SQL Exercise

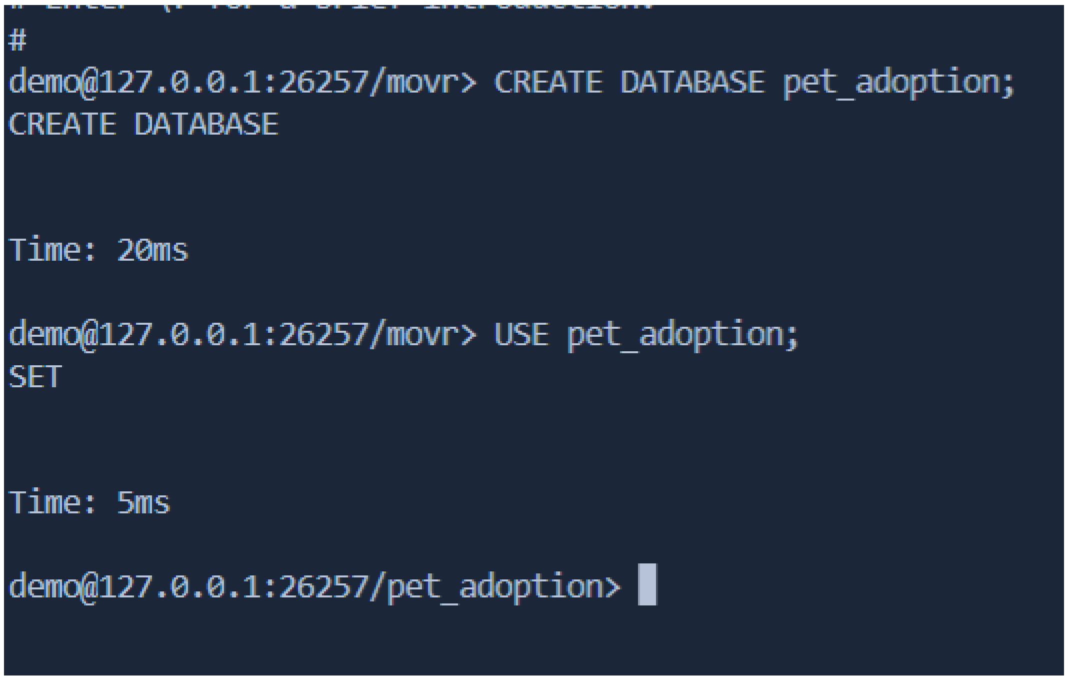
Set Up Dog Shelter Database: CREATE DATABASE

We need to create a database for our data. We use the following SQL statement:

CREATE DATABASE pet_adoption;

Creating a database doesn't automatically set it as the active database. We achieve that with the USE command.

USE pet_adoption;



```
#
demo@127.0.0.1:26257/movr> CREATE DATABASE pet_adoption;
CREATE DATABASE


Time: 20ms

demo@127.0.0.1:26257/movr> USE pet_adoption;
SET


Time: 5ms

demo@127.0.0.1:26257/pet_adoption> 
```

Table #1: A Table for Animals: CREATE TABLE & UUID

And

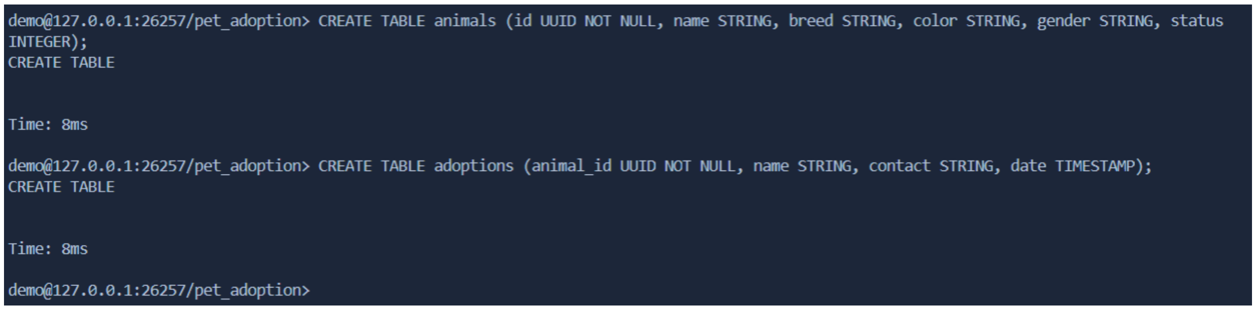
Table #2: The List of Adoptions: TIMESTAMP

We create the animal table with the following command:

CREATE TABLE animals (id UUID NOT NULL, name STRING, breed STRING, color STRING, gender STRING, status INTEGER);

We create the animal table with the following command:

CREATE TABLE adoptions (animalia UUID NOT NULL, name STRING, contact STRING, date TIMESTAMP);



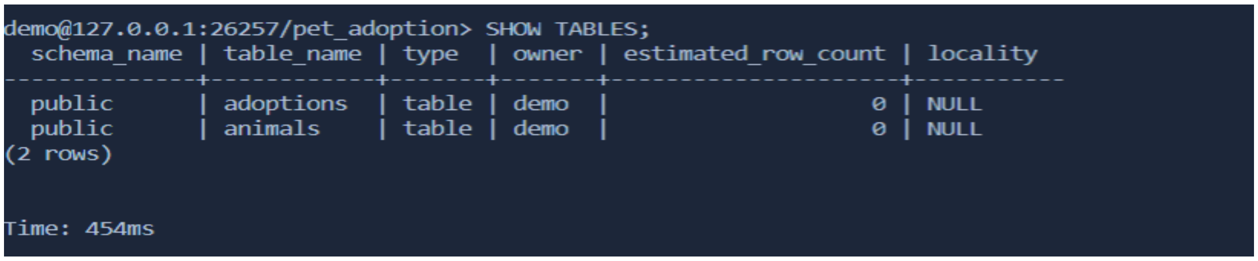
```
demo@127.0.0.1:26257/pet_adoption> CREATE TABLE animals (id UUID NOT NULL, name STRING, breed STRING, color STRING, gender STRING, status
INTEGER);
CREATE TABLE


Time: 8ms

demo@127.0.0.1:26257/pet_adoption> CREATE TABLE adoptions (animal_id UUID NOT NULL, name STRING, contact STRING, date TIMESTAMP);
CREATE TABLE


Time: 8ms

demo@127.0.0.1:26257/pet_adoption>
```

Verify Database Setup: SHOW TABLES & COLUMNS

To get the list of tables in the current database and check that we have both tables.

We achieve that with the following command:

SHOW TABLES;

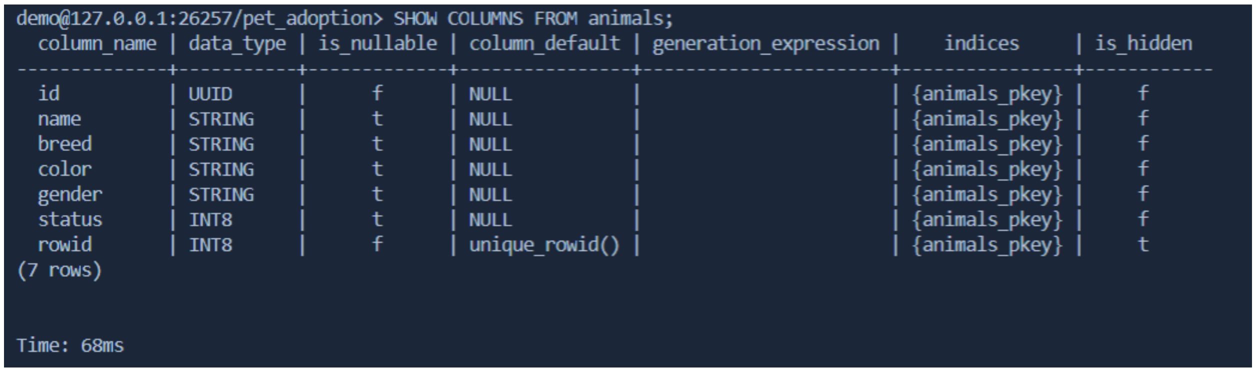


```
demo@127.0.0.1:26257/pet_adoption> SHOW TABLES;
  schema_name | table_name | type  | owner | estimated_row_count | locality
--------------+------------+-------+-------+---------------------+-----------
  public      | adoptions  | table | demo  |                   0 | NULL
  public      | animals    | table | demo  |                   0 | NULL
(2 rows)


Time: 454ms
```

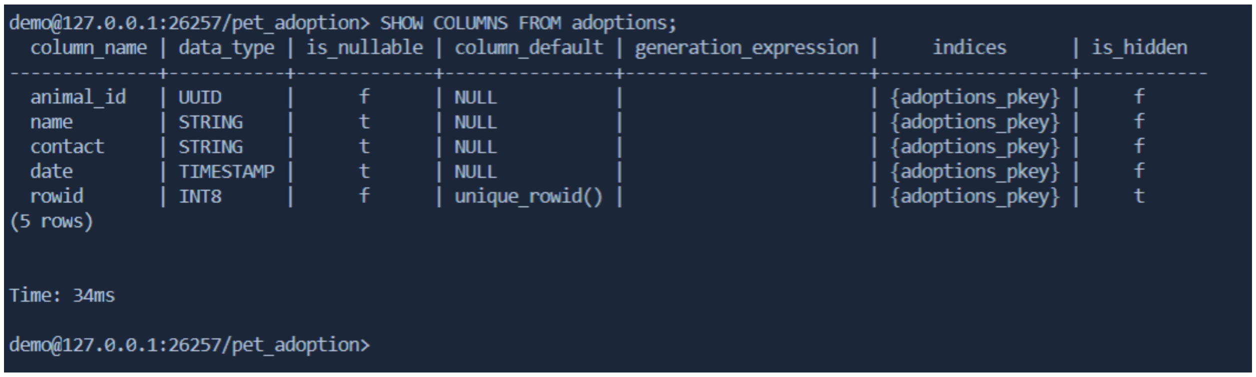
If you see both tables, then we can run these two statements to make sure that the columns of each are correct:

We achieve that with the following statement:

SHOW COLUMNS FROM animals;


```
demo@127.0.0.1:26257/pet_adoption> SHOW COLUMNS FROM animals;
  column_name | data_type | is_nullable | column_default | generation_expression |    indices     | is_hidden
--------------+-----------+-------------+----------------+-----------------------+----------------+------------
  id          | UUID      |      f      | NULL           |                       | {animals_pkey} |     f
  name        | STRING    |      t      | NULL           |                       | {animals_pkey} |     f
  breed       | STRING    |      t      | NULL           |                       | {animals_pkey} |     f
  color       | STRING    |      t      | NULL           |                       | {animals_pkey} |     f
  gender      | STRING    |      t      | NULL           |                       | {animals_pkey} |     f
  status      | INT8      |      t      | NULL           |                       | {animals_pkey} |     f
  rowid       | INT8      |      f      | unique_rowid() |                       | {animals_pkey} |     t
(7 rows)


Time: 68ms
```

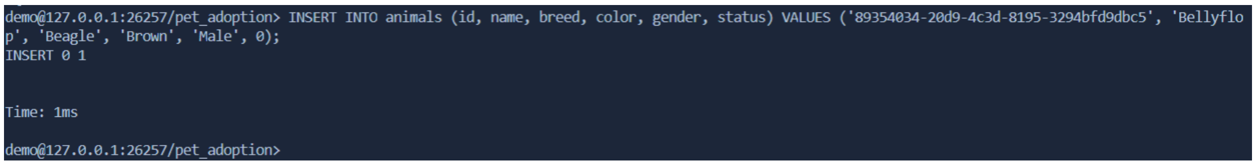
SHOW COLUMNS FROM adoptions;


```
demo@127.0.0.1:26257/pet_adoption> SHOW COLUMNS FROM adoptions;
  column_name | data_type | is_nullable | column_default | generation_expression |     indices      | is_hidden
--------------+-----------+-------------+----------------+-----------------------+------------------+------------
  animal_id   | UUID      |      f      | NULL           |                       | {adoptions_pkey} |     f
  name        | STRING    |      t      | NULL           |                       | {adoptions_pkey} |     f
  contact     | STRING    |      t      | NULL           |                       | {adoptions_pkey} |     f
  date        | TIMESTAMP |      t      | NULL           |                       | {adoptions_pkey} |     f
  rowid       | INT8      |      f      | unique_rowid() |                       | {adoptions_pkey} |     t
(5 rows)


Time: 34ms

demo@127.0.0.1:26257/pet_adoption>
```

Add Dogs to Database: INSERT

For us, to be able to start adding, animals, we use the INSERT statement:

INSERT INTO animals (id, name, breed, color, gender, status) VALUES ('89354034-20d9-4c3d-8195-3294bfd9dbc5', 'Bellyflop', 'Beagle', 'Brown', 'Male', 0);



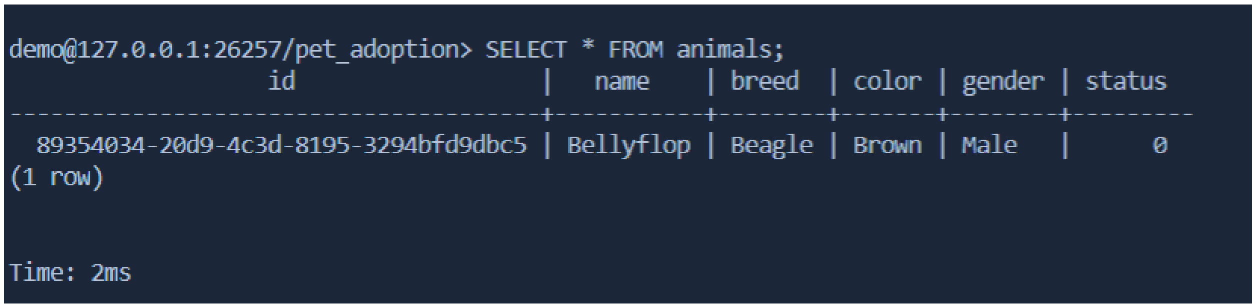
```
demo@127.0.0.1:26257/pet_adoption> INSERT INTO animals (id, name, breed, color, gender, status) VALUES ('89354034-20d9-4c3d-8195-3294bfd9dbc5', 'Bellyflo
p', 'Beagle', 'Brown', 'Male', 0);
INSERT 0 1


Time: 1ms

demo@127.0.0.1:26257/pet_adoption>
```

Retrieve List of Dogs: SELECT * FROM

With the full list added to our database, we can try running some SELECT queries to look through them.

Get the full list of all properties of all dogs (defaults to a limit of 100 rows):

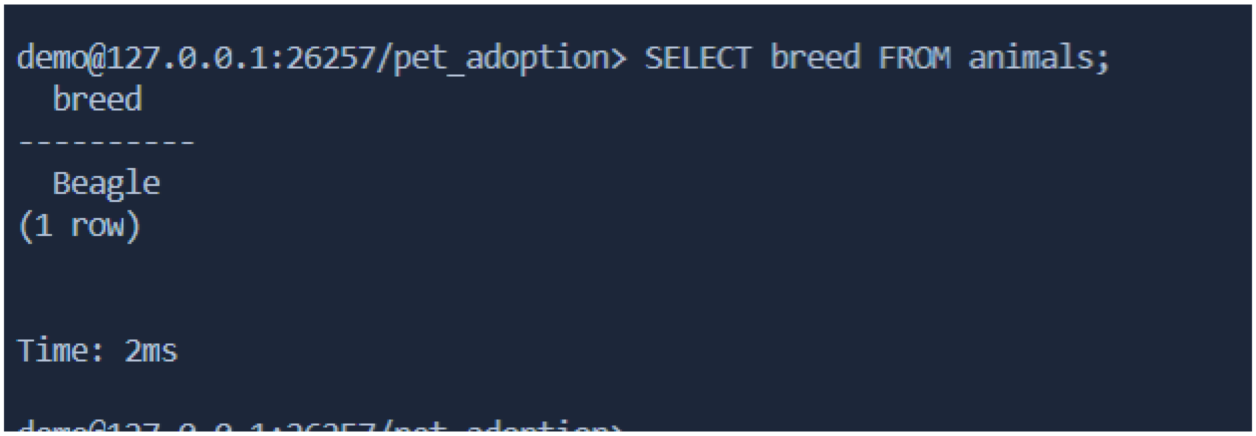
SELECT * FROM animals;



```
demo@127.0.0.1:26257/pet_adoption> SELECT * FROM animals;
                   id                  |   name    | breed  | color | gender | status
---------------------------------------+-----------+--------+-------+--------+---------
  89354034-20d9-4c3d-8195-3294bfd9dbc5 | Bellyflop | Beagle | Brown | Male   |      0
(1 row)


Time: 2ms
```

Get the breeds of all dogs:

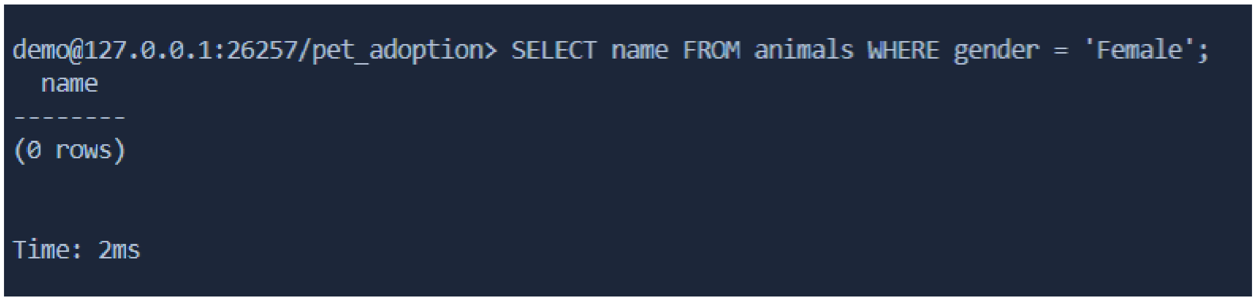
SELECT breed FROM animals;



```
demo@127.0.0.1:26257/pet_adoption> SELECT breed FROM animals;
  breed
----------
  Beagle
(1 row)


Time: 2ms

demo@127.0.0.1:26257/pet_adoption>
```

Get the names of only female dogs by including a WHERE clause:

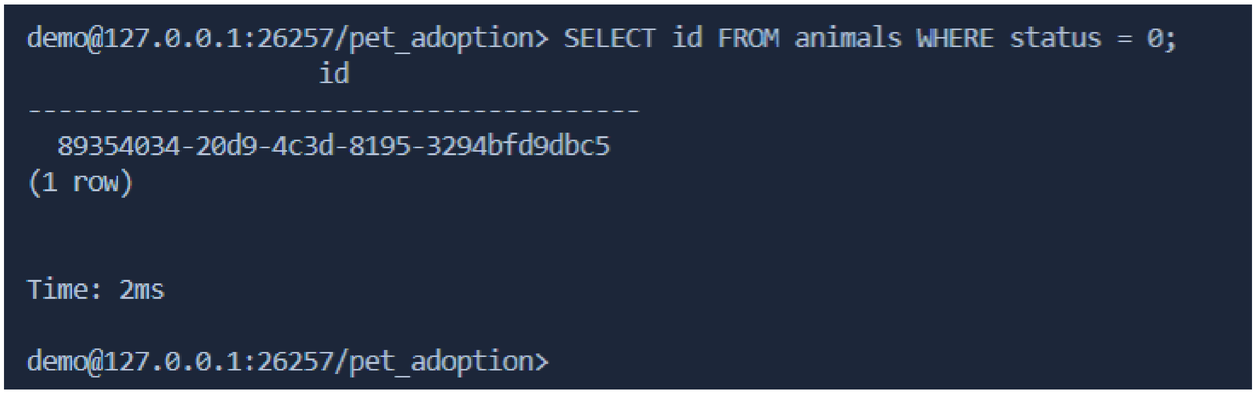
SELECT name FROM animals WHERE gender = 'Female';



```
demo@127.0.0.1:26257/pet_adoption> SELECT name FROM animals WHERE gender = 'Female';
  name
--------
(0 rows)


Time: 2ms
```

Get the IDs of dogs up for adoption:

SELECT id FROM animals WHERE status = 0;



```
demo@127.0.0.1:26257/pet_adoption> SELECT id FROM animals WHERE status = 0;
                   id
----------------------------------------
  89354034-20d9-4c3d-8195-3294bfd9dbc5
(1 row)


Time: 2ms

demo@127.0.0.1:26257/pet_adoption>
```