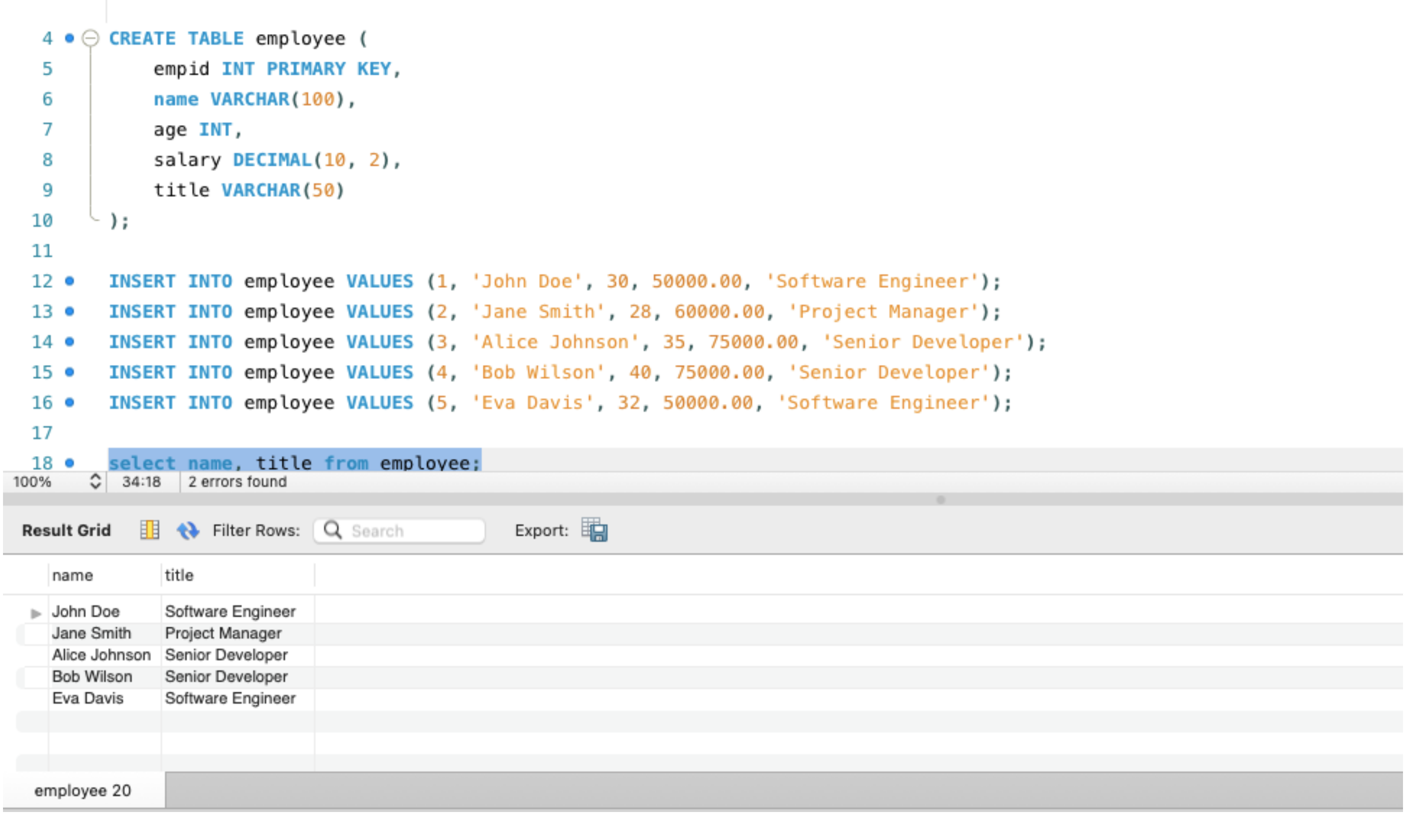
Assignment-03

Name-Ajay Chaudhary
Batch-Data engineering(batch-1)

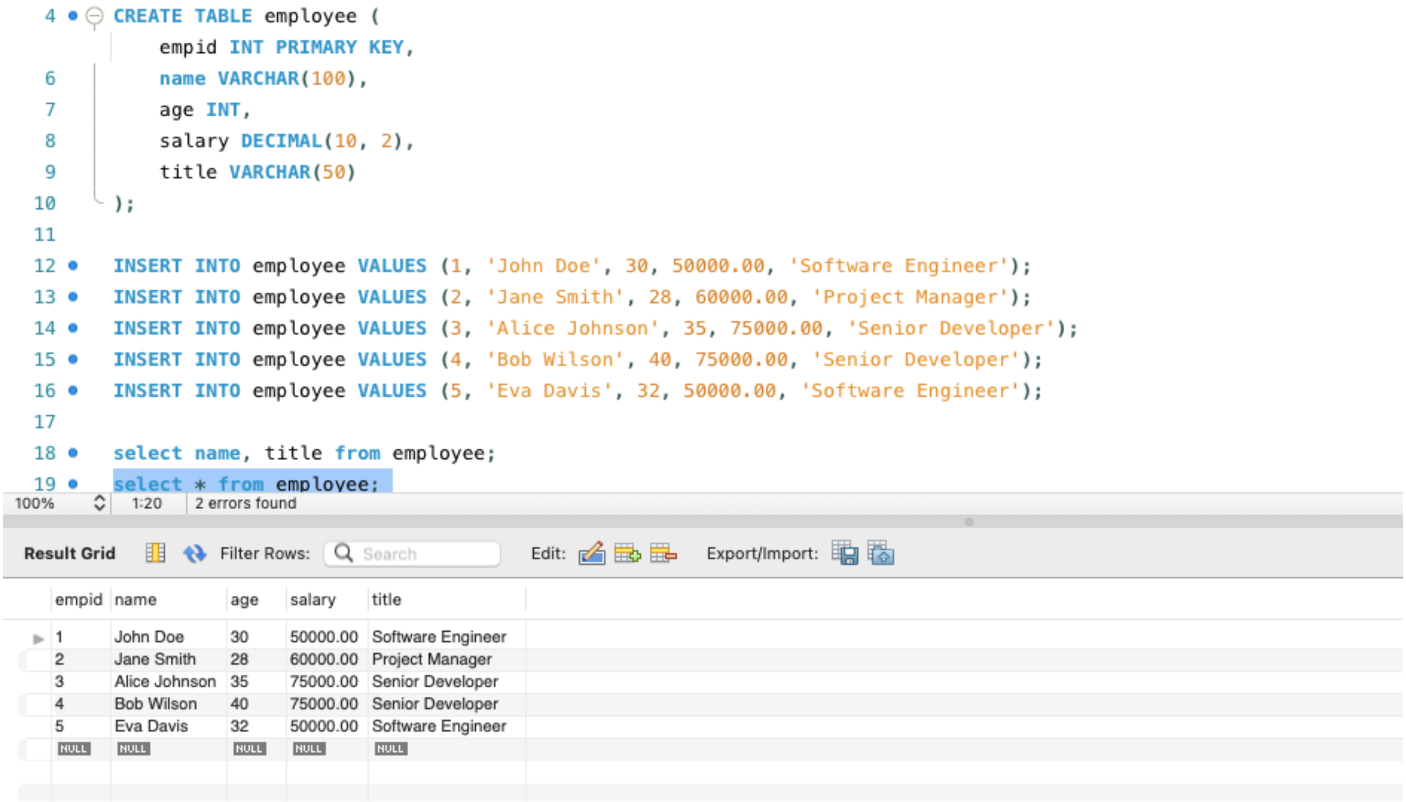
Q. Using Select statement show details of the employees.



```sql
CREATE TABLE employee (
    empid INT PRIMARY KEY,
    name VARCHAR(100),
    age INT,
    salary DECIMAL(10, 2),
    title VARCHAR(50)
);

INSERT INTO employee VALUES (1, 'John Doe', 30, 50000.00, 'Software Engineer');
INSERT INTO employee VALUES (2, 'Jane Smith', 28, 60000.00, 'Project Manager');
INSERT INTO employee VALUES (3, 'Alice Johnson', 35, 75000.00, 'Senior Developer');
INSERT INTO employee VALUES (4, 'Bob Wilson', 40, 75000.00, 'Senior Developer');
INSERT INTO employee VALUES (5, 'Eva Davis', 32, 50000.00, 'Software Engineer');

select name, title from employee;
```

name	title
John Doe	Software Engineer
Jane Smith	Project Manager
Alice Johnson	Senior Developer
Bob Wilson	Senior Developer
Eva Davis	Software Engineer

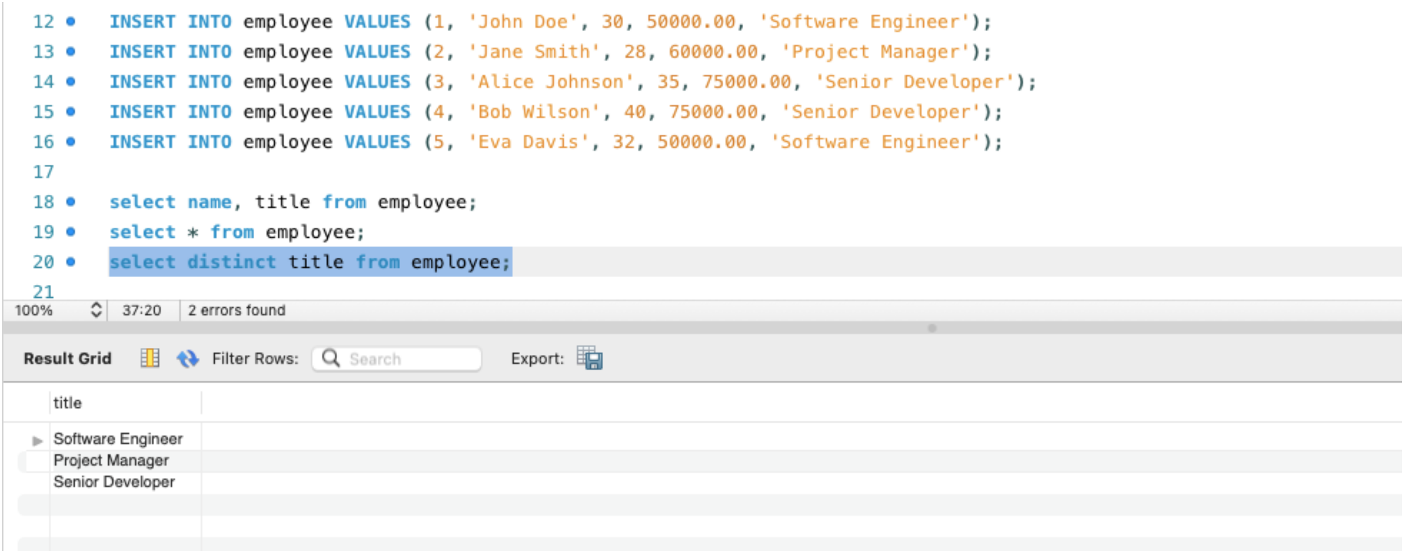
Q.Select all columns of the employee table-



```sql
CREATE TABLE employee (
    empid INT PRIMARY KEY,
    name VARCHAR(100),
    age INT,
    salary DECIMAL(10, 2),
    title VARCHAR(50)
);

INSERT INTO employee VALUES (1, 'John Doe', 30, 50000.00, 'Software Engineer');
INSERT INTO employee VALUES (2, 'Jane Smith', 28, 60000.00, 'Project Manager');
INSERT INTO employee VALUES (3, 'Alice Johnson', 35, 75000.00, 'Senior Developer');
INSERT INTO employee VALUES (4, 'Bob Wilson', 40, 75000.00, 'Senior Developer');
INSERT INTO employee VALUES (5, 'Eva Davis', 32, 50000.00, 'Software Engineer');

select name, title from employee;
select * from employee;
```

empid	name	age	salary	title
1	John Doe	30	50000.00	Software Engineer
2	Jane Smith	28	60000.00	Project Manager
3	Alice Johnson	35	75000.00	Senior Developer
4	Bob Wilson	40	75000.00	Senior Developer
5	Eva Davis	32	50000.00	Software Engineer
NULL	NULL	NULL	NULL	NULL

Q.Select Distinct title from employee table-

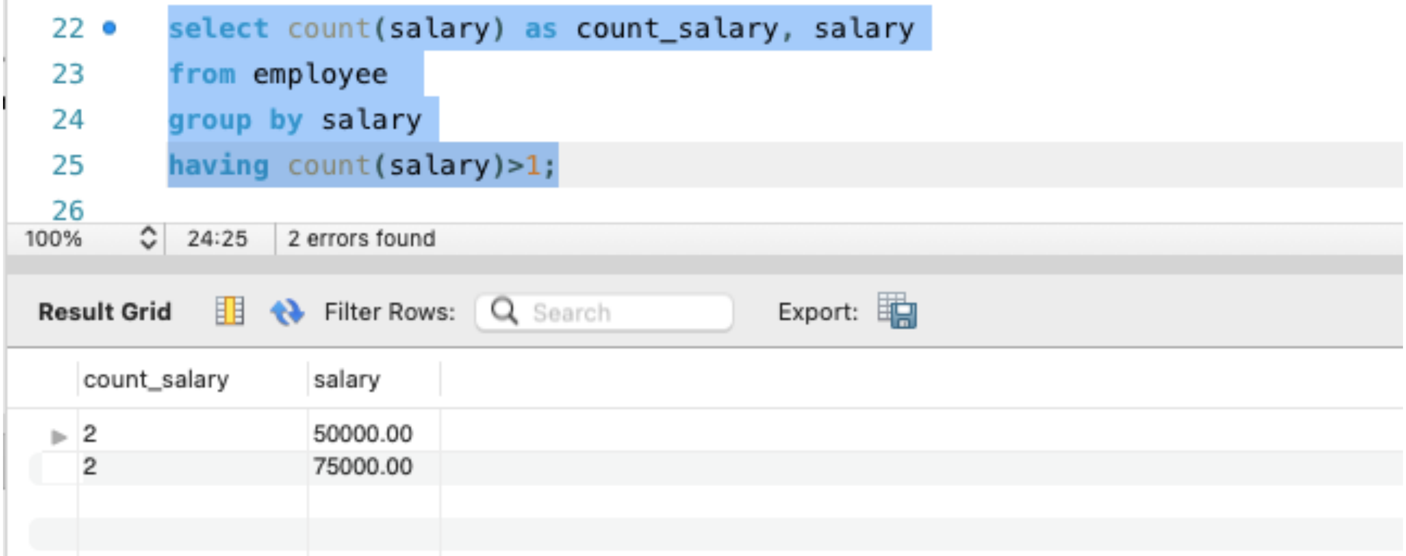


```sql
INSERT INTO employee VALUES (1, 'John Doe', 30, 50000.00, 'Software Engineer');
INSERT INTO employee VALUES (2, 'Jane Smith', 28, 60000.00, 'Project Manager');
INSERT INTO employee VALUES (3, 'Alice Johnson', 35, 75000.00, 'Senior Developer');
INSERT INTO employee VALUES (4, 'Bob Wilson', 40, 75000.00, 'Senior Developer');
INSERT INTO employee VALUES (5, 'Eva Davis', 32, 50000.00, 'Software Engineer');

select name, title from employee;
select * from employee;
select distinct title from employee;
```

```
100%    37:20    2 errors found
```

```
title
Software Engineer
Project Manager
Senior Developer
```

Use of having and group by clause-



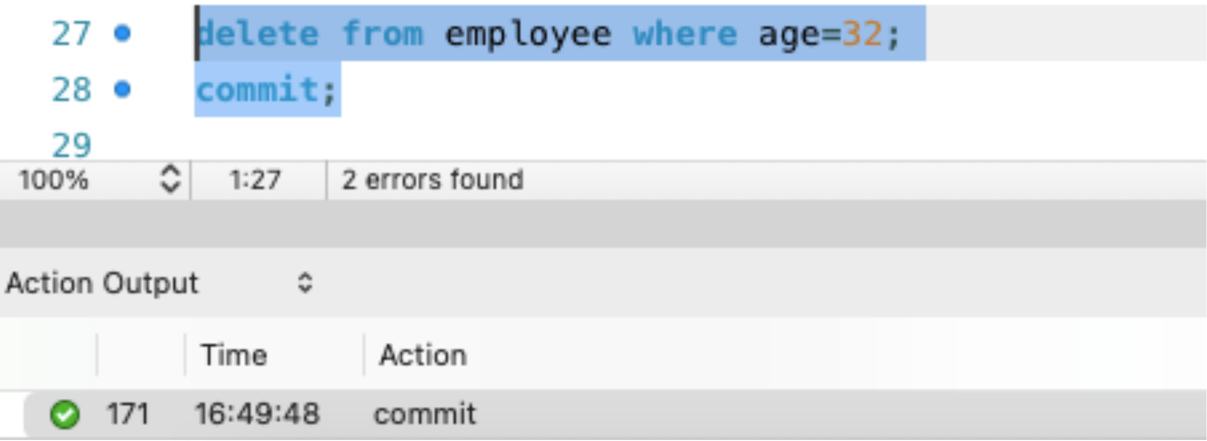
```sql
select count(salary) as count_salary, salary
from employee
group by salary
having count(salary)>1;
```

```
100%    24:25    2 errors found
```

```
count_salary    salary
2               50000.00
2               75000.00
```

Transactional Control commands-

Commit command

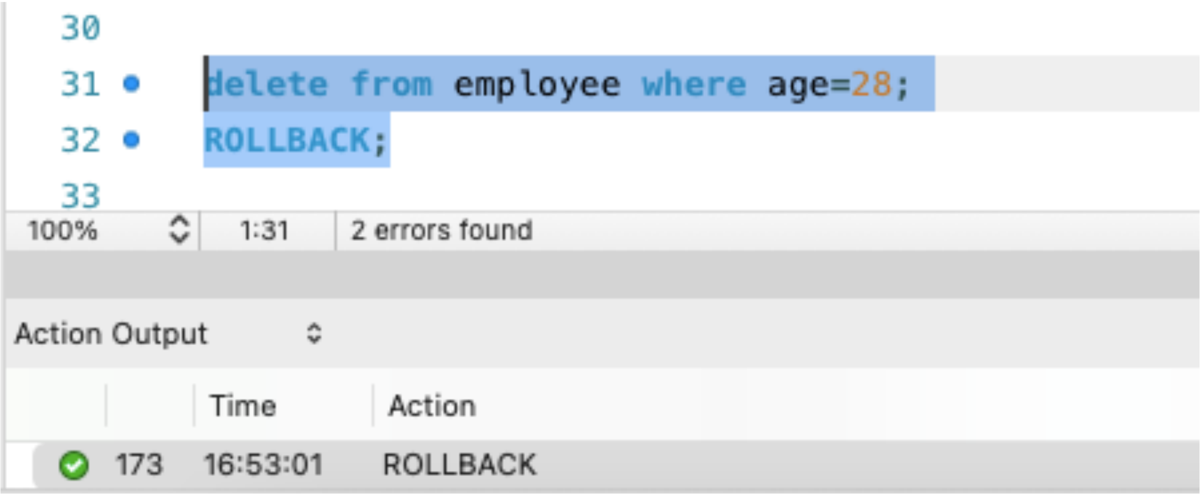


```sql
delete from employee where age=32;
commit;
```

```
100%    1:27    2 errors found
```

```
        Time        Action
✓ 171   16:49:48    commit
```

Rollback command



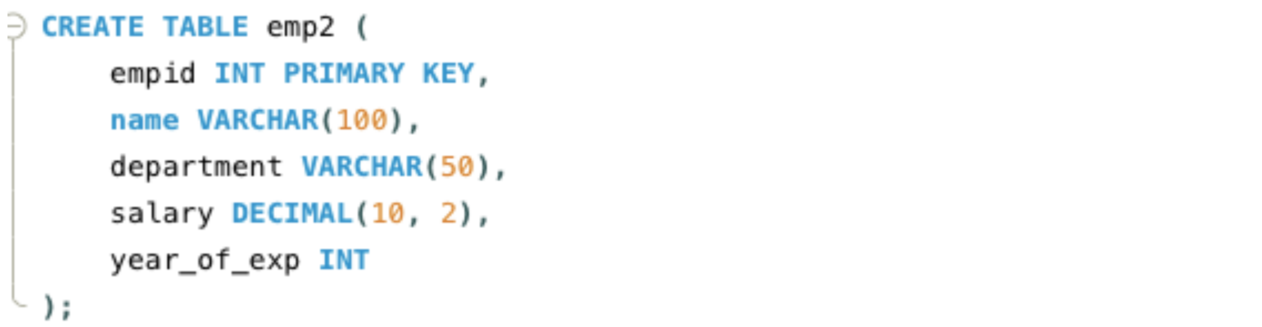
```
delete from employee where age=28;
ROLLBACK;
```

```
   Time      Action
173 16:53:01 ROLLBACK
```

Set operators-

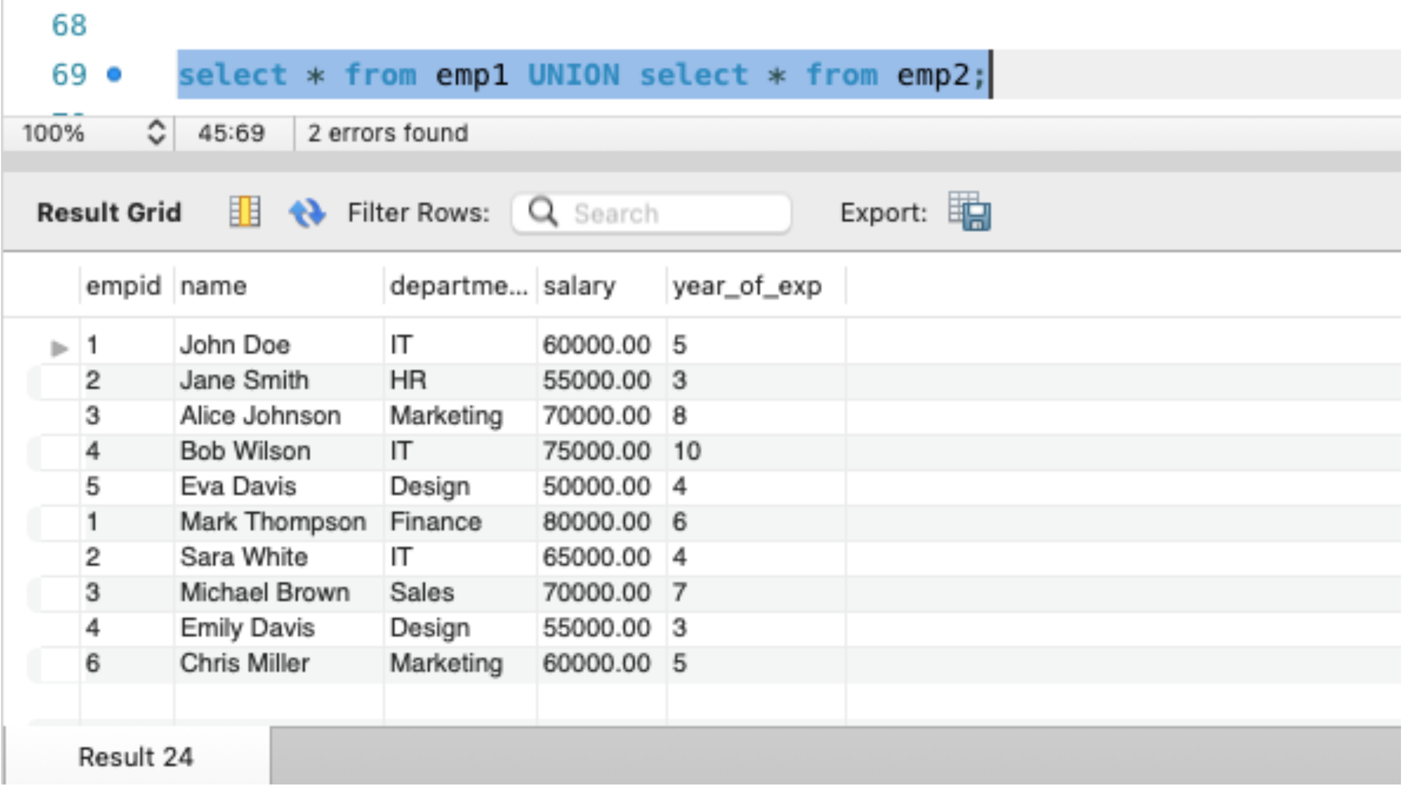
I made 2 table emp1 and emp2 for performing set operations.

```
CREATE TABLE emp1 (
    empid INT PRIMARY KEY,
    name VARCHAR(100),
    department VARCHAR(50),
    salary DECIMAL(10, 2),
    year_of_exp INT
);

INSERT INTO emp1 VALUES (1, 'John Doe', 'IT', 60000.00, 5);
INSERT INTO emp1 VALUES (2, 'Jane Smith', 'HR', 55000.00, 3);
INSERT INTO emp1 VALUES (3, 'Alice Johnson', 'Marketing', 70000.00, 8);
INSERT INTO emp1 VALUES (4, 'Bob Wilson', 'IT', 75000.00, 10);
INSERT INTO emp1 VALUES (5, 'Eva Davis', 'Design', 50000.00, 4);
```



```
CREATE TABLE emp2 (
    empid INT PRIMARY KEY,
    name VARCHAR(100),
    department VARCHAR(50),
    salary DECIMAL(10, 2),
    year_of_exp INT
);

INSERT INTO emp2 VALUES (1, 'Mark Thompson', 'Finance', 80000.00, 6);
INSERT INTO emp2 VALUES (2, 'Sara White', 'IT', 65000.00, 4);
INSERT INTO emp2 VALUES (3, 'Michael Brown', 'Sales', 70000.00, 7);
INSERT INTO emp2 VALUES (4, 'Emily Davis', 'Design', 55000.00, 3);
INSERT INTO emp2 VALUES (5, 'Eva Davis', 'Design', 50000.00, 4);
INSERT INTO emp2 VALUES (6, 'Chris Miller', 'Marketing', 60000.00, 5);
```

Union operator-

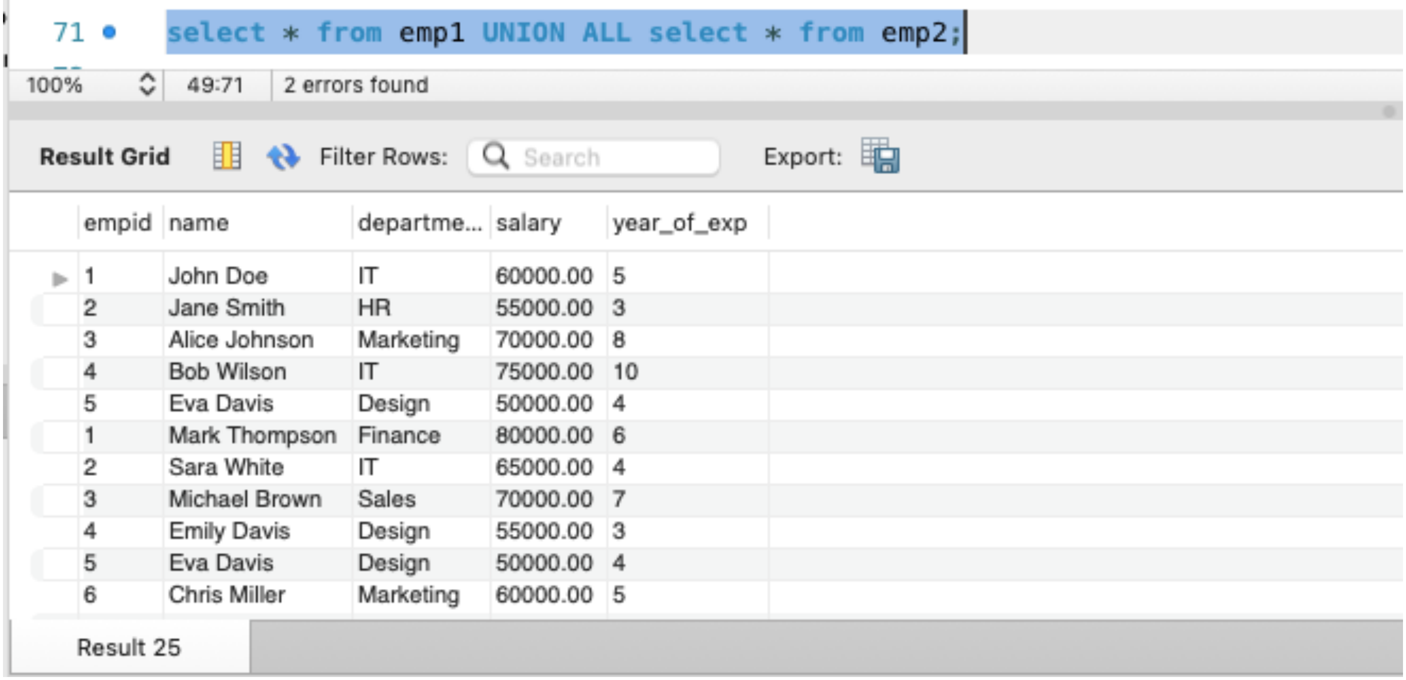


```
select * from emp1 UNION select * from emp2;
```

100%   45:69   2 errors found

empid	name	departme...	salary	year_of_exp
1	John Doe	IT	60000.00	5
2	Jane Smith	HR	55000.00	3
3	Alice Johnson	Marketing	70000.00	8
4	Bob Wilson	IT	75000.00	10
5	Eva Davis	Design	50000.00	4
1	Mark Thompson	Finance	80000.00	6
2	Sara White	IT	65000.00	4
3	Michael Brown	Sales	70000.00	7
4	Emily Davis	Design	55000.00	3
6	Chris Miller	Marketing	60000.00	5

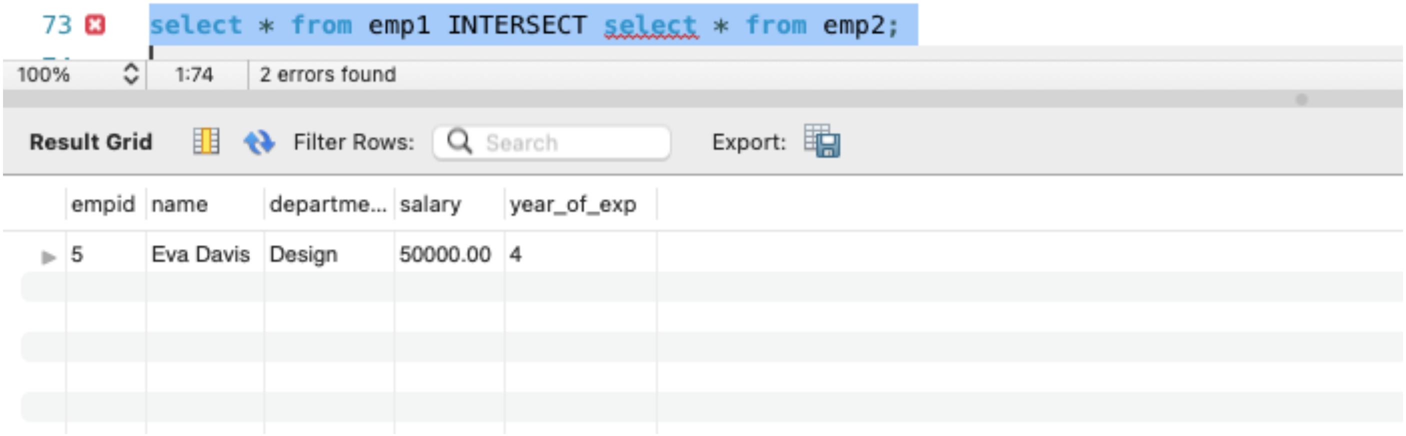
Union ALL operator-



```
select * from emp1 UNION ALL select * from emp2;
```

empid	name	departme...	salary	year_of_exp
1	John Doe	IT	60000.00	5
2	Jane Smith	HR	55000.00	3
3	Alice Johnson	Marketing	70000.00	8
4	Bob Wilson	IT	75000.00	10
5	Eva Davis	Design	50000.00	4
1	Mark Thompson	Finance	80000.00	6
2	Sara White	IT	65000.00	4
3	Michael Brown	Sales	70000.00	7
4	Emily Davis	Design	55000.00	3
5	Eva Davis	Design	50000.00	4
6	Chris Miller	Marketing	60000.00	5

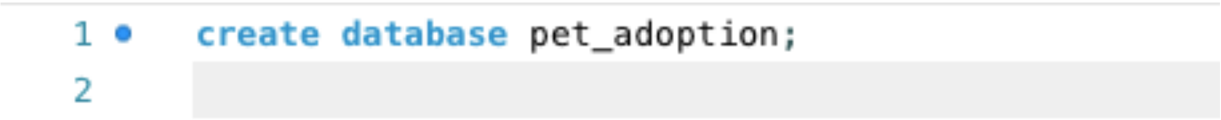
Intersect operation-



```
select * from emp1 INTERSECT select * from emp2;
```

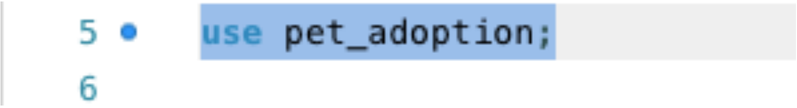
empid	name	departme...	salary	year_of_exp
5	Eva Davis	Design	50000.00	4

CREATE DATABASE pet_adoption



```
create database pet_adoption;
```

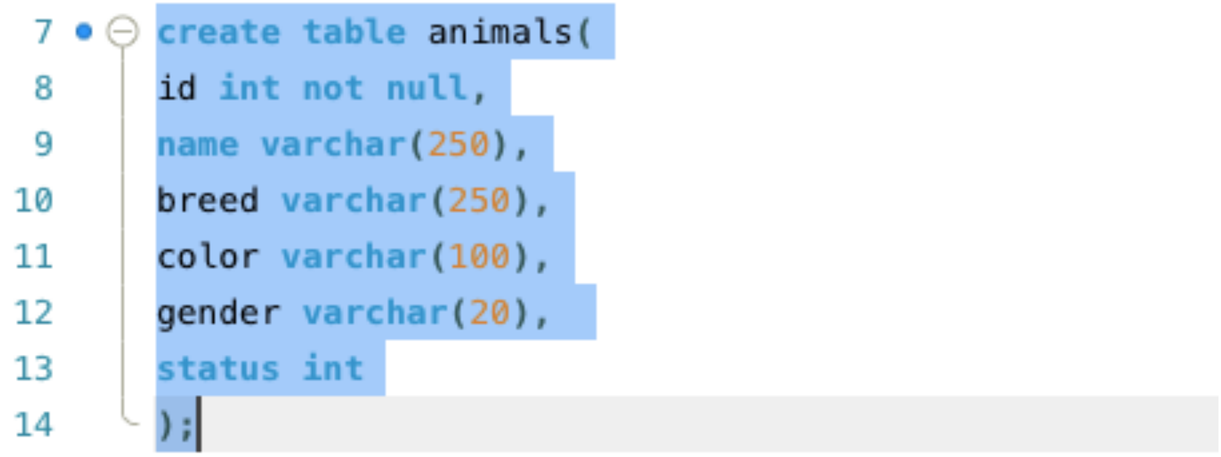
Use database pet_adoption



```
use pet_adoption;
```

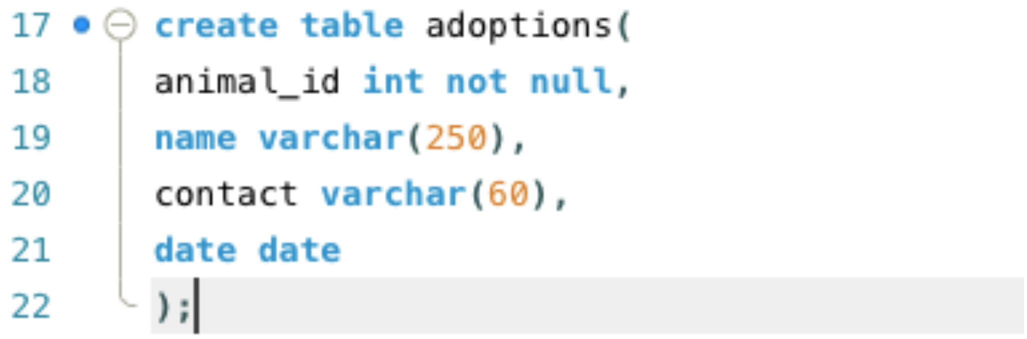
Table for animals

Create table for animals-



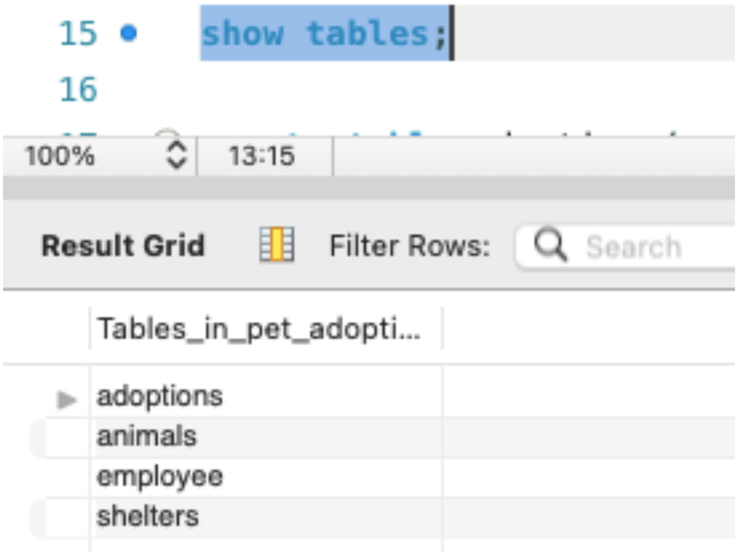
```
create table animals(
id int not null,
name varchar(250),
breed varchar(250),
color varchar(100),
gender varchar(20),
status int
);
```

Create table for adoptions-



```
create table adoptions(
animal_id int not null,
name varchar(250),
contact varchar(60),
date date
);
```

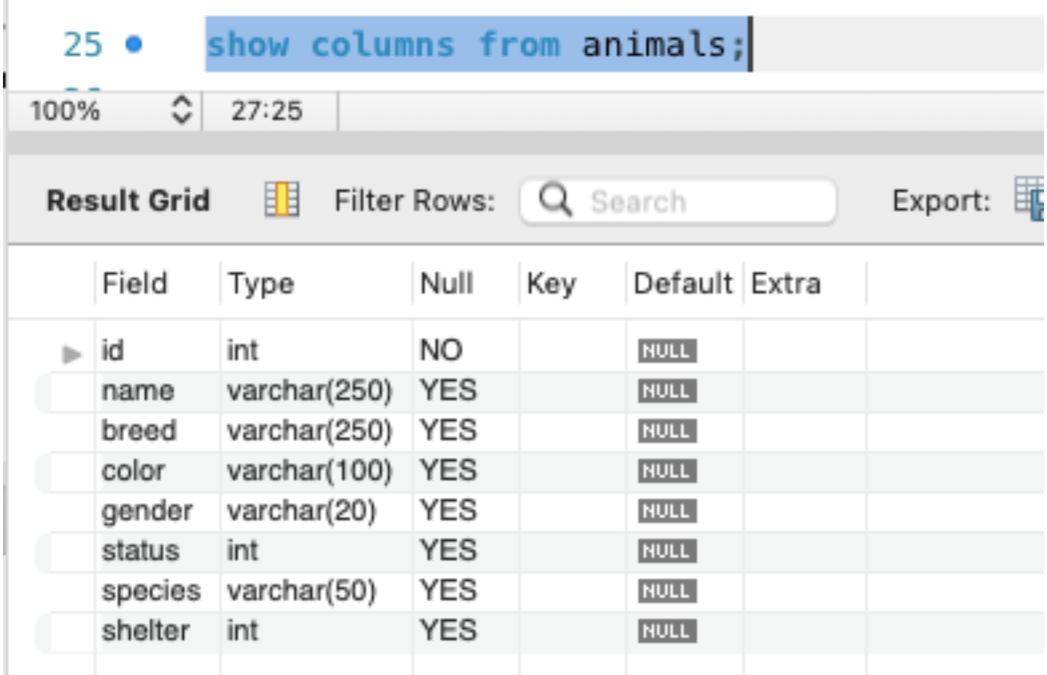
Show tables-



```
show tables;
```

Tables_in_pet_adopti...
adoptions
animals
employee
shelters

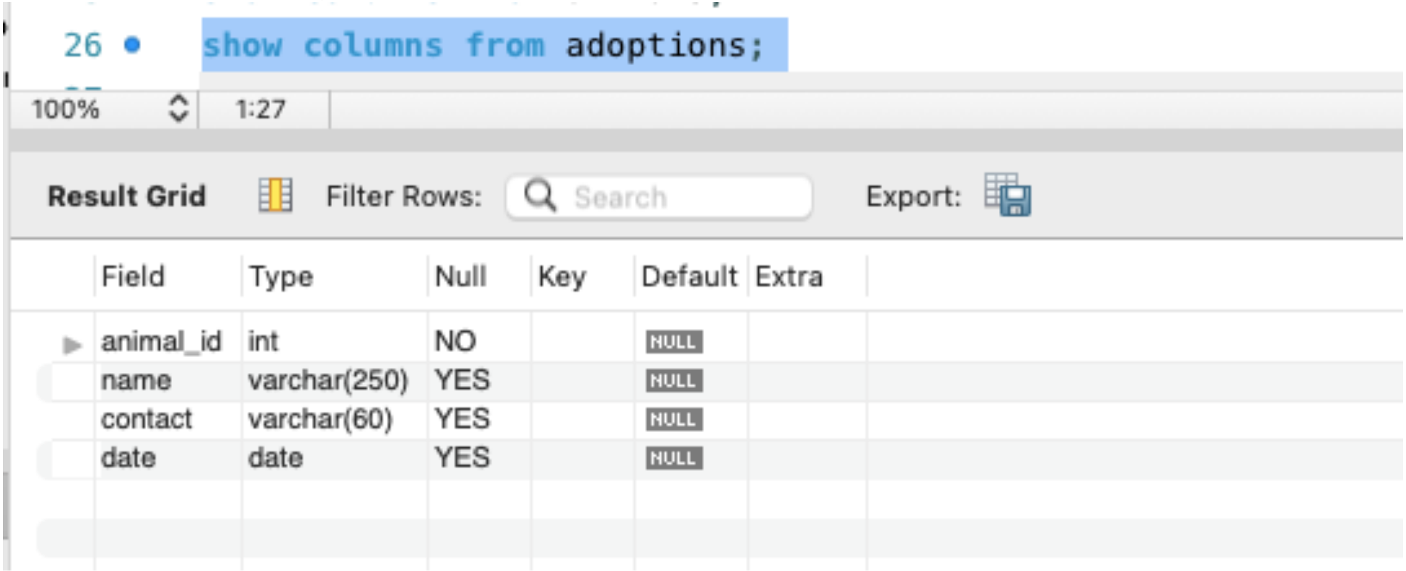
Show columns of animal table-



```
show columns from animals;
```

Field	Type	Null	Key	Default	Extra
id	int	NO		NULL	
name	varchar(250)	YES		NULL	
breed	varchar(250)	YES		NULL	
color	varchar(100)	YES		NULL	
gender	varchar(20)	YES		NULL	
status	int	YES		NULL	
species	varchar(50)	YES		NULL	
shelter	int	YES		NULL	

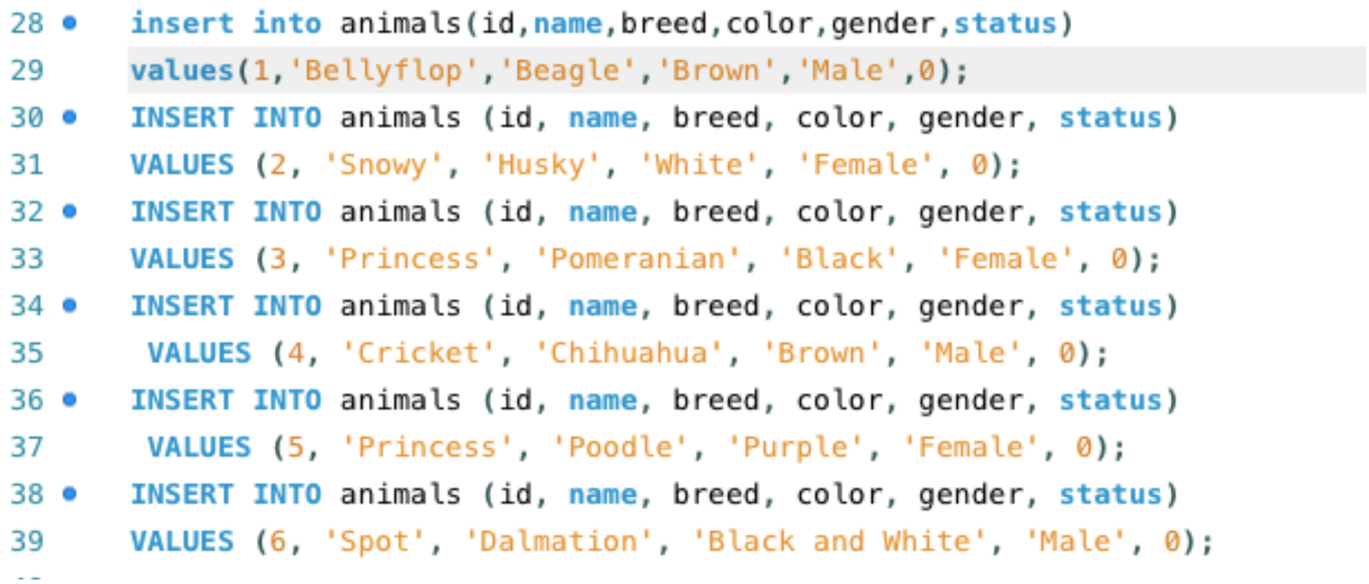
Show columns from adoptions



```
show columns from adoptions;
```

Field	Type	Null	Key	Default	Extra
animal_id	int	NO		NULL	
name	varchar(250)	YES		NULL	
contact	varchar(60)	YES		NULL	
date	date	YES		NULL	

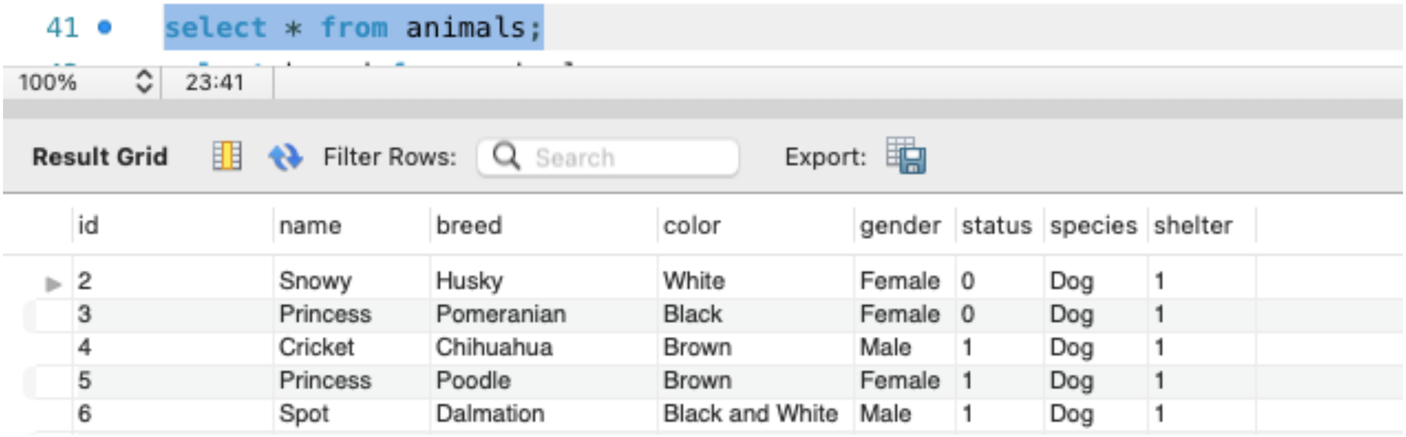
Insert values into animal table-



```
insert into animals(id,name,breed,color,gender,status)
values(1,'Bellyflop','Beagle','Brown','Male',0);
INSERT INTO animals (id, name, breed, color, gender, status)
VALUES (2, 'Snowy', 'Husky', 'White', 'Female', 0);
INSERT INTO animals (id, name, breed, color, gender, status)
VALUES (3, 'Princess', 'Pomeranian', 'Black', 'Female', 0);
INSERT INTO animals (id, name, breed, color, gender, status)
 VALUES (4, 'Cricket', 'Chihuahua', 'Brown', 'Male', 0);
INSERT INTO animals (id, name, breed, color, gender, status)
 VALUES (5, 'Princess', 'Poodle', 'Purple', 'Female', 0);
INSERT INTO animals (id, name, breed, color, gender, status)
VALUES (6, 'Spot', 'Dalmation', 'Black and White', 'Male', 0);
```

Retrieve List of Dogs: SELECT * FROM

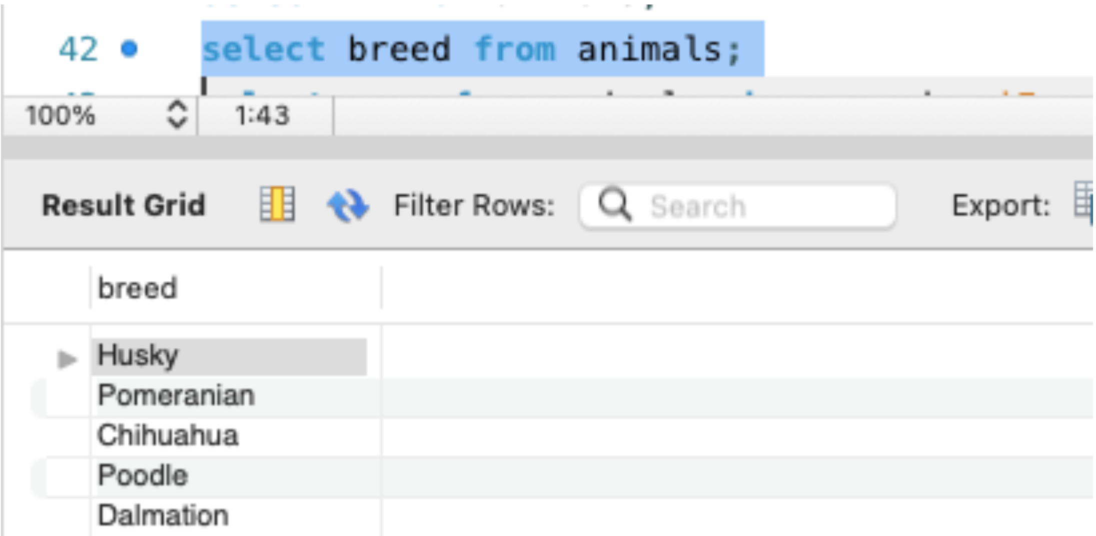
Get the full list of all properties of all dogs (defaults to a limit of 100 rows):



```
select * from animals;
```

id	name	breed	color	gender	status	species	shelter
2	Snowy	Husky	White	Female	0	Dog	1
3	Princess	Pomeranian	Black	Female	0	Dog	1
4	Cricket	Chihuahua	Brown	Male	1	Dog	1
5	Princess	Poodle	Brown	Female	1	Dog	1
6	Spot	Dalmation	Black and White	Male	1	Dog	1

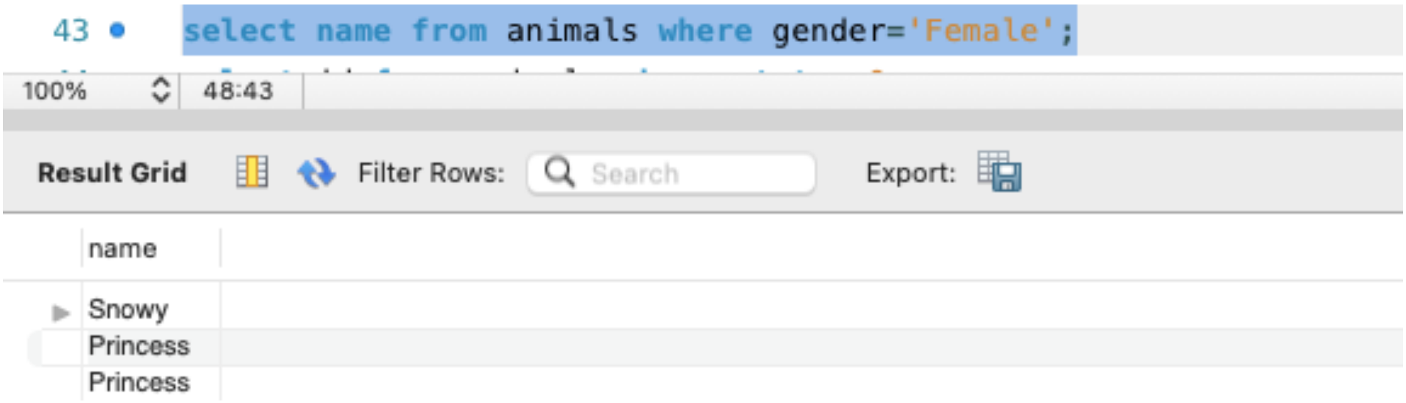
Get the breeds of all dogs:



```
select breed from animals;
```

breed
Husky
Pomeranian
Chihuahua
Poodle
Dalmation

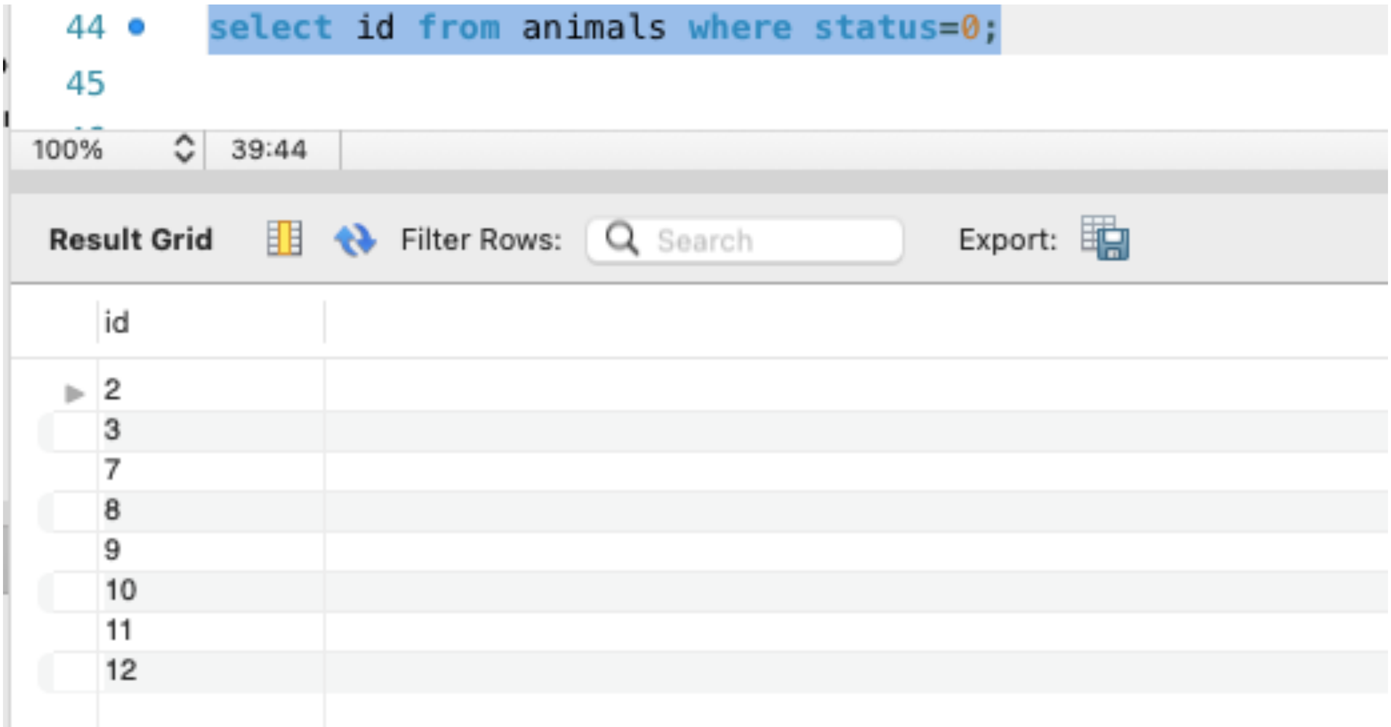
Get the names of only female dogs by including a WHERE clause:



```
select name from animals where gender='Female';
```

```
name
Snowy
Princess
Princess
```

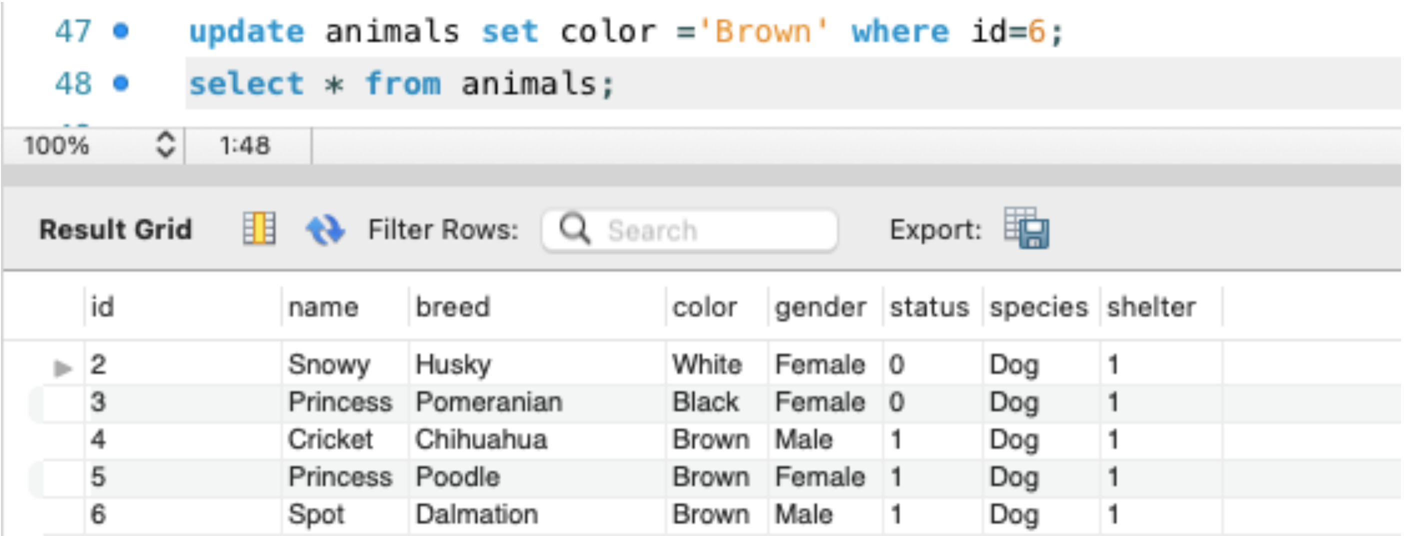
Get the IDs of dogs up for adoption:



```
select id from animals where status=0;
```

```
id
2
3
7
8
9
10
11
12
```

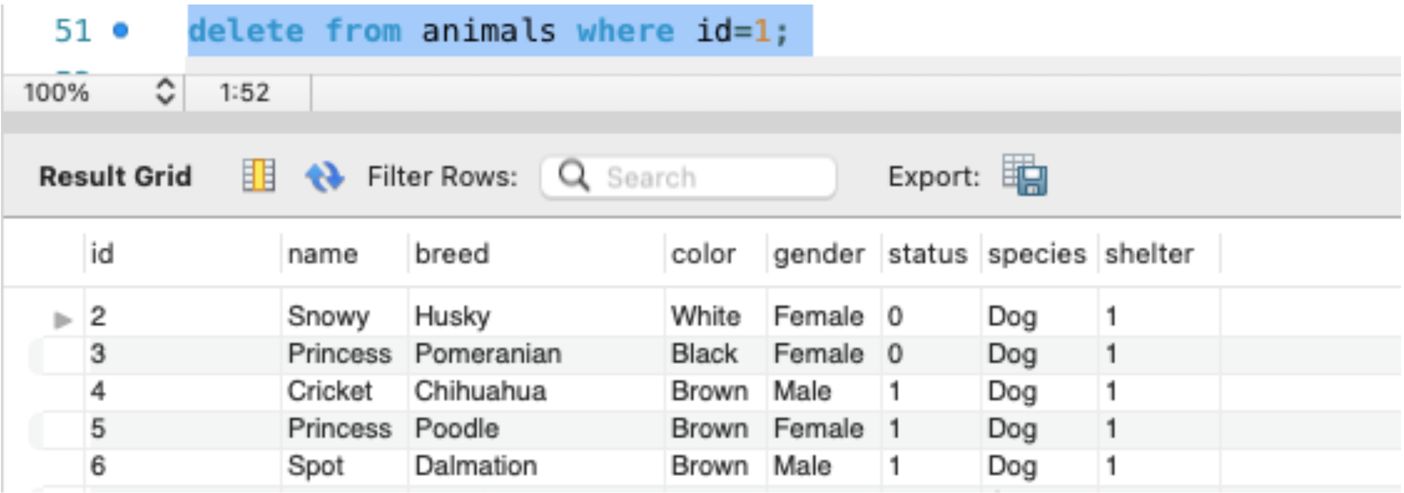
Change the color of the dog whose id is 6



```
update animals set color ='Brown' where id=6;
select * from animals;
```

id	name	breed	color	gender	status	species	shelter
2	Snowy	Husky	White	Female	0	Dog	1
3	Princess	Pomeranian	Black	Female	0	Dog	1
4	Cricket	Chihuahua	Brown	Male	1	Dog	1
5	Princess	Poodle	Brown	Female	1	Dog	1
6	Spot	Dalmation	Brown	Male	1	Dog	1

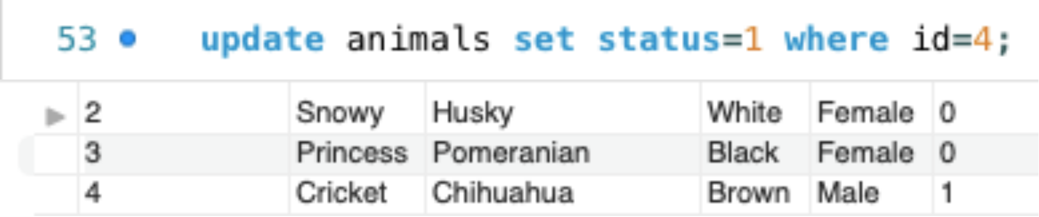
Delete the dog whose id is 1



```
delete from animals where id=1;
```

id	name	breed	color	gender	status	species	shelter
2	Snowy	Husky	White	Female	0	Dog	1
3	Princess	Pomeranian	Black	Female	0	Dog	1
4	Cricket	Chihuahua	Brown	Male	1	Dog	1
5	Princess	Poodle	Brown	Female	1	Dog	1
6	Spot	Dalmation	Brown	Male	1	Dog	1

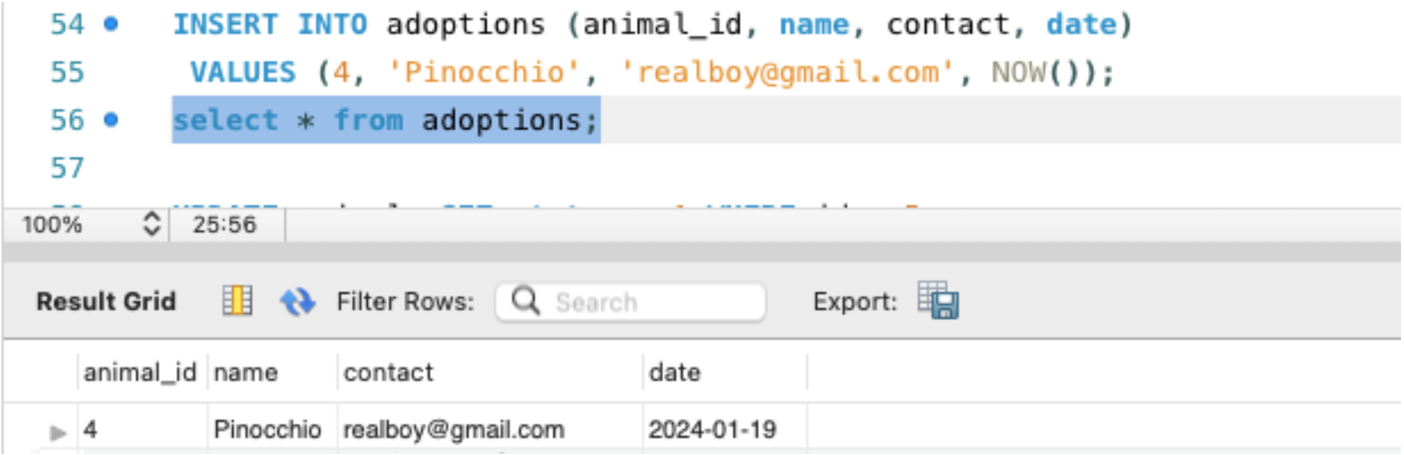
update Cricket's status to 1 for "adopted" in the animals table



```
update animals set status=1 where id=4;
```

2	Snowy	Husky	White	Female	0
3	Princess	Pomeranian	Black	Female	0
4	Cricket	Chihuahua	Brown	Male	1

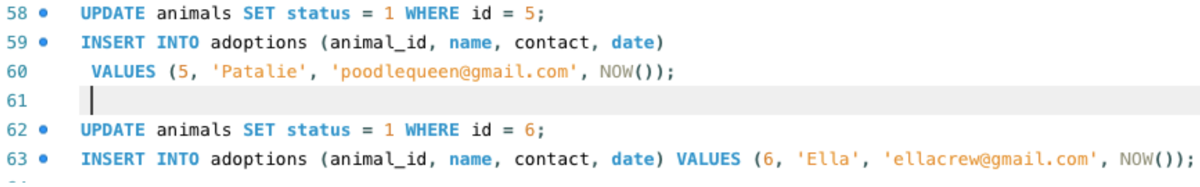
insert the event into the adoptions table



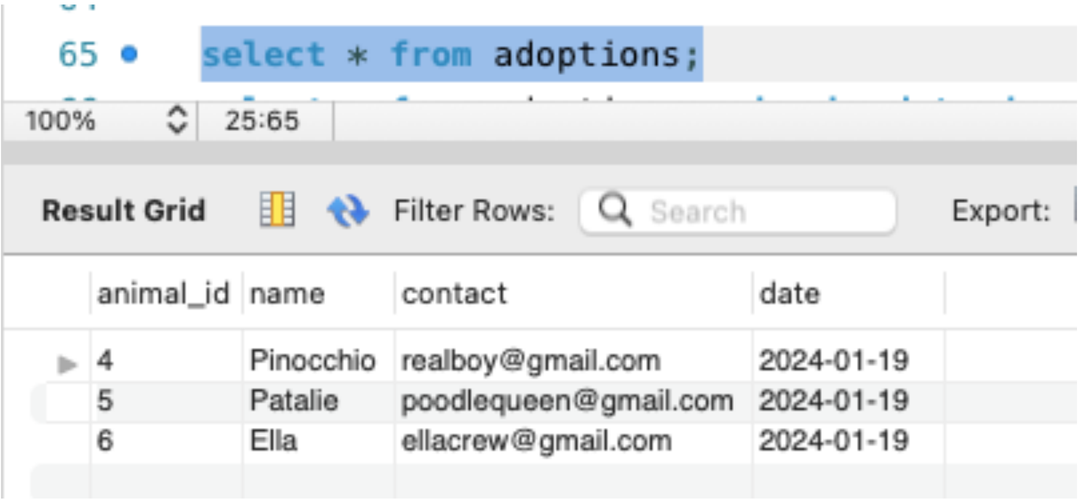
```sql
INSERT INTO adoptions (animal_id, name, contact, date)
 VALUES (4, 'Pinocchio', 'realboy@gmail.com', NOW());
select * from adoptions;
```

animal_id	name	contact	date
4	Pinocchio	realboy@gmail.com	2024-01-19

Inserting values for id=5 and id=6



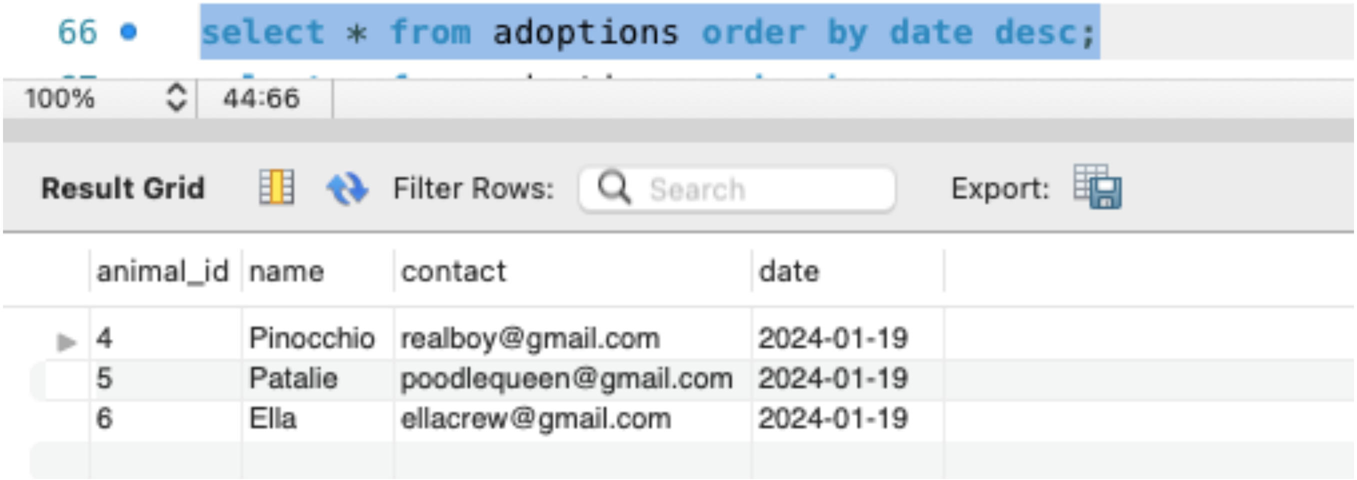
```sql
UPDATE animals SET status = 1 WHERE id = 5;
INSERT INTO adoptions (animal_id, name, contact, date)
 VALUES (5, 'Patalie', 'poodlequeen@gmail.com', NOW());

UPDATE animals SET status = 1 WHERE id = 6;
INSERT INTO adoptions (animal_id, name, contact, date) VALUES (6, 'Ella', 'ellacrew@gmail.com', NOW());
```



```sql
select * from adoptions;
```

animal_id	name	contact	date
4	Pinocchio	realboy@gmail.com	2024-01-19
5	Patalie	poodlequeen@gmail.com	2024-01-19
6	Ella	ellacrew@gmail.com	2024-01-19

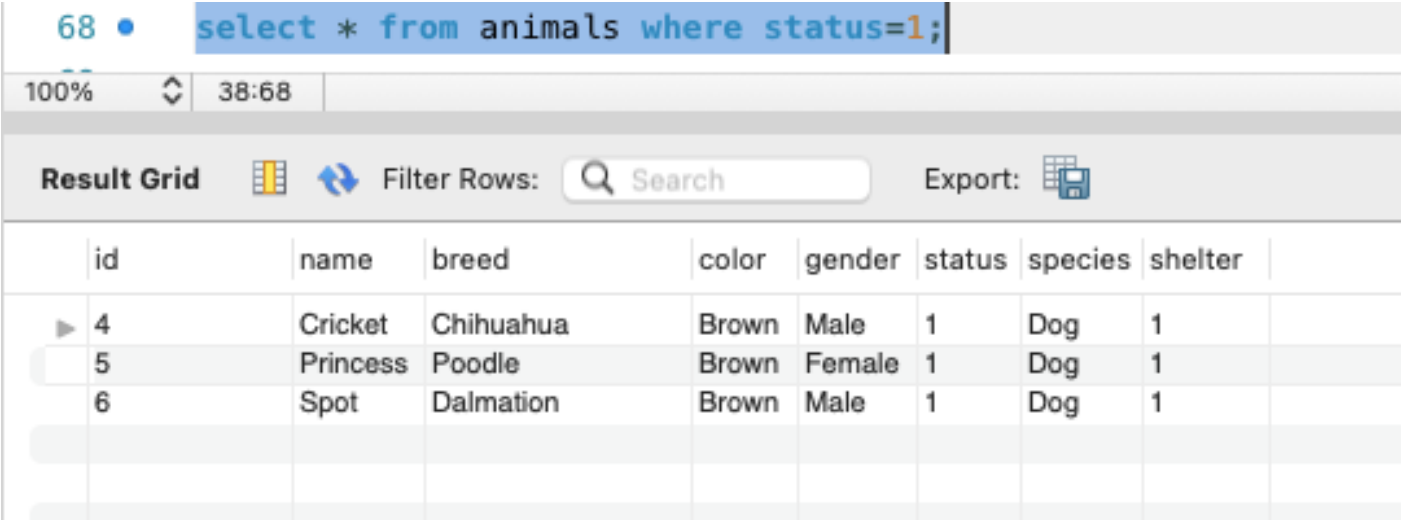
Use order by in adoptions table to show the output in descending order



```sql
select * from adoptions order by date desc;
```

animal_id	name	contact	date
4	Pinocchio	realboy@gmail.com	2024-01-19
5	Patalie	poodlequeen@gmail.com	2024-01-19
6	Ella	ellacrew@gmail.com	2024-01-19

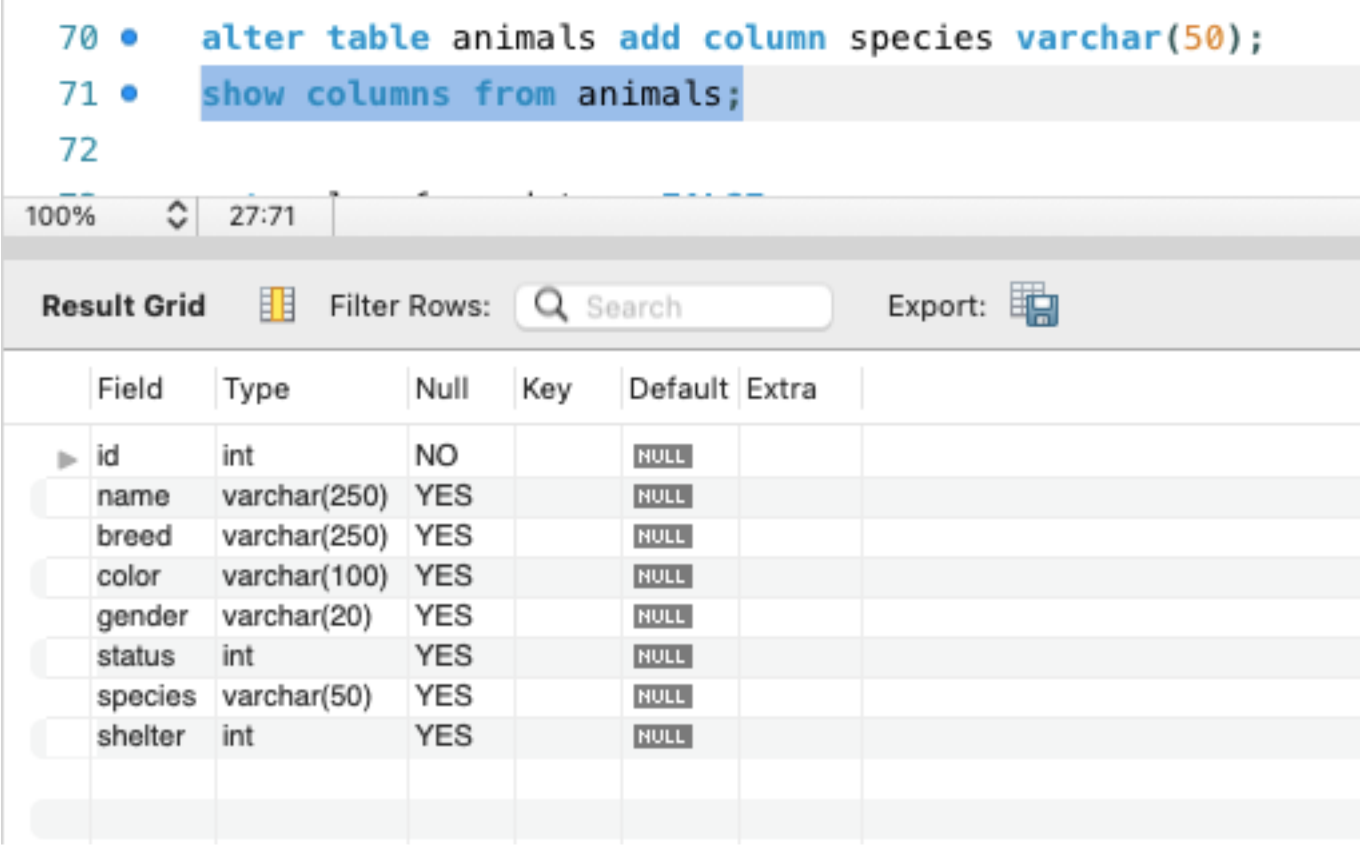
Show all the animals having status=1



```
select * from animals where status=1;
```

id	name	breed	color	gender	status	species	shelter
4	Cricket	Chihuahua	Brown	Male	1	Dog	1
5	Princess	Poodle	Brown	Female	1	Dog	1
6	Spot	Dalmation	Brown	Male	1	Dog	1

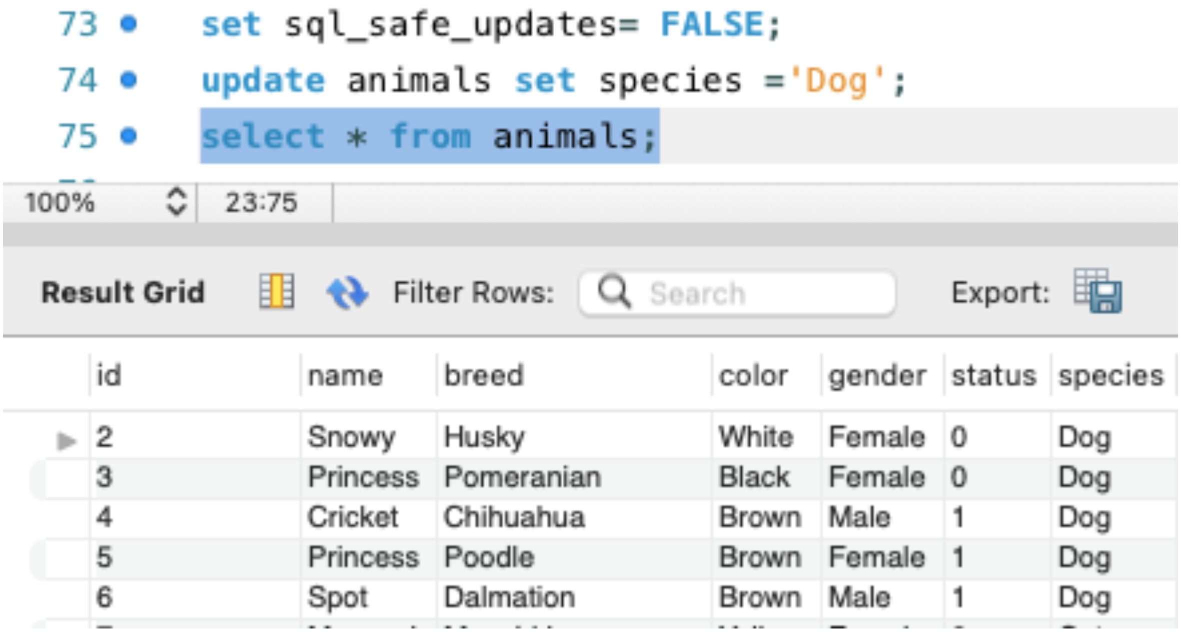
Add a new column species in animal table



```
alter table animals add column species varchar(50);
show columns from animals;
```

Field	Type	Null	Key	Default	Extra
id	int	NO		NULL	
name	varchar(250)	YES		NULL	
breed	varchar(250)	YES		NULL	
color	varchar(100)	YES		NULL	
gender	varchar(20)	YES		NULL	
status	int	YES		NULL	
species	varchar(50)	YES		NULL	
shelter	int	YES		NULL	

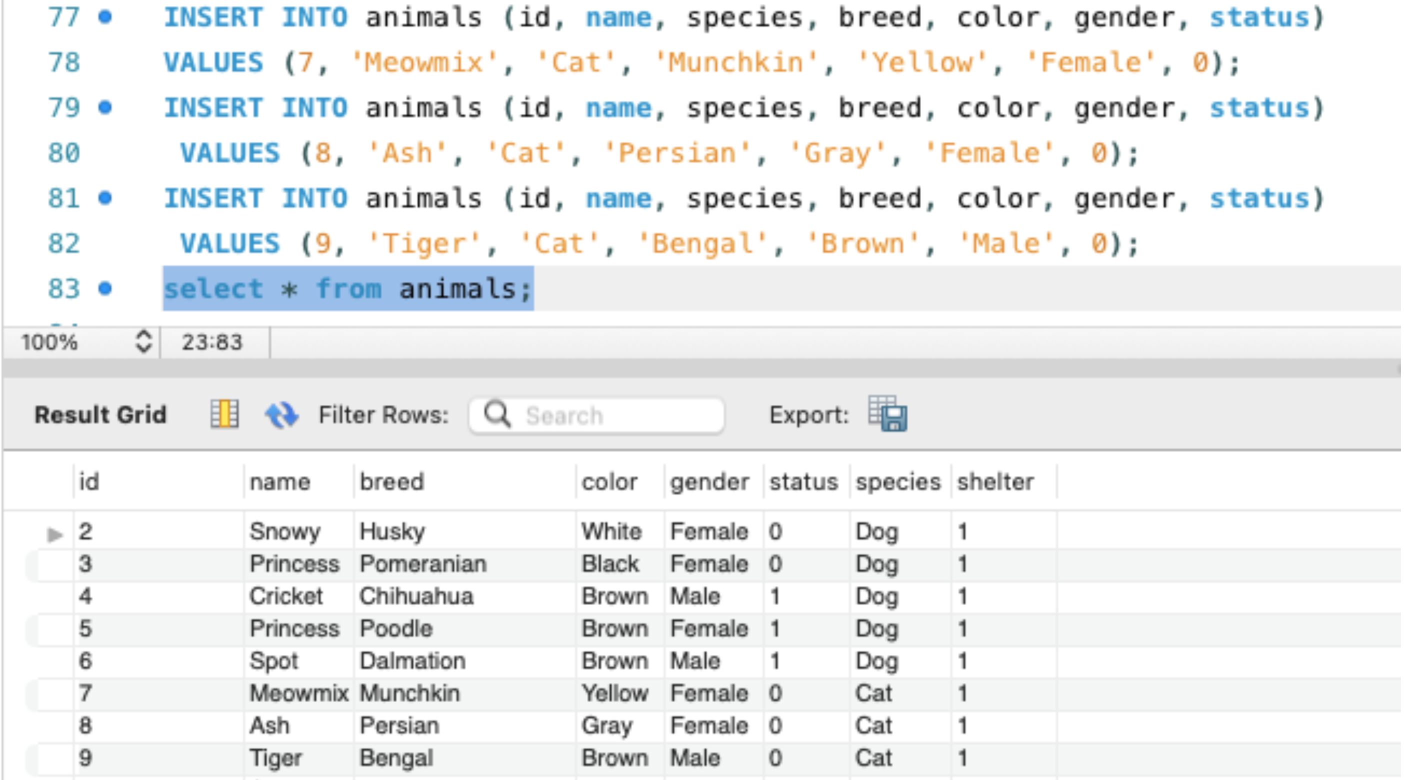
Update all the current animals species to dog



```
set sql_safe_updates= FALSE;
update animals set species ='Dog';
select * from animals;
```

id	name	breed	color	gender	status	species
2	Snowy	Husky	White	Female	0	Dog
3	Princess	Pomeranian	Black	Female	0	Dog
4	Cricket	Chihuahua	Brown	Male	1	Dog
5	Princess	Poodle	Brown	Female	1	Dog
6	Spot	Dalmation	Brown	Male	1	Dog

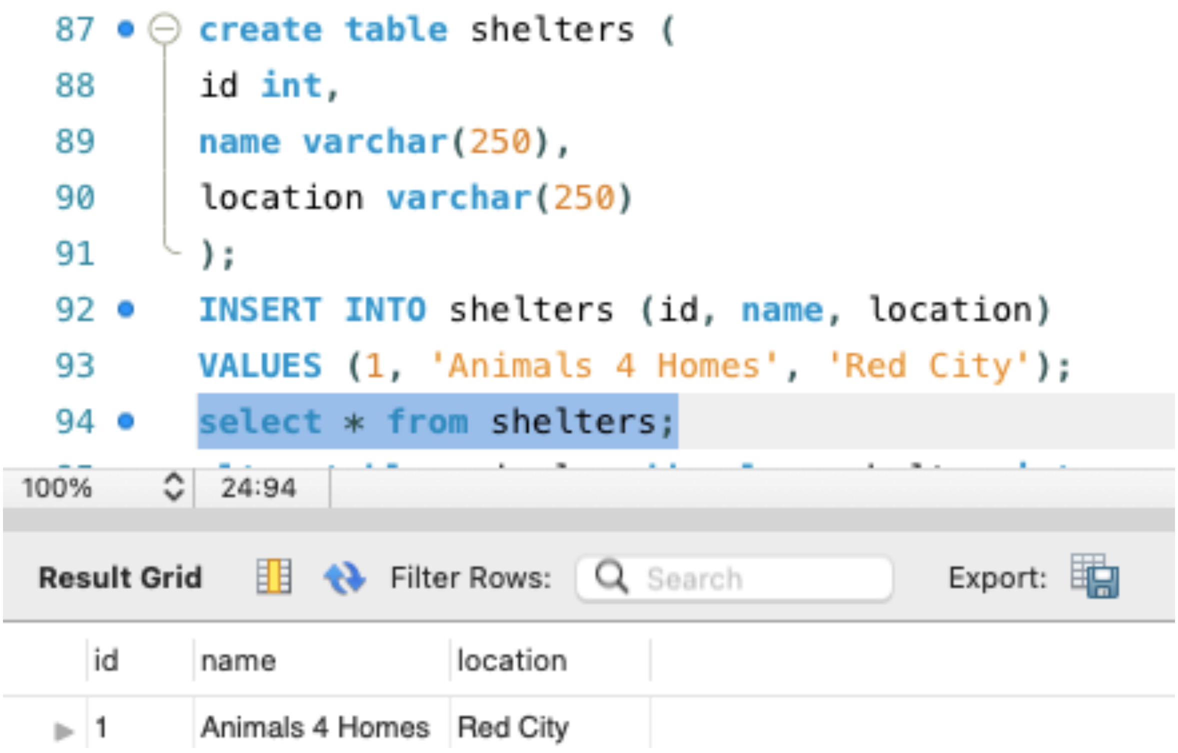
Add some cats in animals table



```
INSERT INTO animals (id, name, species, breed, color, gender, status)
VALUES (7, 'Meowmix', 'Cat', 'Munchkin', 'Yellow', 'Female', 0);
INSERT INTO animals (id, name, species, breed, color, gender, status)
 VALUES (8, 'Ash', 'Cat', 'Persian', 'Gray', 'Female', 0);
INSERT INTO animals (id, name, species, breed, color, gender, status)
 VALUES (9, 'Tiger', 'Cat', 'Bengal', 'Brown', 'Male', 0);
select * from animals;
```

id	name	breed	color	gender	status	species	shelter
2	Snowy	Husky	White	Female	0	Dog	1
3	Princess	Pomeranian	Black	Female	0	Dog	1
4	Cricket	Chihuahua	Brown	Male	1	Dog	1
5	Princess	Poodle	Brown	Female	1	Dog	1
6	Spot	Dalmation	Brown	Male	1	Dog	1
7	Meowmix	Munchkin	Yellow	Female	0	Cat	1
8	Ash	Persian	Gray	Female	0	Cat	1
9	Tiger	Bengal	Brown	Male	0	Cat	1

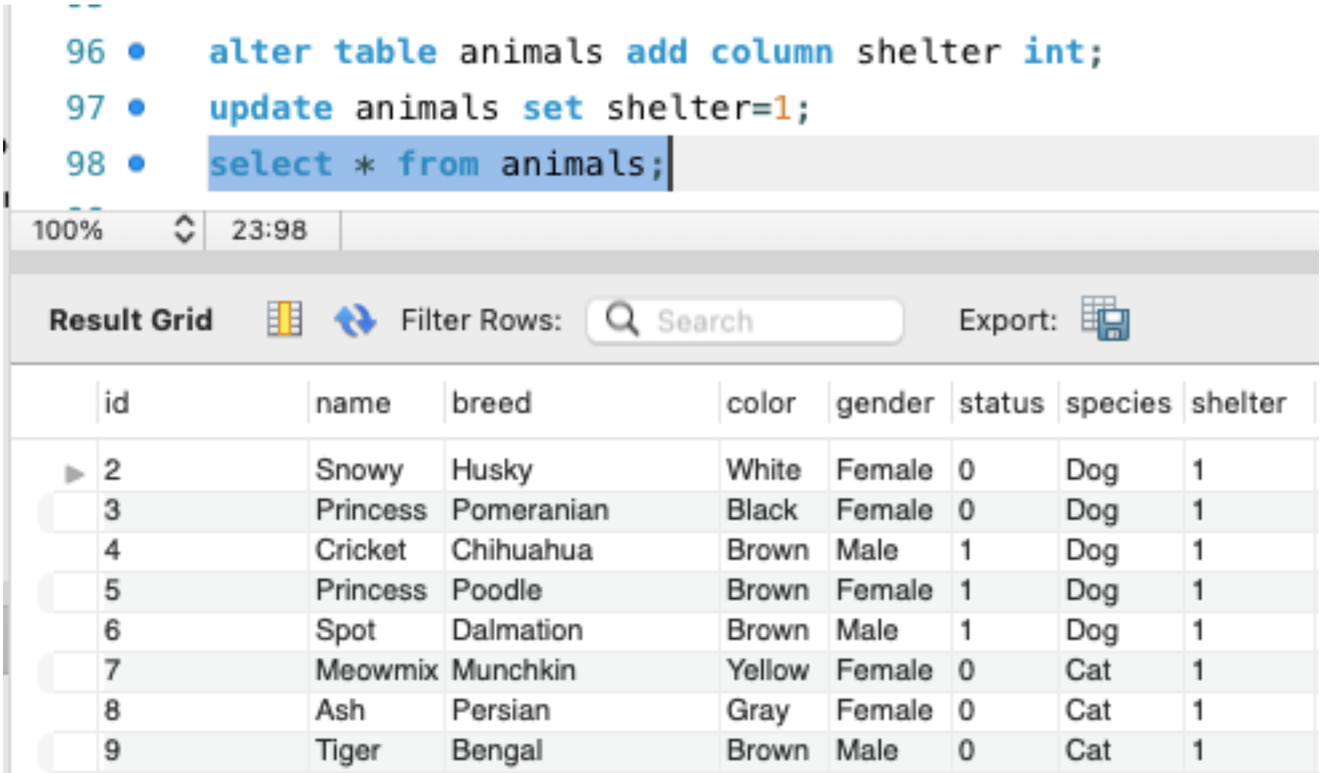
Create a table shelter and add a place with id=1



```sql
create table shelters (
id int,
name varchar(250),
location varchar(250)
);
INSERT INTO shelters (id, name, location)
VALUES (1, 'Animals 4 Homes', 'Red City');
select * from shelters;
```

id	name	location
1	Animals 4 Homes	Red City

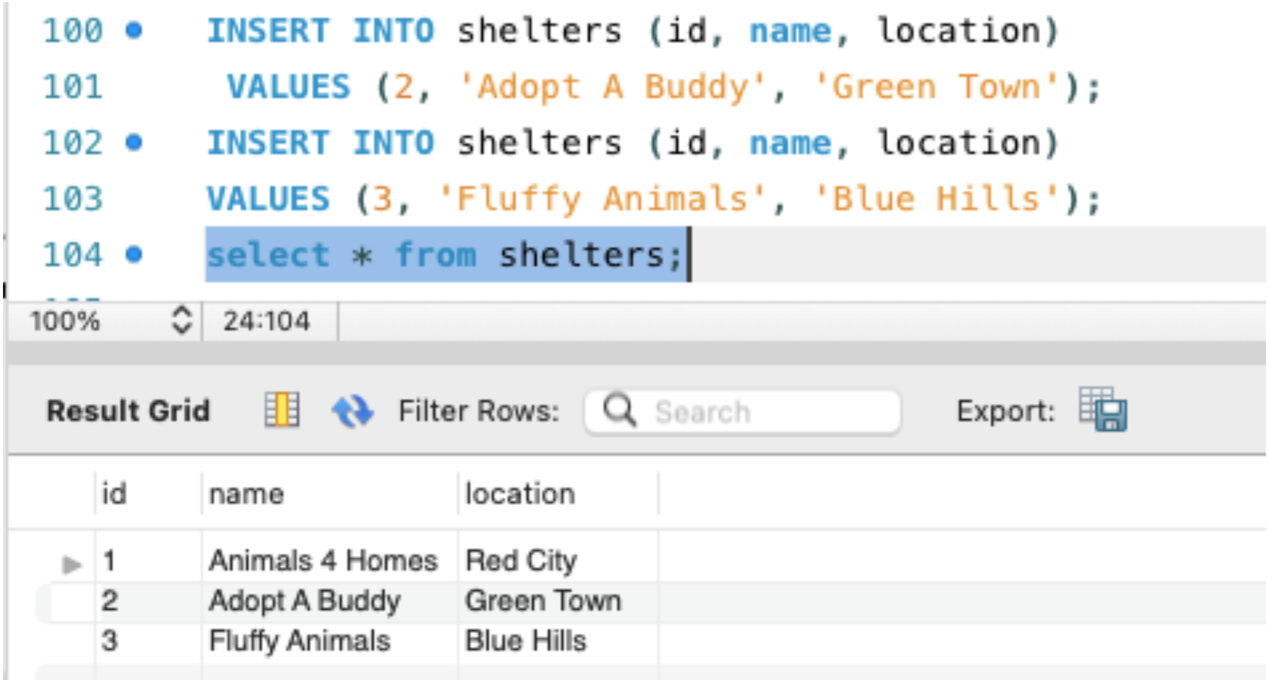
Create new shelter column and update all of the existing rows of animal data to be located in this shelter id = 1.



```sql
alter table animals add column shelter int;
update animals set shelter=1;
select * from animals;
```

id	name	breed	color	gender	status	species	shelter
2	Snowy	Husky	White	Female	0	Dog	1
3	Princess	Pomeranian	Black	Female	0	Dog	1
4	Cricket	Chihuahua	Brown	Male	1	Dog	1
5	Princess	Poodle	Brown	Female	1	Dog	1
6	Spot	Dalmation	Brown	Male	1	Dog	1
7	Meowmix	Munchkin	Yellow	Female	0	Cat	1
8	Ash	Persian	Gray	Female	0	Cat	1
9	Tiger	Bengal	Brown	Male	0	Cat	1

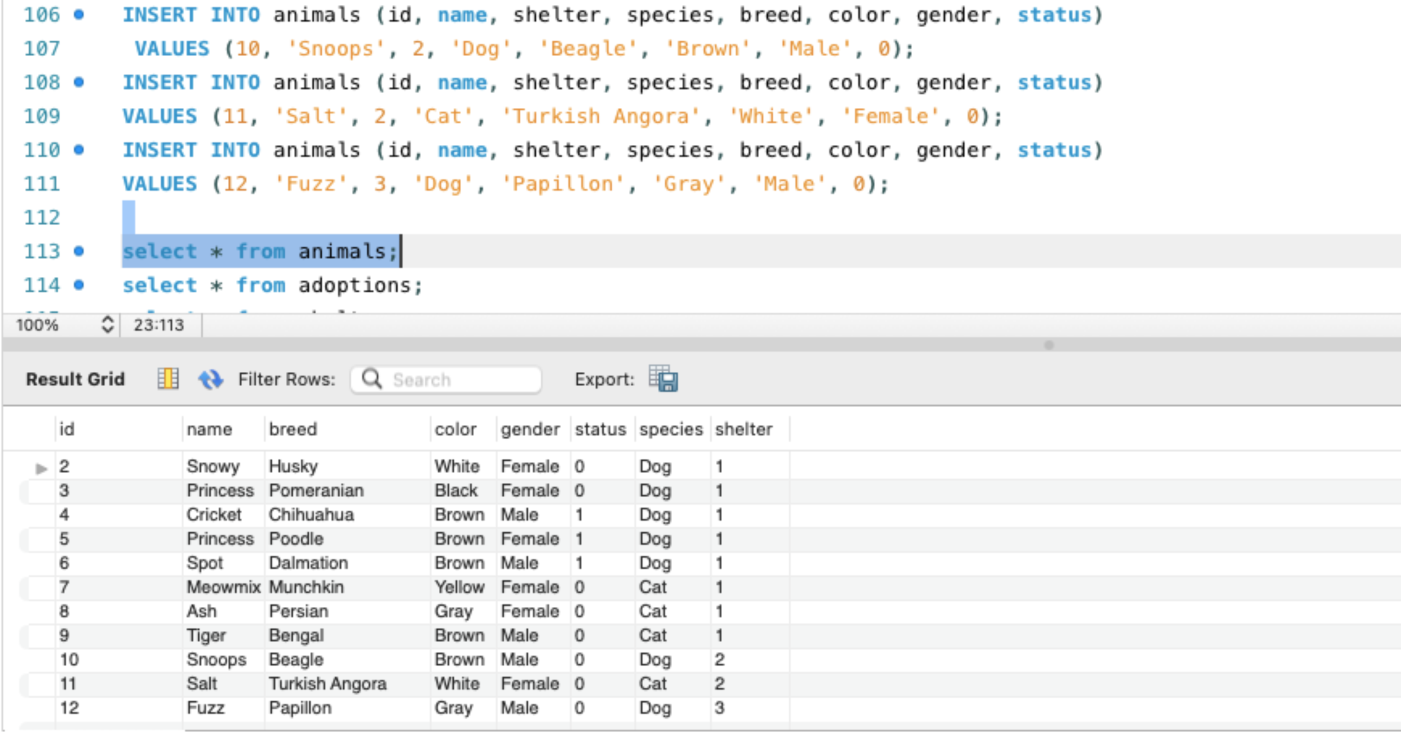
create entries for the other two shelters joining our efforts for pet adoption: Adopt A Buddy and Fluffy animals.



```sql
INSERT INTO shelters (id, name, location)
 VALUES (2, 'Adopt A Buddy', 'Green Town');
INSERT INTO shelters (id, name, location)
VALUES (3, 'Fluffy Animals', 'Blue Hills');
select * from shelters;
```

id	name	location
1	Animals 4 Homes	Red City
2	Adopt A Buddy	Green Town
3	Fluffy Animals	Blue Hills

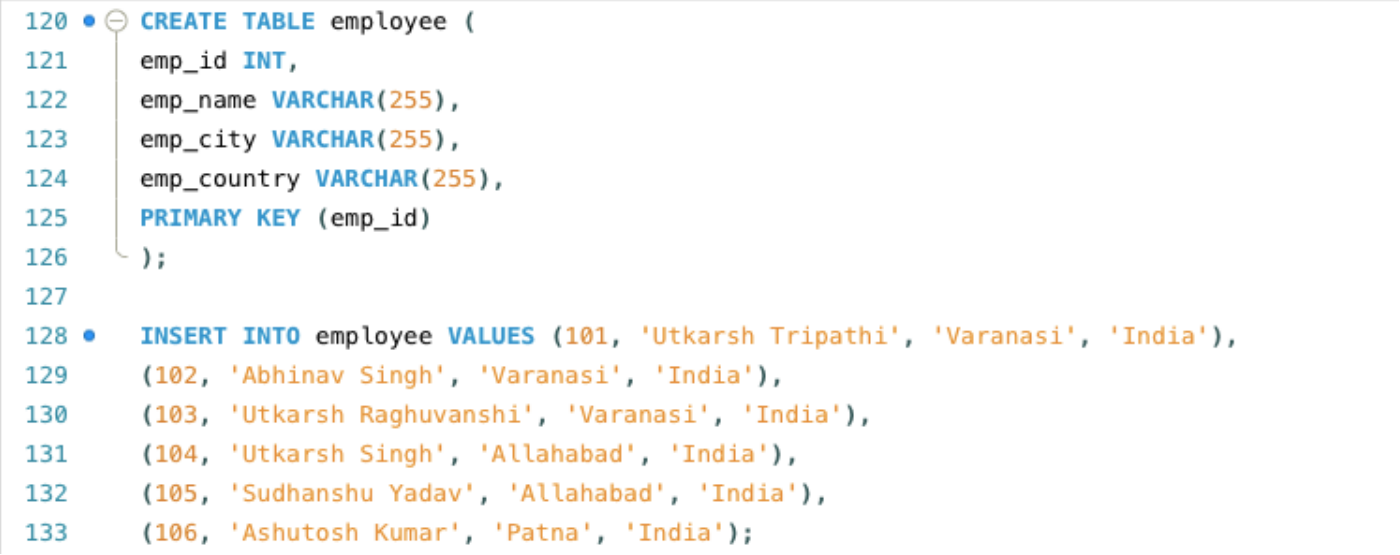
Add some more animals assign them shelter id =2,3



```sql
INSERT INTO animals (id, name, shelter, species, breed, color, gender, status)
 VALUES (10, 'Snoops', 2, 'Dog', 'Beagle', 'Brown', 'Male', 0);
INSERT INTO animals (id, name, shelter, species, breed, color, gender, status)
VALUES (11, 'Salt', 2, 'Cat', 'Turkish Angora', 'White', 'Female', 0);
INSERT INTO animals (id, name, shelter, species, breed, color, gender, status)
VALUES (12, 'Fuzz', 3, 'Dog', 'Papillon', 'Gray', 'Male', 0);

select * from animals;
select * from adoptions;
```

id	name	breed	color	gender	status	species	shelter
2	Snowy	Husky	White	Female	0	Dog	1
3	Princess	Pomeranian	Black	Female	0	Dog	1
4	Cricket	Chihuahua	Brown	Male	1	Dog	1
5	Princess	Poodle	Brown	Female	1	Dog	1
6	Spot	Dalmation	Brown	Male	1	Dog	1
7	Meowmix	Munchkin	Yellow	Female	0	Cat	1
8	Ash	Persian	Gray	Female	0	Cat	1
9	Tiger	Bengal	Brown	Male	0	Cat	1
10	Snoops	Beagle	Brown	Male	0	Dog	2
11	Salt	Turkish Angora	White	Female	0	Cat	2
12	Fuzz	Papillon	Gray	Male	0	Dog	3

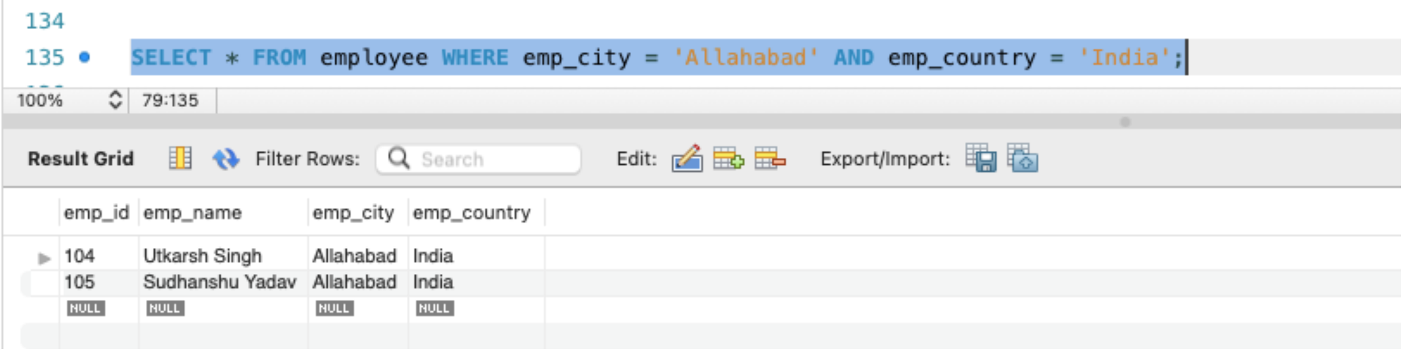
Create a table employee and add some entries to it.



```sql
CREATE TABLE employee (
emp_id INT,
emp_name VARCHAR(255),
emp_city VARCHAR(255),
emp_country VARCHAR(255),
PRIMARY KEY (emp_id)
);

INSERT INTO employee VALUES (101, 'Utkarsh Tripathi', 'Varanasi', 'India'),
(102, 'Abhinav Singh', 'Varanasi', 'India'),
(103, 'Utkarsh Raghuvanshi', 'Varanasi', 'India'),
(104, 'Utkarsh Singh', 'Allahabad', 'India'),
(105, 'Sudhanshu Yadav', 'Allahabad', 'India'),
(106, 'Ashutosh Kumar', 'Patna', 'India');
```

AND Operator

The AND operator is used to combines two or more conditions but if it is true when all the conditions are satisfied.

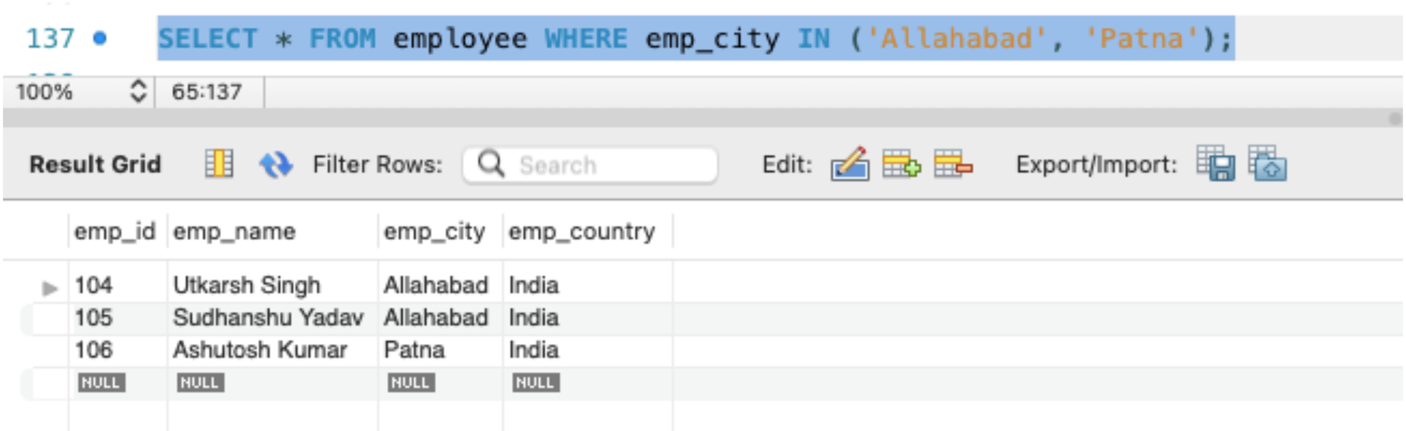


```sql
SELECT * FROM employee WHERE emp_city = 'Allahabad' AND emp_country = 'India';
```

emp_id	emp_name	emp_city	emp_country
104	Utkarsh Singh	Allahabad	India
105	Sudhanshu Yadav	Allahabad	India
NULL	NULL	NULL	NULL

IN Operator

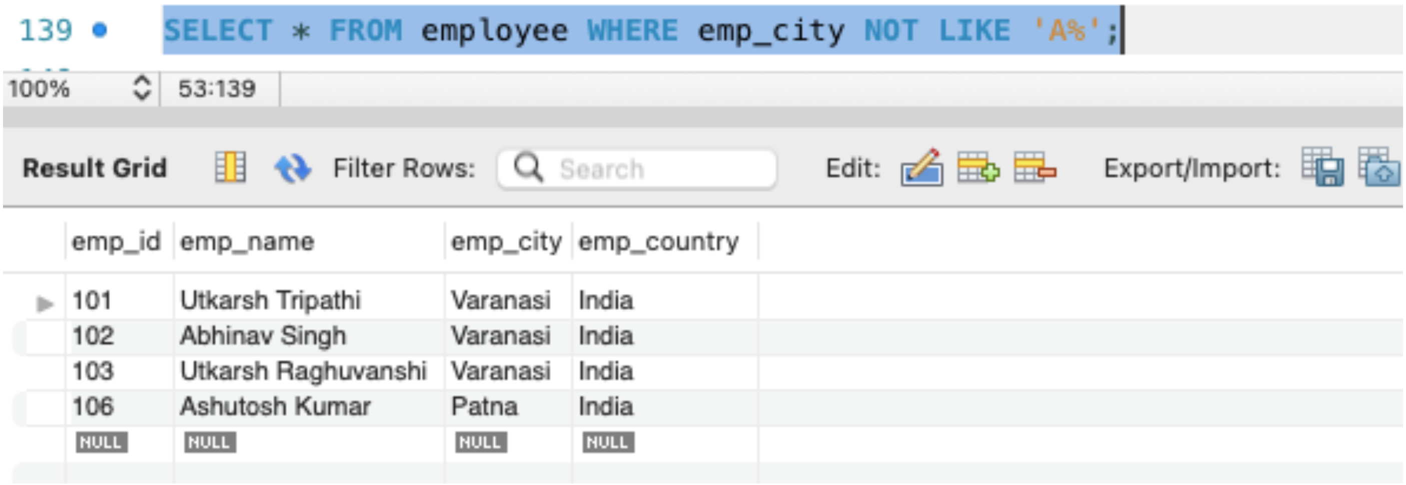
It is used to remove the multiple OR conditions in SELECT, INSERT, UPDATE, or DELETE. and We can also use NOT IN to minimize the rows in your list and any kind of duplicate entry will be retained.



```sql
SELECT * FROM employee WHERE emp_city IN ('Allahabad', 'Patna');
```

emp_id	emp_name	emp_city	emp_country
104	Utkarsh Singh	Allahabad	India
105	Sudhanshu Yadav	Allahabad	India
106	Ashutosh Kumar	Patna	India
NULL	NULL	NULL	NULL

NOT Operator

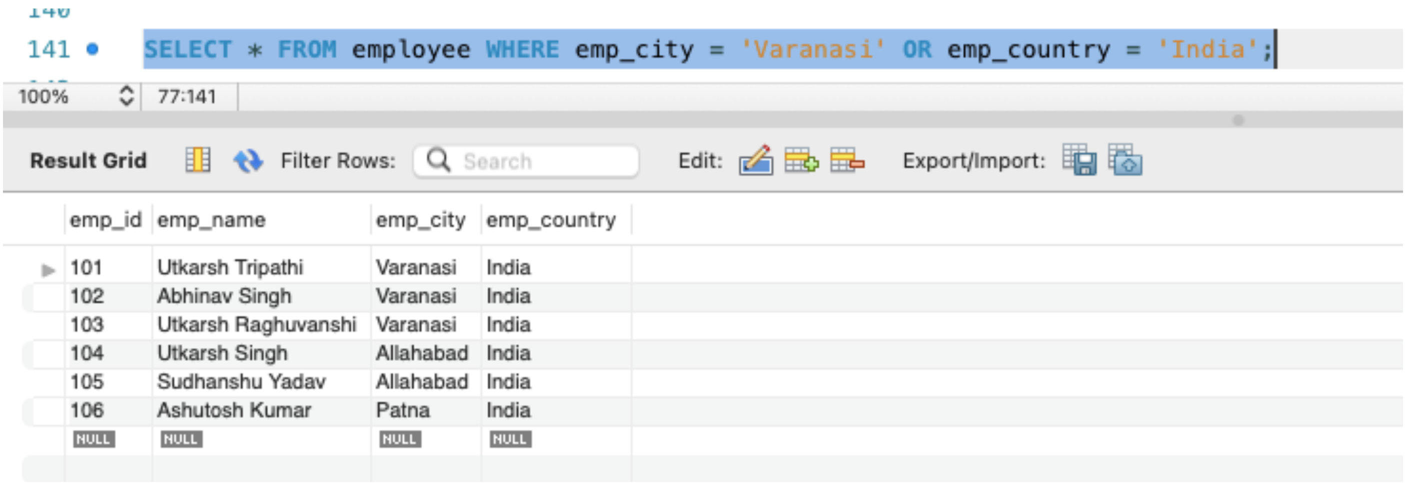


```sql
SELECT * FROM employee WHERE emp_city NOT LIKE 'A%';
```

emp_id	emp_name	emp_city	emp_country
101	Utkarsh Tripathi	Varanasi	India
102	Abhinav Singh	Varanasi	India
103	Utkarsh Raghuvanshi	Varanasi	India
106	Ashutosh Kumar	Patna	India
NULL	NULL	NULL	NULL

OR Operator

The OR operator is used to combines two or more conditions but if it is true when one of the conditions are satisfied.

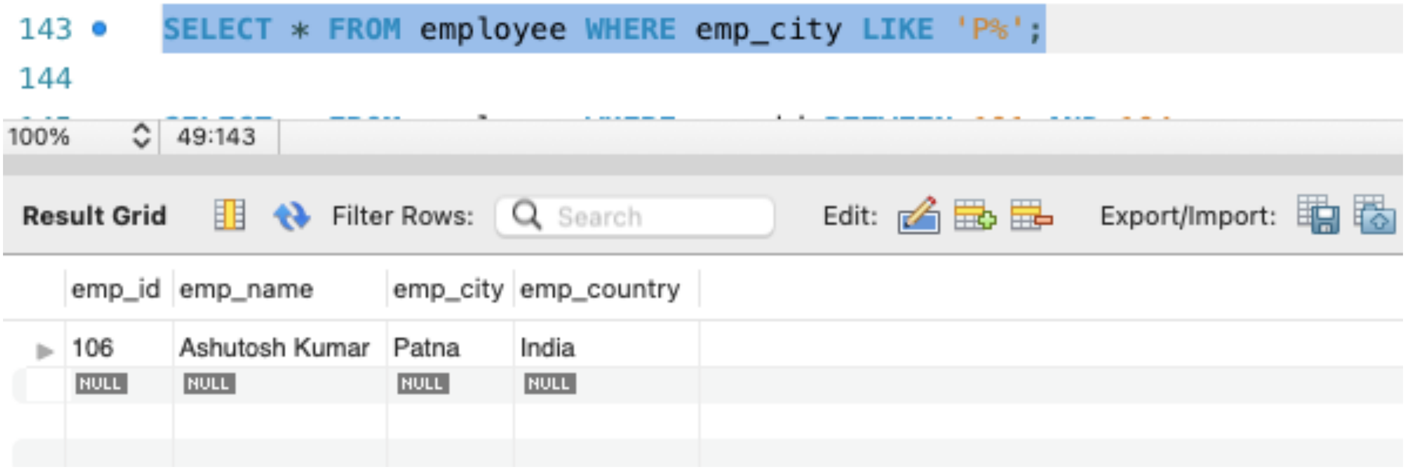


```sql
SELECT * FROM employee WHERE emp_city = 'Varanasi' OR emp_country = 'India';
```

emp_id	emp_name	emp_city	emp_country
101	Utkarsh Tripathi	Varanasi	India
102	Abhinav Singh	Varanasi	India
103	Utkarsh Raghuvanshi	Varanasi	India
104	Utkarsh Singh	Allahabad	India
105	Sudhanshu Yadav	Allahabad	India
106	Ashutosh Kumar	Patna	India
NULL	NULL	NULL	NULL

LIKE Operator

In SQL, the LIKE operator is used in the WHERE clause to search for a specified pattern in a column.

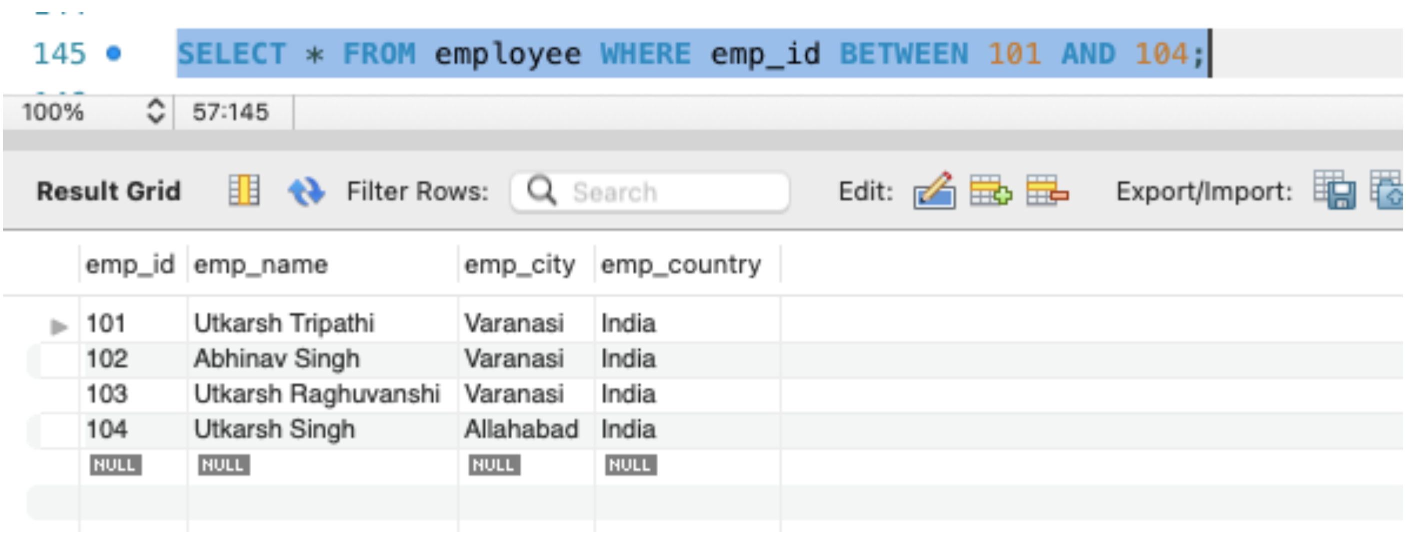


```sql
SELECT * FROM employee WHERE emp_city LIKE 'P%';
```

emp_id	emp_name	emp_city	emp_country
106	Ashutosh Kumar	Patna	India
NULL	NULL	NULL	NULL

BETWEEN Operator

The SQL **BETWEEN** condition allows you to easily test if an expression is within a range of values (inclusive).

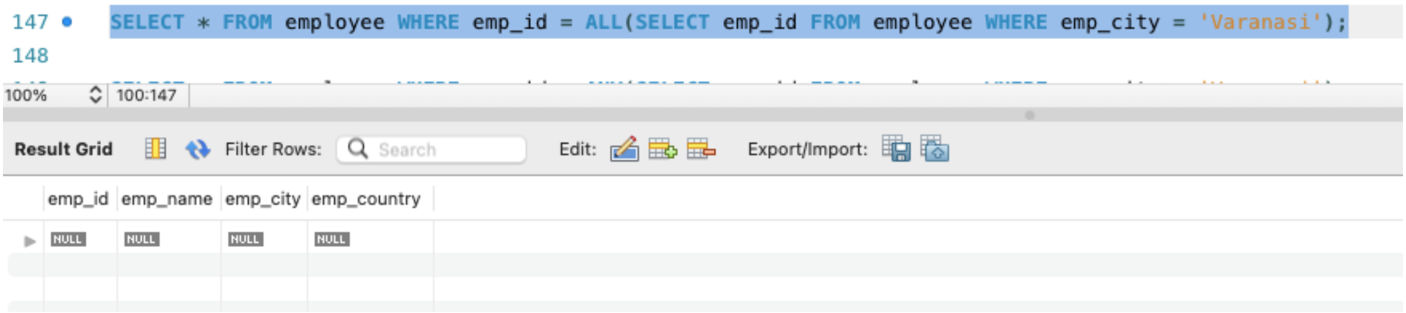


```
SELECT * FROM employee WHERE emp_id BETWEEN 101 AND 104;
```

emp_id	emp_name	emp_city	emp_country
101	Utkarsh Tripathi	Varanasi	India
102	Abhinav Singh	Varanasi	India
103	Utkarsh Raghuvanshi	Varanasi	India
104	Utkarsh Singh	Allahabad	India
NULL	NULL	NULL	NULL

ALL Operator

The ALL operator returns TRUE if all of the subqueries values matches the condition.
All operator is used with SELECT, WHERE, HAVING statement.



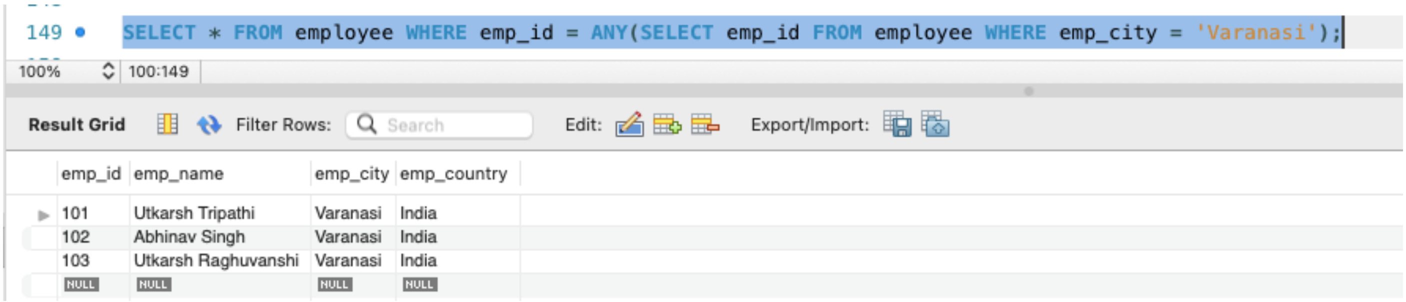
```
SELECT * FROM employee WHERE emp_id = ALL(SELECT emp_id FROM employee WHERE emp_city = 'Varanasi');
```

emp_id	emp_name	emp_city	emp_country
NULL	NULL	NULL	NULL

ANY Operator

The ANY operator:

- It returns a boolean value as a result
- It returns TRUE if ANY of the subquery values match the condition

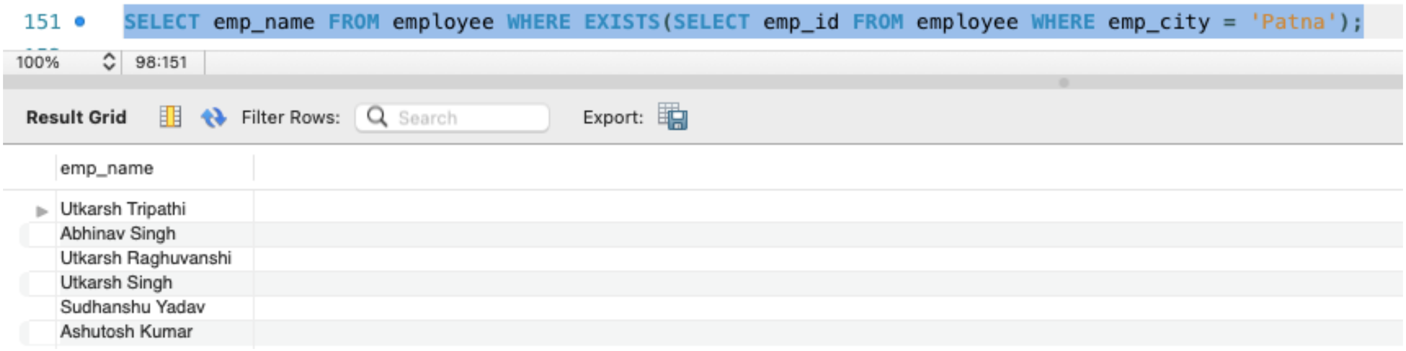


```
SELECT * FROM employee WHERE emp_id = ANY(SELECT emp_id FROM employee WHERE emp_city = 'Varanasi');
```

emp_id	emp_name	emp_city	emp_country
101	Utkarsh Tripathi	Varanasi	India
102	Abhinav Singh	Varanasi	India
103	Utkarsh Raghuvanshi	Varanasi	India
NULL	NULL	NULL	NULL

EXISTS Operator

In SQL,Exists operator is used to check whether the result of a correlated nested query is empty or not. Exists operator is used with SELECT, UPDATE, INSERT or DELETE statement.

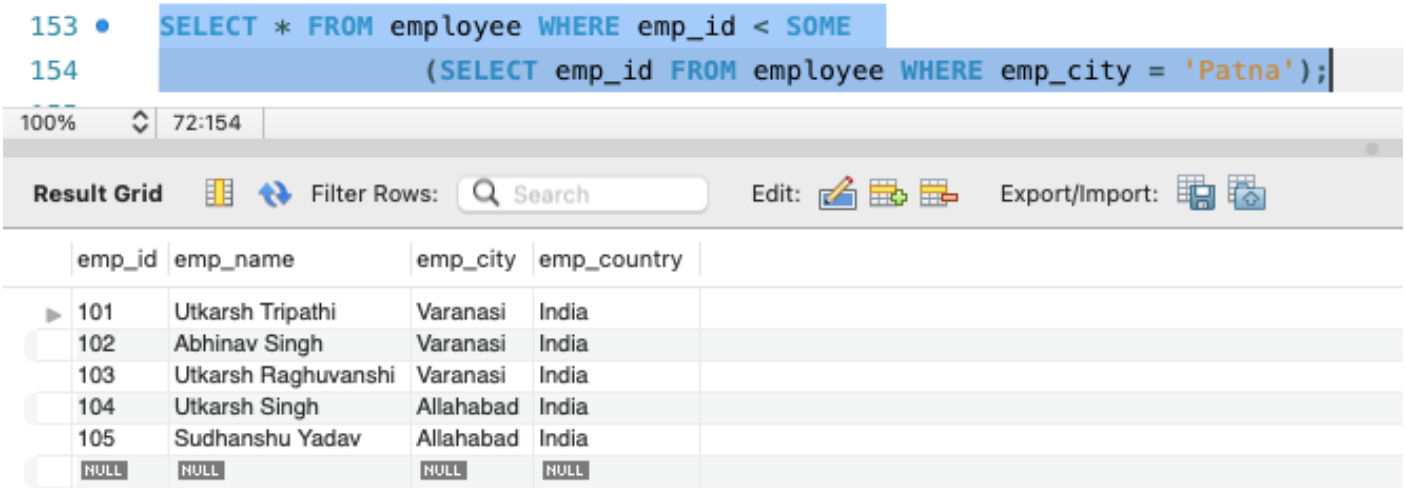


```sql
SELECT emp_name FROM employee WHERE EXISTS(SELECT emp_id FROM employee WHERE emp_city = 'Patna');
```

emp_name
Utkarsh Tripathi
Abhinav Singh
Utkarsh Raghuvanshi
Utkarsh Singh
Sudhanshu Yadav
Ashutosh Kumar

SOME Operator

In SQL, SOME operators are issued with comparison operators (<,>,=,<=, etc) to compare the value with the result of a subquery.

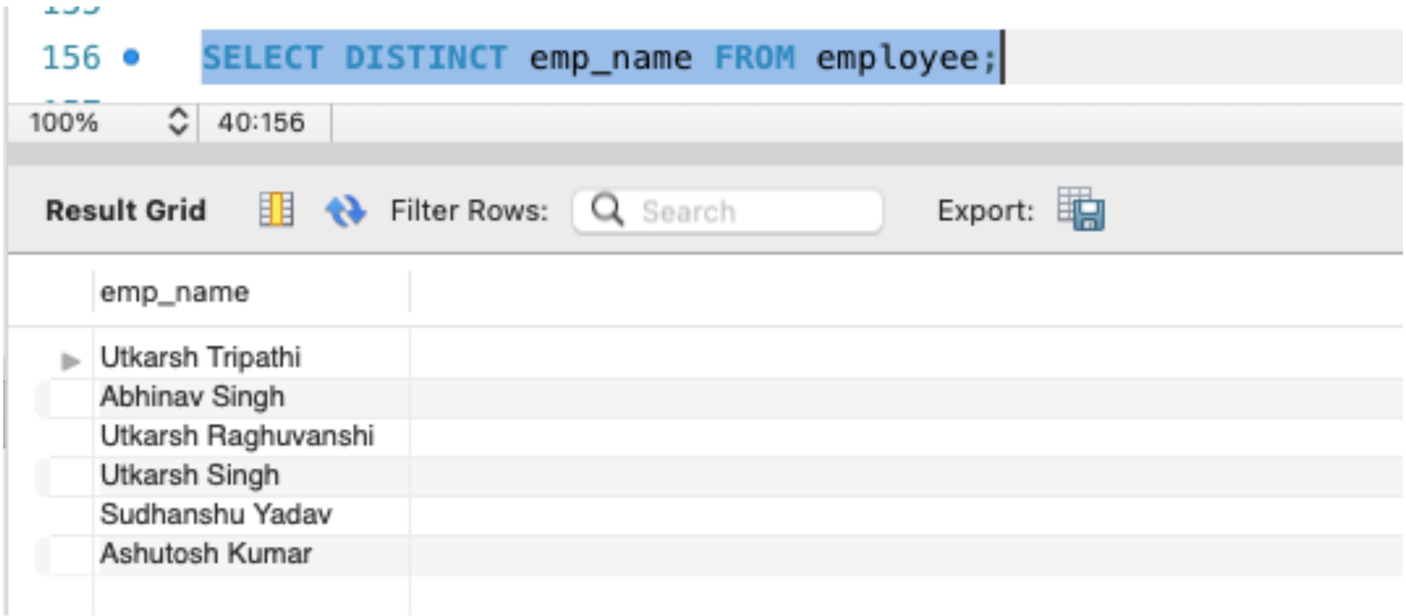


```sql
SELECT * FROM employee WHERE emp_id < SOME
                (SELECT emp_id FROM employee WHERE emp_city = 'Patna');
```

emp_id	emp_name	emp_city	emp_country
101	Utkarsh Tripathi	Varanasi	India
102	Abhinav Singh	Varanasi	India
103	Utkarsh Raghuvanshi	Varanasi	India
104	Utkarsh Singh	Allahabad	India
105	Sudhanshu Yadav	Allahabad	India
NULL	NULL	NULL	NULL

SQL Distinct Clause

The distinct keyword is used in conjunction with the select keyword. It is helpful when there is a need to avoid duplicate values present in any specific columns/table. When we use distinct keywords only the **unique values** are fetched.



```sql
SELECT DISTINCT emp_name FROM employee;
```

emp_name
Utkarsh Tripathi
Abhinav Singh
Utkarsh Raghuvanshi
Utkarsh Singh
Sudhanshu Yadav
Ashutosh Kumar