ISM-671 Organizing Data For Analytics

Wholesale store Database

THE UNIVERSITY of NORTH CAROLINA
GREENSBORO
Bryan School of Business and Economics

Sreekar Gownivaripalli

Table of Contents

Executive Summary:

Wholesale Store is a leading wholesale distribution company that specializes in providing a wide range of products to various retail businesses across multiple sectors, including food and beverage, electronics, and household goods. The company has established a vast network of vendors and logistics partners, enabling efficient and reliable distribution services across the region.

The cornerstone of Wholesale Store's success lies in its customer-centric approach, focusing on understanding the unique needs of each retail client and offering tailored solutions to meet those needs. This approach includes implementing strategic promotional campaigns and offering various payment options to cater to diverse client preferences. Wholesale Store's promotional strategies involve offering discounts, special deals, and targeted marketing campaigns to drive sales and foster customer loyalty. Additionally, the company provides a range of payment methods, including credit cards, online payment platforms, and bank transfers, to ensure a seamless and convenient transaction experience for its clients. Wholesale Store's team of dedicated sales representatives plays a crucial role in building long-lasting relationships with clients, characterized by trust, reliability, and mutual growth. The company leverages its comprehensive database solution to streamline inventory management, order processing, vendor and client management, sales and logistics coordination, and financial tracking. This database system incorporates modules for managing promotions, tracking discounts applied to products and sales, and maintaining records of payment methods used for transactions.

As Wholesale Store looks to the future, it aims to expand its footprint, exploring new markets and embracing innovative technologies to enhance its operational efficiencies and customer service further. The company is dedicated to remaining at the forefront of the wholesale distribution industry, continuously adapting to the changing retail landscape and the evolving needs of its clients while leveraging its robust database solution and strategic promotional and payment capabilities.

Problem Statement

As Wholesale Store continues to expand its operations and client base, it faces increasingly complex challenges that threaten its efficiency, customer satisfaction, and scalability. The company's existing manual and semi-automated systems are proving inadequate to handle the growing complexity and volume of transactions, leading to several critical operational issues:

Inventory Management Inefficiency: Difficulty in tracking and managing stock levels in real-time, leading to overstocking or stockouts.

Order Processing Delays: Slow manual processes for order entry, processing, and fulfillment.

Vendor and Client Management Limitations: Inefficiencies in managing extensive vendor and client databases with varying needs and specifications.

Sales and Logistics Coordination: Challenges in coordinating between sales teams and logistics partners, affecting delivery timelines.

Data Accessibility and Reporting: Limited ability to access real-time data and generate insightful reports for strategic decision-making.

Scalability Concerns: Existing systems are not equipped to scale with the business, limiting potential growth.

Promotion Management Shortcomings: Lack of an integrated system to manage and track the impact of promotional campaigns on sales and inventory.

Payment Processing Inefficiencies: Restricted payment options and slow processing times that affect customer experience and operational fluidity.

While WholesaleStore has implemented various promotional strategies to drive sales and foster customer loyalty, managing these campaigns manually or through disparate systems has become increasingly challenging. The lack of a centralized system for tracking promotions, applying discounts, and analyzing the effectiveness of campaigns hinders the company's ability to optimize its marketing efforts and maximize revenue potential.

Furthermore, the current systems in place offer limited payment options for clients, potentially restricting sales and hindering customer satisfaction. Comprehensive financial tracking and reconciliation processes are also hindered due to the lack of integration between payment methods and sales records.

To address these challenges, "WholesaleStore" requires a comprehensive database solution that streamlines inventory management, order processing, vendor and client management, sales and logistics coordination, and financial tracking. Additionally, the database system must incorporate modules for managing promotions, tracking discounts applied to products and sales, and maintaining records of payment methods used for transactions. Implementing such a database system is essential for enhancing operational efficiencies, improving customer service, optimizing marketing efforts, and ensuring "WholesaleStore" remains competitive in the wholesale distribution industry.

Entities:

We have identified below entities.

- Categories: Classifies products into various categories.
- Clients: Records details of retail businesses purchasing from "Wholesale Store."
- Sales Representative: Information about employees responsible for managing client accounts.
- Logistics: Details of shipping and logistics partners used for order delivery.
- Vendors: Information on suppliers from whom products are sourced.
- Products: The catalog of items available for sale includes vendor and category details.
- Sales: Records of sales transactions, including client, sales representative, and logistics details.
- SaleDetails: Detailed items of each sale, including product and quantity.
- Promotions: Handles various discounts, special deals or promotional campaigns.
- Payment Methods: To track and manage how clients pay for their purchases.

Attributes:

These are the attributes after applying normalization.

Categories: CategoryID (Primary Key), CategoryName, Description,

Clients: ClientID (Primary Key), ClientName, ContactName, Phone, Email, Address, City, ZipCode, Country, RegistrationDate

Sales Representatives: RepresentativeID (Primary Key), LastName, FirstName, Phone, Email, HireDate, Region

Logistics: CarrierID (Primary Key), CarrierName, Phone, Email, Address, City, ZipCode, Country

Vendors: VendorID (Primary Key), VendorName, ContactName, Address, City, ZipCode, Country, Phone, Email, RegistrationDate

Products: ProductID (Primary Key), ProductName, VendorID (Foreign Key), CategoryID (Foreign Key), Unit, Quantity, Price

Sales: SaleID (Primary Key), ClientID (Foreign Key), RepresentativeID (Foreign Key), SaleDate, CarrierID (Foreign Key)

SaleDetails: SaleDetailID (Primary Key), SaleID (Foreign Key), ProductID (Foreign Key), Quantity, TotalAmount

Promotions: PromotionID (Primary Key), Description, StartDate, EndDate, DiscountRate, Conditions

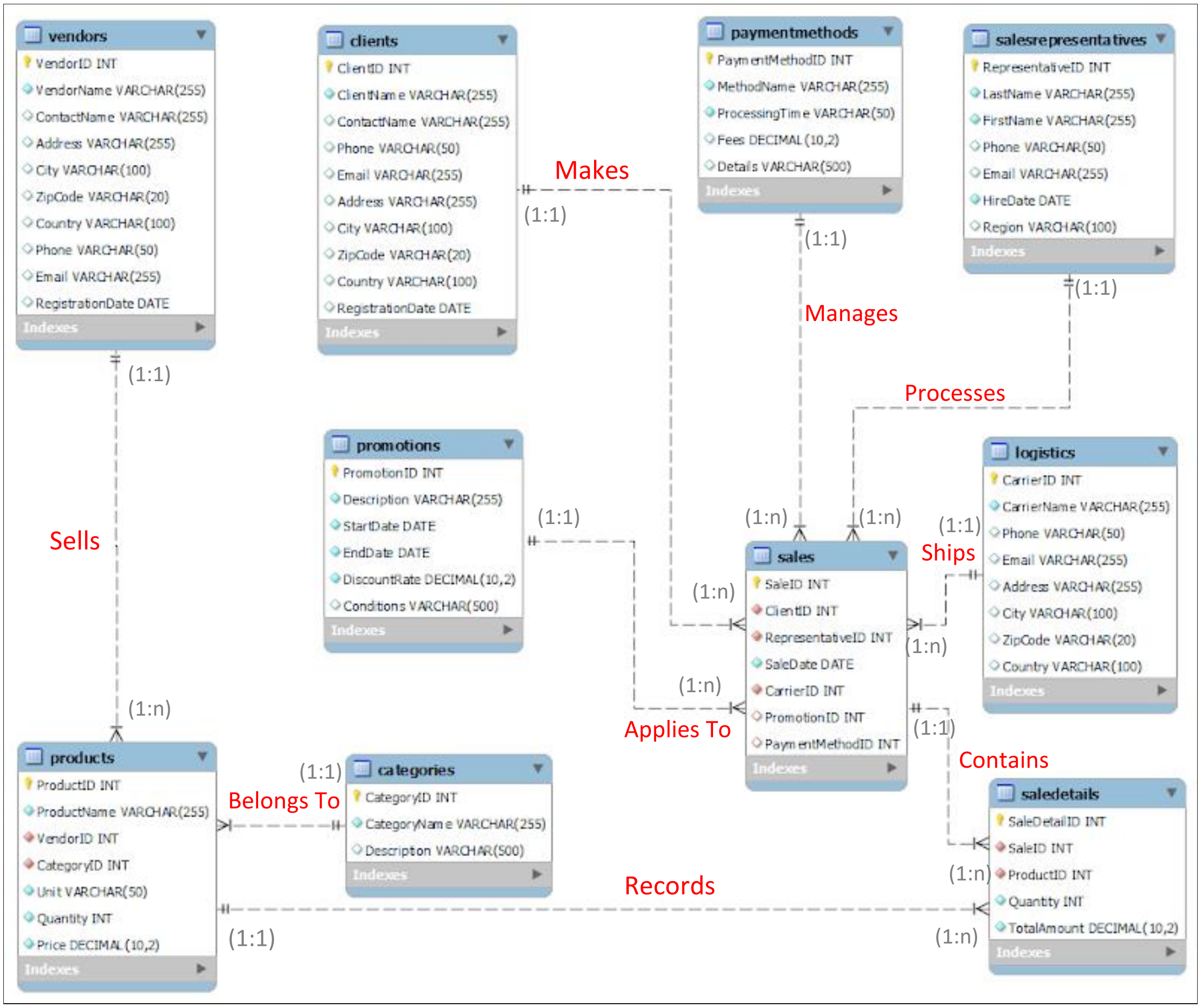
Payment Methods: PaymentMethodID (Primary Key), MethodName, ProcessingTime, Fees, Details

Relationships:
Below are few identified key relations.

Parent Entity	Child Entity	Relationship Type	Foreign Key	Description
Categories	Products	One-to-Many	CategoryID	Each category can include many products, but each product is associated with only one category.
Vendors	Products	One-to-Many	VendorID	Each vendor can supply many products, but each product is sourced from only one vendor.
Clients	Sales	One-to-Many	ClientID	A client can have multiple sales orders, but each sale is made to one client.
Sales Representatives	Sales	One-to-Many	RepresentativeID	A sales representative can handle multiple sales, but each sale is managed by one representative.
Logistics	Sales	One-to-Many	CarrierID	A logistics carrier can be responsible for delivering multiple sales orders, but each sale uses one carrier.
Sales	SaleDetails	One-to-Many	SaleID	Each sale can consist of multiple products (sale details), but each sale detail refers to one sale.
Products	SaleDetails	Many-to-Many	ProductID, SaleID	Products and sales have a many-to-many relationship, realized indirectly through the SaleDetails table.
Promotions	Products	Many-to-Many	PromotionID	Promotions can apply to multiple products, and a product can be involved in multiple promotions.

Promotions	Sales	One-to-Many	PromotionID	Each sale might include one or more promotions, but each promotion is linked to specific sales.
Payment Methods	Sales	One-to-Many	PaymentMethodID	Each sale uses one payment method, but each method can be used in multiple sales.

Entity-Relationship Diagram:
Below is the ERD.



ERD

Functional Requirements for the Database Solution:

Inventory Management:

- Track and manage stock levels in real-time for all products.
- Enable automated alerts for low stock levels to prevent stockouts.
- Provide insights into inventory turnover rates and product popularity.

Order Processing:

- Streamline the process of order entry, processing, and fulfillment.
- Enable automation for order confirmation, invoicing, and shipping.
- Integrate with inventory management to ensure accurate stock allocation.

Vendor and Client Management:

- Maintain comprehensive databases of vendors and clients.
- Record and manage vendor details including contact information and registration dates.
- Track client information such as contact details, registration dates, and purchase history.

Sales and Logistics Coordination:

- Facilitate coordination between sales representatives and logistics partners.
- Assign appropriate carriers for each sale and track delivery statuses.
- Ensure timely delivery by optimizing routing and scheduling.

Data Accessibility and Reporting:

- Provide real-time access to data for strategic decision-making.
- Generate insightful reports on sales performance, inventory levels, and client trends.
- Offer customizable reporting options based on user requirements.

Scalability:

- Design the database system to accommodate growth in transactions and data volume.
- Ensure scalability by optimizing database performance and resource utilization.

Promotion Management:

- Create and manage promotional campaigns with various discounts and special deals.
- Track the effectiveness of promotions by monitoring sales performance.
- Apply discounts automatically during checkout based on promotion criteria.

Payment Processing:

- Support multiple payment methods including credit cards, online platforms, and bank transfers.
- Streamline payment processing for efficient transactions.

Security and Compliance:

- Implement robust security measures to protect sensitive data.
- Ensure compliance with relevant regulations such as GDPR or PCI DSS.
- Restrict access to sensitive functionalities based on user roles and permissions.

Backup and Recovery:

- Implement regular backups of the database to prevent data loss.
- Enable efficient recovery mechanisms in case of system failures or disasters.

These functional requirements are crucial for the successful implementation of the comprehensive database solution to address the challenges faced by Wholesale Store and to enhance its operational efficiency, customer service, and competitiveness in the wholesale distribution industry.

Instance Tables:

Categories:

Categories	Key Type	Null/Unique	Data Type	Max Length	FK Reference Table
Categories	PK	Unique	INT	-	-
Categories	-	Non-null	VARCHAR	255	-
Description	-	Null	VARCHAR	500	-

Clients:

Column Name	Key Type	Null/Unique	Data Type	Max Length	FK Reference Table
ClientID	PK	Unique	INT	-	-
ClientName	-	Non-null	VARCHAR	255	-
ContactName	-	Null	VARCHAR	255	-
Phone	-	Null	VARCHAR	50	-
Email	-	Null	VARCHAR	255	-
Address	-	Null	VARCHAR	255	-
City	-	Null	VARCHAR	100	-
ZipCode	-	Null	VARCHAR	20	-
Country	-	Null	VARCHAR	100	-
RegistrationDate	-	Null	DATE	-	-

Sales Representatives:

Column Name	Key Type	Null/Unique	Data Type	Max Length	FK Reference Table
RepresentativeID	PK	Unique	INT	-	-
LastName	-	Non-null	VARCHAR	255	-
FirstName	-	Non-null	VARCHAR	255	-
Phone	-	Null	VARCHAR	50	-
Email	-	Null	VARCHAR	255	-
HireDate	-	Non-null	DATE	-	-
Region	-	Null	VARCHAR	100	-

Logistics:

Column Name	Key Type	Null/Unique	Data Type	Max Length	FK Reference Table
CarrierID	PK	Unique	INT	-	-
CarrierName	-	Non-null	VARCHAR	255	-
Phone	-	Null	VARCHAR	50	-
Email	-	Null	VARCHAR	255	-
Address	-	Null	VARCHAR	255	-
City	-	Null	VARCHAR	100	-
ZipCode	-	Null	VARCHAR	20	-
Country	-	Null	VARCHAR	100	-

Vendors:

Column Name	Key Type	Null/Unique	Data Type	Max Length	FK Reference Table
VendorID	PK	Unique	INT	-	-
VendorName	-	Non-null	VARCHAR	255	-
ContactName	-	Null	VARCHAR	255	-
Address	-	Null	VARCHAR	255	-
City	-	Null	VARCHAR	100	-
ZipCode	-	Null	VARCHAR	20	-
Country	-	Null	VARCHAR	100	-
Phone	-	Null	VARCHAR	50	-
Email	-	Null	VARCHAR	255	-
RegistrationDate	-	Null	DATE	-	-

Products:

Column Name	Key Type	Null/Unique	Data Type	Max Length	FK Reference Table
ProductID	PK	Unique	INT	-	-
ProductName	-	Non-null	VARCHAR	255	-
VendorID	FK	Non-null	INT	-	Vendors
CategoryID	FK	Non-null	INT	-	Categories
Unit	-	Non-null	VARCHAR	50	-
Quantity	-	Non-null	INT	-	-
Price	-	Non-null	DECIMAL	-	-

Sales:

Column Name	Key Type	Null/Unique	Data Type	Max Length	FK Reference Table
SaleID	PK	Unique	INT	-	-
ClientID	FK	Non-null	INT	-	Clients
RepresentativeID	FK	Non-null	INT	-	Sales Representatives
SaleDate	-	Non-null	DATE	-	-
CarrierID	FK	Non-null	INT	-	Logistics

SaleDetails:

Column Name	Key Type	Null/Unique	Data Type	Max Length	FK Reference Table
SaleDetailID	PK	Unique	INT	-	-
SaleID	FK	Non-null	INT	-	Sales
ProductID	FK	Non-null	INT	-	Products
Quantity	-	Non-null	INT	-	-
TotalAmount	-	Non-null	DECIMAL	-	-

Promotions:

Column Name	Key Type	Null/Unique	Data Type	Max Length	FK Reference Table
PromotionID	PK	Unique	INT	-	-
Description	-	Non-null	VARCHAR	255	-
StartDate	-	Non-null	DATE	-	-
EndDate	-	Non-null	DATE	-	-
DiscountRate	-	Non-null	DECIMAL	-	-

Conditions	-	Null	VARCHAR	500	-

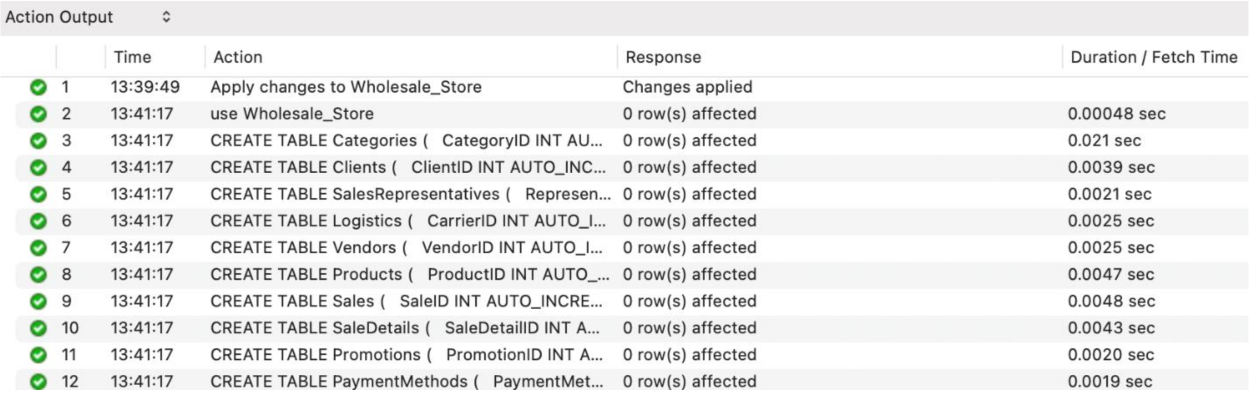
Payment Methods:

Column Name	Key Type	Null/Unique	Data Type	Max Length	FK Reference Table
PaymentMethodID	PK	Unique	INT	-	-
MethodName	-	Non-null	VARCHAR	255	-
ProcessingTime	-	Non-null	VARCHAR	50	-
Fees	-	Null	DECIMAL	-	-
Details	-	Null	VARCHAR	500	-

SQL Queries to create tables:

```
CREATE DATABASE IF NOT EXISTS Wholesale_Store;
use Wholesale_Store;

CREATE TABLE Categories (
  CategoryID INT AUTO_INCREMENT PRIMARY KEY,
  CategoryName VARCHAR(255) NOT NULL,
  Description VARCHAR(500)
);

CREATE TABLE Clients (
  ClientID INT AUTO_INCREMENT PRIMARY KEY,
  ClientName VARCHAR(255) NOT NULL,
  ContactName VARCHAR(255),
  Phone VARCHAR(50),
  Email VARCHAR(255),
  Address VARCHAR(255),
  City VARCHAR(100),
  ZipCode VARCHAR(20),
  Country VARCHAR(100),
  RegistrationDate DATE
);

CREATE TABLE SalesRepresentatives (
  RepresentativeID INT AUTO_INCREMENT PRIMARY KEY,
  LastName VARCHAR(255) NOT NULL,
  FirstName VARCHAR(255) NOT NULL,
  Phone VARCHAR(50),
  Email VARCHAR(255),
  HireDate DATE NOT NULL,
  Region VARCHAR(100)
```

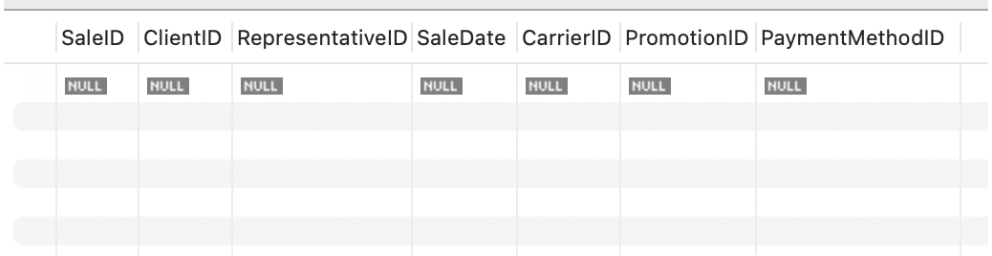
```
);

CREATE TABLE Logistics (
  CarrierID INT AUTO_INCREMENT PRIMARY KEY,
  CarrierName VARCHAR(255) NOT NULL,
  Phone VARCHAR(50),
  Email VARCHAR(255),
  Address VARCHAR(255),
  City VARCHAR(100),
  ZipCode VARCHAR(20),
  Country VARCHAR(100)
);

CREATE TABLE Vendors (
  VendorID INT AUTO_INCREMENT PRIMARY KEY,
  VendorName VARCHAR(255) NOT NULL,
  ContactName VARCHAR(255),
  Address VARCHAR(255),
  City VARCHAR(100),
  ZipCode VARCHAR(20),
  Country VARCHAR(100),
  Phone VARCHAR(50),
  Email VARCHAR(255),
  RegistrationDate DATE
);

CREATE TABLE Products (
  ProductID INT AUTO_INCREMENT PRIMARY KEY,
  ProductName VARCHAR(255) NOT NULL,
  VendorID INT NOT NULL,
  CategoryID INT NOT NULL,
  Unit VARCHAR(50) NOT NULL,
  Quantity INT NOT NULL,
  Price DECIMAL(10,2) NOT NULL,
  FOREIGN KEY (VendorID) REFERENCES Vendors(VendorID),
  FOREIGN KEY (CategoryID) REFERENCES Categories(CategoryID)
);

CREATE TABLE Sales (
  SaleID INT AUTO_INCREMENT PRIMARY KEY,
  ClientID INT NOT NULL,
  RepresentativeID INT NOT NULL,
  SaleDate DATE NOT NULL,
  CarrierID INT NOT NULL,
  FOREIGN KEY (ClientID) REFERENCES Clients(ClientID),
  FOREIGN KEY (RepresentativeID) REFERENCES SalesRepresentatives(RepresentativeID),
  FOREIGN KEY (CarrierID) REFERENCES Logistics(CarrierID)
);

CREATE TABLE SaleDetails (
  SaleDetailID INT AUTO_INCREMENT PRIMARY KEY,
  SaleID INT NOT NULL,
  ProductID INT NOT NULL,
  Quantity INT NOT NULL,
  TotalAmount DECIMAL(10,2) NOT NULL,
  FOREIGN KEY (SaleID) REFERENCES Sales(SaleID),
  FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);

CREATE TABLE Promotions (
  PromotionID INT AUTO_INCREMENT PRIMARY KEY,
  Description VARCHAR(255) NOT NULL,
  StartDate DATE NOT NULL,
  EndDate DATE NOT NULL,
  DiscountRate DECIMAL(10,2) NOT NULL,
  Conditions VARCHAR(500)
);

CREATE TABLE PaymentMethods (
  PaymentMethodID INT AUTO_INCREMENT PRIMARY KEY,
  MethodName VARCHAR(255) NOT NULL,
  ProcessingTime VARCHAR(50) NOT NULL,
  Fees DECIMAL(10,2),
  Details VARCHAR(500)
);
```



		Time	Action	Response	Duration / Fetch Time
✓	1	13:39:49	Apply changes to Wholesale_Store	Changes applied	
✓	2	13:41:17	use Wholesale_Store	0 row(s) affected	0.00048 sec
✓	3	13:41:17	CREATE TABLE Categories (CategoryID INT AU...	0 row(s) affected	0.021 sec
✓	4	13:41:17	CREATE TABLE Clients (ClientID INT AUTO_INC...	0 row(s) affected	0.0039 sec
✓	5	13:41:17	CREATE TABLE SalesRepresentatives (Represen...	0 row(s) affected	0.0021 sec
✓	6	13:41:17	CREATE TABLE Logistics (CarrierID INT AUTO_I...	0 row(s) affected	0.0025 sec
✓	7	13:41:17	CREATE TABLE Vendors (VendorID INT AUTO_I...	0 row(s) affected	0.0025 sec
✓	8	13:41:17	CREATE TABLE Products (ProductID INT AUTO_...	0 row(s) affected	0.0047 sec
✓	9	13:41:17	CREATE TABLE Sales (SaleID INT AUTO_INCRE...	0 row(s) affected	0.0048 sec
✓	10	13:41:17	CREATE TABLE SaleDetails (SaleDetailID INT A...	0 row(s) affected	0.0043 sec
✓	11	13:41:17	CREATE TABLE Promotions (PromotionID INT A...	0 row(s) affected	0.0020 sec
✓	12	13:41:17	CREATE TABLE PaymentMethods (PaymentMet...	0 row(s) affected	0.0019 sec

SQL script to modify a table structure:

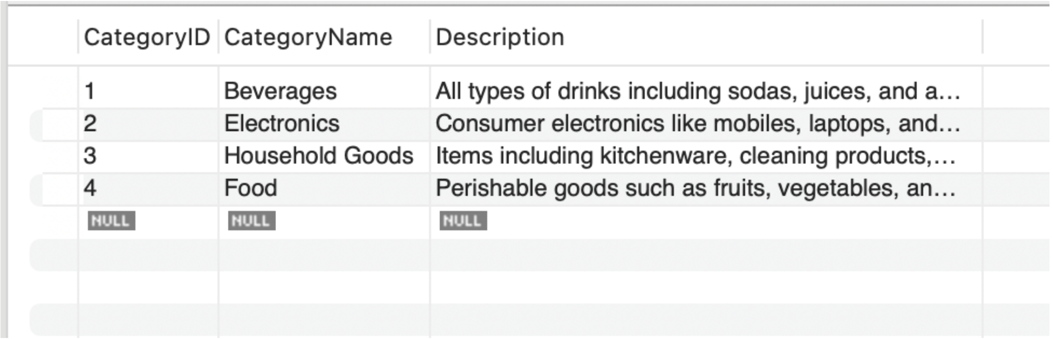
```
use Wholesale_Store;
ALTER TABLE Sales
ADD PromotionID INT NULL,
ADD CONSTRAINT FK_Sales_Promotions FOREIGN KEY (PromotionID) REFERENCES Promotions(PromotionID);

ALTER TABLE Sales
ADD PaymentMethodID INT,
ADD FOREIGN KEY (PaymentMethodID) REFERENCES PaymentMethods(PaymentMethodID);
```



SaleID	ClientID	RepresentativeID	SaleDate	CarrierID	PromotionID	PaymentMethodID
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Inserting data to the tables:

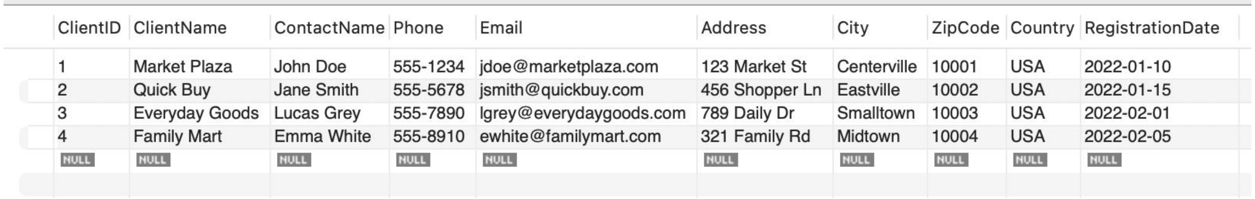
```
INSERT INTO Categories (CategoryName, Description)
VALUES
('Beverages', 'All types of drinks including sodas, juices, and alcoholic beverages.'),
('Electronics', 'Consumer electronics like mobiles, laptops, and other gadgets.'),
('Household Goods', 'Items including kitchenware, cleaning products, and other household necessities.'),
('Food', 'Perishable goods such as fruits, vegetables, and meats.');
```



CategoryID	CategoryName	Description
1	Beverages	All types of drinks including sodas, juices, and a…
2	Electronics	Consumer electronics like mobiles, laptops, and…
3	Household Goods	Items including kitchenware, cleaning products,…
4	Food	Perishable goods such as fruits, vegetables, an…
NULL	NULL	NULL

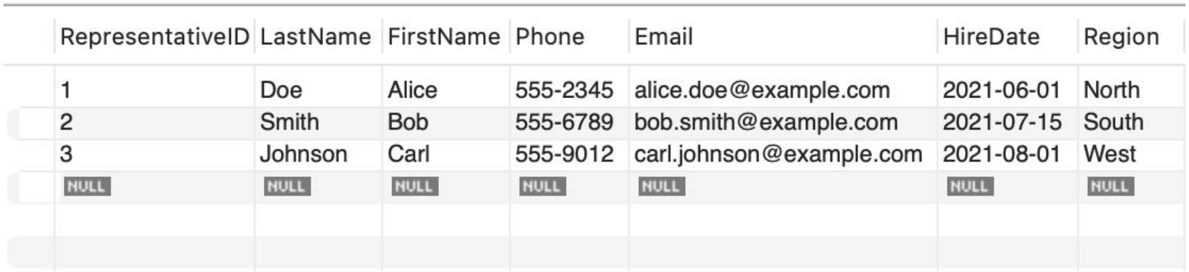
```
INSERT INTO Clients (ClientName, ContactName, Phone, Email, Address, City, ZipCode, Country, RegistrationDate)
VALUES
```

```
('Market Plaza', 'John Doe', '555-1234', 'jdoe@marketplaza.com', '123 Market St', 'Centerville',
'10001', 'USA', '2022-01-10'),
('Quick Buy', 'Jane Smith', '555-5678', 'jsmith@quickbuy.com', '456 Shopper Ln', 'Eastville',
'10002', 'USA', '2022-01-15'),
('Everyday Goods', 'Lucas Grey', '555-7890', 'lgrey@everydaygoods.com', '789 Daily Dr',
'Smalltown', '10003', 'USA', '2022-02-01'),
('Family Mart', 'Emma White', '555-8910', 'ewhite@familymart.com', '321 Family Rd', 'Midtown',
'10004', 'USA', '2022-02-05');
```



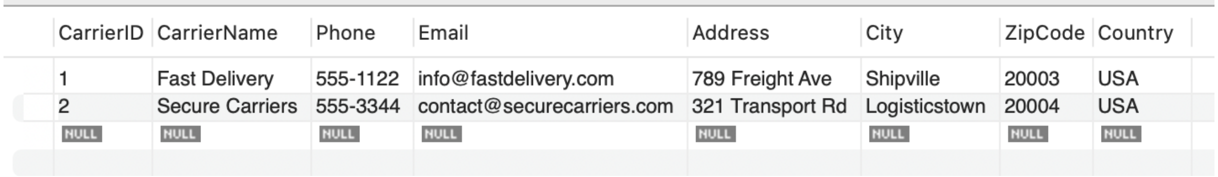
ClientID	ClientName	ContactName	Phone	Email	Address	City	ZipCode	Country	RegistrationDate
1	Market Plaza	John Doe	555-1234	jdoe@marketplaza.com	123 Market St	Centerville	10001	USA	2022-01-10
2	Quick Buy	Jane Smith	555-5678	jsmith@quickbuy.com	456 Shopper Ln	Eastville	10002	USA	2022-01-15
3	Everyday Goods	Lucas Grey	555-7890	lgrey@everydaygoods.com	789 Daily Dr	Smalltown	10003	USA	2022-02-01
4	Family Mart	Emma White	555-8910	ewhite@familymart.com	321 Family Rd	Midtown	10004	USA	2022-02-05
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO SalesRepresentatives (LastName, FirstName, Phone, Email, HireDate, Region)
VALUES
('Doe', 'Alice', '555-2345', 'alice.doe@example.com', '2021-06-01', 'North'),
('Smith', 'Bob', '555-6789', 'bob.smith@example.com', '2021-07-15', 'South'),
('Johnson', 'Carl', '555-9012', 'carl.johnson@example.com', '2021-08-01', 'West');
```



RepresentativeID	LastName	FirstName	Phone	Email	HireDate	Region
1	Doe	Alice	555-2345	alice.doe@example.com	2021-06-01	North
2	Smith	Bob	555-6789	bob.smith@example.com	2021-07-15	South
3	Johnson	Carl	555-9012	carl.johnson@example.com	2021-08-01	West
NULL	NULL	NULL	NULL	NULL	NULL	NULL

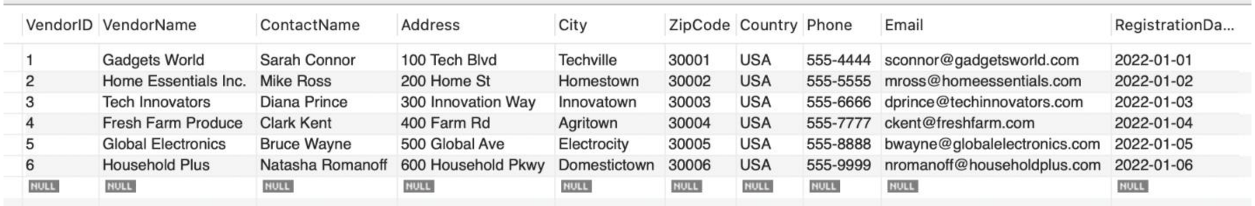
```
INSERT INTO Logistics (CarrierName, Phone, Email, Address, City, ZipCode, Country)
VALUES
('Fast Delivery', '555-1122', 'info@fastdelivery.com', '789 Freight Ave', 'Shipville', '20003', 'USA'),
('Secure Carriers', '555-3344', 'contact@securecarriers.com', '321 Transport Rd', 'Logisticstown',
'20004', 'USA');
```



CarrierID	CarrierName	Phone	Email	Address	City	ZipCode	Country
1	Fast Delivery	555-1122	info@fastdelivery.com	789 Freight Ave	Shipville	20003	USA
2	Secure Carriers	555-3344	contact@securecarriers.com	321 Transport Rd	Logisticstown	20004	USA
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

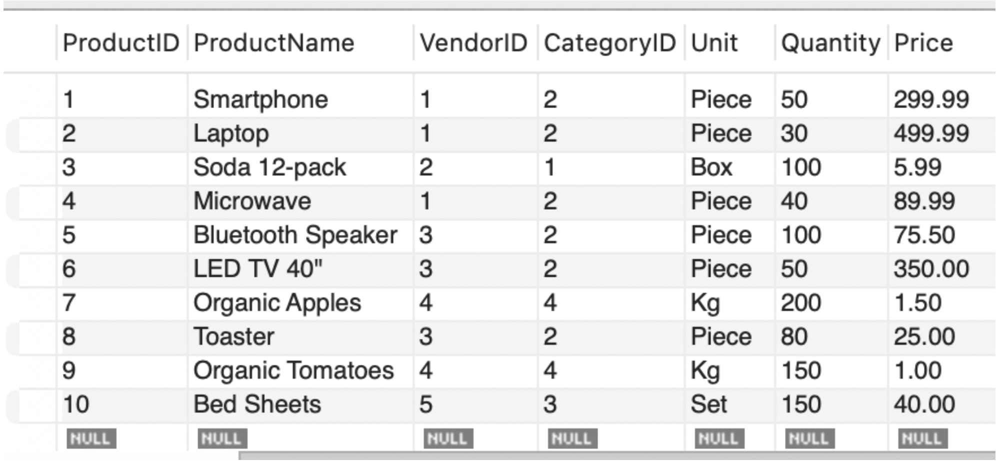
```
INSERT INTO Vendors (VendorName, ContactName, Address, City, ZipCode, Country, Phone,
Email, RegistrationDate)
VALUES
```

```
('Gadgets World', 'Sarah Connor', '100 Tech Blvd', 'Techville', '30001', 'USA', '555-4444',
'sconnor@gadgetsworld.com', '2022-01-01'),
('Home Essentials Inc.', 'Mike Ross', '200 Home St', 'Homestown', '30002', 'USA', '555-5555',
'mross@homeessentials.com', '2022-01-02'),
('Tech Innovators', 'Diana Prince', '300 Innovation Way', 'Innovatown', '30003', 'USA', '555-6666',
'dprince@techinnovators.com', '2022-01-03'),
('Fresh Farm Produce', 'Clark Kent', '400 Farm Rd', 'Agritown', '30004', 'USA', '555-7777',
'ckent@freshfarm.com', '2022-01-04'),
('Global Electronics', 'Bruce Wayne', '500 Global Ave', 'Electrocity', '30005', 'USA', '555-8888',
'bwayne@globalelectronics.com', '2022-01-05'),
('Household Plus', 'Natasha Romanoff', '600 Household Pkwy', 'Domestictown', '30006', 'USA',
'555-9999', 'nromanoff@householdplus.com', '2022-01-06');
```



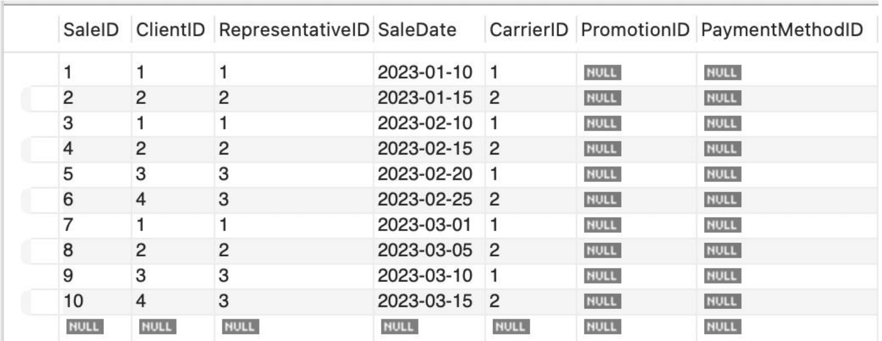
VendorID	VendorName	ContactName	Address	City	ZipCode	Country	Phone	Email	RegistrationDa...
1	Gadgets World	Sarah Connor	100 Tech Blvd	Techville	30001	USA	555-4444	sconnor@gadgetsworld.com	2022-01-01
2	Home Essentials Inc.	Mike Ross	200 Home St	Homestown	30002	USA	555-5555	mross@homeessentials.com	2022-01-02
3	Tech Innovators	Diana Prince	300 Innovation Way	Innovatown	30003	USA	555-6666	dprince@techinnovators.com	2022-01-03
4	Fresh Farm Produce	Clark Kent	400 Farm Rd	Agritown	30004	USA	555-7777	ckent@freshfarm.com	2022-01-04
5	Global Electronics	Bruce Wayne	500 Global Ave	Electrocity	30005	USA	555-8888	bwayne@globalelectronics.com	2022-01-05
6	Household Plus	Natasha Romanoff	600 Household Pkwy	Domestictown	30006	USA	555-9999	nromanoff@householdplus.com	2022-01-06
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO Products (ProductName, VendorID, CategoryID, Unit, Quantity, Price)
VALUES
('Smartphone', 1, 2, 'Piece', 50, 299.99),
('Laptop', 1, 2, 'Piece', 30, 499.99),
('Soda 12-pack', 2, 1, 'Box', 100, 5.99),
('Microwave', 1, 2, 'Piece', 40, 89.99),
('Bluetooth Speaker', 3, 2, 'Piece', 100, 75.50),
('LED TV 40"', 3, 2, 'Piece', 50, 350.00),
('Organic Apples', 4, 4, 'Kg', 200, 1.50),
('Toaster', 3, 2, 'Piece', 80, 25.00),
('Organic Tomatoes', 4, 4, 'Kg', 150, 1.00),
('Bed Sheets', 5, 3, 'Set', 150, 40.00);
```



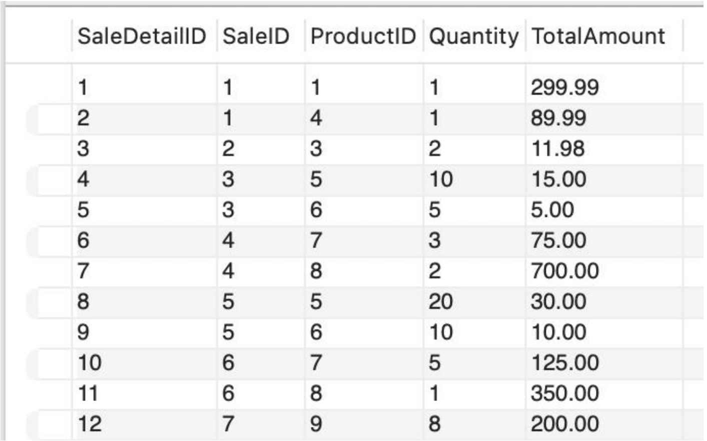
ProductID	ProductName	VendorID	CategoryID	Unit	Quantity	Price
1	Smartphone	1	2	Piece	50	299.99
2	Laptop	1	2	Piece	30	499.99
3	Soda 12-pack	2	1	Box	100	5.99
4	Microwave	1	2	Piece	40	89.99
5	Bluetooth Speaker	3	2	Piece	100	75.50
6	LED TV 40"	3	2	Piece	50	350.00
7	Organic Apples	4	4	Kg	200	1.50
8	Toaster	3	2	Piece	80	25.00
9	Organic Tomatoes	4	4	Kg	150	1.00
10	Bed Sheets	5	3	Set	150	40.00
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO Sales (ClientID, RepresentativeID, SaleDate, CarrierID)
VALUES
(1, 1, '2023-01-10', 1),
(2, 2, '2023-01-15', 2),
(1, 1, '2023-02-10', 1),
(2, 2, '2023-02-15', 2),
(3, 3, '2023-02-20', 1),
(4, 3, '2023-02-25', 2),
(1, 1, '2023-03-01', 1),
(2, 2, '2023-03-05', 2),
(3, 3, '2023-03-10', 1),
(4, 3, '2023-03-15', 2);
```



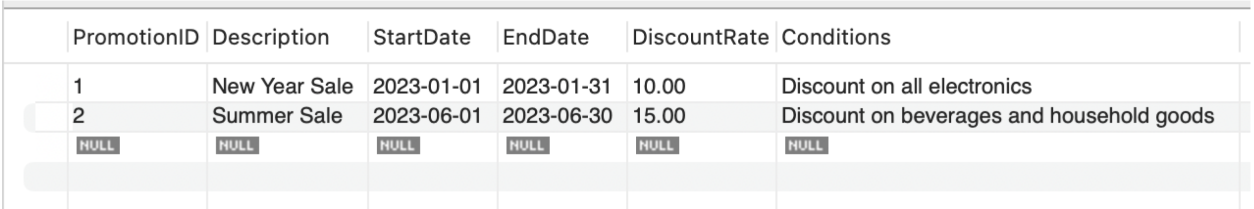
SaleID	ClientID	RepresentativeID	SaleDate	CarrierID	PromotionID	PaymentMethodID
1	1	1	2023-01-10	1	NULL	NULL
2	2	2	2023-01-15	2	NULL	NULL
3	1	1	2023-02-10	1	NULL	NULL
4	2	2	2023-02-15	2	NULL	NULL
5	3	3	2023-02-20	1	NULL	NULL
6	4	3	2023-02-25	2	NULL	NULL
7	1	1	2023-03-01	1	NULL	NULL
8	2	2	2023-03-05	2	NULL	NULL
9	3	3	2023-03-10	1	NULL	NULL
10	4	3	2023-03-15	2	NULL	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO SaleDetails (SaleID, ProductID, Quantity, TotalAmount)
VALUES
(1, 1, 1, 299.99),
(1, 4, 1, 89.99),
(2, 3, 2, 11.98),
(3, 5, 10, 15.00),
(3, 6, 5, 5.00),
(4, 7, 3, 75.00),
(4, 8, 2, 700.00),
(5, 5, 20, 30.00),
(5, 6, 10, 10.00),
(6, 7, 5, 125.00),
(6, 8, 1, 350.00),
(7, 9, 8, 200.00),
(7, 10, 15, 600.00),
(8, 9, 10, 250.00),
(8, 10, 20, 800.00);
```



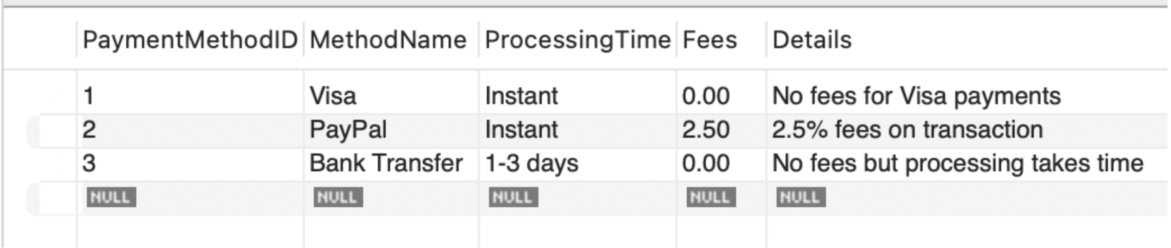
SaleDetailID	SaleID	ProductID	Quantity	TotalAmount
1	1	1	1	299.99
2	1	4	1	89.99
3	2	3	2	11.98
4	3	5	10	15.00
5	3	6	5	5.00
6	4	7	3	75.00
7	4	8	2	700.00
8	5	5	20	30.00
9	5	6	10	10.00
10	6	7	5	125.00
11	6	8	1	350.00
12	7	9	8	200.00

```
INSERT INTO Promotions (Description, StartDate, EndDate, DiscountRate, Conditions)
VALUES
('New Year Sale', '2023-01-01', '2023-01-31', 10, 'Discount on all electronics'),
('Summer Sale', '2023-06-01', '2023-06-30', 15, 'Discount on beverages and household goods');
```



PromotionID	Description	StartDate	EndDate	DiscountRate	Conditions
1	New Year Sale	2023-01-01	2023-01-31	10.00	Discount on all electronics
2	Summer Sale	2023-06-01	2023-06-30	15.00	Discount on beverages and household goods
NULL	NULL	NULL	NULL	NULL	NULL

```
INSERT INTO PaymentMethods (MethodName, ProcessingTime, Fees, Details)
VALUES
('Visa', 'Instant', 0, 'No fees for Visa payments'),
('PayPal', 'Instant', 2.5, '2.5% fees on transaction'),
('Bank Transfer', '1-3 days', 0, 'No fees but processing takes time');
```

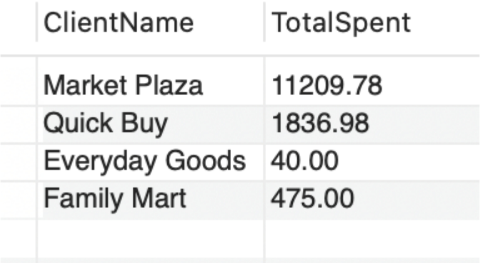


PaymentMethodID	MethodName	ProcessingTime	Fees	Details
1	Visa	Instant	0.00	No fees for Visa payments
2	PayPal	Instant	2.50	2.5% fees on transaction
3	Bank Transfer	1-3 days	0.00	No fees but processing takes time
NULL	NULL	NULL	NULL	NULL

SQL statements (queries, triggers, or functions) to retrieve some meaningful data from the database:

-- Retrieve Total Sales Per Client

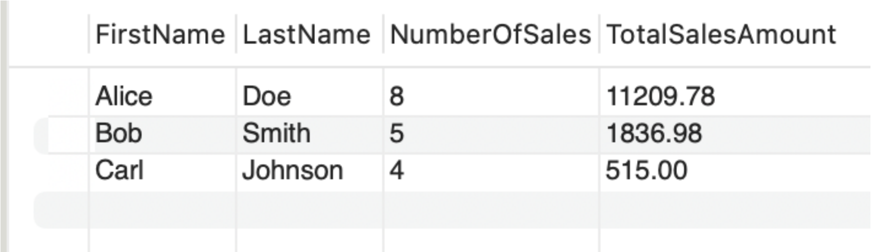
```
SELECT Clients.ClientName, SUM(SaleDetails.TotalAmount) AS TotalSpent
FROM Clients
JOIN Sales ON Clients.ClientID = Sales.ClientID
JOIN SaleDetails ON Sales.SaleID = SaleDetails.SaleID
GROUP BY Clients.ClientName;
```



ClientName	TotalSpent
Market Plaza	11209.78
Quick Buy	1836.98
Everyday Goods	40.00
Family Mart	475.00

-- Get Sales Representatives' Performance

```
SELECT SalesRepresentatives.FirstName, SalesRepresentatives.LastName, COUNT(Sales.SaleID)
AS NumberOfSales, SUM(SaleDetails.TotalAmount) AS TotalSalesAmount
FROM SalesRepresentatives
JOIN Sales ON Sales.RepresentativeID = SalesRepresentatives.RepresentativeID

JOIN SaleDetails ON Sales.SaleID = SaleDetails.SaleID
GROUP BY Sales.RepresentativeID;
```

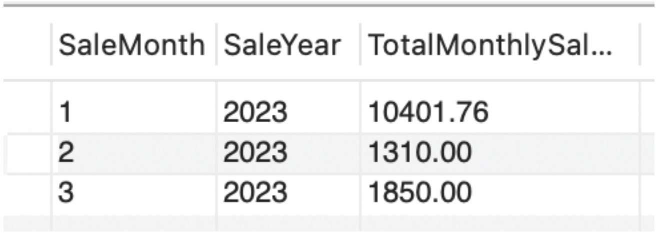


FirstName	LastName	NumberOfSales	TotalSalesAmount
Alice	Doe	8	11209.78
Bob	Smith	5	1836.98
Carl	Johnson	4	515.00

-- Sales Trends Over Time

```
SELECT MONTH(SaleDate) AS SaleMonth, YEAR(SaleDate) AS SaleYear,
SUM(SaleDetails.TotalAmount) AS TotalMonthlySales
FROM Sales
JOIN SaleDetails ON Sales.SaleID = SaleDetails.SaleID
GROUP BY YEAR(SaleDate), MONTH(SaleDate)
```

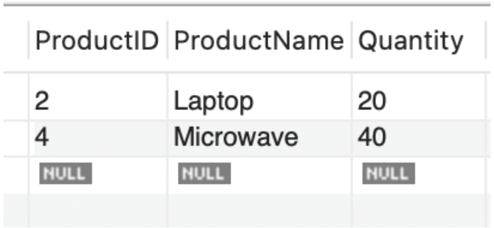
```
ORDER BY YEAR(SaleDate), MONTH(SaleDate);
```



SaleMonth	SaleYear	TotalMonthlySal...
1	2023	10401.76
2	2023	1310.00
3	2023	1850.00

-- List Products Below Minimum Stock

```
SELECT ProductID, ProductName, Quantity
FROM Products
WHERE Quantity < 50;
```

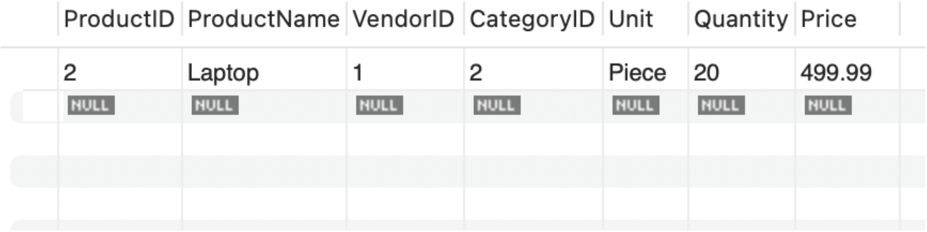


ProductID	ProductName	Quantity
2	Laptop	20
4	Microwave	40
NULL	NULL	NULL

Trigger:

-- Automatically Update Inventory After a Sale

```
DELIMITER $$
CREATE TRIGGER UpdateInventory AFTER INSERT ON SaleDetails
FOR EACH ROW
BEGIN
   UPDATE Products
   SET Quantity = Quantity - NEW.Quantity
   WHERE ProductID = NEW.ProductID;
END$$
DELIMITER ;

SELECT * FROM wholesalestore.products WHERE ProductID= 2;



INSERT INTO SaleDetails (SaleID, ProductID, Quantity, TotalAmount)
VALUES (1, 2, 10, 4999.90);

SELECT * FROM wholesalestore.saledetails;
SELECT * FROM wholesalestore.products WHERE ProductID= 2;
```

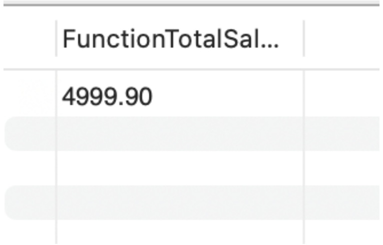


ProductID	ProductName	VendorID	CategoryID	Unit	Quantity	Price
2	Laptop	1	2	Piece	20	499.99
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Previously, the product quantity was 30 here, after inserting 10 Quantities in SaleDetails, Product table is updated with Quantity as 20.

-- Function: to find totalsales for each product

```
DELIMITER $$
CREATE FUNCTION TotalSalesByProduct_(product_id INT)
RETURNS DECIMAL(10,2)
READS SQL DATA
BEGIN
   DECLARE total_sales DECIMAL(10,2);
   SELECT SUM(TotalAmount) INTO total_sales
   FROM SaleDetails
   WHERE ProductID = product_id;
   RETURN total_sales;
END$$
DELIMITER ;

SELECT TotalSalesByProduct_(2) AS FunctionTotalSales;
```

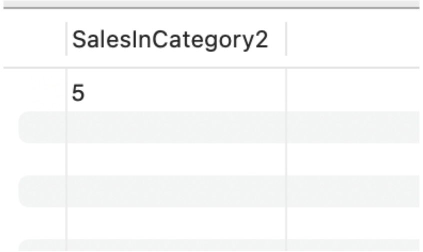


FunctionTotalSal...
4999.90

--Function : to find number sales for each category

```
DELIMITER $$
CREATE FUNCTION SalesCountByCategory(category_id INT)
RETURNS INT
READS SQL DATA
BEGIN
```

```
   DECLARE sales_count INT;
   SELECT COUNT(DISTINCT Sales.SaleID) INTO sales_count
   FROM Sales
   JOIN SaleDetails ON Sales.SaleID = SaleDetails.SaleID
   JOIN Products ON SaleDetails.ProductID = Products.ProductID
   WHERE Products.CategoryID = category_id;
   RETURN sales_count;
END$$
DELIMITER ;
SELECT SalesCountByCategory(2) AS SalesInCategory2;
```



SalesInCategory2
5

Conclusion:

In conclusion, the implementation of a comprehensive database solution for the Wholesale Store has significantly streamlined various aspects of its operations, enhancing overall efficiency and customer satisfaction. The database effectively addresses critical issues such as inventory management, order processing, and client and vendor management, which were previously hindered by manual and semi-automated systems. Moreover, the integration of promotional and payment modules within the database has allowed for smoother transaction processes and more targeted marketing strategies, ultimately leading to increased sales and customer loyalty.

The successful deployment of this database solution underscores the importance of a well-organized data management system in the context of wholesale distribution. It not only supports day-to-day operations but also provides a robust framework for scaling the business amidst an ever-evolving retail landscape.

Looking forward, it is recommended that Wholesale Store continues to enhance its database system by integrating advanced analytics and decision-support tools. These additions will aid in deriving deeper insights from the data, facilitating proactive decision-making, and further refining operational strategies. Such advancements will ensure that Wholesale Store remains competitive and well-positioned to capitalize on new market opportunities.