DATABASE

MANAGEMENT

COURSEWORK 1: PMM Grocery Supermarket

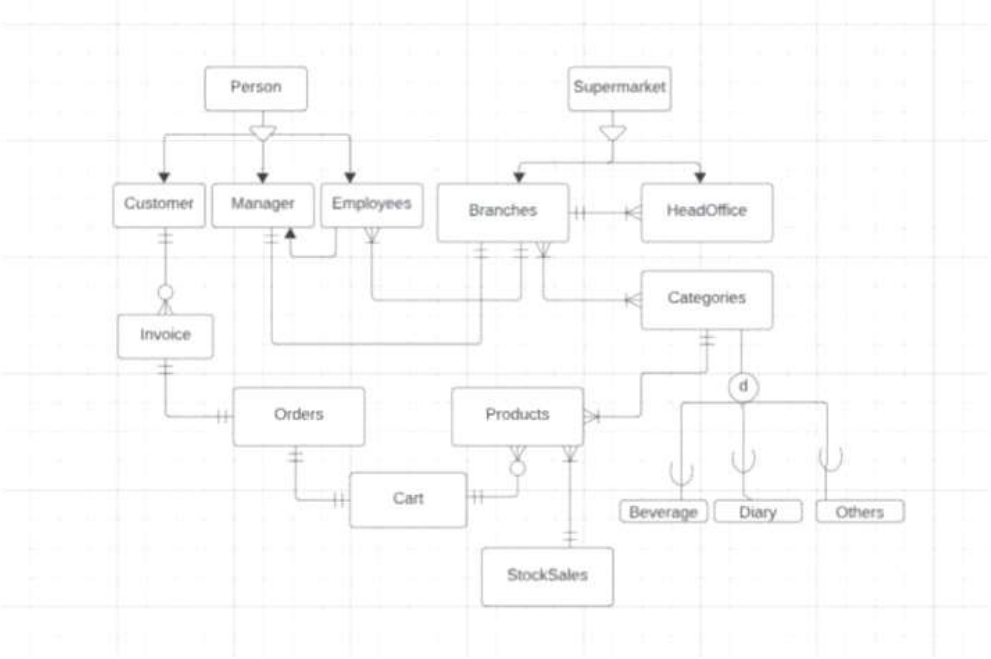
School of Computing, University of Portsmouth– Portsmouth

StudentID:2202479 (Individual work because i dint find)

Date: 10 November 2023

Task T1:

EERD



Generated using Lucid chart

Description: Above figure represent EERD diagram of the PMM Grocery Supermarket has total 13 tables was created for the purpose of this database. Some assumption was made during the creation of this EERD as mentioned in Task 2.

Task T2:

Assumptions:

This project involves designing Grocery Supermarket database.

- Apart from the guidelines given I have used generalization/specialization techniques while designing EERD. For instance, Customer, Employee etc. have inherited basic attributes from it Person entity.
- I have designed separate table for manager other than employee which has one-to-one relationship between branches to help in queries generation later and reduce complexity further.
- For product categories with entity name as “Category” I have used concepts of enhanced ERD for defining multiple categories. It must be noted that no new table is created for each specific category and it is just for sake of representation. StockSale contain stock information.
- In customer table, I have added attributes so customer can access supermarket online i.e. Username, Password etc. Cart entity is introduced to store order details temporarily during transaction. It has one–to-one relationship between orders and one to many relationship with products. It is created for better handling of data.
- In product entity rating attribute is added.
- Invoice entity does handles order and payment related information.
- I have designed one to many relationship between invoice and customer.
- I have created queries for reports management.

Task T3: Script/Data Dictionary:

- **Table: Supermarket**

Attribute Name	PK or AK	DataType & Size	Domain and Constraints	FK Reference	Description
SupermarketID	PK	Int	Not null, AUTO_INCREMENT, UNIQUE		Primary key - Identifier for the supermarket
SupermarketName		varchar(50)	NOT NULL		
SupermarketAddress		varchar(50)	NOT NULL		
SupermarketPhoneNumber		varchar(50)	NOT NULL		
SupermarketEmail		varchar(50)	NOT NULL		

- **Table: HeadOffice**

Attribute Name	PK or AK	DataType & Size	Domain and Constraints	FK Reference	Description
ho_id	PK	int	Not null, AUTO_INCREMENT, UNIQUE	SupermarketID	Identifier for the Headoffice
ho_name		varchar(50)	NOT NULL		
ho_address		varchar(50)	NOT NULL		

ho_phone_number		varchar(50)	NOT NULL		
ho_email		varchar(50)	NOT NULL		

- **Table: Branch**

Attribute Name	PK or AK	DataType & Size	Domain and Constraints	FK Reference	Description
BranchID	PK	int	Not null, AUTO_INCREMENT, UNIQUE	SupermarketID	
BranchName		varchar(50)	NOT NULL		
HeadOfficeID		int	NOT NULL	ho_id	
BranchAddress		varchar(50)	NOT NULL		
BranchPhoneNumber		varchar(50)	NOT NULL,		
BranchEmail		varchar(50)	NOT NULL,		

- **Table: Product Category**

Attribute Name	PK or AK	DataType & Size	Domain and Constraints	FK Reference	Description
CategoryID	PK	int	Not null,AUTO_INCREMENT, UNIQUE		
Category_Name		varchar(50)	NOT NULL		

- **Table: BranchProductCategory**

Attribute Name	PK or AK	DataType & Size	Domain and Constraints	FK Reference	Description
BranchProductCategoryI	PK	int	Not		

D			null,AUTO_INCREMENT, UNIQUE		
BranchID		int	NOT NULL	BranchID	Branch table
ProductCategoryID		int	NOT NULL	CategoryID	Product Category table

- **Table: Person**

Attribute Name	PK or AK	DataType & Size	Domain and Constraints	FK Reference	Description
PersonID	PK	int	Not null,AUTO_INCREMENT, UNIQUE		
Username	AK	varchar(50)	NOT NULL, UNIQUE		
Password		varchar(50)	NOT NULL		
FirstName		varchar(50)	NOT NULL		
LastName		varchar(50)			
PhoneNumber		varchar(50)			

- **Table: Employee**

Attribute Name	PK or AK	DataType & Size	Domain and Constraints	FK Reference	Description
EmployeeID	PK	int	Not null,AUTO_INCREMENT, UNIQUE	PersonID	
FirstName		varchar(50)	NOT NULL		
LastName		varchar(50)			
PhoneNumber		int			
Branch_ID		int	NOT NULL	Branch_ID	
Email		varchar(50)			
Department		varchar(50)			

JobTitle		varchar(50)			
Salary		varchar(50)			
StartDate		date			
EndDate		date			
Username	AK	varchar(50)	NOT NULL, UNIQUE		
Password		varchar(50)	NOT NULL		

- **Table: Customer**

Attribute Name	PK or AK	DataType & Size	Domain and Constraints	FK Reference	Description
CU_ID	PK	int	Not null,AUTO_INCREMENT, UNIQUE	PersonID	
First_name		varchar(50)	NOT NULL		
Last_Name		varchar(50)			
Email		varchar(50)			
Address		varchar(50)			
Phone_number		varchar(50)			
Username	AK	varchar(50)	NOT NULL, UNIQUE		
Password		varchar(50)	NOT NULL		
IsOnline		boolean			

- **Manager**

Attribute Name	PK or AK	DataType & Size	Domain and Constraints	FK Reference	Description
ManagerID	PK	int	Not null,AUTO_INCREMENT, UNIQUE	Employee_ID	Employee Table
FirstName		varchar(50)	NOT NULL		
LastName		varchar(50)			
PhoneNumber		varchar(50)			
Branch_ID		int	NOT NULL	Branch_ID	Branch Table
Email		varchar(50)			
Department		varchar(50)			
JobTitle		varchar(50)			
Salary		varchar(50)			
StartDate		date			
EndDate		date			

Username	AK	varchar(50)	NOT NULL, UNIQUE		

- **Invoice:**

Attribute Name	PK or AK	DataType & Size	Domain and Constraints	FK Reference	Description
InvoiceID	PK	int	AUTO_INCREMENT, UNIQUE		Primary Key - Identifier for the invoice
OrderID	AK	int	NOT NULL	Order_Id	Unique constraint for the OrderID
CustomerID		int	NOT NULL	CU_ID	Identifier for the associated customer
InvoiceDate		timestamp	NOT NULL		Date and time of the invoice
TotalAmount		varchar(50)	NOT NULL		Total amount in the invoice
TaxAmount		varchar(50)			Amount of tax in the invoice
Discount		varchar(50)			Discount applied to the invoice
ShippingCost		varchar(50)			Cost of shipping in the invoice
DueDate		timestamp			Date by which payment is due
PaymentStatus		varchar(50)			Status of payment (Paid, Pending, Credited, Declined)

- **Order**

Attribute Name	PK or AK	DataType & Size	Domain and Constraints	FK Reference	Description
order_id	PK	int	Not null,AUTO_INCREMENT, UNIQUE		Primary Key - Identifier for the order
customer_id		int	NOT NULL, DEFAULT	CU_Id	Identifier for the associated customer
order_date		timestamp			Date and time of the order
shipping_address		varchar(50)			Shipping address for the order
order_total		int			Total cost of the order
order_status		varchar(50)	NOT NULL		Status of the order (Delivered, Pending, In Process

- **Cart**

Attribute Name	PK or AK	DataType & Size	Domain and Constraints	FK Reference	Description
CartID	PK	int	Not null,AUTO_INCREMENT, UNIQUE		Primary Key - Identifier for the cart
OrderID		int	NOT NULL	Order_id	Identifier for the associated order
ProductID		int	NOT NULL		Identifier for the associated product
Quantity		int	NOT NULL		Quantity of the product in the cart
Unit_price		varchar(50)	NOT NULL		Price per unit

					of the product
TotalPrice		varchar(50)	NOT NULL		Total cost for the products in the cart

- **Product**

Attribute Name	PK or AK	DataType & Size	Domain and Constraints	FK Reference	Description
ProductID	PK	int	Not null,AUTO_INCREMENT, UNIQUE		Primary Key - Identifier for the product
ProductName		varchar(50)	NOT NULL		Name of the product
ProductPrice		varchar(50)	NOT NULL		Price of the product
Ingredients		varchar(50)	NOT NULL		Ingredients used in the product
NetWeight		varchar(50)	NOT NULL		Net weight of the product
Lifestyle		varchar(50)	NOT NULL		Lifestyle description for the product
NutritionInfo		varchar(50)	NOT NULL		Nutritional information
StorageInstructions		varchar(50)	NOT NULL		Instructions for storage
ProductCategoryID		int	NOT NULL	ProductCategoryID	Identifier for associated product category
Rating		varchar(50)			Rating of the product

Onsale		boolean			Indicates if the product is on sale

- **Sale**

Attribute Name	PK or AK	DataType & Size	Domain and Constraints	FK Reference	Description
sale_id	PK	int	Not null,AUTO_INCREMENT, UNIQUE		Primary Key - Identifier for the sale
product_id		int		Product (ProductID)	Identifier for associated product from Product table
sale_date		date			Date of the sale
sold_quantity		int			Quantity of products sold in the sale
Revenue		varchar(50)			Revenue generated from the sale
Productprice		varchar(50)			Price of the product at the time of the sale

SQL of database:

1: Table name: Supermarket

```
CREATE TABLE Supermarket
(
    SupermarketID  int GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    SupermarketName varchar(50) NOT NULL,
    SupermarketAddress varchar(50) NOT NULL,
    SupermarketPhoneNumber varchar(50) NOT NULL,
```

```
    SupermarketEmail varchar(50) NOT NULL



);
```

2: Table name: HeadOffice

```
CREATE TABLE Headoffice
(
    ho_id int GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    ho_name varchar(50) NOT NULL,
    ho_address varchar(50) NOT NULL,
    ho_phone_number varchar(50) NOT NULL,
    ho_email varchar(50) NOT NULL

    );
```

3: Table name: Branch

```
CREATE TABLE Branch
(
    BranchID int GENERATED ALWAYS AS IDENTITY,
    BranchName varchar(50) NOT NULL,
    BranchAddress varchar(50) NOT NULL,
     BranchPhoneNumber varchar(50) NOT NULL,
     BranchEmail varchar(50) NOT NULL,
     ho_id int NOT NULL,

PRIMARY KEY(BranchID),
  CONSTRAINT fk_Headoffice
     FOREIGN KEY(ho_id)
           REFERENCES Headoffice(ho_id)



    );

INSERT INTO Branch(BranchName , BranchAddress ,BranchPhoneNumber ,BranchEmail,ho_id  )
VALUES ('Londonbridge', 'Lo3 2qwt', '07856444','branch@gmail.com','2');

INSERT INTO Branch(BranchName , BranchAddress ,BranchPhoneNumber ,BranchEmail,ho_id  )
VALUES ('charlthddh', 'cho3 2qwt', '07844','branch@gmail.com','1');
```

4: Table name: Product Category

```
CREATE TABLE ProductCategory
(
   CategoryID int GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
   CategoryName varchar(50) NOT NULL
);

INSERT INTO ProductCategory(CategoryName ) VALUES ('Drinks');
```

5: Table name: Branch Product Category

```
CREATE TABLE BranchProductCategory
(
   BranchProductCategoryID  int GENERATED ALWAYS,
   BranchID int NOT NULL,
   ProductCategoryID int NOT NULL,


PRIMARY KEY(BranchProductCategoryID ),
  CONSTRAINT fk_Branch
    FOREIGN KEY(BranchID )
          REFERENCES Headoffice(BranchID ),

 CONSTRAINT fk_Headoffice
    FOREIGN KEY(CategoryID )
          REFERENCES ProductCategory(CategoryID )


);
INSERT INTO BranchProductCategory(BranchID,ProductCategoryID ) VALUES ('1','3');
INSERT INTO BranchProductCategory(BranchID,ProductCategoryID ) VALUES ('3','1');
```

6: Table name: Person

```
CREATE TABLE Person
(
   PersonID int GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
   Username varchar(50) NOT NULL,
   Password varchar(50) NOT NULL,
   FirstName varchar(50) NOT NULL,
   LastName varchar(50) NOT NULL,
```

```
    PhoneNumber varchar(50) NOT NULL,
    Email varchar(50) NOT NULL

);
```

7: Table name: Manager

```
CREATE TABLE Manager
(
    ManagerID int GENERATED ALWAYS AS IDENTITY,
    FirstName varchar(50) NOT NULL,
    LastName varchar(50) NOT NULL,
    PhoneNumber varchar(50) NOT NULL,
    Email varchar(50) NOT NULL,
    Department varchar(50) NOT NULL,
    JobTitle varchar(50) NOT NULL,
    Salary int NOT NULL,
    StartDate varchar(50) NOT NULL,
    EndDate varchar(50) NOT NULL,
    EmployeeID int NOT NULL,
    Username varchar(50) NOT NULL,
    Password varchar(50) NOT NULL,
    BranchID int NOT NULL,

PRIMARY KEY(ManagerID ),
  CONSTRAINT fk_Employee
    FOREIGN KEY(EmployeeID)
          REFERENCES Employee(EmployeeID),

  CONSTRAINT fk_Branch
    FOREIGN KEY(BranchID )
          REFERENCES Branch(BranchID )

);
```

8: Table name: Employee

```
CREATE TABLE Employee
(
```

```
    EmployeeID int GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
    FirstName varchar(50) NOT NULL,
    LastName varchar(50) NOT NULL,
    PhoneNumber varchar(50) NOT NULL,
    Email varchar(50) NOT NULL,
    Department varchar(50) NOT NULL,
    JobTitle varchar(50) NOT NULL,
    Salary varchar(50) NOT NULL,
    StartDate varchar(50) NOT NULL,
    EndDate varchar(50) NOT NULL
);
```

9: Table name: Customer

```
CREATE TABLE Customer
 (
          Customerid int GENERATED ALWAYS AS IDENTITY PRIMARY KEY,
          FirstName VARCHAR ( 50 ) NOT NULL,
          LastName VARCHAR ( 50 ) NOT NULL,
         EmailAddress VARCHAR ( 50 ) NOT NULL,
         PhoneNumber VARCHAR ( 50 ) NOT NULL,
         Username VARCHAR ( 50 ) NOT NULL,
         Password VARCHAR ( 50 ) NOT NULL
);
INSERT INTO Customer(FirstName ,LastName ,EmailAddress , PhoneNumber,Username
,Password) VALUES ('roy','jjason','hony@gmail.com','070544235','roy1','12345');

INSERT INTO Customer(FirstName ,LastName ,EmailAddress , PhoneNumber,Username
,Password) VALUES ('youn','faqe','youn@gmail.com','0745674235','youn12','178745');

INSERT INTO Customer(FirstName ,LastName ,EmailAddress , PhoneNumber,Username
,Password) VALUES ('wqas','ali','wqas@gmail.com','072344235','wqas1234','173445');
```

10: Table name: Invoice

```
CREATE TABLE Invoice
(
    InvoiceID int GENERATED ALWAYS AS IDENTITY,
    OrderID int NOT NULL,
    CustomerID int NOT NULL,
```

```
    InvoiceDate VARCHAR ( 50 ) NOT NULL,
    TotalAmount int NOT NULL,
    TaxAmount int NOT NULL,
    Discount int NOT NULL,
    ShippingCost int NOT NULL,
    DueDate VARCHAR ( 50 ) NOT NULL,
    PaymentStatus VARCHAR ( 50 ) NOT NULL,

PRIMARY KEY(InvoiceID ),
   CONSTRAINT fk_Order1
      FOREIGN KEY(orderid )
            REFERENCES Order1(orderid ),

  CONSTRAINT fk_Customer
      FOREIGN KEY(Custo merid )
            REFERENCES Customer(Customerid )

);
```

11: Table name: Order1

```
CREATE TABLE Order1
(
    orderid int GENERATED ALWAYS AS IDENTITY,
    customerid int NOT NULL,
    orderdate VARCHAR ( 50 ) NOT NULL,
    shippingaddress VARCHAR ( 50 ) NOT NULL,
    ordertotal VARCHAR ( 50 ) NOT NULL,
    orderstatus VARCHAR ( 50 ) NOT NULL,

 PRIMARY KEY(orderid ),
  CONSTRAINT fk_Customer
      FOREIGN KEY(Customerid )
            REFERENCES Customer(Customerid )


       );
INSERT INTO Order1(customerid ,orderdate , shippingaddress ,ordertotal ,orderstatus) VALUES
('2','20-11-2023','p03 2re','1510','paid');

INSERT INTO Order1(customerid ,orderdate , shippingaddress ,ordertotal ,orderstatus) VALUES
('3','21-11-2019','p04 2ue','17810','paid');
```

```
INSERT INTO Order1(customerid ,orderdate , shippingaddress ,ordertotal ,orderstatus) VALUES
('4','1-09-2019','p01 2ue','17810','unpaid');
```

12: Table name: Cart

```
CREATE TABLE Cart
(
   CartID int GENERATED ALWAYS AS IDENTITY ,
   OrderID int NOT NULL,
   ProductID int NOT NULL,
   Quantity int NOT NULL,
   Unitprice int NOT NULL,
   TotalPrice int NOT NULL,
PRIMARY KEY(CartID ),
  CONSTRAINT fk_Order1
     FOREIGN KEY(orderid )
          REFERENCES Order1(orderid )

);
```

13: Table name: Product

```
CREATE TABLE Product
(
   ProductID int GENERATED ALWAYS AS IDENTITY ,
   ProductName VARCHAR ( 50 ) NOT NULL,
   ProductPrice int NOT NULL,
   Ingredients VARCHAR ( 50 ) NOT NULL,
   NetWeight int NOT NULL,
   Lifestyle VARCHAR ( 50 ) NOT NULL,
   NutritionInfo VARCHAR ( 50 ) NOT NULL,
   StorageInstructions VARCHAR ( 50 ) NOT NULL,
   CategoryID int NOT NULL,
   ratingn int NOT NULL,
   onsale int NOT NULL,
PRIMARY KEY(ProductID ),
  CONSTRAINT fk_ProductCategory
     FOREIGN KEY(CategoryID )
          REFERENCES ProductCategory(CategoryID)
```

```
);
```

14: Table name: Sales

```
CREATE TABLE Sales
(
    saleid int GENERATED ALWAYS AS IDENTITY,
    ProductID int NOT NULL,
    saledate VARCHAR ( 50 ) NOT NULL,
    soldquantity int NOT NULL,
    revenue int NOT NULL,
    productprice int NOT NULL,

PRIMARY KEY(saleid),
  CONSTRAINT fk_Product
    FOREIGN KEY(ProductID)
          REFERENCES Product(ProductID)
);
```

Task T4: Queries

Query No 1: Report of product sold

```
SELECT
```

```
    P.ProductName, P.ProductID, SUM(S.soldquantity) AS total_sold
FROM
    Sales S
INNER JOIN
    Product P ON S.ProductID = P.ProductID
GROUP BY
    P.ProductName, P.ProductID;
```

Description:

This query is the report of all the product sold in a month .It shows the product Name product id and total of sold 1 item and shows in a table

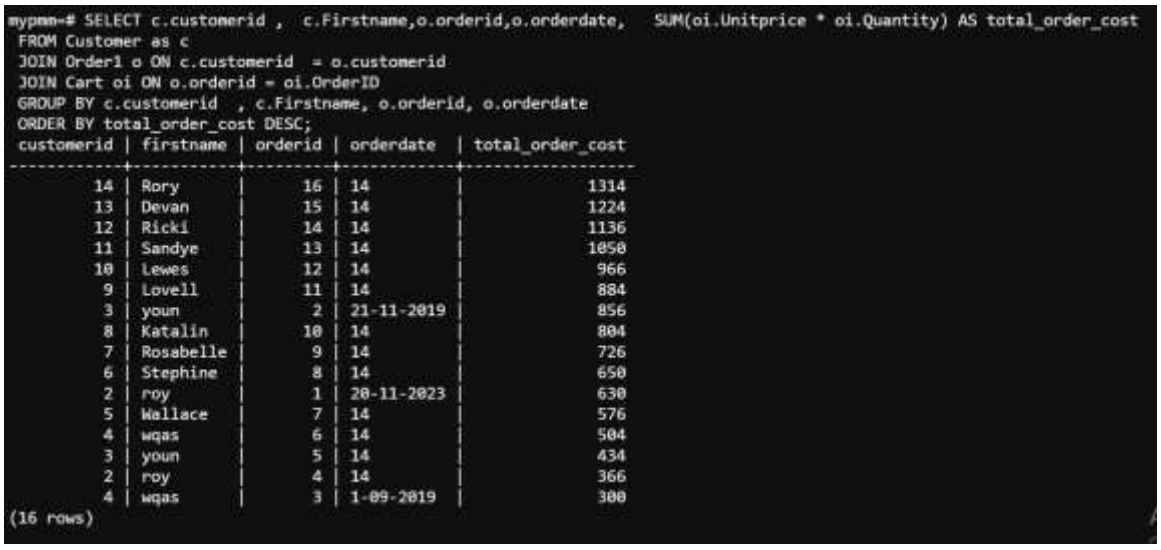
```
mypmm=# SELECT
    P.ProductName, P.ProductID, SUM(S.soldquantity) AS total_sold
FROM
    Sales S
INNER JOIN
    Product P ON S.ProductID = P.ProductID
GROUP BY
    P.ProductName, P.ProductID;
 productname | productid | total_sold
-------------+-----------+------------
 Milk        |         3 |          9
 butter      |         2 |         55
 Milk        |         1 |         12
(3 rows)
```

Query no 2: Customer Order Cost Summary

```
SELECT c.customerid , c.Firstname,o.orderid,o.orderdate, SUM(oi.Unitprice * oi.Quantity)
AS total_order_cost

FROM Customer as c

JOIN Order1 o ON c.customerid = o.customerid

JOIN Cart oi ON o.orderid = oi.OrderID

GROUP BY c.customerid , c.Firstname, o.orderid, o.orderdate

ORDER BY total_order_cost DESC;
```

Description:

This SQL query retrieves information about the total order cost for each customer's orders. It accomplishes this by joining three tables: Customer, Order, and Cart from the public schema.



```
mypmn-# SELECT c.customerid , c.Firstname,o.orderid,o.orderdate, SUM(oi.Unitprice * oi.Quantity) AS total_order_cost
 FROM Customer as c
 JOIN Order1 o ON c.customerid = o.customerid
 JOIN Cart oi ON o.orderid = oi.OrderID
 GROUP BY c.customerid , c.Firstname, o.orderid, o.orderdate
 ORDER BY total_order_cost DESC;
 customerid | firstname | orderid | orderdate  | total_order_cost
------------+-----------+---------+------------+------------------
         14 | Rory      |      16 | 14         |             1314
         13 | Devan     |      15 | 14         |             1224
         12 | Ricki     |      14 | 14         |             1136
         11 | Sandye    |      13 | 14         |             1050
         10 | Lewes     |      12 | 14         |              966
          9 | Lovell    |      11 | 14         |              884
          3 | youn      |       2 | 21-11-2019 |              856
          8 | Katalin   |      10 | 14         |              804
          7 | Rosabelle |       9 | 14         |              726
          6 | Stephine  |       8 | 14         |              650
          2 | roy       |       1 | 20-11-2023 |              630
          5 | Wallace   |       7 | 14         |              576
          4 | wqas      |       6 | 14         |              504
          3 | youn      |       5 | 14         |              434
          2 | roy       |       4 | 14         |              366
          4 | wqas      |       3 | 1-09-2019  |              300
(16 rows)
```

Query no 3: Summary of Customers and Their Total Orders

```
SELECT c.customerid , c.Firstname, c.LastName, c.emailaddress, COUNT(o.orderid) AS Total_Orders

 FROM Customer c

 LEFT JOIN Order1 o ON c.customerid = o.customerid

 GROUP BY c.customerid, c.Firstname, c.LastName, c.emailaddress;
```

Description:

This SQL query retrieves a summary of customers and their orders. It retrieves and consolidates customer-related information alongside the total number of orders each customer has placed.

```
mypmm=# SELECT c.customerid , c.Firstname, c.LastName, c.emailaddress, COUNT(o.orderid) AS Total_Orders
 FROM Customer c
 LEFT JOIN Order1 o ON c.customerid = o.customerid
 GROUP BY c.customerid, c.Firstname, c.LastName, c.emailaddress;
 customerid | firstname | lastname |         emailaddress         | total_orders
------------+-----------+----------+------------------------------+--------------
          4 | wqas      | ali      | wqas@gmail.com               |            2
         10 | Lewes     | Wards    | lwards5@prlog.org            |            1
          6 | Stephine  | Burgess  | sburgess1@blogger.com        |            1
         14 | Rory      | Cherrett | rcherrett9@ameblo.jp         |            1
         13 | Devan     | Allbrook | dallbrook8@de.vu             |            1
          2 | roy       | jjason   | hony@gmail.com               |            2
         16 | Gerri     | Arenson  | garensonb@unblog.fr          |            1
         11 | Sandye    | Fulloway | sfulloway6@icio.us           |            1
          9 | Lovell    | France   | lfrance4@tmall.com           |            1
          7 | Rosabelle | Lakenden | rlakenden2@jalbum.net        |            1
         15 | Eduino    | Hastler  | ehastlera@nyu.edu            |            1
         12 | Ricki     | Frudd    | rfrudd7@microsoft.com        |            1
         19 | Cyndi     | Cherrett | ccherrette@printfriendly.com |            0
          3 | youn      | faqe     | youn@gmail.com               |            2
         17 | Adiana    | Dron     | adronc@jigsy.com             |            0
          1 | roy       | jjason   | hony@gmail.com               |            0
          5 | Wallace   | Mecco    | wmecco0@seesaa.net           |            1
         18 | Konstanze | Piller   | kpillerd@tumblr.com          |            0
          8 | Katalin   | MacGahey | kmacgahey3@adobe.com         |            1
(19 rows)
```

\

Query no 4: Retrieving Specific Product Information with Associated Category Name

SELECT p.ProductID, p.ProductName, p.ProductPrice, p.Ingredients, p.NetWeight, ProductCategory .CategoryID

FROM Product p

JOIN ProductCategory ON p.CategoryID = ProductCategory .CategoryID;

Description:

This SQL query retrieves data from the "Product" and "ProductCategory" tables .It gives specific product information with associated category name.

```
mypmm=# SELECT p.ProductID, p.ProductName, p.ProductPrice, p.Ingredients, p.NetWeight, ProductCategory .CategoryID
FROM Product p
JOIN ProductCategory  ON p.CategoryID = ProductCategory .CategoryID;
 productid |        productname         | productprice | ingredients | netweight | categoryid
-----------+----------------------------+--------------+-------------+-----------+------------
         2 | butter                     |            1 | protein     |        10 |          1
         3 | Milk                       |            2 | minerals    |       110 |          2
         1 | Milk                       |            2 | vitamint    |       110 |          2
        34 | Mustard Prepared           |          322 | 4           |        59 |          3
        35 | Monkfish - Fresh           |          323 | 5           |        60 |          4
        36 | Spic And Span All Purpose  |          324 | 6           |        61 |          5
        37 | Eggplant Italian           |          325 | 7           |        62 |          6
        38 | Pickle - Dill              |          326 | 8           |        63 |          7
        39 | Salt - Celery              |          327 | 9           |        64 |          8
        40 | Croissant, Raw - Mini      |          328 | 10          |        65 |          9
        41 | Spice - Greek 1 Step       |          329 | 11          |        66 |         10
        42 | Sesame Seed                |          330 | 12          |        67 |         11
        43 | Garlic - Peeled            |          331 | 13          |        68 |         12
        44 | Spic And Span All Purpose  |          332 | 14          |        69 |         13
        45 | Hand Towel                 |          333 | 15          |        70 |         14
        46 | Pork - Loin, Center Cut    |          334 | 16          |        71 |         15
        47 | Wood Chips - Regular       |          335 | 17          |        72 |         16
        48 | Sprite - 355 Ml            |          336 | 18          |        73 |         17
(18 rows)
```

Appendix:

Following date was inserted in respective tables and these data was generated using https://mockaroo.com.

```
insert into Branch (CategoryName) values ('Beef - Flank Steak');
insert into Branch (CategoryName) values ('Cheese - Gouda');
insert into Branch (CategoryName) values ('Container - Clear 32 Oz');
insert into Branch (CategoryName) values ('Cabbage - Red');
insert into Branch (CategoryName) values ('Wine - Red, Metus Rose');
insert into Branch (CategoryName) values ('Venison - Racks Frenched');
insert into Branch (CategoryName) values ('Nut - Pine Nuts, Whole');
insert into Branch (CategoryName) values ('Pork - Backs - Boneless');
insert into Branch (CategoryName) values ('Bread - Pita');
insert into Branch (CategoryName) values ('Pastry - Raisin Muffin - Mini');
insert into Branch (CategoryName) values ('Cream - 10%');
insert into Branch (CategoryName) values ('Pasta - Agnolotti - Butternut');
insert into Branch (CategoryName) values ('Cake - Lemon Chiffon');
insert into Branch (CategoryName) values ('Trueblue - Blueberry Cranberry');
insert into Branch (CategoryName) values ('Bread - Corn Muffaletta');



insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values
('bchristou0', 'iG7/pXMM$', 'Bart', 'Christou', '+62 173 755 6623', 'bchristou0@flickr.com');
insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values ('jrex1',
'pL7,p2rle`l/I', 'Jere', 'Rex', '+63 431 920 9724', 'jrex1@alexa.com');
insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values
('ebiglin2', 'tL9_pj9o}&w6,0)x', 'Eddie', 'Biglin', '+86 856 206 0619', 'ebiglin2@fc2.com');
insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values
('dhostan3', 'jF0_?x4d~Ye', 'Demetri', 'Hostan', '+504 519 648 2419', 'dhostan3@theatlantic.com');
insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values
('jboreland4', 'gE6/IibY&Y@S##L', 'Jarad', 'Boreland', '+61 604 159 0927', 'jboreland4@123-reg.co.uk');
insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values
('bnickell5', 'kP0+`J\\', 'Bowie', 'Nickell', '+46 270 983 9537', 'bnickell5@soundcloud.com');
insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values
('hkneel6', 'mQ2''d"LYo<m9', 'Holt', 'Kneel', '+52 258 827 0767', 'hkneel6@senate.gov');
insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values
('mrosson7', 'fJ7.H6Ti!$E<V8', 'Mitchel', 'Rosson', '+380 861 163 3510', 'mrosson7@digg.com');
insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values
('aspillard8', 'xV9.K>jnTZ', 'Angie', 'Spillard', '+86 496 759 2350', 'aspillard8@tuttocitta.it');
insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values
('tcamps9', 'mD2<@qV''Y.j', 'Truman', 'Camps', '+86 335 742 2723', 'tcamps9@bing.com');
insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values
('usharera', 'eN0#dXZk', 'Ulrick', 'Sharer', '+351 279 768 8238', 'usharera@themeforest.net');
```

```
insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values
('sbeecraftb', 'pB6<dlLg', 'Saunders', 'Beecraft', '+52 333 388 8638', 'sbeecraftb@symantec.com');
insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values
('wrubinovc', 'yX0%AA.Q=4x7HU', 'Wendell', 'Rubinov', '+86 350 774 9464', 'wrubinovc@umich.edu');
insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values
('wrappaportd', 'jZ0`t<IG_5**0', 'Walt', 'Rappaport', '+63 906 594 7550', 'wrappaportd@altervista.org');
insert into Person(Username , Password , FirstName , LastName , PhoneNumber , Email ) values
('dgowansone', 'cK6)"2G1w', 'Demetri', 'Gowanson', '+7 836 769 0119', 'dgowansone@unc.edu');



insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Wallace', 'Mecco', 'wmecco0@seesaa.net', '337-731-5104', 'wmecco0', 'qG4$@I#P7uR');
insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Stephine', 'Burgess', 'sburgess1@blogger.com', '701-317-9555', 'sburgess1', 'aX6_4XzZu5D=l');
insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Rosabelle', 'Lakenden', 'rlakenden2@jalbum.net', '720-245-1393', 'rlakenden2', 'zR0@E8.m');
insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Katalin', 'MacGahey', 'kmacgahey3@adobe.com', '994-837-9344', 'kmacgahey3', 'hG6=XZB#15');
insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Lovell', 'France', 'lfrance4@tmall.com', '218-861-3456', 'lfrance4', 'nS5,QF7K`cCUEZDt');
insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Lewes', 'Wards', 'lwards5@prlog.org', '473-992-7985', 'lwards5', 'sX1`u9G3p');
insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Sandye', 'Fulloway', 'sfulloway6@icio.us', '182-251-0885', 'sfulloway6', 'qR3\BaT_(');
insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Ricki', 'Frudd', 'rfrudd7@microsoft.com', '799-608-9066', 'rfrudd7', 'aW8{LU6I');
insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Devan', 'Allbrook', 'dallbrook8@de.vu', '189-804-2781', 'dallbrook8', 'aC2\H$bfoEef%.`g');
insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Rory', 'Cherrett', 'rcherrett9@ameblo.jp', '829-681-4480', 'rcherrett9', 'cF4%Fx%=O~*d');
insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Eduino', 'Hastler', 'ehastlera@nyu.edu', '734-709-7283', 'ehastlera', 'kL2%X7n?Nej');
insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Gerri', 'Arenson', 'garensonb@unblog.fr', '262-290-8941', 'garensonb', 'aL3?UC9.~(');
insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Adiana', 'Dron', 'adronc@jigsy.com', '984-296-6820', 'adronc', 'zN3}OJd_d=WC');
insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Konstanze', 'Piller', 'kpillerd@tumblr.com', '957-877-8154', 'kpillerd', 'xC7\AqV\');
insert into Employee(firstname, lastname, emailaddress  , phonenumber , username , password ) values
('Cyndi', 'Cherrett', 'ccherrette@printfriendly.com', '609-823-9388', 'ccherrette', 'hE3>6)N?');
```

```
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (2, 3, 14, '7180', 4, 1, 3, '8/26/2022', 'ante nulla justo aliquam quis');
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (3, 4, 14, '3759', 5, 2, 4, '6/3/2023', 'adipiscing molestie hendrerit');
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (4, 5, 14, '8780', 6, 3, 5, '3/14/2023', 'volutpat eleifend donec ut dolor');
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (5, 6, 14, '432', 7, 4, 6, '2/9/2022', 'ligula nec sem duis');
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (6, 7, 14, '616', 8, 5, 7, '8/14/2023', 'rutrum neque aenean auctor');
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (7, 8, 14, '40', 9, 6, 8, '10/9/2022', 'primis in faucibus orci');
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (8, 9, 14, '04181', 10, 7, 9, '8/25/2022', 'ipsum primis in faucibus');
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (9, 10, 14, '90', 11, 8, 10, '10/19/2021', 'odio curabitur convallis');
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (10, 11, 14, '6', 12, 9, 11, '11/24/2021', 'sed sagittis nam congue');
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (11, 12, 14, '0593', 13, 10, 12, '10/26/2022', 'non velit nec');
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (12, 13, 14, '023', 14, 11, 13, '4/10/2023', 'nulla pede ullamcorper');
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (13, 14, 14, '9', 15, 12, 14, '3/4/2022', 'vivamus vestibulum sagittis sapien');
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (14, 15, 14, '708', 16, 13, 15, '10/15/2023', 'leo rhoncus sed');
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (15, 16, 14, '338', 17, 14, 16, '4/28/2022', 'augue vestibulum rutrum');
```

```
insert into Invoice (OrderID , CustomerID , InvoiceDate , TotalAmount , TaxAmount , Discount , ShippingCost , DueDate , PaymentStatus) values (16, 17, 14, '69', 18, 15, 17, '2/26/2023', 'eleifend luctus ultricies eu nibh');



insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (2, 14, 'Room 1220', 59, 1);
insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (3, 14, 'Apt 575', 60, 2);
insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (4, 14, 'PO Box 535', 61, 3);
insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (5, 14, 'Suite 96', 62, 4);
insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (6, 14, 'Apt 1375', 63, 5);
insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (7, 14, '2nd Floor', 64, 6);
insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (8, 14, 'Room 1147', 65, 7);
insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (9, 14, '8th Floor', 66, 8);
insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (10, 14, '15th Floor', 67, 9);
insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (11, 14, 'PO Box 4330', 68, 10);
insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (12, 14, 'Room 1822', 69, 11);
insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (13, 14, 'Suite 47', 70, 12);
insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (14, 14, 'Suite 44', 71, 13);
insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (15, 14, 'Apt 787', 72, 14);
insert into Order1 (customerid, orderdate  , shippingaddress , ordertotal , orderstatus) values (16, 14, 'Room 279', 73, 15);


insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (2, 2, 4, 59, 68);
insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (3, 3, 5, 60, 69);
insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (4, 4, 6, 61, 70);
insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (5, 5, 7, 62, 71);
insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (6, 6, 8, 63, 72);
insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (7, 7, 9, 64, 73);
insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (8, 8, 10, 65, 74);
```

```
insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (9, 9, 11, 66, 75);
insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (10, 10, 12, 67, 76);
insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (11, 11, 13, 68, 77);
insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (12, 12, 14, 69, 78);
insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (13, 13, 15, 70, 79);
insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (14, 14, 16, 71, 80);
insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (15, 15, 17, 72, 81);
insert into Cart (orderid, productid , quantity , unitprice, totalprice) values (16, 16, 18, 73, 82);



insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo ,
StorageInstructions , CategoryID, ratingn , onsale ) values ('Mustard Prepared', 322, 4, 59, 101, 2, 1, 3, 2,
1);
insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo ,
StorageInstructions , CategoryID, ratingn , onsale ) values ('Monkfish - Fresh', 323, 5, 60, 102, 3, 2, 4, 3,
2);
insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo ,
StorageInstructions , CategoryID, ratingn , onsale ) values ('Spic And Span All Purpose', 324, 6, 61, 103,
4, 3, 5, 4, 3);
insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo ,
StorageInstructions , CategoryID, ratingn , onsale ) values ('Eggplant Italian', 325, 7, 62, 104, 5, 4, 6, 5,
4);
insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo ,
StorageInstructions , CategoryID, ratingn , onsale ) values ('Pickle - Dill', 326, 8, 63, 105, 6, 5, 7, 6, 5);
insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo ,
StorageInstructions , CategoryID, ratingn , onsale ) values ('Salt - Celery', 327, 9, 64, 106, 7, 6, 8, 7, 6);
insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo ,
StorageInstructions , CategoryID, ratingn , onsale ) values ('Croissant, Raw - Mini', 328, 10, 65, 107, 8, 7,
9, 8, 7);
insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo ,
StorageInstructions , CategoryID, ratingn , onsale ) values ('Spice - Greek 1 Step', 329, 11, 66, 108, 9, 8,
10, 9, 8);
insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo ,
StorageInstructions , CategoryID, ratingn , onsale ) values ('Sesame Seed', 330, 12, 67, 109, 10, 9, 11, 10,
9);
insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo ,
StorageInstructions , CategoryID, ratingn , onsale ) values ('Garlic - Peeled', 331, 13, 68, 110, 11, 10, 12,
11, 10);
insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo ,
StorageInstructions , CategoryID, ratingn , onsale ) values ('Spic And Span All Purpose', 332, 14, 69, 111,
12, 11, 13, 12, 11);
insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo ,
StorageInstructions , CategoryID, ratingn , onsale ) values ('Hand Towel', 333, 15, 70, 112, 13, 12, 14, 13,
12);
insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo ,
```

```
StorageInstructions , CategoryID, ratingn , onsale ) values ('Pork - Loin, Center Cut', 334, 16, 71, 113, 14, 13, 15, 14, 13);
insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo , StorageInstructions , CategoryID, ratingn , onsale ) values ('Wood Chips - Regular', 335, 17, 72, 114, 15, 14, 16, 15, 14);
insert into Product (ProductName, ProductPrice, Ingredients, NetWeight, Lifestyle, NutritionInfo , StorageInstructions , CategoryID, ratingn , onsale ) values ('Sprite - 355 Ml', 336, 18, 73, 115, 16, 15, 17, 16, 15);




insert into Sales (soldquantity, revenue, productprice) values (8, 58, 100);
insert into Sales (soldquantity, revenue, productprice) values (9, 116, 200);
insert into Sales (soldquantity, revenue, productprice) values (10, 174, 300);
insert into Sales (soldquantity, revenue, productprice) values (11, 232, 400);
insert into Sales (soldquantity, revenue, productprice) values (12, 290, 500);
insert into Sales (soldquantity, revenue, productprice) values (13, 348, 600);
insert into Sales (soldquantity, revenue, productprice) values (14, 406, 700);
insert into Sales (soldquantity, revenue, productprice) values (15, 464, 800);
insert into Sales (soldquantity, revenue, productprice) values (16, 522, 900);
insert into Sales (soldquantity, revenue, productprice) values (17, 580, 1000);
insert into Sales (soldquantity, revenue, productprice) values (18, 638, 1100);
insert into Sales (soldquantity, revenue, productprice) values (19, 696, 1200);
insert into Sales (soldquantity, revenue, productprice) values (20, 754, 1300);
insert into Sales (soldquantity, revenue, productprice) values (21, 812, 1400);
insert into Sales (soldquantity, revenue, productprice) values (22, 870, 1500);
```