Name:	
Roll No:	
Class/Sem:	TE/V
Experiment No.:	2
Title:	Implementation of Dimension tables and Fact tables and perform OLAP operations.
Date of Performance:	
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.

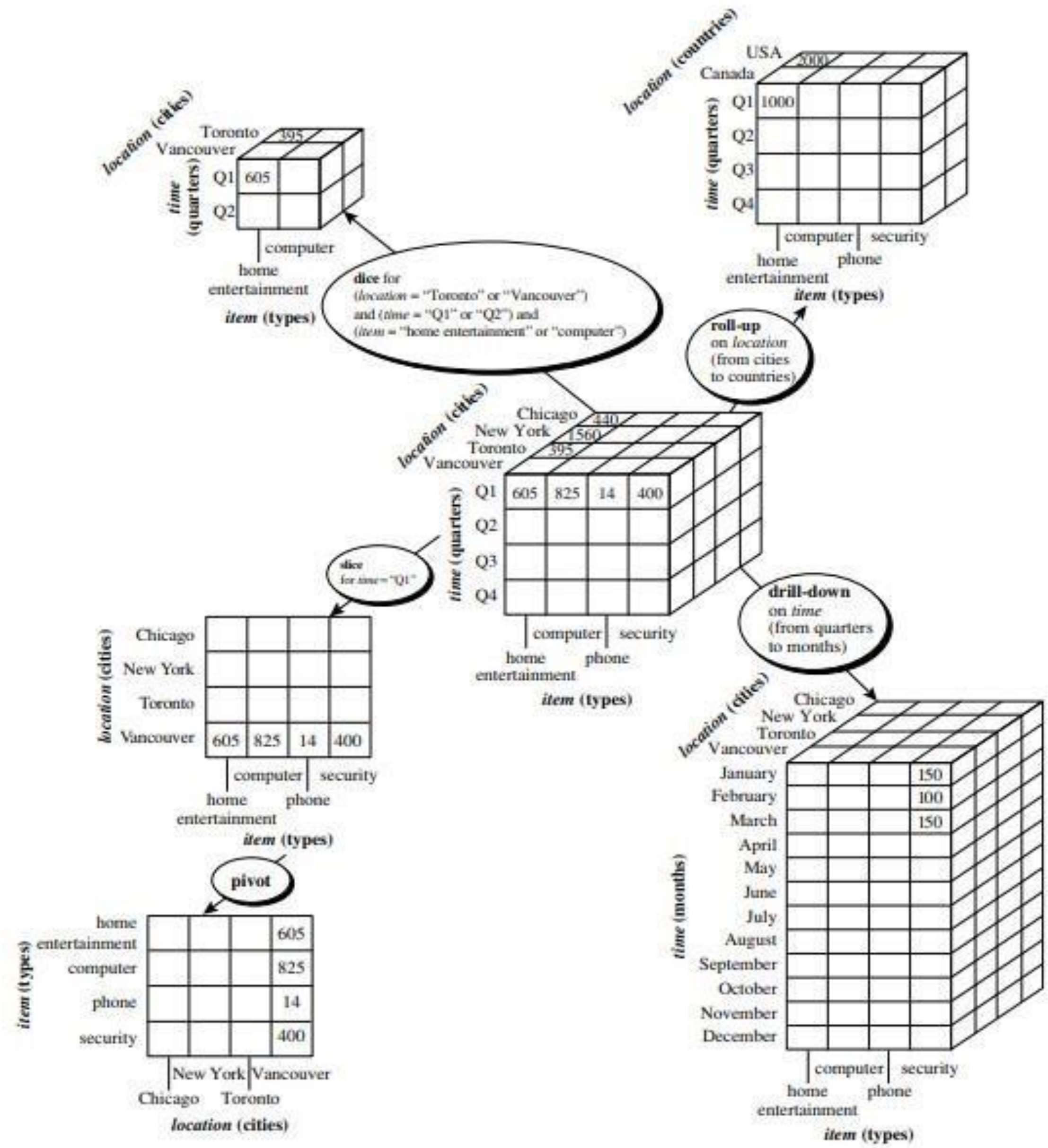


Figure 2.1: Examples of typical OLAP operations on multidimensional data.

Slice:

- The slice operation performs a selection on one dimension of the given cube, resulting in a subcube.
- 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 subcube 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:

Suppose that a data warehouse consists of four dimensions as patient,doctor,location and treatment.The two measures are count and fees,where fees is the treatment charge paid by the patient to the doctor on a weekly basis. Draw a star and snowflake schema diagram for the above data warehouse.

Output:

1. Creating the Dimension Tables

```
Create Database Hospital;


CREATE TABLE patient(
    Patient_id int(10) PRIMARY KEY,
    Patient_name varchar(50),
    Patient_age int(10),
    Patient_address varchar(250),
    Report_id int(10)
    );
CREATE TABLE doctor(
   Doctor_id int(10) PRIMARY KEY,
```

```
Doctor_name varchar(50),
Doctor_type varchar(50),
Doctor_age int(10),
Doctor_experience varchar(250),
1st_week int(10),
2nd_week int(10),
3rd_week int(10),
4th_week int(10)
);
    CREATE TABLE Location (
     location_id int(10) PRIMARY KEY,
     city varchar(50), state varchar(50),
     pincode int(10), country varchar(50)
    );

create table treatment(
        Treatment_id int(10) PRIMARY KEY,
        Treatment_name varchar(50),
        Treatment_duration varchar(20),
        Treatment_type varchar(20)
    );
```

2. Creating the Fact Table

```
CREATE TABLE fact_table(
  Doctor_id int(10) REFERENCES doctor(Doctor_id), Patient_id
  int(10) REFERENCES patient(Patient_id), location_id int(10)
```

```
REFERENCES location(location_id), treatment_id int(10)
REFERENCES treatment(Treatment_id),
Count int(10),
Fees int(10)
);
```

3. Inserting values in both dimension and fact tables

```
INSERT INTO location(location_id,city,state,pincode,country)
VALUES(1234567789,'vasai','maharashtra',301303,'India'),
(1234567779,'vasai','maharashtra',401202,'India'),
(1234567079,'vasai','maharashtra',401206,'India'),
(1234567059,'vasai','maharashtra',401106,'India');

INSERT INTO
treatment(Treatment_id,Treatment_name,Treatment_duration,Treatment_type)
VALUES(101,"Angioplasty","3 Hours","Heart Disease"),
(102,"Chemotherapy","8 Hours","Lungs Disease"),
(103,"Heart Bypass Surgery","14 Hours","Heart Disease"), (104,"Lead Extraction","12
Hours","Heart Disease"),
(105,"Dialysis","5 Hours","Kidney Disease");

INSERT INTO fact_table(Doctor_id,Patient_id,location_id,treatment_id,Count,Fees)
VALUES (01,62,1234567789,101,31,500),
(02,66,1234567779,102,32,700),
(01,60,1234567079,103,33,800),
(02,61,1234567059,104,34,900);
```

INSERT INTO doctor(Doctor_id,Doctor_name,Doctor_type,Doctor_age,Doctor_experience,1st_week,2nd _week,3rd_week,4th_week)

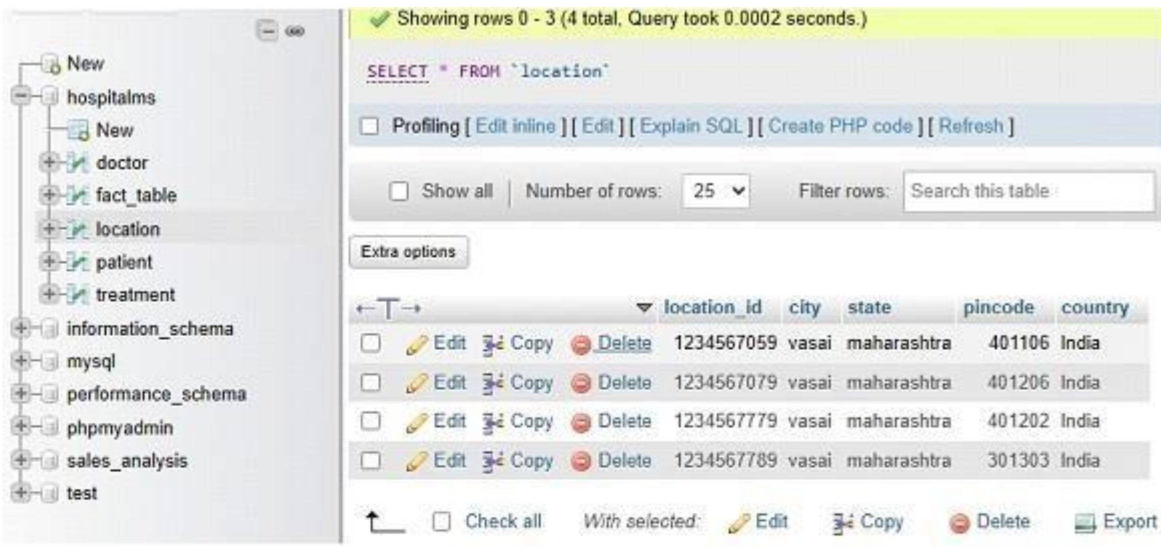
VALUES (01,'Sairaj','Brain',20,'1 years',100,200,100,100),

(02,'Viraj','Kidney',20,'5 years',200,100,100,300),

(03,'Shivansh','Heart',21,'7 years',200,200,200,200),

(04,'Anand','Lung',19,'4 years',300,300,200,100); 4.

Displaying the tables Location:



Patient:

Patient_id	Patient_name	Patient_age	Patient_address	Report_id
60	Shivansh	20	Vasai	4
61	Anand	20	Vasai	5
62	Sairaj	20	Virar	2
66	Viraj	20	Goregaon	3

.

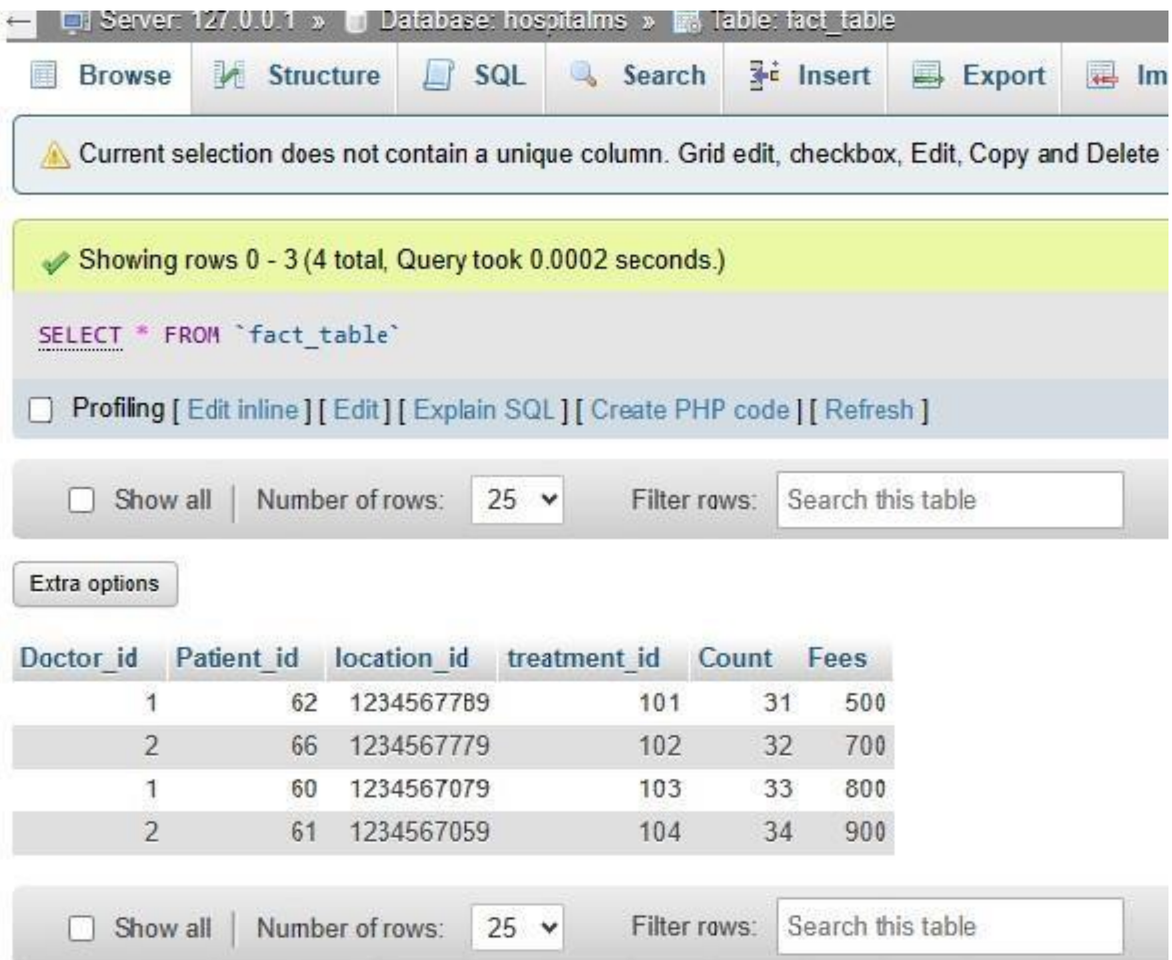
Doctor:

Doctor_Id	Doctor_name	Doctor_type	Doctor_age	Doctor_experience	1st_week	2nd_week	3rd_week	4th_week
1	Sairaj	Brain	20	1 years	100	200	100	100
2	Viraj	Kidney	20	6 years	200	100	100	300
3	Shivansh	Heart	21	7 years	200	200	200	200
4	Anand	Lung	19	4 years	300	300	200	100

Treatment:

Treatment_id	Treatment_name	Treatment_duration	Treatment_type
101	Angioplasty	3 Hours	Heart Disease
102	Chemotherapy	8 Hours	Lungs Disease
103	Heart Bypass Surgery	14 Hours	Heart Disease
104	Lead Extraction	12 Hours	Heart Disease
105	Dialysis	5 Hours	Kidney Disease

Fact Table:



Doctor_id	Patient_id	location_id	treatment_id	Count	Fees
1	62	1234567789	101	31	500
2	66	1234567779	102	32	700
1	60	1234567079	103	33	800
2	61	1234567059	104	34	900

5. Write SQL Queries for all the above OLAP operations.

Roll Up :

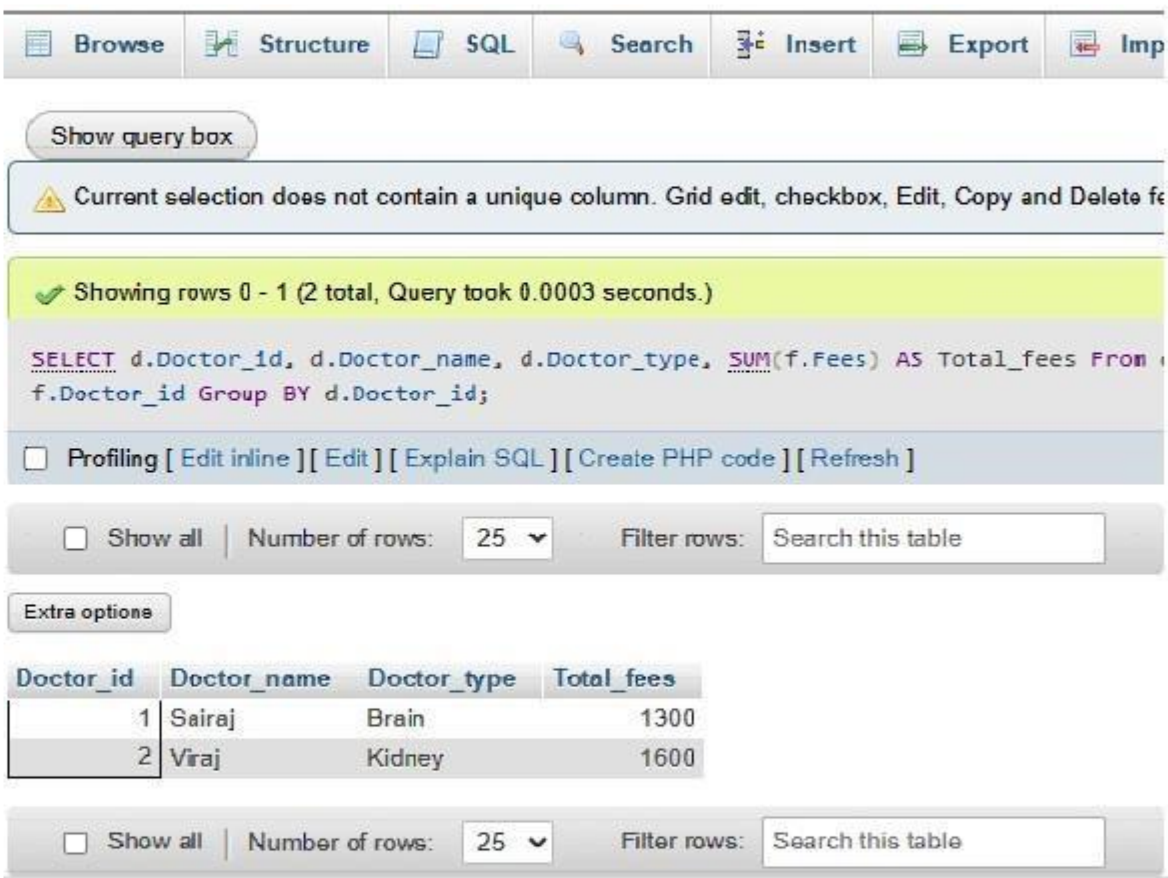
SELECT d.Doctor_id, d.Doctor_name, d.Doctor_type, SUM(f.Fees) AS Total_fees

From doctor AS d

JOIN fact_table AS f

Where d.Doctor_id = f.Doctor_id

Group BY d.Doctor_id ;



Doctor_id	Doctor_name	Doctor_type	Total_fees
1	Sairaj	Brain	1300
2	Viraj	Kidney	1600

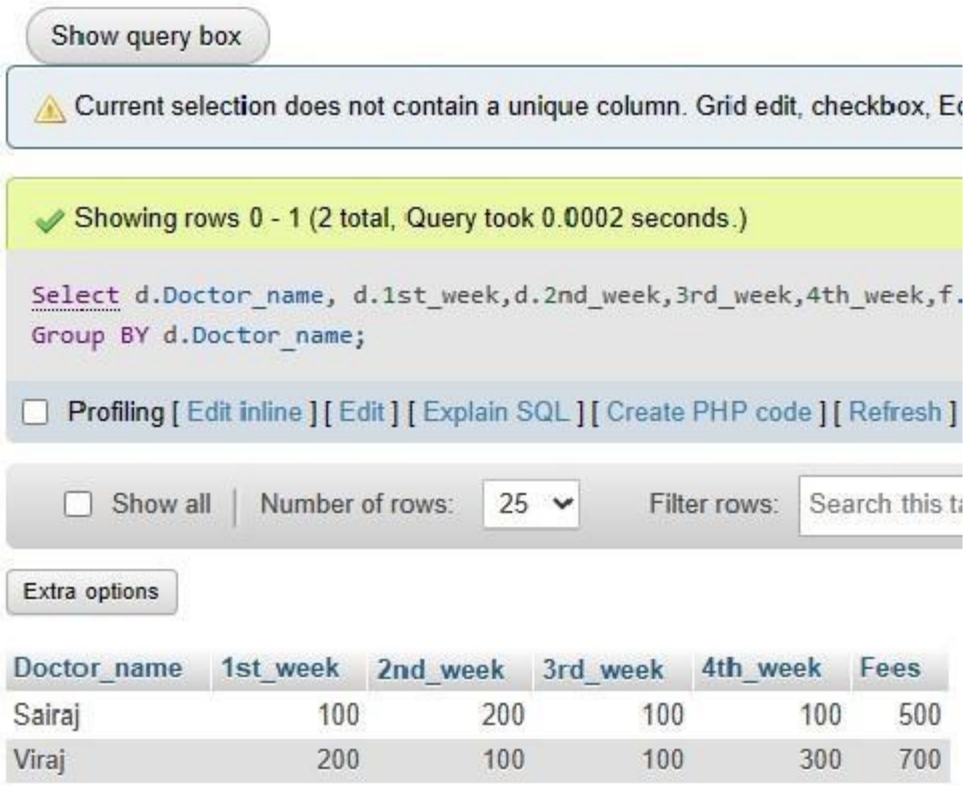
Drill Down :

Select d.Doctor_name, d.1st_week,d.2nd_week,3rd_week,4th_week,f.Fees

FROM doctor as d

JOIN fact_table AS f Where d.Doctor_id = f.Doctor_id

Group BY d.Doctor_name ;



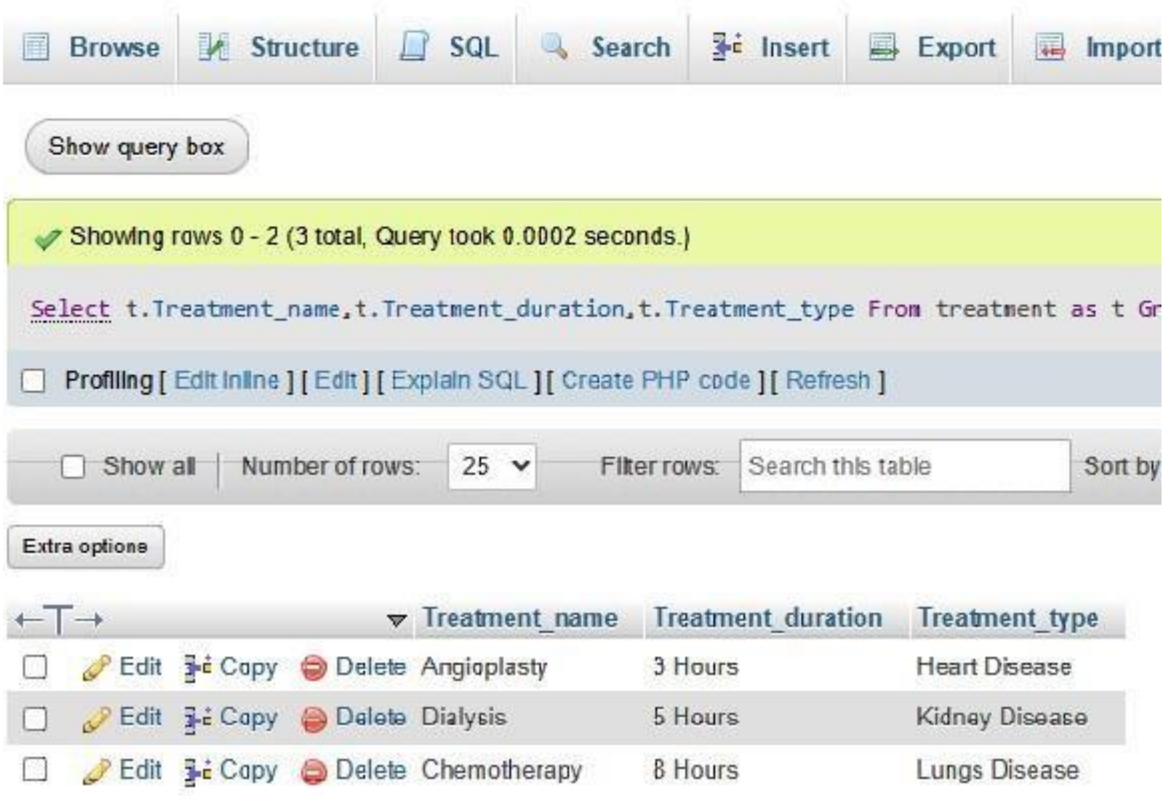
Doctor_name	1st_week	2nd_week	3rd_week	4th_week	Fees
Sairaj	100	200	100	100	500
Viraj	200	100	100	300	700

Slice :

Select t.Treatment_name,t.Treatment_duration,t.Treatment_type

From treatment as t

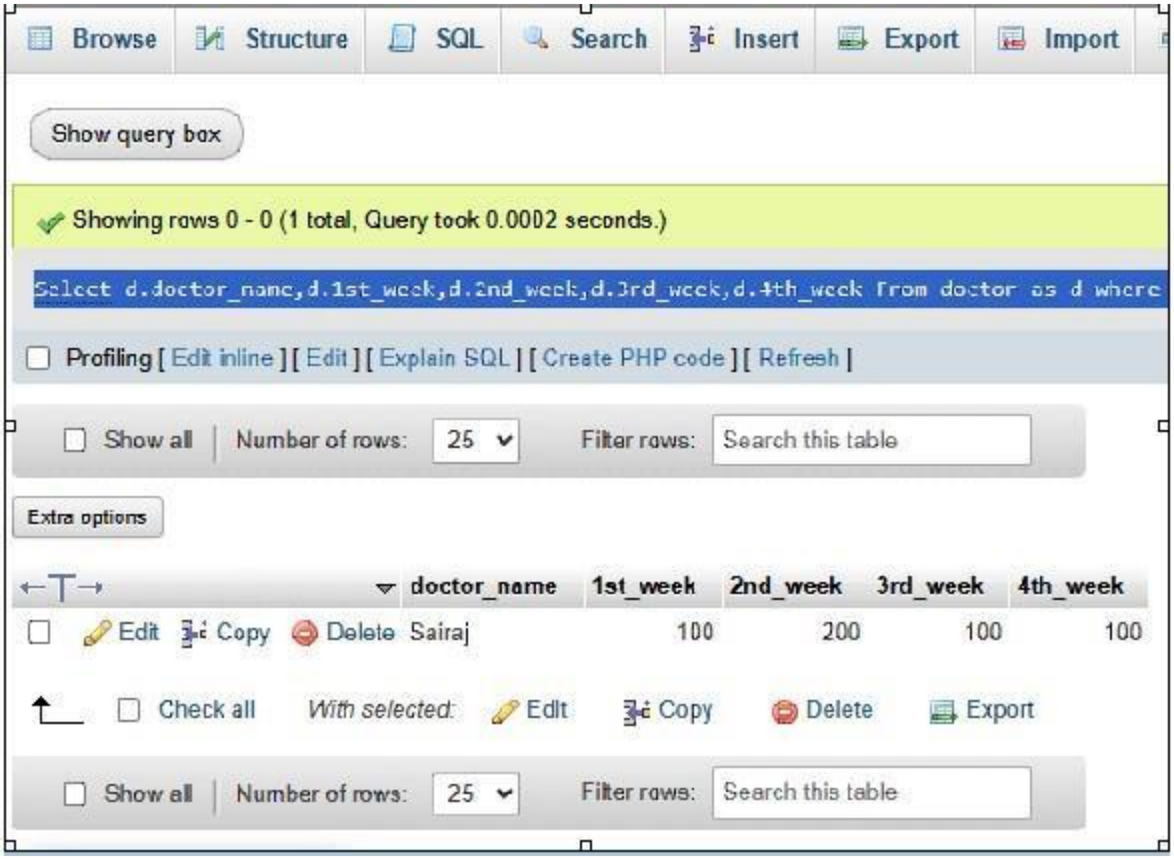
Group by t.Treatment_type;



Dice :

Select d.doctor_name,d.1st_week,d.2nd_week,d.3rd_week,d.4th_week

From doctor as d where d.Doctor_id = 1;

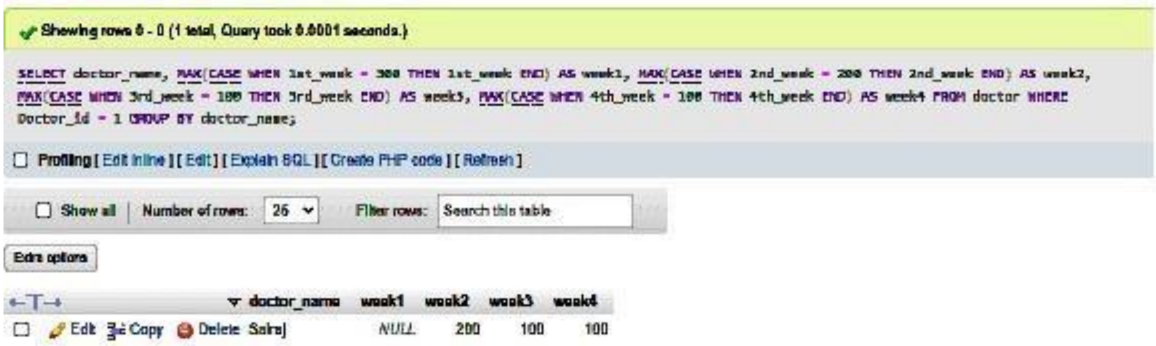


pivot:

```
SELECT
    doctor_name,
    MAX(CASE WHEN 1st_week = 300 THEN 1st_week END) AS week1, MAX(CASE
    WHEN 2nd_week = 200 THEN 2nd_week END) AS week2, MAX(CASE WHEN
    3rd_week = 100 THEN 3rd_week END) AS week3,
    MAX(CASE WHEN 4th_week = 100 THEN 4th_week END) AS week4
FROM
    doctor
WHERE
    Doctor_id = 1
```

GROUP BY

doctor_name;



Conclusion:
The implementation of dimension and fact tables, along with OLAP operations, is crucial for building a robust and efficient data analytics environment. It empowers organizations to extract valuable insights from their data, make data-driven decisions, and adapt to changing business needs. However, it's important to continuously monitor and maintain the data warehouse to ensure data accuracy and relevance.