



PRIMEHEAVEN (REAL-ESTATE)

1. Project Overview

Project Name:

PrimeHeaven – Real Estate Platform

Description:

PrimeHeaven is an online real estate platform designed to streamline property listing, buying, selling, and renting. It provides features like property search, filtering options, secure transactions, and user management to support secure and efficient real estate transactions.

Core Technologies:

HTML, CSS, Frontend Framework (Bootstrap), Database (SQL).

Objectives:

- The objective of the PrimeHeaven project is to create a user-friendly online platform where people can easily list, buy, rent, and sell properties. By bringing together property listings and important search tools, it helps users find their ideal home or investment quickly and efficiently.
- The platform is designed to make the process of finding, comparing, and securing a property smoother and safer. With built-in search filters, secure transaction features, and user-friendly design.
- PrimeHeaven aims to simplify real estate transactions, making them accessible and transparent for everyone involved.

2. Database Design

Objectives:

Outline key database requirements for real estate data, such as property listings, user information, transactions, and filtering capabilities.

Schema Design Overview:

The schema for PrimeHeaven's database is designed to support a real estate platform that enables users to list, search, inquire about, and transact properties. Key tables represent entities like **Customer**, **Property**, **PropertyCategory**, **PropertyFeatures**, **PropertySeller**, **PropertyInquiry**, and **Payment**. Relationships among these tables ensure that data is linked efficiently, providing a foundation for complex queries, reporting, and real-time interactions.

1. Table Structure

Each table in the schema is designed to capture specific aspects of the platform's requirements, as follows:

1.1 Customer Table

Stores information about the platform's users, who can be buyers or sellers.

Key Attributes:

- Cid (Primary Key): Unique identifier for each customer.
- Cname, Cemail, Cphone: Basic contact details for the customer.
- Ctype: Specifies if the customer is a buyer, seller, or both.

Purpose: Supports user authentication, profile management, and linking inquiries or transactions to specific customers.

1.2 Property Table

Contains all property listings on the platform.

Key Attributes:

- Proid (Primary Key): Unique identifier for each property.
- Proname, Prolocation, Proprice, Prostatus: Property details such as name, location, price, and availability status.

- Category_id & Profeatureid (Foreign Key): Links each property to a specific category in the PropertyCategory table and specific feature in the Property.

Purpose: Allows properties to be listed, categorized, searched, and linked with their features, inquiries, and payment history.

1.3 PropertyCategory Table

Holds various categories for properties, like Residential, Commercial, etc.

Key Attributes:

- Category_id (Primary Key): Unique identifier for each category.
- Category_name: Descriptive name of the property category.

Purpose: Allows properties to be grouped under categories, making it easier for users to filter and search for specific types.

1.4 PropertyFeatures Table

Stores additional property features and amenities.

Key Attributes:

- Profeatureid (Primary Key): Unique identifier for each feature entry.
- Proid (Foreign Key): Links the features to a specific property in the Property table.
- Attributes like Floorspace, Balcony, NumofBedroom, Gym, and Swimmingpool: Describe property-specific features and amenities.

Purpose: Enables enhanced filtering of properties based on features, helping users find properties that match specific criteria.

1.5 PropertySeller Table

Manages information about property sellers.

Key Attributes:

- ProSellerid (Primary Key): Unique identifier for each seller.
- ProSellername, ProSelleremail, ProSellerphone: Contact information for the seller.
- Proid (Foreign Key): Links the seller to a specific property in the Property table.

Purpose: Supports seller profiles, links sellers to their listed properties, and enables tracking of seller information during transactions.

1.6 PropertyInquiry Table

Manages inquiries sent by customers about specific properties.

Key Attributes:

- Inquiryid (Primary Key): Unique identifier for each inquiry.
- Proid (Foreign Key): Links the inquiry to a specific property in the Property table.
- SenderCid (Foreign Key): Links the inquiry to the customer making it.
- Inquirymessage, Inquirydate: Content of the inquiry and its date.

Purpose: Allows potential buyers to inquire about properties, enabling communication between customers and sellers.

1.7 Payment Table

Records payment transactions for properties.

Key Attributes:

- Paymentid (Primary Key): Unique identifier for each payment.
- Proid (Foreign Key): Links the payment to a specific property in the Property table.
- BuyerCid (Foreign Key): Links the payment to the buyer (customer) making the purchase.
- ProSellerid (Foreign Key): Links the payment to the seller.
- Attributes like Paymentamount, Paymentmethod, and Paymentstatus: Track transaction details.

Purpose: Facilitates secure transactions, linking properties to buyers and sellers, and records payment status and methods.

2. Relationships Between Tables

The schema design emphasizes the relationships between these entities to create a fully integrated database that supports complex real estate workflows.

One-to-Many Relationships:

Customer - Property: A customer can list multiple properties, but each property is owned by one customer (likely sellers).

PropertyCategory - Property: Each property belongs to one category, but each category can include multiple properties.

Property - PropertyFeatures: Each property can have multiple features (like floorspace, gym, parking), but each feature entry belongs to one property.

Customer - PropertyInquiry: A customer can make multiple inquiries, but each inquiry is linked to one customer.

Property - Payment: Each payment is associated with one property, but a property could involve multiple payments (e.g., installments).

PropertySeller - Property: Each seller can list multiple properties, but each property has one seller.

Property - PropertyInquiry: Each property can have multiple inquiries, but each inquiry pertains to one property.

Customer – Payment: A Customer (typically a buyer) can make multiple payments for properties, but each payment record is linked to one specific customer. This relationship allows tracking of all payments made by each customer.

3. Entity-Relationship Diagram (ERD):

An **Entity-Relationship Diagram (ER Diagram)** is a visual representation of the data model for a system, showing how different entities (or concepts) relate to one another within that system.

ER diagrams are widely used in database design to illustrate the logical structure of databases and to define the relationships between various data elements.

The main components of ER model are:

Entity set: A group of similar entities with the same attributes.

Relationship set: A group of associations between entities, showing how they relate.

Here are the geometric shapes and their meaning in an ER Diagram:

Rectangle : Represents Entity sets.

Ellipses : Attributes.

Diamonds: Relationship set.

Lines : They link attributes to Entity Sets and this to Relationship Set.

Dotted Ellipses : Add foreign key to that attribute.

ER Diagram:

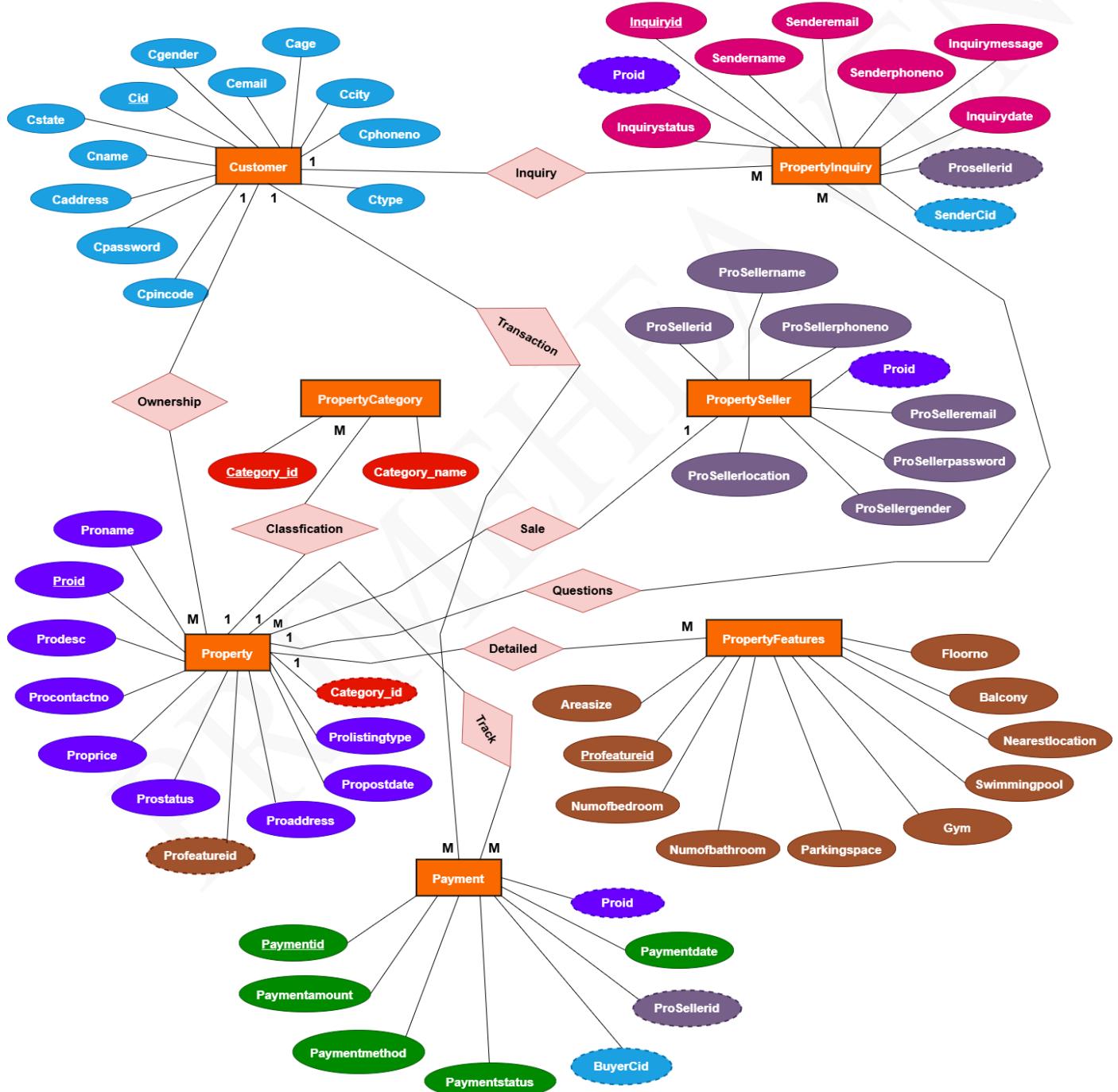


Fig 1: Entity Relationship Diagram for PRIMEHEAVEN (Real-Estate)

3. NORMALIZATION:

Normalization is a systematic process in database design aimed at reducing data redundancy and improving data integrity.

The Objective of Normalization:

1. Eliminate Redundancy: Reduce duplication of data across table/database.

2. Ensure Data Integrity: It maintains accuracy and consistency of data.

There are three main types of normal forms:

- a) First normal form(1NF)
- b) Second normal form(2NF)
- c) Third normal form(3NF)

(a) First normal form(1NF):

A table is in 1NF if it only contains atomic values and each column contains only a single value per row.

Requirement:

No repeating groups.

Each cell contains a single value.

Each record is unique.

(b) Second normal form(2NF):

A table is in 2NF if it is in 1NF and all non-key attributes are fully functionally dependent on entire primary key.

Requirement:

No partial dependency (every non key attribute must be dependent on whole primary key not just part of its).

(c) Third normal form(3NF):

A table is in 3NF if it is in 2NF and all non-key attributes are not only fully functionally dependent on the primary key but also directly dependent on the primary key.

Requirement:

No (Avoid) transitive dependency.

NORMALIZED PRIMEHEAVEN DATABASE TABLE USING 1NF, 2NF AND 3NF:

Initial Unnormalized Table:

To normalize the table and demonstrate 1NF, 2NF, and 3NF, let's start by creating an unnormalized table using input values based on the attributes.

Cname	Cage	Caddress	Cgender	Cemail	Cphoneno	Ctype
Ramesh	30	Mumbai	Male	Ramesh@email.com	9876543210	Buyer
Sita	25	Pune	Female	sita@email.com	9876509876	Saler
Anjali	28	Nashik	Female	anjali@email.com	9898989898	Buyer
Arjun	32	Thane	Male	arjun@email.com	9765432109	Buyer

Priname	Procity	Proprice	Procontactno	Prolistingtype	Prostatus	Category_name
Sea Breeze	Mumbai	5000000	8765432109	For Sale	Available	Residential
Hil View	Pune	3000000	9876509876	For Rent	Rented	IndependentHome
Green Meadows	Nashik	4000000	9888776655	For Sale	Available	Commercial
Royal Heights	Thane, Dombivli	(6000000)(15000)	9765123456	For Sale, For Rent	Sold, Rented	Plot, PG Room

Prosellername	Prosellerlocation	Prosellerphoneno	Floorno	Areasize	Numofbedroom	Furnished
Sunil	Mumbai	9988776655	5	1500	2	Yes
Radha	Pune	9988665544	NO	3200	4	Semi
Kiran	Nashik	9988555522	2	2000	NO	Yes
Meera	Thane	9876767676	NO,3	(2500)(1000)	NO, 1	No, Semi

Parkingspace	Sendername	Senderphoneno	Inquirymessage	Inquirydate
Available	Ramesh	9876543210	Interested in property	07-11-2024
Available	Sita	9876509876	Need more details	06-11-2024
Available	Anjali	9898989898	Scheduled for site visit	08-11-2024
Unavailable, Available	Arjun	9765432109	Request for price reduction	09-11-2024

Inquirystatus	Paymentamount	Paymentdate	Paymentstatus
Completed	5000000	11-11-2024	Done
In Progress	3000000	20-11-2024	Pending
Pending	4000000	18-11-2024	Pending
Completed	(6000000)(15000)	19-11-2024	Done

Normalization:

1NF:

- 1. Every column/attribute need to have a single value.**
- 2. Each row should be unique. Either through a single or multiple columns. Not mandatory to have primary key.**

Cname	Cage	Caddress	Cgender	Cemail	Cphoneno
Ramesh	30	Mumbai	Male	Ramesh@email.com	9876543210
Sita	25	Pune	Female	sita@email.com	9876509876
Anjali	28	Nashik	Female	anjali@email.com	9898989898
Arjun	32	Thane	Male	arjun@email.com	9765432109
Arjun	32	Thane	Male	arjun@email.com	9765432109

Ctype	Proname	Procity	Proprice	Procontactno	Prolistingtype
Buyer	Sea Breeze	Mumbai	5000000	8765432109	For Sale
Saler	Hil View	Pune	3000000	9876509876	For Rent
Buyer	Green Meadows	Nashik	4000000	9888776655	For Sale
Buyer	Royal Heights	Thane	6000000	9765123456	For Sale
Buyer	Royal Heights	Dombivli	15000	9765123456	For Rent

Prostatus	Category_name	Prosellername	Prosellerlocation	Prosellerphoneno	Floorno
Available	Residential	Sunil	Mumbai	9988776655	5
Rented	IndepentdentHome	Radha	Pune	9988665544	NO
Available	Commercial	Kiran	Nashik	9988555522	2
Sold	Plot	Meera	Thane	9876767676	NO
Rented	PG Room	Meera	Thane	9876767676	3

Areasize	Numofbedroom	Furnished	Parkingspace	Sendername	Senderphoneno
1500	2	Yes	Available	Ramesh	9876543210
3200	4	Semi	Available	Sita	9876509876
2000	NO	Yes	Available	Anjali	9898989898
2500	NO	No	Unavailable	Arjun	9765432109
1000	1	Semi	Available	Arjun	9765432109

Inquirymessage	Inquirydate	Inquirystatus	Paymentamount	Paymentdate	Paymentstatus
Interested in property	07-11-2024	Completed	5000000	11-11-2024	Done
Need more details	06-11-2024	In Progress	3000000	20-11-2024	Pending
Scheduled for site visit	08-11-2024	Pending	4000000	18-11-2024	Pending
Request for price reduction	09-11-2024	Completed	6000000	19-11-2024	Done
Request for price reduction	09-11-2024	Completed	15000	19-11-2024	Done

2NF:

- 1. Must be in 1NF.**
- 2. All non-key attributes must be fully dependent on candidate key.**

If a non-key column in partially dependent on candidate key (subset of column forming candidate key) then split them into separate table.

- 3. Every table should have primary key and relationship between the tables should be formed using foreign key.**

CANDIDATE KEY:

Set of columns which uniquely identify a record.

A table can have multiple candidate key.

CANDIDATE KEY PRESENT IN 1NF TABLE:

Cname || Proname || Prosellename || Floorno || Sendname || Paymentamount

Customer Details:							
Cid (PK)	Cname	Cage	Caddress	Cgender	Cemail	Cphoneno	Ctype
C1	Ramesh	30	Mumbai	Male	Ramesh@email.com	9876543210	Buyer
C2	Sita	25	Pune	Female	sita@email.com	9876509876	Saler
C3	Anjali	28	Nashik	Female	anjali@email.com	9898989898	Buyer
C4	Arjun	32	Thane	Male	arjun@email.com	9765432109	Buyer

Property Details:							
Proid (PK)	Proname	Procity	Proprice	Procontactno	Prolistingtype	Prostatus	Category_name
101	Sea Breeze	Mumbai	5000000	8765432109	For Sale	Available	Residential
102	Hil View	Pune	3000000	9876509876	For Rent	Rented	IndependentHome
103	Green Meadows	Nashik	4000000	9888776655	For Sale	Available	Commercial
104	Royal Heights	Thane	6000000	9765123456	For Sale	Sold	Plot
105	Royal Heights	Dombivli	15000	9765123456	For Rent	Rented	PG Room

PropertySeller Details:			
Prosellerid (PK)	Prosellername	Prosellerlocation	Prosellerphoneno
1	Sunil	Mumbai	9988776655
2	Radha	Pune	9988665544
3	Kiran	Nashik	9988555522
4	Meera	Thane	9876767676

PropertyFeatures Details:				
Profeaturesid (PK)	Floorno	Areasize	Numofbedroom	Furnished
1	5	1500	2	Yes
2	NO	3200	4	Semi
3	2	2000	NO	Yes
4	NO	2500	NO	No
5	3	1000	1	Semi

PropertyInquiry Details:					
Inquiryid (PK)	Sendername	Senderphoneno	Inquirymessage	Inquirydate	Inquirystatus
1	Ramesh	9876543210	Interested in property	07-11-2024	Completed
2	Sita	9876509876	Need more details	06-11-2024	In Progress
3	Anjali	9898989898	Scheduled for site visit	08-11-2024	Pending
4	Arjun	9765432109	Request for price reduction	09-11-2024	Completed
5	Arjun	9765432109	Request for price reduction	09-11-2024	Completed

Payment Details:			
Paymentid (PK)	Paymentamount	Paymentdate	Paymentstatus
1	5000000	11-11-2024	Done
2	3000000	20-11-2024	Pending
3	4000000	18-11-2024	Pending
4	6000000	19-11-2024	Done
5	15000	19-11-2024	Done

Cid (PK)	Proid (PK)	Prosellerid (PK)	Profeaturesid (PK)	Inquiryid (PK)	Paymentid (PK)
C1	101	1	1	1	1
C2	102	2	2	2	2
C3	103	3	3	3	3
C4	104	4	4	4	4
C4	105	4	5	5	5

3NF:

- 1. Must be in 2NF.**
- 2. Direct dependency: Each non-key attribute should depend directly on the primary key.**
- 3. Avoid transitive dependency: If a non-key attribute depends on another non-key attribute, which in turn depends on the primary key, then there's a transitive dependency.**

Customer Details:							
Cid (PK)	Cname	Cage	Caddress	Cgender	Cemail	Cphoneno	Ctype
C1	Ramesh	30	Mumbai	Male	Ramesh@email.com	9876543210	Buyer
C2	Sita	25	Pune	Female	sita@email.com	9876509876	Saler
C3	Anjali	28	Nashik	Female	anjali@email.com	9898989898	Buyer
C4	Arjun	32	Thane	Male	arjun@email.com	9765432109	Buyer

Property Details:						
Proid (PK)	Prname	Procity	Proprice	Procontactno	Prolistingtype	Prostatus
101	Sea Breeze	Mumbai	5000000	8765432109	For Sale	Available
102	Hil View	Pune	3000000	9876509876	For Rent	Rented
103	Green Meadows	Nashik	4000000	9888776655	For Sale	Available
104	Royal Heights	Thane	6000000	9765123456	For Sale	Sold
105	Royal Heights	Dombivli	15000	9765123456	For Rent	Rented

PropertySeller Details:			
Prosellerid (PK)	Prosellername	Prosellerlocation	Prosellerphoneno
1	Sunil	Mumbai	9988776655
2	Radha	Pune	9988665544
3	Kiran	Nashik	9988555522
4	Meera	Thane	9876767676

PropertyFeatures Details:				
Profeaturesid (PK)	Floorno	Areasize	Numofbedroom	Furnished
1	5	1500	2	Yes
2	NO	3200	4	Semi
3	2	2000	NO	Yes
4	NO	2500	NO	No
5	3	1000	1	Semi

PropertyInquiry Details:					
Inquiryid (PK)	Sendername	Senderphoneno	Inquirymessage	Inquirydate	Inquirystatus
1	Ramesh	9876543210	Interested in property	07-11-2024	Completed
2	Sita	9876509876	Need more details	06-11-2024	In Progress
3	Anjali	9898989898	Scheduled for site visit	08-11-2024	Pending
4	Arjun	9765432109	Request for price reduction	09-11-2024	Completed
5	Arjun	9765432109	Request for price reduction	09-11-2024	Completed

Payment Details:			
Paymentid (PK)	Paymentamount	Paymentdate	Paymentstatus
1	5000000	11-11-2024	Done
2	3000000	20-11-2024	Pending
3	4000000	18-11-2024	Pending
4	6000000	19-11-2024	Done
5	15000	19-11-2024	Done

Category Details:	
Category_id (PK)	Category_name
1	Residential
2	IndependentHome
3	Commercial
4	Plot
5	PG Room

Cid (PK)	Proid (PK)	Prosellerid (PK)	Profeaturesid (PK)	Inquiryid (PK)	Paymentid (PK)	Category_id (PK)
C1	101	1	1	1	1	1
C2	102	2	2	2	2	2
C3	103	3	3	3	3	3
C4	104	4	4	4	4	4
C4	105	4	5	5	5	5

4. DATABASE QUERIES:

CONSTRAINTS

Constraints in SQL are rules applied to a column or a combination of columns in a table to enforce data integrity and ensure accuracy and reliability of the data.

1.NOT NULL:

Ensures that a column cannot contain NULL values.

Used when you want to make sure a column always has a value.

Syntax:

column_name data_type NOT NULL

Example:

```
CREATE TABLE Property (
    Proid INT PRIMARY KEY,                      -- Unique identifier for each property
    Proname VARCHAR(50) NOT NULL;                -- Name of the property, cannot be NULL
```

2.UNIQUE:

Ensures that all values in a column or set of columns are unique across the table.

Syntax:

column_name data_type UNIQUE

Example:

```
CREATE TABLE Property (Proid INT Unique);
```

3.ENUM:

Specifies that a column can only have one of a predefined set of values.

Syntax:

column_name ENUM('value1', 'value2', ...)

Example:

```
CREATE TABLE Customer (
    Cid INT PRIMARY KEY,                      -- Unique identifier for each customer
    Cgender ENUM('Male', 'Female');           -- Customer's gender, can only be 'Male' or 'Female'
```

4.PRIMARY KEY:

Uniquely identifies each record in a table.

Only one primary key is allowed per table, and it cannot contain NULL values.

Syntax:

column_name data_type PRIMARY KEY

Example:

```
CREATE TABLE Customer (Cid INT PRIMARY KEY);      -- Unique identifier for each customer
```

5.FOREIGN KEY:

Establishes a relationship between two tables by linking the foreign key column in one table to the primary key in another.

Syntax:

column_name data_type,

FOREIGN KEY (column_name) REFERENCES other_table(primary_key_column)

Example:

```
CREATE TABLE Property (
    Proid INT PRIMARY KEY,                      -- Unique identifier for each property
    Proname VARCHAR(50) NOT NULL,                -- Name of the property, cannot be NULL
    Category_id INT,                            -- Foreign key to reference the property category
    FOREIGN KEY (Category_id)
    REFERENCES propertycategory(Category_id)     -- Links to Category_id in the propertycategory table
);
```

6.DEFAULT:

Specifies that a column can only have one of a predefined set of values.

Syntax:

column_name data_type DEFAULT default_value

Example:

```
CREATE TABLE Property (
    Proid INT PRIMARY KEY,                      -- Unique identifier for each property
    Procountry VARCHAR(100) DEFAULT 'India';     -- Country of the property, defaults to 'India'
```

7.CHECK:

Specifies that a column can only have one of a predefined set of values.

Syntax:

column_name data_type CHECK (condition)

Example:

```
CREATE TABLE Customer (
    Cid INT PRIMARY KEY,          -- Unique identifier for each customer
    Cname VARCHAR(50),            -- Customer's name
    Cage INT CHECK (Cage > 18)); -- Customer's age, must be greater than 18
```

DDL (DATA DEFINITION LANGUAGE)

Data Definition Language (DDL) is used to create and modify the structure of objects in a database using predefined commands and a specific syntax.

DDL is a set of SQL commands used to **Create, Alter, Rename, Drop** and **Truncate** database structures but not data.

1. CREATE (Database & Tables):

The **CREATE TABLE** command in **SQL** allows users to define the structure of a new table. It helps users to set up the table structure by specifying **columns, datatypes and constraints**.

Syntax:

```
CREATE TABLE table_name (column1 data_type, column2 data_type, ...);
```

First, we create database that is **PrimeHeaven** then we use that database.

```
mysql> Create database PrimeHeaven;
Query OK, 1 row affected (0.02 sec)

mysql> use PrimeHeaven;
Database changed
```

Second, we create table with the use of **Create** command for entity **Customer, Property, PropertyCategory, PropertyFeatures, PropertyInquiry, PropertySeller and Payment** with some attribute.

Table1: Customer

```
CREATE TABLE Customer (
    Cid INT PRIMARY KEY,          -- Unique identifier for each customer
    Cname VARCHAR(50),            -- Customer's name
    Cage INT CHECK (Cage > 18), -- Customer's age, must be greater than 18
```

NIKHIL VANAJI WALUNJ

```

Caddress VARCHAR(50), -- Customer's address
Cgender ENUM('Male', 'Female'), -- Customer's gender, can only be 'Male' or 'Female'
Cemail VARCHAR(50), -- Customer's email address
Cpassword VARCHAR(50), -- Customer's password
Cphoneno VARCHAR(50), -- Customer's phone number
Ccity VARCHAR(50), -- City where the customer resides
Cpincode INT, -- Postal code of the customer's address
Cstate VARCHAR(50), -- State where the customer resides
Ctype ENUM('Saler', 'Buyer', 'None') -- Customer type, can be 'Saler', 'Buyer', or 'None'
);

```

```

mysql> create table Customer(Cid int primary key,Cname varchar(50),Cage int Check(Cage>18),Caddress varc
har(50),Cgender enum("Male","Female"),Cemail varchar(50),Cpassword varchar(50),Cphoneno varchar(50),Ccit
y varchar(50),Cpincode int,Cstate varchar(50),Ctype enum("Saler", "Buyer", "None"));
Query OK, 0 rows affected (0.12 sec)

mysql> desc Customer;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Cid | int | NO | PRI | NULL | |
| Cname | varchar(50) | YES | | NULL | |
| Cage | int | YES | | NULL | |
| Caddress | varchar(50) | YES | | NULL | |
| Cgender | enum('Male', 'Female') | YES | | NULL | |
| Cemail | varchar(50) | YES | | NULL | |
| Cpassword | varchar(50) | YES | | NULL | |
| Cphoneno | varchar(50) | YES | | NULL | |
| Ccity | varchar(50) | YES | | NULL | |
| Cpincode | int | YES | | NULL | |
| Cstate | varchar(50) | YES | | NULL | |
| Ctype | enum('Saler', 'Buyer', 'None') | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.02 sec)

```

Table2: Property

CREATE TABLE Property (

```

Proid INT PRIMARY KEY, -- Unique identifier for each property
Proname VARCHAR(50) NOT NULL, -- Name of the property, cannot be NULL
Prodesc VARCHAR(100), -- Description of the property
Proaddress VARCHAR(255) NOT NULL, -- Address of the property, cannot be NULL
Procity VARCHAR(100), -- City where the property is located
Prostate VARCHAR(100), -- State where the property is located
Proprice DECIMAL(15,2) NOT NULL, -- Price of the property, cannot be NULL
Procountry VARCHAR(100) DEFAULT 'India', -- Country of the property, defaults to 'India'
Procontactno VARCHAR(20), -- Contact number for property inquiries
Prolistingtype ENUM('For Sale', 'For Rent'), -- Type of listing, either 'For Sale' or 'For Rent'
Propostdate DATETIME DEFAULT CURRENT_TIMESTAMP, -- Date and time when the property was posted,
defaults to current timestamp
Prostatus ENUM('Available', 'Sold', 'Rented') DEFAULT 'Available', -- Status of the property, defaults to 'Available'

```

```

Category_id INT, -- Foreign key to reference the property category
FOREIGN KEY (Category_id) REFERENCES propertycategory(Category_id) -- Links to Category_id in the
propertycategory table
);

mysql> create table Property(Proid int primary key,Proname varchar(50) NOT NULL,Prodesc varchar(100),Proaddress varchar(255) NOT NULL,Procity varchar(100),Prostate varchar(100),Proprice decimal(15,2) NOT NULL,Procountry varchar(100) default "India",Procontactno varchar(20),Prolistingtype enum("For Sale","For Rent"),Propostdate datetime default current_timestamp,Prostatus enum('Available', 'Sold', 'Rented') DEFAULT 'Available',Category_id int,foreign key(Category_id) references propertycategory(Category_id));
Query OK, 0 rows affected (0.78 sec)

mysql> desc Property;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Proid | int  | NO   | PRI  | NULL    |          |
| Proname | varchar(50) | NO   | NULL | NULL    |          |
| Prodesc | varchar(100) | YES  | NULL | NULL    |          |
| Proaddress | varchar(255) | NO   | NULL | NULL    |          |
| Procity | varchar(100) | YES  | NULL | NULL    |          |
| Prostate | varchar(100) | YES  | NULL | NULL    |          |
| Proprice | decimal(15,2) | NO   | NULL | NULL    |          |
| Procountry | varchar(100) | YES  | NULL | India   |          |
| Procontactno | varchar(20) | YES  | NULL | NULL    |          |
| Prolistingtype | enum('For Sale','For Rent') | YES  | NULL | NULL    |          |
| Propostdate | datetime | YES  | NULL | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| Prostatus | enum('Available', 'Sold', 'Rented') | YES  | MUL  | Available |          |
| Category_id | int  | YES  | MUL  | NULL    |          |
+-----+-----+-----+-----+-----+-----+
13 rows in set (0.21 sec)

```

Table3: PropertyCategory

```

CREATE TABLE PropertyCategory (
    Category_id INT PRIMARY KEY, -- Unique identifier for each property category
    Category_name VARCHAR(50) -- Name of the property category
);

```

```

mysql> create table PropertyCategory(Category_id int primary key,Category_name varchar(50));
Query OK, 0 rows affected (0.42 sec)

mysql> desc PropertyCategory;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Category_id | int  | NO   | PRI  | NULL    |          |
| Category_name | varchar(50) | YES  | NULL | NULL    |          |
+-----+-----+-----+-----+-----+
2 rows in set (0.09 sec)

```

Table4: PropertyFeatures

```

CREATE TABLE PropertyFeatures (
    Profeatureid INT PRIMARY KEY, -- Unique identifier for each property feature entry
    Floorno VARCHAR(20), -- Floor number of the property
    Nearestlocation VARCHAR(50), -- Nearest landmark or location to the property
    Areasize VARCHAR(50), -- Size of the property area
    Numofbedroom VARCHAR(50), -- Number of bedrooms in the property
    Numofbathroom INT, -- Number of bathrooms in the property
);

```

NIKHIL VANAJI WALUNJ

Balcony ENUM('Available', 'Unavailable'), -- Indicates if a balcony is available or unavailable
 Parkingspace ENUM('Available', 'Unavailable'), -- Indicates if parking space is available or unavailable
 Gym ENUM('Available', 'Unavailable'), -- Indicates if a gym is available or unavailable
 Swimmingpool ENUM('Available', 'Unavailable') -- Indicates if a swimming pool is available or unavailable
);

```
mysql> create table PropertyFeatures(Profeatureid int primary key,Floorno varchar(20),Nearestlocation varchar(50),Areasize varchar(50),Numofbedroom varchar(50),Numofbathroom int,Balcony enum("Available", "Unavailable"),Parkingspace enum("Available", "Unavailable"),Gym enum("Available", "Unavailable"),Swimmingpool enum("Available", "Unavailable"));
Query OK, 0 rows affected (0.12 sec)

mysql> desc PropertyFeatures;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Profeatureid | int | NO | PRI | NULL |       |
| Floorno | varchar(20) | YES |       | NULL |       |
| Nearestlocation | varchar(50) | YES |       | NULL |       |
| Areasize | varchar(50) | YES |       | NULL |       |
| Numofbedroom | varchar(50) | YES |       | NULL |       |
| Numofbathroom | int | YES |       | NULL |       |
| Balcony | enum('Available', 'Unavailable') | YES |       | NULL |       |
| Parkingspace | enum('Available', 'Unavailable') | YES |       | NULL |       |
| Gym | enum('Available', 'Unavailable') | YES |       | NULL |       |
| Swimmingpool | enum('Available', 'Unavailable') | YES |       | NULL |       |
+-----+-----+-----+-----+-----+-----+
10 rows in set (0.01 sec)
```

Table5: PropertyInquiry

CREATE TABLE PropertyInquiry (

Inquiryid INT PRIMARY KEY, -- Unique identifier for each property inquiry

Proid INT, -- Foreign key to reference the property being inquired about
 FOREIGN KEY (Proid) REFERENCES Property(Proid),

SenderCid INT, -- Foreign key to reference the customer making the inquiry
 FOREIGN KEY (SenderCid) REFERENCES Customer(Cid),

Sendername VARCHAR(255) NOT NULL, -- Name of the sender (customer making the inquiry)

Senderemail VARCHAR(255) NOT NULL, -- Email of the sender

Senderphoneno VARCHAR(15), -- Phone number of the sender

Inquirymessage TEXT, -- Message content of the inquiry

Inquirydate DATETIME DEFAULT CURRENT_TIMESTAMP, -- Date and time of inquiry, defaults to current timestamp

Inquirystatus ENUM('Pending', 'In Progress', 'Completed') DEFAULT 'Pending', -- Status of the inquiry, defaults to 'Pending'

ProSellerid INT, -- Foreign key to reference the property seller
 FOREIGN KEY (ProSellerid) REFERENCES PropertySeller(ProSellerid)

);

```
mysql> create table PropertyInquiry(Inquiryid int primary key,Proid int,foreign key(Proid) references Property(Proid)
,SenderCid int,foreign key(SenderCid) references Customer(Cid),Sendername varchar(255) NOT NULL,Senderemail varchar(2
55) NOT NULL,Senderphoneno varchar(15),Inquirymessage text,Inquirydate datetime default current_timestamp,Inquirystat
us enum('Pending', 'In Progress', 'Completed') defa
ult 'Pending',ProSellerid int,foreign key(ProSellerid) references PropertySeller(ProSellerid));
Query OK, 0 rows affected (0.50 sec)
```

```
mysql> desc PropertyInquiry;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Inquiryid | int | NO | PRI | NULL |
| Proid | int | YES | MUL | NULL |
| SenderCid | int | YES | MUL | NULL |
| Sendername | varchar(255) | NO | NULL |
| Senderemail | varchar(255) | NO | NULL |
| Senderphoneno | varchar(15) | YES | NULL |
| Inquirymessage | text | YES | NULL |
| Inquirydate | datetime | YES | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
| Inquirystatus | enum('Pending', 'In Progress', 'Completed') | YES | Pending |
| ProSellerid | int | YES | MUL | NULL |
+-----+-----+-----+-----+-----+
10 rows in set (0.01 sec)
```

Table6: PropertySeller

CREATE TABLE PropertySeller (

Prosellerid INT PRIMARY KEY, -- Unique identifier for each property seller

Prosellename VARCHAR(50), -- Name of the property seller

Prosellergender ENUM('Male', 'Female'), -- Gender of the property seller, either 'Male' or 'Female'

Prosellerlocation VARCHAR(50), -- Location of the property seller

Proselleremail VARCHAR(50), -- Email of the property seller

Prosellerpassword VARCHAR(50), -- Password for the property seller's account

Prosellerphoneno VARCHAR(50) -- Phone number of the property seller

);

```
mysql> create table PropertySeller(Prosellerid int primary key,Prosellename varchar(50),
Prosellergender enum("Male","Female"),Prosellerlocation varchar(50),Proselleremail varchar(50),Prosellerpassword varchar(50),Prosellerphoneno varchar(50));
Query OK, 0 rows affected (0.03 sec)
```

```
mysql> desc PropertySeller;
```

```
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Prosellerid | int | NO | PRI | NULL |
| Prosellename | varchar(50) | YES | NULL |
| Prosellergender | enum('Male', 'Female') | YES | NULL |
| Prosellerlocation | varchar(50) | YES | NULL |
| Proselleremail | varchar(50) | YES | NULL |
| Prosellerpassword | varchar(50) | YES | NULL |
| Prosellerphoneno | varchar(50) | YES | NULL |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

Table7: Payment

```

CREATE TABLE Payment (
    Paymentid INT PRIMARY KEY, -- Unique identifier for each payment
    BuyerCid INT, -- Foreign key to reference the buyer (customer) making the payment
    FOREIGN KEY (BuyerCid) REFERENCES Customer(Cid),
    Proid INT, -- Foreign key to reference the property involved in the transaction
    FOREIGN KEY (Proid) REFERENCES Property(Proid),
    ProSellerid INT, -- Foreign key to reference the property seller
    FOREIGN KEY (ProSellerid) REFERENCES PropertySeller(ProSellerid),
    Paymentamount DECIMAL(10,2) NOT NULL, -- Amount of the payment, cannot be NULL
    Paymentmethod VARCHAR(100), -- Method of payment (e.g., credit card, bank transfer)
    Paymentstatus ENUM('Done', 'Pending') DEFAULT 'Pending', -- Status of the payment, defaults to 'Pending'
    Paymentdate DATETIME DEFAULT CURRENT_TIMESTAMP -- Date and time of the payment, defaults to current timestamp
);

```

```

mysql> create table Payment(Paymentid int primary key,BuyerCid int,foreign key(BuyerCid) references Customer(Cid),Proid int,foreign key(Proid) references Property(Proid),ProSellerid int,foreign key(ProSellerid) references PropertySeller(ProSellerid),Paymentamount decimal(10,2) NOT NULL,Paymentmethod varchar(100),Paymentstatus enum("Done","Pending") default "Pending",Paymentdate datetime default current_timestamp);
Query OK, 0 rows affected (0.08 sec)

mysql> desc Payment;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Paymentid | int | NO | PRI | NULL |       |
| BuyerCid | int | YES | MUL | NULL |       |
| Proid | int | YES | MUL | NULL |       |
| ProSellerid | int | YES | MUL | NULL |       |
| Paymentamount | decimal(10,2) | NO |       | NULL |       |
| Paymentmethod | varchar(100) | YES |       | NULL |       |
| Paymentstatus | enum('Done', 'Pending') | YES |       | Pending |       |
| Paymentdate | datetime | YES |       | CURRENT_TIMESTAMP | DEFAULT_GENERATED |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

```

2. ALTER (Table):

Modifying the structure of a table in databases involves altering its schema, which includes adding, modifying, or dropping columns, constraints, or indexes.

1) Add column in existing table:

Syntax:

```
ALTER TABLE table_name ADD column_name data_type;
```

Query:

```
ALTER TABLE PropertyFeatures
```

```
ADD Furnished ENUM('Yes', 'No', 'Semi');
```

```
mysql> alter table PropertyFeatures add Furnished enum('Yes', 'No', 'Semi');
Query OK, 0 rows affected (0.44 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> desc PropertyFeatures;
```

Field	Type	Null	Key	Default	Extra
Profeatureid	int	NO	PRI	NULL	
Floorno	varchar(20)	YES		NULL	
Nearestlocation	varchar(50)	YES		NULL	
Areasize	varchar(50)	YES		NULL	
Numofbedroom	varchar(50)	YES		NULL	
Numofbathroom	int	YES		NULL	
Balcony	enum('Available','Unavailable')	YES		NULL	
Parkingspace	enum('Available','Unavailable')	YES		NULL	
Gym	enum('Available','Unavailable')	YES		NULL	
Swimmingpool	enum('Available','Unavailable')	YES		NULL	
Furnished	enum('Yes','No','Semi')	YES		NULL	

11 rows in set (0.00 sec)

2) Add column after specific column:

Syntax:

```
ALTER TABLE table_name ADD column_name data_type AFTER column_name;
```

Query:

```
ALTER TABLE PropertyFeatures
```

```
ADD GardenArea enum('YES','NO') AFTER parkingspace;
```

```
mysql> alter table PropertyFeatures add GardenArea enum('YES','NO') after parkingspace;
Query OK, 0 rows affected (2.08 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> desc PropertyFeatures;
```

Field	Type	Null	Key	Default	Extra
Profeatureid	int	NO	PRI	NULL	
Floorno	varchar(20)	YES		NULL	
Nearestlocation	varchar(50)	YES		NULL	
Areasize	varchar(50)	YES		NULL	
Numofbedroom	varchar(50)	YES		NULL	
Numofbathroom	int	YES		NULL	
Balcony	enum('Available','Unavailable')	YES		NULL	
Parkingspace	enum('Available','Unavailable')	YES		NULL	
GardenArea	enum('YES','NO')	YES		NULL	
Gym	enum('Available','Unavailable')	YES		NULL	
Swimmingpool	enum('Available','Unavailable')	YES		NULL	
Furnished	enum('Yes','No','Semi')	YES		NULL	

12 rows in set (0.09 sec)

3) Add column at first position in table:

Syntax:

```
ALTER TABLE table_name ADD column_name data_type FIRST;
```

Query:

```
ALTER TABLE PropertyFeatures ADD Profeatureid int primary key FIRST;
```

```
mysql> alter table PropertyFeatures add Profeatureid int primary key first;
Query OK, 0 rows affected (0.61 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc PropertyFeatures;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Profeatureid | int | NO | PRI | NULL |
| Floorno | varchar(20) | YES | NULL |
| Nearestlocation | varchar(50) | YES | NULL |
| Areasize | varchar(50) | YES | NULL |
| Numofbedroom | varchar(50) | YES | NULL |
| Numofbathroom | int | YES | NULL |
| Balcony | enum('Available','Unavailable') | YES | NULL |
| Parkingspace | enum('Available','Unavailable') | YES | NULL |
| GardenArea | enum('YES','NO') | YES | NULL |
| Gym | enum('Available','Unavailable') | YES | NULL |
| Swimmingpool | enum('Available','Unavailable') | YES | NULL |
| Furnished | enum('Yes','No','Semi') | YES | NULL |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.01 sec)
```

4) Change data type of existing column:

Syntax:

```
ALTER TABLE table_name MODIFY column_name data_type;
```

Query:

```
ALTER TABLE PropertyFeatures MODIFY Numofbedroom varchar(100);
```

```
mysql> alter table PropertyFeatures modify Numofbedroom varchar(100);
Query OK, 0 rows affected (0.72 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc PropertyFeatures;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Profeatureid | int | NO | PRI | NULL |
| Floorno | varchar(20) | YES | NULL |
| Nearestlocation | varchar(50) | YES | NULL |
| Areasize | varchar(50) | YES | NULL |
| Numofbedroom | varchar(100) | YES | NULL |
| Numofbathroom | int | YES | NULL |
| Balcony | enum('Available','Unavailable') | YES | NULL |
| Parkingspace | enum('Available','Unavailable') | YES | NULL |
| GardenArea | enum('YES','NO') | YES | NULL |
| Gym | enum('Available','Unavailable') | YES | NULL |
| Swimmingpool | enum('Available','Unavailable') | YES | NULL |
| Furnished | enum('Yes','No','Semi') | YES | NULL |
+-----+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```

5) Change existing column name:

Syntax:

```
ALTER TABLE    table_name    CHANGE    COLUMN    existing_column_name
new_column_name data_type;
```

Query:

```
ALTER TABLE PropertyFeatures
```

```
CHANGE Areasize Property_Areasize varchar(100);
```

```
mysql> alter table PropertyFeatures change Areasize Property_Areasize varchar(100);
Query OK, 0 rows affected (0.41 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> desc PropertyFeatures;
```

Field	Type	Null	Key	Default	Extra
Profeatureid	int	NO	PRI	NULL	
Floorno	varchar(20)	YES		NULL	
Nearestlocation	varchar(50)	YES		NULL	
Property_Areasize	varchar(100)	YES		NULL	
Numofbedroom	varchar(100)	YES		NULL	
Numofbathroom	int	YES		NULL	
Balcony	enum('Available', 'Unavailable')	YES		NULL	
Parkingspace	enum('Available', 'Unavailable')	YES		NULL	
GardenArea	enum('YES', 'NO')	YES		NULL	
Gym	enum('Available', 'Unavailable')	YES		NULL	
Swimmingpool	enum('Available', 'Unavailable')	YES		NULL	
Furnished	enum('Yes', 'No', 'Semi')	YES		NULL	

```
12 rows in set (0.00 sec)
```

6) Add multiple columns at same time in table:

Syntax:

```
ALTER TABLE tablename ADD column_name data_type, ADD column_name data_type,..;
```

Query:

```
ALTER TABLE PropertyFeatures
```

```
ADD Furnished ENUM('Yes', 'No', 'Semi'),
```

```
ADD GardenArea ENUM('YES', 'NO');
```

7) Rearrange column in table:

Syntax:

```
ALTER TABLE  table_name  MODIFY  column  column_name  data_type  AFTER
column_name;
```

Query:

ALTER TABLE PropertyFeatures

MODIFY column Property_Areaseize varchar(100)

AFTER Profeatureid;

```
mysql> alter table PropertyFeatures modify column Property_Areaseize varchar(100) after Profeatureid;
Query OK, 0 rows affected (0.45 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc PropertyFeatures;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Profeatureid | int | NO | PRI | NULL |
| Property_Areaseize | varchar(100) | YES | NULL |
| Floorno | varchar(20) | YES | NULL |
| Nearestlocation | varchar(50) | YES | NULL |
| Numofbedroom | varchar(100) | YES | NULL |
| Numofbathroom | int | YES | NULL |
| Balcony | enum('Available','Unavailable') | YES | NULL |
| Parkingspace | enum('Available','Unavailable') | YES | NULL |
| GardenArea | enum('YES','NO') | YES | NULL |
| Gym | enum('Available','Unavailable') | YES | NULL |
| Swimmingpool | enum('Available','Unavailable') | YES | NULL |
| Furnished | enum('Yes','No','Semi') | YES | NULL |
+-----+-----+-----+-----+-----+
12 rows in set (0.14 sec)
```

8) Add Primary Key to existing column:

Syntax:

ALTER TABLE table_name ADD primary key(column_name);

Query:

ALTER TABLE PropertyFeatures

ADD primary key(Profeatureid);

```
mysql> alter table PropertyFeatures add primary key(Profeatureid);
Query OK, 0 rows affected (0.63 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc PropertyFeatures;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Profeatureid | int | NO | PRI | NULL |
| Property_Areaseize | varchar(100) | YES | NULL |
| Floorno | varchar(20) | YES | NULL |
| Nearestlocation | varchar(50) | YES | NULL |
| Numofbedroom | varchar(100) | YES | NULL |
| Numofbathroom | int | YES | NULL |
| Balcony | enum('Available','Unavailable') | YES | NULL |
| Parkingspace | enum('Available','Unavailable') | YES | NULL |
| GardenArea | enum('YES','NO') | YES | NULL |
| Gym | enum('Available','Unavailable') | YES | NULL |
| Swimmingpool | enum('Available','Unavailable') | YES | NULL |
| Furnished | enum('Yes','No','Semi') | YES | NULL |
+-----+-----+-----+-----+-----+
12 rows in set (0.00 sec)
```

3. RENAME (Table):

Rename a table existing in the database.

Syntax:

```
RENAME TABLE old_table_name TO new_table_name;
```

Query:

```
RENAME TABLE PropertyFeatures TO Property_Features;
```

```
mysql> rename table PropertyFeatures To Property_Features;
Query OK, 0 rows affected (0.22 sec)

mysql> desc PropertyFeatures;
ERROR 1146 (42S02): Table 'primeheaven.propertyfeatures' doesn't exist
mysql> desc Property_Features;
+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| Profeatureid | int | NO | PRI | NULL |
| Property_Areasize | varchar(100) | YES | NULL |
| Floorno | varchar(20) | YES | NULL |
| Nearestlocation | varchar(50) | YES | NULL |
| Numofbedroom | varchar(100) | YES | NULL |
| Numofbathroom | int | YES | NULL |
| Balcony | enum('Available','Unavailable') | YES | NULL |
| Parkingspace | enum('Available','Unavailable') | YES | NULL |
| GardenArea | enum('YES','NO') | YES | NULL |
| Gym | enum('Available','Unavailable') | YES | NULL |
| Swimmingpool | enum('Available','Unavailable') | YES | NULL |
| Furnished | enum('Yes','No','Semi') | YES | NULL |
+-----+-----+-----+-----+-----+
12 rows in set (0.03 sec)
```

4. DROP (Table):

Delete objects/table from the database.

Dropping a table in SQL means permanently removing the table's structure and all its data from the database.

(A) Delete whole table

Syntax for one table:

```
DROP TABLE table_name;
```

Query:

```
DROP TABLE PropertyFeatures;
```

Syntax for multiple table:

```
DROP TABLE table_name1,table_name2,table_name3,...;
```

Query:

```
DROP TABLE PropertyFeatures,Property,Payment;
```

(B) Delete particular column from table

Syntax:

```
ALTER TABLE table_name DROP column column_name;
```

Query:

```
ALTER TABLE PropertyFeatures
```

```
DROP column GardenArea;
```

```
mysql> alter table PropertyFeatures drop column GardenArea;
Query OK, 0 rows affected (0.10 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> desc PropertyFeatures;
+-----+-----+-----+-----+-----+-----+
| Field        | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| Profeatureid | int       | NO   | PRI | NULL    |       |
| Property_Areasize | varchar(100) | YES  |     | NULL    |       |
| Floorno      | varchar(20)  | YES  |     | NULL    |       |
| Nearestlocation | varchar(50) | YES  |     | NULL    |       |
| Numofbedroom  | varchar(100) | YES  |     | NULL    |       |
| Numofbathroom | int       | YES  |     | NULL    |       |
| Balcony       | enum('Available','Unavailable') | YES  |     | NULL    |       |
| Parkingspace  | enum('Available','Unavailable') | YES  |     | NULL    |       |
| Gym           | enum('Available','Unavailable') | YES  |     | NULL    |       |
| Swimmingpool  | enum('Available','Unavailable') | YES  |     | NULL    |       |
| Furnished     | enum('Yes','No','Semi')        | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

5. Truncate (Table):

Remove all records from a table, including all spaces allocated for the records are removed.

The Truncate Command is used to delete the data inside a table, but not the table itself.

Syntax:

```
TRUNCATE TABLE table_name;
```

Query:

```
TRUNCATE TABLE PropertyFeatures;
```

DML (DATA MANIPULATION LANGUAGE)

Data Manipulation Language (DML) is a subset of SQL (Structured Query Language) that is used to manage and manipulate data within database tables.

DML commands are responsible for operation like **INSERT, UPDATE and DELETE** on the data.

1. INSERT statement:

The **INSERT** statement is used in SQL to add new records (rows) into a table. It allows you to insert one or more rows of data into a specified table.

Syntax:

```
INSERT INTO table_name (column1, column2, ...)  
VALUES (value1, value2, ...);
```

1.1 Insert value in Customer table:

Query:

```
mysql> INSERT INTO customer (Cid, Cname, Cage, Caddress, Cgender, Cemail, Cpassword, Cphoneno, Ccity, Cpincode, Cstate, Ctype) VALUES  
> (1, 'Rohit Sharma', 30, '21 Marine Drive', 'Male', 'rohit.sharma@gmail.com', 'rohit@123', '9876543211', 'Mumbai', 400020, 'Maharashtra', 'Buyer'),  
> (2, 'Sakshi Mehta', 28, '12 Vasant Kunj', 'Female', 'sakshi.mehta@gmail.com', 'sakshi@123', '9765432102', 'Delhi', 110070, 'Delhi', 'Saler'),  
> (3, 'Vikram Desai', 35, '43 JP Nagar', 'Male', 'vikram.desai@gmail.com', 'vikram@123', '9876043124', 'Bangalore', 560078, 'Karnataka', 'Buyer'),  
> (4, 'Priya Reddy', 26, '17 Hitec City', 'Female', 'priya.reddy@gmail.com', 'priya@123', '9876087651', 'Hyderabad', 500081, 'Telangana', 'Saler'),  
> (5, 'Arjun Singh', 32, '98 South Avenue', 'Male', 'arjun.singh@gmail.com', 'arjun@123', '9876532198', 'Delhi', 110011, 'Delhi', 'Buyer'),  
> (6, 'Neha Sharma', 27, '55 Powai', 'Female', 'neha.sharma@gmail.com', 'neha@123', '9876012347', 'Mumbai', 400076, 'Maharashtra', 'Saler'),  
> (7, 'Kiran Patel', 34, '14 Maninagar', 'Male', 'kiran.patel@gmail.com', 'kiran@123', '9876554322', 'Ahmedabad', 380008, 'Gujarat', 'Buyer'),  
> (8, 'Pooja Bhatt', 30, '27 Anna Nagar', 'Female', 'pooja.bhatt@gmail.com', 'pooja@123', '9876087656', 'Chennai', 600040, 'Tamil Nadu', 'Saler'),  
> (9, 'Rahul Joshi', 31, '75 MG Road', 'Male', 'rahul.joshi@gmail.com', 'rahul@123', '9876078452', 'Pune', 411001, 'Maharashtra', 'Buyer'),  
> (10, 'Swati Desai', 29, '66 Alkapuri', 'Female', 'swati.desai@gmail.com', 'swati@123', '9876543287', 'Vadodara', 390007, 'Gujarat', 'Saler'),  
> (11, 'Vivek Sharma', 38, '15 Connaught Place', 'Male', 'vivek.sharma@gmail.com', 'vivek@123', '9876076534', 'Delhi', 110001, 'Delhi', 'Buyer'),  
> (12, 'Ritika Nair', 33, '44 Whitefield', 'Female', 'ritika.nair@gmail.com', 'ritika@123', '9876021236', 'Bangalore', 560066, 'Karnataka', 'Buyer'),  
> (13, 'Rajesh Kumar', 36, '5 Janakpuri', 'Male', 'rajesh.kumar@gmail.com', 'rajesh@123', '9876043217', 'Delhi', 110058, 'Delhi', 'Saler'),  
> (14, 'Sunita Tiwari', 27, '18 Sector 56', 'Female', 'sunita.tiwari@gmail.com', 'sunita@123', '9876098763', 'Noida', 201301, 'Uttar Pradesh', 'Buyer'),  
> (15, 'Amit Agarwal', 40, '29 Salt Lake', 'Male', 'amit.agarwal@gmail.com', 'amit@123', '9876067891', 'Kolkata', 700091, 'West Bengal', 'Buyer'),  
> (16, 'Anjali Verma', 31, '37 Sector 10', 'Female', 'anjali.verma@gmail.com', 'anjali@123', '9876087653', 'Noida', 201301, 'Uttar Pradesh', 'Saler'),  
> (17, 'Manish Chawla', 29, '22 DLF Phase 2', 'Male', 'manish.chawla@gmail.com', 'manish@123', '9876054323', 'Gurgaon', 122002, 'Haryana', 'Buyer'),  
> (18, 'Geeta Iyer', 26, '11 Koramangala', 'Female', 'geeta.iyer@gmail.com', 'geeta@123', '9876021239', 'Bangalore', 560095, 'Karnataka', 'None'),  
> (19, 'Nikhil Sinha', 34, '19 Connaught Place', 'Male', 'nikhil.sinha@gmail.com', 'nikhil@123', '9876078941', 'Delhi', 110001, 'Delhi', 'Buyer'),
```

NIKHIL VANAJI WALUNJ

```
-> (20, 'Divya Shah', 30, '3 Ellisbridge', 'Female', 'divya.shah@gmail.com', 'divya@123', '9876076523', 'Ahmedabad', 380006, 'Gujarat', 'Saler'),  
-> (21, 'Ramesh Mehta', 32, '8 Sector 15', 'Male', 'ramesh.mehta@gmail.com', 'ramesh@123', '9876012348', 'Chandigarh', 160015, 'Punjab', 'Buyer'),  
-> (22, 'Meera Khan', 28, '23 Kalina', 'Female', 'meera.khan@gmail.com', 'meera@123', '9876065421', 'Mumbai', 400098, 'Maharashtra', 'Saler'),  
-> (23, 'Suresh Yadav', 30, '67 Baner', 'Male', 'suresh.yadav@gmail.com', 'suresh@123', '9876098769', 'Pune', 411045, 'Maharashtra', 'Buyer'),  
-> (24, 'Shalini Ghosh', 29, '77 Salt Lake', 'Female', 'shalini.ghosh@gmail.com', 'shalini@123', '9876543215', 'Kolkata', 700064, 'West Bengal', 'Saler'),  
-> (25, 'Mohit Singh', 33, '10 Golf Course Road', 'Male', 'mohit.singh@gmail.com', 'mohit@123', '9876087657', 'Gurgaon', 122003, 'Haryana', 'Buyer'),  
-> (26, 'Anupam Bhatia', 37, '91 Ballygunge', 'Male', 'anupam.bhatia@gmail.com', 'anupam@123', '9876098768', 'Kolkata', 700019, 'West Bengal', 'None'),  
-> (27, 'Ravi Rao', 29, '5 Jayanagar', 'Male', 'ravi.rao@gmail.com', 'ravi@123', '9876076524', 'Bangalore', 560041, 'Karnataka', 'Buyer');
```

Query OK, 27 rows affected (0.47 sec)

Records: 27 Duplicates: 0 Warnings: 0

1.2 Insert value in PropertyCategory table:

Query:

```
mysql> INSERT INTO PropertyCategory (Category_id, Category_name) VALUES
```

```
-> (1, 'Residential'),  
-> (2, 'IndependentHome'),  
-> (3, 'Commercial'),  
-> (4, 'Plots'),  
-> (5, 'PG Room');
```

Query OK, 5 rows affected (0.50 sec)

Records: 5 Duplicates: 0 Warnings: 0

1.3 Insert value in PropertySeller table:

Query:

```
mysql> INSERT INTO PropertySeller (Prosellerid, Prosellename, Prosellergender, Prosellerlocation, Proselleremail, Prosellerpassword, Prosellerphoneno) VALUES
```

```
-> (1, 'Sakshi Mehta', 'Female', 'Vasant Kunj', 'sakshi.mehta@gmail.com', 'sakshi@123', '9765432102'),  
-> (2, 'Priya Reddy', 'Female', 'Hitec City', 'priya.reddy@gmail.com', 'priya@123', '9876087651'),  
-> (3, 'Neha Sharma', 'Female', 'Powai', 'neha.sharma@gmail.com', 'neha@123', '9876012347'),  
-> (4, 'Pooja Bhatt', 'Female', 'Anna Nagar', 'pooja.bhatt@gmail.com', 'pooja@123', '9876087656'),  
-> (5, 'Swati Desai', 'Female', 'Alkapuri', 'swati.desai@gmail.com', 'swati@123', '9876543287'),  
-> (6, 'Rajesh Kumar', 'Male', 'Janakpuri', 'rajesh.kumar@gmail.com', 'rajesh@123', '9876043217'),  
-> (7, 'Anjali Verma', 'Female', 'Sector 10', 'anjali.verma@gmail.com', 'anjali@123', '9876087653'),  
-> (8, 'Divya Shah', 'Female', 'Ellisbridge', 'divya.shah@gmail.com', 'divya@123', '9876076523'),  
-> (9, 'Meera Khan', 'Female', 'Kalina', 'meera.khan@gmail.com', 'meera@123', '9876065421'),  
-> (10, 'Shalini Ghosh', 'Female', 'Salt Lake', 'shalini.ghosh@gmail.com', 'shalini@123', '9876543215');
```

Query OK, 10 rows affected (0.05 sec)

Records: 10 Duplicates: 0 Warnings: 0

1.4 Insert value in Property table:

NIKHIL VANAJI WALUNJ

Query:

```
mysql> INSERT INTO Property (Proid, Proname, Prodesc, Proaddress, Procity, Proprice, Procountry, Procontactno, Prolistingtype, Prostatus, Category_id)
-> VALUES
-> (1, 'Luxury 2BHK Apartment', 'A spacious 2BHK apartment in a prime location.', '123 Park Street, Mumbai', 'Mumbai', 'Maharashtra', 7500000.00, 'India',
'9876543210', 'For Sale', 'Available', 1),
-> (2, 'Green View Independent House', 'A beautiful independent house with a garden and ample space.', '45 Green Lane, Pune', 'Pune', 'Maharashtra', 12000000.00,
'India', '9898989898', 'For Sale', 'Available', 2),
-> (3, 'Commercial Office Space', 'A commercial office space located in the business hub.', '10 Business Park, Delhi', 'Delhi', 'Delhi', 25000000.00, 'India',
'9999999999', 'For Rent', 'Available', 3),
-> (4, 'Residential Plot', 'A well-located residential plot in a developing area.', '60 Greenfield, Bangalore', 'Bangalore', 'Karnataka', 5000000.00, 'India',
'9876543210', 'For Sale', 'Available', 4),
-> (5, 'PG Room for Rent', 'A fully furnished PG room with all amenities included.', '101 Comfort Street, Chennai', 'Chennai', 'Tamil Nadu', 15000.00, 'India',
'9500000000', 'For Rent', 'Available', 5),
-> (6, 'Oceanview 3BHK', 'A luxury 3BHK apartment with ocean views and modern amenities.', '234 Sea Breeze Road, Goa', 'Goa', 'Goa', 9500000.00, 'India',
'8888888888', 'For Sale', 'Available', 1),
-> (7, 'Villa in Green Valley', 'A spacious villa with a private pool and large garden.', '12 Green Valley, Ahmedabad', 'Ahmedabad', 'Gujarat', 18000000.00, 'India',
'9612345678', 'For Sale', 'Available', 2),
-> (8, 'Tech Hub Office', 'An office space designed for tech companies with modern infrastructure.', '56 Silicon Street, Hyderabad', 'Hyderabad', 'Telangana',
15000000.00, 'India', '9176543210', 'For Rent', 'Available', 3),
-> (9, 'Luxury Plot in Prime Area', 'A plot for residential development in a prime location.', '500 Main Road, Kolkata', 'Kolkata', 'West Bengal', 8000000.00, 'India',
'9712345678', 'For Sale', 'Available', 4),
-> (10, 'PG Room Near IT Park', 'A comfortable PG room with food and laundry services.', '78 IT Park Lane, Bangalore', 'Bangalore', 'Karnataka', 12000.00, 'India',
'9512345678', 'For Rent', 'Available', 5),
-> (11, 'Modern 2BHK Apartment', 'Modern apartment with all necessary amenities.', '15 Sunrise Avenue, Mumbai', 'Mumbai', 'Maharashtra', 5000000.00, 'India',
'9369876543', 'For Sale', 'Available', 1),
-> (12, 'Independent House with Garden', 'Spacious independent house with a beautiful garden.', '23 Hilltop Lane, Pune', 'Pune', 'Maharashtra', 8500000.00, 'India',
'9843276543', 'For Sale', 'Available', 2),
-> (13, 'Commercial Space for Retail', 'A commercial space located in a prime shopping area.', '80 Mall Road, Delhi', 'Delhi', 'Delhi', 12000000.00, 'India',
'9543678921', 'For Rent', 'Available', 3),
-> (14, 'Residential Plot for Sale', 'Prime location residential plot available for sale.', '900 City Center, Bangalore', 'Bangalore', 'Karnataka', 6500000.00, 'India',
'9643721890', 'For Sale', 'Available', 4),
-> (15, 'PG Room with AC', 'Well-maintained PG room with air conditioning and other amenities.', '30 Liberty Road, Chennai', 'Chennai', 'Tamil Nadu', 18000.00,
'India', '9445678910', 'For Rent', 'Available', 5),
-> (16, 'Luxury 1BHK Apartment', 'Stylish and compact 1BHK apartment in a premium location.', '40 Skyline Building, Delhi', 'Delhi', 'Delhi', 3500000.00, 'India',
'9912345678', 'For Sale', 'Available', 1),
-> (17, 'Detached Independent Home', 'A beautiful independent home with a spacious backyard.', '77 Sunset Blvd, Ahmedabad', 'Ahmedabad', 'Gujarat',
15000000.00, 'India', '9976543210', 'For Sale', 'Available', 2),
-> (18, 'Co-Working Office Space', 'A modern co-working space with high-speed internet and conference rooms.', '12 Tech Valley, Hyderabad', 'Hyderabad',
'Telangana', 8500000.00, 'India', '9176543210', 'For Rent', 'Available', 3),
-> (19, 'Prime Residential Plot', 'A plot with excellent development potential in a growing neighborhood.', '55 North Street, Kolkata', 'Kolkata', 'West Bengal',
4000000.00, 'India', '9800000000', 'For Sale', 'Available', 4),
-> (20, 'PG Room in Safe Area', 'Safe and secure PG room with easy access to public transport.', '60 Safe Lane, Pune', 'Pune', 'Maharashtra', 10000.00, 'India',
'9823456789', 'For Rent', 'Available', 5);
Query OK, 20 rows affected (1.83 sec)
Records: 20 Duplicates: 0 Warnings: 0
```

1.5 Insert value in PropertyFeatures table:

Query:

```
mysql> INSERT INTO PropertyFeatures (Profeatureid, Property_Areasize, Floorno, Nearestlocation, Numofbedroom, Numofbathroom, Balcony, Parkingspace,
Gym, Swimmingpool, Furnished)
```

NIKHIL VANAJI WALUNJ

-> VALUES

```
-> (1, '1200 sq ft', '5th', 'Park Street, Mumbai', '2', 2, 'Available', 'Available', 'Available', 'Unavailable', 'Semi'),
-> (2, '2000 sq ft', '2nd', 'Green Lane, Pune', '3', 3, 'Available', 'Unavailable', 'Unavailable', 'Unavailable', 'No'),
-> (3, '3000 sq ft', '8th', 'Business Park, Delhi', '0', 2, 'Unavailable', 'Available', 'Unavailable', 'Unavailable', 'No'),
-> (4, '1500 sq ft', 'Ground', 'Greenfield, Bangalore', '0', 0, 'Unavailable', 'Available', 'Unavailable', 'Unavailable', 'No'),
-> (5, '250 sq ft', '1st', 'Comfort Street, Chennai', '1', 1, 'Available', 'Unavailable', 'Unavailable', 'Unavailable', 'Yes'),
-> (6, '1800 sq ft', '10th', 'Sea Breeze Road, Goa', '3', 2, 'Available', 'Available', 'Available', 'Available', 'Semi'),
-> (7, '3500 sq ft', '4th', 'Green Valley, Ahmedabad', '4', 4, 'Available', 'Available', 'Unavailable', 'Available', 'Yes'),
-> (8, '4000 sq ft', '6th', 'Silicon Street, Hyderabad', '0', 3, 'Unavailable', 'Available', 'Available', 'Unavailable', 'No'),
-> (9, '5000 sq ft', '7th', 'Main Road, Kolkata', '0', 0, 'Unavailable', 'Unavailable', 'Unavailable', 'Unavailable', 'No'),
-> (10, '350 sq ft', '3rd', 'IT Park Lane, Bangalore', '1', 1, 'Unavailable', 'Available', 'Unavailable', 'Unavailable', 'Yes'),
-> (11, '1100 sq ft', '12th', 'Sunrise Avenue, Mumbai', '2', 2, 'Available', 'Available', 'Unavailable', 'Available', 'Semi'),
-> (12, '2200 sq ft', '5th', 'Hilltop Lane, Pune', '3', 2, 'Unavailable', 'Available', 'Unavailable', 'Unavailable', 'No'),
-> (13, '3500 sq ft', '3rd', 'Mall Road, Delhi', '0', 1, 'Available', 'Available', 'Unavailable', 'Unavailable', 'No'),
-> (14, '1800 sq ft', 'Ground', 'City Center, Bangalore', '0', 0, 'Unavailable', 'Unavailable', 'Unavailable', 'Unavailable', 'No'),
-> (15, '200 sq ft', '4th', 'Liberty Road, Chennai', '1', 1, 'Available', 'Unavailable', 'Unavailable', 'Unavailable', 'Yes'),
-> (16, '900 sq ft', '15th', 'Skyline Building, Delhi', '1', 1, 'Unavailable', 'Available', 'Unavailable', 'Unavailable', 'Semi'),
-> (17, '3500 sq ft', '2nd', 'Sunset Blvd, Ahmedabad', '4', 3, 'Available', 'Available', 'Available', 'Yes'),
-> (18, '1500 sq ft', '9th', 'Tech Valley, Hyderabad', '0', 2, 'Unavailable', 'Available', 'Available', 'Unavailable', 'No'),
-> (19, '5000 sq ft', 'Ground', 'North Street, Kolkata', '0', 0, 'Unavailable', 'Unavailable', 'Unavailable', 'Unavailable', 'No'),
-> (20, '300 sq ft', '1st', 'Safe Lane, Pune', '1', 1, 'Available', 'Unavailable', 'Unavailable', 'Unavailable', 'Yes');
```

Query OK, 20 rows affected (0.07 sec)

Records: 20 Duplicates: 0 Warnings: 0

1.6 Insert value in PropertyInquiry table:

Query:

```
mysql> INSERT INTO PropertyInquiry (Inquiryid, Proid, SenderCid, Sendername, Senderemail, Senderphoneno, Inquirymessage, Inquirystatus, ProSellerid)
-> VALUES
-> (1, 2, 9, 'rahul.joshi@gmail.com', 'Rahul Joshi', '9876078452', 'Can you schedule a viewing for this property?', 'Pending', 1),
-> (2, 6, 1, 'rohit.sharma@gmail.com', 'Rohit Sharma', '9876543211', 'I am interested in this property. Please provide more details.', 'In Progress', 3),
-> (3, 10, 19, 'nikhil.sinha@gmail.com', 'Nikhil Sinha', '9876078941', 'What are the maintenance costs for this property?', 'Completed', 8),
-> (4, 15, 5, 'arjun.singh@gmail.com', 'Arjun Singh', '9876532198', 'What is the expected closing date for this property?', 'Completed', 6),
-> (5, 19, 14, 'sunita.tiwari@gmail.com', 'Sunita Tiwari', '9876098763', 'Is this property pet-friendly?', 'Pending', 10);
```

Query OK, 5 rows affected (0.01 sec)

Records: 5 Duplicates: 0 Warnings: 0

1.7 Insert value in Payment table:

Query:

```
mysql> INSERT INTO Payment (Paymentid, BuyerCid, Proid, ProSellerid, Paymentamount, Paymentmethod, Paymentstatus)
```

```
-> VALUES
-> (1, 1, 3, 1, 500000, 'Credit Card', 'Done'),
-> (2, 7, 7, 3, 750000, 'Bank Transfer', 'Pending'),
-> (3, 12, 2, 5, 1200000, 'Cash', 'Done'),
-> (4, 15, 12, 7, 600000, 'Debit Card', 'Pending'),
-> (5, 19, 5, 9, 450000, 'UPI', 'Done'),
-> (6, 5, 20, 6, 800000, 'Credit Card', 'Done'),
-> (7, 25, 15, 2, 1000000, 'Bank Transfer', 'Pending');
```

Query OK, 7 rows affected (0.01 sec)

Records: 7 Duplicates: 0 Warnings: 0

2. UPDATE statement:

The **UPDATE** statement in SQL is used to modify existing records (rows) in a table. It allows you to update one or more columns in a table with new values based on specified conditions.

Syntax:

UPDATE table_name

SET column1 = value1, column2 = value2,

WHERE condition;

2.1 Update value in Property table:

Proid	Proname	Prodesc	Proaddress	Procity	Prostate	Propri	Procountry	Procontactno	Prolistingtype	Propostdate	Prostatus	Category_id	Profeatureid
1	Luxury 2BHK Apartment	A spacious 2BHK apartment in a prime location.	123 Park Street, Mumbai	Mumbai	Maharashtra	7500000.00	India	9876543210	For Sale	2024-11-11 20:51:32	Available	1	NULL
2	Green View Independent House	A beautiful independent house with a garden and ample space.	45 Green Lane, Pune	Pune	Maharashtra	12000000.00	India	9898989898	For Sale	2024-11-11 20:51:32	Available	2	NULL
3	Commercial Office Space	A commercial office space located in the business hub.	10 Business Park, Delhi	Delhi	Delhi	2500000.00	India	9999999999	For Rent	2024-11-11 20:51:32	Available	3	NULL
4	Residential Plot	A well-located residential plot in a developing area.	68 Greenfield, Bangalore	Bangalore	Karnataka	5000000.00	India	9876543210	For Sale	2024-11-11 20:51:32	Available	4	NULL
5	PG Room in PG Area	A PG room available for rent in a safe area.	1100 Sector 10, Gurgaon	Gurgaon	Haryana	100000.00	India	9898989898	For Rent	2024-11-11 20:51:32	Available	5	NULL
6	Oceanview 3BHK	A luxury 3BHK apartment with ocean views and modern amenities.	234 Sea Breeze Road, Goa	Goa	Goa	9500000.00	India	8888888888	For Sale	2024-11-11 20:51:32	Available	1	NULL
7	Villa in Green Valley	A spacious villa with a private pool and large garden.	12 Green Valley, Ahmedabad	Ahmedabad	Gujarat	18000000.00	India	9612345678	For Sale	2024-11-11 20:51:32	Available	2	NULL
8	Tech Hub Office	An office space designed for tech companies with modern infrastructure.	56 Silicon Street, Hyderabad	Hyderabad	Telangana	1500000.00	India	9176543210	For Rent	2024-11-11 20:51:32	Available	3	NULL
9	Luxury Plot in Pimpri Area	A plot for residential development in a prime location.	580 Main Road, Kolkata	Kolkata	West Bengal	8000000.00	India	9712345678	For Sale	2024-11-11 20:51:32	Available	4	NULL
10	PG Room in Safe Area	A PG room available for rent in a safe area.	101 Sector 10, Bangalore	Bangalore	Karnataka	150000.00	India	9898989898	For Rent	2024-11-11 20:51:32	Available	5	NULL
11	Modern 2BHK Apartment	Modern apartment with all necessary amenities.	15 Sunrise Avenue, Mumbai	Mumbai	Maharashtra	500000.00	India	9369876543	For Sale	2024-11-11 20:51:32	Available	1	NULL
12	Independent House with Garden	Spacious independent house with a beautiful garden.	23 Hiltop Lane, Pune	Pune	Maharashtra	5500000.00	India	9843276543	For Sale	2024-11-11 20:51:32	Available	2	NULL
13	Commercial Space for Rent	A commercial space located in a prime shopping area.	80 Mall Road, Delhi	Delhi	Delhi	1200000.00	India	9543878921	For Rent	2024-11-11 20:51:32	Available	3	NULL
14	Residential Plot for Sale	Prime location residential plot available for sale.	990 City Center, Bangalore	Bangalore	Karnataka	6500000.00	India	9647721898	For Sale	2024-11-11 20:51:32	Available	4	NULL
15	PG Room in Safe Area	Well-maintained PG room with air conditioning and other amenities.	100 Sector 10, Gurgaon	Gurgaon	Haryana	120000.00	India	9898989898	For Rent	2024-11-11 20:51:32	Available	5	NULL
16	Luxury 1BHK Apartment	Stylish and compact 1BHK apartment in a premium location.	48 Skyline Building, Delhi	Delhi	Delhi	350000.00	India	9912345678	For Sale	2024-11-11 20:51:32	Available	1	NULL
17	Detached Independent Home	A beautiful independent home with a spacious backyard.	77 Sunset Blvd, Ahmedabad	Ahmedabad	Gujarat	1500000.00	India	9976543210	For Sale	2024-11-11 20:51:32	Available	2	NULL
18	Co-Working Office Space	A modern co-working space with high-speed internet and conference rooms.	12 Tech Valley, Hyderabad	Hyderabad	Telangana	8500000.00	India	9176543210	For Rent	2024-11-11 20:51:32	Available	3	NULL
19	Prime Residential Plot	A plot with excellent development potential in a growing neighborhood.	55 North Street, Kolkata	Kolkata	West Bengal	4000000.00	India	9898989898	For Sale	2024-11-11 20:51:32	Available	4	NULL
20	PG Room in Safe Area	Safe and secure PG room with easy access to public transport.	68 Safe Lane, Pune	Pune	Maharashtra	10000.00	India	9823456789	For Rent	2024-11-11 20:51:32	Available	5	NULL

20 rows in set (0.01 sec)

In the Property table, there is a column called **Profeatureid** that contains **NULL** values, and we need to update these values.

Query:

mysql> update Property set Profeatureid=1 where Proid=1;

Query OK, 1 row affected (0.02 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=2 where Proid=2;

Query OK, 1 row affected (0.12 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=3 where Proid=3;

Query OK, 1 row affected (0.00 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=4 where Proid=4;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=5 where Proid=5;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=6 where Proid=6;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=7 where Proid=7;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

NIKHIL VANAJI WALUNJ

```
mysql> update Property set Profeatureid=8 where Proid=8;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=9 where Proid=9;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=10 where Proid=10;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=11 where Proid=11;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=12 where Proid=12;
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=13 where Proid=13;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=14 where Proid=14;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=15 where Proid=15;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=16 where Proid=16;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=17 where Proid=17;
Query OK, 1 row affected (0.01 sec)
```

```
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=18 where Proid=18;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=19 where Proid=19;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0

mysql> update Property set Profeatureid=20 where Proid=20;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

3. DELETE statement:

The **DELETE** statement in SQL is used to remove one or more records (rows) from a table based on specified conditions.

Syntax:

DELETE FROM table_name WHERE condition;

Query:

Delete from Customer where Cid=20;

DQL (DATA QUERY LANGUAGE)

Data Query Language (DQL) is a subset of SQL (Structured Query Language) primarily focused on retrieving data from a database.

DQL commands are responsible for operation like **SELECT** on the data.

1. **SELECT statement:**

The **SELECT** statement is a fundamental query in SQL (Structured Query Language) used to retrieve data from a database.

It allows you to specify which columns to retrieve and from which table or tables.

Syntax:

SELECT column1, column2, ...

FROM table_name

WHERE condition

ORDER BY column_name [ASC | DESC];

Component:

SELECT clause:

Specifies the columns you want to retrieve from the database table.

FROM clause:

Specifies the table or tables from which to retrieve data.

WHERE clause:

Optional clause that filters rows based on specified conditions.

ORDER BY clause:

Optional clause that sorts the result set based on specified column(s) in ascending or descending order.

1.1 Retrieve data of Customer table:

Query:

```
mysql> select * from Customer;
```

NIKHIL VANAJI WALUNJ

Result:

```
mysql> select*from Customer;
+---+---+---+---+---+---+---+---+---+---+---+---+
| Cid | Cname | Cage | Caddress | Cgender | Cemail | Cpassword | Cphonenumber | Ccity | Cpincode | Cstate | Ctype |
+---+---+---+---+---+---+---+---+---+---+---+---+
1 | Rohit Sharma | 30 | 21 Marine Drive | Male | rohit.sharma@gmail.com | rohit@123 | 9876543211 | Mumbai | 400020 | Maharashtra | Buyer
2 | Sakshi Mehta | 28 | 12 Vasant Kunj | Female | sakshi.mehta@gmail.com | sakshi@123 | 9765432102 | Delhi | 110070 | Delhi | Saler
3 | Vikram Desai | 35 | 43 JP Nagar | Male | vikram.desai@gmail.com | vikram@123 | 9876043124 | Bangalore | 560078 | Karnataka | Buyer
4 | Priya Reddy | 26 | 17 Hitec City | Female | priya.reddy@gmail.com | priya@123 | 9876087651 | Hyderabad | 500081 | Telangana | Saler
5 | Arjun Singh | 32 | 98 South Avenue | Male | arjun.singh@gmail.com | arjun@123 | 9876532198 | Delhi | 110011 | Delhi | Buyer
6 | Neha Sharma | 27 | 55 Powai | Female | neha.sharma@gmail.com | neha@123 | 9876012347 | Mumbai | 400076 | Maharashtra | Saler
7 | Kiran Patel | 34 | 14 Maninagar | Male | kiran.patel@gmail.com | kiran@123 | 9876545322 | Ahmedabad | 380008 | Gujarat | Buyer
8 | Pooja Bhatt | 30 | 27 Anna Nagar | Female | pooja.bhatt@gmail.com | pooja@123 | 9876087656 | Chennai | 600040 | Tamil Nadu | Saler
9 | Rahul Joshi | 31 | 75 MG Road | Male | rahul.joshi@gmail.com | rahul@123 | 9876078452 | Pune | 411001 | Maharashtra | Buyer
10 | Swati Desai | 29 | 66 Alkapuri | Female | swati.desai@gmail.com | swati@123 | 9876543287 | Vadodara | 390007 | Gujarat | Saler
11 | Vivek Sharma | 38 | 15 Connaught Place | Male | vivek.sharma@gmail.com | vivek@123 | 9876076534 | Delhi | 110001 | Delhi | Buyer
12 | Ritika Nair | 33 | 44 Whitefield | Female | ritika.nair@gmail.com | ritika@123 | 9876021236 | Bangalore | 560066 | Karnataka | Buyer
13 | Rajesh Kumar | 36 | 5 Janakpuri | Male | rajesh.kumar@gmail.com | rajesh@123 | 9876043217 | Delhi | 110058 | Delhi | Saler
14 | Sunita Tiwari | 27 | 18 Sector 56 | Female | sunita.tiwari@gmail.com | sunita@123 | 9876098763 | Noida | 201301 | Uttar Pradesh | Buyer
15 | Amit Agarwal | 40 | 29 Salt Lake | Male | amit.agarwal@gmail.com | amit@123 | 9876067891 | Kolkata | 700091 | West Bengal | Buyer
16 | Anjali Verma | 31 | 37 Sector 10 | Female | anjali.verma@gmail.com | anjali@123 | 9876087653 | Noida | 201301 | Uttar Pradesh | Saler
17 | Manish Chawla | 29 | 22 DLF Phase 2 | Male | manish.chawla@gmail.com | manish@123 | 9876054323 | Gurgaon | 122002 | Haryana | Buyer
18 | Geeta Iyer | 26 | 11 Koramangala | Female | geeta.iyer@gmail.com | geeta@123 | 98760621239 | Bangalore | 560095 | Karnataka | None
19 | Nikhil Sinha | 34 | 19 Connaught Place | Male | nikhil.sinha@gmail.com | nikhil@123 | 9876078941 | Delhi | 110001 | Delhi | Buyer
20 | Divya Shah | 30 | 3 Ellisbridge | Female | divya.shah@gmail.com | divya@123 | 9876076523 | Ahmedabad | 380006 | Gujarat | Saler
21 | Ramesh Mehta | 32 | 8 Sector 15 | Male | ramesh.mehta@gmail.com | ramesh@123 | 9876012348 | Chandigarh | 160015 | Punjab | Buyer
22 | Meera Khan | 28 | 23 Kalina | Female | meera.khan@gmail.com | meera@123 | 9876065421 | Mumbai | 400098 | Maharashtra | Saler
23 | Suresh Yadav | 30 | 67 Baner | Male | suresh.yadav@gmail.com | suresh@123 | 9876098769 | Pune | 411045 | Maharashtra | Buyer
24 | Shalini Ghosh | 29 | 77 Salt Lake | Female | shalini.ghosh@gmail.com | shalini@123 | 9876543215 | Kolkata | 700064 | West Bengal | Saler
25 | Mohit Singh | 33 | 10 Golf Course Road | Male | mohit.singh@gmail.com | mohit@123 | 9876087657 | Gurgaon | 122003 | Haryana | Buyer
26 | Anupam Bhatia | 37 | 91 Ballygunge | Male | anupam.bhatia@gmail.com | anupam@123 | 9876098768 | Kolkata | 700019 | West Bengal | None
27 | Ravi Rao | 29 | 5 Jayanagar | Male | ravi.rao@gmail.com | ravi@123 | 9876076524 | Bangalore | 560041 | Karnataka | Buyer
+---+---+---+---+---+---+---+---+---+---+---+---+
27 rows in set (0.16 sec)
```

1.2 Retrieve data of PropertyCategory table:

Query:

```
mysql> select * from PropertyCategory;
```

Result:

```
mysql> select*from PropertyCategory;
+---+---+
| Category_id | Category_name |
+---+---+
1 | Residential
2 | IndependentHome
3 | Commercial
4 | Plots
5 | PG Room
+---+---+
5 rows in set (0.07 sec)
```

1.3 Retrieve data of PropertySeller table:

Query:

```
mysql> select * from PropertySeller;
```

Result:

```
mysql> select*from Propertyseller;
+---+---+---+---+---+---+---+---+
| Prosellerid | Prosellename | Prosellergender | Prosellerlocation | Proselleremail | Prosellerpassword | Prosellerphoneno |
+---+---+---+---+---+---+---+---+
1 | Sakshi Mehta | Female | Vasant Kunj | sakshi.mehta@gmail.com | sakshi@123 | 9765432102
2 | Priya Reddy | Female | Hitec City | priya.reddy@gmail.com | priya@123 | 9876087651
3 | Neha Sharma | Female | Powai | neha.sharma@gmail.com | neha@123 | 9876012347
4 | Pooja Bhatt | Female | Anna Nagar | pooja.bhatt@gmail.com | pooja@123 | 9876087656
5 | Swati Desai | Female | Alkapuri | swati.desai@gmail.com | swati@123 | 9876543287
6 | Rajesh Kumar | Male | Janakpuri | rajesh.kumar@gmail.com | rajesh@123 | 9876043217
7 | Anjali Verma | Female | Sector 10 | anjali.verma@gmail.com | anjali@123 | 9876087653
8 | Divya Shah | Female | Ellisbridge | divya.shah@gmail.com | divya@123 | 9876076523
9 | Meera Khan | Female | Kalina | meera.khan@gmail.com | meera@123 | 9876065421
10 | Shalini Ghosh | Female | Salt Lake | shalini.ghosh@gmail.com | shalini@123 | 9876543215
+---+---+---+---+---+---+---+---+
10 rows in set (0.01 sec)
```

1.4 Retrieve data of Property table:

Query:

```
mysql> select * from Property;
```

Result:

mysql> select * from Property;					
Proid	Proname	Prodesc	Proaddress	Procity	Prostate
1	Luxury 2BHK Apartment	A spacious 2BHK apartment in a prime location.	123 Park Street, Mumbai	Mumbai	Maharashtra
2	Green View Independent House	A beautiful independent house with a garden and ample space.	45 Green Lane, Pune	Pune	Maharashtra
3	Commercial Office Space	A commercial office space located in the business hub.	10 Business Park, Delhi	Delhi	Delhi
4	Residential Plot	A well-located residential plot in a developing area.	60 Greenfield, Bangalore	Bangalore	Karnataka
5	PG Room for Rent	A fully furnished PG room with all amenities included.	101 Comfort Street, Chennai	Chennai	Tamil Nadu
6	Oceanview 3BHK	A luxury 3BHK apartment with ocean views and modern amenities.	234 Sea Breeze Road, Goa	Goa	Goa
7	Villa in Green Valley	A spacious villa with a private pool and large garden.	12 Green Valley, Ahmedabad	Ahmedabad	Gujarat
8	Tech Hub Office	An office space designed for tech companies with modern infrastructure.	56 Silicon Street, Hyderabad	Hyderabad	Telangana
9	Luxury Plot in Prime Area	A plot for residential development in a prime location.	500 Main Road, Kolkata	Kolkata	West Bengal
10	PG Room Near IT Park	A comfortable PG room with food and laundry services.	78 IT Park Lane, Bangalore	Bangalore	Karnataka
11	Modern 2BHK Apartment	Modern apartment with all necessary amenities.	15 Sunrise Avenue, Mumbai	Mumbai	Maharashtra
12	Independent House with Garden	Spacious independent house with a beautiful garden.	23 Hilltop Lane, Pune	Pune	Maharashtra
13	Commercial Space for Retail	A commercial space located in a prime shopping area.	88 Mall Road, Delhi	Delhi	Delhi
14	Residential Plot for Sale	Prime location residential plot available for sale.	900 City Center, Bangalore	Bangalore	Karnataka
15	PG Room with AC	Well-maintained PG room with air conditioning and other amenities.	30 Liberty Road, Chennai	Chennai	Tamil Nadu
16	Luxury 1BHK Apartment	Stylish and compact 1BHK apartment in a premium location.	40 Skyline Building, Delhi	Delhi	Delhi
17	Detached Independent Home	A beautiful independent home with a spacious backyard.	77 Sunset Blvd, Ahmedabad	Ahmedabad	Gujarat
18	Co-Working Office Space	A modern co-working space with high-speed internet and conference rooms.	12 Tech Valley, Hyderabad	Hyderabad	Telangana
19	Prime Residential Plot	A plot with excellent development potential in a growing neighborhood.	55 North Street, Kolkata	Kolkata	West Bengal
20	PG Room in Safe Area	Safe and secure PG room with easy access to public transport.	60 Safe Lane, Pune	Pune	Maharashtra

20 rows in set (0.00 sec)

Proprice	Procountry	Procontactno	Prolistingtype	Propostdate	Prostatus	Category_id	Profeatureid
7500000.00	India	9876543210	For Sale	2024-11-11 20:51:32	Available	1	1
12000000.00	India	9898989898	For Sale	2024-11-11 20:51:32	Available	2	2
25000000.00	India	9999999999	For Rent	2024-11-11 20:51:32	Available	3	3
5000000.00	India	9876543210	For Sale	2024-11-11 20:51:32	Available	4	4
15000.00	India	9500000000	For Rent	2024-11-11 20:51:32	Available	5	5
9500000.00	India	8888888888	For Sale	2024-11-11 20:51:32	Available	1	6
18000000.00	India	9612345678	For Sale	2024-11-11 20:51:32	Available	2	7
15000000.00	India	9176543210	For Rent	2024-11-11 20:51:32	Available	3	8
8000000.00	India	9712345678	For Sale	2024-11-11 20:51:32	Available	4	9
12000.00	India	9512345678	For Rent	2024-11-11 20:51:32	Available	5	10
5000000.00	India	9369876543	For Sale	2024-11-11 20:51:32	Available	1	11
8500000.00	India	9843276543	For Sale	2024-11-11 20:51:32	Available	2	12
12000000.00	India	9543678921	For Rent	2024-11-11 20:51:32	Available	3	13
6500000.00	India	9643721890	For Sale	2024-11-11 20:51:32	Available	4	14
18000.00	India	9445678910	For Rent	2024-11-11 20:51:32	Available	5	15
3500000.00	India	9912345678	For Sale	2024-11-11 20:51:32	Available	1	16
15000000.00	India	9976543210	For Sale	2024-11-11 20:51:32	Available	2	17
850000.00	India	9176543210	For Rent	2024-11-11 20:51:32	Available	3	18
4000000.00	India	9800000000	For Sale	2024-11-11 20:51:32	Available	4	19
10000.00	India	9823456789	For Rent	2024-11-11 20:51:32	Available	5	20

1.5 Retrieve data of Payment table:

Query:

```
mysql> select * from Payment;
```

Result:

mysql> select * from Payment;							
Paymentid	BuyerCid	Proid	ProSellerid	Paymentamount	Paymentmethod	Paymentstatus	Paymentdate
1	1	3	1	500000.00	Credit Card	Done	2024-11-11 23:59:47
2	7	7	3	750000.00	Bank Transfer	Pending	2024-11-11 23:59:47
3	12	2	5	1200000.00	Cash	Done	2024-11-11 23:59:47
4	15	12	7	600000.00	Debit Card	Pending	2024-11-11 23:59:47
5	19	5	9	450000.00	UPI	Done	2024-11-11 23:59:47
6	5	20	6	800000.00	Credit Card	Done	2024-11-11 23:59:47
7	25	15	2	1000000.00	Bank Transfer	Pending	2024-11-11 23:59:47

7 rows in set (0.00 sec)

1.6 Retrieve data of PropertyFeatures table:

Query:

```
mysql> select * from PropertyFeatures;
```

Result:

Profeatureid	Property_Areasize	Floorno	Nearestlocation	Numofbedroom	Numofbathroom	Balcony	Parkingspace	Gym	Swimmingpool	Furnished
1	1200 sq ft	5th	Park Street, Mumbai	2		2	Available	Available	Available	Unavailable
2	2000 sq ft	2nd	Green Lane, Pune	3		3	Available	Unavailable	Unavailable	No
3	3000 sq ft	8th	Business Park, Delhi	0		2	Unavailable	Available	Unavailable	No
4	1500 sq ft	Ground	Greenfield, Bangalore	0		0	Unavailable	Available	Unavailable	No
5	250 sq ft	1st	Comfort Street, Chennai	1		1	Available	Unavailable	Unavailable	Yes
6	1800 sq ft	10th	Sea Breeze Road, Goa	3		2	Available	Available	Available	Semi
7	3500 sq ft	4th	Green Valley, Ahmedabad	4		4	Available	Available	Unavailable	Yes
8	4000 sq ft	6th	Silicon Street, Hyderabad	0		3	Unavailable	Available	Available	No
9	5000 sq ft	7th	Main Road, Kolkata	0		0	Unavailable	Unavailable	Unavailable	No
10	350 sq ft	3rd	IT Park Lane, Bangalore	1		1	Unavailable	Available	Unavailable	Yes
11	1100 sq ft	12th	Sunrise Avenue, Mumbai	2		2	Available	Available	Available	Semi
12	2200 sq ft	5th	Hilltop Lane, Pune	3		2	Unavailable	Available	Unavailable	No
13	3500 sq ft	3rd	Mall Road, Delhi	0		1	Available	Available	Unavailable	No
14	1800 sq ft	Ground	City Center, Bangalore	0		0	Unavailable	Unavailable	Unavailable	No
15	200 sq ft	4th	Liberty Road, Chennai	1		1	Available	Unavailable	Unavailable	Yes
16	900 sq ft	15th	Skyline Building, Delhi	1		1	Unavailable	Available	Unavailable	Semi
17	3500 sq ft	2nd	Sunset Blvd, Ahmedabad	4		3	Available	Available	Available	Yes
18	1500 sq ft	9th	Tech Valley, Hyderabad	0		2	Unavailable	Available	Available	No
19	5000 sq ft	Ground	North Street, Kolkata	0		0	Unavailable	Unavailable	Unavailable	No
20	300 sq ft	1st	Safe Lane, Pune	1		1	Available	Unavailable	Unavailable	Yes

20 rows in set (0.00 sec)

1.7 Retrieve data of PropertyInquiry table:

Query:

```
mysql> select * from PropertyInquiry;
```

Result:

Inquiryid	Proid	SenderCid	Senderemail	Sendername	Senderphoneno
1	2	9	rahul.joshi@gmail.com	Rahul Joshi	9876078452
2	6	1	rohit.sharma@gmail.com	Rohit Sharma	9876543211
3	10	19	nikhil.sinha@gmail.com	Nikhil Sinha	9876078941
4	15	5	arjun.singh@gmail.com	Arjun Singh	9876532198
5	19	14	sunita.tiwari@gmail.com	Sunita Tiwari	9876098763

5 rows in set (0.00 sec)

Inquirymessage	Inquirydate	Inquirystatus	ProSellerid
Can you schedule a viewing for this property?	2024-11-11 23:49:06	Pending	1
I am interested in this property. Please provide more details.	2024-11-11 23:49:06	In Progress	3
What are the maintenance costs for this property?	2024-11-11 23:49:06	Completed	8
What is the expected closing date for this property?	2024-11-11 23:49:06	Completed	6
Is this property pet-friendly?	2024-11-11 23:49:06	Pending	10

WHERE Clause Search Condition

The WHERE clause in SQL is used to filter records based on specified conditions.

It allows you to retrieve rows that meet certain criteria from a table. It is used in SELECT, UPDATE, DELETE statement etc.

Syntax:

SELECT column1, column2, ...

FROM table_name

WHERE condition;

Query:

mysql> SELECT Cname, Cage

-> FROM Customer

-> WHERE Ccity = 'Mumbai';

Result:

```
mysql> SELECT Cname, Cage
-> FROM Customer
-> WHERE Ccity = 'Mumbai';
+-----+-----+
| Cname | Cage |
+-----+-----+
| Rohit Sharma | 30 |
| Neha Sharma | 27 |
| Meera Khan | 28 |
+-----+-----+
3 rows in set (0.00 sec)
```

WHERE clause with Arithmetic operators:

Addition, Subtraction, Multiply, Division, Modulo.

1. ADDITION:

Query:

mysql> update Property set Proprice=Proprice+20000 where Proid=20;

Query OK, 20 rows affected (0.36 sec)

```
mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property;
+-----+-----+-----+-----+-----+
| Proid | Proname | Prodesc | Procity | Proprice |
+-----+-----+-----+-----+-----+
| 1 | Luxury 2BHK Apartment | A spacious 2BHK apartment in a prime location. | Mumbai | 7520000.00 |
| 2 | Green View Independent House | A beautiful independent house with a garden and ample space. | Pune | 12020000.00 |
| 3 | Commercial Office Space | A commercial office space located in the business hub. | Delhi | 25020000.00 |
| 4 | Residential Plot | A well-located residential plot in a developing area. | Bangalore | 5020000.00 |
| 5 | PG Room for Rent | A fully furnished PG room with all amenities included. | Chennai | 35000.00 |
| 6 | Oceanview 3BHK | A luxury 3BHK apartment with ocean views and modern amenities. | Goa | 9520000.00 |
| 7 | Villa in Green Valley | A spacious villa with a private pool and large garden. | Ahmedabad | 18020000.00 |
| 8 | Tech Hub Office | An office space designed for tech companies with modern infrastructure. | Hyderabad | 15020000.00 |
| 9 | Luxury Plot in Prime Area | A plot for residential development in a prime location. | Kolkata | 8020000.00 |
| 10 | PG Room Near IT Park | A comfortable PG room with food and laundry services. | Bangalore | 32000.00 |
| 11 | Modern 2BHK Apartment | Modern apartment with all necessary amenities. | Mumbai | 5020000.00 |
| 12 | Independent House with Garden | Spacious independent house with a beautiful garden. | Pune | 8520000.00 |
| 13 | Commercial Space for Retail | A commercial space located in a prime shopping area. | Delhi | 12020000.00 |
| 14 | Residential Plot for Sale | Prime location residential plot available for sale. | Bangalore | 6520000.00 |
| 15 | PG Room with AC | Well-maintained PG room with air conditioning and other amenities. | Chennai | 38000.00 |
| 16 | Luxury 1BHK Apartment | Stylish and compact 1BHK apartment in a premium location. | Delhi | 3520000.00 |
| 17 | Detached Independent Home | A beautiful independent home with a spacious backyard. | Ahmedabad | 15020000.00 |
| 18 | Co-Working Office Space | A modern co-working space with high-speed internet and conference rooms. | Hyderabad | 8520000.00 |
| 19 | Prime Residential Plot | A plot with excellent development potential in a growing neighborhood. | Kolkata | 4020000.00 |
| 20 | PG Room in Safe Area | Safe and secure PG room with easy access to public transport. | Pune | 30000.00 |
+-----+-----+-----+-----+-----+
20 rows in set (0.00 sec)
```

NIKHIL VANAJI WALUNJ

2. SUBTRACTION:

Query:

```
mysql> update Property set Proprice=Proprice-20000 where Proid=20;
```

Query OK, 20 rows affected (0.01 sec)

mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property;				
Proid	Proname	Prodesc	Procity	Proprice
1	Luxury 2BHK Apartment	A spacious 2BHK apartment in a prime location.	Mumbai	7500000.00
2	Green View Independent House	A beautiful independent house with a garden and ample space.	Pune	12000000.00
3	Commercial Office Space	A commercial office space located in the business hub.	Delhi	25000000.00
4	Residential Plot	A well-located residential plot in a developing area.	Bangalore	5000000.00
5	PG Room for Rent	A fully furnished PG room with all amenities included.	Chennai	15000.00
6	Oceanview 3BHK	A luxury 3BHK apartment with ocean views and modern amenities.	Goa	9500000.00
7	Villa in Green Valley	A spacious villa with a private pool and large garden.	Ahmedabad	18000000.00
8	Tech Hub Office	An office space designed for tech companies with modern infrastructure.	Hyderabad	15000000.00
9	Luxury Plot in Prime Area	A plot for residential development in a prime location.	Kolkata	8000000.00
10	PG Room Near IT Park	A comfortable PG room with food and laundry services.	Bangalore	12000.00
11	Modern 2BHK Apartment	Modern apartment with all necessary amenities.	Mumbai	5000000.00
12	Independent House with Garden	Spacious independent house with a beautiful garden.	Pune	8500000.00
13	Commercial Space for Retail	A commercial space located in a prime shopping area.	Delhi	12000000.00
14	Residential Plot for Sale	Prime location residential plot available for sale.	Bangalore	6500000.00
15	PG Room with AC	Well-maintained PG room with air conditioning and other amenities.	Chennai	18000.00
16	Luxury 1BHK Apartment	Stylish and compact 1BHK apartment in a premium location.	Delhi	3500000.00
17	Detached Independent Home	A beautiful independent home with a spacious backyard.	Ahmedabad	15000000.00
18	Co-Working Office Space	A modern co-working space with high-speed internet and conference rooms.	Hyderabad	8500000.00
19	Prime Residential Plot	A plot with excellent development potential in a growing neighborhood.	Kolkata	4000000.00
20	PG Room in Safe Area	Safe and secure PG room with easy access to public transport.	Pune	10000.00

3. DIVISION:

Query:

```
mysql> update Property set Proprice=Proprice/2 where Proid=20;
```

Query OK, 20 rows affected (0.00 sec)

mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property;				
Proid	Proname	Prodesc	Procity	Proprice
1	Luxury 2BHK Apartment	A spacious 2BHK apartment in a prime location.	Mumbai	3750000.00
2	Green View Independent House	A beautiful independent house with a garden and ample space.	Pune	6000000.00
3	Commercial Office Space	A commercial office space located in the business hub.	Delhi	12500000.00
4	Residential Plot	A well-located residential plot in a developing area.	Bangalore	2500000.00
5	PG Room for Rent	A fully furnished PG room with all amenities included.	Chennai	7500.00
6	Oceanview 3BHK	A luxury 3BHK apartment with ocean views and modern amenities.	Goa	4750000.00
7	Villa in Green Valley	A spacious villa with a private pool and large garden.	Ahmedabad	9000000.00
8	Tech Hub Office	An office space designed for tech companies with modern infrastructure.	Hyderabad	7500000.00
9	Luxury Plot in Prime Area	A plot for residential development in a prime location.	Kolkata	4000000.00
10	PG Room Near IT Park	A comfortable PG room with food and laundry services.	Bangalore	6000.00
11	Modern 2BHK Apartment	Modern apartment with all necessary amenities.	Mumbai	2500000.00
12	Independent House with Garden	Spacious independent house with a beautiful garden.	Pune	4250000.00
13	Commercial Space for Retail	A commercial space located in a prime shopping area.	Delhi	6000000.00
14	Residential Plot for Sale	Prime location residential plot available for sale.	Bangalore	3250000.00
15	PG Room with AC	Well-maintained PG room with air conditioning and other amenities.	Chennai	9000.00
16	Luxury 1BHK Apartment	Stylish and compact 1BHK apartment in a premium location.	Delhi	1750000.00
17	Detached Independent Home	A beautiful independent home with a spacious backyard.	Ahmedabad	7500000.00
18	Co-Working Office Space	A modern co-working space with high-speed internet and conference rooms.	Hyderabad	4250000.00
19	Prime Residential Plot	A plot with excellent development potential in a growing neighborhood.	Kolkata	2000000.00
20	PG Room in Safe Area	Safe and secure PG room with easy access to public transport.	Pune	5000.00

4. MULTIPLICATION:

Query:

```
mysql> update Property set Proprice=Proprice*2 where Proid=20;
```

Query OK, 20 rows affected (0.00 sec)

Proid	Proname	Prodsc	Procity	Proprice
1	Luxury 2BHK Apartment	A spacious 2BHK apartment in a prime location.	Mumbai	7500000.00
2	Green View Independent House	A beautiful independent house with a garden and ample space.	Pune	12000000.00
3	Commercial Office Space	A commercial office space located in the business hub.	Delhi	25000000.00
4	Residential Plot	A well-located residential plot in a developing area.	Bangalore	5000000.00
5	PG Room for Rent	A fully furnished PG room with all amenities included.	Chennai	15000.00
6	Oceanview 3BHK	A luxury 3BHK apartment with ocean views and modern amenities.	Goa	9500000.00
7	Villa in Green Valley	A spacious villa with a private pool and large garden.	Ahmedabad	18000000.00
8	Tech Hub Office	An office space designed for tech companies with modern infrastructure.	Hyderabad	15000000.00
9	Luxury Plot in Prime Area	A plot for residential development in a prime location.	Kolkata	8000000.00
10	PG Room Near IT Park	A comfortable PG room with food and laundry services.	Bangalore	12000.00
11	Modern 2BHK Apartment	Modern apartment with all necessary amenities.	Mumbai	5000000.00
12	Independent House with Garden	Spacious independent house with a beautiful garden.	Pune	8500000.00
13	Commercial Space for Retail	A commercial space located in a prime shopping area.	Delhi	12000000.00
14	Residential Plot for Sale	Prime location residential plot available for sale.	Bangalore	6500000.00
15	PG Room with AC	Well-maintained PG room with air conditioning and other amenities.	Chennai	18000.00
16	Luxury 1BHK Apartment	Stylish and compact 1BHK apartment in a premium location.	Delhi	3500000.00
17	Detached Independent Home	A beautiful independent home with a spacious backyard.	Ahmedabad	15000000.00
18	Co-Working Office Space	A modern co-working space with high-speed internet and conference rooms.	Hyderabad	8500000.00
19	Prime Residential Plot	A plot with excellent development potential in a growing neighborhood.	Kolkata	4000000.00
20	PG Room in Safe Area	Safe and secure PG room with easy access to public transport.	Pune	10000.00

20 rows in set (0.00 sec)

5. MODULO:

Query:

```
mysql> select Proid,Proname,Prodsc,Procity,Proprice from Property where Proprice%7=2;
```

Proid	Proname	Prodsc	Procity	Proprice
10	PG Room Near IT Park	A comfortable PG room with food and laundry services.	Bangalore	12000.00

1 row in set (0.00 sec)

WHERE clause with Comparison operators:

Less than, greater than, equal to, less than or equal to, greater than or equal to, not equal to.

1. LESS THAN (<):

Query:

```
mysql> select Proid,Proname,Prodsc,Procity,Proprice from Property
where Proprice < 100000;
```

Proid	Proname	Prodsc	Procity	Proprice
5	PG Room for Rent	A fully furnished PG room with all amenities included.	Chennai	15000.00
10	PG Room Near IT Park	A comfortable PG room with food and laundry services.	Bangalore	12000.00
15	PG Room with AC	Well-maintained PG room with air conditioning and other amenities.	Chennai	18000.00
20	PG Room in Safe Area	Safe and secure PG room with easy access to public transport.	Pune	10000.00

4 rows in set (0.00 sec)

2. GREATER THAN (>):

Query:

```
mysql> select Proid,Proname,Prodsc,Procity,Proprice from Property
where Proprice > 1000000;
```

```
mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property where Proprice > 10000000;
+-----+-----+-----+-----+-----+
| Proid | Proname      | Prodesc          | Procity | Proprice |
+-----+-----+-----+-----+
| 2    | Green View Independent House | A beautiful independent house with a garden and ample space. | Pune   | 12000000.00 |
| 3    | Commercial Office Space     | A commercial office space located in the business hub.        | Delhi  | 25000000.00 |
| 7    | Villa in Green Valley       | A spacious villa with a private pool and large garden.       | Ahmedabad | 18000000.00 |
| 8    | Tech Hub Office            | An office space designed for tech companies with modern infrastructure. | Hyderabad | 15000000.00 |
| 13   | Commercial Space for Retail | A commercial space located in a prime shopping area.           | Delhi  | 12000000.00 |
| 17   | Detached Independent Home | A beautiful independent home with a spacious backyard.        | Ahmedabad | 15000000.00 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

3. EQUAL TO (=):

Query:

mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property
where Proprice = 15000000;

```
mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property where Proprice = 15000000;
+-----+-----+-----+-----+-----+
| Proid | Proname      | Prodesc          | Procity | Proprice |
+-----+-----+-----+-----+
| 8    | Tech Hub Office | An office space designed for tech companies with modern infrastructure. | Hyderabad | 15000000.00 |
| 17   | Detached Independent Home | A beautiful independent home with a spacious backyard. | Ahmedabad | 15000000.00 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

4. LESS THAN OR EQUAL TO (<=):

Query:

mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property
where Proprice <= 18000;

```
mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property where Proprice <= 18000;
+-----+-----+-----+-----+-----+
| Proid | Proname      | Prodesc          | Procity | Proprice |
+-----+-----+-----+-----+
| 5    | PG Room for Rent | A fully furnished PG room with all amenities included. | Chennai | 15000.00 |
| 10   | PG Room Near IT Park | A comfortable PG room with food and laundry services. | Bangalore | 12000.00 |
| 15   | PG Room with AC | Well-maintained PG room with air conditioning and other amenities. | Chennai | 18000.00 |
| 20   | PG Room in Safe Area | Safe and secure PG room with easy access to public transport. | Pune   | 10000.00 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

5. GREATER THAN OR EQUAL TO (>=):

Query:

mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property
where Proprice >= 11000000;

```
mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property where Proprice >= 11000000;
+-----+-----+-----+-----+-----+
| Proid | Proname      | Prodesc          | Procity | Proprice |
+-----+-----+-----+-----+
| 2    | Green View Independent House | A beautiful independent house with a garden and ample space. | Pune   | 12000000.00 |
| 3    | Commercial Office Space     | A commercial office space located in the business hub.        | Delhi  | 25000000.00 |
| 7    | Villa in Green Valley       | A spacious villa with a private pool and large garden.       | Ahmedabad | 18000000.00 |
| 8    | Tech Hub Office            | An office space designed for tech companies with modern infrastructure. | Hyderabad | 15000000.00 |
| 13   | Commercial Space for Retail | A commercial space located in a prime shopping area.           | Delhi  | 12000000.00 |
| 17   | Detached Independent Home | A beautiful independent home with a spacious backyard.        | Ahmedabad | 15000000.00 |
+-----+-----+-----+-----+
6 rows in set (0.00 sec)
```

6. NOT EQUAL TO (\neq):

Query:

```
mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property
where Proprice <> 15000000;
```

mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property where Proprice <> 15000000;				
Proid	Proname	Prodesc	Procity	Proprice
1	Luxury 2BHK Apartment	A spacious 2BHK apartment in a prime location.	Mumbai	7500000.00
2	Green View Independent House	A beautiful independent house with a garden and ample space.	Pune	12000000.00
3	Commercial Office Space	A commercial office space located in the business hub.	Delhi	25000000.00
4	Residential Plot	A well-located residential plot in a developing area.	Bangalore	5000000.00
5	PG Room for Rent	A fully furnished PG room with all amenities included.	Chennai	15000.00
6	Oceanview 3BHK	A luxury 3BHK apartment with ocean views and modern amenities.	Goa	9500000.00
7	Villa in Green Valley	A spacious villa with a private pool and large garden.	Ahmedabad	18000000.00
9	Luxury Plot in Prime Area	A plot for residential development in a prime location.	Kolkata	8000000.00
10	PG Room Near IT Park	A comfortable PG room with food and laundry services.	Bangalore	12000.00
11	Modern 2BHK Apartment	Modern apartment with all necessary amenities.	Mumbai	5000000.00
12	Independent House with Garden	Spacious independent house with a beautiful garden.	Pune	8500000.00
13	Commercial Space for Retail	A commercial space located in a prime shopping area.	Delhi	12000000.00
14	Residential Plot for Sale	Prime location residential plot available for sale.	Bangalore	6500000.00
15	PG Room with AC	Well-maintained PG room with air conditioning and other amenities.	Chennai	18000.00
16	Luxury 1BHK Apartment	Stylish and compact 1BHK apartment in a premium location.	Delhi	3500000.00
18	Co-Working Office Space	A modern co-working space with high-speed internet and conference rooms.	Hyderabad	8500000.00
19	Prime Residential Plot	A plot with excellent development potential in a growing neighborhood.	Kolkata	4000000.00
20	PG Room in Safe Area	Safe and secure PG room with easy access to public transport.	Pune	10000.00

18 rows in set (0.00 sec)

WHERE clause with Logical operators:

AND, OR, NOT.

1. AND (&& or AND):

Returns true if both conditions are true.

Query:

```
mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property
where Proprice < 20000 AND Procity="Pune";
```

mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property where Proprice < 20000 AND Procity="Pune";				
Proid	Proname	Prodesc	Procity	Proprice
20	PG Room in Safe Area	Safe and secure PG room with easy access to public transport.	Pune	10000.00

1 row in set (0.00 sec)

2. OR (|| or OR):

Returns true if at least one condition is true.

Query:

```
mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property
where Proprice < 20000 OR Procity="Pune";
```

```
mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property where Proprice < 20000 OR Procity="Pune";
```

Proid	Proname	Prodesc	Procity	Proprice
2	Green View Independent House	A beautiful independent house with a garden and ample space.	Pune	12000000.00
5	PG Room for Rent	A fully furnished PG room with all amenities included.	Chennai	15000.00
10	PG Room Near IT Park	A comfortable PG room with food and laundry services.	Bangalore	12000.00
12	Independent House with Garden	Spacious independent house with a beautiful garden.	Pune	8500000.00
15	PG Room with AC	Well-maintained PG room with air conditioning and other amenities.	Chennai	18000.00
20	PG Room in Safe Area	Safe and secure PG room with easy access to public transport.	Pune	10000.00

6 rows in set (0.00 sec)

3. NOT (! or NOT):

Reverses the result of the condition.

Query:

```
mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property
```

```
where NOT Proprice > 20000;
```

```
mysql> select Proid,Proname,Prodesc,Procity,Proprice from Property where NOT Proprice > 20000;
```

Proid	Proname	Prodesc	Procity	Proprice
5	PG Room for Rent	A fully furnished PG room with all amenities included.	Chennai	15000.00
10	PG Room Near IT Park	A comfortable PG room with food and laundry services.	Bangalore	12000.00
15	PG Room with AC	Well-maintained PG room with air conditioning and other amenities.	Chennai	18000.00
20	PG Room in Safe Area	Safe and secure PG room with easy access to public transport.	Pune	10000.00

RANGE OPERATOR:

The range operator in SQL is used to specify a range of values within a certain criterion.

1. BETWEEN Operator

The **BETWEEN** operator in SQL is commonly used as a range operator to filter data within a specified lower and upper limit (inclusive of both boundaries).

Syntax:

```
SELECT * FROM table_name
```

```
WHERE column_name BETWEEN lower AND upper;
```

Query:

```
select Proid,Proname,Procity,Proprice from Property
```

```
where Proprice between 8500000 and 15000000;
```

```
mysql> select Proid,Proname,Procity,Proprice from Property where Proprice between 8500000 and 15000000;
```

Proid	Proname	Procity	Proprice
2	Green View Independent House	Pune	12000000.00
6	Oceanview 3BHK	Goa	9500000.00
8	Tech Hub Office	Hyderabad	15000000.00
12	Independent House with Garden	Pune	8500000.00
13	Commercial Space for Retail	Delhi	12000000.00
17	Detached Independent Home	Ahmedabad	15000000.00
18	Co-Working Office Space	Hyderabad	8500000.00

7 rows in set (0.11 sec)

2. NOT BETWEEN Operator

The **NOT BETWEEN** operator is used to filter records where a column's value does **not** fall within a specified range. It is the opposite of the BETWEEN operator.

Syntax:

```
SELECT * FROM table_name
```

```
WHERE column_name NOT BETWEEN lower AND upper;
```

Query:

```
select Proid,Proname,Procity,Proprice from Property  
where Proprice not between 5000000 and 15000000;
```

```
mysql> select Proid,Proname,Procity,Proprice from Property where Proprice not between 5000000 and 15000000;  
+----+-----+-----+-----+  
| Proid | Proname           | Procity   | Proprice |  
+----+-----+-----+-----+  
|    3 | Commercial Office Space | Delhi     | 25000000.00 |  
|    5 | PG Room for Rent      | Chennai   | 15000.00  |  
|    7 | Villa in Green Valley | Ahmedabad | 18000000.00 |  
|   10 | PG Room Near IT Park | Bangalore | 12000.00  |  
|   15 | PG Room with AC       | Chennai   | 18000.00  |  
|   16 | Luxury 1BHK Apartment | Delhi     | 3500000.00 |  
|   19 | Prime Residential Plot | Kolkata   | 4000000.00 |  
|   20 | PG Room in Safe Area  | Pune      | 10000.00  |  
+----+-----+-----+-----+  
8 rows in set (0.00 sec)
```

LIST OPERATOR:

The list operator in SQL allows you to filter data based on a list of specified values

1. IN Operator

Checks if a value matches any value in a specified list.

Syntax:

```
SELECT * FROM table_name
```

```
WHERE column_name IN (value1, value2, ..., valueN);
```

Query:

```
select Proid,Proname,Procity,Proprice from Property  
where Proprice IN (15000.00, 12000.00, 18000.00);
```

```
mysql> select Proid,Proname,Procity,Proprice from Property where Proprice IN (15000.00, 12000.00, 18000.00);  
+----+-----+-----+-----+  
| Proid | Proname           | Procity   | Proprice |  
+----+-----+-----+-----+  
|    5 | PG Room for Rent      | Chennai   | 15000.00 |  
|   10 | PG Room Near IT Park | Bangalore | 12000.00 |  
|   15 | PG Room with AC       | Chennai   | 18000.00 |  
+----+-----+-----+-----+  
3 rows in set (0.00 sec)
```

2. NOT IN Operator

Checks if a value does not match any value in a specified list.

Syntax:

```
SELECT * FROM table_name
```

```
WHERE column_name NOT IN (value1, value2, ..., valueN);
```

Query:

```
select Proid,Proname,Procity,Proprice from Property
```

```
where Proprice NOT IN (7500000.00, 12000000.00, 18000000.00);
```

Proid	Proname	Procity	Proprice
3	Commercial Office Space	Delhi	25000000.00
4	Residential Plot	Bangalore	5000000.00
5	PG Room for Rent	Chennai	15000.00
6	Oceanview 3BHK	Goa	9500000.00
8	Tech Hub Office	Hyderabad	15000000.00
9	Luxury Plot in Prime Area	Kolkata	8000000.00
10	PG Room Near IT Park	Bangalore	12000.00
11	Modern 2BHK Apartment	Mumbai	5000000.00
12	Independent House with Garden	Pune	8500000.00
14	Residential Plot for Sale	Bangalore	6500000.00
15	PG Room with AC	Chennai	18000.00
16	Luxury 1BHK Apartment	Delhi	3500000.00
17	Detached Independent Home	Ahmedabad	15000000.00
18	Co-Working Office Space	Hyderabad	8500000.00
19	Prime Residential Plot	Kolkata	4000000.00
20	PG Room in Safe Area	Pune	10000.00

16 rows in set (0.00 sec)

LIKE OPERATOR:

The LIKE operator in SQL is used to displays only those data from the table which matches the pattern specified in the query.

Syntax:

```
SELECT * FROM table_name WHERE column_name LIKE pattern;
```

1. %: It represents Zero, one or multiple characters

Query:

```
select Prosellerid,Prosellename,Prosellerphoneno from PropertySeller
```

```
where Prosellename like "S%";
```

Prosellerid	Prosellename	Prosellerphoneno
1	Sakshi Mehta	9765432102
5	Swati Desai	9876543287
10	Shalini Ghosh	9876543215

3 rows in set (0.42 sec)

2. Underscore (_): It represents single characters.

Query:

```
select Prosellerid,Prosellername,Prosellerphoneno from PropertySeller  
where Prosellername like "_e%";
```

```
mysql> select Prosellerid,Prosellername,Prosellerphoneno from PropertySeller where Prosellername like "_e%";  
+-----+-----+-----+  
| Prosellerid | Prosellername | Prosellerphoneno |  
+-----+-----+-----+  
| 3 | Neha Sharma | 9876012347 |  
| 9 | Meera Khan | 9876065421 |  
+-----+-----+-----+  
2 rows in set (0.00 sec)
```

3. [charlist]: Matches any single character in the list.

Syntax:

```
SELECT * FROM PropertySeller  
WHERE Prosellername LIKE '[SPD]%';
```

USING ORDER BY, DISTINCT and TOP

1. ORDER BY Clause:

The **ORDER BY** clause is used in SQL to sort the result set of a query in ascending or descending order based on one or more columns.

ASC Order:

ASC (ascending) is the default sorting order.

Syntax:

```
SELECT * FROM PropertySeller  
ORDER BY column_name ASC;
```

```
mysql> select Prosellerid,Prosellername,Prosellerphoneno from PropertySeller order by Prosellerid ASC;  
+-----+-----+-----+  
| Prosellerid | Prosellername | Prosellerphoneno |  
+-----+-----+-----+  
| 1 | Sakshi Mehta | 9765432102 |  
| 2 | Priya Reddy | 9876087651 |  
| 3 | Neha Sharma | 9876012347 |  
| 4 | Pooja Bhatt | 9876087656 |  
| 5 | Swati Desai | 9876543287 |  
| 6 | Rajesh Kumar | 9876043217 |  
| 7 | Anjali Verma | 9876087653 |  
| 8 | Divya Shah | 9876076523 |  
| 9 | Meera Khan | 9876065421 |  
| 10 | Shalini Ghosh | 9876543215 |  
+-----+-----+-----+  
10 rows in set (0.00 sec)
```

Query:

```
select Prosellerid,Prosellename,Prosellerphoneno from PropertySeller  
order by Prosellerid ASC;
```

DESC Order:

DESC (descending) sorts in reverse order.

Syntax:

```
SELECT * FROM PropertySeller  
ORDER BY column_name DESC;
```

Query:

```
select Prosellerid,Prosellename,Prosellerphoneno from PropertySeller  
order by Prosellerid DESC;
```

```
mysql> select Prosellerid,Prosellename,Prosellerphoneno from PropertySeller order by Prosellerid DESC;  
+-----+-----+-----+  
| Prosellerid | Prosellename | Prosellerphoneno |  
+-----+-----+-----+  
| 10 | Shalini Ghosh | 9876543215 |  
| 9 | Meera Khan | 9876065421 |  
| 8 | Divya Shah | 9876076523 |  
| 7 | Anjali Verma | 9876087653 |  
| 6 | Rajesh Kumar | 9876043217 |  
| 5 | Swati Desai | 9876543287 |  
| 4 | Pooja Bhatt | 9876087656 |  
| 3 | Neha Sharma | 9876012347 |  
| 2 | Priya Reddy | 9876087651 |  
| 1 | Sakshi Mehta | 9765432102 |  
+-----+-----+-----+  
10 rows in set (0.11 sec)
```

2. DISTINCT keyword:

The **DISTINCT** keyword is used to retrieve unique values from a specified column or combination of columns in a query result set.

Syntax:

```
SELECT DISTINCT column_name FROM table_name;
```

```
mysql> select distinct Procity from Property;  
+-----+  
| Procity |  
+-----+  
| Mumbai |  
| Pune |  
| Delhi |  
| Bangalore |  
| Chennai |  
| Goa |  
| Ahmedabad |  
| Hyderabad |  
| Kolkata |  
+-----+  
9 rows in set (0.00 sec)
```

Query:

```
select distinct Procity from Property;
```

3. LIMIT clause:

LIMIT keyword (used in MySQL) is used to limit the number of rows returned in the result set.

Syntax:

```
SELECT * FROM table_name order by column_name LIMIT 5;
```

Query:

```
select * from PropertySeller order by Prosellerid limit 5;
```

```
mysql> select * from PropertySeller order by Prosellerid limit 5;
+-----+-----+-----+-----+-----+-----+-----+
| Prosellerid | Prosellename | Prosellergender | Prosellerlocation | Proselleremail | Prosellerpassword | Prosellerphoneno |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Sakshi Mehta | Female | Vasant Kunj | sakshi.mehta@gmail.com | sakshi@123 | 9765432102 |
| 2 | Priya Reddy | Female | Hitec City | priya.reddy@gmail.com | priya@123 | 9876087651 |
| 3 | Neha Sharma | Female | Powai | neha.sharma@gmail.com | neha@123 | 9876012347 |
| 4 | Pooja Bhatt | Female | Anna Nagar | pooja.bhatt@gmail.com | pooja@123 | 9876087656 |
| 5 | Swati Desai | Female | Alkapuri | swati.desai@gmail.com | swati@123 | 9876543287 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

Combining DISTINCT, ORDER BY, and TOP (or LIMIT)

Query:

```
SELECT DISTINCT Prosellerlocation FROM PropertySeller
ORDER BY Prosellerlocation ASC LIMIT 3;
```

```
mysql> SELECT DISTINCT Prosellerlocation
-> FROM PropertySeller
-> ORDER BY Prosellerlocation ASC
-> LIMIT 3;
+-----+
| Prosellerlocation |
+-----+
| Alkapuri |
| Anna Nagar |
| Ellisbridge |
+-----+
3 rows in set (0.03 sec)
```

USING IS NULL and IS NOT NULL

1. IS NULL Operator:

The **IS NULL** operator is used to check if a column contains a NULL value.

Syntax:

```
SELECT * FROM table_name WHERE column_name IS NULL;
```

Query:

```
select Prosellerid,Prosellername,Prosellergender from PropertySeller  
where Prosellerphoneno is null;
```

```
mysql> select Prosellerid,Prosellername,Prosellergender from PropertySeller where Prosellerphoneno is null;  
+-----+-----+  
| Prosellerid | Prosellername | Prosellergender |  
+-----+-----+  
| 1 | Sakshi Mehta | Female |  
+-----+-----+  
1 row in set (0.00 sec)
```

2. IS NOT NULL Operator:

The **IS NOT NULL** operator is used to check if a column does not contain a NULL value.

Syntax:

```
SELECT * FROM table_name WHERE column_name IS NOT NULL;
```

Query:

```
select Prosellerid,Prosellername,Prosellergender from PropertySeller  
where Prosellerphoneno is not null;
```

```
mysql> select Prosellerid,Prosellername,Prosellergender from PropertySeller where Prosellerphoneno is not null;  
+-----+-----+  
| Prosellerid | Prosellername | Prosellergender |  
+-----+-----+  
| 2 | Priya Reddy | Female |  
| 3 | Neha Sharma | Female |  
| 4 | Pooja Bhatt | Female |  
| 5 | Swati Desai | Female |  
| 6 | Rajesh Kumar | Male |  
| 7 | Anjali Verma | Female |  
| 8 | Divya Shah | Female |  
| 9 | Meera Khan | Female |  
| 10 | Shalini Ghosh | Female |  
+-----+-----+  
9 rows in set (0.00 sec)
```

CASE STATEMENT

The **CASE** statement in SQL allows you to implement conditional logic directly in your query, similar to if-else statements in programming.

Syntax:

```
SELECT
    column1,
    column2,
CASE
    WHEN condition1 THEN result1
    WHEN condition2 THEN result2
    ...
    ELSE default_result
END AS new_column_name
FROM table_name;
```

- **WHEN:** Specifies a condition.
- **THEN:** Specifies the result if the corresponding WHEN condition is true.
- **ELSE:** Specifies the result if none of the conditions are true (optional).
- **END:** Closes the CASE statement.

1. Simple CASE statement

Query:

```
mysql> select Proid,Proname,Procity,Category_id,
-> case
-> when Category_id=1 then 'Residential'
-> when Category_id=2 then 'IndependentHome'
-> when Category_id=3 then 'Commercial'
-> when Category_id=4 then 'Plots'
-> when Category_id=5 then 'PG Room'
-> else 'Other'
-> end as Category
-> from Property;
```

Proid	Proname	Procity	Category_id	Category
1	Luxury 2BHK Apartment	Mumbai	1	Residential
2	Green View Independent House	Pune	2	IndependentHome
3	Commercial Office Space	Delhi	3	Commercial
4	Residential Plot	Bangalore	4	Plots
5	PG Room for Rent	Chennai	5	PG Room
6	Oceanview 3BHK	Goa	1	Residential
7	Villa in Green Valley	Ahmedabad	2	IndependentHome
8	Tech Hub Office	Hyderabad	3	Commercial
9	Luxury Plot in Prime Area	Kolkata	4	Plots
10	PG Room Near IT Park	Bangalore	5	PG Room
11	Modern 2BHK Apartment	Mumbai	1	Residential
12	Independent House with Garden	Pune	2	IndependentHome
13	Commercial Space for Retail	Delhi	3	Commercial
14	Residential Plot for Sale	Bangalore	4	Plots
15	PG Room with AC	Chennai	5	PG Room
16	Luxury 1BHK Apartment	Delhi	1	Residential
17	Detached Independent Home	Ahmedabad	2	IndependentHome
18	Co-Working Office Space	Hyderabad	3	Commercial
19	Prime Residential Plot	Kolkata	4	Plots
20	PG Room in Safe Area	Pune	5	PG Room

20 rows in set (0.00 sec)

2. Searched CASE Statement:

Query:

```
mysql> select Proid,Proname,Procity,Proprice,
```

```
-> case
-> when Proprice > 10000000 then Proprice * 0.3
-> when Proprice between 400000 and 10000000 then Proprice * 0.2
-> when Proprice between 50000 and 400000 then Proprice * 0.1
-> else Proprice
-> end as DiscountPropertyPrice
-> from Property;
```

Proid	Proname	Procity	Proprice	DiscountPropertyPrice
1	Luxury 2BHK Apartment	Mumbai	7500000.00	1500000.000
2	Green View Independent House	Pune	12000000.00	3600000.000
3	Commercial Office Space	Delhi	25000000.00	7500000.000
4	Residential Plot	Bangalore	5000000.00	1000000.000
5	PG Room for Rent	Chennai	15000.00	15000.00
6	Oceanview 3BHK	Goa	9500000.00	1900000.000
7	Villa in Green Valley	Ahmedabad	18000000.00	5400000.000
8	Tech Hub Office	Hyderabad	15000000.00	4500000.000
9	Luxury Plot in Prime Area	Kolkata	8000000.00	1600000.000
10	PG Room Near IT Park	Bangalore	12000.00	12000.00
11	Modern 2BHK Apartment	Mumbai	5000000.00	1000000.000
12	Independent House with Garden	Pune	8500000.00	1700000.000
13	Commercial Space for Retail	Delhi	12000000.00	3600000.000
14	Residential Plot for Sale	Bangalore	6500000.00	1300000.000
15	PG Room with AC	Chennai	18000.00	18000.00
16	Luxury 1BHK Apartment	Delhi	3500000.00	700000.000
17	Detached Independent Home	Ahmedabad	15000000.00	4500000.000
18	Co-Working Office Space	Hyderabad	8500000.00	1700000.000
19	Prime Residential Plot	Kolkata	4000000.00	800000.000
20	PG Room in Safe Area	Pune	10000.00	10000.00

20 rows in set (0.00 sec)

BUILT IN FUNCTION

A **built-in function** is a function provided natively by the language or system. These functions are predefined and ready to use, allowing you to perform common operations without writing custom code for each one.

String Function:

String functions in SQL are used to manipulate and operate on character data stored in columns of a table.

1. CONCAT: Joins two or more strings together.

Query:

```
select concat(Cname,",",Cage) as Customer_Info from Customer limit 5;
```

```
mysql> select concat(Cname,' ',Cage) as Customer_Info from Customer limit 5;
+-----+
| Customer_Info |
+-----+
| Rohit Sharma30 |
| Sakshi Mehta28 |
| Vikram Desai35 |
| Priya Reddy26 |
| Arjun Singh32 |
+-----+
5 rows in set (0.00 sec)
```

2. UPPER: Converts a string to uppercase.

Query:

```
select upper(Cname) as Name from customer limit 2;
```

```
mysql> select upper(Cname) as Name from customer limit 2;
+-----+
| Name |
+-----+
| ROHIT SHARMA |
| SAKSHI MEHTA |
+-----+
2 rows in set (0.11 sec)
```

3. LOWER: Converts a string to lowercase.

Query:

```
select lower(Cname) as Name from customer limit 2;
```

```
mysql> select lower(Cname) as Name from customer limit 2;
+-----+
| Name |
+-----+
| rohit sharma |
| sakshi mehta |
+-----+
2 rows in set (0.00 sec)
```

4. LENGTH: Returns the length of a string.

Query:

```
select Cname,length(Cname) as Name from customer limit 5;
```

```
mysql> select Cname,length(Cname) as Name from customer limit 5;
+-----+-----+
| Cname      | Name   |
+-----+-----+
| Rohit Sharma |    12 |
| Sakshi Mehta |    12 |
| Vikram Desai |    12 |
| Priya Reddy  |    11 |
| Arjun Singh   |    11 |
+-----+
5 rows in set (0.00 sec)
```

5. SUBSTRING: Extracts a substring from a string.

Query:

```
select substring(Cname,3,6) as NameSubstring from Customer limit 5;
```

```
mysql> select substring(Cname,3,6) as NameSubstring from Customer limit 5;
+-----+
| NameSubstring |
+-----+
| hit Sh       |
| kshi M       |
| kram D       |
| iya Re       |
| jun Si       |
+-----+
5 rows in set (0.11 sec)
```

Math Function:

Math functions in SQL are used to perform mathematical operations on numeric data stored in columns of a table.

1. ROUND(): Rounds a number to a specified number of decimal places.

Query: SELECT ROUND(123.456, 2) AS RoundedValue;

```
mysql> SELECT ROUND(123.456, 2) AS RoundedValue;
+-----+
| RoundedValue |
+-----+
|      123.46 |
+-----+
1 row in set (0.00 sec)
```

2. **CEIL()**: Returns the smallest integer greater than or equal to a given number.

Query: SELECT CEIL(4.7) AS CeilValue;

```
mysql> SELECT CEIL(4.7) AS CeilValue;
+-----+
| CeilValue |
+-----+
|      5   |
+-----+
1 row in set (0.09 sec)
```

3. **FLOOR()**: Returns the largest integer less than or equal to a given number.

Query: SELECT FLOOR(4.7) AS FloorValue;

```
mysql> SELECT FLOOR(4.7) AS FloorValue;
+-----+
| FloorValue |
+-----+
|      4    |
+-----+
1 row in set (0.00 sec)
```

4. **ABS()**: Returns the absolute value of a number.

Query: SELECT ABS(-18) AS AbsoluteValue;

```
mysql> SELECT ABS(-18) AS AbsoluteValue;
+-----+
| AbsoluteValue |
+-----+
|      18     |
+-----+
1 row in set (0.11 sec)
```

5. **POWER()**: Returns the result of raising a number to a specified power.

Query: SELECT POWER(4, 4) AS PowerValue;

```
mysql> SELECT POWER(4,4) AS PowerValue;
+-----+
| PowerValue |
+-----+
|      256   |
+-----+
1 row in set (0.00 sec)
```

Date Function:

Date functions in SQL are used to perform operations on date and time values stored in columns of a table.

1. **curdate() or current_date:** Returns the current date.

Query: select curdate();

```
mysql> select curdate();
+-----+
| curdate() |
+-----+
| 2024-11-13 |
+-----+
1 row in set (0.00 sec)
```

2. **now() or current_timestamp:** Returns the date and time.

Query: select now();

```
mysql> select now();
+-----+
| now() |
+-----+
| 2024-11-13 01:31:27 |
+-----+
1 row in set (0.00 sec)
```

3. **current_time:** Return the current time.

Query: select current_time();

```
mysql> select current_time();
+-----+
| current_time |
+-----+
| 01:31:38 |
+-----+
1 row in set (0.00 sec)
```

4. **year(), month(), day():** Extracts the year, month, or day from a date.

Query: SELECT YEAR(Propostdate), MONTH(Propostdate), DAY(Propostdate) FROM Property LIMIT 3;

```
mysql> SELECT YEAR(Propostdate), MONTH(Propostdate), DAY(Propostdate) FROM Property LIMIT 3;
+-----+-----+-----+
| YEAR(Propostdate) | MONTH(Propostdate) | DAY(Propostdate) |
+-----+-----+-----+
| 2024 | 11 | 11 |
| 2024 | 11 | 11 |
| 2024 | 11 | 11 |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

5. **datediff():** Calculates the difference between two dates.

Query: SELECT DATEDIFF(NOW(), Propostdate) AS DaysSinceCreated FROM Property limit 3;

```
mysql> SELECT DATEDIFF(NOW(), Propostdate) AS DaysSinceCreated FROM Property limit 3;
+-----+
| DaysSinceCreated |
+-----+
| 2 |
| 2 |
| 2 |
+-----+
3 rows in set (0.12 sec)
```

6. date_format(): Formats a date based on a specified format.

Query: SELECT DATE_FORMAT(Propostdate, "%Y-%m-%d") AS Dateformat FROM Property LIMIT 3;

```
mysql> SELECT DATE_FORMAT(Propostdate, '%Y-%m-%d') AS Dateformat FROM Property LIMIT 3;
+-----+
| Dateformat |
+-----+
| 2024-11-11 |
| 2024-11-11 |
| 2024-11-11 |
+-----+
3 rows in set (0.00 sec)
```

Aggregate Function:

These functions perform calculations on multiple rows and return a single result.

They are commonly used with the GROUP BY clause to perform calculations on groups of rows.

1. COUNT(): Counts the number of rows.

Query: SELECT COUNT(*) AS TotalRows FROM PropertyFeatures;

```
mysql> SELECT COUNT(*) AS TotalRows FROM PropertyFeatures;
+-----+
| TotalRows |
+-----+
|      20 |
+-----+
1 row in set (0.01 sec)
```

2. SUM(): Calculates the sum of values in a column.

Query: SELECT SUM(Proprice) FROM Property WHERE Category_id = 1;

```
mysql> SELECT SUM(Proprice) FROM Property WHERE Category_id = 1;
+-----+
| SUM(Proprice) |
+-----+
| 25500000.00 |
+-----+
1 row in set (0.15 sec)
```

3. AVG(): Calculates the average of values in a column.

Query: SELECT AVG(Proprice) FROM Property;

```
mysql> SELECT AVG(Proprice) FROM Property;
+-----+
| AVG(Proprice) |
+-----+
| 8152750.000000 |
+-----+
1 row in set (0.01 sec)
```

4. MIN(): Returns the minimum value in a column.

Query: SELECT MIN(Proprice) FROM Property;

```
mysql> SELECT MIN(Proprice) FROM Property;
+-----+
| MIN(Proprice) |
+-----+
| 10000.00 |
+-----+
1 row in set (0.36 sec)
```

5. MAX(): Returns the maximum value in a column.

Query: SELECT MAX(Proprice) FROM Property;

```
mysql> SELECT MAX(Proprice) FROM Property;
+-----+
| MAX(Proprice) |
+-----+
| 25000000.00 |
+-----+
1 row in set (0.00 sec)
```

GROUP BY Clause with HAVING

1. GROUP BY Clause

Groups rows that have the same values into summary rows.

Syntax:

```
SELECT column1, aggregate_function (column2)
FROM table_name
WHERE condition
GROUP BY column1;
```

Query:

```
select Proname,count(Prostate) from Property
where Prostate='Maharashtra' group by Proname;
```

```
mysql> select Proname,count(Prostate) from Property where Prostate='Maharashtra' group by Proname;
+-----+-----+
| Proname          | count(Prostate) |
+-----+-----+
| Luxury 2BHK Apartment |      1 |
| Green View Independent House |      1 |
| Modern 2BHK Apartment |      1 |
| Independent House with Garden |      1 |
| PG Room in Safe Area |      1 |
+-----+-----+
5 rows in set (0.00 sec)
```

2. HAVING Clause

Filters groups based on specified conditions after the **GROUP BY** clause.

Syntax:

```
SELECT column1, aggregate_function(column2)
FROM table_name
WHERE condition
GROUP BY column1
HAVING aggregate_function (column2) > 1000;
```

Query:

```
select Proname,max(Proprice) from Property
where Category_id=1 group by Proname having max(Proprice)>5000000;
```

```
mysql> select Proname,max(Proprice) from Property where Category_id=1 group by Proname having max(Proprice)>5000000;
+-----+-----+
| Proname | max(Proprice) |
+-----+-----+
| Luxury 2BHK Apartment | 7500000.00 |
| Oceanview 3BHK | 9500000.00 |
+-----+-----+
2 rows in set (0.00 sec)
```

SUBQUERY

A **subquery** is a query within another SQL query. It is enclosed in parentheses and provides a result that the main query can use.

Subqueries are helpful for filtering results, performing calculations, and creating complex query logic.

1. Single Row Subquery:

A single-row subquery returns only one row. It's often used with comparison operators (**=, <, >, etc.**) in the main query to compare a single value.

Syntax:

```
SELECT column1, column2, ...
```

```
FROM table1
```

```
WHERE column_name = (SELECT single_column FROM table2 WHERE condition);
```

Query:

```
SELECT Proid, Proname, Proprice FROM Property
```

```
WHERE Proprice > (SELECT AVG(Proprice) FROM Property);
```

```
mysql> SELECT Proid, Proname, Proprice
-> FROM Property
-> WHERE Proprice > (SELECT AVG(Proprice) FROM Property);
+-----+-----+-----+
| Proid | Proname | Proprice |
+-----+-----+-----+
| 2 | Green View Independent House | 12000000.00 |
| 3 | Commercial Office Space | 25000000.00 |
| 6 | Oceanview 3BHK | 9500000.00 |
| 7 | Villa in Green Valley | 18000000.00 |
| 8 | Tech Hub Office | 15000000.00 |
| 12 | Independent House with Garden | 8500000.00 |
| 13 | Commercial Space for Retail | 12000000.00 |
| 17 | Detached Independent Home | 15000000.00 |
| 18 | Co-Working Office Space | 8500000.00 |
+-----+-----+-----+
9 rows in set (0.16 sec)
```

2. Multiple-row Subquery:

A multiple-row subquery returns more than one row. You must use operators that can handle multiple values, such as **IN, ANY, ALL**, etc., in the outer query.

Syntax:

```
SELECT column1, column2, ...
```

```
FROM table1
```

```
WHERE column IN (SELECT single_column FROM table2 WHERE condition);
```

Query:

```
SELECT Proid, Proname, Proprice FROM Property
```

```
WHERE Category_id IN (SELECT Category_id FROM PropertyCategory WHERE Category_id IN (1, 2, 3));
```

```
mysql> SELECT Proid, Proname, Proprice
    -> FROM Property
    -> WHERE Category_id IN (SELECT Category_id FROM PropertyCategory WHERE Category_id IN (1, 2, 3));
+----+-----+-----+
| Proid | Proname           | Proprice |
+----+-----+-----+
|   1  | Luxury 2BHK Apartment | 7500000.00 |
|   6  | Oceanview 3BHK       | 9500000.00 |
|  11  | Modern 2BHK Apartment | 5000000.00 |
|  16  | Luxury 1BHK Apartment | 3500000.00 |
|   2  | Green View Independent House | 12000000.00 |
|   7  | Villa in Green Valley | 18000000.00 |
|  12  | Independent House with Garden | 8500000.00 |
|  17  | Detached Independent Home | 15000000.00 |
|   3  | Commercial Office Space | 25000000.00 |
|   8  | Tech Hub Office       | 15000000.00 |
|  13  | Commercial Space for Retail | 12000000.00 |
|  18  | Co-Working Office Space | 8500000.00 |
+----+-----+-----+
12 rows in set (0.36 sec)
```

Example 2:

```
mysql> SELECT Proid, Proname, Proprice
    -> FROM Property
    -> where Proprice < any (select Proprice from Property where Category_id=1);
+----+-----+-----+
| Proid | Proname           | Proprice |
+----+-----+-----+
|   1  | Luxury 2BHK Apartment | 7500000.00 |
|   4  | Residential Plot     | 5000000.00 |
|   5  | PG Room for Rent     | 15000.00  |
|   9  | Luxury Plot in Prime Area | 8000000.00 |
|  10  | PG Room Near IT Park | 12000.00  |
|  11  | Modern 2BHK Apartment | 5000000.00 |
|  12  | Independent House with Garden | 8500000.00 |
|  14  | Residential Plot for Sale | 6500000.00 |
|  15  | PG Room with AC      | 18000.00  |
|  16  | Luxury 1BHK Apartment | 3500000.00 |
|  18  | Co-Working Office Space | 8500000.00 |
|  19  | Prime Residential Plot | 4000000.00 |
|  20  | PG Room in Safe Area | 10000.00  |
+----+-----+-----+
13 rows in set (0.38 sec)
```

Example 3&4:

```

mysql> SELECT Proid, Proname, Proprice
-> FROM Property
-> where Proprice > any (select Proprice from Property where Category_id=1);
+-----+-----+
| Proid | Proname           | Proprice |
+-----+-----+
| 1     | Luxury 2BHK Apartment | 7500000.00 |
| 2     | Green View Independent House | 12000000.00 |
| 3     | Commercial Office Space | 25000000.00 |
| 4     | Residential Plot | 5000000.00 |
| 6     | Oceanview 3BHK | 9500000.00 |
| 7     | Villa in Green Valley | 18000000.00 |
| 8     | Tech Hub Office | 15000000.00 |
| 9     | Luxury Plot in Prime Area | 8000000.00 |
| 11    | Modern 2BHK Apartment | 5000000.00 |
| 12    | Independent House with Garden | 8500000.00 |
| 13    | Commercial Space for Retail | 12000000.00 |
| 14    | Residential Plot for Sale | 6500000.00 |
| 17    | Detached Independent Home | 15000000.00 |
| 18    | Co-Working Office Space | 8500000.00 |
| 19    | Prime Residential Plot | 4000000.00 |
+-----+-----+
15 rows in set (0.00 sec)

mysql> SELECT Proid, Proname, Proprice
-> FROM Property
-> where Proprice < all (select Proprice from Property where Category_id=1);
+-----+-----+
| Proid | Proname           | Proprice |
+-----+-----+
| 5     | PG Room for Rent | 15000.00 |
| 10    | PG Room Near IT Park | 12000.00 |
| 15    | PG Room with AC | 18000.00 |
| 20    | PG Room in Safe Area | 10000.00 |
+-----+-----+
4 rows in set (0.00 sec)

```

3. Multiple-column Subquery:

A multiple column subquery returns more than one column to the outer query and can be listed in outer query from where and having clause.

Syntax:

SELECT column1, column2, ...

FROM table1

WHERE (column1, column2)

IN (SELECT column1, column2 FROM table2 WHERE condition);

Query:

SELECT Proid, Proname, Proprice, Category_id from Property

where (Proprice,Category_id) in (select Proprice,Category_id from Property where Proprice>3500000 and Category_id=1);

```
mysql> SELECT Proid, Proname, Proprice, Category_id from Property where (Proprice,Category_id) in (select Proprice,Category_id from Property where Proprice>3500000 and Category_id=1);
+----+-----+-----+-----+
| Proid | Proname | Proprice | Category_id |
+----+-----+-----+-----+
|    1 | Luxury 2BHK Apartment | 7500000.00 |          1 |
|    6 | Oceanview 3BHK         | 9500000.00 |          1 |
|   11 | Modern 2BHK Apartment | 5000000.00 |          1 |
+----+-----+-----+-----+
3 rows in set (0.00 sec)
```

JOINS

Joins in SQL are used to combine rows from two or more tables based on a related column between them. They allow you to retrieve data from multiple tables simultaneously and create a single result set.

1. INNER JOIN:

Returns rows with matching values in both tables.

Syntax:

SELECT column1, column2 FROM table1

INNER JOIN table2 ON table1.column = table2.column;

Query: If columnname are same in both table then use Using(columnname).

select p.Proid,p.Proname,p.Procity,c.Category_name from Property p

inner join PropertyCategory c on p.Category_id=c.Category_id / using(Category_id);

```
mysql> select p.Proid,p.Proname,p.Procity,c.Category_name from Property p inner join PropertyCategory c on p.Category_id=c.Category_id;
+----+-----+-----+-----+
| Proid | Proname | Procity | Category_name |
+----+-----+-----+-----+
|    1 | Luxury 2BHK Apartment | Mumbai | Residential |
|    6 | Oceanview 3BHK         | Goa   | Residential |
|   11 | Modern 2BHK Apartment | Mumbai | Residential |
|  16 | Luxury 1BHK Apartment | Delhi | Residential |
|    2 | Green View Independent House | Pune | IndependentHome |
|    7 | Villa in Green Valley | Ahmedabad | IndependentHome |
|  12 | Independent House with Garden | Pune | IndependentHome |
|  17 | Detached Independent Home | Ahmedabad | IndependentHome |
|    3 | Commercial Office Space | Delhi | Commercial |
|    8 | Tech Hub Office | Hyderabad | Commercial |
|  13 | Commercial Space for Retail | Delhi | Commercial |
|  18 | Co-Working Office Space | Hyderabad | Commercial |
|    4 | Residential Plot | Bangalore | Plots |
|    9 | Luxury Plot in Prime Area | Kolkata | Plots |
| 14 | Residential Plot for Sale | Bangalore | Plots |
| 19 | Prime Residential Plot | Kolkata | Plots |
|    5 | PG Room for Rent | Chennai | PG Room |
| 10 | PG Room Near IT Park | Bangalore | PG Room |
| 15 | PG Room with AC | Chennai | PG Room |
| 20 | PG Room in Safe Area | Pune | PG Room |
+----+-----+-----+-----+
20 rows in set (0.00 sec)
```

2. LEFT JOIN:

Returns all rows from the left table and matched rows from the right; unmatched rows from the right are NULL.

Syntax:

```
SELECT column1, column2 FROM table1
LEFT JOIN table2 ON table1.column = table2.column;
```

Query:

```
select p.Proid,p.Proname,p.Procity,c.Category_name from Property p
left join PropertyCategory c on p.Category_id=c.Category_id;
```

Proid	Proname	Procity	Category_name
1	Luxury 2BHK Apartment	Mumbai	Residential
2	Green View Independent House	Pune	IndependentHome
3	Commercial Office Space	Delhi	Commercial
4	Residential Plot	Bangalore	Plots
5	PG Room for Rent	Chennai	PG Room
6	Oceanview 3BHK	Goa	Residential
7	Villa in Green Valley	Ahmedabad	IndependentHome
8	Tech Hub Office	Hyderabad	Commercial
9	Luxury Plot in Prime Area	Kolkata	Plots
10	PG Room Near IT Park	Bangalore	PG Room
11	Modern 2BHK Apartment	Mumbai	Residential
12	Independent House with Garden	Pune	IndependentHome
13	Commercial Space for Retail	Delhi	Commercial
14	Residential Plot for Sale	Bangalore	Plots
15	PG Room with AC	Chennai	PG Room
16	Luxury 1BHK Apartment	Delhi	Residential
17	Detached Independent Home	Ahmedabad	IndependentHome
18	Co-Working Office Space	Hyderabad	Commercial
19	Prime Residential Plot	Kolkata	Plots
20	PG Room in Safe Area	Pune	PG Room
21	Indian Independent House	New Delhi	NULL
22	Indian Co-Working Space	Mumbai	NULL

3. RIGHT JOIN:

Returns all rows from the right table and matched rows from the left; unmatched rows from the left are NULL.

Syntax:

```
SELECT column1, column2 FROM table1
RIGHT JOIN table2 ON table1.column = table2.column;
```

Query:

```
select p.Proid,p.Proname,p.Procity,c.Category_name from Property p
right join PropertyCategory c on p.Category_id=c.Category_id;
```

```
mysql> select p.Proid,p.Proname,p.Procity,c.Category_name from Property p right join PropertyCategory c on p.Category_id=c.Category_id;
+----+-----+-----+-----+
| Proid | Proname | Procity | Category_name |
+----+-----+-----+-----+
| 1 | Luxury 2BHK Apartment | Mumbai | Residential |
| 6 | Oceanview 3BHK | Goa | Residential |
| 11 | Modern 2BHK Apartment | Mumbai | Residential |
| 16 | Luxury 1BHK Apartment | Delhi | Residential |
| 2 | Green View Independent House | Pune | IndependentHome |
| 7 | Villa in Green Valley | Ahmedabad | IndependentHome |
| 12 | Independent House with Garden | Pune | IndependentHome |
| 17 | Detached Independent Home | Ahmedabad | IndependentHome |
| 3 | Commercial Office Space | Delhi | Commercial |
| 8 | Tech Hub Office | Hyderabad | Commercial |
| 13 | Commercial Space for Retail | Delhi | Commercial |
| 18 | Co-Working Office Space | Hyderabad | Commercial |
| 4 | Residential Plot | Bangalore | Plots |
| 9 | Luxury Plot in Prime Area | Kolkata | Plots |
| 14 | Residential Plot for Sale | Bangalore | Plots |
| 19 | Prime Residential Plot | Kolkata | Plots |
| 5 | PG Room for Rent | Chennai | PG Room |
| 10 | PG Room Near IT Park | Bangalore | PG Room |
| 15 | PG Room with AC | Chennai | PG Room |
| 20 | PG Room in Safe Area | Pune | PG Room |
| NULL | NULL | NULL | Others |
+----+-----+-----+-----+
21 rows in set (0.00 sec)
```

4. FULL OUTER JOIN:

Retrieves all rows when there is a match in either the left or right table.; unmatched rows from each side are NULL.

Syntax:

```
SELECT table1.column1, table2.column2, ... FROM table1
```

```
LEFT JOIN table2
```

```
ON table1.common_column = table2.common_column
```

```
UNION
```

```
SELECT table1.column1, table2.column2, ... FROM table1
```

```
RIGHT JOIN table2
```

```
ON table1.common_column = table2.common_column;
```

Query:

```
select p.Proid,p.Proname,p.Procity,c.Category_name from Property p
```

```
left join PropertyCategory c
```

```
on p.Category_id=c.Category_id
```

```
union
```

```
select p.Proid,p.Proname,p.Procity,c.Category_name from Property p
```

```
right join PropertyCategory c
```

```
on p.Category_id=c.Category_id;
```

NIKHIL VANAJI WALUNJ

```
mysql> select p.Proid,p.Proname,p.Procity,c.Category_name from Property p left join PropertyCategory c on p.Category_id=c.Category_id union select p.Proid,p.Proname,p.Procity,c.Category_name from Property p right join PropertyCategory c on p.Category_id=c.Category_id;
+----+-----+-----+-----+
| Proid | Proname | Procity | Category_name |
+----+-----+-----+-----+
| 1 | Luxury 2BHK Apartment | Mumbai | Residential |
| 2 | Green View Independent House | Pune | IndependentHome |
| 3 | Commercial Office Space | Delhi | Commercial |
| 4 | Residential Plot | Bangalore | Plots |
| 5 | PG Room for Rent | Chennai | PG Room |
| 6 | Oceanview 3BHK | Goa | Residential |
| 7 | Villa in Green Valley | Ahmedabad | IndependentHome |
| 8 | Tech Hub Office | Hyderabad | Commercial |
| 9 | Luxury Plot in Prime Area | Kolkata | Plots |
| 10 | PG Room Near IT Park | Bangalore | PG Room |
| 11 | Modern 2BHK Apartment | Mumbai | Residential |
| 12 | Independent House with Garden | Pune | IndependentHome |
| 13 | Commercial Space for Retail | Delhi | Commercial |
| 14 | Residential Plot for Sale | Bangalore | Plots |
| 15 | PG Room with AC | Chennai | PG Room |
| 16 | Luxury 1BHK Apartment | Delhi | Residential |
| 17 | Detached Independent Home | Ahmedabad | IndependentHome |
| 18 | Co-Working Office Space | Hyderabad | Commercial |
| 19 | Prime Residential Plot | Kolkata | Plots |
| 20 | PG Room in Safe Area | Pune | PG Room |
| 21 | Indian Independent House | New Delhi | NULL |
| 22 | Indian Co-Working Space | Mumbai | NULL |
| NULL | NULL | NULL | Others |
+----+-----+-----+-----+
23 rows in set (0.33 sec)
```

5. CROSS JOIN:

Returns the Cartesian product of both tables (all possible row combinations).

Syntax:

SELECT column1, column2 FROM table1

CROSS JOIN table2;

Query:

```
select p.Proid,p.Proname,p.Procity,c.Category_name from Property p
cross join PropertyCategory c limit 7;
```

```
mysql> select p.Proid,p.Proname,p.Procity,c.Category_name from Property p cross join PropertyCategory c limit 7;
+----+-----+-----+-----+
| Proid | Proname | Procity | Category_name |
+----+-----+-----+-----+
| 1 | Luxury 2BHK Apartment | Mumbai | Others |
| 1 | Luxury 2BHK Apartment | Mumbai | PG Room |
| 1 | Luxury 2BHK Apartment | Mumbai | Plots |
| 1 | Luxury 2BHK Apartment | Mumbai | Commercial |
| 1 | Luxury 2BHK Apartment | Mumbai | IndependentHome |
| 1 | Luxury 2BHK Apartment | Mumbai | Residential |
| 2 | Green View Independent House | Pune | Others |
+----+-----+-----+-----+
7 rows in set (0.00 sec)
```

6. NATURAL JOIN:

A **Natural Join** is a type of join that automatically joins two tables based on all columns with the same names and compatible data types in both tables.

Syntax For One Table:

SELECT * FROM table1 NATURAL JOIN table2;

Syntax For Multiple Table:

SELECT * FROM table1

NATURAL JOIN table2 NATURAL JOIN table3;

Query:

select * from Property natural join PropertyCategory;

mysql> select * from Property natural join PropertyCategory;					
Category_id	Proid	Proname	Prodsc	Proaddress	Procity
1	1	Luxury 2BHK Apartment	A spacious 2BHK apartment in a prime location.	123 Park Street, Mumbai	Mumbai
1	6	Oceanview 3BHK	A luxury 3BHK apartment with ocean views and modern amenities.	234 Sea Breeze Road, Goa	Goa
1	11	Modern 2BHK Apartment	Modern apartment with all necessary amenities.	15 Sunrise Avenue, Mumbai	Mumbai
1	16	Luxury 1BHK Apartment	Stylish and compact 1BHK apartment in a premium location.	40 Skyline Building, Delhi	Delhi
2	2	Green View Independent House	A beautiful independent house with a garden and ample space.	45 Green Valley, Pune	Pune
2	7	Villa in Green Valley	A spacious villa with a private pool and large garden.	12 Green Valley, Ahmedabad	Ahmedabad
2	12	Independent House with Garden	Spacious independent house with a beautiful garden.	23 Hilltop Lane, Pune	Pune
2	17	Detached Independent Home	A beautiful independent home with a spacious backyard.	77 Sunset Blvd, Ahmedabad	Ahmedabad
3	3	Commercial Office Space	A commercial office space located in the business hub.	18 Business Park, Delhi	Delhi
3	8	Tech Hub Office	An office space designed for tech companies with modern infrastructure.	56 Silicon Street, Hyderabad	Hyderabad
3	13	Commercial Space for Retail	A commercial space located in a prime shopping area.	80 Mall Road, Delhi	Delhi
3	18	Co-Working Office Space	A modern co-working space with high-speed internet and conference rooms.	12 Tech Valley, Hyderabad	Hyderabad
4	4	Residential Plot	A well-located residential plot in a developing area.	60 Greenfield, Bangalore	Bangalore
4	9	Luxury Plot in Prime Area	A plot for residential development in a prime location.	500 Main Road, Kolkata	Kolkata
4	14	Residential Plot for Sale	Prime location residential plot available for sale.	900 City Center, Bangalore	Bangalore
4	19	Prime Residential Plot	A plot with excellent development potential in a growing neighborhood.	55 North Street, Kolkata	Kolkata
5	5	PG Room for Rent	A fully furnished PG room with all amenities included.	101 Comfort Street, Chennai	Chennai
5	10	PG Room Near IT Park	A comfortable PG room with food and laundry services.	78 IT Park Lane, Bangalore	Bangalore
5	15	PG Room with AC	Well-maintained PG room with air conditioning and other amenities.	30 Liberty Road, Chennai	Chennai
5	20	PG Room in Safe Area	Safe and secure PG room with easy access to public transport.	60 Safe Lane, Pune	Pune

20 rows in set (0.00 sec)

Prostate	Proprice	Procountry	Procontactno	Prolistingtype	Propostdate	Prostatus	Profeatureid	Category_name
Maharashtra	7500000.00	India	9876543210	For Sale	2024-11-11 20:51:32	Available	1	Residential
Goa	9500000.00	India	8888888888	For Sale	2024-11-11 20:51:32	Available	6	Residential
Maharashtra	5600000.00	India	9369876543	For Sale	2024-11-11 20:51:32	Available	11	Residential
Delhi	3500000.00	India	9912345678	For Sale	2024-11-11 20:51:32	Available	16	Residential
Maharashtra	12000000.00	India	9898989898	For Sale	2024-11-11 20:51:32	Available	2	IndependentHome
Gujarat	18000000.00	India	9612345678	For Sale	2024-11-11 20:51:32	Available	7	IndependentHome
Maharashtra	8500000.00	India	9843276543	For Sale	2024-11-11 20:51:32	Available	12	IndependentHome
Gujarat	15000000.00	India	9976543210	For Sale	2024-11-11 20:51:32	Available	17	IndependentHome
Delhi	25000000.00	India	9999999999	For Rent	2024-11-11 20:51:32	Available	3	Commercial
Telangana	15000000.00	India	9176543210	For Rent	2024-11-11 20:51:32	Available	8	Commercial
Delhi	12000000.00	India	9543678921	For Rent	2024-11-11 20:51:32	Available	13	Commercial
Telangana	9500000.00	India	9176543210	For Rent	2024-11-11 20:51:32	Available	18	Commercial
Karnataka	5000000.00	India	9876543210	For Sale	2024-11-11 20:51:32	Available	4	Plots
West Bengal	8000000.00	India	9712345678	For Sale	2024-11-11 20:51:32	Available	9	Plots
Karnataka	6500000.00	India	9643721890	For Sale	2024-11-11 20:51:32	Available	14	Plots
West Bengal	4000000.00	India	9800000000	For Sale	2024-11-11 20:51:32	Available	19	Plots
Tamil Nadu	15000.00	India	9500000000	For Rent	2024-11-11 20:51:32	Available	5	PG Room
Karnataka	12000.00	India	9512345678	For Rent	2024-11-11 20:51:32	Available	10	PG Room
Tamil Nadu	10000.00	India	9445678910	For Rent	2024-11-11 20:51:32	Available	15	PG Room
Maharashtra	10000.00	India	9823456789	For Rent	2024-11-11 20:51:32	Available	20	PG Room

7. SELF JOIN:

A **Self Join** is a join in which a table is joined with itself.

It is useful when you want to compare rows within the same table or when you want to create a relationship between rows in the same table.

Syntax:

SELECT a.column1, b.column2

FROM table_name AS a JOIN table_name AS b

ON a.common_column = b.common_column;

Query:

SELECT

```

ps1.Prosellerid AS Seller1_ID,
ps1.Prosellername AS Seller1_Name,
ps1.Prosellergender AS Seller1_Gender,
ps1.Prosellerlocation AS Location,
ps2.Prosellerid AS Seller2_ID,
ps2.Prosellername AS Seller2_Name,
ps2.Prosellergender AS Seller2_Gender

```

FROM

Propertyseller ps1

JOIN

Propertyseller ps2 ON ps1.Prosellerlocation = ps2.Prosellerlocation

WHERE

ps1.Prosellerid < ps2.Prosellerid;

```

mysql> SELECT
->     ps1.Prosellerid AS Seller1_ID,
->     ps1.Prosellername AS Seller1_Name,
->     ps1.Prosellergender AS Seller1_Gender,
->     ps1.Prosellerlocation AS Location,
->     ps2.Prosellerid AS Seller2_ID,
->     ps2.Prosellername AS Seller2_Name,
->     ps2.Prosellergender AS Seller2_Gender
-> FROM
->     Propertyseller ps1
-> JOIN
->     Propertyseller ps2 ON ps1.Prosellerlocation = ps2.Prosellerlocation
-> WHERE
->     ps1.Prosellerid < ps2.Prosellerid;
+-----+-----+-----+-----+-----+-----+
| Seller1_ID | Seller1_Name | Seller1_Gender | Location | Seller2_ID | Seller2_Name | Seller2_Gender |
+-----+-----+-----+-----+-----+-----+
|        4 | Pooja Bhatt | Female       | Pune      |        10 | Shalini Ghosh | Female      |
|        4 | Pooja Bhatt | Female       | Pune      |        14 | Priyanka Tanpure | Female      |
|       10 | Shalini Ghosh | Female       | Pune      |        14 | Priyanka Tanpure | Female      |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)

```

8. EQUI JOIN:

An **Equi Join** is a type of join that uses equality operators (**usually =**) to match rows between two tables based on a related column.

It retrieves rows where values in the specified columns are equal.

Syntax:

```
SELECT columns FROM table1
JOIN table2 ON table1.column = table2.column;
```

Query:

```
select p.Proid,p.Proname,p.Procity,c.Category_name
from Property p
join PropertyCategory c
on p.Category_id=c.Category_id;
```

```
mysql> select p.Proid,p.Proname,p.Procity,c.Category_name from Property p
-> join PropertyCategory c on p.Category_id=c.Category_id;
+----+-----+-----+-----+
| Proid | Proname           | Procity | Category_name |
+----+-----+-----+-----+
|    1 | Luxury 2BHK Apartment | Mumbai   | Residential
|    6 | Oceanview 3BHK       | Goa      | Residential
|   11 | Modern 2BHK Apartment | Mumbai   | Residential
|  16  | Luxury 1BHK Apartment | Delhi    | Residential
|    2 | Green View Independent House | Pune    | IndependentHome
|    7 | Villa in Green Valley | Ahmedabad | IndependentHome
|  12  | Independent House with Garden | Pune    | IndependentHome
|  17  | Detached Independent Home | Ahmedabad | IndependentHome
|    3 | Commercial Office Space | Delhi    | Commercial
|    8 | Tech Hub Office       | Hyderabad | Commercial
|  13  | Commercial Space for Retail | Delhi    | Commercial
|  18  | Co-Working Office Space | Hyderabad | Commercial
|    4 | Residential Plot       | Bangalore | Plots
|    9 | Luxury Plot in Prime Area | Kolkata  | Plots
|  14  | Residential Plot for Sale | Bangalore | Plots
|  19  | Prime Residential Plot  | Kolkata  | Plots
|    5 | PG Room for Rent       | Chennai  | PG Room
|  10  | PG Room Near IT Park  | Bangalore | PG Room
|  15  | PG Room with AC        | Chennai  | PG Room
|  20  | PG Room in Safe Area   | Pune     | PG Room
+----+-----+-----+-----+
20 rows in set (0.00 sec)
```

9. NON-EQUI JOIN:

A **Non-Equi Join** is a join condition that uses operators other than equality (**like <, >, <=, >=, !=**) to join rows between two tables.

It retrieves rows based on a range or comparison condition rather than strict equality.

Syntax:

SELECT columns

FROM table1

JOIN table2

ON table1.column < table2.column;

Query:

```
select p.Proid,p.Proname,p.Procity,p.Proprice,c.AveragePrice, c.Category_name
from Property p
join PropertyCategory c
on p.Category_id=c.Category_id
where p.Proprice > c.AveragePrice;
```

```
mysql> select p.Proid,p.Proname,p.Procity,p.Proprice,c.AveragePrice, c.Category_name from Property p
-> join PropertyCategory c on p.Category_id=c.Category_id where p.Proprice > c.AveragePrice;
+----+-----+-----+-----+-----+-----+
| Proid | Proname           | Procity   | Proprice | AveragePrice | Category_name |
+----+-----+-----+-----+-----+-----+
| 1   | Luxury 2BHK Apartment | Mumbai    | 7500000.00 | 6000000.00  | Residential  |
| 6   | Oceanview 3BHK       | Goa       | 9500000.00 | 6000000.00  | Residential  |
| 2   | Green View Independent House | Pune     | 12000000.00 | 10000000.00 | IndependentHome |
| 7   | Villa in Green Valley | Ahmedabad | 18000000.00 | 10000000.00 | IndependentHome |
| 17  | Detached Independent Home | Ahmedabad | 15000000.00 | 10000000.00 | IndependentHome |
| 3   | Commercial Office Space | Delhi     | 25000000.00 | 8000000.00  | Commercial   |
| 8   | Tech Hub Office       | Hyderabad | 15000000.00 | 8000000.00  | Commercial   |
| 13  | Commercial Space for Retail | Delhi     | 12000000.00 | 8000000.00  | Commercial   |
| 18  | Co-Working Office Space | Hyderabad | 8500000.00  | 8000000.00  | Commercial   |
| 9   | Luxury Plot in Prime Area | Kolkata   | 8000000.00  | 5000000.00  | Plots        |
| 14  | Residential Plot for Sale | Bangalore | 6500000.00  | 5000000.00  | Plots        |
| 5   | PG Room for Rent      | Chennai   | 15000.00    | 7000.00     | PG Room     |
| 10  | PG Room Near IT Park | Bangalore | 12000.00    | 7000.00     | PG Room     |
| 15  | PG Room with AC       | Chennai   | 18000.00    | 7000.00     | PG Room     |
| 20  | PG Room in Safe Area  | Pune     | 10000.00    | 7000.00     | PG Room     |
+----+-----+-----+-----+-----+-----+
15 rows in set (0.00 sec)
```

VIEWS

A **view** is essentially a virtual table that is based on the result of a query. It doesn't store data itself but rather displays data from one or more tables.

We can create a view by selecting field from one or more tables present in the database.

1. SIMPLE VIEW:

It is based on a single table. It doesn't include group by, subquery, join, aggregate. We can perform the DML operation.

Syntax:

```
CREATE VIEW view_name
```

```
AS
```

```
SELECT column1, column2, ...
```

```
FROM table_name WHERE condition;
```

```
Select * from view_name;
```

Query:

```
create view PropertyDetails
```

```
as
```

```
select Proid,Proname,Proaddress,Procity,Proprice
```

```
from Property
```

```
where Prostate="Maharashtra";
```

```
Query OK, 0 rows affected (0.42 sec)
```

```
select * from PropertyDetails;
```

```
mysql> create view PropertyDetails
-> as
-> select Proid,Proname,Proaddress,Procity,Proprice from Property where Prostate="Maharashtra";
Query OK, 0 rows affected (0.42 sec)

mysql> select * from PropertyDetails;
+----+-----+-----+-----+-----+
| Proid | Proname          | Proaddress        | Procity | Proprice |
+----+-----+-----+-----+-----+
|    1 | Luxury 2BHK Apartment | 123 Park Street, Mumbai | Mumbai | 7500000.00 |
|    2 | Green View Independent House | 45 Green Lane, Pune | Pune | 12000000.00 |
|   11 | Modern 2BHK Apartment | 15 Sunrise Avenue, Mumbai | Mumbai | 5000000.00 |
|   12 | Independent House with Garden | 23 Hilltop Lane, Pune | Pune | 8500000.00 |
|   20 | PG Room in Safe Area | 60 Safe Lane, Pune | Pune | 10000.00 |
|   22 | Indian Co-Working Space | 101 India Avenue, Mumbai | Mumbai | 5000000.00 |
+----+-----+-----+-----+-----+
6 rows in set (0.37 sec)
```

2. COMPLEX VIEW:

It is based on two or more table. It includes group by, subquery, join, aggregate.

Syntax:

CREATE VIEW ComplexView

AS

SELECT column1, column2, ...

FROM table1 JOIN table2 ON condition WHERE condition;

Query:

create view Propertyprice

as

select p.Proid,p.Proname,p.Procity,p.Proprice,c.AveragePrice, c.Category_name
from Property p

join PropertyCategory c

on p.Category_id=c.Category_id

where p.Proprice > c.AveragePrice;

Query OK, 0 rows affected (0.02 sec)

select * from Propertyprice;

```
mysql> create view Propertyprice
-> as
-> select p.Proid,p.Proname,p.Procity,p.Proprice,c.AveragePrice, c.Category_name from Property p
-> join PropertyCategory c on p.Category_id=c.Category_id where p.Proprice > c.AveragePrice;
Query OK, 0 rows affected (0.02 sec)

mysql> select * from Propertyprice;
+----+-----+-----+-----+-----+-----+
| Proid | Proname | Procity | Proprice | AveragePrice | Category_name |
+----+-----+-----+-----+-----+-----+
| 1 | Luxury 2BHK Apartment | Mumbai | 7500000.00 | 6000000.00 | Residential |
| 6 | Oceanview 3BHK | Goa | 9500000.00 | 6000000.00 | Residential |
| 2 | Green View Independent House | Pune | 12000000.00 | 10000000.00 | IndependentHome |
| 7 | Villa in Green Valley | Ahmedabad | 18000000.00 | 10000000.00 | IndependentHome |
| 17 | Detached Independent Home | Ahmedabad | 15000000.00 | 10000000.00 | IndependentHome |
| 3 | Commercial Office Space | Delhi | 25000000.00 | 8000000.00 | Commercial |
| 8 | Tech Hub Office | Hyderabad | 15000000.00 | 8000000.00 | Commercial |
| 13 | Commercial Space for Retail | Delhi | 12000000.00 | 8000000.00 | Commercial |
| 18 | Co-Working Office Space | Hyderabad | 8500000.00 | 8000000.00 | Commercial |
| 9 | Luxury Plot in Prime Area | Kolkata | 8000000.00 | 5000000.00 | Plots |
| 14 | Residential Plot for Sale | Bangalore | 6500000.00 | 5000000.00 | Plots |
| 5 | PG Room for Rent | Chennai | 15000.00 | 7000.00 | PG Room |
| 10 | PG Room Near IT Park | Bangalore | 12000.00 | 7000.00 | PG Room |
| 15 | PG Room with AC | Chennai | 18000.00 | 7000.00 | PG Room |
| 20 | PG Room in Safe Area | Pune | 10000.00 | 7000.00 | PG Room |
+----+-----+-----+-----+-----+-----+
15 rows in set (0.01 sec)
```

DML USING VIEWS:

Views can be used for performing DML operations, which allows manipulating data through views instead of directly on tables.

DML operations on views can simplify data access and provide a level of abstraction.

Firstly, we create view:

Query:

create view SellerView

as

```
select Prosellerid,Prosellername,Prosellergender,Prosellerlocation from PropertySeller;
```

Query OK, 0 rows affected (0.13 sec)

```
mysql> select * from SellerView;
```

```
mysql> create view SellerView
    -> as
    -> select Prosellerid,Prosellername,Prosellergender,Prosellerlocation from PropertySeller;
Query OK, 0 rows affected (0.13 sec)

mysql> select * from SellerView;
+-----+-----+-----+-----+
| Prosellerid | Prosellername | Prosellergender | Prosellerlocation |
+-----+-----+-----+-----+
|      1 | Sakshi Mehta | Female        | Vasant Kunj       |
|      2 | Priya Reddy  | Female        | Hitec City       |
|      3 | Neha Sharma  | Female        | Powai            |
|      4 | Pooja Bhatt  | Female        | Anna Nagar       |
|      5 | Swati Desai  | Female        | Alkapuri         |
|      6 | Rajesh Kumar | Male          | Janakpuri        |
|      7 | Anjali Verma | Female        | Sector 10        |
|      8 | Divya Shah   | Female        | Ellisbridge      |
|      9 | Meera Khan   | Female        | Kalina           |
|     10 | Shalini Ghosh | Female        | Salt Lake        |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

1. Inserting Data into a View

Syntax:

```
INSERT INTO view_name (column1, column2, ...)
```

```
VALUES (value1, value2, ...);
```

Query:

```
insert into SellerView(Prosellerid,Prosellername,Prosellergender,Prosellerlocation)
values(11,'Nikhil Walunj','Male','Dombivli');
```

```
mysql> insert into SellerView(Prosellerid,Prosellename,Prosellergender,Prosellerlocation)
-> values(11,'Nikhil Walunj','Male','Dombivli');
Query OK, 1 row affected (0.11 sec)

mysql> select * from SellerView;
+-----+-----+-----+-----+
| Prosellerid | Prosellename | Prosellergender | Prosellerlocation |
+-----+-----+-----+-----+
| 1 | Sakshi Mehta | Female | Vasant Kunj |
| 2 | Priya Reddy | Female | Hitec City |
| 3 | Neha Sharma | Female | Powai |
| 4 | Pooja Bhatt | Female | Anna Nagar |
| 5 | Swati Desai | Female | Alkapuri |
| 6 | Rajesh Kumar | Male | Janakpuri |
| 7 | Anjali Verma | Female | Sector 10 |
| 8 | Divya Shah | Female | Ellisbridge |
| 9 | Meera Khan | Female | Kalina |
| 10 | Shalini Ghosh | Female | Salt Lake |
| 11 | Nikhil Walunj | Male | Dombivli |
+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

2. Updating Data into a View

Syntax:

UPDATE view_name SET column1 = value1, column2 = value2, ...

WHERE condition;

Query:

update SellerView set Prosellerlocation="Sector 25"

where Prosellerid=7;

```
mysql> update SellerView set Prosellerlocation="Sector 25" where Prosellerid=7;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select * from SellerView;
+-----+-----+-----+-----+
| Prosellerid | Prosellename | Prosellergender | Prosellerlocation |
+-----+-----+-----+-----+
| 1 | Sakshi Mehta | Female | Vasant Kunj |
| 2 | Priya Reddy | Female | Hitec City |
| 3 | Neha Sharma | Female | Powai |
| 4 | Pooja Bhatt | Female | Anna Nagar |
| 5 | Swati Desai | Female | Alkapuri |
| 6 | Rajesh Kumar | Male | Janakpuri |
| 7 | Anjali Verma | Female | Sector 25 |
| 8 | Divya Shah | Female | Ellisbridge |
| 9 | Meera Khan | Female | Kalina |
| 10 | Shalini Ghosh | Female | Salt Lake |
| 11 | Nikhil Walunj | Male | Dombivli |
+-----+-----+-----+-----+
11 rows in set (0.00 sec)
```

3. Deleting Data into a View

Syntax:

```
DELETE FROM view_name WHERE condition;
```

Query:

```
delete from SellerView where Prosellerid=11;
```

```
mysql> delete from SellerView where Prosellerid=11;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from SellerView;
+-----+-----+-----+-----+
| Prosellerid | Prosellename | Prosellergender | Prosellerlocation |
+-----+-----+-----+-----+
| 1 | Sakshi Mehta | Female | Vasant Kunj
| 2 | Priya Reddy | Female | Hitec City
| 3 | Neha Sharma | Female | Powai
| 4 | Pooja Bhatt | Female | Anna Nagar
| 5 | Swati Desai | Female | Alkapuri
| 6 | Rajesh Kumar | Male | Janakpuri
| 7 | Anjali Verma | Female | Sector 25
| 8 | Divya Shah | Female | Ellisbridge
| 9 | Meera Khan | Female | Kalina
| 10 | Shalini Ghosh | Female | Salt Lake
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

4. CHECK Constraints into view:

Specify conditions that each row must satisfy when data is modified through the view.

Syntax:

```
CREATE VIEW view_name
```

```
AS
```

```
SELECT column1, column2, ... FROM table_name WHERE condition
```

```
WITH CHECK OPTION;
```

Query:

```
create view CustomerDetails
```

```
as
```

```
select * from Customer where Cage > 30 with check option;
```

```
Query OK, 0 rows affected (0.01 sec)
```

```
select * from CustomerDetails;
```

```
mysql> create view CustomerDetails
-> as
-> select * from Customer where Cage > 30 with check option;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from CustomerDetails;
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Cid | Cname | Cage | Caddress | Cgender | Cemail | Cpassword | Cphoneno | Ccity | Cpincode | Cstate | Ctype |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 3  | Vikram Desai | 35 | 43 JP Nagar | Male | vikram.desai@gmail.com | vikram@123 | 9876043124 | Bangalore | 560078 | Karnataka | Buyer |
| 5  | Arjun Singh | 32 | 98 South Avenue | Male | arjun.singh@gmail.com | arjun@123 | 9876532198 | Delhi | 110011 | Delhi | Buyer |
| 7  | Kiran Patel | 34 | 14 Maninagar | Male | kiran.patel@gmail.com | kiran@123 | 9876554322 | Ahmedabad | 380008 | Gujarat | Buyer |
| 9  | Rahul Joshi | 31 | 75 MG Road | Male | rahul.joshi@gmail.com | rahul@123 | 9876078452 | Pune | 411001 | Maharashtra | Buyer |
| 11 | Vivek Sharma | 38 | 15 Connaught Place | Male | vivek.sharma@gmail.com | vivek@123 | 9876076534 | Delhi | 110001 | Delhi | Buyer |
| 12 | Ritika Nair | 33 | 44 Whitefield | Female | ritika.nair@gmail.com | ritika@123 | 9876021236 | Bangalore | 560066 | Karnataka | Buyer |
| 13 | Rajesh Kumar | 36 | 5 Janakpuri | Male | rajesh.kumar@gmail.com | rajesh@123 | 9876043217 | Delhi | 110058 | Delhi | Saler |
| 15 | Amit Agarwal | 40 | 29 Salt Lake | Male | amit.agarwal@gmail.com | amit@123 | 9876067891 | Kolkata | 700091 | West Bengal | Buyer |
| 16 | Anjali Verma | 31 | 37 Sector 10 | Female | anjali.verma@gmail.com | anjali@123 | 9876087653 | Noida | 201301 | Uttar Pradesh | Saler |
| 19 | Nikhil Sinha | 34 | 19 Connaught Place | Male | nikhil.sinha@gmail.com | nikhil@123 | 9876078941 | Delhi | 110001 | Delhi | Buyer |
| 21 | Ramesh Mehta | 32 | 8 Sector 15 | Male | ramesh.mehta@gmail.com | ramesh@123 | 9876012348 | Chandigarh | 160015 | Punjab | Buyer |
| 25 | Mohit Singh | 33 | 10 Golf Course Road | Male | mohit.singh@gmail.com | mohit@123 | 9876087657 | Gurgaon | 122003 | Haryana | Buyer |
| 26 | Anupam Bhatia | 37 | 91 Ballygunge | Male | anupam.bhatia@gmail.com | anupam@123 | 9876098768 | Kolkata | 700019 | West Bengal | None |
+----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
13 rows in set (0.00 sec)
```

5. UNIQUE Constraints into view:

Ensure no duplicate values exist in specified columns.

Syntax:

CREATE VIEW view_name

AS SELECT DISTINCT column1, column2, ...

FROM table_name;

Query:

create view UniqueCustomerDetails

as

select distinct Cstate from Customer;

Query OK, 0 rows affected (0.02 sec)

select * from UniqueCustomerDetails;

```
mysql> create view UniqueCustomerDetails
-> as
-> select distinct Cstate from Customer;
Query OK, 0 rows affected (0.02 sec)

mysql> select * from UniqueCustomerDetails;
+-----+
| Cstate |
+-----+
| Maharashtra |
| Delhi |
| Karnataka |
| Telangana |
| Gujarat |
| Tamil Nadu |
| Uttar Pradesh |
| West Bengal |
| Haryana |
| Punjab |
+-----+
10 rows in set (0.00 sec)
```

6. IS NOT NULL Constraints into view:

Ensure columns cannot contain null values.

Syntax:

```
CREATE VIEW view_name
```

```
AS SELECT column1, column2, ... FROM table_name
```

```
WHERE column1 IS NOT NULL;
```

Query:

```
create view notnullcustomerdetails
```

```
as
```

```
select * from Customer where Cemail IS NOT NULL;
```

```
Query OK, 0 rows affected (0.02 sec)
```

```
select * from notnullcustomerdetails;
```

Cid	Cname	Cage	Caddress	Cgender	Cemail	Cpassword	Cphoneno	Ccity	Cpincode	Cstate	Ctype
1	Rohit Sharma	30	21 Marine Drive	Male	rohit.sharma@gmail.com	rohit@123	9876543211	Mumbai	400020	Maharashtra	Buyer
2	Sakshi Mehta	28	12 Vasant Kunj	Female	sakshi.mehta@gmail.com	sakshi@123	9765432102	Delhi	110070	Delhi	Saler
3	Vikram Desai	35	43 JP Nagar	Male	vikram.desai@gmail.com	vikram@123	9876043124	Bangalore	560078	Karnataka	Buyer
4	Priya Reddy	26	17 Hitec City	Female	priya.reddy@gmail.com	priya@123	9876087651	Hyderabad	500081	Telangana	Saler
5	Arjun Singh	32	98 South Avenue	Male	arjun.singh@gmail.com	arjun@123	9876532198	Delhi	110011	Delhi	Buyer
6	Neha Sharma	27	55 Powai	Female	neha.sharma@gmail.com	neha@123	98760012347	Mumbai	400076	Maharashtra	Saler
7	Kiran Patel	34	14 Maninagar	Male	kiran.patel@gmail.com	kiran@123	9876554322	Ahmedabad	380008	Gujarat	Buyer
8	Pooja Bhatt	30	27 Anna Nagar	Female	pooja.bhatt@gmail.com	pooja@123	9876087656	Chennai	600040	Tamil Nadu	Saler
9	Rahul Joshi	31	75 MG Road	Male	rahul.joshi@gmail.com	rahul@123	9876078452	Pune	411001	Maharashtra	Buyer
10	Swati Desai	29	66 Alkapuri	Female	swati.desai@gmail.com	swati@123	9876543287	Vadodara	390007	Gujarat	Saler
11	Vivek Sharma	38	15 Connaught Place	Male	vivek.sharma@gmail.com	vivek@123	9876076534	Delhi	110001	Delhi	Buyer
12	Ritika Nair	33	44 Whitefield	Female	ritika.nair@gmail.com	ritika@123	9876021236	Bangalore	560066	Karnataka	Buyer
13	Rajesh Kumar	36	5 Janakpuri	Male	rajesh.kumar@gmail.com	rajesh@123	9876043217	Delhi	110058	Delhi	Saler
14	Sunita Tiwari	27	18 Sector 56	Female	sunita.tiwari@gmail.com	sunita@123	9876098763	Noida	201301	Uttar Pradesh	Buyer
15	Amit Agarwal	40	29 Salt Lake	Male	amit.agarwal@gmail.com	amit@123	9876067891	Kolkata	700091	West Bengal	Buyer
16	Anjali Verma	31	37 Sector 10	Female	anjali.verma@gmail.com	anjali@123	9876087653	Noida	201301	Uttar Pradesh	Saler
17	Manish Chawla	29	22 DLF Phase 2	Male	manish.chawla@gmail.com	manish@123	9876054323	Gurgaon	122002	Haryana	Buyer
18	Geeta Iyer	26	11 Koramangala	Female	geeta.iyer@gmail.com	geeta@123	9876021239	Bangalore	560095	Karnataka	None
19	Nikhil Sinha	34	19 Connaught Place	Male	nikhil.sinha@gmail.com	nikhil@123	9876078941	Delhi	110001	Delhi	Buyer
20	Divya Shah	30	3 Ellisbridge	Female	divya.shah@gmail.com	divya@123	9876076523	Ahmedabad	380006	Gujarat	Saler
21	Ramesh Mehta	32	8 Sector 15	Male	ramesh.mehta@gmail.com	ramesh@123	98760012348	Chandigarh	160015	Punjab	Buyer
22	Meera Khan	28	23 Kalina	Female	meera.khan@gmail.com	meera@123	9876065421	Mumbai	400098	Maharashtra	Saler
23	Suresh Yadav	30	67 Baner	Male	suresh.yadav@gmail.com	suresh@123	9876098769	Pune	411045	Maharashtra	Buyer
24	Shalini Ghosh	29	77 Salt Lake	Female	shalini.ghosh@gmail.com	shalini@123	9876543215	Kolkata	700064	West Bengal	Saler
25	Mohit Singh	33	10 Golf Course Road	Male	mohit.singh@gmail.com	mohit@123	9876087657	Gurgaon	122003	Haryana	Buyer
26	Anupam Bhatia	37	91 Ballygunge	Male	anupam.bhatia@gmail.com	anupam@123	9876098768	Kolkata	700019	West Bengal	None
27	Ravi Rao	29	5 Jayanagar	Male	ravi.rao@gmail.com	ravi@123	9876076524	Bangalore	560041	Karnataka	Buyer

7. DEFAULT Constraints into view:

Specify a default value for a column if no value is specified during an insert operation.

Syntax:

```
CREATE VIEW view_name
```

```
AS SELECT column1, column2,
```

```
COALESCE(column3,'value')...
```

```
FROM table_name
```

Query:

create view Defaultseller

as

```
select Prosellerid,Prosellername,Prosellergender,Prosellerlocation,  
coalesce(Prosellerphoneno,'NOT AVAILABLE') as Prosellerphoneno  
from PropertySeller;
```

Query OK, 0 rows affected (0.02 sec)

```
select * from Defaultseller;
```

```
mysql> create view Defaultseller  
-> as  
-> select Prosellerid,Prosellername,Prosellergender,Prosellerlocation,coalesce(Prosellerphoneno,'NOT AVAILABLE') as Prosellerphoneno  
-> from PropertySeller;  
Query OK, 0 rows affected (0.02 sec)  
  
mysql> select * from Defaultseller;  
+-----+-----+-----+-----+-----+  
| Prosellerid | Prosellername | Prosellergender | Prosellerlocation | Prosellerphoneno |  
+-----+-----+-----+-----+-----+  
| 1 | Sakshi Mehta | Female | Vasant Kunj | 9765432102 |  
| 2 | Priya Reddy | Female | Hitec City | 9876087651 |  
| 3 | Neha Sharma | Female | Powai | 9876012347 |  
| 4 | Pooja Bhatt | Female | Anna Nagar | 9876087656 |  
| 5 | Swati Desai | Female | Alkapuri | 9876543287 |  
| 6 | Rajesh Kumar | Male | Janakpuri | 9876043217 |  
| 7 | Anjali Verma | Female | Sector 25 | 9876087653 |  
| 8 | Divya Shah | Female | Ellisbridge | 9876076523 |  
| 9 | Meera Khan | Female | Kalina | 9876065421 |  
| 10 | Shalini Ghosh | Female | Salt Lake | 9876543215 |  
| 11 | Nikhil Walunj | Male | Dombivli | NOT AVAILABLE |  
+-----+-----+-----+-----+-----+  
11 rows in set (0.00 sec)
```

STORED PROCEDURE

A stored procedure is a set of SQL statements that can be stored in the database server and executed later by calling the procedure.

This is precompiled SQL statement.

Stored procedures are especially useful when you want to automate repetitive tasks.

Syntax:

DELIMITER &&

```
CREATE PROCEDURE procedure_name([IN | OUT | INOUT ] parameter_name datatype)
```

BEGIN

 Declaration_section

 Executable_section

END &&

DELIMITER;

1. FOR IN:

Query:

```
mysql> delimiter &&
mysql> create procedure showdetails(IN var int)
-> begin
-> select * from PropertySeller where Prosellerid=var;
-> end &&
```

Query OK, 0 rows affected (0.43 sec)

```
mysql> delimiter ;
mysql> call showdetails(5);
```

```
mysql> delimiter &&
mysql> create procedure showdetails(IN var int)
-> begin
-> select * from PropertySeller where Prosellerid=var;
-> end &&
Query OK, 0 rows affected (0.43 sec)

mysql> delimiter ;
mysql> call showdetails(5);
+-----+-----+-----+-----+-----+-----+
| Prosellerid | Prosellername | Prosellergender | Prosellerlocation | Proselleremail | Prosellerpassword | Prosellerphoneno |
+-----+-----+-----+-----+-----+-----+
|      5 | Swati Desai | Female       | Alkapuri        | swati.desai@gmail.com | swati@123        | 9876543287      |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.01 sec)
```

2. FOR OUT:

Query:

```
mysql> delimiter &&
mysql> create procedure Prosellerdata(OUT var int)
-> begin
-> select count(Prosellername) into var from Propertyseller;
-> end &&
```

Query OK, 0 rows affected (0.38 sec)

```
mysql> delimiter ;
mysql> call Prosellerdata(@var);
Query OK, 1 row affected (0.01 sec)

mysql> select @var;
```

```
mysql> delimiter &&
mysql> create procedure Prosellerdata(OUT var int)
      -> begin
      -> select count(Prosellename) into var from Propertyseller;
      -> end &&
Query OK, 0 rows affected (0.38 sec)

mysql> delimiter ;
mysql> call Prosellerdata(@var);
Query OK, 1 row affected (0.01 sec)

mysql> select @var;
+-----+
| @var |
+-----+
|    11 |
+-----+
1 row in set (0.01 sec)
```

3. FOR INOUT:

Query:

```
mysql> delimiter &&
mysql> create procedure Propertydata(IN price int,OUT result int)
      -> begin
      -> select count(Proid) into result from Property where Proprice > price;
      -> end &&
Query OK, 0 rows affected (0.50 sec)
```

```
mysql> delimiter ;
mysql> call Propertydata(12000000,@result);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select @result;
```

```
mysql> delimiter &&
mysql> create procedure Propertydata(IN price int,OUT result int)
      -> begin
      -> select count(Proid) into result from Property where Proprice > price;
      -> end &&
Query OK, 0 rows affected (0.50 sec)

mysql> delimiter ;
mysql> call Propertydata(12000000,@result);
Query OK, 1 row affected (0.01 sec)

mysql> select @result;
+-----+
| @result |
+-----+
|      4 |
+-----+
1 row in set (0.00 sec)
```

4. CONDITIONAL STATEMENT:

Conditional statements in SQL stored procedures and functions allow logic to be applied based on specified conditions, enhancing flexibility and control over data processing.

Query:

```
mysql> delimiter &&
mysql> create procedure Propertydiscount_price(IN proprice int)
-> begin
-> set @discount=NULL;
-> if proprice > 10000000 then
-> set @discount=proplice * 0.2;
-> elseif proprice >20000 then
-> set @discount=proplice * 0.1;
-> else
-> set @discount="NO DISCOUNT APPLY";
-> end if;
-> end &&
```

Query OK, 0 rows affected (0.01 sec)

```
mysql> delimiter ;
mysql> call Propertydiscount_price(15000000);
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> select @discount;
```

```
mysql> delimiter &&
mysql> create procedure Propertydiscount_price(IN proprice int)
-> begin
-> set @discount=NULL;
-> if proprice > 10000000 then
-> set @discount=proplice * 0.2;
-> elseif proprice >20000 then
-> set @discount=proplice * 0.1;
-> else
-> set @discount="NO DISCOUNT APPLY";
-> end if;
-> end &&
Query OK, 0 rows affected (0.01 sec)
```

```
mysql> delimiter ;
mysql> call Propertydiscount_price(15000000);
Query OK, 0 rows affected (0.01 sec)

mysql> select @discount;
+-----+
| @discount |
+-----+
| 3000000.0 |
+-----+
1 row in set (0.00 sec)
```

5. LOOPING STATEMENT:

Looping statements in SQL are used to iterate over a set of data or perform repetitive tasks based on specified conditions.

Syntax:

[labelname]: Loop

If condition then

 Leave [labelname];

End if;

End Loop;

Query:

```
mysql> DELIMITER //
mysql> CREATE PROCEDURE DisplayProperties()
-> BEGIN
->     DECLARE total_rows INT;
->     DECLARE counter INT DEFAULT 1;
->
->     -- Get the total number of rows in the Property table
->     SELECT COUNT(*) INTO total_rows FROM Property;
->
--Looping statement
->     property_loop: LOOP
->         -- Exit the loop if the counter exceeds the total rows
->         IF counter > total_rows THEN
->             LEAVE property_loop;
->         END IF;
->
->         -- Display details of the current property
```

NIKHIL VANAJI WALUNJ

```
-> SELECT CONCAT('ID: ', Proid, ', Name: ', Proname, ', City: ', Procity, ', Price: ', Proprice, ', Status: ',  
Prostatus) AS PropertyDetails  
-> FROM Property  
-> WHERE Proid = counter;  
  
-> -- Increment the counter to move to the next property  
-> SET counter = counter + 1;  
-> END LOOP property_loop;  
-> END //
```

Query OK, 0 rows affected (2.15 sec)

mysql> DELIMITER ;

mysql> CALL DisplayProperties();

Result show property detail of each property in one column

```
mysql> DELIMITER //  
mysql>  
mysql> CREATE PROCEDURE DisplayProperties()  
-> BEGIN  
->     DECLARE total_rows INT;  
->     DECLARE counter INT DEFAULT 1;  
->  
->     -- Get the total number of rows in the Property table  
->     SELECT COUNT(*) INTO total_rows FROM Property;  
->  
->     property_loop: LOOP  
->         -- Exit the loop if the counter exceeds the total rows  
->         IF counter > total_rows THEN  
->             LEAVE property_loop;  
->         END IF;  
->  
->         -- Display details of the current property  
->         SELECT CONCAT('ID: ', Proid, ', Name: ', Proname, ', City: ', Procity, ', Price: ', Proprice, ', Status: ', Prostatus) AS PropertyDetails  
->             FROM Property  
->             WHERE Proid = counter;  
->  
->         -- Increment the counter to move to the next property  
->         SET counter = counter + 1;  
->     END LOOP property_loop;  
->  
-> END //  
Query OK, 0 rows affected (2.15 sec)  
  
mysql>  
mysql> DELIMITER ;  
mysql> CALL DisplayProperties();  
+-----+  
| PropertyDetails |  
+-----+  
| ID: 1, Name: Luxury 2BHK Apartment, City: Mumbai, Price: 7500000.00, Status: Sold |  
+-----+  
1 row in set (0.49 sec)  
  
+-----+  
| PropertyDetails |  
+-----+  
| ID: 2, Name: Green View Independent House, City: Pune, Price: 12000000.00, Status: Sold |  
+-----+  
1 row in set (0.49 sec)  
  
+-----+  
| PropertyDetails |  
+-----+  
| ID: 21, Name: Indian Independent House, City: New Delhi, Price: 12000000.00, Status: Sold |  
+-----+  
1 row in set (0.53 sec)  
  
+-----+  
| PropertyDetails |  
+-----+  
| ID: 22, Name: Indian Co-Working Space, City: Mumbai, Price: 5000000.00, Status: Sold |  
+-----+  
1 row in set (0.54 sec)  
  
Query OK, 0 rows affected (0.54 sec)
```

STORED FUNCTION

Functions in SQL are reusable blocks of code that accept parameters, perform calculations, and return a single value. They enhance SQL's capability by encapsulating logic for common operations.

Syntax:

Delimiter &&

```
CREATE FUNCTION function_name (parameters)
RETURNS data_type
BEGIN
    -- SQL statements
RETURN some_value;
END &&;
```

Query:

```
mysql> delimiter &&
mysql> create function getsellerdetails(Prosellerid int)
->     returns varchar(100)
->     deterministic
->     begin
->         declare sellerdetails varchar(100);
->         select CONCAT(Prosellername, ', ', Prosellerlocation)
->             into sellerdetails
->             from PropertySeller
->             where Prosellerid = Prosellerid
->             limit 1;
->         return sellerdetails;
->     end &&
```

Query OK, 0 rows affected (0.38 sec)

```
mysql> delimiter ;
mysql> select getsellerdetails(1);
```

```
mysql> delimiter &&
mysql> create function getsellerdetails(Prosellerid int)
->      returns varchar(100)
->      deterministic
->      begin
->          declare sellerdetails varchar(100);
->          select CONCAT(Prosellernname, ' ', ' ', Prosellerlocation)
->          into sellerdetails
->          from PropertySeller
->          where Prosellerid = Prosellerid
->          limit 1;
->          return sellerdetails;
->      end &&
Query OK, 0 rows affected (0.38 sec)

mysql> delimiter ;
mysql> select getsellerdetails(1);
+-----+
| getsellerdetails(1) |
+-----+
| Sakshi Mehta, Vasant Kunj |
+-----+
1 row in set (0.00 sec)
```

TRIGGERS

A **trigger** in SQL is a set of actions or SQL statements that are automatically executed (or "triggered") in response to specific events occurring in a database.

Syntax:

Create trigger triggername {before|after}

{Insert|Update|Delete}

On

Tablename

For each row

Begin

Trigger body

End

1. FOR INSERT:

Triggered when a new record is inserted into the table.

Query:

```
mysql> create trigger insert_seller after insert on PropertySeller
```

-> for each row

-> begin

```
-> insert into sellerauditlog(sellerid,Action,seller_name,Action_time) values(New.Prosellerid,"ProSeller Added",New.Prosellename,now());
```

-> end &&

Query OK, 0 rows affected (0.15 sec)

```
mysql> delimiter ;
```

```
mysql> select * from sellerauditlog;
```

Empty set (0.10 sec)

```
mysql> insert into PropertySeller values(13, "Rutika Bhirwandekar", "Female", "Jogeshwari", "rutika.bhi@gmail.com", "rutika@123", "9345245363");
```

Query OK, 1 row affected (0.14 sec)

```
mysql> insert into PropertySeller values(14, "Priyanka Tanpure", "Female", "Pune", "priyanka.tanpure@gmail.com", "priyanka@123", "9397749836");
```

Query OK, 1 row affected (0.36 sec)

```
mysql> select * from sellerauditlog;
```

```
mysql> create trigger insert_seller after insert on PropertySeller
-> for each row
-> begin
-> insert into sellerauditlog(sellerid,Action,seller_name,Action_time) values(New.Prosellerid,"ProSeller Added",New.Prosellename,now());
-> end &&
Query OK, 0 rows affected (0.15 sec)
```

```
mysql> delimiter ;
```

```
mysql> select * from sellerauditlog;
Empty set (0.10 sec)
```

```
mysql> insert into PropertySeller values(13,"Rutika Bhirwandekar","Female","Jogeshwari","rutika.bhi@gmail.com","rutika@123","9345245363");
Query OK, 1 row affected (0.14 sec)
```

```
mysql> insert into PropertySeller values(14,"Priyanka Tanpure","Female","Pune","priyanka.tanpure@gmail.com","priyanka@123","9397749836");
Query OK, 1 row affected (0.36 sec)
```

```
mysql> select * from sellerauditlog;
+-----+-----+-----+-----+
| sellerid | Action      | seller_name   | Action_time  |
+-----+-----+-----+-----+
|    13 | ProSeller Added | Rutika Bhirwandekar | 2024-11-14 19:09:15 |
|    14 | ProSeller Added | Priyanka Tanpure   | 2024-11-14 19:10:33 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

2. FOR UPDATE:

Triggered when an existing record is updated into the table.

Query:

```
mysql> delimiter &&
```

```
mysql> create trigger update_seller before update on PropertySeller
```

```
    -> for each row
```

```
    -> begin
```

```
        -> insert into sellerauditlog(sellerid, Action, seller_name, Action_time, previous_data, new_data)
values(New.Prosellerid, "ProSeller Updated", New.Prosellename, now(), Old.Prosellerlocation,
New.Prosellerlocation);
```

```
    -> end &&
```

```
Query OK, 0 rows affected (0.47 sec)
```

```
mysql> delimiter ;
```

```
mysql> update PropertySeller set Prosellerlocation="Pimpri" where Prosellerid=1;
```

```
Query OK, 1 row affected (0.12 sec)
```

```
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from sellerauditlog;
```

```
mysql> delimiter &&
mysql> create trigger update_seller before update on PropertySeller
    -> for each row
    -> begin
        -> insert into sellerauditlog(sellerid,Action,seller_name,Action_time,previous_data,new_data) values(New.Prosellerid,"ProSeller Updated",New.Prosellename,now(),Old.Prosellerlocation,New.Prosellerlocation);
    -> end &&
Query OK, 0 rows affected (0.47 sec)
```

```
mysql> delimiter ;
```

```
mysql> update PropertySeller set Prosellerlocation="Pimpri" where Prosellerid=1;
Query OK, 1 row affected (0.12 sec)
```

```
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from sellerauditlog;
```

sellerid	Action	seller_name	Action_time	previous_data	new_data
1	ProSeller Updated	Sakshi Mehta	2024-11-14 19:39:44	Vasant Kunj	Pimpri
13	ProSeller Added	Rutika Bhirwandekar	2024-11-14 19:09:15	NULL	NULL
14	ProSeller Added	Priyanka Tanpure	2024-11-14 19:10:33	NULL	NULL

```
3 rows in set (0.00 sec)
```

3. FOR DELETE:

Triggered when an existing record is deleted into the table.

Query:

NIKHIL VANAJI WALUNJ

```
mysql> delimiter &&
mysql> create trigger delete_seller after delete on PropertySeller
-> for each row
-> begin
-> insert into sellerauditlog(sellerid, Action, seller_name, Action_time) values(Old.Prosellerid, "ProSeller deleted", Old.Prosellername, now());
-> end &&
Query OK, 0 rows affected (0.37 sec)

mysql> delimiter ;
mysql> delete from PropertySeller where Prosellerid=11;
Query OK, 1 row affected (0.12 sec)

mysql> select * from sellerauditlog;
```

```
mysql> delimiter &&
mysql> create trigger delete_seller after delete on PropertySeller
-> for each row
-> begin
-> insert into sellerauditlog(sellerid, Action, seller_name, Action_time) values(Old.Prosellerid, "ProSeller deleted", Old.Prosellername, now());
-> end &&
Query OK, 0 rows affected (0.37 sec)

mysql> delimiter ;
mysql> delete from PropertySeller where Prosellerid=11;
Query OK, 1 row affected (0.12 sec)

mysql> select * from sellerauditlog;
+-----+-----+-----+-----+-----+
| sellerid | Action | seller_name | Action_time | previous_data | new_data |
+-----+-----+-----+-----+-----+
|      1 | ProSeller Updated | Sakshi Mehta | 2024-11-14 19:39:44 | Vasant Kunj | Pimpri
|     11 | ProSeller deleted | Nikhil Walunj | 2024-11-14 19:53:27 | NULL       | NULL
|    13 | ProSeller Added   | Rutika Bhirwandekar | 2024-11-14 19:09:15 | NULL       | NULL
|    14 | ProSeller Added   | Priyanka Tanpure | 2024-11-14 19:10:33 | NULL       | NULL
+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

CURSOR

A **Cursor** in MySQL is a database object that allows you to retrieve a result set row by row, rather than all at once.

Cursors are particularly useful in stored procedures, triggers, or scripts where you need to perform operations on each individual row returned by a query.

Basic Cursor Operations

Declare the Cursor: Define the cursor with a SQL query.

Open the Cursor: Initialize the cursor and populate it with the result set.

Fetch Data: Retrieve individual rows one by one from the cursor.

Close the Cursor: Release the memory associated with the cursor.

Query:

```
create procedure Propertiesdata()
```

```
begin
```

--Declare the Variables

```
declare done int default false;
```

```
declare P_Proid int;
```

```
declare P_Proname varchar(50);
```

```
declare P_Procity varchar(100);
```

```
declare P_Proprice decimal(15,2);
```

--Declare the Cursor

```
declare status_cursor cursor for
```

```
select Proid,Proname,Procity,Proprice from Property
```

```
where Prostatus="Available";
```

-- Declare CONTINUE HANDLER to handle when the cursor has finished fetching rows

```
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
```

--Open the Cursor

```
open status_cursor;
```

--Fetch the Data

```
status_loop:LOOP
```

```
fetch status_cursor into P_Proid,P_Proname,P_Procity,P_Proprice;
```

```
if done=TRUE then
```

```
leave status_loop;
```

```
end if;
```

```
update Property set Prostatus="Sold" where Proid=P_Proid;
```

```
end loop;
```

--Close the Cursor

```
close status_cursor;
```

```
end &&
```

```
mysql> delimiter &&
mysql> create procedure Propertiesdata()
-> begin
-> declare done int default false;
-> declare P_Proid int;
-> declare P_Proname varchar(50);
-> declare P_Procity varchar(100);
-> declare P_Proprice decimal(15,2);
-> declare status_cursor cursor for
-> select Proid,Proname,Procity,Proprice from Property
-> where Prostatus="Available";
-> DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = TRUE;
-> open status_cursor;
-> status_loop:LOOP
-> fetch status_cursor into P_Proid,P_Proname,P_Procity,P_Proprice;
-> if done=TRUE then
-> leave status_loop;
-> end if;
-> update Property set Prostatus="Sold" where Proid=P_Proid;
-> end loop;
-> close status_cursor;
-> end &&
```

Query OK, 0 rows affected (0.12 sec)

```
mysql> delimiter ;
mysql> call Propertiesdata();
Query OK, 0 rows affected (0.18 sec)
```

```
mysql> select Proid,Proname,Procity,Proprice,Prostatus from Property;
```

Proid	Proname	Procity	Proprice	Prostatus
1	Luxury 2BHK Apartment	Mumbai	7500000.00	Sold
2	Green View Independent House	Pune	12000000.00	Sold
3	Commercial Office Space	Delhi	25000000.00	Sold
4	Residential Plot	Bangalore	5000000.00	Sold
5	PG Room for Rent	Chennai	15000.00	Sold
6	Oceanview 3BHK	Goa	9500000.00	Sold
7	Villa in Green Valley	Ahmedabad	18000000.00	Sold
8	Tech Hub Office	Hyderabad	15000000.00	Sold
9	Luxury Plot in Prime Area	Kolkata	8000000.00	Sold
10	PG Room Near IT Park	Bangalore	12000.00	Sold
11	Modern 2BHK Apartment	Mumbai	5000000.00	Sold
12	Independent House with Garden	Pune	8500000.00	Sold
13	Commercial Space for Retail	Delhi	12000000.00	Sold
14	Residential Plot for Sale	Bangalore	6500000.00	Sold
15	PG Room with AC	Chennai	18000.00	Sold
16	Luxury 1BHK Apartment	Delhi	3500000.00	Sold
17	Detached Independent Home	Ahmedabad	15000000.00	Sold
18	Co-Working Office Space	Hyderabad	8500000.00	Sold
19	Prime Residential Plot	Kolkata	4000000.00	Sold
20	PG Room in Safe Area	Pune	10000.00	Sold
21	Indian Independent House	New Delhi	12000000.00	Sold
22	Indian Co-Working Space	Mumbai	5000000.00	Sold

```
22 rows in set (0.00 sec)
```

PRAMEEVA