VU VISHWAKARMA UNIVERSITY

Maximising Human Potential

T. Y. B. Tech Computer Engineering
2024-2025
Pursued in Department of Computer Engineering Faculty of Science & Technology
Vishwakarma University, Pune-411048

NAME	Swanand Garge
YEAR	THIRD YEAR
DIV	D
BATCH	D2
ROLL NO	39
SRN NO	202201589
PRN NO	2280030433
COURSE NAME	DATAWAREHOUSE & DATA MINING LAB
COURSE CODE	BTECCE22509
COURSE TEACHER NAME	PROF. RAHUL PAPALKAR

ASSIGNMENT NO: 6

Problem Statement:

Assignment 6

Gather Businesse Requirements for Rental and sales Analysis and design it using Multi-dimensional data model namely star schema.

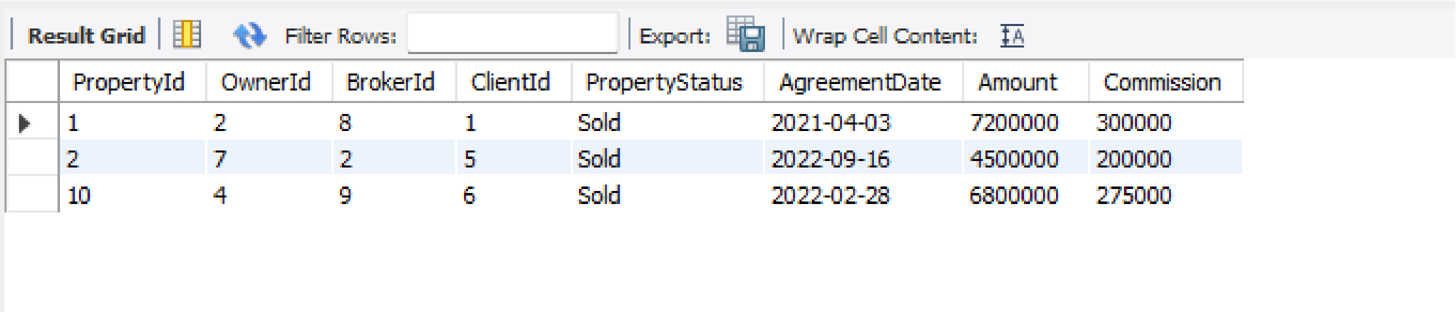
THEORY:

- **Star Schema:**

  - A type of database schema that organizes data into fact and dimension tables.
  - The fact table is at the center, and it connects to dimension tables, which "radiate" out like the points of a star.
  - It is simple, easy to understand, and supports efficient queries.

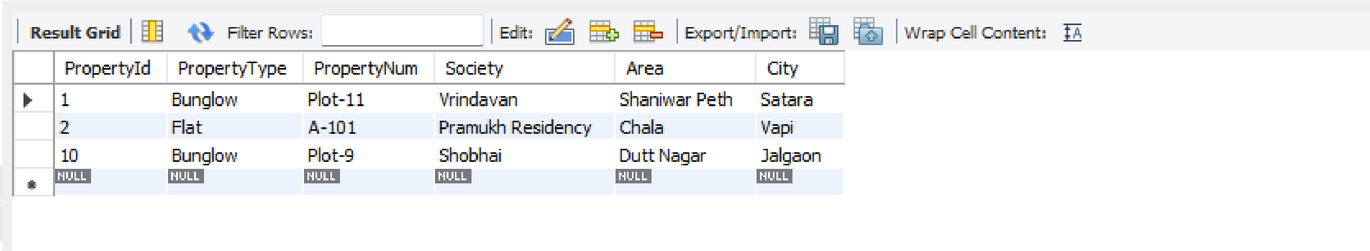
- **Fact Table:**

  - Stores quantitative data (measurable metrics) related to business processes, like sales, revenue, or performance metrics.
  - Each record in a fact table is a combination of foreign keys to dimension tables and facts (numeric data).
  - Typically, it has many rows and fewer columns.

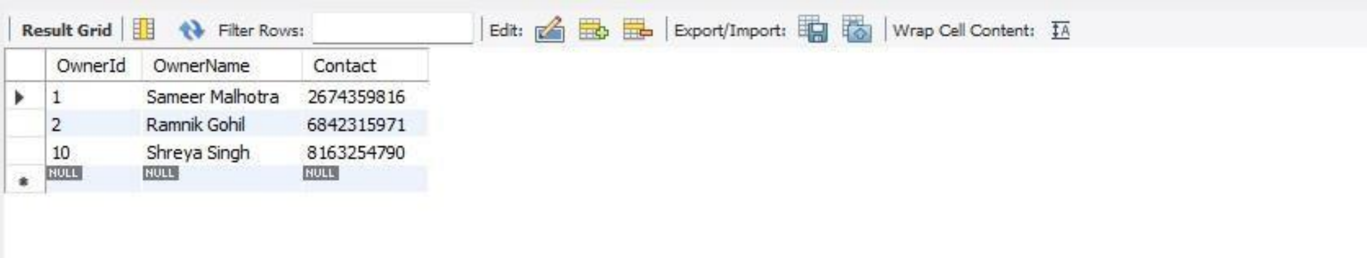


PropertyId	OwnerId	BrokerId	ClientId	PropertyStatus	AgreementDate	Amount	Commission
1	2	8	1	Sold	2021-04-03	7200000	300000
2	7	2	5	Sold	2022-09-16	4500000	200000
10	4	9	6	Sold	2022-02-28	6800000	275000

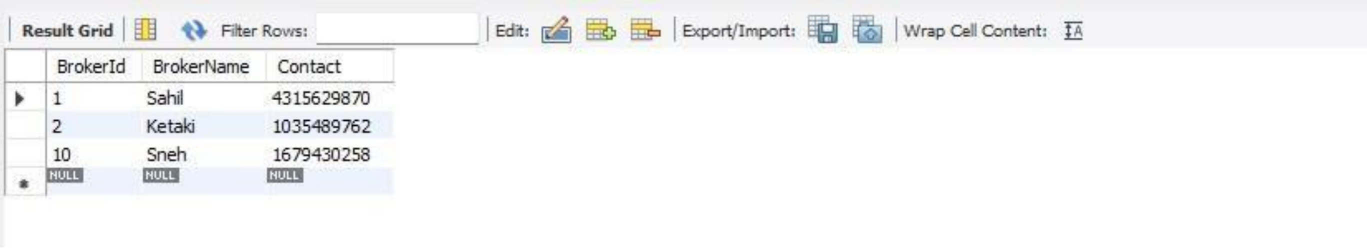
- **Dimension Table:**
  - Contains descriptive attributes (context) about the facts, like product details, time, location, or customer information.
  - Dimension tables are used to filter, group, and label the data in the fact table.
  - They usually have fewer rows but more columns compared to fact tables.



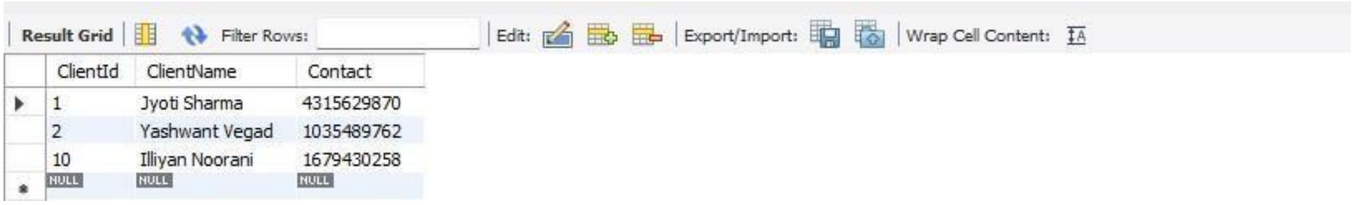
PropertyId	PropertyType	PropertyNum	Society	Area	City
1	Bunglow	Plot-11	Vrindavan	Shaniwar Peth	Satara
2	Flat	A-101	Pramukh Residency	Chala	Vapi
10	Bunglow	Plot-9	Shobhai	Dutt Nagar	Jalgaon
NULL	NULL	NULL	NULL	NULL	NULL



OwnerId	OwnerName	Contact
1	Sameer Malhotra	2674359816
2	Ramnik Gohil	6842315971
10	Shreya Singh	8163254790
NULL	NULL	NULL



BrokerId	BrokerName	Contact
1	Sahil	4315629870
2	Ketaki	1035489762
10	Sneh	1679430258
NULL	NULL	NULL



ClientId	ClientName	Contact
1	Jyoti Sharma	4315629870
2	Yashwant Vegad	1035489762
10	Illiyan Noorani	1679430258
NULL	NULL	NULL

1. Fact_Sales Table (Fact Table)
Attributes:

Sales_Rep_id (FK)
Time_id (FK)
Order_id (FK)
Customer_id (FK)
Product_id (FK)
Net_amount_per_customer
Net_amount_per_product
Net_amount_per_promotion

Relationships:

Many-to-One with Dim_Sales_Rep (via Sales_Rep_id)
Many-to-One with Dim_Time (via Time_id)
Many-to-One with Dim_Customer (via Customer_id)
Many-to-One with Dim_Product (via Product_id)
Many-to-One with Dim_Order (via Order_id)
2. Dim_Sales_Rep Table (Dimension Table)

3. Attributes:

Sales_Rep_id (PK)
Name
Deal
Discount
Relationships:

One-to-Many with Fact_Sales (via Sales_Rep_id)

4. Dim_Time Table (Dimension Table)
Attributes:

Time_id (PK)
Day
Month
Year
Relationships:

One-to-Many with Fact_Sales (via Time_id)

SOURCE CODE:

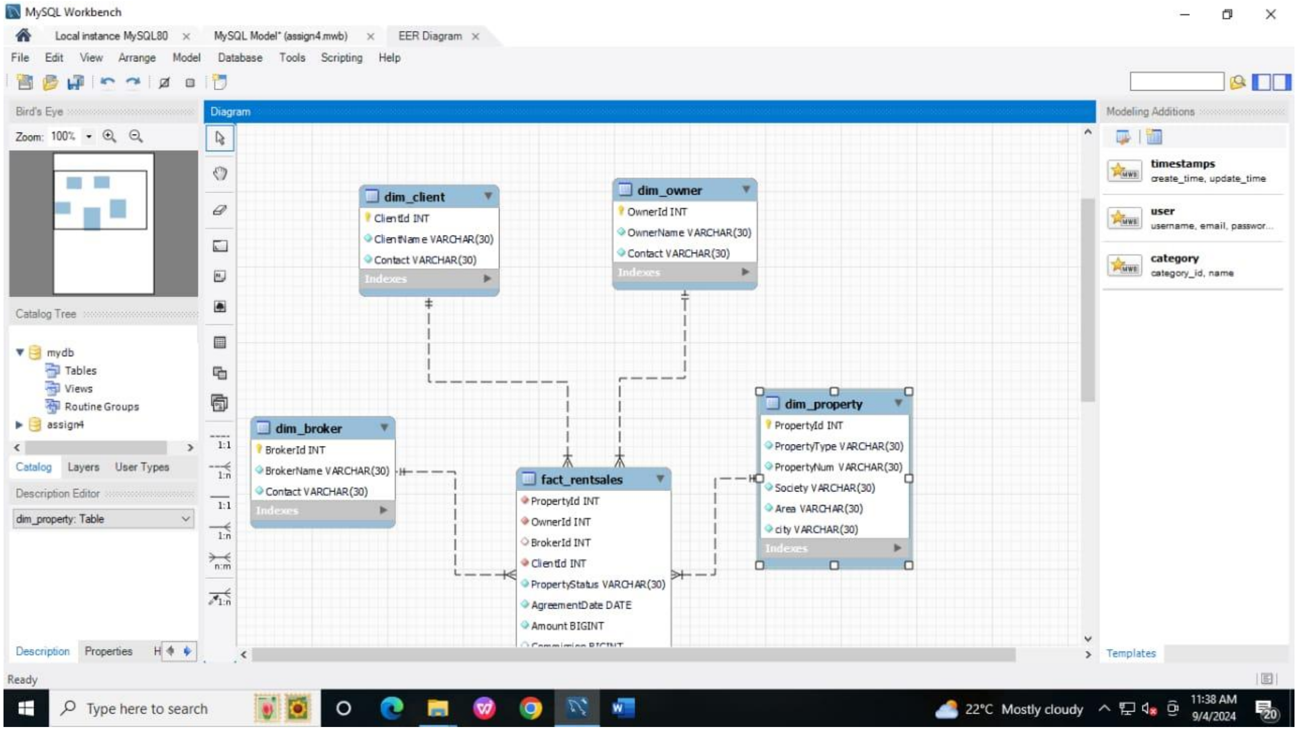
```
CREATE DATABASE rental;
USE rental;

-- Fact Table
CREATE TABLE Fact_RentSales (
    PropertyId int not null,
    OwnerId int not null,
    BrokerId int,
    ClientId int not null,
    PropertyStatus varchar(30) not null,
    AgreementDate date not null,
    Amount bigint not null,
    Commission bigint,
    PaymentMethodId int not null,  -- Added foreign key to Payment Method
    DateId int not null,           -- Added foreign key to Time Dimension
    FOREIGN KEY(PropertyId) REFERENCES Dim_Property(PropertyId),
    FOREIGN KEY(OwnerId) REFERENCES Dim_Owner(OwnerId),
    FOREIGN KEY(BrokerId) REFERENCES Dim_Broker(BrokerId),
    FOREIGN KEY(ClientId) REFERENCES Dim_Client(ClientId),
    FOREIGN KEY(PaymentMethodId) REFERENCES Dim_PaymentMethod(PaymentMethodId),
    FOREIGN KEY(DateId) REFERENCES Dim_Time(DateId)
);

-- Property Dimension
CREATE TABLE Dim_Property (
    PropertyId int not null,
    PropertyType varchar(30) not null,
    PropertyNum varchar(30) not null,
    Society varchar(30) not null,
    Area varchar(30) not null,
    City varchar(30) not null,
    Bedrooms int,                  -- Added new column for bedrooms
    YearBuilt int,                 -- Added new column for year built
    PRIMARY KEY(PropertyId)
);

-- Owner Dimension
```

```
CREATE TABLE Dim_Owner (
    OwnerId int not null,
    OwnerName varchar(30) not null,
    Contact varchar(30) not null,
    Email varchar(50),                -- Added new column for email
    DateOfBirth date,                 -- Added new column for date of birth
    PRIMARY KEY(OwnerId)
);

-- Broker Dimension
CREATE TABLE Dim_Broker (
    BrokerId int not null,
    BrokerName varchar(30) not null,
    Contact varchar(30) not null,
    Email varchar(50),                -- Added new column for email
    PRIMARY KEY(BrokerId)
);

-- Client Dimension
CREATE TABLE Dim_Client (
    ClientId int not null,
    ClientName varchar(30) not null,
    Contact varchar(30) not null,
    Email varchar(50),                -- Added new column for email
    Address varchar(100),             -- Added new column for client address
    PRIMARY KEY(ClientId)
);

-- Location Dimension (new)
CREATE TABLE Dim_Location (
    LocationId int not null,
    City varchar(30) not null,
    State varchar(30) not null,
    Country varchar(30) not null,
    PRIMARY KEY(LocationId)
);

-- Time Dimension (new)
CREATE TABLE Dim_Time (
    DateId int not null AUTO_INCREMENT,
    Date date not null,
    Day int not null,
    Month int not null,
    Year int not null,
    Quarter int not null,
    PRIMARY KEY(DateId)
);

-- Payment Method Dimension (new)
CREATE TABLE Dim_PaymentMethod (
    PaymentMethodId int not null AUTO_INCREMENT,
    PaymentMethod varchar(30) not null,
    PRIMARY KEY(PaymentMethodId)
);

-- Insert values into Fact_RentSales
INSERT INTO Fact_RentSales VALUES
('1', '2', '8', '1', 'Sold', '2021-04-03', '7200000', '300000', 1, 1),
('2', '7', '2', '5', 'Sold', '2022-09-16', '4500000', '200000', 2, 2),
('10', '4', '9', '6', 'Sold', '2022-02-28', '6800000', '275000', 3, 3);

-- Insert values into Dim_Property
INSERT INTO Dim_Property VALUES
('1', 'Bunglow', 'Plot-11', 'Vrindavan', 'Shaniwar Peth', 'Satara', 4, 2010),
('2', 'Flat', 'A-101', 'Pramukh Residency', 'Chala', 'Vapi', 3, 2015),
('10', 'Bunglow', 'Plot-9', 'Shobhai', 'Dutt Nagar', 'Jalgaon', 5, 2005);

-- Insert values into Dim_Owner
INSERT INTO Dim_Owner VALUES
('1', 'Sameer Malhotra', '2674359816', 'sameer@example.com', '1975-08-12'),
('2', 'Ramnik Gohil', '6842315971', 'ramnik@example.com', '1980-11-05'),
('10', 'Shreya Singh', '8163254790', 'shreya@example.com', '1988-04-22');

-- Insert values into Dim_Broker
INSERT INTO Dim_Broker VALUES
('1', 'Sahil', '4315629870', 'sahil@example.com'),
('2', 'Ketaki', '1035489762', 'ketaki@example.com'),
('10', 'Sneh', '1679430258', 'sneh@example.com');

-- Insert values into Dim_Client
```

```
INSERT INTO Dim_Client VALUES
('1', 'Jyoti Sharma', '4315629870', 'jyoti@example.com', '123 Park Street, Mumbai'),
('2', 'Yashwant Vegad', '1035489762', 'yashwant@example.com', '45 Green Road, Surat'),
('10', 'Illiyan Noorani', '1679430258', 'illiyan@example.com', '67 Kings Avenue, Pune');

-- Insert values into Dim_Location
INSERT INTO Dim_Location VALUES
('1', 'Satara', 'Maharashtra', 'India'),
('2', 'Vapi', 'Gujarat', 'India'),
('3', 'Jalgaon', 'Maharashtra', 'India');

-- Insert values into Dim_Time
INSERT INTO Dim_Time (Date, Day, Month, Year, Quarter) VALUES
('2021-04-03', 3, 4, 2021, 2),
('2022-09-16', 16, 9, 2022, 3),
('2022-02-28', 28, 2, 2022, 1);

-- Insert values into Dim_PaymentMethod
INSERT INTO Dim_PaymentMethod (PaymentMethod) VALUES
('Credit Card'), ('Cash'), ('Bank Transfer');

-- Queries for Sales Analysis
SELECT dp.City, SUM(frs.Amount) AS TotalAmount
FROM Fact_RentSales frs
JOIN Dim_Property dp ON frs.PropertyId = dp.PropertyId
GROUP BY dp.City;

SELECT dp.PropertyType, AVG(frs.Commission) AS AvgCommission
FROM Fact_RentSales frs
JOIN Dim_Property dp ON frs.PropertyId = dp.PropertyId
GROUP BY dp.PropertyType;

SELECT SUM(Amount) AS TotalSalesAmount
FROM Fact_RentSales
WHERE PropertyStatus = 'Sold';

SELECT SUM(Commission) AS TotalSalesCommission
FROM Fact_RentSales
WHERE PropertyStatus = 'Sold';
```

```
SELECT COUNT(*) AS TotalSalesTransactions
FROM Fact_RentSales
WHERE PropertyStatus = 'Sold';

SELECT *
FROM Fact_RentSales;
```

OUTPUT:



Description Editor: dim_property: Table

timestamps
create_time, update_time

user
username, email, passwor...

category
category_id, name

dim_client
ClientId INT
ClientName VARCHAR(30)
Contact VARCHAR(30)
Indexes

dim_owner
OwnerId INT
OwnerName VARCHAR(30)
Contact VARCHAR(30)
Indexes

dim_broker
BrokerId INT
BrokerName VARCHAR(30)
Contact VARCHAR(30)
Indexes

fact_rentsales
PropertyId INT
OwnerId INT
BrokerId INT
ClientId INT
PropertyStatus VARCHAR(30)
AgreementDate DATE
Amount BIGINT
Commision BIGINT

dim_property
PropertyId INT
PropertyType VARCHAR(30)
PropertyNum VARCHAR(30)
Society VARCHAR(30)
Area VARCHAR(30)
city VARCHAR(30)
Indexes

CONCLUSION:

The SQL code implements key concepts of data warehousing, specifically using the star schema design to efficiently organize and manage rental and sales data for analysis. The schema consists of a central fact table (`Fact_RentSales`) that records transactional data such as the `Amount`, `Commission`, and `AgreementDate`, surrounded by multiple dimension tables like `Dim_Property`, `Dim_Owner`, `Dim_Client`, and `Dim_Time`, which provide descriptive context to the facts. These dimension tables are connected to the fact table via foreign keys, ensuring data integrity and facilitating accurate query results.

The granularity of the fact table is at the transaction level, capturing detailed records of each rent or sale. This design supports aggregations and analyses such as total sales by city or average commission by property type. By separating dimensions like property, owner, and time, the schema allows for flexibility and efficiency in queries without redundancy. For instance, details about properties are stored only once in `Dim_Property`, allowing the fact table to reference them multiple times.

Additionally, the `Dim_Time` table enables efficient time-based analysis, a critical feature in data warehousing, while the foreign key constraints maintain strong data relationships between the fact and dimension tables. Aggregation functions like `SUM`, `COUNT`, and `AVG` further exemplify how the schema supports analytical queries, making it easier to derive meaningful insights. Overall, this star schema structure provides a highly organized, scalable approach to managing data, allowing for powerful reporting and analysis in a sales and rental context.