

In this project, I practiced key concepts of SQL by designing, creating, and analyzing multiple related tables. The objective was to understand how to manage structured data, perform filtering and sorting, and implement joins between tables to extract meaningful insights.

## Step 1: Creating Tables and Inserting Data

### 1. Customers Table:

I started by creating a Customers table to practice basic SQL operations. I inserted customer records and used this table to apply filtering and grouping logic.

### 2. Pets Table:

I created a Pets table containing details such as pet\_id, name, kind, gender, age, and owner\_id. This table represents individual pets and links each to their respective owner through owner\_id.

### 3. Owners Table:

The Owners table includes information about pet owners such as owner\_id, name, gender, occupation, age, city, and state.

### 4. Vet Table:

I added a Vet table that includes veterinarian details like vet\_id, name, clinic\_name, pet\_id, and disease. This table maps pets to their respective health issues and doctors.

## Step 2: Applying Filtering & Grouping

I used the **WHERE** clause to filter specific rows based on conditions (e.g., age > 20, gender = 'male').

I used the **GROUP BY** clause to group data (e.g., counting pets by kind or grouping customers by city).

I used the **HAVING** clause to filter the grouped results (e.g., cities having more than 2 customers).

### Step 3: Sorting Data

I used the **\*\*ORDER BY\*\*** clause to sort records alphabetically or numerically (e.g., sorting pets by age or owners by name).

This helped in presenting data in a structured and understandable way.

### Step 4: Performing SQL Joins

To combine data across tables and derive deeper insights, I performed:

**\*\*INNER JOIN\*\*** between Pets and Owners on owner\_id to get pet and owner information together.

**\*\*INNER JOIN\*\*** between Pets and Vet on pet\_id to analyze which pet was treated by which vet and for what disease.

Multi-table joins to connect Pets, Owners, and Vet in a single query — extracting pet name, owner name, disease, and vet details in one result.

These joins simulated a real-world relational database structure, which is commonly used in applications like pet clinics, veterinary hospitals, and pet adoption systems.

### Conclusion:

This task helped me gain hands-on experience with:

Table creation and data normalization

Data filtering using WHERE and HAVING

Data sorting using ORDER BY

Relational joins using INNER JOIN

Through this, I understood how to build and interact with complex datasets, making me more confident in using SQL for real-life data analysis and database management tasks.



