| **Name:** | **James Lewis** |
| --- | --- |
| **Roll No:** | **32** |
| **Class/Sem:** | TE/V |
| **Experiment No.:** | 2 |
| **Title:** | Implementation of Dimension and Fact tables and perform OLAP operations. |
| **Date of Performance:** | 29/07/25 |
| **Date of Submission:** |  |
| **Marks:** |  |
| **Sign of Faculty:** |  |

**Aim:** Implementation of Dimension and Fact tables and perform OLAP operations.

**Objective:** OLAP stands for Online Analytical Processing. The objective of OLAP is to analyze information from multiple database systems at the same time. It is based on multidimensional data model and allows the user to query on multi-dimensional data.

**Theory:**

* Online Analytical Processing Server (OLAP) is based on the multidimensional data model.
* The main aim of OLAP is to provide multidimensional analysis to the underlying data.

Following is the list of OLAP operations:

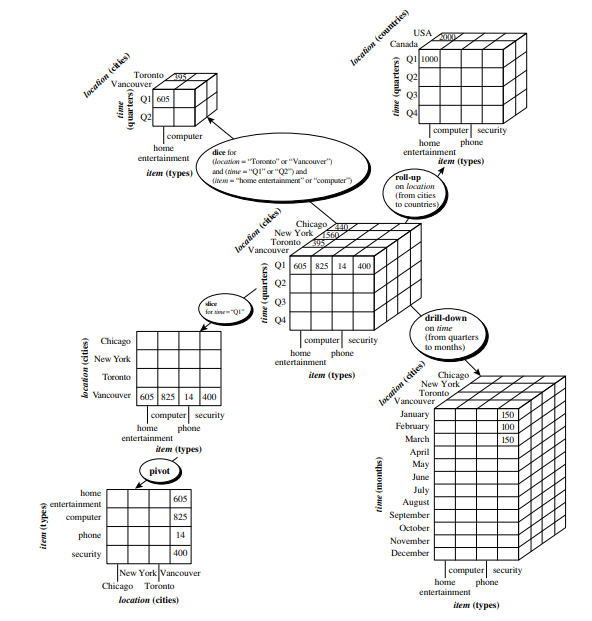
1. Roll-up
2. Drill-down
3. Slice
4. Dice
5. Pivot (rotate)

**Roll-up:**

* The roll-up operation (also called the drill-up operation) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension reduction.
* Figure 2.1 shows the result of a roll-up operation performed on the central cube by climbing up the concept hierarchy for location.
* This hierarchy was defined as the total order “street < city < province or state < country.”
* The roll-up operation aggregates the data by ascending the location hierarchy from the level of city to the level of country.
* In other words, rather than grouping the data by city, the resulting cube groups the data by country.

**Drill-down:**

* Drill-down is the reverse of roll-up. It navigates from less detailed data to more detailed data.
* Drill-down can be realized by either stepping down a concept hierarchy for a dimension or introducing additional dimensions.
* Figure 2.1 shows the result of a drill-down operation performed on the central cube by stepping down a concept hierarchy for time defined as “day < month < quarter < year.”
* Drill-down occurs by descending the time hierarchy from the level of quarter to the more detailed level of month.
* The resulting data cube details the total sales per month rather than summarizing them by quarter.

****

**Figure 2.1: Examples of typical OLAP operations on multidimensional data.**

**Slice:**

* The slice operation performs a selection on one dimension of the given cube, resulting in a sub cube.
* Figure 2.1 below shows a slice operation where the sales data are selected from the central cube for the dimension time using the criterion time = “Q1.”

**Dice:**

* The dice operation defines a sub cube by performing a selection on two or more dimensions.
* Figure 2.1 shows a dice operation on the central cube based on the following selection criteria that involve three dimensions: (location = “Toronto” or “Vancouver”) and (time = “Q1” or “Q2”) and (item = “home entertainment” or “computer”).

**Pivot:**

* Pivot (also called rotate) is a visualization operation that rotates the data axes in view to provide an alternative data presentation.
* Figure 2.1 shows a pivot operation where the item and location axes in a 2-D slice are rotated.

**Problem Statement:**

The problem is to design and implement a data warehousing solution for a bookstore that optimally organizes and manages its vast data, including sales, inventory, customer information, and more, to facilitate efficient reporting and analytics. This involves creating both a star schema and a snowflake schema to support various business intelligence and decision-making processes, while ensuring data accuracy, integrity, and performance

# Code:

**Create Database “book”:**

Create database book;

# Create Dimension Tables:

CREATE TABLE Book ( Book\_Id INT PRIMARY KEY,

Book\_Name VARCHAR(100), Book\_Genre VARCHAR(50), Book\_Author VARCHAR(100), Book\_Cost INT

);

CREATE TABLE Store ( Store\_Id INT PRIMARY KEY,

Store\_Name VARCHAR(100), Store\_Address VARCHAR(200), Store\_Pincode INT,

Store\_City VARCHAR(100)

);

CREATE TABLE Supplier ( Supplier\_Id INT PRIMARY KEY, Supplier\_Name VARCHAR(100), Supplier\_Address VARCHAR(200)

);

CREATE TABLE Users ( User\_Id INT PRIMARY KEY,

User\_Name VARCHAR(100), User\_Address VARCHAR(200), User\_Pincode INT,

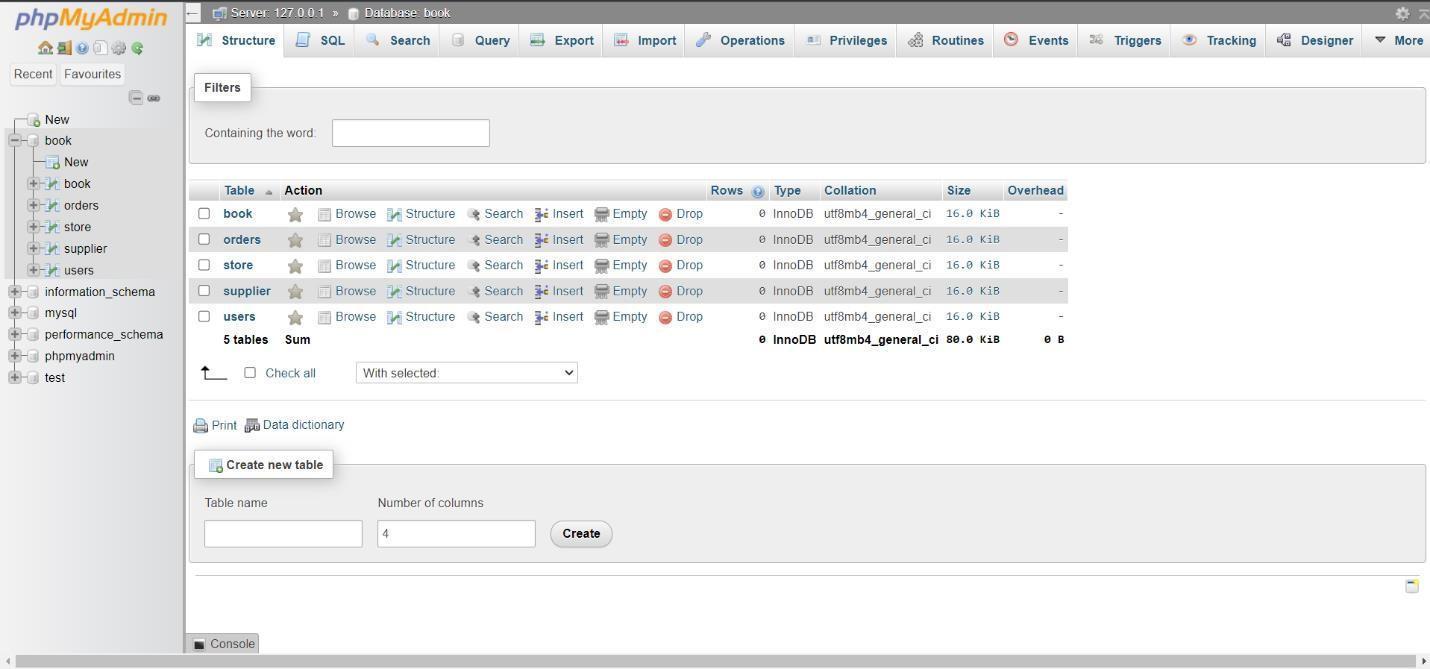
User\_City VARCHAR(100)

);

CREATE TABLE Orders ( Order\_Id INT PRIMARY KEY,

Order\_Cost INT, Order\_Quantity INT

);



# Create Fact Table:

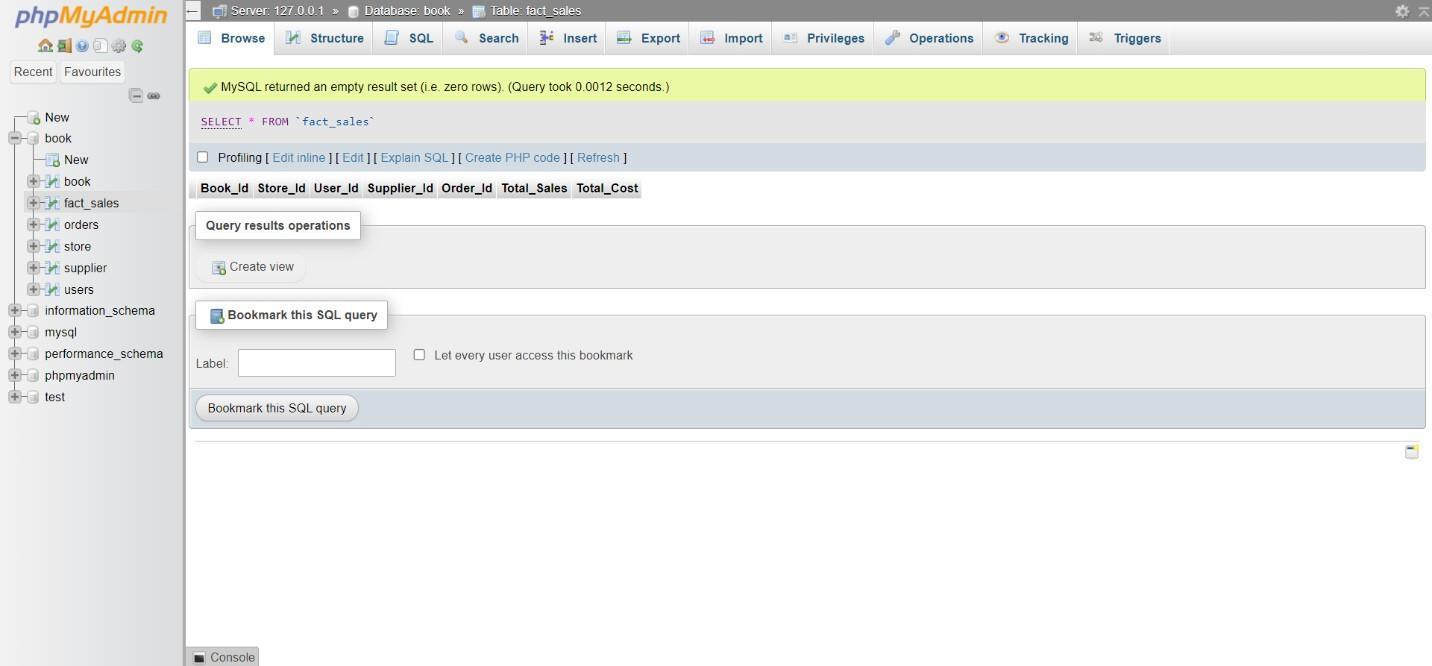
CREATE TABLE Fact\_Sales ( Book\_Id INT,

Store\_Id INT, User\_Id INT, Supplier\_Id INT, Order\_Id INT, Total\_Sales INT, Total\_Cost INT,

PRIMARY KEY (Book\_Id, Store\_Id, Order\_Id),

FOREIGN KEY (Book\_Id) REFERENCES Book(Book\_Id), FOREIGN KEY (Store\_Id) REFERENCES Store(Store\_Id), FOREIGN KEY (User\_Id) REFERENCES Users(User\_Id), FOREIGN KEY (Supplier\_Id) REFERENCES Supplier(Supplier\_Id), FOREIGN KEY (Order\_Id) REFERENCES Orders(Order\_Id)

);



# Inserting values in both dimension and fact tables:

INSERT INTO Book VALUES

(1, 'To Kill a Mockingbird', 'Fiction', 'Harper Lee', 250), (2, '1984', 'Fiction', 'George Orwell', 200),

(3, 'The Great Gatsby', 'Fiction', 'F. Scott Fitzgerald', 180);

INSERT INTO Store VALUES

(1, 'A Store', '123 Main St', 12345, 'New York'),

(2, 'B Store', '456 Oak Rd', 67890, 'Chicago'),

(3, 'C Store', '789 Elm St', 23456, 'Los Angeles');

INSERT INTO Supplier VALUES

(1, 'ABC Supplier', '111 Industry Ave'),

(2, 'XYZ Supplier', '222 Commerce St'),

(3, '123 Supplier', '333 Trade Blvd');

INSERT INTO Users VALUES

(1, 'John Doe', '100 1st St', 12345, 'New York'),

(2, 'Jane Smith', '200 2nd St', 23456, 'Los Angeles'),

(3, 'Bob Johnson', '300 3rd St', 34567, 'Chicago');

INSERT INTO Orders VALUES (1, 500, 10),

(2, 400, 8),

(3, 600, 12);

INSERT INTO Fact\_Sales VALUES (1, 1, 1, 1, 1, 5000, 2500),

(2, 2, 2, 2, 2, 4000, 1600),

(3, 3, 3, 3, 3, 7200, 2160);

# 

# Displaying the tables:

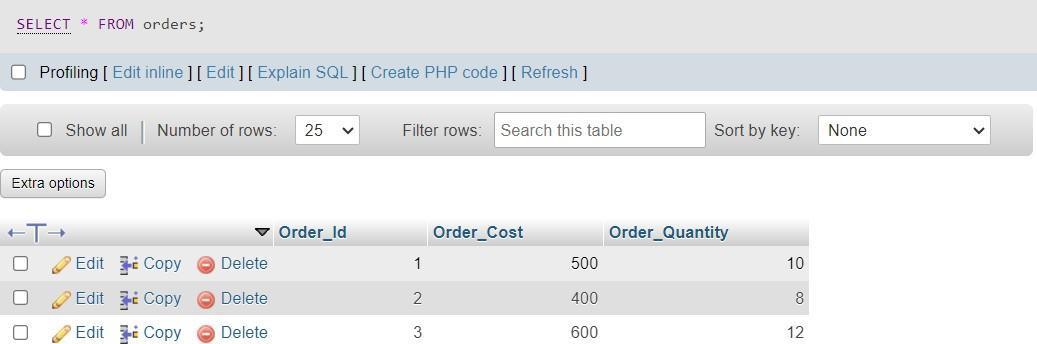
SELECT \* FROM book;



SELECT \* FROM fact\_sales;

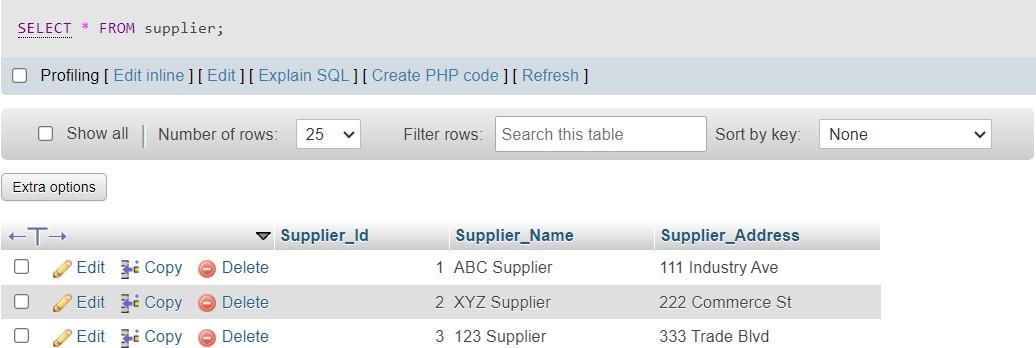


SELECT \* FROM orders;



SELECT \* FROM store;



SELECT \* FROM supplier;

SELECT \* FROM users;

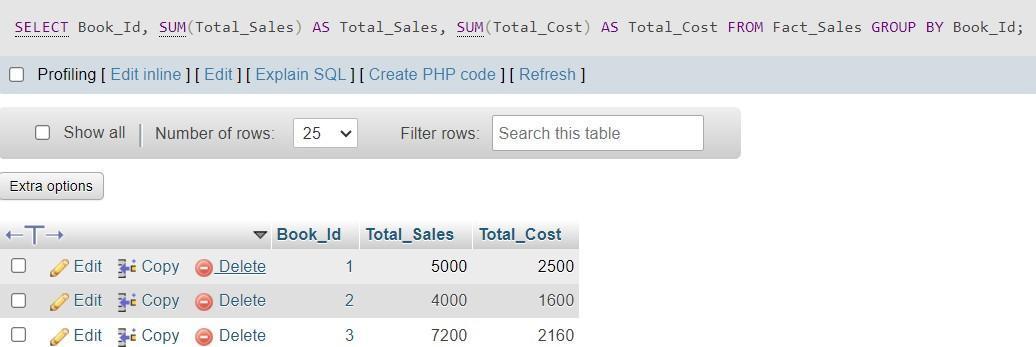


# Write SQL Queries for all the above OLAP operations:

**Rollup:**

SELECT Book\_Id, SUM(Total\_Sales) AS Total\_Sales, SUM(Total\_Cost) AS Total\_Cost FROM Fact\_Sales

GROUP BY Book\_Id;



# Drill Down:

SELECT Book\_Id, Store\_Id, SUM(Total\_Sales) AS Total\_Sales, SUM(Total\_Cost) AS Total\_Cost

FROM Fact\_Sales

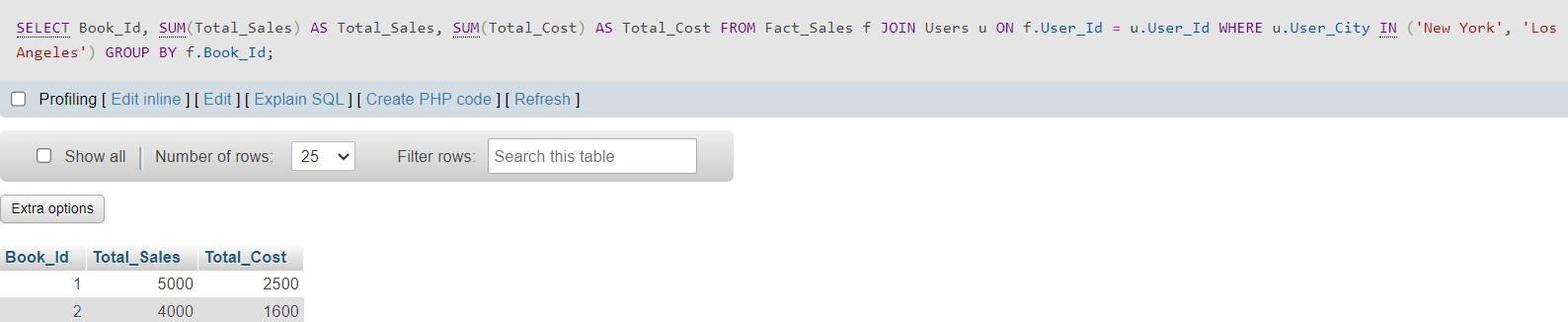
GROUP BY Book\_Id, Store\_Id;



# Slice:

SELECT Book\_Id, SUM(Total\_Sales) AS Total\_Sales, SUM(Total\_Cost) AS Total\_Cost FROM Fact\_Sales f

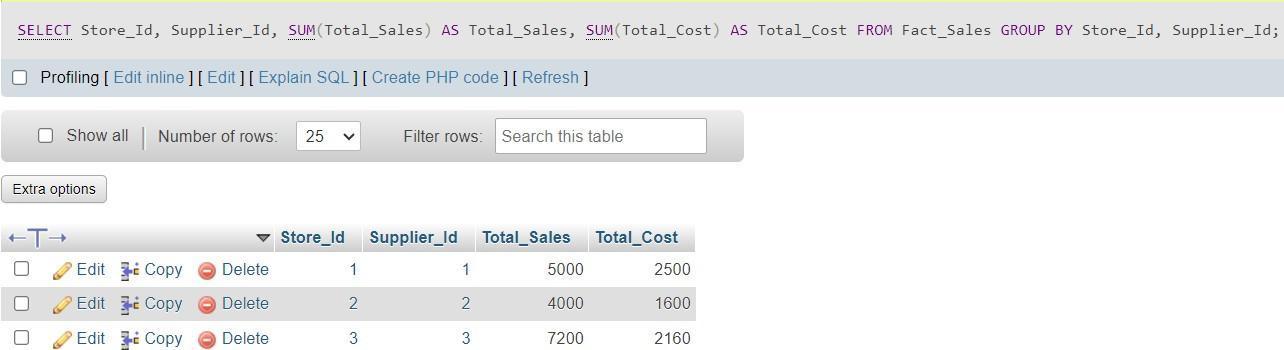
JOIN User u ON f.User\_Id = u.User\_Id

WHERE u.User\_City IN ('New York', 'Los Angeles') GROUP BY f.Book\_Id;

# Dice:

SELECT Store\_Id, Supplier\_Id, SUM(Total\_Sales) AS Total\_Sales, SUM(Total\_Cost) AS Total\_Cost

FROM Fact\_Sales

GROUP BY Store\_Id, Supplier\_Id;

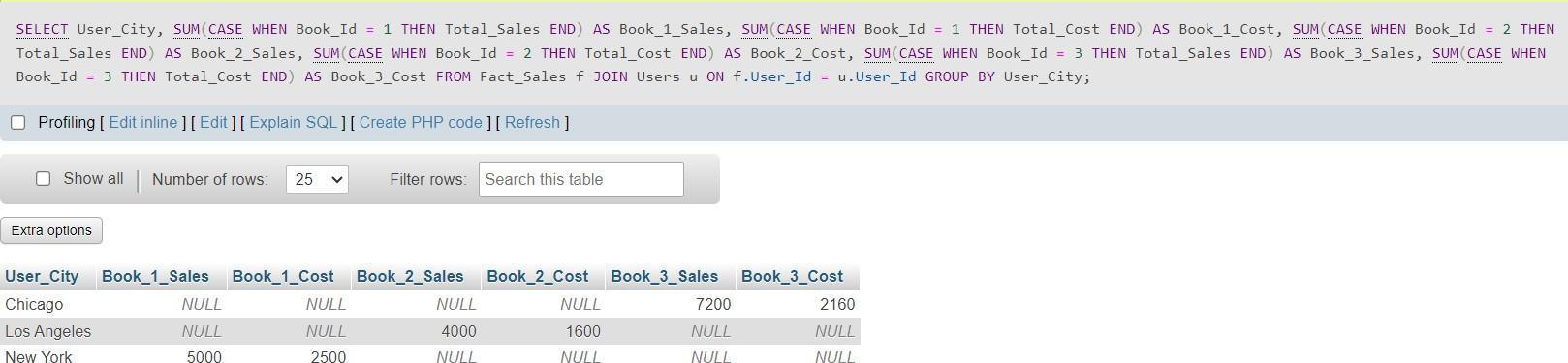
# Pivot:

SELECT

User\_City,

SUM(CASE WHEN Book\_Id = 1 THEN Total\_Sales END) AS Book\_1\_Sales, SUM(CASE WHEN Book\_Id = 1 THEN Total\_Cost END) AS Book\_1\_Cost, SUM(CASE WHEN Book\_Id = 2 THEN Total\_Sales END) AS Book\_2\_Sales, SUM(CASE WHEN Book\_Id = 2 THEN Total\_Cost END) AS Book\_2\_Cost, SUM(CASE WHEN Book\_Id = 3 THEN Total\_Sales END) AS Book\_3\_Sales, SUM(CASE WHEN Book\_Id = 3 THEN Total\_Cost END) AS Book\_3\_Cost FROM Fact\_Sales f

JOIN User u ON f.User\_Id = u.User\_Id GROUP BY User\_City;



# Conclusion:

Thus, we have learned implementation of Dimension and Fact tables and perform OLAP operations on database. OLAP stands for Online Analytical Processing. The objective of OLAP is to analyze information from multiple database systems at the same time.