**,** CountryOfOrigin**,** StorageInstruction**,** Manufacturer**)** **VALUES**

**(**'Sliced Malted Bloomer Bread'**,** 'Bread/Bakery'**,** 1.35**,** 'Wheat Flour, Water, Malted Wheat, Yeast, Rye Flour, Rapeseed Oil, Salt, Wheat Bran, Wheat Semolina, Malted Barley Flour, Preservative: Calcium Propionate, Flour Treatment Agent: Ascorbic Acid'**,** 'Contains: Wheat, Rye, Barley'**,** 'Vegetarian'**,** 'Large'**,** '800g'**,** 'Use as directed'**,** 'Energy 1064kJ, 251kcal, Fat 1.7g, Carbohydrate 48g, Fibre 4.1g, Protein 8.5g, Salt 0.87g'**,** 'UK'**,** 'Store in a cool, dry place'**,** 'BreadCo'**);**

-- Sample Data for Customer

**INSERT** **INTO** Customer **(**FirstName**,** LastName**,** Email**,** Address**,** City**)** **VALUES**

**(**'Alice'**,** 'Johnson'**,** 'alice.johnson@example.com'**,** '12 Cedar St'**,** 'Waterlooville'**),**

**(**'Bob'**,** 'Martin'**,** 'bob.martin@example.com'**,** '34 Spruce St'**,** 'Fareham'**);**

-- Sample Data for Orders

**INSERT** **INTO** Orders **(**OrderDate**,** CustomerID**,** BranchID**)** **VALUES**

**(**'2024-07-01'**,** 1**,** 1**),**

**(**'2024-07-02'**,** 2**,** 2**);**

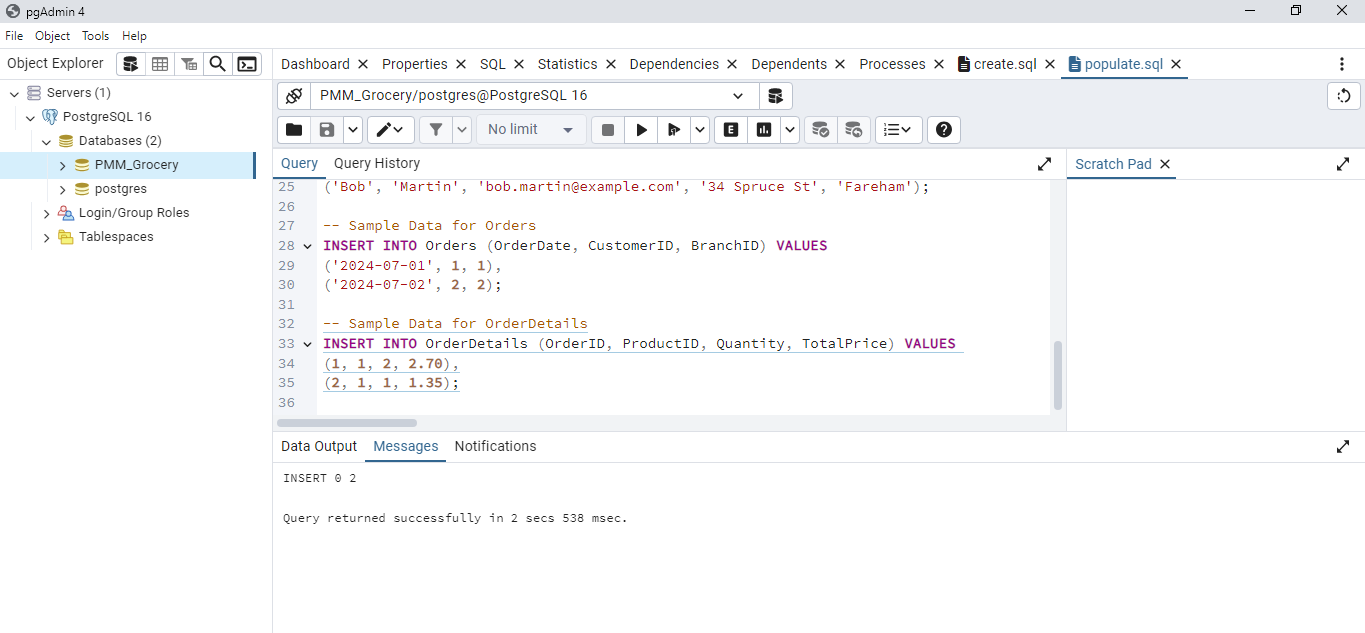
-- Sample Data for OrderDetails

**INSERT** **INTO** OrderDetails **(**OrderID**,** ProductID**,** Quantity**,** TotalPrice**)** **VALUES**

**(**1**,** 1**,** 2**,** 2.70**),**

**(**2**,** 1**,** 1**,** 1.35**);**

Screenshot of output:



Queries:

1. Getting basic statistics on customers per city for a specific time period

**SELECT** City**,** **COUNT(**CustomerID**)** **AS** CustomerCount

**FROM** Customer

**WHERE** CustomerID **IN** **(**

**SELECT** CustomerID

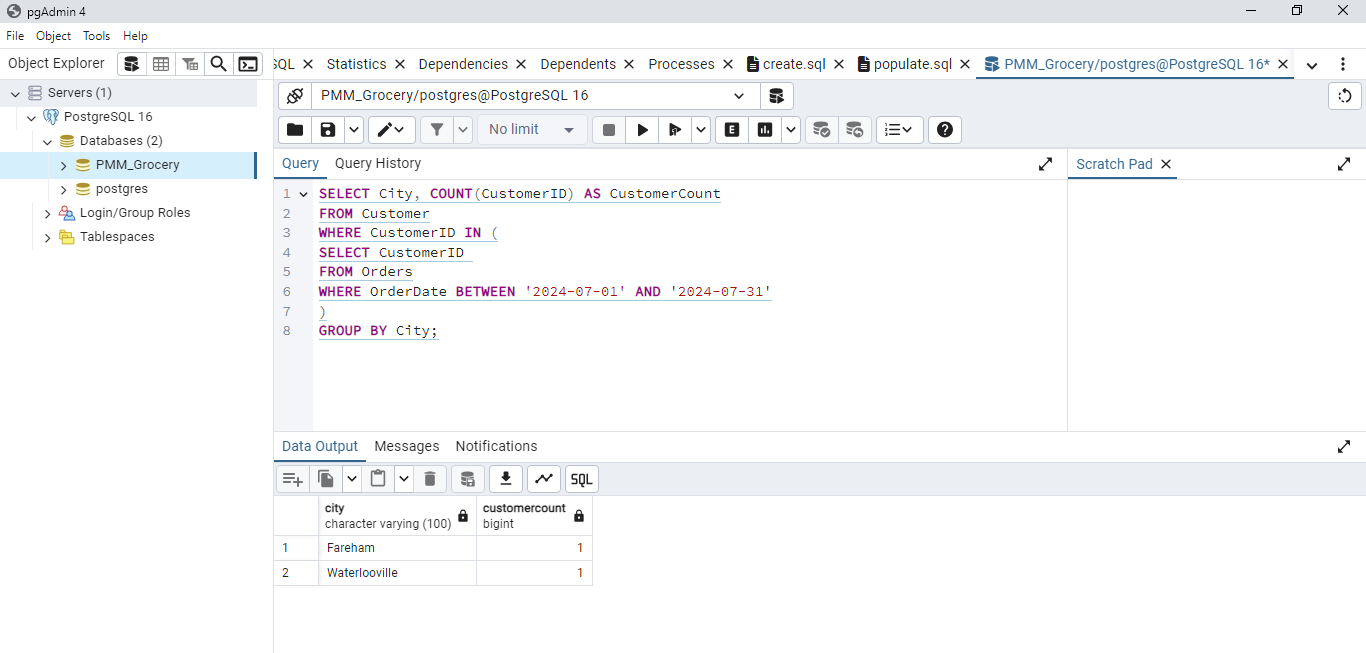
**FROM** Orders

**WHERE** OrderDate **BETWEEN** '2024-07-01' **AND** '2024-07-31'

**)**

**GROUP** **BY** City**;**

Screenshot of query running:

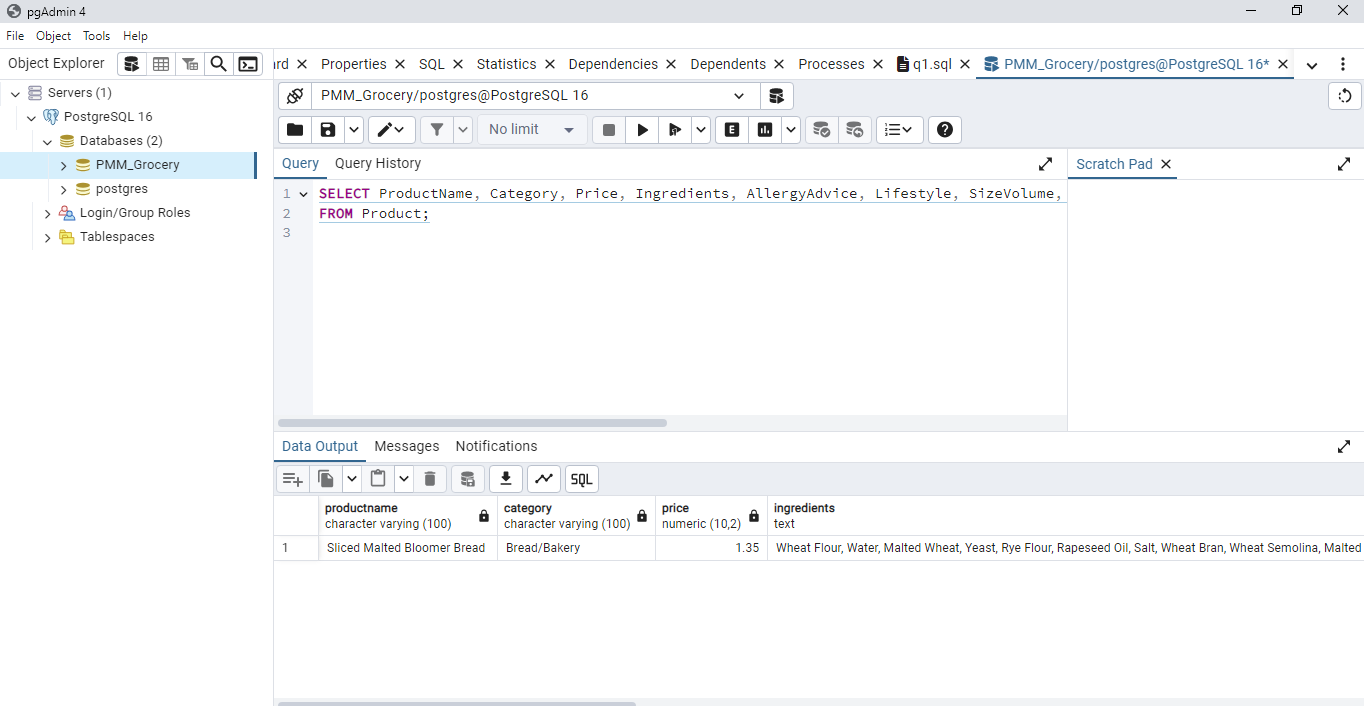


1. Listing all products with details and prices

**SELECT** ProductName**,** Category**,** Price**,** Ingredients**,** AllergyAdvice**,** Lifestyle**,** SizeVolume**,** NetWeight**,** DirectionForUse**,** NutritionInfo**,** CountryOfOrigin**,** StorageInstruction**,** Manufacturer

**FROM** Product**;**

Screenshot of query running:



1. Getting order records and delivery details

**SELECT** O**.**OrderID**,** O**.**OrderDate**,** C**.**FirstName**,** C**.**LastName**,** C**.**Address**,** C**.**City**,** B**.**BranchName**,** P**.**ProductName**,** OD**.**Quantity**,** OD**.**TotalPrice

**FROM** Orders O

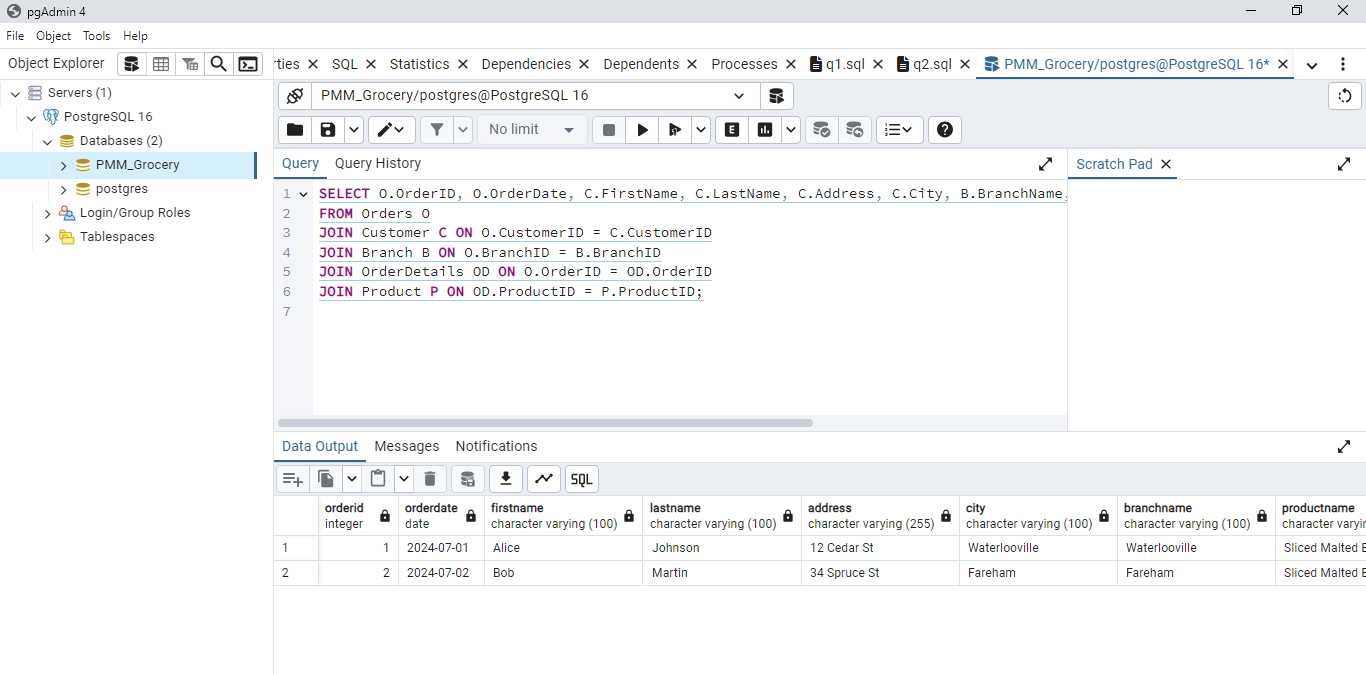
**JOIN** Customer C **ON** O**.**CustomerID **=** C**.**CustomerID

**JOIN** Branch B **ON** O**.**BranchID **=** B**.**BranchID

**JOIN** OrderDetails OD **ON** O**.**OrderID **=** OD**.**OrderID

**JOIN** Product P **ON** OD**.**ProductID **=** P**.**ProductID**;**

Screenshot of query running:



1. Report of product availability and their location

**SELECT** B**.**BranchName**,** P**.**ProductName**,** **SUM(**OD**.**Quantity**)** **AS** TotalQuantity

**FROM** Orders O

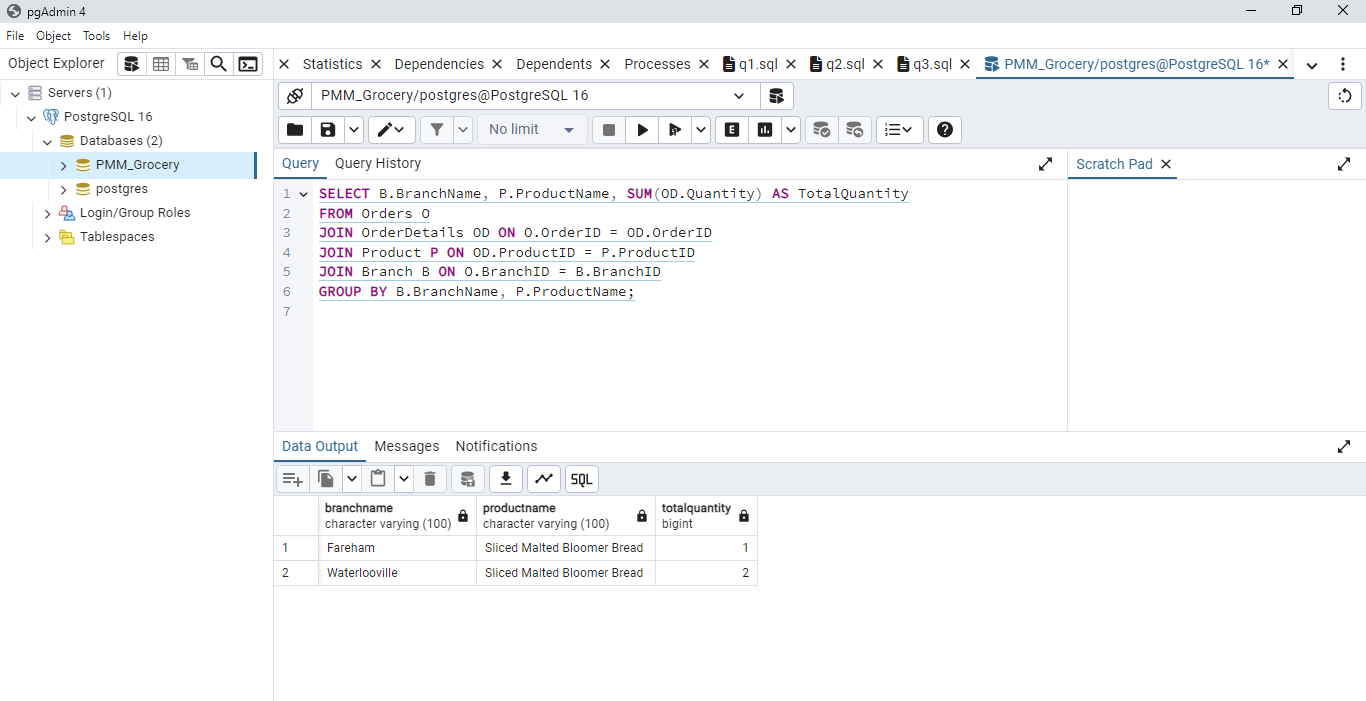
**JOIN** OrderDetails OD **ON** O**.**OrderID **=** OD**.**OrderID

**JOIN** Product P **ON** OD**.**ProductID **=** P**.**ProductID

**JOIN** Branch B **ON** O**.**BranchID **=** B**.**BranchID

**GROUP** **BY** B**.**BranchName**,** P**.**ProductName**;**

Screenshot of query running:



1. Monthly income generated per city/location

**SELECT** B**.**City**,** **SUM(**OD**.**TotalPrice**)** **AS** MonthlyIncome

**FROM** Orders O

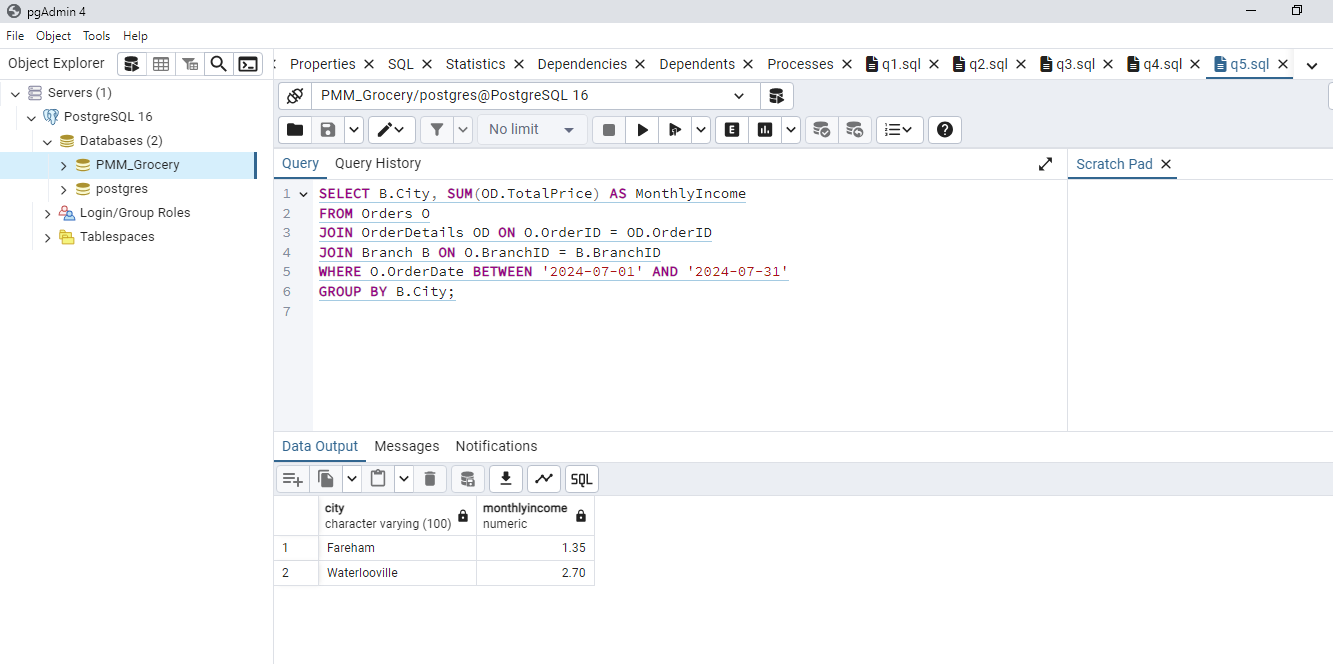
**JOIN** OrderDetails OD **ON** O**.**OrderID **=** OD**.**OrderID

**JOIN** Branch B **ON** O**.**BranchID **=** B**.**BranchID

**WHERE** O**.**OrderDate **BETWEEN** '2024-07-01' **AND** '2024-07-31'

**GROUP** **BY** B**.**City**;**

Screenshot of query running:



**TASK 2**

Implementing MongoDB:

**import** pymongo

**from** pymongo **import** MongoClient

**import** random

# Connect to MongoDB

client **=** MongoClient**(**"mongodb://localhost:27017/"**)**

# Create database

db **=** client**[**'PMM\_Grocery\_Supermarket'**]**

# Create collection

products\_collection **=** db**[**'products'**]**

# Sample product categories

categories **=** **[**"Beverages"**,** "Bread/Bakery"**,** "Canned/Jarred Goods"**,** "Dairy"**,** "Dry/Baking Goods"**,** "Frozen Foods"**,** "Meats"**,** "Produce"**,** "Cleaners"**,** "Personal Care"**,** "Paper Goods"**]**

# Function to generate random product data

**def** generate\_product\_data**():**

product\_data **=** **[]**

**for** i **in** **range(**100**):**

product **=** **{**

"ProductID"**:** i **+** 1**,**

"ProductName"**:** f"Product\_{i **+** 1}"**,**

"Category"**:** random**.**choice**(**categories**),**

"Price"**:** **round(**random**.**uniform**(**0.5**,** 50.0**),** 2**),**

"Ingredients"**:** "Sample ingredients list"**,**

"AllergyAdvice"**:** "Sample allergy advice"**,**

"Lifestyle"**:** "Sample lifestyle"**,**

"SizeVolume"**:** "Sample size/volume"**,**

"NetWeight"**:** f"{random**.**randint**(**100**,** 2000**)**}g"**,**

"DirectionForUse"**:** "Sample direction for use"**,**

"NutritionInfo"**:** "Sample nutrition info"**,**

"CountryOfOrigin"**:** "Sample country of origin"**,**

"StorageInstruction"**:** "Sample storage instruction"**,**

"Manufacturer"**:** "Sample manufacturer"

**}**

product\_data**.**append**(**product**)**

**return** product\_data

# Generate product data

products **=** generate\_product\_data**()**

# Insert data into collection

products\_collection**.**insert\_many**(**products**)**

**print(**"Database populated with sample products."**)**

# Sample Queries

**print(**"\nProducts in the 'Beverages' category:"**)**

**for** product **in** products\_collection**.**find**({**"Category"**:** "Beverages"**}):**

**print(**product**)**

**print(**"\nProducts with price less than £10.00:"**)**

**for** product **in** products\_collection**.**find**({**"Price"**:** **{**"$lt"**:** 10.0**}}):**

**print(**product**)**

**print(**"\nCount of products in each category:"**)**

pipeline **=** **[**

**{**"$group"**:** **{**"\_id"**:** "$Category"**,** "count"**:** **{**"$sum"**:** 1**}}}**

**]**

**for** result **in** products\_collection**.**aggregate**(**pipeline**):**

**print(**result**)**

**print(**"\nProduct with name 'Product\_1':"**)**

product\_name **=** "Product\_1"

product **=** products\_collection**.**find\_one**({**"ProductName"**:** product\_name**})**

**print(**product**)**

**print(**"\nProducts with specific ingredient 'Sample ingredients list':"**)**

ingredient **=** "Sample ingredients list"

**for** product **in** products\_collection**.**find**({**"Ingredients"**:** ingredient**}):**

**print(**product**)**

**TASK 3**

1. Looking for specific seasonal products

season **=** "Winter"

specific\_ingredient **=** "Malted Barley Flour"

description\_keyword **=** "Bread"

query **=** **{**

"Season"**:** season**,**

"Ingredients"**:** **{**"$regex"**:** specific\_ingredient**,** "$options"**:** "i"**},**

"ProductName"**:** **{**"$regex"**:** description\_keyword**,** "$options"**:** "i"**}**

**}**

results **=** products\_collection**.**find**(**query**)**

**for** product **in** results**:**

**print(**product**)**

1. Availability of products and locations

availability\_query **=** **[**

**{**

"$lookup"**:** **{**

"from"**:** "branches"**,**

"localField"**:** "BranchID"**,**

"foreignField"**:** "BranchID"**,**

"as"**:** "branch\_info"

**}**

**},**

**{**

"$unwind"**:** "$branch\_info"

**},**

**{**

"$project"**:** **{**

"ProductName"**:** 1**,**

"branch\_info.BranchName"**:** 1**,**

"branch\_info.Address"**:** 1**,**

"branch\_info.City"**:** 1**,**

"Quantity"**:** 1

**}**

**}**

**]**

results **=** products\_collection**.**aggregate**(**availability\_query**)**

**for** result **in** results**:**

**print(**result**)**

1. Order deliveries history

delivery\_history\_query **=** **[**

**{**

"$lookup"**:** **{**

"from"**:** "order\_details"**,**

"localField"**:** "OrderID"**,**

"foreignField"**:** "OrderID"**,**

"as"**:** "order\_details"

**}**

**},**

**{**

"$unwind"**:** "$order\_details"

**},**

**{**

"$lookup"**:** **{**

"from"**:** "products"**,**

"localField"**:** "order\_details.ProductID"**,**

"foreignField"**:** "ProductID"**,**

"as"**:** "product\_info"

**}**

**},**

**{**

"$unwind"**:** "$product\_info"

**},**

**{**

"$project"**:** **{**

"OrderID"**:** 1**,**

"OrderDate"**:** 1**,**

"CustomerID"**:** 1**,**

"product\_info.ProductName"**:** 1**,**

"order\_details.Quantity"**:** 1**,**

"order\_details.TotalPrice"**:** 1

**}**

**}**

**]**

results **=** db**.**orders**.**aggregate**(**delivery\_history\_query**)**

**for** result **in** results**:**

**print(**result**)**

1. Record of products purchased based on locations of stores

purchasing\_record\_query **=** **[**

**{**

"$lookup"**:** **{**

"from"**:** "orders"**,**

"localField"**:** "OrderID"**,**

"foreignField"**:** "OrderID"**,**

"as"**:** "order\_info"

**}**

**},**

**{**

"$unwind"**:** "$order\_info"

**},**

**{**

"$lookup"**:** **{**

"from"**:** "branches"**,**

"localField"**:** "order\_info.BranchID"**,**

"foreignField"**:** "BranchID"**,**

"as"**:** "branch\_info"

**}**

**},**

**{**

"$unwind"**:** "$branch\_info"

**},**

**{**

"$project"**:** **{**

"ProductName"**:** 1**,**

"branch\_info.BranchName"**:** 1**,**

"branch\_info.City"**:** 1**,**

"order\_info.OrderDate"**:** 1**,**

"order\_info.CustomerID"**:** 1**,**

"Quantity"**:** 1**,**

"TotalPrice"**:** 1

**}**

**},**

**{**

"$sort"**:** **{**"branch\_info.City"**:** 1**,** "order\_info.OrderDate"**:** 1**}**

**}**

**]**

results **=** products\_collection**.**aggregate**(**purchasing\_record\_query**)**

**for** result **in** results**:**

**print(**result**)**

1. Finding specific lifestyle and nutrition products

**TASK 4**

The criteria for switching from a manual system to a digitalized approach that could handle large amounts of data were carefully analysed before the database for PMM Grocery Supermarket was developed. The Entity-Relationship Diagram (ERD), which helped depict the relationships between various entities like items, branches, staff, customers, and orders, was the first thing I designed. PostgreSQL was used to structure the relational database based on this ERD. To assure data integrity, I built tables for each entity, specified primary and foreign keys, and established relationships.

Writing scripts to produce realistic sample data for goods, personnel, and consumers was part of populating the database. To improve query efficiency, indexes were made on columns that were often requested. Data validation and integrity were upheld by applying constraints like {NOT NULL` and `UNIQUE`.

I decided on MongoDB as my NoSQL database because of its adaptability to handle unstructured data. I made a Python script to produce and insert one hundred sample product documents after creating a collection for products. Each paper reflected the full qualities listed in the case study and included comprehensive product information, such as price, ingredients, nutrition information, and storage directions.

CHALLENGES AND SOLUTION

Over the course of the endeavour, I ran into a few obstacles. For the relationships and dependencies between entities to be accurately represented, substantial consideration was needed throughout the ERD's design. Retaining data integrity and maximizing query efficiency were essential elements requiring careful preparation and implementation. Furthermore, transferring from a rigid relational database to an adaptable NoSQL database presented unique difficulties with regard to data representation and query creation.

I used extensive research and database design best practices to overcome these obstacles. Frequent verification and assessment of the database structure and queries guaranteed that the system fulfilled the necessary conditions and operated well.

SELECTION OF QUERIES

The following questions were chosen to answer the pragmatic requirements of PMM Grocery Supermarket's operations:

1. Season-Specific Product Search: This search function assists consumers in locating goods that are appropriate for particular seasons and comprise particular ingredients, meeting dietary needs and seasonal incentives.
2. Product Locations and Availability: It's critical for inventory management to guarantee product availability across various branches. A snapshot of the locations and stock levels is given by this query.
3. Delivery Order History: For the purposes of logistics management and customer service, delivery history tracking is crucial. This search yields a whole order history, which is useful for answering client questions and evaluating delivery efficiency.
4. purchase Records by Location: Stock replenishment and strategic planning are facilitated by an understanding of purchase patterns among various locations. This search offers information on branch-specific sales performance.
5. Criteria for Nutrition and Lifestyle: This search improves the user experience by assisting in the discovery of products that satisfy particular dietary and lifestyle needs, appealing to health-conscious consumers.

These inquiries were created with PMM Grocery Supermarket's operational needs in mind, guaranteeing that the system offers insightful data and facilitates effective administration. Every inquiry was painstakingly designed to extract valuable information, enabling well-informed decision-making and effective business operations.

To sum up, creating this database required careful design, extensive testing, and an emphasis on real-world business requirements. The selected queries are in line with PMM Grocery Supermarket's operational needs, guaranteeing that the system offers insightful data and facilitates effective administration.