**ACTIVITY 8.1:** **SQL Joins**

**Objectives:**

1. To know the underlying concepts of the SQL join statements
2. To know how to write and invoke SQL join statements

**Materials:**

PC or Laptop

WAMP/XAMPP Installer

Web Browser or CLI

**Background**

A **JOIN** clause is used to combine rows from two or more tables, based on a related column between them.

**Here are the different types of the JOINs in SQL:**

* **(INNER) JOIN**: Returns records that have matching values in both tables
* **LEFT (OUTER) JOIN**: Return all records from the left table, and the matched records from the right table
* **RIGHT (OUTER) JOIN**: Return all records from the right table, and the matched records from the left table



Graphical user interface, text, application

Description automatically generated

Looking at our **ERD** above, we can say that all tables enforced referential integrity. This is one great advantage of a relational database. However, if we try to invoke SQL command using only **SELECT** statement or some simple queries, we might produce undesired reports or not meaningful and readable summarized data. Look at the display output below:

Graphical user interface, text

Description automatically generated

Yes, you may have shown information of pet, but if you try to look at it specifically ***pet\_gender, pet\_breed, status\_id*** may seem to have less meaning as we know. This is because of the relationship you have established. A data of a table is related to any other data of the other tables in a single database. Look at the display below:

Graphical user interface

Description automatically generated with medium confidence

You might think it is magic. Personally, I can say it is. Everything we do in computer is a magic. Because tech guy is considered as a wizard.

And everything about computer is mathematically inclined. So, it is mathematically magic. Anyhow, you see there above that you have meaningful and readable information compare to the previous summarized data. That is the power of join statements. This is most underrated SQL statements by the students because it can be complex or too hard to understand. For me it just takes practice to master this concept. So, let us get started.

## SQL LEFT JOIN Keyword

The LEFT JOIN keyword returns all records from the left table (table1), and the matched records from the right table (table2). The result is NULL from the right side, if there is no match.

### LEFT JOIN Syntax

SELECT column\_name(s)  
FROM table1  
LEFT JOIN table2ON table1.column\_name=table2.column\_name;

**Note:** In some databases LEFT JOIN is called LEFT OUTER JOIN.



**Example:**

Text

Description automatically generated with medium confidenceLet us look at the two tables of your database pet; **tbl\_type** and **tbl\_breed**. Of course, you see the relation between the two tables since *every pet has breed, and every single breed belongs to a type of pet*. Examine the two tables below:

Text

Description automatically generated

READ: Note that there are breed values which were not assigned to any type. To name them, we frog, cockroach, and pickachu. Probably, as of the moment they were set as null as if it was not yet decided which type they should belong. And yes, there instances we do that to our database.

Using LEFT join, let’s say how our summarized data should look like now:

Text

Description automatically generated with medium confidence

Red ones are original fields of **tbl\_breed**. **‘b’** means an alias/shortcut for **tbl\_breed**. You can have any character to represent. This is used to avoid ambiguity of the statement. Like from which table and what column you want to use and show. There are instances that two tables may have the same field name like if you have tables pet and owner which can both contains **‘name’** as a field. So, to explicitly call them you can use **p.name** for pet and **o.name** for owner.

Blue ones are fields related to **tbl\_type**. It means to say you are combining two different tables and use some fields of the other tables to obtain desired reports.

In neon color you will see results with ‘null’ values. This is what you normally obtain when you use **LEFT** join to two or more different tables. In our example we joined **tbl\_breed (left)** and **tbl\_type(right)**. Practically, it means that it will display all values from the left table including records that does not match to right table. Observe that **frog, cockroach, and pikachu** were NOT assigned yet to any type. But you can improve your reports using the statement below:

Text

Description automatically generated with medium confidence

That’s it for LEFT join.

**SQL RIGHT JOIN** Keyword

The **RIGHT JOIN** keyword returns all records from the right table (table2), and the matched records from the left table (table1). The result is NULL from the left side, when there is no match.

**RIGHT JOIN Syntax**

SELECT column\_name(s)  
FROM table1  
RIGHT JOIN table2ON table1.column\_name=table2.column\_name;

**Note:** In some databases RIGHT JOIN is called RIGHT OUTER JOIN.



Example:

You will observe in this join concept as an opposite of the previous join. All fields from the right table must be shown even if there was no matched from the table on the **LEFT**.

Text

Description automatically generated

And it can be improved using the statement below:

Text

Description automatically generated with medium confidence

Now, you can use the same statement by changing the ***‘left join’ or ‘right join’ to ‘inner join’ or just ‘join’***. Observe the display below:

Text

Description automatically generated with low confidence

Text

Description automatically generated

NOTE: Personally, I can say that most of the time the most useful join statement is **‘inner join’**.

Inner join with where clause

Text

Description automatically generated

Text

Description automatically generated

Sometimes when you realize that there is too much field you want to include. If not all but almost. And then suddenly you decide to include everything. You don’t need to specify each one of them. Use the SQL statement below instead:

Text

Description automatically generated

Inner join using **‘Using’** clause

Text

Description automatically generated

NOTE that in using **‘Using’** clause in join, you must make sure that two tables must have same field name that appears. For instance, since we use same **type\_id** to both tables **namely tbl\_breed and tbl\_type** hence there is nothing wrong about our statement above. But if you try to invoke statement below, you will see some errors. Below is the example:

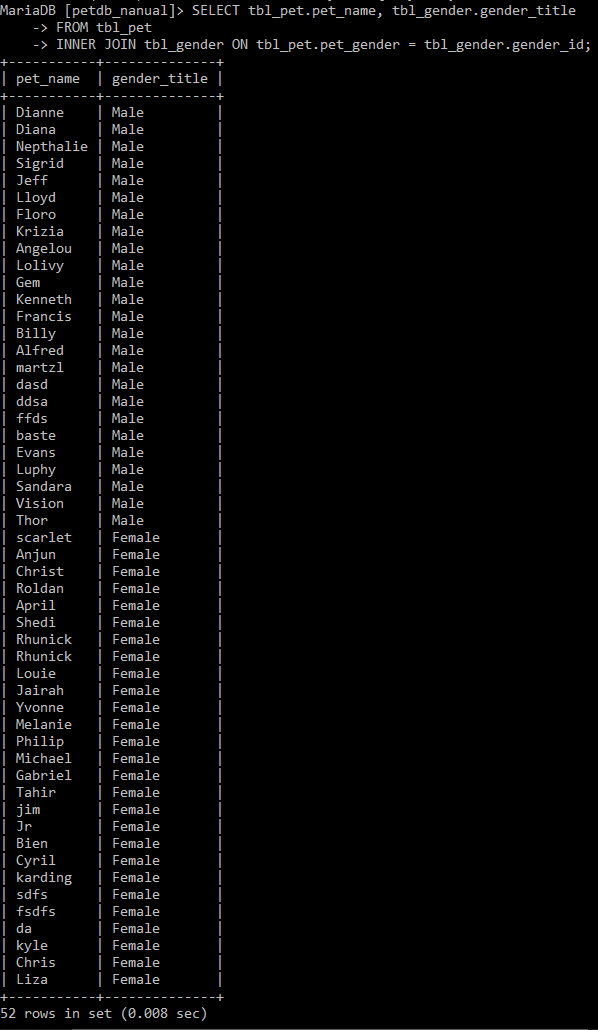
Text

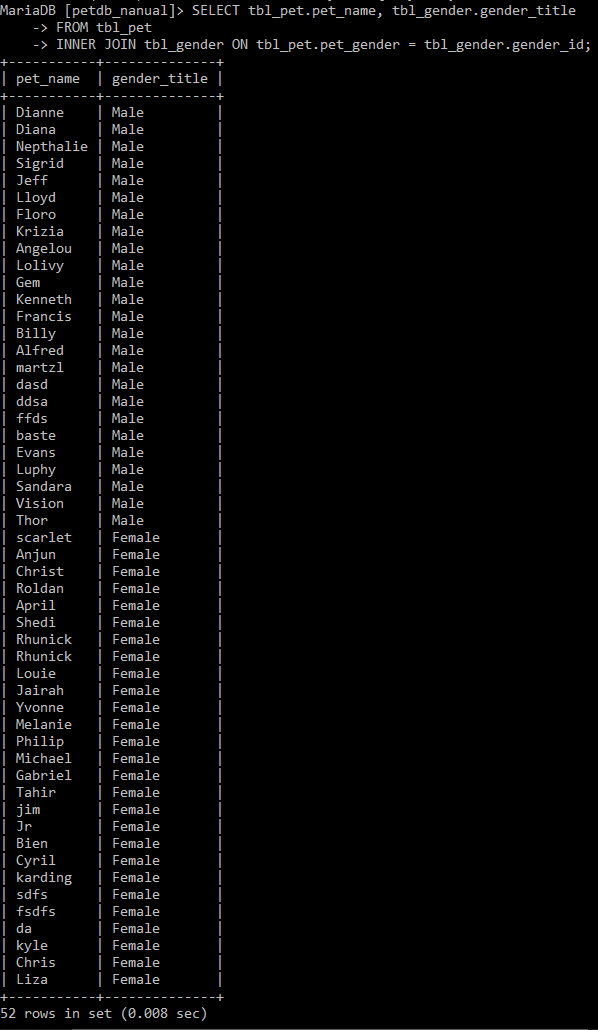
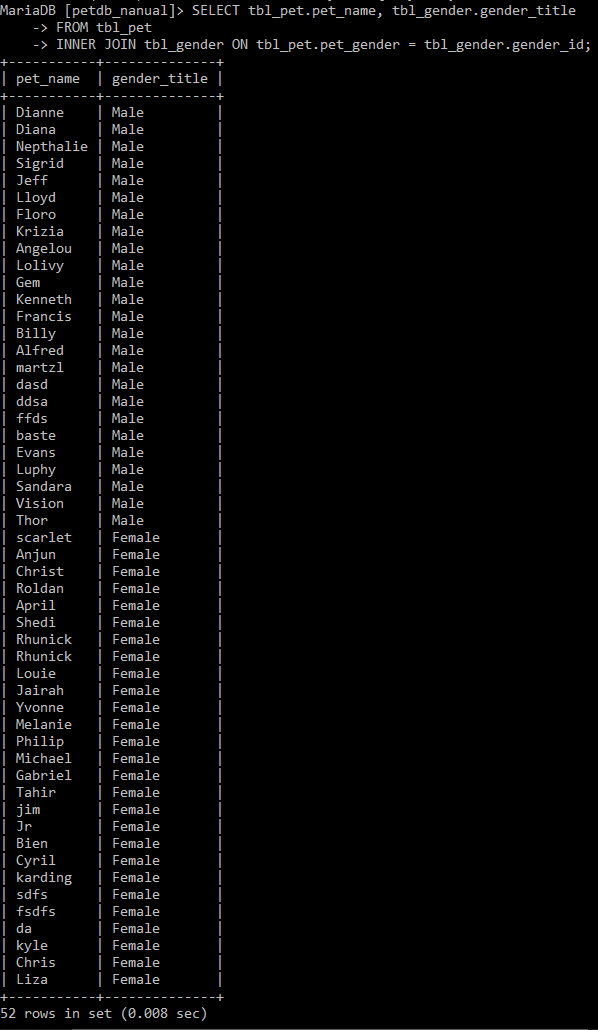
Description automatically generated

**TASKS:**

1. Show name and gender of pet.

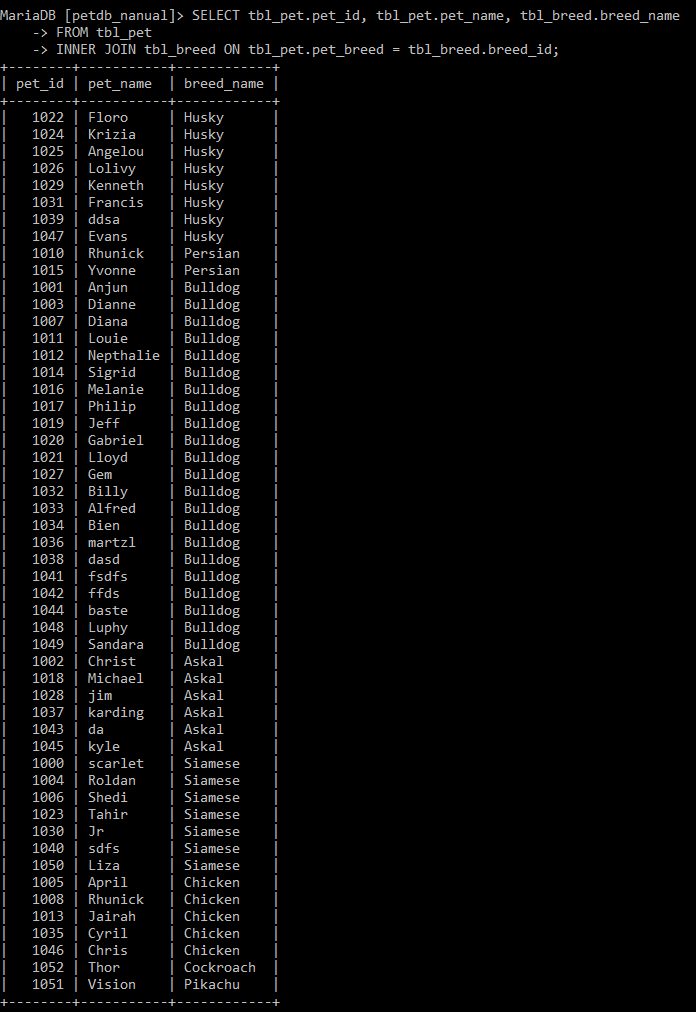
**My command prompt:**

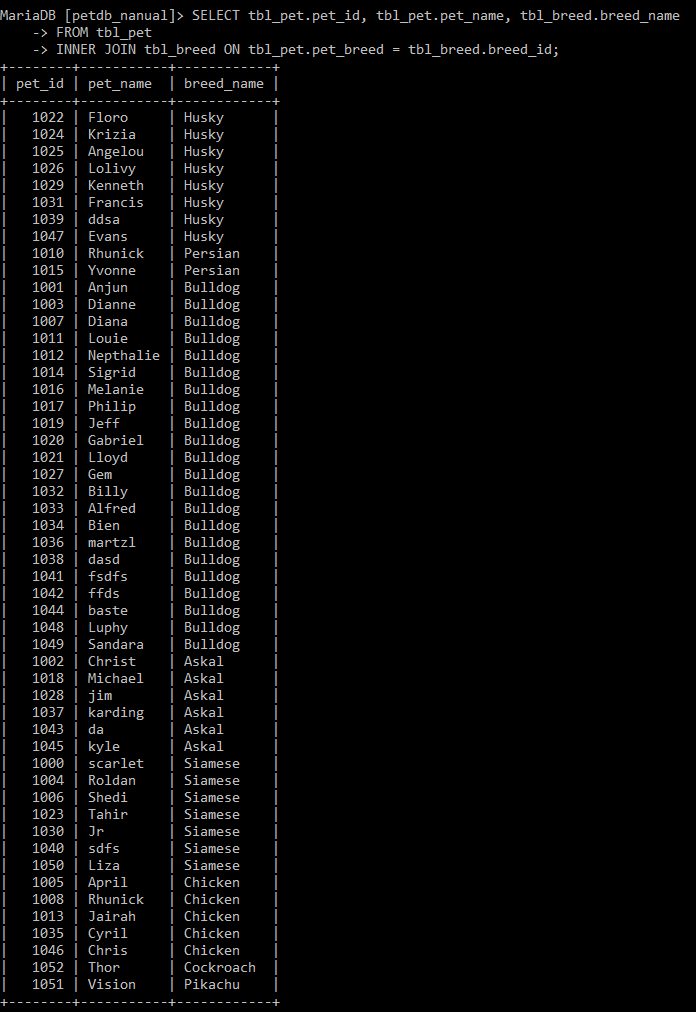
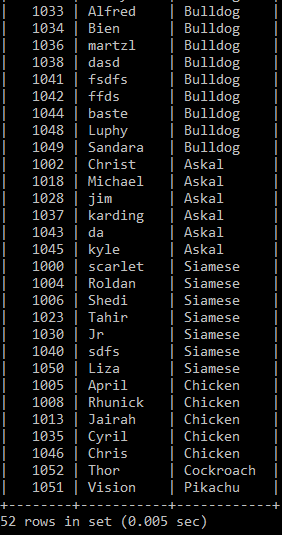


1. Show id, name, and breed of pet.

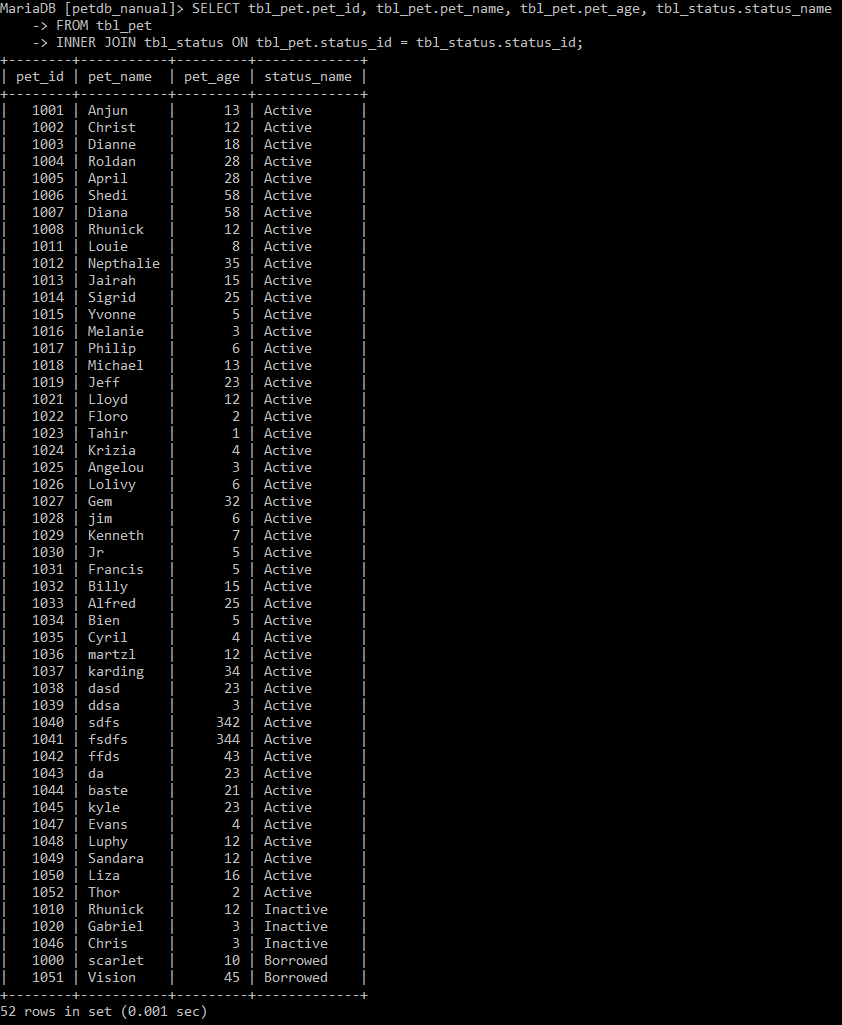
**My command prompt:**

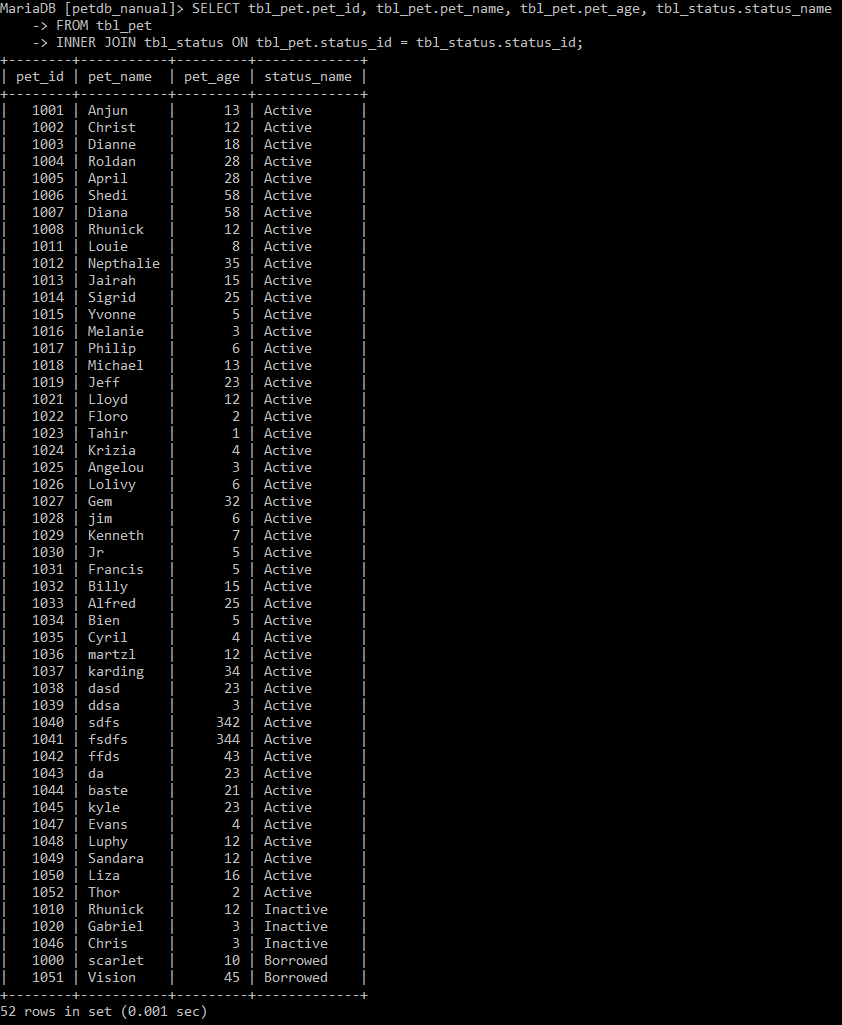
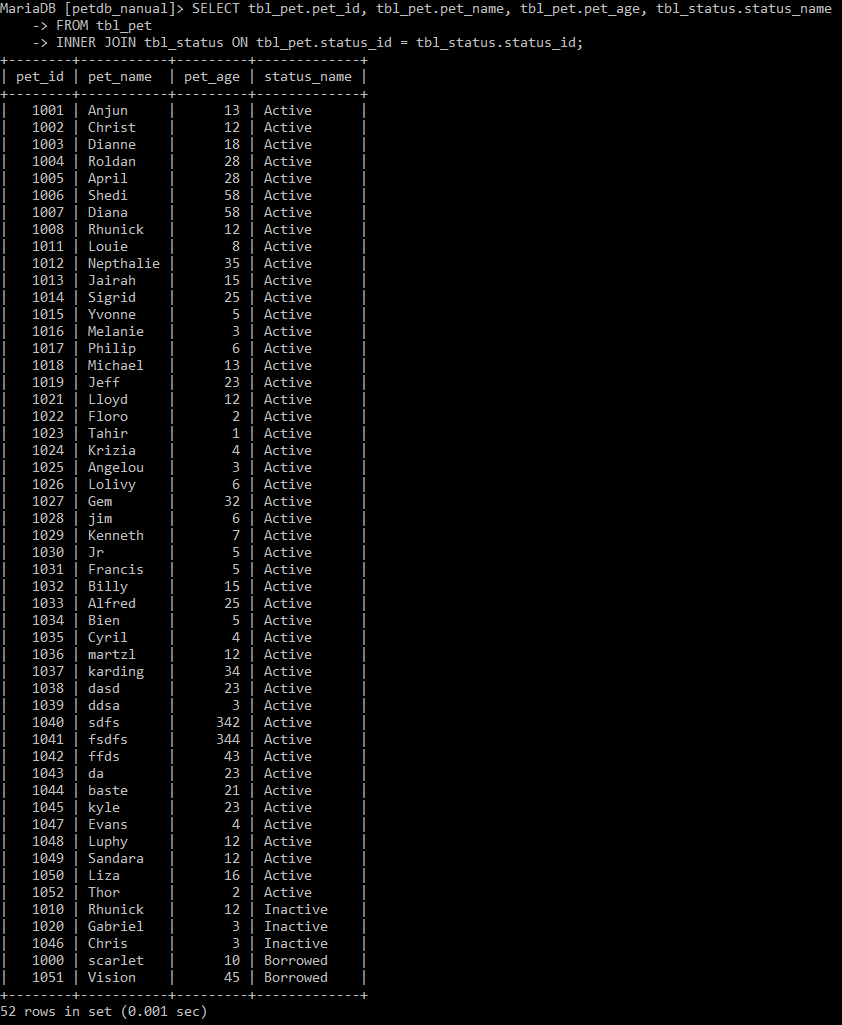


1. Show id, name, age, and status of pet.

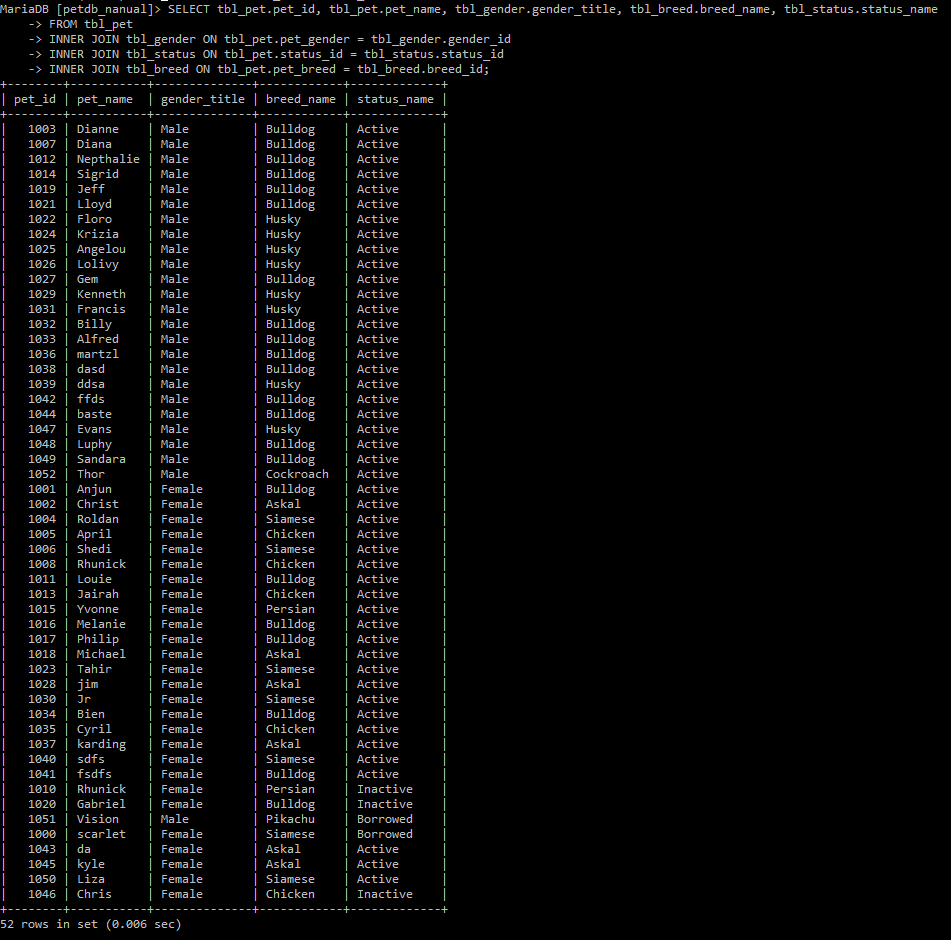
**My command prompt:**

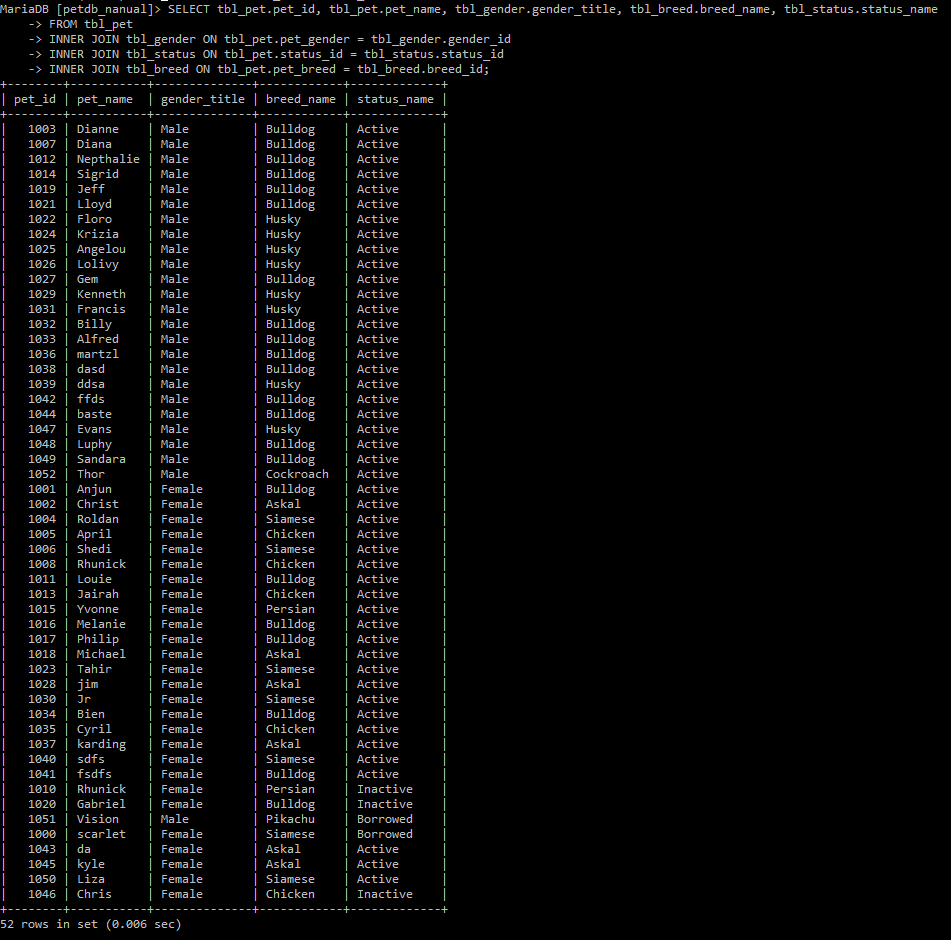
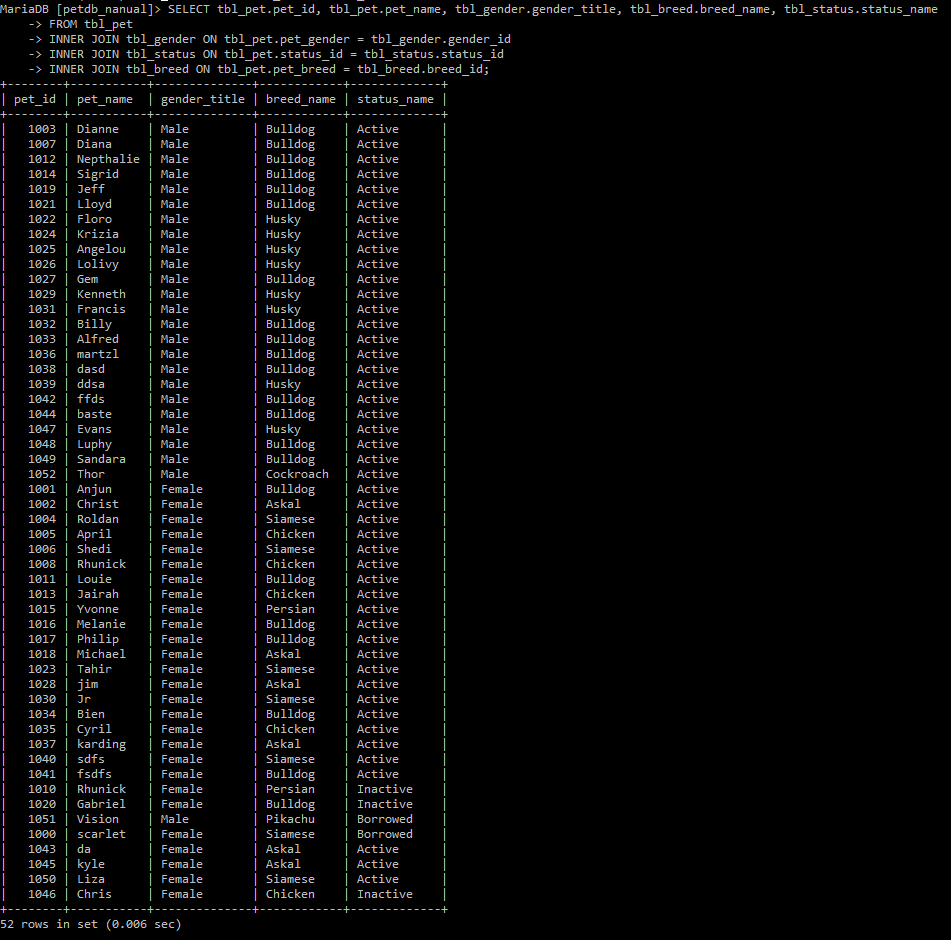


1. View id, name, gender, breed, and status of pet.

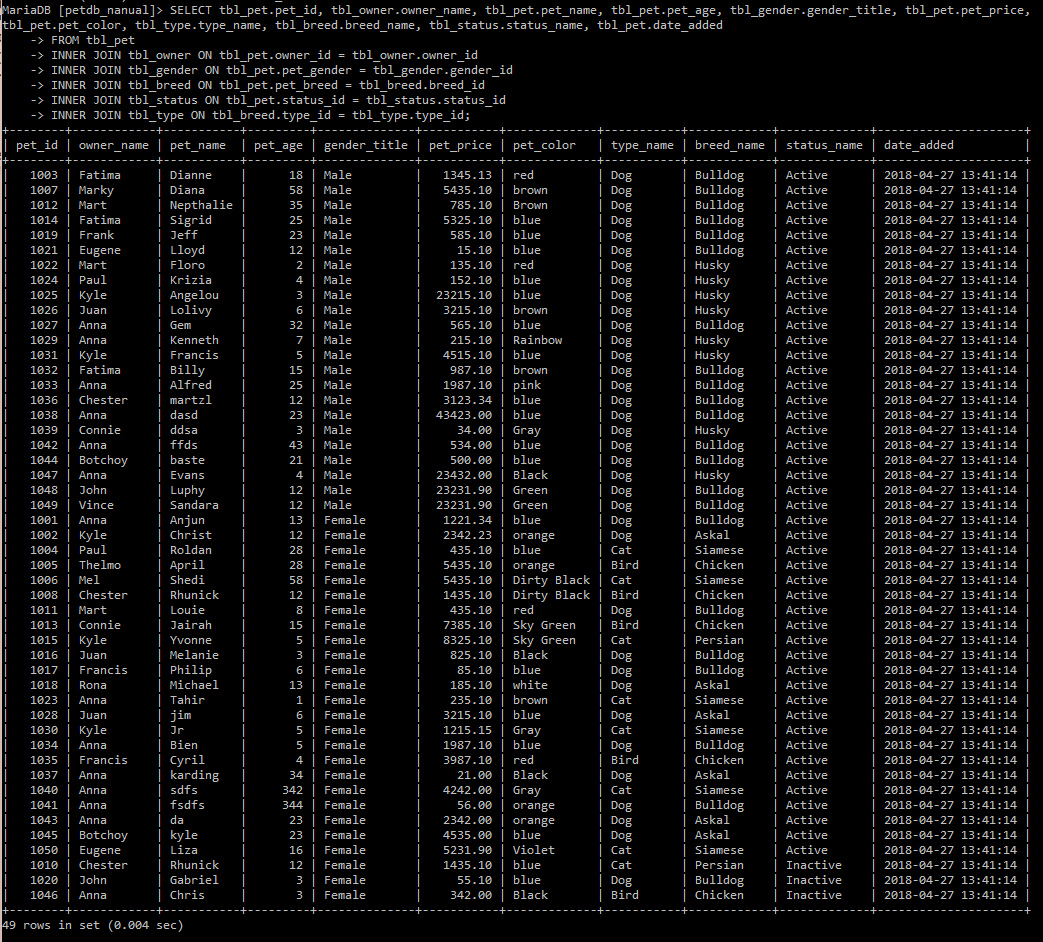
**My command prompt:**



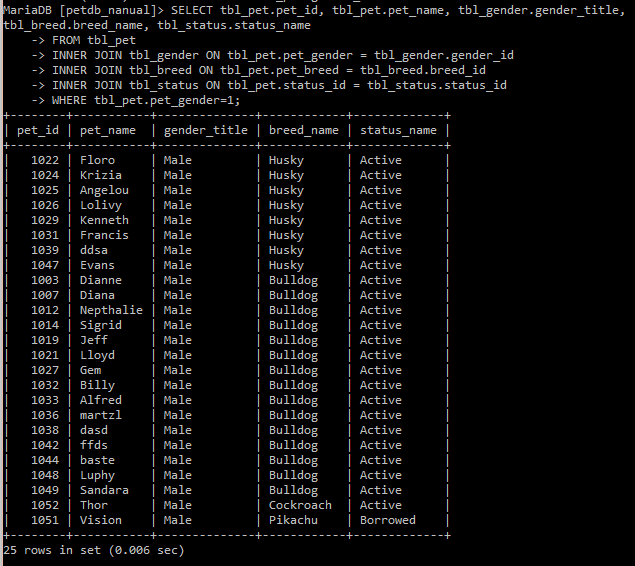
1. Display all pet’s information.

**My command prompt:**



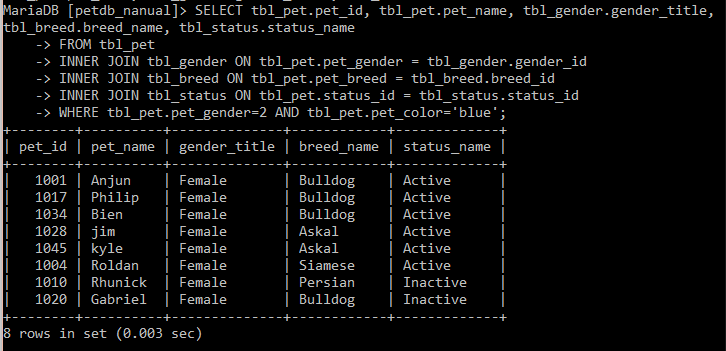
1. Display pet’s id, name, gender, breed, and status whose gender is male.

**My command prompt:**



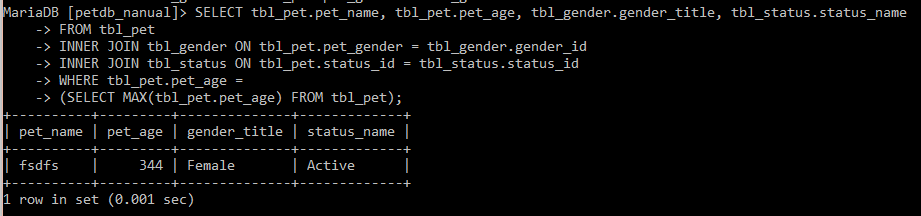
1. Display pet’s id, name, gender, breed, and status whose gender is female and color is ‘blue’.

**My command prompt:**



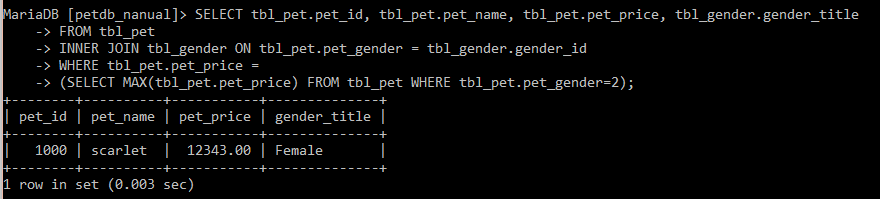
1. Show the pet’s name, age, gender, status of the eldest pet.

**My command prompt:**



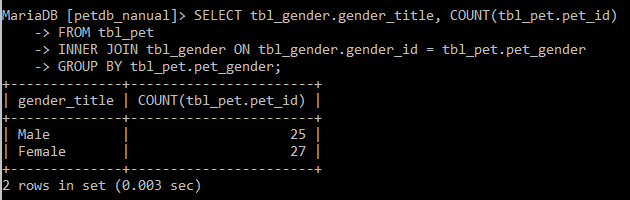
1. Show the most expensive pet who are female. Show pet’s id, name, price, gender.

**My command prompt:**



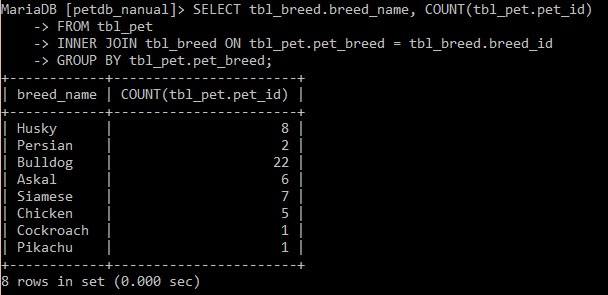
1. Show how many pets for each gender.

**My command prompt:**



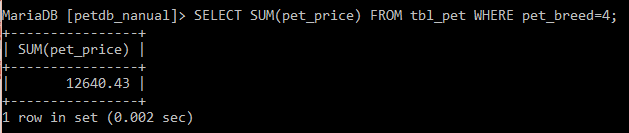
1. Show how many pets for each breed.

**My command prompt:**



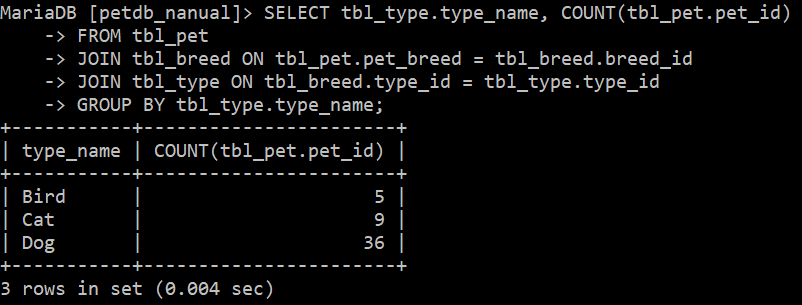
1. Show the total price of all pet whose breed is Askal.

**My command prompt:**



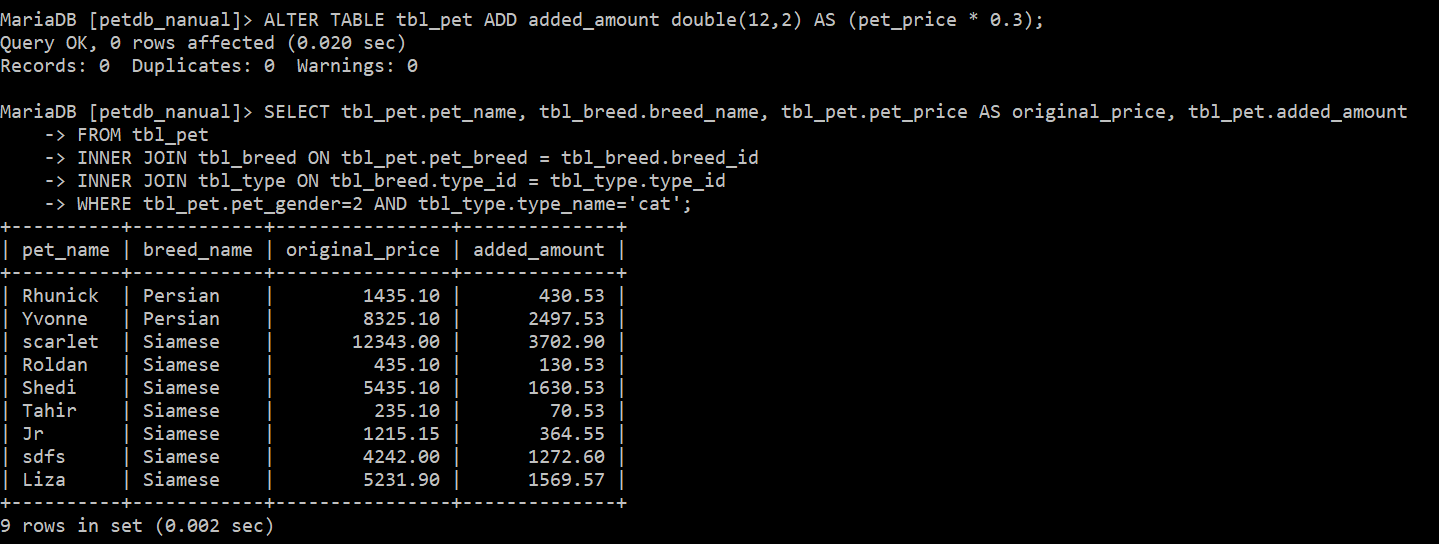
1. Show how many pets for each of the type.

**My command prompt:**



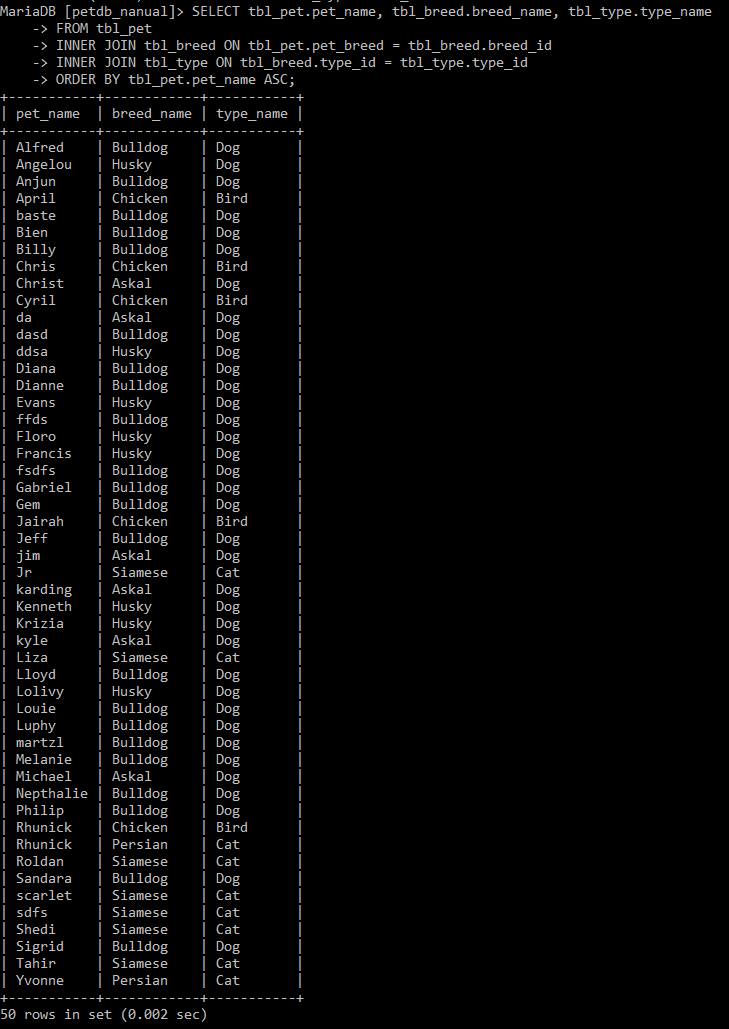
1. Show how much increase of the price of all female cats when it has mark up of 30% to its price.

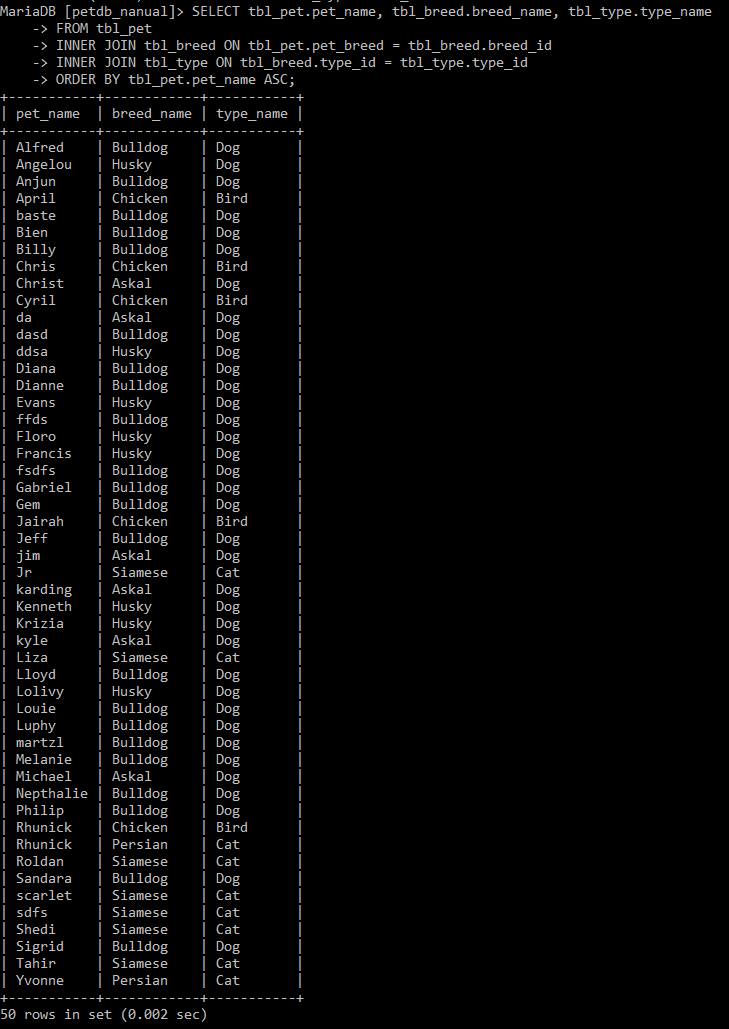
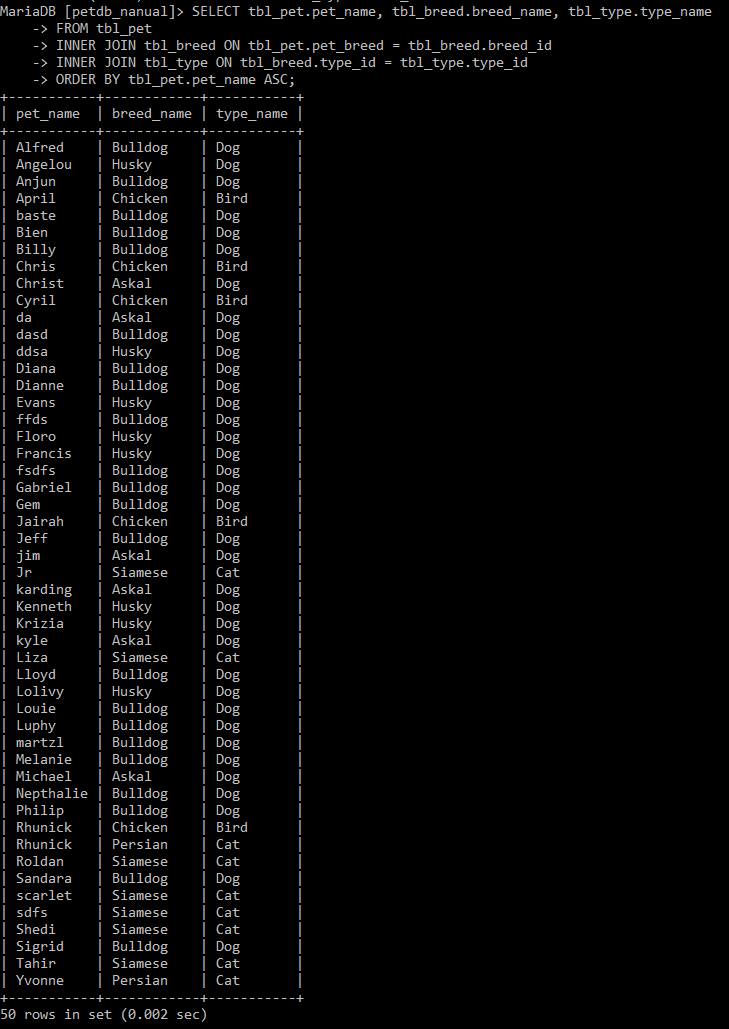
**My command prompt:**



1. Display what are the breeds and type of each pet.

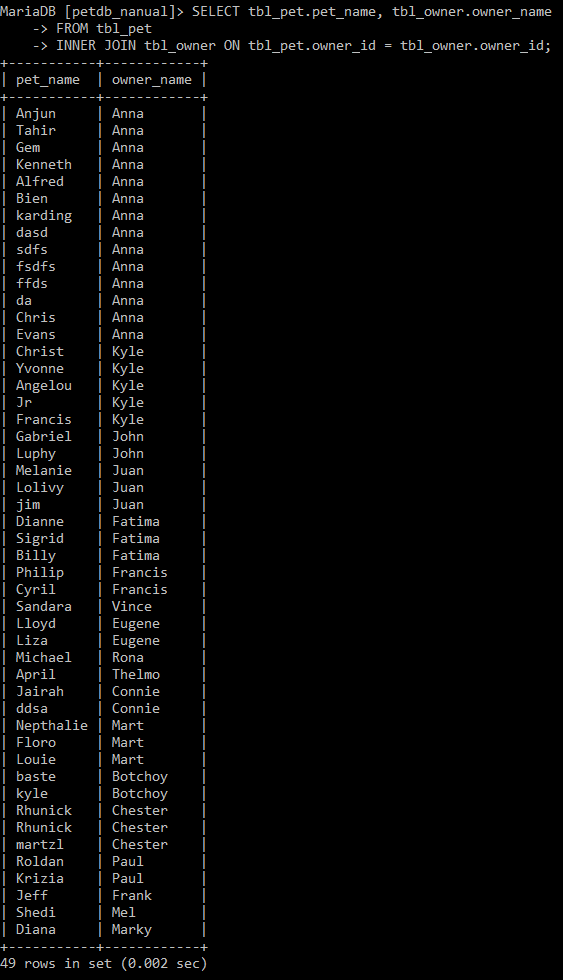
**My command prompt:**

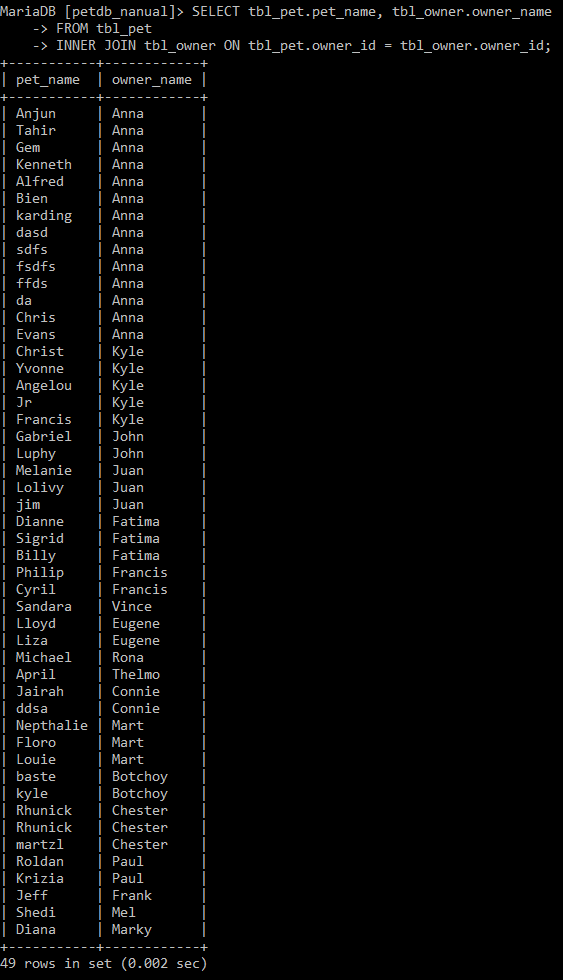
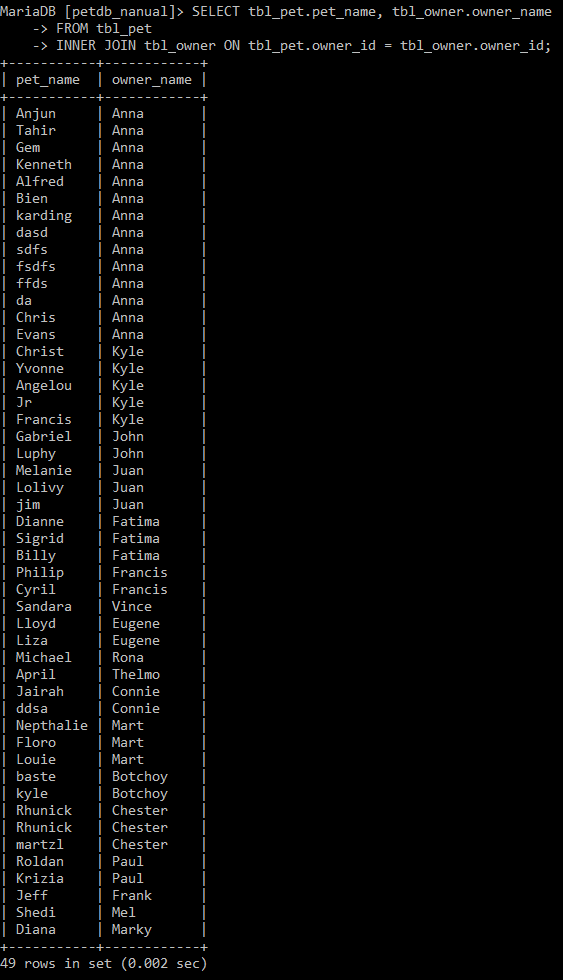


1. Show the pet’s name and its owner’s name.

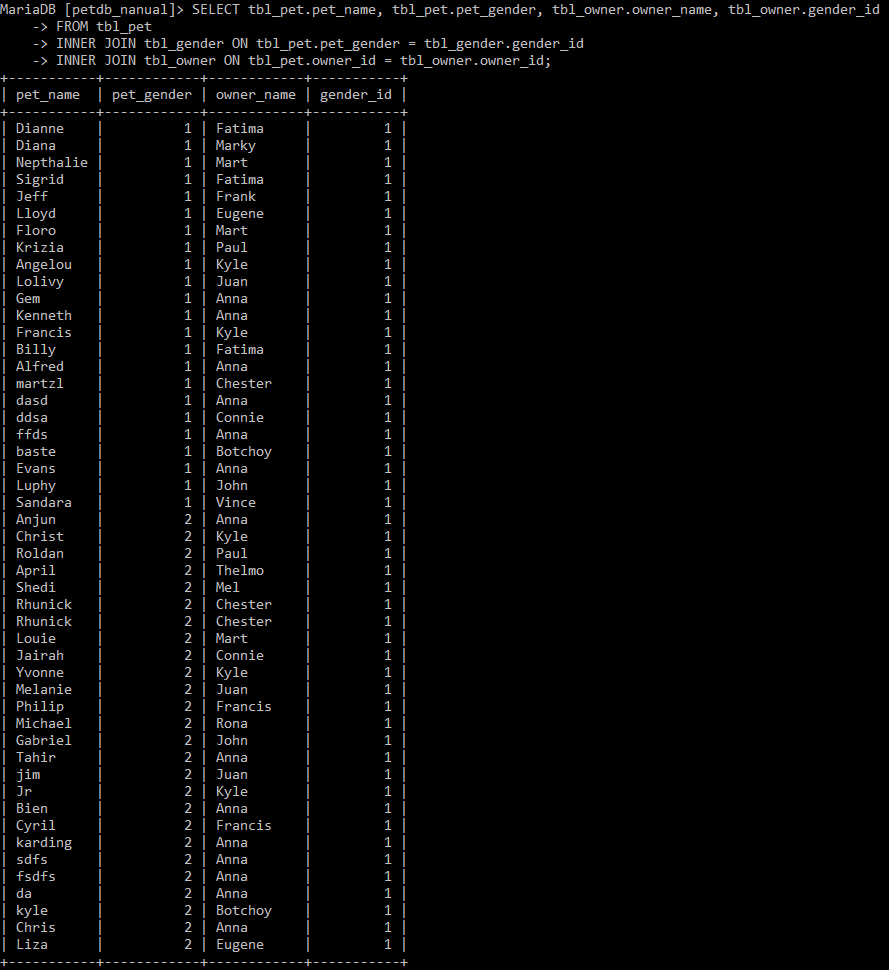
**My command prompt:**

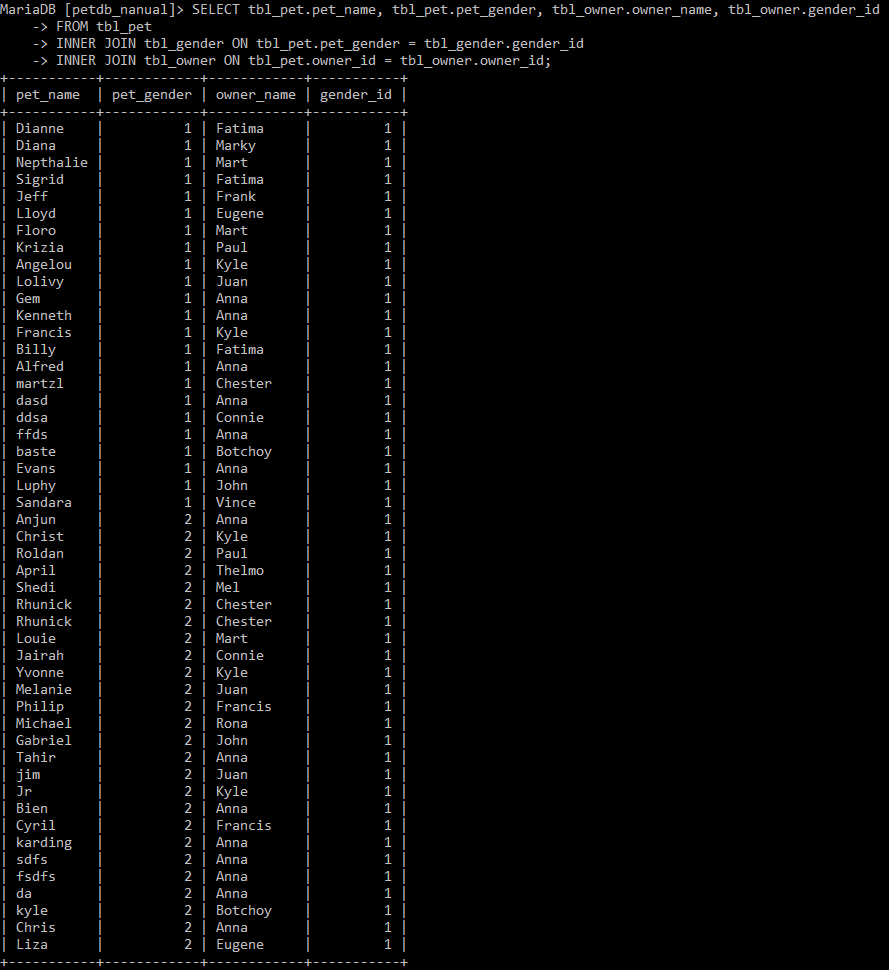
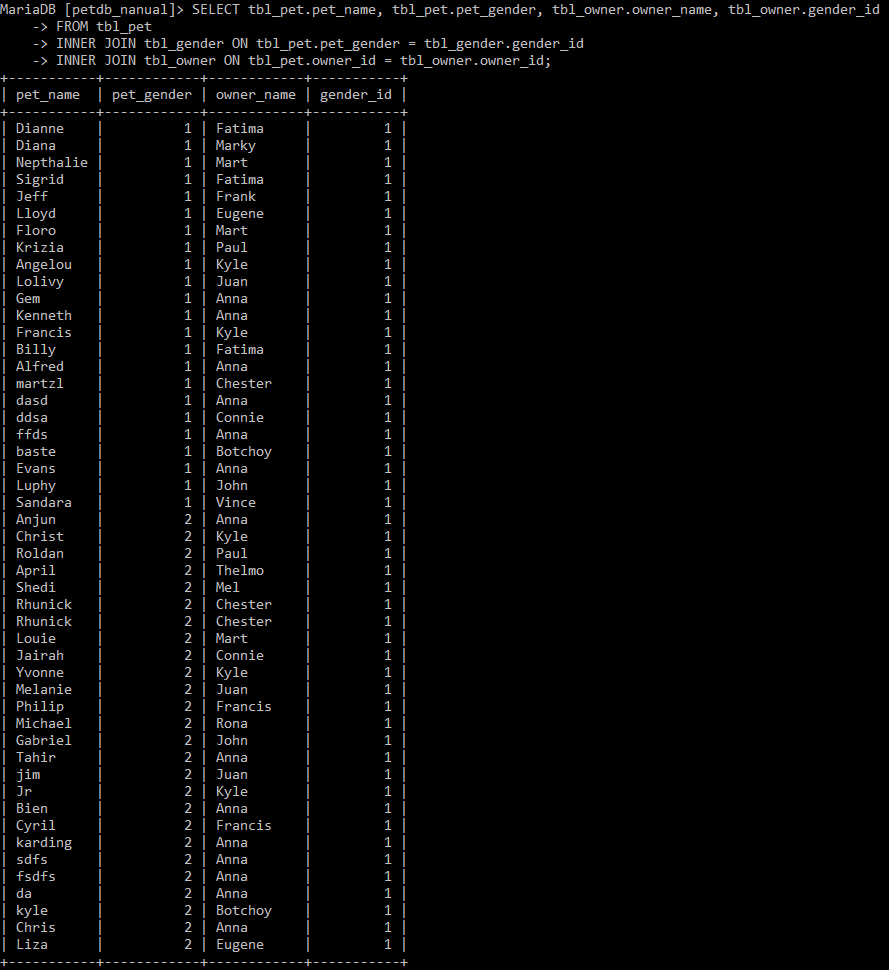


1. Show both the name and gender of pets and owners.

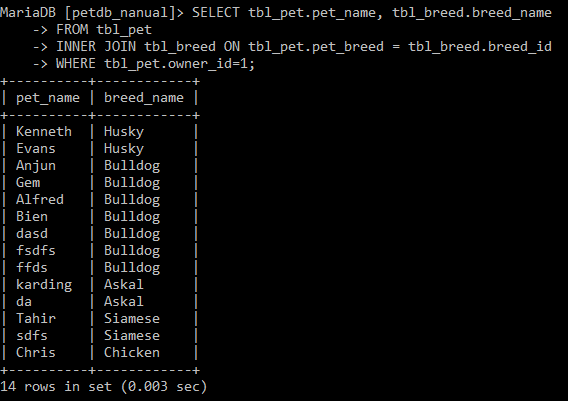
**My command prompt:**



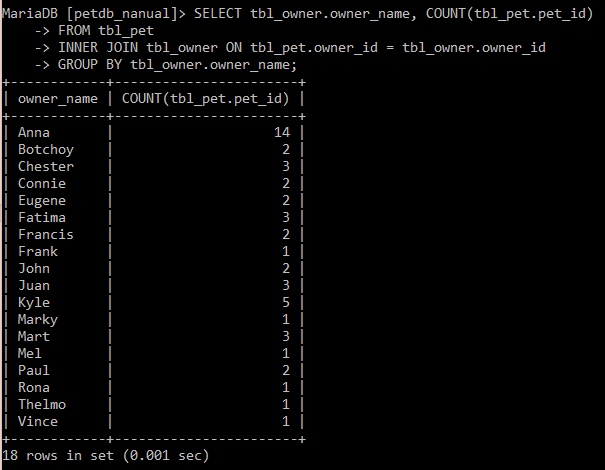
1. Display all pets [only name and breed] of owner 1.

**My command prompt:**



1. Show owners plus how many pets they owned.

**My command prompt:**



1. Show pets with no owner.

**My command prompt:**

