PharmaSys

Group #3

Fernanda Carrillo Escarcega *Likitha Bommasani*

Manisha Radhakrishna *Miguel Bravo Martinez Del Valle*

Overview

The purpose of this project is to design an efficient Database Management system to handle various aspects of pharmacy operations including inventory management, financial tracking, customer service, and workflow management. The system is intended to provide data integrity, accuracy, and consistency throughout the pharmacy's activities.

For the completion of the project, we adequately split tasks. All the group members participated in the topic selection and framing of the Business Rules. Fernanda and Manisha focused on Conceptual Data Modeling the ER/EER Diagram, Transforming the ER/EER Diagram to the Relational Model and Normalization. Miguel and Likitha Focused on creating DDL SQL Statements, Deploying the Database to the AWS cloud, and App-Database Interaction. We all pitched in to create a simple dashboard. However, we each helped our teammates in terms of troubleshooting and confirming results.

STEP 1: Topic/Domain Selection

Introduction:

The requirement for a robust and efficient Database Management System (DBMS) becomes critical in the ever-changing pharmaceutical industry landscape. This project attempts to address the need by developing a sophisticated database management system (DBMS) customized to handle the many features of pharmacy operations. PharmaSys, the envisioned system, is strategically designed to change inventory management, financial tracking, customer service, and workflow management in the pharmaceutical area.

Advantages and contributions:

PharmaSys has a transformative impact on pharmacy operations due to several distinct benefits and contributions. First and foremost, the system maintains a commitment to data integrity by ensuring the dependability and trustworthiness of information through the adoption of strong business rules. Its second key benefit is operational efficiency, which is achieved by streamlining activities and decreasing manual labor, reducing the possibility of errors. PharmaSys is a pioneer in inventory management, enabling pharmacies to proactively optimize stock levels while successfully decreasing associated expenses. Furthermore, by accurately tracking sales, expenses, and revenue, the system promotes informed financial decision-making. Its capabilities extend to customer service, where efficient record management translates into individualized interactions for improved client experiences. Finally, PharmaSys improves workflow management by coordinating.

Use Cases:

PharmaSys finds application in various use cases within the pharmacy domain:

- **Inventory Management:** Real-time tracking of product quantities and optimizing stock levels.
- **Financial Management/Sales Tracking:** Accurate tracking of sales, expenses, and revenue for informed financial decision-making.
- **Customer Service:** Efficient management of customer records and prescription details for personalized customer interactions.
- **Workflow Management:** Organizing employee roles and optimizing the handling of prescriptions for a streamlined workflow.

STEP 2: Conceptual Data Modeling and Database Design

Business Rules:

This Database Management System focuses on capturing end-to-end activities within a pharmacy, spanning from supplier interactions to consumer transactions. A transaction is a prerequisite for inclusion in the system.

Product Rules:

- Each product in the system is associated with its unique ID and its name
- The quantity in stock for a product should never be negative.
- The price per unit for a product should be greater than or equal to zero.
- Each product should also have its expiry date, to understand its validity on-shelf.

Supplier Rules:

- Each pharmacy supplier should have a unique ID and its name.
- The phone numbers and addresses of the suppliers are recorded in the system

Associate Entity: SupplierProduct Rules:

- A supplier can supply multiple products, and a product can be supplied by one or more suppliers.
- The product quantity supplied for each transaction should be greater than zero and is recorded as an associative entity in the system between supplier and product.

Employee Rules:

- All employees are assigned an Employee ID, which is kept along with the Employee's Name.
- The end date of employment for an employee should be '9999-12-31' or after the start date.
- An Employee's Role can be either a Manager or a Salesperson.
- An employee's salary should be a non-negative value.
- For Sales person, hourly wages and target weekly work hours are stored.
- For managers, an annual wage is stored which is considered as their base salary in the system.

Prescription Rules:

- Each prescription has a unique prescription ID and a list of products in the prescription.
- The prescription quantity should be greater than zero.
- The system also stores the prescription details like the hospital name and date when it was prescribed.
- The prescription date should not be NULL and must have a hospital seal for validating them.

Customer Rules:

- Every customer purchasing in the pharmacy should have their records stored in the system.
- Details like their Name, SSN, phone number, Address, and gender are stored for future reference.
- The SSN of a customer should be unique.
- Gender should be limited to predefined values (e.g., Male, Female, Other, PreferNotToSay).

Transaction Rules:

- Every transaction in the pharmacy is recorded in the system as a separate value.
- It records the customer and employee details involved with the transaction.
- A transaction must cater to at least one product that a hospital prescribes.
- The total quantity sold in a transaction should be greater than zero.
- The transaction date must be on or after the prescription date.
- The total price in a transaction should be the sum of the prices of the individual products.

Associate Entity: ProductTransaction Rules:

- The system stores the relation between a product and a transaction as multiple products can be purchased in a single transaction.
- The quantity purchased for each transaction should be greater than zero to be recorded in the system.

General Rules:

- All foreign keys reference existing primary keys.
- Dates (e.g., TransactionDate, ExpiryDate, PrescriptionDate, StartDate, EndDate) should be valid and in a consistent format.
- Phone numbers should adhere to a specific format.
- Access to sensitive information (e.g., SSN) should be restricted based on user roles.
- Changes to employee roles or salaries should be logged for auditing purposes.
- The address format for suppliers and customers should follow a standardized structure.
- Product names should follow a consistent naming convention.

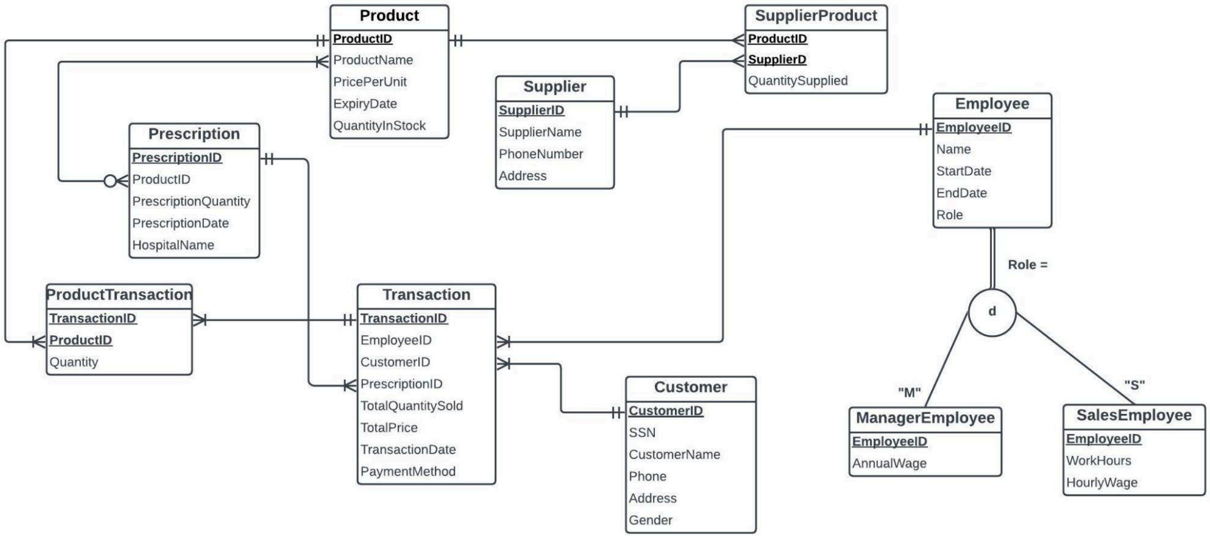
These rules help ensure data integrity, accuracy, and consistency within the database. Adjust them based on specific requirements and constraints in our application.

ER/EER Diagram:

Link: PharmaSys Lucichart

PharmaSys

Conceptual Data Modeling: ER/EER Diagram:



Relational Diagram:

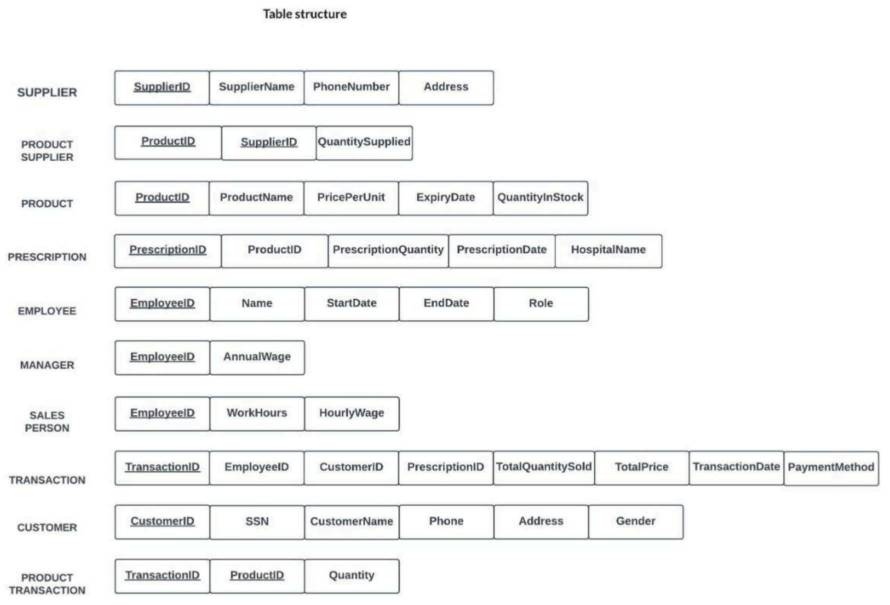
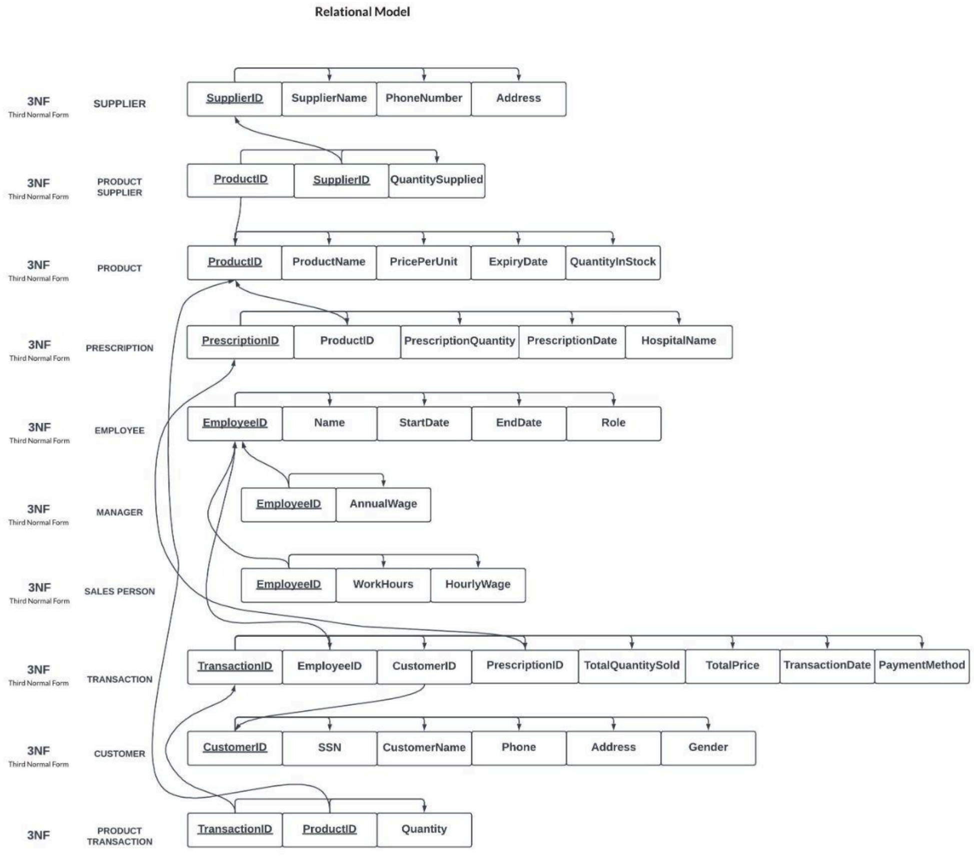


Table structure

SUPPLIER	SupplierID	SupplierName	PhoneNumber	Address				
PRODUCT SUPPLIER	ProductID	SupplierID	QuantitySupplied					
PRODUCT	ProductID	ProductName	PricePerUnit	ExpiryDate	QuantityInStock			
PRESCRIPTION	PrescriptionID	ProductID	PrescriptionQuantity	PrescriptionDate	HospitalName			
EMPLOYEE	EmployeeID	Name	StartDate	EndDate	Role			
MANAGER	EmployeeID	AnnualWage						
SALES PERSON	EmployeeID	WorkHours	HourlyWage					
TRANSACTION	TransactionID	EmployeeID	CustomerID	PrescriptionID	TotalQuantitySold	TotalPrice	TransactionDate	PaymentMethod
CUSTOMER	CustomerID	SSN	CustomerName	Phone	Address	Gender		
PRODUCT TRANSACTION	TransactionID	ProductID	Quantity					



Relational Model

STEP 3: Database Implementation

SQL Command to Create Tables in Database:

Creating tables in a pharmacy system is a foundational step in organizing and managing data efficiently. This process involves defining the structure of the database by specifying the table's fields and their data types. Once designed, the SQL command 'CREATE TABLE' is employed to instantiate the table. This methodical approach ensures a structured and organized database, laying the groundwork for the systematic storage and retrieval of crucial pharmacy-related information. Below are the SQL commands that were used to create tables for the database. The SQL commands used to create the tables for the database can also be found in an attached text file [PharmaSys_Create.txt]

- **Create Suppliers Table**

```
DROP TABLE IF EXISTS Supplier;
CREATE TABLE IF NOT EXISTS Supplier (
    SupplierID INT PRIMARY KEY,
    SupplierName VARCHAR(100),
    PhoneNumber VARCHAR(15),
    Address VARCHAR(255));
```

- **Create Product Table**

```
DROP TABLE IF EXISTS Product;
CREATE TABLE  IF NOT EXISTS  Product (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100) NOT NULL,
    PricePerUnit DECIMAL(10, 2),
    ExpiryDate DATE,
    QuantityInStock INT
);
```

- **Create SupplierProduct Table**

```
DROP TABLE IF EXISTS SupplierProduct;
CREATE TABLE IF NOT EXISTS SupplierProduct (
    ProductID INT,
    SupplierID INT,
    QuantitySupplied INT,
    PRIMARY KEY (ProductID, SupplierID),
      CONSTRAINT  SupplierProduct_FK1  FOREIGN  KEY  (SupplierID)
REFERENCES Supplier(SupplierID),
            CONSTRAINT  SupplierProduct_FK2  FOREIGN  KEY  (ProductID)
REFERENCES Product(ProductID)
      );
```

- **Create Employee Table**

```
      DROP TABLE IF EXISTS Employee;
      CREATE TABLE IF NOT EXISTS Employee (
          EmployeeID INT PRIMARY KEY,
          Name VARCHAR(100) NOT NULL,
          StartDate DATE,
          EndDate DATE DEFAULT '9999-12-31',
          Role VARCHAR(50)
      );
      ALTER TABLE Employee
      ADD CONSTRAINT check_end_dates CHECK (EndDate > StartDate);
```

- **Create Manager_Employee Table**

```
      DROP TABLE IF EXISTS ManagerEmployee;
      CREATE TABLE  IF NOT EXISTS  ManagerEmployee (
          EmployeeID INT PRIMARY KEY,
          AnnualWage DECIMAL(10, 2)
      );
```

- **Create Sales_Employee Table**

```
      DROP TABLE IF EXISTS SalesEmployee;
      CREATE TABLE  IF NOT EXISTS  SalesEmployee (
          EmployeeID INT PRIMARY KEY,
          workhours DECIMAL(10, 2),
          hourlywage  DECIMAL(10, 2)
      );
```

- **Create Customer Table**

```
      DROP TABLE IF EXISTS Customer;
      CREATE TABLE  IF NOT EXISTS  Customer (
          CustomerID INT PRIMARY KEY,
          SSN VARCHAR(11) NOT NULL,
          CustomerName VARCHAR(100),
          Phone VARCHAR(15),
          Address VARCHAR(255),
          Gender VARCHAR(50) DEFAULT 'PreferNotToSay');
```

- **Create Prescription Table**

```
     DROP TABLE IF EXISTS Prescription;
     CREATE TABLE  IF NOT EXISTS  Prescription (
         PrescriptionID INT PRIMARY KEY,
         ProductID INT NOT NULL,
         PrescriptionQuantity INT,
         PrescriptionDate DATE  NOT NULL,
         HospitalName VARCHAR(100),
         FOREIGN KEY (ProductID) REFERENCES Product(ProductID)
     );
```

- **Create Transaction Table**

```
     DROP TABLE IF EXISTS Transaction;
     CREATE TABLE  IF NOT EXISTS  Transaction (
         TransactionID INT PRIMARY KEY,
         EmployeeID INT  NOT NULL,
         CustomerID INT  NOT NULL,
         TotalQuantitySold INT,
         PrescriptionID INT,
         TotalPrice DECIMAL(10, 2),
         TransactionDate DATE  NOT NULL,
         PaymentMethod VARCHAR(50),
          FOREIGN       KEY       (PrescriptionID)       REFERENCES
Prescription(PrescriptionID),
         FOREIGN KEY (EmployeeID) REFERENCES Employee(EmployeeID),
         FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
     );
```

- **Create ProductTransaction Table**

```
     DROP TABLE IF EXISTS ProductTransaction;
     CREATE TABLE  IF NOT EXISTS  ProductTransaction (
         TransactionID INT,
         ProductID INT,
         Quantity INT  NOT NULL,
         PRIMARY KEY (TransactionID,ProductID),
         FOREIGN KEY (ProductID) REFERENCES Product(ProductID),
                    FOREIGN    KEY    (TransactionID)    REFERENCES
Transaction(TransactionID)
     );
```

SQL Commands to Insert Data Into Database:

For managing pharmaceutical information, we are manually inserting data into a pharmacy system's tables. This process involves accessing the database, navigating to the relevant table, and using SQL commands to insert new records. Each entry requires careful manual input of data, ensuring accuracy and alignment with predefined data types. Executing the query finalizes the insertion, contributing to the comprehensive management of pharmaceutical information within the system. Below are the SQL commands that were used to insert tables for the database. The SQL commands used to insert the tables for the database can also be found in an attached text file [PharmaSys_Insert.txt].

- **Insert data into Supplier table**

```
      INSERT INTO Supplier (SupplierID, SupplierName, PhoneNumber,
Address)
      VALUES
      (301, 'MedicoSource Inc.', '555-1234', '123 Main Street'),
      (302, 'HealthLink Suppliers', '555-5678', '456 Oak Avenue'),
      (303, 'MediCare Solutions Co.', '555-9876', '789 Maple
Street'),
      (304, 'Apex Medical Distributors', '555-4321', '890 Pine
Avenue'),
      (305, 'MedTech Innovations', '555-8765', '901 Elm Street'),
      (306, 'VitalCare Suppliers', '555-3456', '234 Cedar Avenue'),
      (307, 'MedSupply Partners', '555-6543', '567 Oak Street'),
      (308, 'WellnessPro Enterprises', '555-2109', '890 Maple
Avenue'),
      (309, 'MediGoods Global', '555-7890', '123 Cherry Street'),
      (310, 'MedEquip Solutions', '555-1234', '456 Pine Street');
```

- **Insert data into Product table**

```
      INSERT INTO Product (ProductID, ProductName, PricePerUnit,
ExpiryDate, QuantityInStock)
      VALUES
      (101, 'Painkiller X', 2.50, '2023-12-31', 200),
      (102, 'Cough Syrup Y', 4.00, '2024-06-30', 150),
      (103, 'Antibiotic Z', 6.00, '2023-10-15', 100),
      (104, 'Headache Relief', 3.75, '2023-11-30', 180),
      (105, 'Allergy Medicine', 5.25, '2024-03-31', 120),
      (106, 'Digestive Aid', 4.50, '2023-09-30', 90),
      (107, 'Cold Medicine', 3.25, '2024-01-15', 220),
      (108, 'Pain Relief Gel', 6.75, '2023-08-31', 75),
      (109, 'Eye Drops', 5.00, '2024-04-30', 140),
      (110, 'Vitamin Supplement', 7.00, '2023-07-31', 100);
```

- **Insert data into SupplierProduct table**

```
      INSERT INTO SupplierProduct (ProductID, SupplierID,
QuantitySupplied)
      VALUES
      (101, 301, 100),
      (102, 301, 150),
      (101, 302, 100),
      (103, 302, 120),
      (104, 302, 200),
      (105, 303, 90),
      (106, 304, 180),
      (107, 305, 160),
      (108, 305, 130),
      (109, 306, 110),
      (110, 307, 100),
      (105, 308, 90),
      (101, 309, 40),
      (109, 310, 120),
      (101, 307, 50),
      (101, 308, 50),
      (104, 309, 70),
      (105, 310, 90),
      (109, 307, 130),
      (110, 308, 140),
      (102, 309, 180),
      (102, 310, 20);
```

- **Insert data into Employee table**

```
      INSERT INTO Employee (EmployeeID, Name, StartDate, EndDate,
Role) VALUES
      (1, 'Miguel Bravo', '2023-01-01', '9999-12-31', 'Manager'),
      (2, 'Benjamin Williams', '2023-01-15', '9999-12-31',
'Manager'),
      (3, 'Sophia Brown', '2023-02-01', '9999-12-31', 'Sales
Person'),
      (4, 'Daniel Miller', '2023-02-05', '9999-12-31', 'Sales
Person'),
      (5, 'Olivia Davis', '2023-02-10', '9999-12-31', 'Sales
Person'),
      (6, 'Liam Wilson', '2023-02-15', '9999-12-31', 'Sales
Person'),
      (7, 'Ava Moore', '2023-03-01', '9999-12-31', 'Sales Person'),
      (8, 'Ethan Taylor', '2023-03-05', '2025-12-31', 'Sales
Person'),
      (9, 'Mia Anderson', '2023-03-10', '9999-12-31', 'Sales
Person'),
      (10, 'Alexander Martinez', '2023-03-15', '9999-12-31', 'Sales
Person');
```

- **Insert data into ManagerEmployee table**

```
      INSERT INTO ManagerEmployee (EmployeeID, AnnualWage) VALUES
      (1, 120000.00),
      (2, 90000.00);
```

- **Insert data into SalesEmployee table**

```
      INSERT INTO SalesEmployee (EmployeeID, workhours, hourlywage)
VALUES
      (3, 40.00, 20.00),
      (4, 35.00, 18.00),
      (5, 37.50, 19.50),
      (6, 40.00, 20.00),
      (7, 38.00, 19.00),
      (8, 35.50, 18.50),
      (9, 39.00, 19.75),
      (10, 37.00, 18.25);
```

- **Insert data into Prescription table**

```
      INSERT INTO Prescription (PrescriptionID, ProductID,
PrescriptionQuantity, PrescriptionDate, HospitalName) VALUES
      (1, 101, 2, '2023-01-10', 'General Hospital'),
      (2, 102, 1, '2023-02-05', 'City Medical Center'),
      (3, 103, 3, '2023-03-15', 'Community Health Clinic'),
      (4, 104, 2, '2023-04-20', 'City Hospital'),
      (5, 105, 1, '2023-05-12', 'University Medical Center'),
      (6, 106, 2, '2023-06-25', 'St. Marys Hospital'),
      (7, 107, 1, '2023-07-08', 'Community Health Clinic'),
      (8, 108, 3, '2023-08-18', 'General Hospital'),
      (9, 109, 2, '2023-09-22', 'City Medical Center'),
      (10, 110, 1, '2023-10-30', 'University Medical Center');
```

- **Insert data into Customer table**

```
      INSERT INTO Customer (CustomerID, SSN, CustomerName, Phone,
Address, Gender) VALUES
      (1, '123-45-6789', 'John Doe', '555-1234', '123 Main St',
'Male'),
      (2, '234-56-7890', 'Jane Smith', '555-5678', '456 Oak St',
'Female'),
      (3, '345-67-8901', 'Robert Johnson', '555-9876', '789 Pine
St', 'Male'),
      (4, '456-78-9012', 'Susan Williams', '555-4321', '987 Elm St',
'PreferNotToSay'),
      (5, '567-89-0123', 'Michael Davis', '555-8765', '654 Birch
St', 'Male'),
      (6, '678-90-1234', 'Emily Taylor', '555-2345', '321 Cedar St',
'Female'),
      (7, '789-01-2345', 'Daniel Brown', '555-6543', '876 Maple St',
'Male'),
      (8, '890-12-3456', 'Olivia White', '555-8765', '543 Walnut
St', 'Female'),
      (9, '901-23-4567', 'Matthew Miller', '555-2345', '210
Pineapple St', 'Male'),
      (10, '012-34-5678', 'Sophia Wilson', '555-7654', '789 Orange
St', 'Female');
```

- **Insert data into Transaction table**

```
      INSERT INTO Transaction (TransactionID, EmployeeID,
CustomerID, TotalQuantitySold, TotalPrice, TransactionDate,
PaymentMethod,PrescriptionID) VALUES
      (1,3, 1,  2, 40.00, '2023-01-15','Credit Card',8),
      (2,6, 2,  1, 25.00, '2023-02-10','Cash',6),
      (3,3, 3,  3, 90.00, '2023-03-20','Credit Card',3),
      (4,4, 4,  2, 50.00, '2023-04-25','Cash',4),
      (5,5, 5,  1, 30.00, '2023-05-18','Credit Card',1),
      (6,3, 6,  2, 60.00, '2023-06-30','Cash',9),
      (7,7, 7,  1, 35.00, '2023-07-15','Credit Card',2),
      (8,8, 8,  3, 105.00 ,'2023-08-25','Cash',5),
      (9,9, 9,  2, 70.00, '2023-09-28','Credit Card',7),
      (10,5, 10, 1, 40.00, '2023-10-31','Cash',10);
```

- **Insert data into Orders table**

```
      INSERT  INTO  ProductTransaction  (TransactionID,  ProductID,
Quantity) VALUES
      (1,101, 200),
      (1,102, 3),
      (1,103, 1),
      (1,110, 5),
      (2,105, 7),
      (3,107, 4),
      (3,103, 10),
      (4,109, 11),
      (5,110, 2),
      (6,103, 4),
      (6,102, 24),
      (7,107, 5),
      (8,110, 56),
      (8,106, 21),
      (9,102, 500),
      (9,105, 47),
      (10,106, 55);
```

Analytical Questions and SQL Queries:

Our data-centric analyses aim to provide actionable insights into supplier dynamics, employee salary structures, revenue distribution, and transaction details. By leveraging this data, the pharmacy can make informed decisions to enhance operational efficiency, optimize inventory, and improve overall business performance.

- **Supplier Analytics**: In Supplier Analytics, we address the challenge of understanding and managing supplier relationships. We gather and analyze data about each supplier and their associated products. The primary metric considered is the quantity supplied by each supplier, providing insights into supply patterns and helping optimize inventory management.

```
SELECT
      s.SupplierName,
      p.ProductName,
      sp.QuantitySupplied
```

```
FROM PharmaSys.SupplierProduct sp
LEFT JOIN PharmaSys.Supplier s
ON s.SupplierID=sp.SupplierID
LEFT JOIN PharmaSys.Product p
ON p.ProductID=sp.ProductID;
```

- **Employee Analytics**: Employee Analytics revolves around exploring the salary structure and correlations among different roles, such as Managers and Salespersons. By collecting weekly salary data, we aim to identify trends and distributions. The data includes information about employees' roles, allowing for a comprehensive overview of workforce-related financial patterns.

```
SELECT
      E.Name,
      E.Role,
      COALESCE(M.AnnualWage/(4*12),(S.workhours*S.hourlywage))as
WeeklySalary
FROM PharmaSys.Employee E
LEFT JOIN PharmaSys.ManagerEmployee M
ON E.EmployeeID=M.EmployeeID
LEFT JOIN PharmaSys.SalesEmployee S
ON E.EmployeeID=S.EmployeeID
GROUP BY
      E.Name,
      E.Role
ORDER BY
      E.Role,
      E.Name;
```

- **Revenue Analysis**: Revenue Analysis focuses on the distribution of revenue across all products in the pharmacy. Data is collected to understand how much revenue each product contributes to the overall income. This information is crucial for strategic decision-making, allowing the pharmacy to prioritize and optimize product offerings based on their revenue-generating potential.

```
SELECT
      p.ProductName,
      COALESCE(SUM(pt.Quantity * p.PricePerUnit), 0) AS TotalRevenue
FROM Product p
LEFT JOIN ProductTransaction pt
ON p.ProductID = pt.ProductID
GROUP BY
```

```
      p.ProductID,
      p.ProductName
ORDER BY
   TotalRevenue DESC;
```

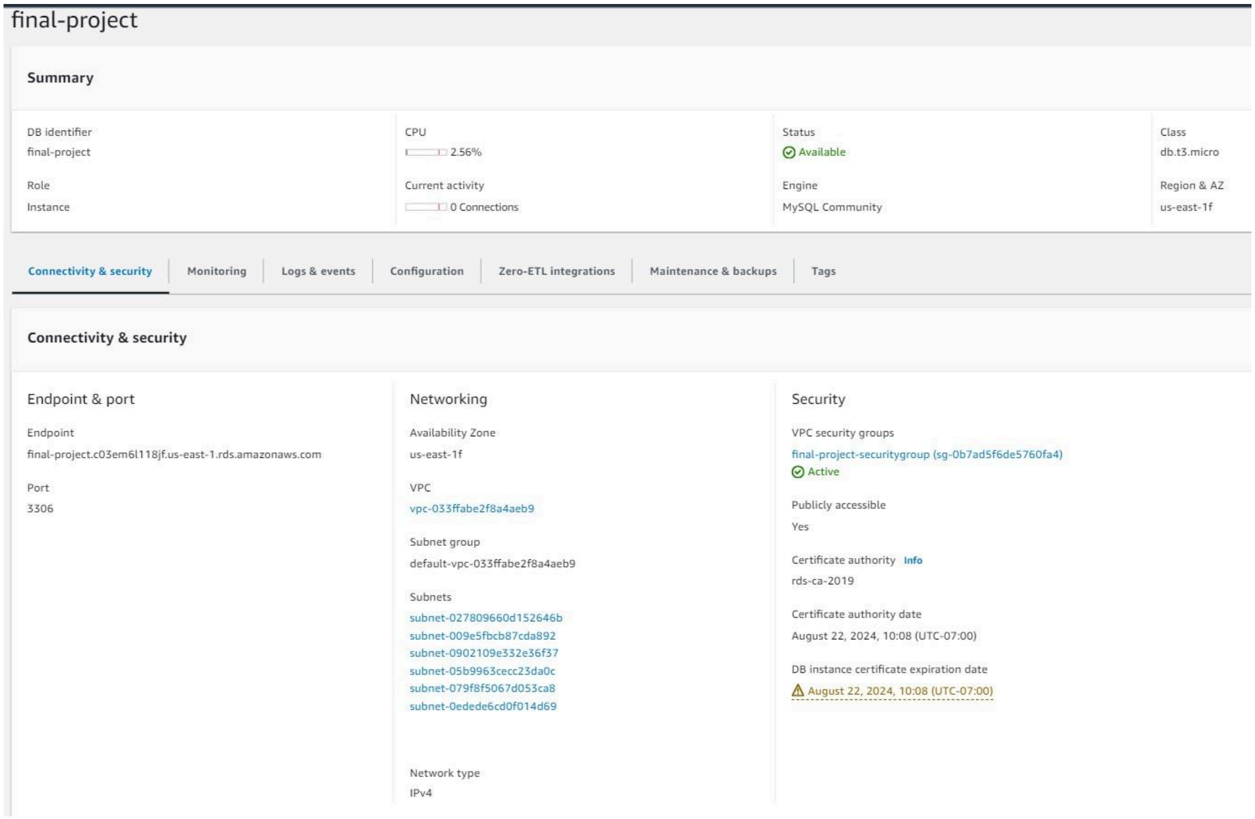
- **Transaction Analytics/Customer Sales Analytics**: In Transaction Analytics, we delve into the details of customer transactions and sales relationships. The data includes information about employees involved in the sales process, customer interactions, and the products sold. This analysis is essential for understanding customer preferences, improving sales strategies, and enhancing overall customer satisfaction.

```
SELECT
      t.TransactionId,
      c.CustomerName,
      e.Name,
      p.HospitalName,
      t.TotalQuantitySold,
      t.TotalPrice,
      t.TransactionDate,
      t.PaymentMethod
FROM PharmaSys.Transaction t
LEFT JOIN PharmaSys.Customer c
ON c.CustomerId=t.CustomerId
LEFT JOIN PharmaSys.Employee e
ON e.EmployeeId=t.EmployeeId
LEFT JOIN PharmaSys.Prescription p
ON p.PrescriptionID=t.PrescriptionId;
```

STEP 4: Deploy The Database on AWS Cloud

We followed the AWS tutorial for creating an environment to run our MySQL database.

```
host = 'final-project.c03em6l118jf.us-east-1.rds.amazonaws.com'
database = 'PharmaSys'
```



final-project

Summary

DB identifier
final-project

Role
Instance

CPU
2.56%

Current activity
0 Connections

Status
Available

Engine
MySQL Community

Class
db.t3.micro

Region & AZ
us-east-1f

Connectivity & security

Endpoint & port

Endpoint
final-project.c03em6l118jf.us-east-1.rds.amazonaws.com

Port
3306

Networking

Availability Zone
us-east-1f

VPC
vpc-033ffabe2f8a4aeb9

Subnet group
default-vpc-033ffabe2f8a4aeb9

Subnets
subnet-027809660d152646b
subnet-009e5fbcb87cda892
subnet-0902109e332e36f37
subnet-05b9963cecc23da0c
subnet-079f8f5067d053ca8
subnet-0edede6cd0f014d69

Network type
IPv4

Security

VPC security groups
final-project-securitygroup (sg-0b7ad5f6de5760fa4)
Active

Publicly accessible
Yes

Certificate authority Info
rds-ca-2019

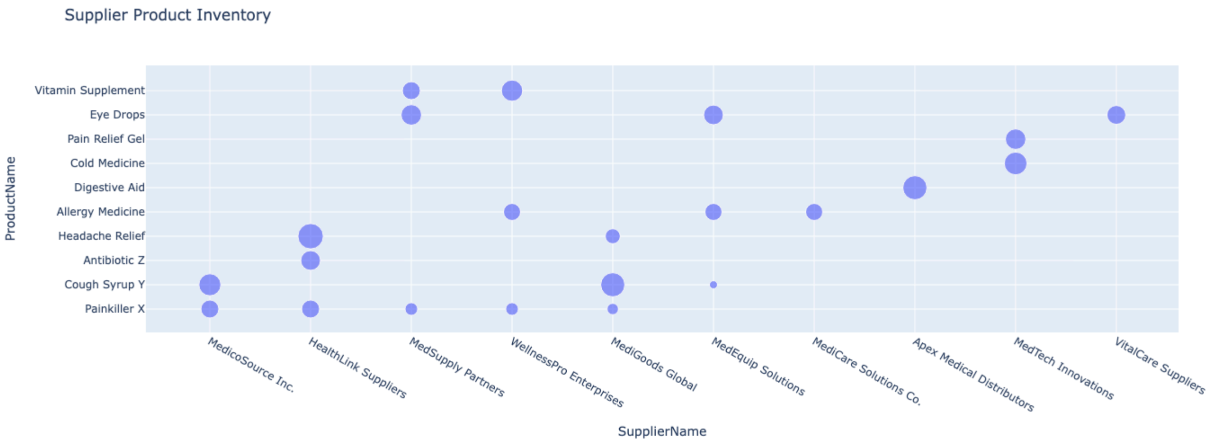
Certificate authority date
August 22, 2024, 10:08 (UTC-07:00)

DB instance certificate expiration date
⚠ August 22, 2024, 10:08 (UTC-07:00)

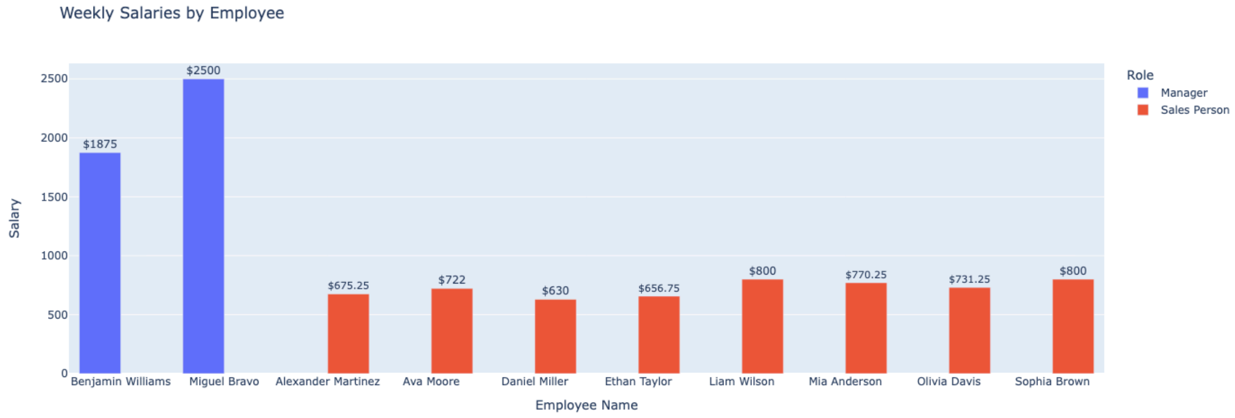
STEP 5: Enterprise (web) Database Dashboard

In our Enterprise Web Database Dashboard – a centralized hub designed to provide insights into critical aspects of our business operations. This intuitive and user-friendly platform is tailored to offer a comprehensive overview of our enterprise data, empowering users to make informed decisions. From tracking key performance indicators to analyzing trends and facilitating data-driven strategies, our dashboard serves as a powerful tool for optimizing efficiency and fostering a data-driven approach across the organization.

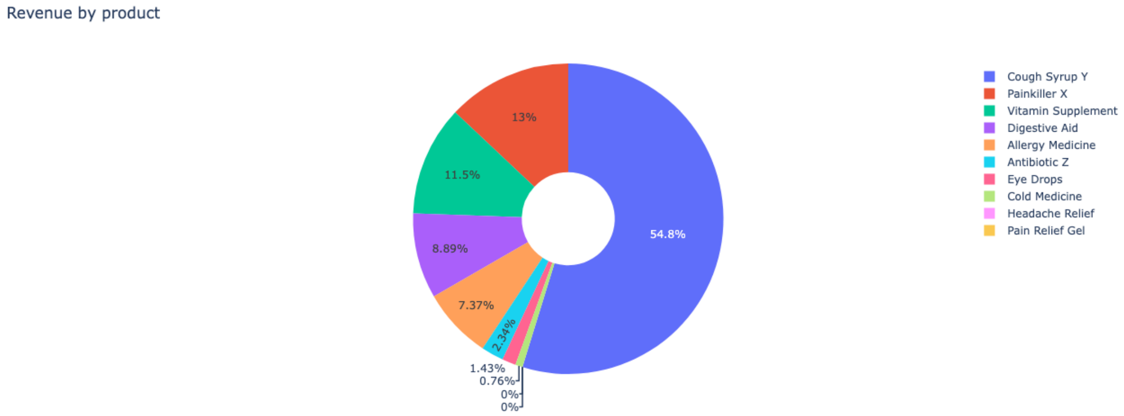
- **Supplier Analytics**: In tackling the first challenge, we delve into Supplier Analytics. This involves a comprehensive analysis that visualizes supplier and product details through a scatter plot. The size of each point on the scatter plot dynamically represents the quantity supplied by each supplier. This provides an insightful overview of the distribution and impact of various suppliers on the pharmacy's inventory.



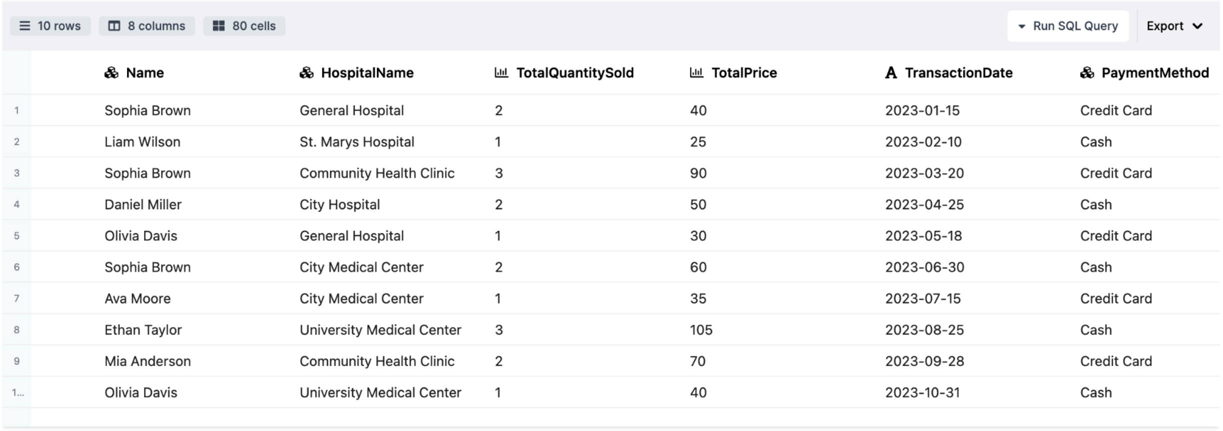
- **Employee Analytics**: Our second focus is on Employee Analytics, where we explore the weekly salary correlations among different employee roles such as Managers and Salespersons. This analysis aims to uncover patterns in the distribution of employee salaries every week. The use of color-coded bars allows for easy classification and comparison among different types of employees, offering a valuable perspective on workforce-related financial trends.



- **Revenue Analysis**: The third challenge addresses Revenue Analysis, aiming to understand the distribution of revenue across all products in the pharmacy. This is visualized through a pie chart that vividly illustrates the proportion of revenue contributed by each product. The pie chart serves as an effective tool for quick insights into the product-wise revenue distribution, aiding strategic decision-making.

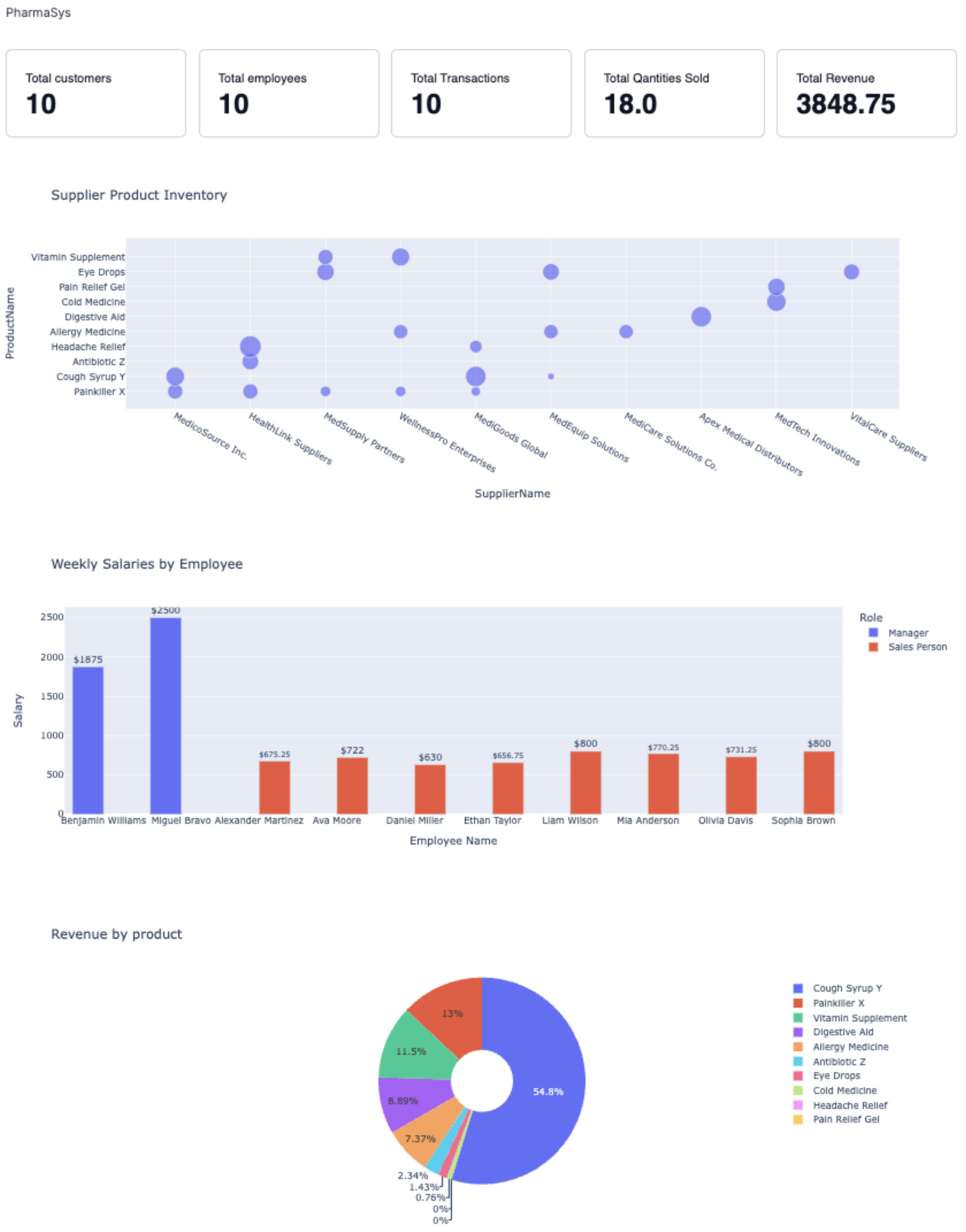


- **Transaction Analytics/Customer Sales Analytics**: Lastly, we tackle Transaction Analytics or Customer Sales Analytics, presenting a detailed view of the transaction table. This table provides valuable insights into the relationships among employees, customers, and sales. By visually representing these interactions, we aim to enhance understanding and optimize the sales process for better customer engagement and satisfaction.



	Name	HospitalName	TotalQuantitySold	TotalPrice	TransactionDate	PaymentMethod
1	Sophia Brown	General Hospital	2	40	2023-01-15	Credit Card
2	Liam Wilson	St. Marys Hospital	1	25	2023-02-10	Cash
3	Sophia Brown	Community Health Clinic	3	90	2023-03-20	Credit Card
4	Daniel Miller	City Hospital	2	50	2023-04-25	Cash
5	Olivia Davis	General Hospital	1	30	2023-05-18	Credit Card
6	Sophia Brown	City Medical Center	2	60	2023-06-30	Cash
7	Ava Moore	City Medical Center	1	35	2023-07-15	Credit Card
8	Ethan Taylor	University Medical Center	3	105	2023-08-25	Cash
9	Mia Anderson	Community Health Clinic	2	70	2023-09-28	Credit Card
1...	Olivia Davis	University Medical Center	1	40	2023-10-31	Cash

The analytical group dashboard can be found under the following link:
PharmaSys Dashboard



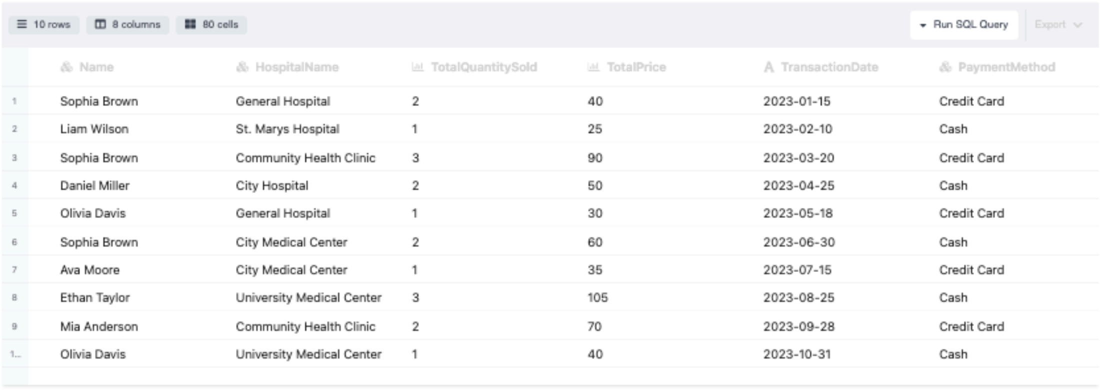
PharmaSys

Total customers	Total employees	Total Transactions	Total Qantities Sold	Total Revenue
10	10	10	18.0	3848.75

Supplier Product Inventory

Weekly Salaries by Employee

Revenue by product



	Name	HospitalName	TotalQuantitySold	TotalPrice	TransactionDate	PaymentMethod
1	Sophia Brown	General Hospital	2	40	2023-01-15	Credit Card
2	Liam Wilson	St. Marys Hospital	1	25	2023-02-10	Cash
3	Sophia Brown	Community Health Clinic	3	90	2023-03-20	Credit Card
4	Daniel Miller	City Hospital	2	50	2023-04-25	Cash
5	Olivia Davis	General Hospital	1	30	2023-05-18	Credit Card
6	Sophia Brown	City Medical Center	2	60	2023-06-30	Cash
7	Ava Moore	City Medical Center	1	35	2023-07-15	Credit Card
8	Ethan Taylor	University Medical Center	3	105	2023-08-25	Cash
9	Mia Anderson	Community Health Clinic	2	70	2023-09-28	Credit Card
1...	Olivia Davis	University Medical Center	1	40	2023-10-31	Cash

Conclusion

Our project aimed to develop an efficient Database Management System designed for comprehensive pharmacy operations, encompassing inventory management, financial tracking, customer service, and workflow management. The system's main goal is to ensure data integrity, accuracy, and consistency across all facets of the pharmacy's activities. This project not only enhanced our understanding of Database Management but also highlighted the significance of teamwork in achieving shared goals.

Thank you