**PET SHOP (Management System)**

**OBJECTIVE OF THE PROJECT**

* The objectives of the project is to provide web based interface to a **petshop** owner to manages his **petshop** activities.
* To provide an option for storing and managing the basic information about pets and pet products in the shop.
* To provide an option for storing and managing the sales details of the shop.
* To provide an option for storing and managing the basic information about the customer.
* To track the information about sold pets and products to a customer

**IMPLEMENTATION**

**ER DIAGRAM**

An entity-relationship model (ER model) describes inter-related things of

Interest in a specific domain of knowledge. An ER model is composed of entity types(which classify the things of interest) and specifies relationships that can exist between instances of those entity types.

ER model is commonly formed to represent things that a business needs to

remember in order to perform business processes. Consequently, the ER model becomes an abstract data model that defines a data or information structure that can be implemented in a database, typically a relational database.

The main components of ER model are: entity set and relationship set.

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.



**ER DIAGRAM FOR PET SHOP MANAGEMENT SYSTEM**

**NORMALIZE THE RELATIONS**

Normalization is a process of organizing the data in database to avoid data redundancy,

insertion anomaly, update anomaly & deletion anomaly.

There are three main types of normal forms:

a) First normal form(1NF)

b) Second normal form(2NF)

c) Third normal form(3NF)

**1. First normal form (1NF):**

a) As per the rule of first normal form, an attribute (column) of a table cannot hold multiple

values.

b) It should hold only atomic values.

This table holds only the atomic values company id and the company name and no multiple

values are stored in this table so it can be considered as the 1NF.

**2.Second normal form (2NF):**

A table is said to be in 2NF if both the following conditions hold:

a) Table is in 1NF (First normal form)

b) No non-prime attribute is dependent on the proper subset of any candidate key of table.

c) An attribute that is not part of any candidate key is known as non-prime attribute.

**3. Third Normal form (3NF):**

A table design is said to be in 3NF if both the following conditions hold:

a) Table must be in 2NF

b) Transitive functional dependency of non-prime attribute on any super key should be

removed.

c) An attribute that is not part of any candidate key is known as non-prime attribute.

In other words 3NF can be explained like this: A table is in 3NF if it is in 2NF and for each

functional dependency X-> Y at least one of the following conditions hold:

X is a super key of table

Y is a prime attribute of table

An attribute that is a part of one of the candidate keys is known as prime attribute.

The relations are already in the normalized form in the schema diagram without any redundancy.

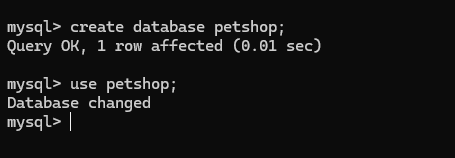
**Database and Tables Creation**

* **Table for storing different types of PETS**

1. **Create Database:**

create database petshop;

use petshop;



1. **Create Tables And Insert Values:**

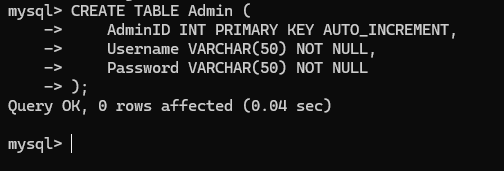
**i. Create Table Admin :**

CREATE TABLE Admin (

AdminID INT PRIMARY KEY AUTO\_INCREMENT,

Username VARCHAR(50) NOT NULL,

Password VARCHAR(50) NOT NULL );

****

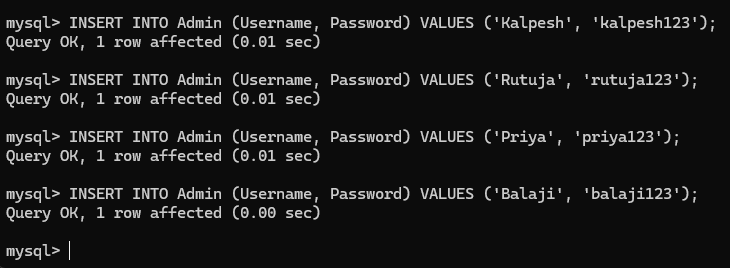
**ii. Insert Admin Data:**

INSERT INTO Admin (Username, Password) VALUES ('Kalpesh', 'kalpesh123');

INSERT INTO Admin (Username, Password) VALUES ('Rutuja', 'rutuja123');

INSERT INTO Admin (Username, Password) VALUES ('Priya', 'priya123');

INSERT INTO Admin (Username, Password) VALUES ('Balaji', 'balaji123');



* **Table for storing CUSTOMER information**

**i. Create Table For Customers :**

CREATE TABLE Customer (

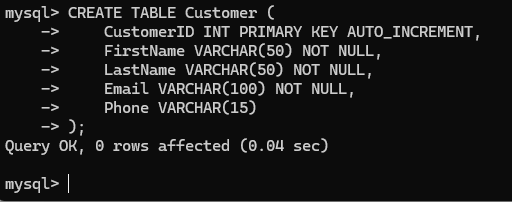
CustomerID INT PRIMARY KEY AUTO\_INCREMENT,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Email VARCHAR(100) NOT NULL,

Phone VARCHAR(15) );

****

**i. Insert Customer Data :**

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Amit', 'Sharma', 'amit.sharma@example.com', '9123456780');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Priya', 'Verma', 'priya.verma@example.com', '9123456781');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Rahul', 'Gupta', 'rahul.gupta@example.com', '9123456782');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Anita', 'Rao', 'anita.rao@example.com', '9123456783');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Vikram', 'Patel', 'vikram.patel@example.com', '9123456784');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Sneha', 'Singh', 'sneha.singh@example.com', '9123456785');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Karan', 'Joshi', 'karan.joshi@example.com', '9123456786');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Neha', 'Mehta', 'neha.mehta@example.com', '9123456787');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Ravi', 'Kumar', 'ravi.kumar@example.com', '9123456788');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Pooja', 'Yadav', 'pooja.yadav@example.com', '9123456789');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Arjun', 'Shukla', 'arjun.shukla@example.com', '9123456790');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Lakshmi', 'Iyer', 'lakshmi.iyer@example.com', '9123456791');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Rohan', 'Nair', 'rohan.nair@example.com', '9123456792');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Anjali', 'Desai', 'anjali.desai@example.com', '9123456793');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Suresh', 'Reddy', 'suresh.reddy@example.com', '9123456794');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Asha', 'Ghosh', 'asha.ghosh@example.com', '9123456795');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Raj', 'Singh', 'raj.singh@example.com', '9123456796');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Meera', 'Chopra', 'meera.chopra@example.com', '9123456797');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Nitin', 'Agarwal', 'nitin.agarwal@example.com', '9123456798');

INSERT INTO Customer (FirstName, LastName, Email, Phone) VALUES ('Divya', 'Khan', 'divya.khan@example.com', '9123456799');

****

* **Table for storing PET**

**i. Create Table For Pet :**

CREATE TABLE Pet ( PetID INT PRIMARY KEY AUTO\_INCREMENT,

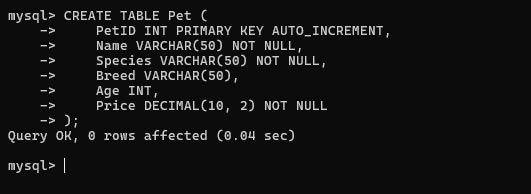
Name VARCHAR(50) NOT NULL,

Species VARCHAR(50) NOT NULL,

Breed VARCHAR(50),

Age INT,

Price DECIMAL(10, 2) NOT NULL );



**i. Insert Pet Data:**

**-- Dogs**

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Sheru', 'Dog', 'Indian Pariah', 3, 200.00);

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Tommy', 'Dog', 'Rajapalayam', 2, 500.00);

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Moti', 'Dog', 'Mudhol Hound', 4, 600.00);

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Raja', 'Dog', 'Kanni', 3, 550.00);

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Bruno', 'Dog', 'Chippiparai', 2, 450.00);

**-- Cats**

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Misty', 'Cat', 'Indian Shorthair', 2, 150.00);

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Simba', 'Cat', 'Indian Billi', 3, 200.00);

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Lucy', 'Cat', 'Bengal', 2, 300.00);

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Leo', 'Cat', 'Persian', 4, 700.00);

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Snowy', 'Cat', 'Himalayan', 3, 500.00);

**-- Birds**

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Mithu', 'Bird', 'Budgerigar', 1, 50.00);

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Chirpy', 'Bird', 'Indian Ringneck Parakeet', 2, 100.00);

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Sunny', 'Bird', 'Sun Conure', 1, 150.00);

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Kiki', 'Bird', 'Cockatiel', 2, 120.00);

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Peppy', 'Bird', 'Lovebird', 1, 80.00);

**-- Additional Dogs**

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Rocky', 'Dog', 'Kombai', 3, 400.00);

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Tiger', 'Dog', 'Gaddi Kutta', 4, 450.00);

**-- Additional Cats**

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Bella', 'Cat', 'Exotic Shorthair', 2, 600.00);

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Coco', 'Cat', 'Sphynx', 3, 800.00);

**-- Additional Birds**

INSERT INTO Pet (Name, Species, Breed, Age, Price) VALUES ('Raja', 'Bird', 'Macaw', 1, 500.00);

****

* **Table for storing PRODUCT**

**i. Create Table For Product :**

CREATE TABLE Product (

ProductID INT PRIMARY KEY AUTO\_INCREMENT,

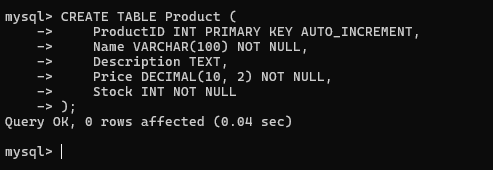
Name VARCHAR(100) NOT NULL,

Description TEXT,

Price DECIMAL(10, 2) NOT NULL,

Stock INT NOT NULL

);



**ii. Insert Product Data :**

**-- Dog Products**

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Royal Canin Dog Food', 'High-quality dog food for all breeds', 800.00, 50);

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Pedigree Dog Food', 'Nutritious dog food for healthy growth', 600.00, 80);

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Dog Collar', 'Adjustable collar for dogs', 250.00, 100);

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Dog Leash', 'Durable leash for walks', 300.00, 75);

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Dog Treats', 'Tasty treats for training', 150.00, 120);

**-- Cat Products**

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Whiskas Cat Food', 'Delicious cat food for all ages', 500.00, 60);

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Meow Mix Cat Food', 'Nutritious food for healthy cats', 450.00, 70);

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Cat Litter', 'Clumping cat litter for easy cleaning', 300.00, 50);

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Cat Toy', 'Interactive toys for playful cats', 200.00, 100);

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Cat Scratching Post', 'Essential for cat grooming', 700.00, 30);

**-- Bird Products**

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Budgie Food', 'Specialized food for budgerigars', 200.00, 90);

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Parakeet Cage', 'Spacious cage for parakeets', 1500.00, 20);

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Bird Toys', 'Fun toys for birds', 100.00, 150);

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Bird Bath', 'A refreshing bath for birds', 300.00, 40);

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Cockatiel Food', 'Nutritional food for cockatiels', 250.00, 60);

**-- Additional Dog Products**

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Dog Bed', 'Comfortable bed for dogs', 1200.00, 25);

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Flea and Tick Treatment', 'Effective treatment for fleas and ticks', 400.00, 50);

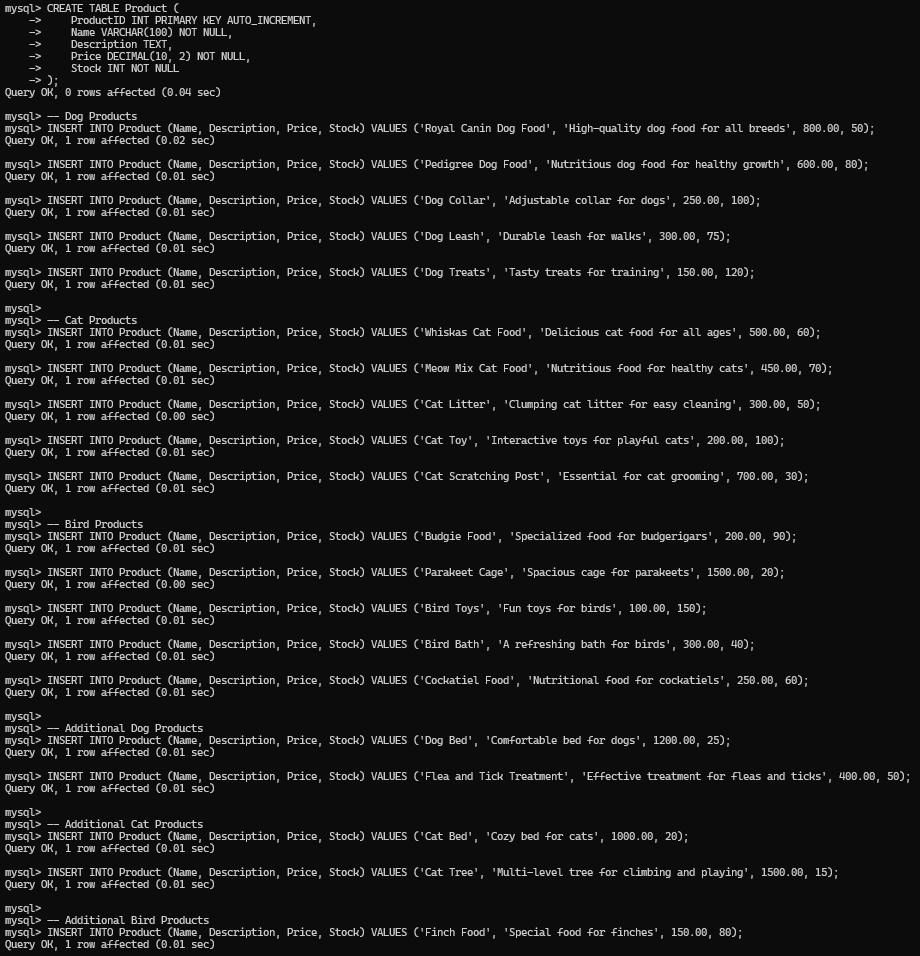
**-- Additional Cat Products**

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Cat Bed', 'Cozy bed for cats', 1000.00, 20);

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Cat Tree', 'Multi-level tree for climbing and playing', 1500.00, 15);

**-- Additional Bird Products**

INSERT INTO Product (Name, Description, Price, Stock) VALUES ('Finch Food', 'Special food for finches', 150.00, 80);

****

* **Table for storing SALES**

**i. Create Table For Sales :**

CREATE TABLE Sales (

SalesID INT PRIMARY KEY AUTO\_INCREMENT,

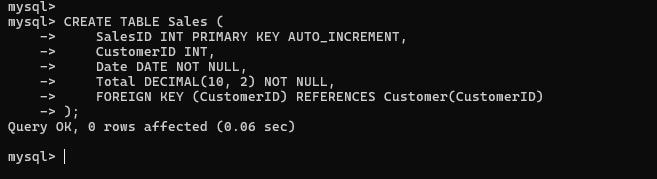
CustomerID INT,

Date DATE NOT NULL,

Total DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);



**ii. Insert Sales Data :**

INSERT INTO Sales (CustomerID, Date, Total) VALUES (1, '2024-08-01', 850.00);

INSERT INTO Sales (CustomerID, Date, Total) VALUES (2, '2024-08-02', 1200.00);

INSERT INTO Sales (CustomerID, Date, Total) VALUES (3, '2024-08-03', 450.00);

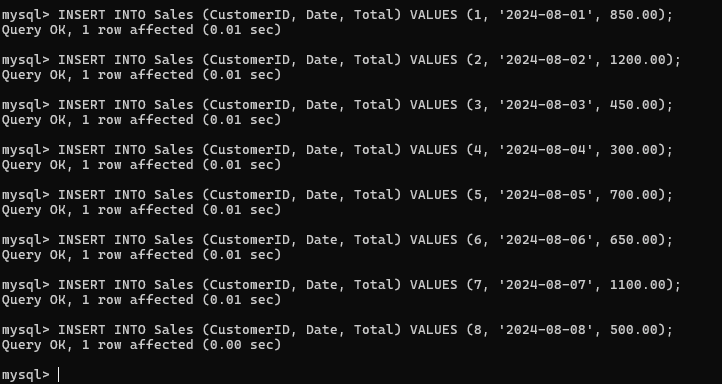
INSERT INTO Sales (CustomerID, Date, Total) VALUES (4, '2024-08-04', 300.00);

INSERT INTO Sales (CustomerID, Date, Total) VALUES (5, '2024-08-05', 700.00);

INSERT INTO Sales (CustomerID, Date, Total) VALUES (6, '2024-08-06', 650.00);

INSERT INTO Sales (CustomerID, Date, Total) VALUES (7, '2024-08-07', 1100.00);

INSERT INTO Sales (CustomerID, Date, Total) VALUES (8, '2024-08-08', 500.00);



* **Table for storing SALES**

**i. Create Table For Sales Details :**

CREATE TABLE SaleDetails (

SaleDetailsID INT PRIMARY KEY AUTO\_INCREMENT,

SalesID INT,

ProductID INT,

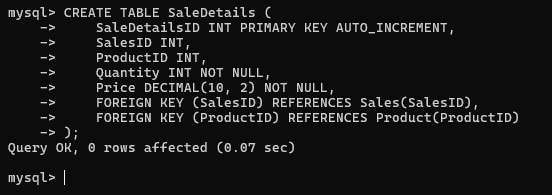
Quantity INT NOT NULL,

Price DECIMAL(10, 2) NOT NULL,

FOREIGN KEY (SalesID) REFERENCES Sales(SalesID),

FOREIGN KEY (ProductID) REFERENCES Product(ProductID)

);



**ii. Insert Sales Details Data :**

INSERT INTO SaleDetails (SalesID, ProductID, Quantity, Price) VALUES (1, 1, 2, 800.00);

INSERT INTO SaleDetails (SalesID, ProductID, Quantity, Price) VALUES (2, 2, 3, 600.00);

INSERT INTO SaleDetails (SalesID, ProductID, Quantity, Price) VALUES (3, 3, 1, 250.00);

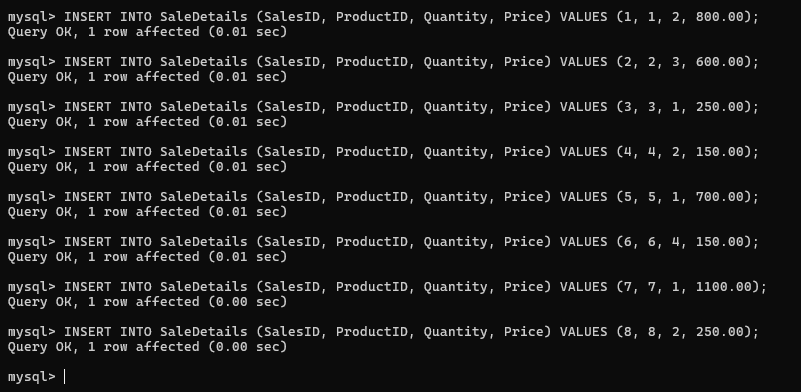
INSERT INTO SaleDetails (SalesID, ProductID, Quantity, Price) VALUES (4, 4, 2, 150.00);

INSERT INTO SaleDetails (SalesID, ProductID, Quantity, Price) VALUES (5, 5, 1, 700.00);

INSERT INTO SaleDetails (SalesID, ProductID, Quantity, Price) VALUES (6, 6, 4, 150.00);

INSERT INTO SaleDetails (SalesID, ProductID, Quantity, Price) VALUES (7, 7, 1, 1100.00);

INSERT INTO SaleDetails (SalesID, ProductID, Quantity, Price) VALUES (8, 8, 2, 250.00);



* **Table for storing PAYMENT**

**i. Create Table For Payment :**

CREATE TABLE Payment (

PaymentID INT PRIMARY KEY AUTO\_INCREMENT,

SalesID INT,

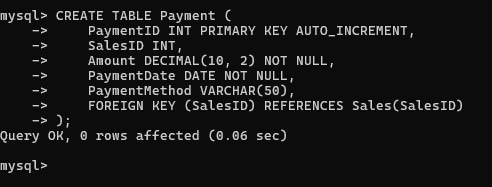
Amount DECIMAL(10, 2) NOT NULL,

PaymentDate DATE NOT NULL,

PaymentMethod VARCHAR(50),

FOREIGN KEY (SalesID) REFERENCES Sales(SalesID)

);



**ii. Insert Payment Data :**

INSERT INTO Payment (SalesID, Amount, PaymentDate, PaymentMethod) VALUES (1, 850.00, '2024-08-01', 'Credit Card');

INSERT INTO Payment (SalesID, Amount, PaymentDate, PaymentMethod) VALUES (2, 1200.00, '2024-08-02', 'Debit Card');

INSERT INTO Payment (SalesID, Amount, PaymentDate, PaymentMethod) VALUES (3, 450.00, '2024-08-03', 'Cash');

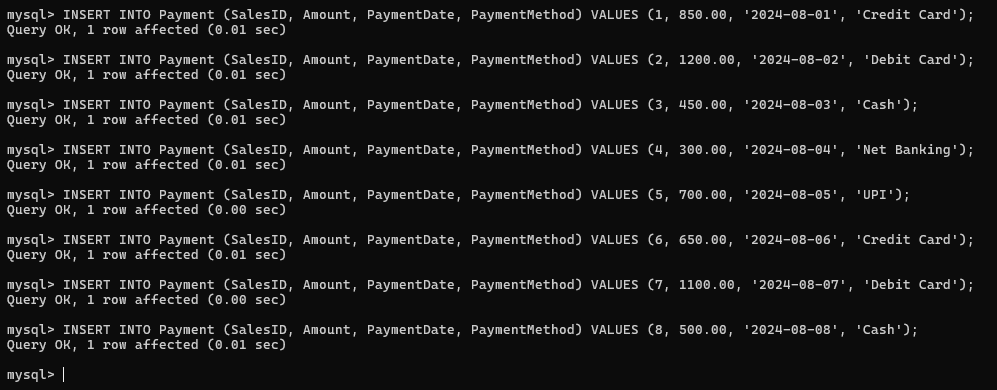
INSERT INTO Payment (SalesID, Amount, PaymentDate, PaymentMethod) VALUES (4, 300.00, '2024-08-04', 'Net Banking');

INSERT INTO Payment (SalesID, Amount, PaymentDate, PaymentMethod) VALUES (5, 700.00, '2024-08-05', 'UPI');

INSERT INTO Payment (SalesID, Amount, PaymentDate, PaymentMethod) VALUES (6, 650.00, '2024-08-06', 'Credit Card');

INSERT INTO Payment (SalesID, Amount, PaymentDate, PaymentMethod) VALUES (7, 1100.00, '2024-08-07', 'Debit Card');

INSERT INTO Payment (SalesID, Amount, PaymentDate, PaymentMethod) VALUES (8, 500.00, '2024-08-08', 'Cash');



* **Table for storing ORDERS**

**i. Create Table For Orders :**

CREATE TABLE Orders (

OrderID INT PRIMARY KEY AUTO\_INCREMENT,

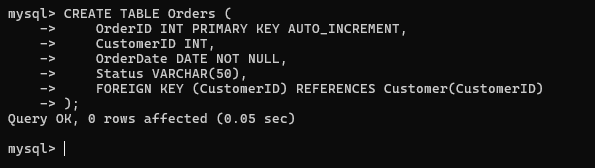
CustomerID INT,

OrderDate DATE NOT NULL,

Status VARCHAR(50),

FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)

);



**ii. Insert Orders Data :**

INSERT INTO Orders (CustomerID, OrderDate, Status) VALUES (2, '2024-08-01', 'Shipped');

INSERT INTO Orders (CustomerID, OrderDate, Status) VALUES (3, '2024-08-02', 'Processing');

INSERT INTO Orders (CustomerID, OrderDate, Status) VALUES (4, '2024-08-03', 'Delivered');

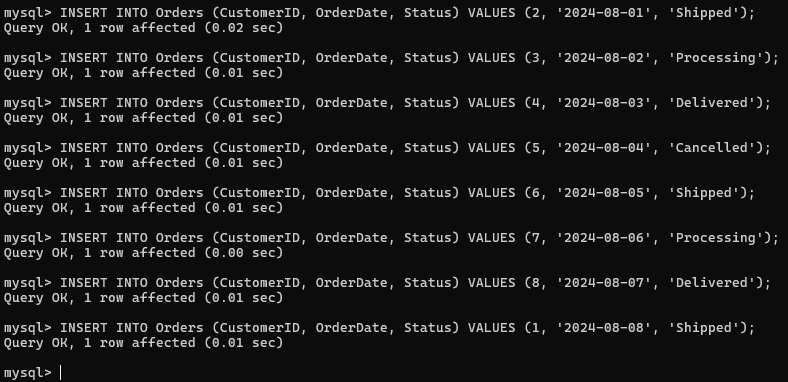
INSERT INTO Orders (CustomerID, OrderDate, Status) VALUES (5, '2024-08-04', 'Cancelled');

INSERT INTO Orders (CustomerID, OrderDate, Status) VALUES (6, '2024-08-05', 'Shipped');

INSERT INTO Orders (CustomerID, OrderDate, Status) VALUES (7, '2024-08-06', 'Processing');

INSERT INTO Orders (CustomerID, OrderDate, Status) VALUES (8, '2024-08-07', 'Delivered');

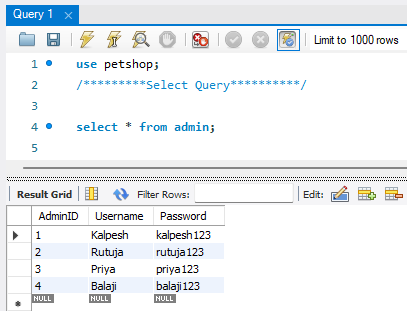
INSERT INTO Orders (CustomerID, OrderDate, Status) VALUES (1, '2024-08-08', 'Shipped');



**Select Query :**

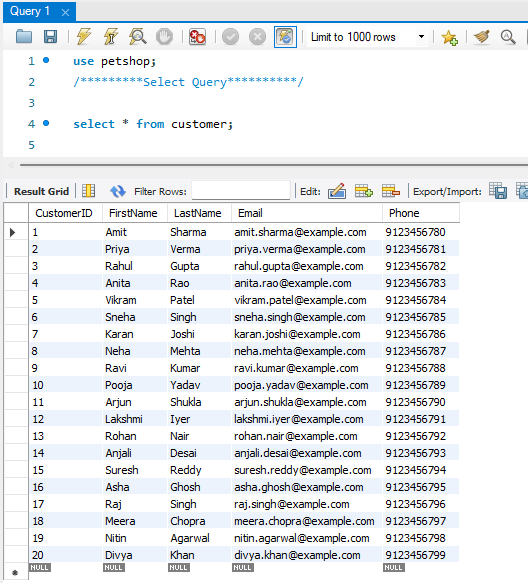
**Select Query For Admin :**

select \* from admin;



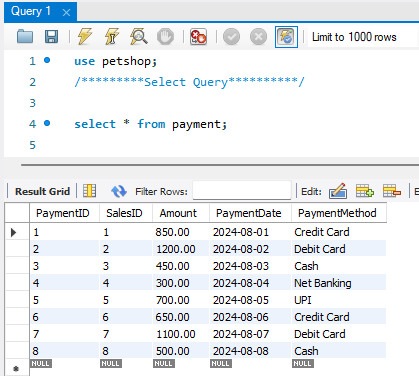
**Select Query For Customer :**

select \* from customer;



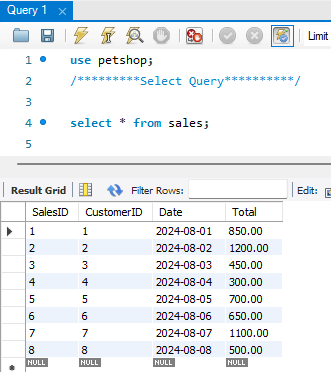
**Select Query For Payment :**

select \* from payment;



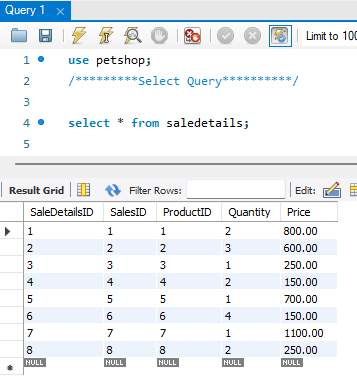
**Select Query For Sales :**

select \* from sales;

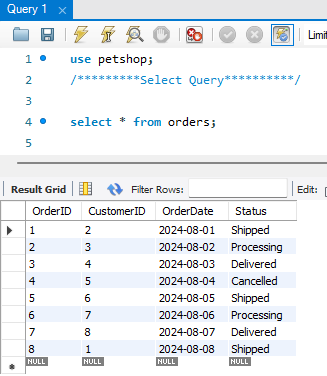


**Select Query For Sale Details :**

select \* from salesdetails;

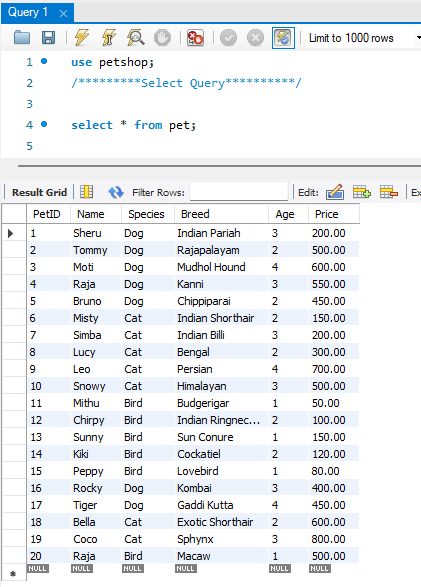


**Select Query For Orders :**

****select \* from orders;

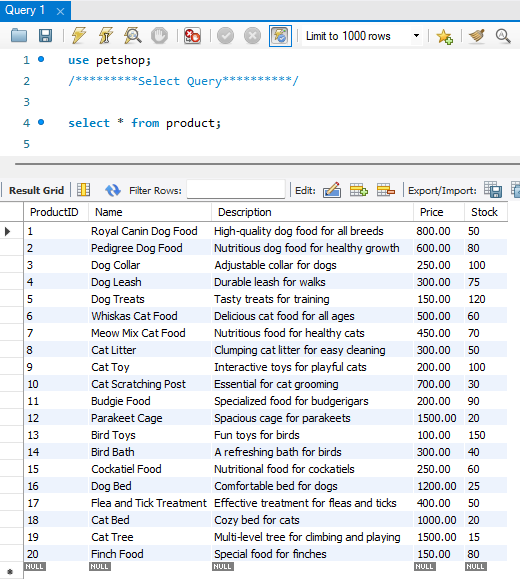
**Select Query For Pet :**

select \* from pet;

****

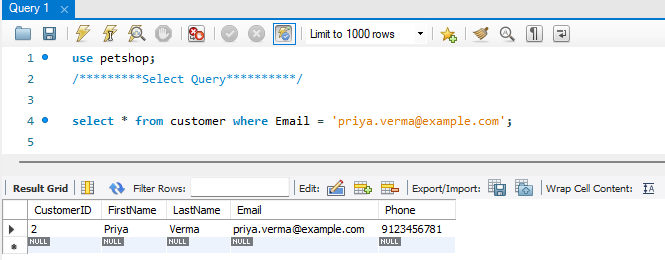
**Select Query For Product :**

select \* from product;

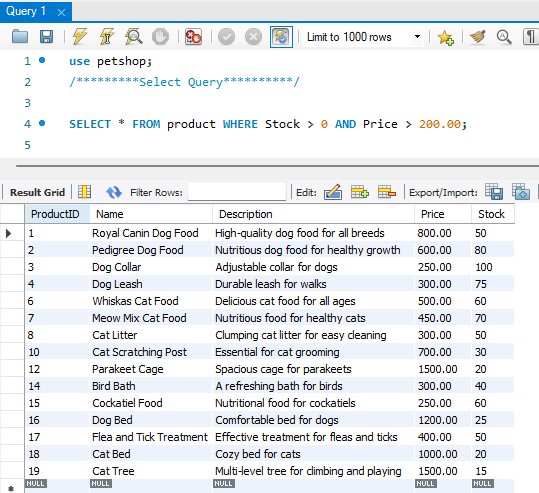
****

* **WHERE Clause :**

1) select \* from customer where Email = 'priya.verma@example.com';

****

2) SELECT \* FROM product WHERE Stock > 0 AND Price > 200.00;

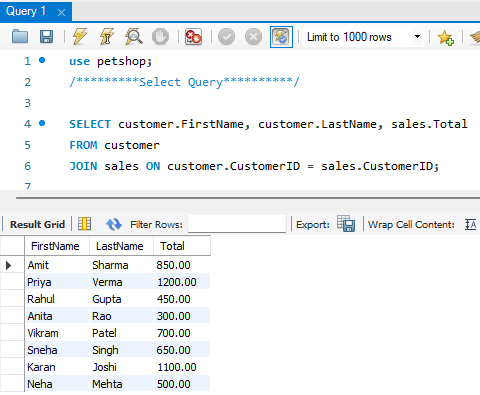
****

* **JOIN Query :**

SELECT customer.FirstName, customer.LastName, sales.Total

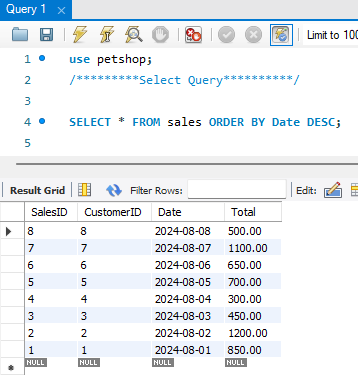
FROM customer

JOIN sales ON customer.CustomerID = sales.CustomerID;



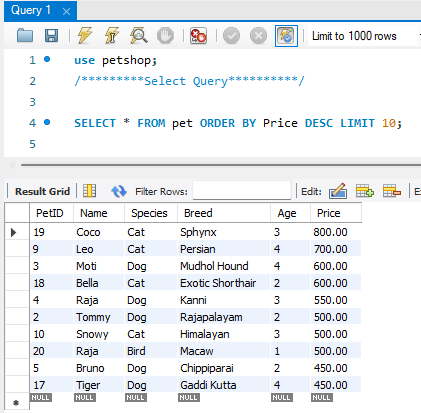
* **ORDER BY Clause :**

SELECT \* FROM sales ORDER BY Date DESC;



* **LIMIT Clause :**

SELECT \* FROM pet ORDER BY Price DESC LIMIT 10;



* **GROUP BY Clause :**

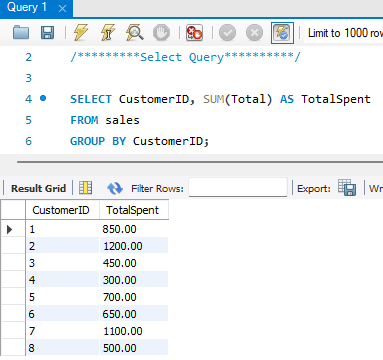
**Example 1: Group By Customer's Total Spend :**

**Let's find out how much each customer has spent in total:**

SELECT CustomerID, SUM(Total) AS TotalSpent

FROM sales

GROUP BY CustomerID;



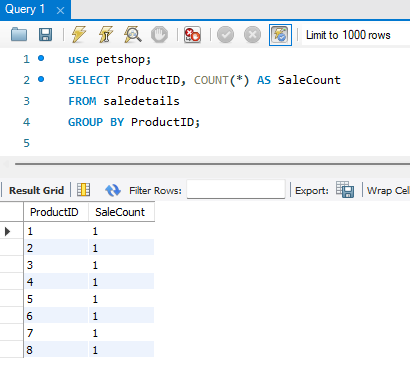
**Example 2: Group By Product Sales Count :**

**Let's see how many times each product has been sold:**

SELECT ProductID, COUNT(\*) AS SaleCount

FROM saledetails

GROUP BY ProductID;



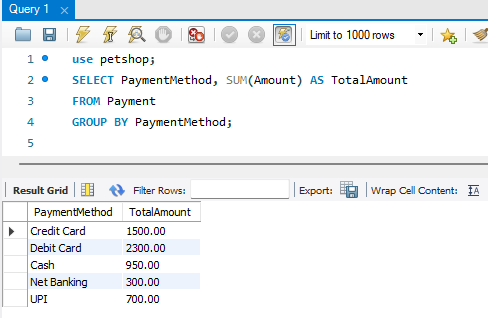
**Example 3: Group By Payment Method :**

**Let's see the total amount paid using each payment method:**

SELECT PaymentMethod, SUM(Amount) AS TotalAmount

FROM Payment

GROUP BY PaymentMethod;



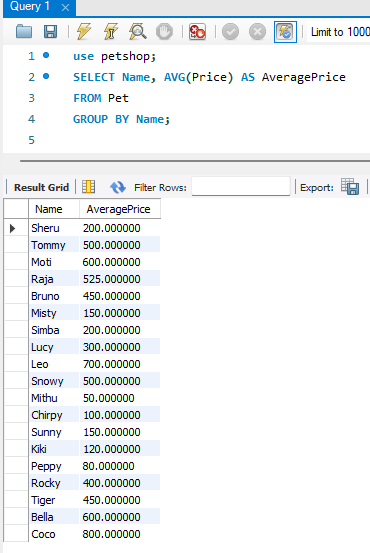
**Example 4: Group By Pet Name :**

**Let's see the average price of pets by Name:**

SELECT Name, AVG(Price) AS AveragePrice

FROM Pet

GROUP BY Name;



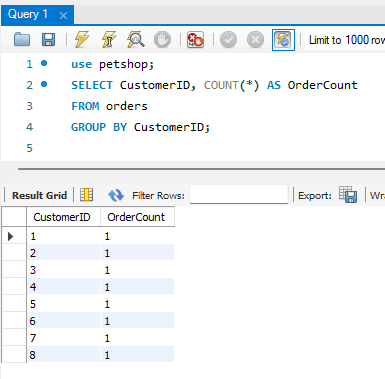
### Example 5: Group By Customer's Order Count :

**Let's count the number of orders each customer has placed:**

SELECT CustomerID, COUNT(\*) AS OrderCount

FROM orders

GROUP BY CustomerID;



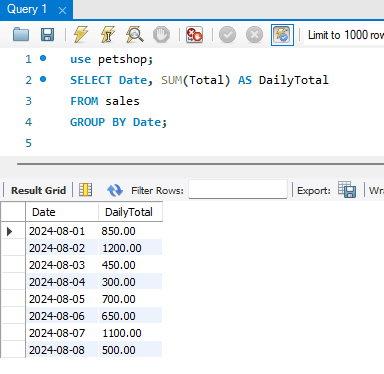
### Example 6: Group By Sales Date :

**Let's see the total sales amount for each date:**

SELECT Date, SUM(Total) AS DailyTotal

FROM sales

GROUP BY Date;



### Example 7: Group By Customer's Name

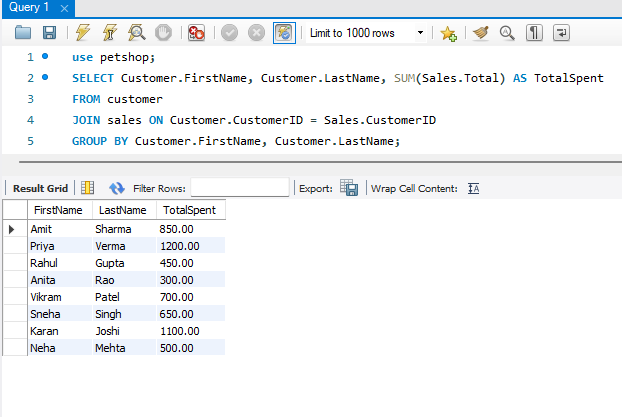
**Let's find the total amount each customer has spent, grouped by their name (assuming names are unique for simplicity):**

SELECT Customer.FirstName, Customer.LastName, SUM(Sales.TotalAmount) AS TotalSpent

FROM Customer

JOIN Sales ON Customer.CustomerID = Sales.CustomerID

GROUP BY Customer.FirstName, Customer.LastName;

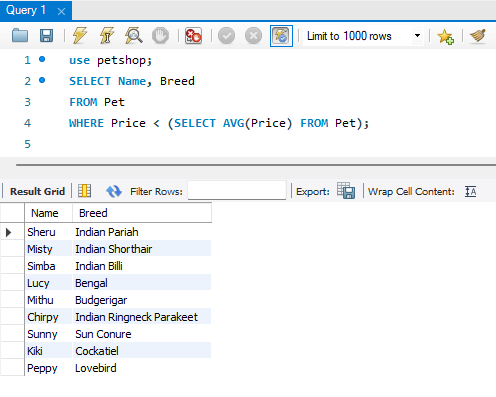


* **SUB Query :**

SELECT Name, Breed

FROM Pet

WHERE Price < (SELECT AVG(Price) FROM Pet);



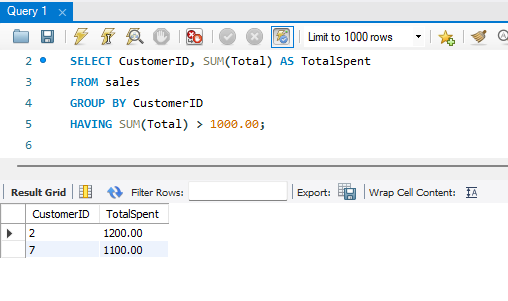
* **HAVING Clause :**

SELECT CustomerID, SUM(Total) AS TotalSpent

FROM sales

GROUP BY CustomerID

HAVING SUM(Total) > 1000.00;



* **DELIMITER Query :**

DELIMITER //

CREATE PROCEDURE GetHighValueCustomers()

BEGIN

SELECT CustomerID, SUM(Total) AS TotalSpent

FROM sales

GROUP BY CustomerID

HAVING SUM(Total) > 1000.00;

END //

DELIMITER ;

CALL GetHighValueCustomers();

