**BIS1002 Data and Information Management**

**Assessment 3: Applied Project 2**

**Documentation Report**

1. **Start the Project**

* Created a new MS Word assignment document with a cover sheet.
* Set up the XAMPP environment.
* Created a new database in MySQL named **AppliedProject2**.

1. **Database Table Setup**

* We were given initial scripts to create the dimension tables. We updated them to:
* Declare **primary keys**.
* Set them as **AUTO INCREMENT**.
* Changed SaleDate in **DateDim** from TEXT → DATE.
* Made all fields **NOT NULL**.

Below is its code

CREATE TABLE IF NOT EXISTS CustomerDim (

CustomerKey INT AUTO\_INCREMENT PRIMARY KEY,

CustomerName VARCHAR(255) NOT NULL,

CustomerAddress VARCHAR(255) NOT NULL,

CustomerState VARCHAR(100) NOT NULL,

CustomerEmail VARCHAR(255) NOT NULL

);

CREATE TABLE IF NOT EXISTS DateDim (

DateKey INT AUTO\_INCREMENT PRIMARY KEY,

SaleDate DATE NOT NULL,

`Year` INT NOT NULL,

`Month` INT NOT NULL,

`Day` INT NOT NULL,

`Quarter` INT NOT NULL

);

CREATE TABLE IF NOT EXISTS ProductDim (

ProductKey INT AUTO\_INCREMENT PRIMARY KEY,

ProductName VARCHAR(255) NOT NULL,

Brand VARCHAR(100) NOT NULL,

UnitPrice DECIMAL(10,2) NOT NULL

);

CREATE TABLE IF NOT EXISTS ShopDim (

ShopKey INT AUTO\_INCREMENT PRIMARY KEY,

ShopState VARCHAR(100) NOT NULL,

ShopAddress VARCHAR(255) NOT NULL,

ShopEmail VARCHAR(255) NOT NULL

);

CREATE TABLE IF NOT EXISTS WarehouseDim (

WarehouseKey INT AUTO\_INCREMENT PRIMARY KEY,

WarehouseDivision VARCHAR(100) NOT NULL,

WarehouseRegion VARCHAR(100) NOT NULL

);

1. **Create the SaleFacts Table**

We created the fact table with foreign keys linking to all dimension tables.

Below is its code

CREATE TABLE IF NOT EXISTS SaleFacts (

ID INT AUTO\_INCREMENT PRIMARY KEY,

DateKey INT NOT NULL,

ShopKey INT NOT NULL,

CustomerKey INT NOT NULL,

ProductKey INT NOT NULL,

WarehouseKey INT NOT NULL,

SalesQuantity INT NOT NULL,

Discount TINYINT(1) NOT NULL CHECK (Discount IN (0,1)),

CONSTRAINT fk\_date FOREIGN KEY (DateKey) REFERENCES DateDim(DateKey),

CONSTRAINT fk\_shop FOREIGN KEY (ShopKey) REFERENCES ShopDim(ShopKey),

CONSTRAINT fk\_customer FOREIGN KEY (CustomerKey) REFERENCES CustomerDim(CustomerKey),

CONSTRAINT fk\_product FOREIGN KEY (ProductKey) REFERENCES ProductDim(ProductKey),

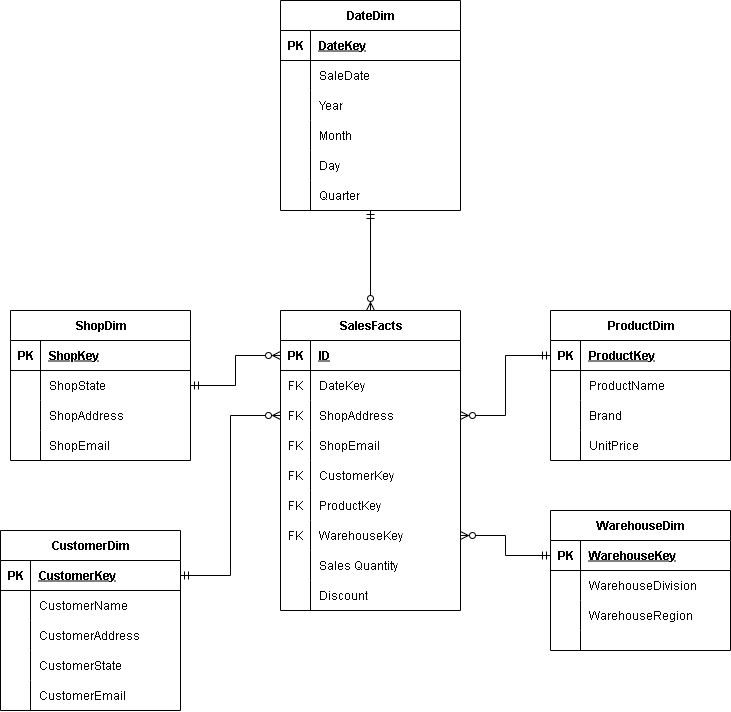
CONSTRAINT fk\_warehouse FOREIGN KEY (WarehouseKey) REFERENCES WarehouseDim(WarehouseKey)

);

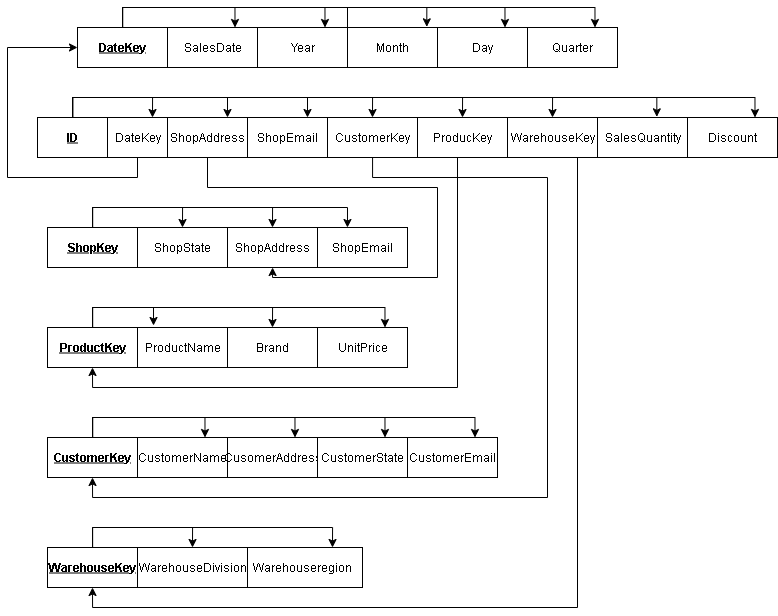
1. **Populating the database**

We populated the dimension tables using provided sample data (ap2\_t2\_2025\_data.txt).  
This ensured that queries on **SaleFacts** could successfully reference keys in each dimension.

1. **Diagrams**
2. **Conceptual Model**

****

1. **Relational Model**

****

1. **Data Insertion**

We inserted a new customer (using student details):

INSERT INTO CustomerDim (CustomerName, CustomerAddress, CustomerState, CustomerEmail)

VALUES ('Keith Wanguru', 'APIC Campus, Brisbane', 'Queensland', '202500378@student.apicollege.edu.au');

We then added a new sale referencing this customer (assuming supporting dimension data exists with IDs = 1):

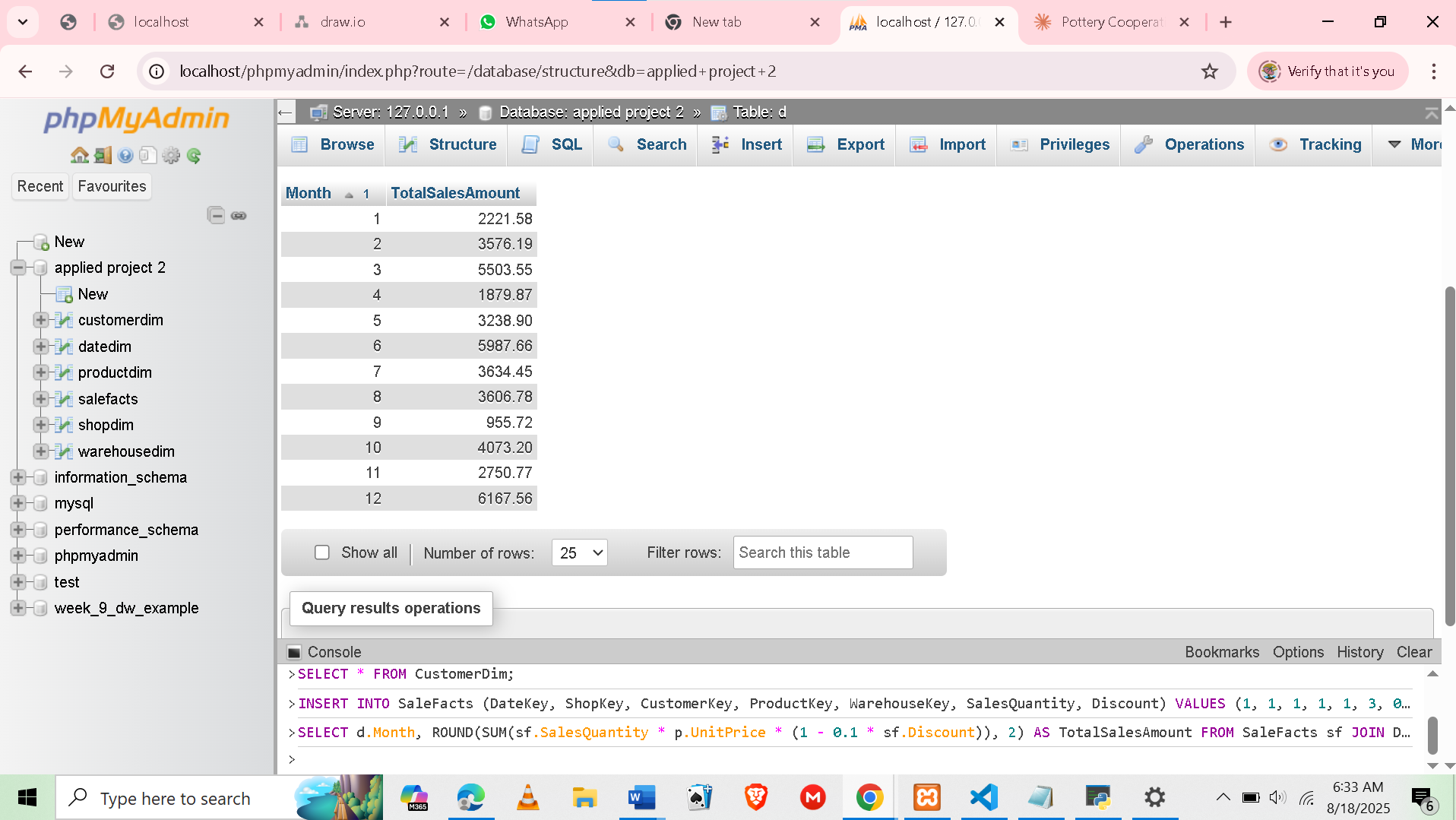
INSERT INTO SaleFacts (DateKey, ShopKey, CustomerKey, ProductKey, WarehouseKey, SalesQuantity, Discount)

VALUES (1, 1, 1, 1, 1, 3, 0);

1. **SQL Queries and Outputs**

We executed the following queries and included **screenshots from XAMPP** showing the query and result.

1. Total sales per month in 2017



SELECT

d.Month,

ROUND(SUM(sf.SalesQuantity \* p.UnitPrice \* (1 - 0.1 \* sf.Discount)), 2) AS TotalSalesAmount

FROM SaleFacts sf

JOIN DateDim d ON sf.DateKey = d.DateKey

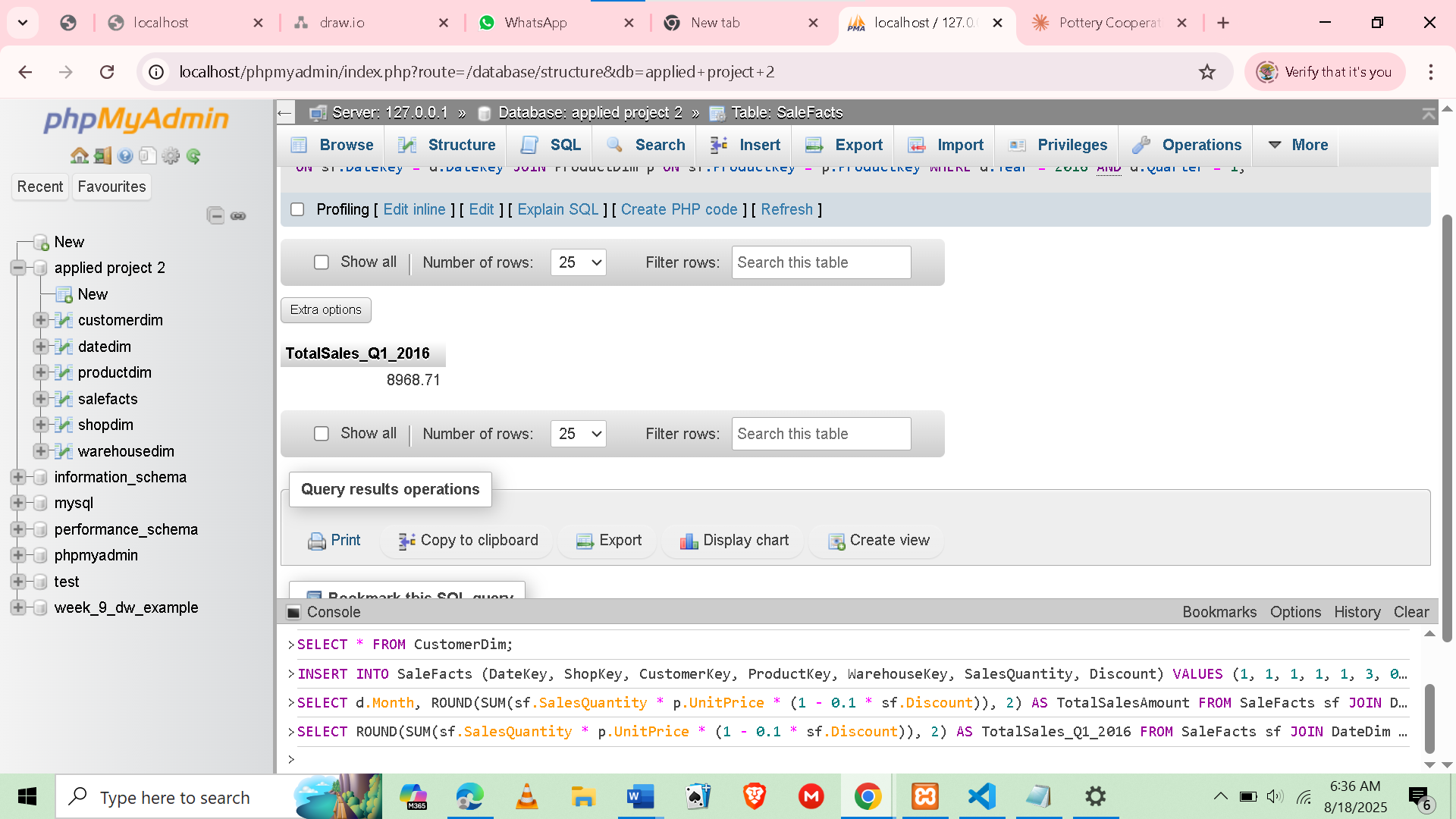
JOIN ProductDim p ON sf.ProductKey = p.ProductKey

WHERE d.Year = 2017

GROUP BY d.Month

ORDER BY d.Month;

1. Total sales in Q1 2016



SELECT

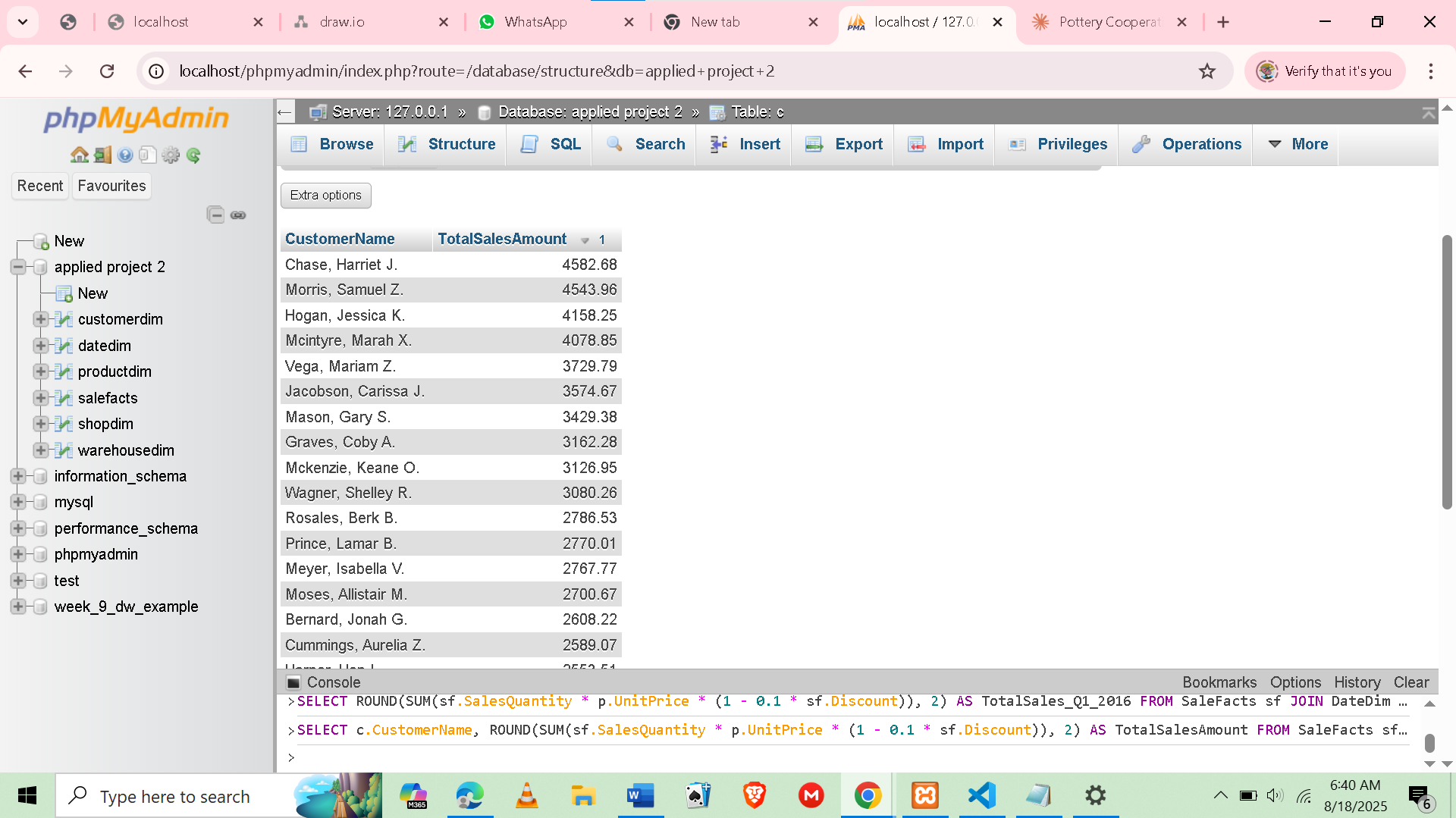
ROUND(SUM(sf.SalesQuantity \* p.UnitPrice \* (1 - 0.1 \* sf.Discount)), 2) AS TotalSales\_Q1\_2016

FROM SaleFacts sf

JOIN DateDim d ON sf.DateKey = d.DateKey

JOIN ProductDim p ON sf.ProductKey = p.ProductKey

WHERE d.Year = 2016 AND d.Quarter = 1;

1. Total sales per customer

SELECT

c.CustomerName,

ROUND(SUM(sf.SalesQuantity \* p.UnitPrice \* (1 - 0.1 \* sf.Discount)), 2) AS TotalSalesAmount

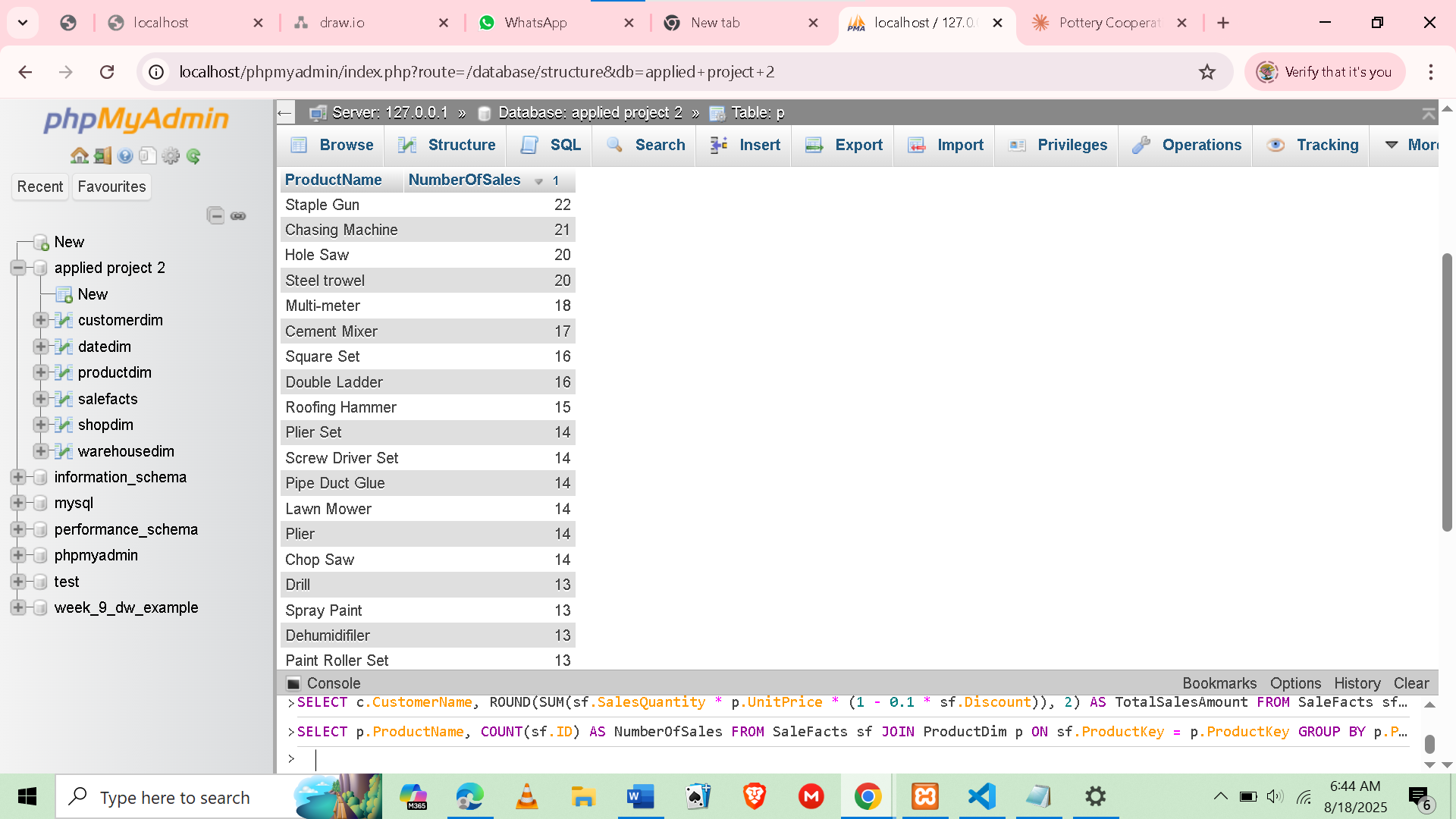
FROM SaleFacts sf

JOIN CustomerDim c ON sf.CustomerKey = c.CustomerKey

JOIN ProductDim p ON sf.ProductKey = p.ProductKey

GROUP BY c.CustomerName

ORDER BY TotalSalesAmount DESC;

1. Number of sales per product

SELECT

p.ProductName,

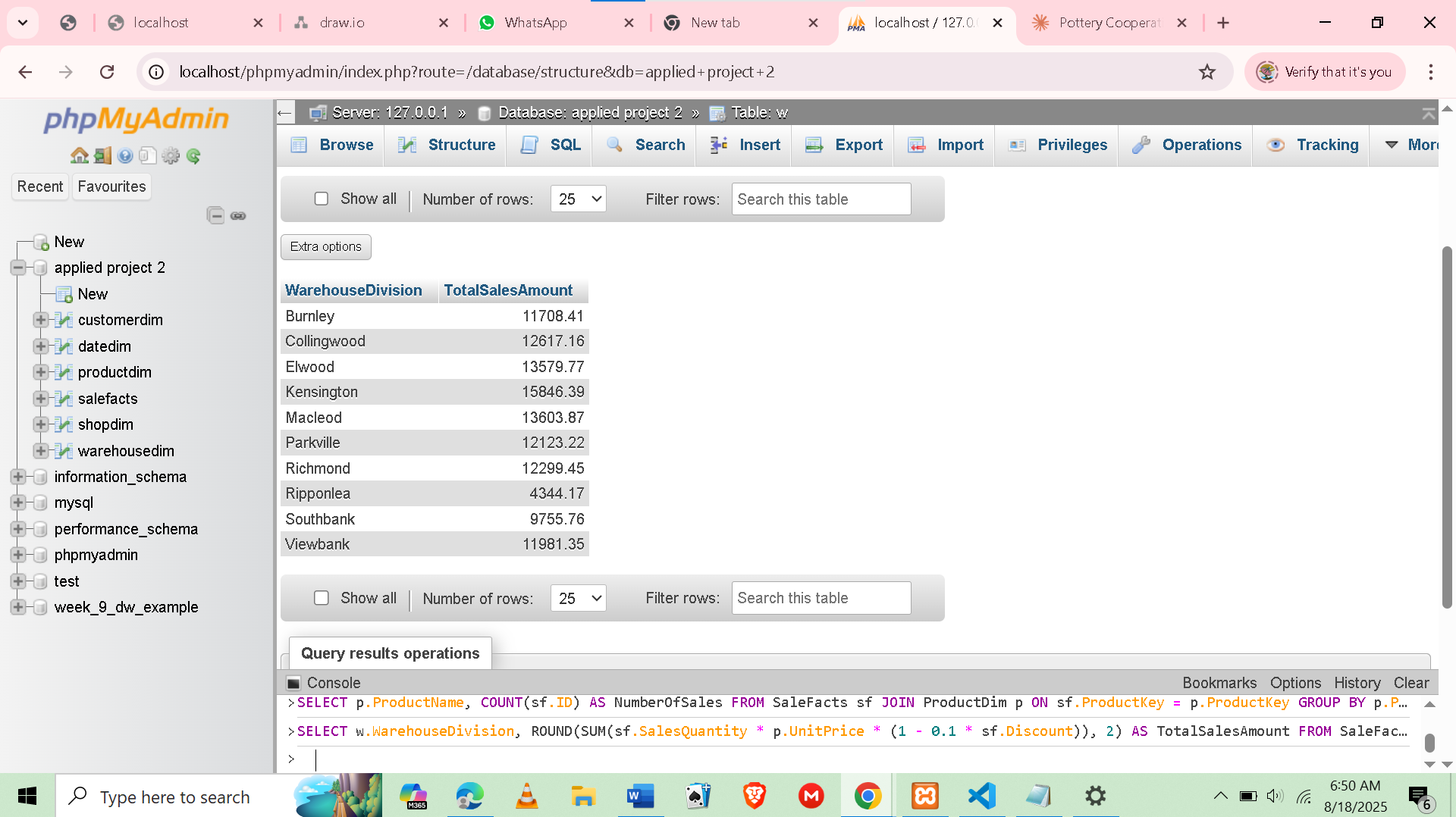
COUNT(sf.ID) AS NumberOfSales

FROM SaleFacts sf

JOIN ProductDim p ON sf.ProductKey = p.ProductKey

GROUP BY p.ProductName

ORDER BY NumberOfSales DESC;

1. Total sales per warehouse

SELECT

w.WarehouseDivision,

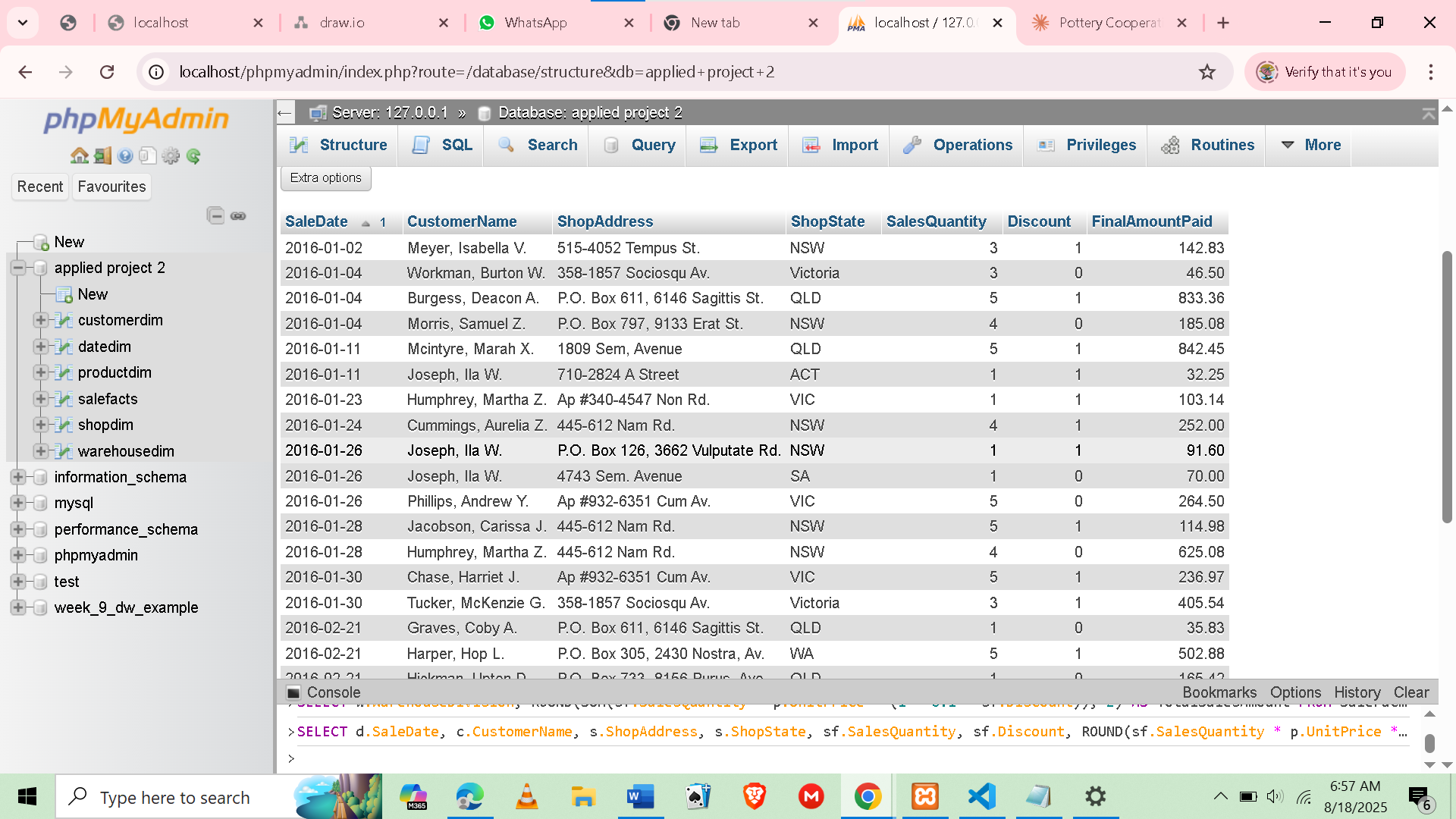
ROUND(SUM(sf.SalesQuantity \* p.UnitPrice \* (1 - 0.1 \* sf.Discount)), 2) AS TotalSalesAmount

FROM SaleFacts sf

JOIN WarehouseDim w ON sf.WarehouseKey = w.WarehouseKey

JOIN ProductDim p ON sf.ProductKey = p.ProductKey

GROUP BY w.WarehouseDivision;

1. Sales info for each transaction

SELECT

d.SaleDate,

c.CustomerName,

s.ShopAddress,

s.ShopState,

sf.SalesQuantity,

sf.Discount,

ROUND(sf.SalesQuantity \* p.UnitPrice \* (1 - 0.1 \* sf.Discount), 2) AS FinalAmountPaid

FROM SaleFacts sf

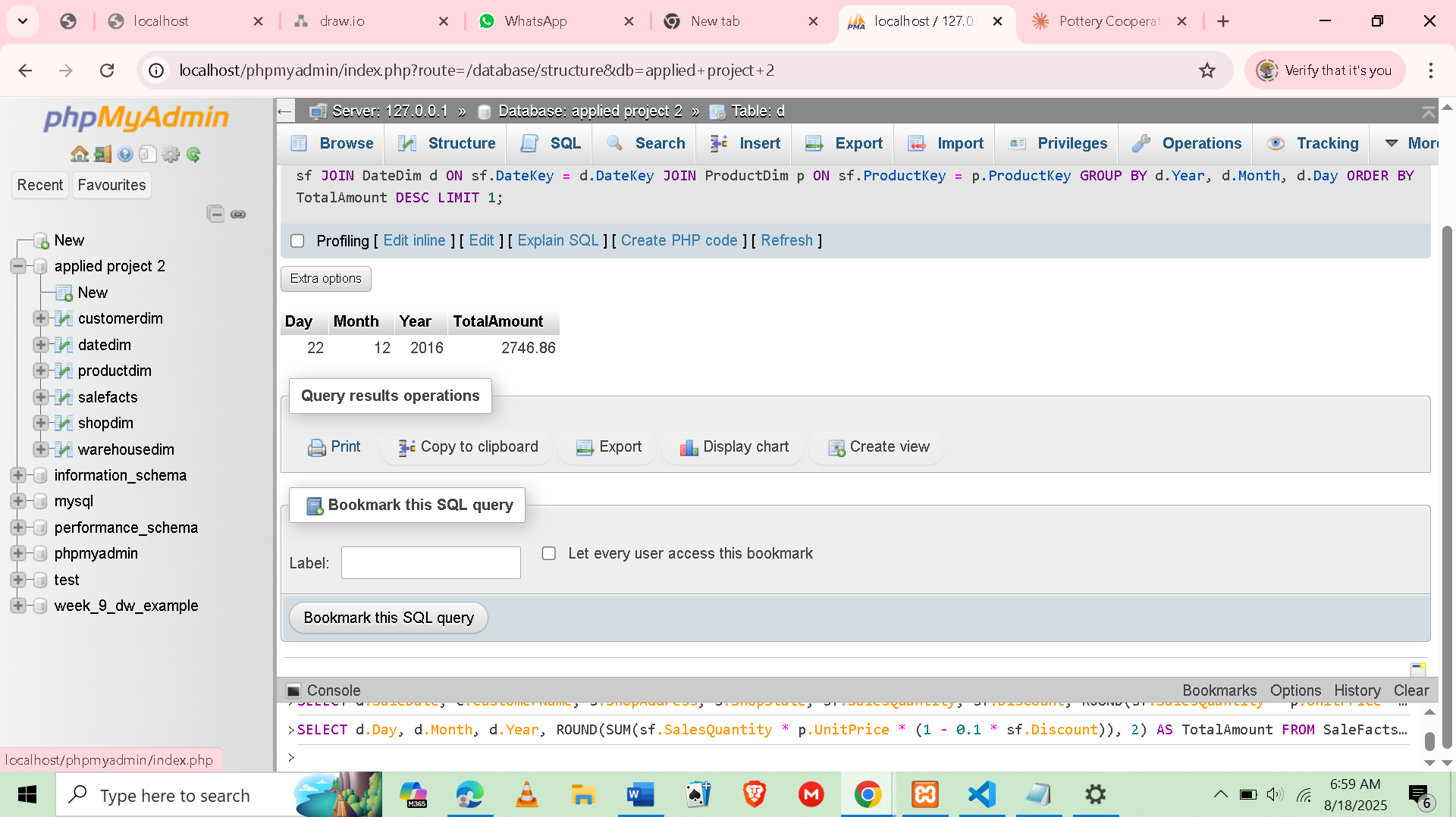
JOIN DateDim d ON sf.DateKey = d.DateKey

JOIN CustomerDim c ON sf.CustomerKey = c.CustomerKey

JOIN ShopDim s ON sf.ShopKey = s.ShopKey

JOIN ProductDim p ON sf.ProductKey = p.ProductKey

ORDER BY d.SaleDate;

1. Day with highest total amount

SELECT

d.Day,

d.Month,

d.Year,

ROUND(SUM(sf.SalesQuantity \* p.UnitPrice \* (1 - 0.1 \* sf.Discount)), 2) AS TotalAmount

FROM SaleFacts sf

JOIN DateDim d ON sf.DateKey = d.DateKey

JOIN ProductDim p ON sf.ProductKey = p.ProductKey

GROUP BY d.Year, d.Month, d.Day

ORDER BY TotalAmount DESC

LIMIT 1;

1. **AI Use Declaration**

In this project, we used generative AI tools (ChatGPT) strictly for **debugging SQL queries** and clarifying certain database concepts. The AI was not used to generate the full assessment but only to assist in identifying syntax issues (e.g., correcting AUTOINCREMENT to AUTO\_INCREMENT in MySQL) and improving the formatting of SQL queries. I take full responsibility for the content of this assessment.

1. **Appendix:**

The following resources were referenced during the preparation of this project:

* ECA Library Catalogue and Databases:  
  <https://eca.libguides.com/az/databases?preview=4b60ca002bf80c92f2ec9d7a8332fda2>
* Subject Guide: Information Technology:  
  <https://eca.libguides.com/c.php?g=962328&p=6993297>
* AI Use Guidelines (ECA Library and Learning Support):  
  <https://eca.libguides.com/AI>