National College of Ireland

Advanced Databases
Section A
Group Y
Food Inventory Data Mart

Eoin Fitzsimons- x23151374
Conor Judge- x22165398

FOOD INVENTORY

Eoin Fitzsimons (Profit) – Conor Judge (Perish)

Introduction

When faced with the task of designing a data mart for a business within food inventory we were unsure which of the many options was best to pick; we ended up settling on a restaurant with the onus of the mart being the stockroom of individual ingredients. Two thoughts immediately emerged when deciding upon the business requirements; perishability and profit. We did not find a ready-made database that suited our needs enough, so endeavoured to make our own. We started with what information we would like to obtain from our queries and then worked backwards to flesh out the table and make it appear more realistic akin to how a genuine restaurant would keep track of its produce.

We devised an artisanal culinary experience that stays true to the roots of our vegetables and wants to be carbon-positive. All our food is prepared without electricity, and we only refrigerate and freeze a minuscule amount of ingredients to have the lowest emissions. We have a direct link to the local vegetable farmers and even have our plots within the restaurant to provide that mesmerising organic dining moment™.

We assume this is popular and not a complete waste of time.

Business Requirements.

Perishability

A key issue for any business that deals with food will be managing waste. Spoiled items cannot be used or sold and have another further cost in the form of waste disposal. Restaurants must invest in facilities to store the food in appropriate environments to prevent waste.

Being able to identify items that are most and least likely to spoil can be used to change buying and storage patterns.

Storage areas where a significant amount of waste occurs can be identified to adjust storage conditions, implementing better inventory management practices, or reevaluating the allocation of ingredients to different storage areas.

Profits

All businesses need to make money to run. A restaurant is no exception, our restaurant needs to purchase ingredients and then the meals will make use of ingredients and sell to customers.

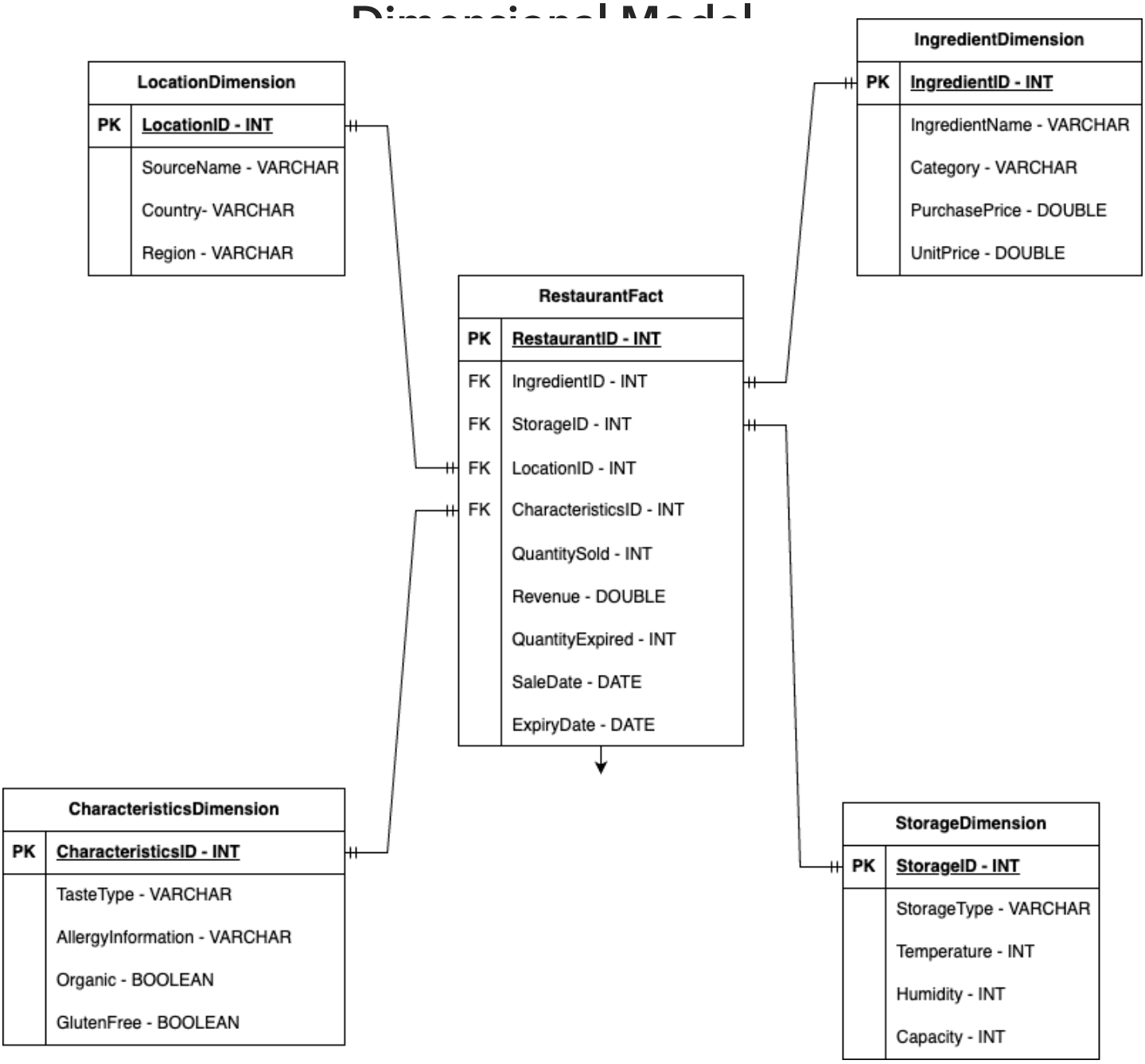
The inventory manager for a restaurant may want to observe multiple dimensions to examine why ingredients are profitable. They should see whether they ought to raise prices or cut costs and obtain the ingredients differently.

To do this they could examine the ingredient dimension to see what produce is bought and sold. Then take note of how much is spent in procurement, and how much an ingredient brings in when used in one of the restaurant's wonderous dishes.

They could also see which country the produce has arrived from to inspect whether the distance the product travels has proportionality with the profit margin. Is the premium on local artisanal goods worth it when compared to mass-produced goods from elsewhere?

Then the inventory manager would examine the characteristics of the ingredients to see which taste has the most profound impact on the profit margin. Is it those sweet ingredients that you only need a minuscule amount of to tie a dessert together? Or should the restaurant focus on those foundational vegetables of society that grow in abundance and are bought at dirt-cheap prices?

Dimensional Model



SQL – Fact & Dimension Tables.

```sql
-- Delete Database
DROP DATABASE IF EXISTS Restaurant;
-- Create Database
CREATE DATABASE IF NOT EXISTS Restaurant;

-- Switch to the Restaurant Database
USE Restaurant;

-- Create Dimension Tables
CREATE TABLE IngredientDimension (
    IngredientID INT PRIMARY KEY,
    IngredientName VARCHAR(50),
    Category VARCHAR(30),
    UnitPrice DECIMAL(10, 2),
    PurchasePrice DECIMAL(10, 2)
);

CREATE TABLE StorageDimension (
    StorageID INT PRIMARY KEY,
    StorageType VARCHAR(30),
    Temperature VARCHAR(20),
    Humidity VARCHAR(20),
    Capacity DECIMAL(10, 2)
);

CREATE TABLE LocationDimension (
    LocationID INT PRIMARY KEY,
    SourceName VARCHAR(50),
    Country VARCHAR(50),
    Region VARCHAR(50)
);

CREATE TABLE CharacteristicsDimension (
    CharacteristicsID INT PRIMARY KEY,
    TasteProfile VARCHAR(100),
    AllergyInformation VARCHAR(100),
    Organic BOOLEAN,
    GlutenFree BOOLEAN
);

-- Create Fact Table
CREATE TABLE RestaurantFact (
    RestaurantID INT PRIMARY KEY,
    IngredientID INT,
    QuantityBought INT,
    QuantitySold INT,
    Revenue DECIMAL(10, 2),
    QuantityExpired INT,
    StorageID INT,
    CharacteristicsID INT,
    LocationID INT,
    SaleDate DATE,
    WastageDate DATE,
    FOREIGN KEY (IngredientID) REFERENCES IngredientDimension(IngredientID),
    FOREIGN KEY (StorageID) REFERENCES StorageDimension(StorageID),
    FOREIGN KEY (CharacteristicsID) REFERENCES CharacteristicsDimension(CharacteristicsID),
    FOREIGN KEY (LocationID) REFERENCES LocationDimension(LocationID)
);

-- Indexes
-- Index for IngredientID in RestaurantFact table (for JOIN operations)
CREATE INDEX idx_IngredientID ON RestaurantFact (IngredientID);

-- Index for SaleDate in RestaurantFact table (for WHERE clause)
CREATE INDEX idx_SaleDate ON RestaurantFact (SaleDate);

-- Index for WastageDate in RestaurantFact table (for WHERE clause)
CREATE INDEX idx_WastageDate ON RestaurantFact (WastageDate);

-- Index for QuantityExpired in RestaurantFact table (for Food Waste query)
CREATE INDEX idx_QuantityExpired ON RestaurantFact (QuantityExpired);
```

SQL – Data into Mart.

```sql
-- Insert data into IngredientDimension
INSERT INTO IngredientDimension (IngredientID, IngredientName, Category, UnitPrice, PurchasePrice)
VALUES
(1, 'Potatoes', 'Vegetables', 1.00, 0.80),
(2, 'Lamb', 'Meat', 8.50, 7.00),
(3, 'Carrots', 'Vegetables', 0.80, 0.60),
(4, 'Butter', 'Dairy', 3.00, 2.50),
(5, 'Salmon', 'Seafood', 10.00, 8.00),
(6, 'Mushrooms', 'Vegetables', 2.50, 2.00),
(7, 'Oats', 'Grains', 1.20, 1.00),
(8, 'Beef', 'Meat', 9.00, 7.50),
(9, 'Apples', 'Fruits', 1.50, 1.20),
(10, 'Cheese', 'Dairy', 4.00, 3.50),
(11, 'Honey', 'Condiments', 5.00, 4.00),
(12, 'Eggs', 'Dairy', 0.90, 0.70),
(13, 'Spinach', 'Vegetables', 2.20, 1.80),
(14, 'Barley', 'Grains', 1.80, 1.50),
(15, 'Lettuce', 'Vegetables', 2.00, 1.50),
(16, 'Sweet Potatoes', 'Vegetables', 1.20, 1.00),
(17, 'Chicken', 'Meat', 7.50, 6.00),
(18, 'Broccoli', 'Vegetables', 1.00, 0.80),
(19, 'Cream', 'Dairy', 2.50, 2.00),
(20, 'Shrimp', 'Seafood', 12.00, 10.00),
(21, 'Onions', 'Vegetables', 1.50, 1.20),
(22, 'Quinoa', 'Grains', 2.00, 1.80),
(23, 'Pork', 'Meat', 8.00, 6.50),
(24, 'Oranges', 'Fruits', 1.80, 1.50),
(25, 'Cheddar', 'Dairy', 4.50, 3.80),
(26, 'Mustard', 'Condiments', 2.50, 2.00),
(27, 'Whole Milk', 'Dairy', 1.00, 0.90),
(28, 'Kale', 'Vegetables', 2.00, 1.80),
(29, 'Rice', 'Grains', 1.50, 1.20),
(30, 'Tomatoes', 'Vegetables', 1.80, 1.50);

-- Insert data into StorageDimension
INSERT INTO StorageDimension (StorageID, StorageType, Temperature, Humidity, Capacity)
VALUES
(1, 'Pantry', 'Room Temp', 'Normal', 500.00),
(2, 'Freezer', '-18°C', 'Low', 300.00),
(3, 'Refrigerator', '4°C', 'Medium', 200.00),
(4, 'Dry Storage', 'Room Temp', 'Low', 700.00),
(5, 'Cool Room', '12°C', 'High', 400.00),
(6, 'Freezer', '-20°C', 'Low', 250.00),
(7, 'Refrigerator', '5°C', 'Medium', 180.00),
(8, 'Pantry', 'Room Temp', 'Normal', 550.00),
(9, 'Dry Storage', 'Room Temp', 'Low', 600.00),
(10, 'Cool Room', '10°C', 'High', 350.00),
```

```
    (10, 'Cool Room', '10°C', 'High', 350.00),
    (11, 'Freezer', '-15°C', 'Low', 280.00),
    (12, 'Refrigerator', '3°C', 'Medium', 220.00),
    (13, 'Dry Storage', 'Room Temp', 'Low', 800.00),
    (14, 'Pantry', 'Room Temp', 'Normal', 450.00),
    (15, 'Cool Room', '11°C', 'High', 320.00),
    (16, 'Pantry', 'Room Temp', 'Normal', 450.00),
    (17, 'Freezer', '-20°C', 'Low', 320.00),
    (18, 'Refrigerator', '5°C', 'Medium', 180.00),
    (19, 'Dry Storage', 'Room Temp', 'Low', 600.00),
    (20, 'Cool Room', '10°C', 'High', 350.00),
    (21, 'Freezer', '-15°C', 'Low', 280.00),
    (22, 'Refrigerator', '3°C', 'Medium', 220.00),
    (23, 'Pantry', 'Room Temp', 'Normal', 500.00),
    (24, 'Dry Storage', 'Room Temp', 'Low', 700.00),
    (25, 'Cool Room', '12°C', 'High', 400.00),
    (26, 'Freezer', '-18°C', 'Low', 250.00),
    (27, 'Refrigerator', '4°C', 'Medium', 200.00),
    (28, 'Dry Storage', 'Room Temp', 'Low', 800.00),
    (29, 'Pantry', 'Room Temp', 'Normal', 550.00),
    (30, 'Cool Room', '11°C', 'High', 320.00);

    -- Insert data into LocationDimension
    INSERT INTO LocationDimension (LocationID, SourceName, Country, Region)
    VALUES
    (1, 'Local Farmer', 'Ireland', 'Leinster'),
    (2, 'Irish Seafood Co-op', 'Ireland', 'Munster'),
    (3, 'Dairy Co-op', 'Ireland', 'Connaught'),
    (4, 'Green Vegetable Farms', 'Ireland', 'Ulster'),
    (5, 'British Meat Supplier', 'United Kingdom', 'Various'),
    (6, 'French Cheese Distributor', 'France', 'Various'),
    (7, 'Spanish Fruit Co-op', 'Spain', 'Various'),
    (8, 'Italian Olive Oil Supplier', 'Italy', 'Various'),
    (9, 'Dutch Dairy Co-op', 'Netherlands', 'Various'),
    (10, 'Scottish Honey Producer', 'United Kingdom', 'Various'),
    (11, 'German Egg Farm', 'Germany', 'Various'),
    (12, 'Belgian Mushroom Co-op', 'Belgium', 'Various'),
    (13, 'Norwegian Salmon Fisheries', 'Norway', 'Various'),
    (14, 'Swedish Barley Farm', 'Sweden', 'Various'),
    (15, 'Danish Lettuce Grower', 'Denmark', 'Various'),
    (16, 'Scottish Potato Farms', 'United Kingdom', 'Various'),
    (17, 'New Zealand Lamb Co-op', 'New Zealand', 'Various'),
    (18, 'Caribbean Roots Farms', 'Jamaica', 'Various'),
    (19, 'Irish Dairy Collective', 'Ireland', 'Munster'),
    (20, 'Japanese Seafood Importers', 'Japan', 'Various'),
    (21, 'Brazilian Mushroom Co-op', 'Brazil', 'Various'),
    (22, 'Canadian Oats Distributors', 'Canada', 'Various'),
```

```
    (22, 'Canadian Oats Distributors', 'Canada', 'Various'),
    (23, 'Australian Beef Producers', 'Australia', 'Various'),
    (24, 'South African Apple Orchards', 'South Africa', 'Various'),
    (25, 'Swiss Cheese Importers', 'Switzerland', 'Various'),
    (26, 'Indian Honey Farms', 'India', 'Various'),
    (27, 'Greek Egg Producers', 'Greece', 'Various'),
    (28, 'Mexican Spinach Farms', 'Mexico', 'Various'),
    (29, 'Argentinian Barley Fields', 'Argentina', 'Various'),
    (30, 'Chinese Lettuce Growers', 'China', 'Various');

    -- Insert data into CharacteristicsDimension
    INSERT INTO CharacteristicsDimension (CharacteristicsID, TasteProfile, AllergyInformation, Organic, GlutenFree)
    VALUES
    (1, 'Starchy', 'None', TRUE, TRUE),
    (2, 'Savory', 'None', TRUE, FALSE),
    (3, 'Sweet', 'None', TRUE, TRUE),
    (4, 'Rich', 'Lactose Intolerance', TRUE, FALSE),
    (5, 'Umami', 'Fish Allergy', TRUE, TRUE),
    (6, 'Earthy', 'None', TRUE, TRUE),
    (7, 'Nutty', 'Gluten Intolerance', TRUE, TRUE),
    (8, 'Savory', 'None', TRUE, FALSE),
    (9, 'Sweet', 'None', TRUE, TRUE),
    (10, 'Creamy', 'Lactose Intolerance', TRUE, FALSE),
    (11, 'Sweet', 'None', TRUE, TRUE),
    (12, 'Protein-Rich', 'None', TRUE, TRUE),
    (13, 'Leafy', 'Oxalate Sensitivity', TRUE, TRUE),
    (14, 'Nutty', 'Gluten Intolerance', TRUE, TRUE),
    (15, 'Crisp', 'None', TRUE, TRUE),
    (16, 'Sweet', 'None', TRUE, TRUE),
    (17, 'Lean', 'None', TRUE, FALSE),
    (18, 'Fresh', 'None', TRUE, TRUE),
    (19, 'Creamy', 'Lactose Intolerance', TRUE, FALSE),
    (20, 'Delicate', 'Shellfish Allergy', TRUE, TRUE),
    (21, 'Aromatic', 'None', TRUE, TRUE),
    (22, 'Nutrient-Rich', 'Gluten Intolerance', TRUE, TRUE),
    (23, 'Juicy', 'None', TRUE, FALSE),
    (24, 'Citrusy', 'None', TRUE, TRUE),
    (25, 'Sharp', 'Lactose Intolerance', TRUE, FALSE),
    (26, 'Spicy', 'Mustard Allergy', TRUE, TRUE),
    (27, 'Creamy', 'Lactose Intolerance', TRUE, FALSE),
    (28, 'Leafy', 'Oxalate Sensitivity', TRUE, TRUE),
    (29, 'Versatile', 'Gluten Intolerance', TRUE, TRUE),
    (30, 'Crunchy', 'None', TRUE, TRUE);
    -- Insert data into RestaurantFact
    INSERT INTO RestaurantFact (RestaurantID, IngredientID, QuantityBought, QuantitySold, Revenue, QuantityExpired, StorageID, CharacteristicsID, LocationID, SaleDate, WastageDate)
    VALUES
    (1, 1, 200, 180, 180.00, 20, 1, 1, 1, '2023-01-15', '2023-01-20'),
    (2, 2, 30, 25, 212.50, 5, 2, 5, 5, '2023-02-10', '2023-02-15'),
    (3, 3, 150, 140, 112.00, 10, 3, 7, 8, '2023-03-05', '2023-03-10'),
    (4, 4, 50, 45, 135.00, 5, 4, 8, 10, '2023-04-20', '2023-04-25'),
    (5, 5, 80, 75, 150.00, 5, 5, 11, 2, '2023-05-12', '2023-05-17'),
    (6, 6, 100, 90, 225.00, 10, 6, 13, 12, '2023-06-08', '2023-06-13'),
    (7, 7, 120, 110, 144.00, 10, 7, 14, 7, '2023-07-25', '2023-07-30'),
    (8, 8, 60, 55, 270.00, 5, 8, 1, 14, '2023-08-17', '2023-08-22'),
    (9, 9, 40, 35, 60.00, 5, 9, 10, 3, '2023-09-03', '2023-09-08'),
    (10, 10, 110, 100, 140.00, 10, 10, 4, 9, '2023-10-19', '2023-10-24'),
    (11, 11, 65, 60, 200.00, 5, 11, 8, 11, '2023-11-14', '2023-11-19'),
    (12, 12, 95, 90, 105.00, 5, 12, 2, 7, '2023-12-02', '2023-12-07'),
    (13, 13, 55, 50, 120.00, 5, 13, 3, 13, '2024-01-08', '2024-01-13'),
    (14, 14, 70, 65, 165.00, 5, 14, 9, 15, '2024-02-14', '2024-02-19'),
    (15, 15, 85, 80, 180.00, 5, 15, 12, 9, '2024-03-21', '2024-03-26'),
    (16, 16, 120, 110, 132.00, 10, 16, 16, 16, '2024-04-02', '2024-04-07'),
    (17, 17, 40, 35, 262.50, 5, 17, 17, 17, '2024-05-19', '2024-05-24'),
    (18, 18, 90, 80, 72.00, 10, 18, 18, 18, '2024-06-14', '2024-06-19'),
    (19, 19, 30, 25, 75.00, 5, 19, 19, 19, '2024-07-02', '2024-07-07'),
    (20, 20, 70, 65, 165.00, 5, 20, 20, 20, '2024-08-17', '2024-08-22'),
    (21, 21, 80, 70, 180.00, 10, 21, 21, 21, '2024-09-03', '2024-09-08'),
    (22, 22, 110, 100, 132.00, 10, 22, 22, 22, '2024-10-19', '2024-10-24'),
    (23, 23, 65, 60, 180.00, 5, 23, 23, 23, '2024-11-14', '2024-11-19'),
    (24, 24, 95, 90, 96.00, 5, 24, 24, 24, '2024-12-02', '2024-12-07'),
    (25, 25, 55, 50, 135.00, 5, 25, 25, 25, '2025-01-08', '2025-01-13'),
    (26, 26, 70, 65, 180.00, 5, 26, 26, 26, '2025-02-14', '2025-02-19'),
    (27, 27, 85, 80, 120.00, 5, 27, 27, 27, '2025-03-21', '2025-03-26'),
    (28, 28, 40, 35, 90.00, 5, 28, 28, 28, '2025-04-08', '2025-04-13'),
    (29, 29, 60, 55, 150.00, 5, 29, 29, 29, '2025-05-15', '2025-05-20'),
    (30, 30, 75, 70, 165.00, 5, 30, 30, 30, '2025-06-22', '2025-06-27');
```

Queries

Food Waste.

```sql
-- Food Waste
SELECT
    IngredientName,
    SUM(QuantityExpired) AS TotalWaste
FROM RestaurantFact
JOIN IngredientDimension ON RestaurantFact.IngredientID = IngredientDimension.IngredientID
WHERE WastageDate BETWEEN '2000-01-01' AND '3024-01-11'
GROUP BY IngredientName;
```

Profit Per Ingredient.

```sql
-- Profit Per Ingredient
SELECT
    IngredientName,
        Category,
        SUM(UnitPrice - PurchasePrice) AS Profit
FROM RestaurantFact
JOIN IngredientDimension ON RestaurantFact.IngredientID = IngredientDimension.IngredientID
WHERE SaleDate BETWEEN '2000-01-01' AND '3000-01-01'
GROUP BY IngredientName;
```

Help

Search

untitled.py

Settings

C: > Users > eoin0 > untitled.py > ...

```python
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy as np
import mysql.connector

# connect to database
connection = mysql.connector.connect(
    host="localhost",
    user="root",
    database="Restaurant"
)

# Create a cursor to execute SQL queries
cursor = connection.cursor()

# Execute your SQL query - the years are stupid to catch all data
query = """
SELECT
    IngredientName,
    Category,
    SUM(UnitPrice - PurchasePrice) AS TotalProfit,
    (QuantitySold) AS TotalSold
FROM RestaurantFact
JOIN IngredientDimension ON RestaurantFact.IngredientID = IngredientDimension.IngredientID
WHERE SaleDate BETWEEN '2000-01-01' AND '3024-01-11'
GROUP BY IngredientName;
"""

cursor.execute(query)

# Fetch the result set
result = cursor.fetchall()

# Close the database connection
cursor.close()
connection.close()

# Check if the result is not empty
if not result:
    print("No data found.")
else:
    # Extract data for plotting
    ingredient_names, categories, total_profit, total_quantity_sold = zip(*result)

    # Convert Decimal values to float - needed for plotting the data due to library limitations
    total_profit = [float(value) for value in total_profit]
    total_quantity_sold = [float(value) for value in total_quantity_sold]

    # Define ingredient_indices - sets an order for the names
    ingredient_indices = np.arange(len(ingredient_names))
```

Help

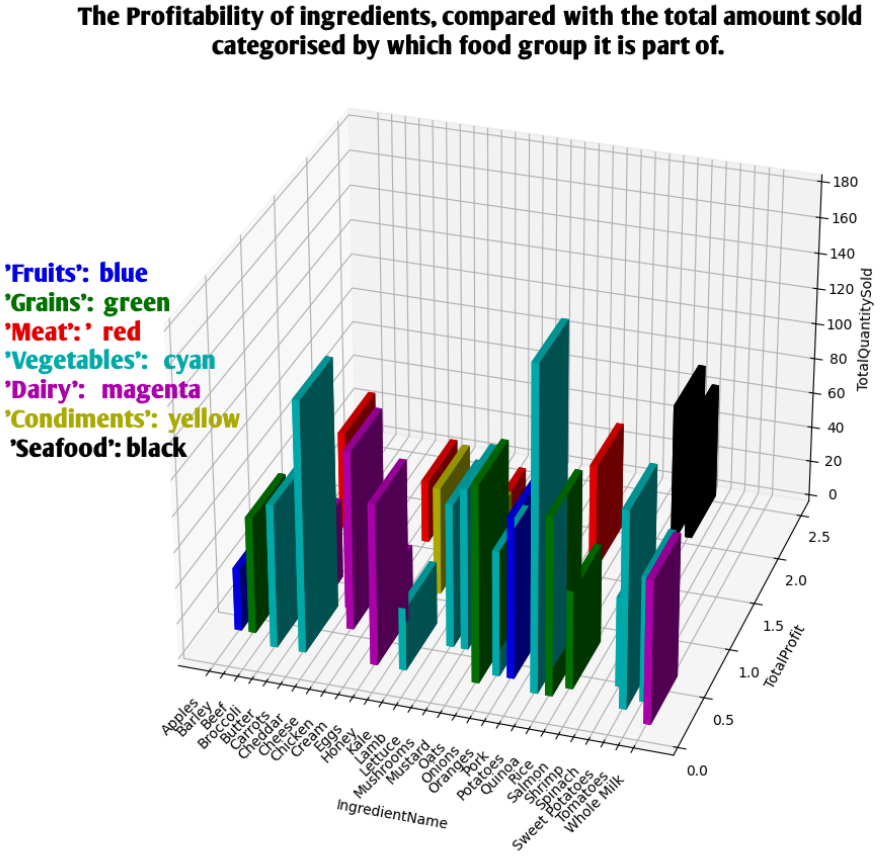
Search

untitled.py

Settings

C: > Users > eoin0 > untitled.py > ...

```python
    # Map categories to colours
    category_color_mapping = {
        'Fruits': 'b',          # blue
        'Grains': 'g',          # green
        'Meat': 'r',            # red
        'Vegetables': 'c',      # cyan
        'Dairy': 'm',           # magenta
        'Condiments': 'y',      # yellow
        'Seafood': 'k',         # black
    }

    # Get the colours for each category - have to use usa spelling ew
    colors = [category_color_mapping[category] for category in categories]

    # Plot the 3D bar chart with thicker bars - it was difficult to see prior
    fig = plt.figure(figsize=(5, 5))
    ax = fig.add_subplot(111, projection='3d')

    bar_width = 0.5
    ax.bar3d(ingredient_indices, total_profit, np.zeros_like(total_profit),
             bar_width, bar_width, total_quantity_sold, color=colors)

    ax.set_xticks(ingredient_indices)
    ax.set_xticklabels(ingredient_names, rotation=45, ha='right')  # Rotate it to see it better
    ax.set_xlabel('IngredientName', labelpad=30)

    ax.set_ylabel('TotalProfit')
    ax.set_zlabel('TotalQuantitySold')

    plt.show()
```



Potatoes have the highest quantity sold but a rather meagre profit, this tells us that the price we sell it at can be increased.

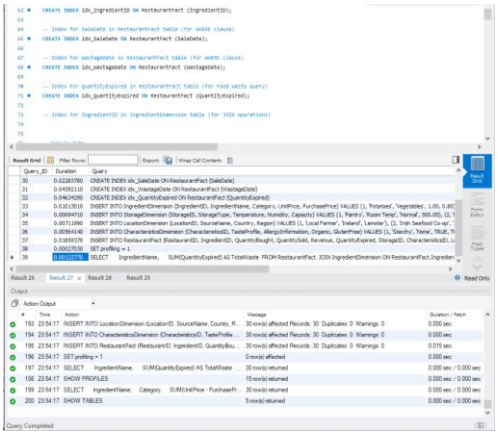
Kale has the lowest profit and the second lowest amount sold; we need to reduce the price we buy it.

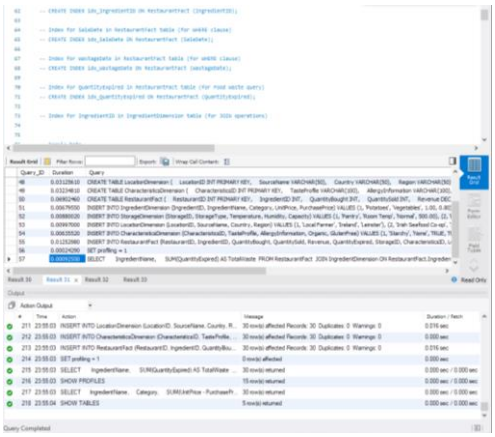
All types of seafood have a good profit margin and a middling amount of sales; this would imply we are selling it correctly.

Changes made to optimise read access.

A star schema was chosen for our relationships rather than a snowflake as all our customers are stars of the world. The star schema allows us to achieve faster querying speeds due to its relative lack of complexity in comparison to the snowflake. This will allow us to gain insight into the ever-changing situation of stockroom inventory. The perishable nature of many of our products is what encourages us to use a system of less normalisation with a focus on greater write and read speed, which will allow those who need to read queries to get the answers they seek faster. This allows us to better reach our business requirements of minimising waste and maximising wealth.

We indexed our table through the field of ingredient name, this was done by





References

We used ChatGPT to generate the data, we found it was the most reliable for pairing foodstuff to its real characteristics. It was also used for assistance in formatting the graph of profit.