HEXAWARE CODING CHALLENGE

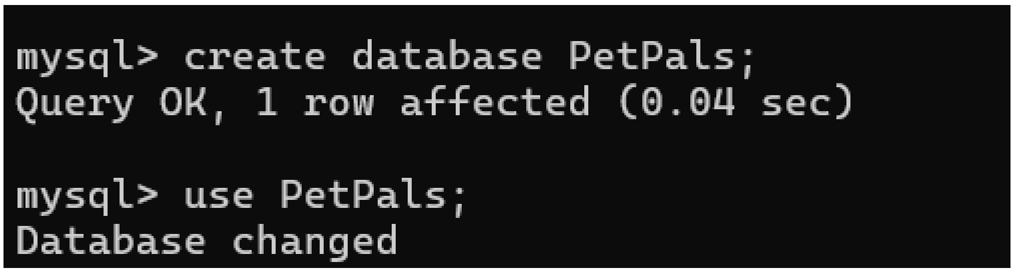
NAME : AISHWARYA B
SUPERSET ID: 5006869
Project Name: Pets Pal
Date: 17/06/2025

<u>Tasks:</u>

1. Provide a SQL script that initializes the database for the Pet Adoption Platform ”PetPals”.

Query:

```
->  create database PetPals;
Query OK, 1 row affected (0.04 sec)

mysql> use PetPals;
Database changed
```



```
mysql> create database PetPals;
Query OK, 1 row affected (0.04 sec)

mysql> use PetPals;
Database changed
```

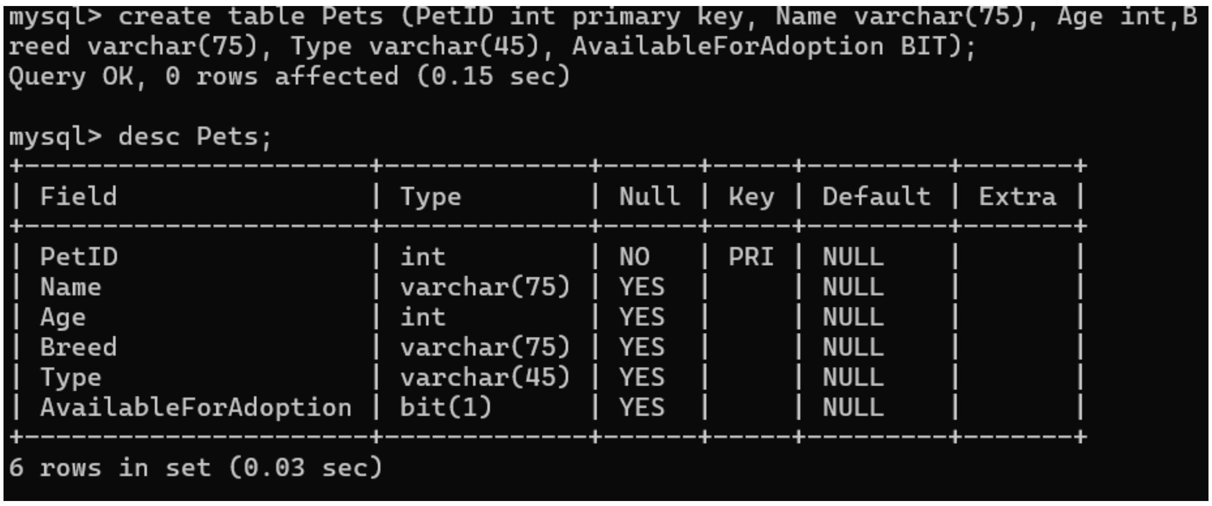
2. Create tables for pets, shelters, donations, adoption events, and participants.

Query:

TABLE NAME: Pets

```
->  create table Pets (PetID int primary key, Name varchar(75), Age int,Breed
varchar(75), Type varchar(45), AvailableForAdoption BIT);
Query OK, 0 rows affected (0.15 sec)
```

mysql> desc Pets;

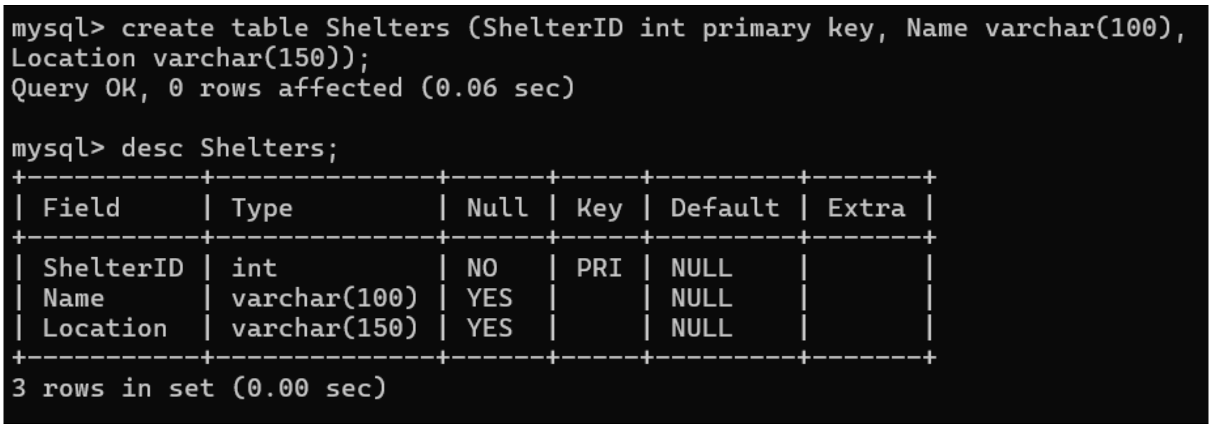


```
mysql> create table Pets (PetID int primary key, Name varchar(75), Age int,B
reed varchar(75), Type varchar(45), AvailableForAdoption BIT);
Query OK, 0 rows affected (0.15 sec)

mysql> desc Pets;
+----------------------+-------------+------+-----+---------+-------+
| Field                | Type        | Null | Key | Default | Extra |
+----------------------+-------------+------+-----+---------+-------+
| PetID                | int         | NO   | PRI | NULL    |       |
| Name                 | varchar(75) | YES  |     | NULL    |       |
| Age                  | int         | YES  |     | NULL    |       |
| Breed                | varchar(75) | YES  |     | NULL    |       |
| Type                 | varchar(45) | YES  |     | NULL    |       |
| AvailableForAdoption | bit(1)      | YES  |     | NULL    |       |
+----------------------+-------------+------+-----+---------+-------+
6 rows in set (0.03 sec)
```

TABLE NAME: Shelters

-> create table Shelters (ShelterID int primary key, Name varchar(100), Location varchar(150));
Query OK, 0 rows affected (0.06 sec)

mysql> desc Shelters;

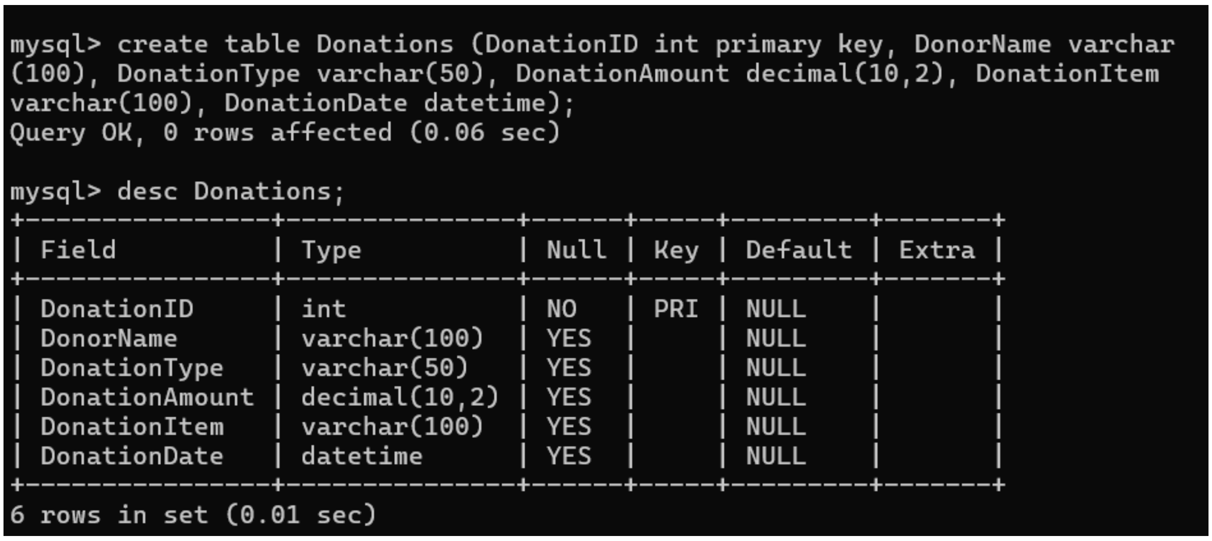


```
mysql> create table Shelters (ShelterID int primary key, Name varchar(100),
Location varchar(150));
Query OK, 0 rows affected (0.06 sec)

mysql> desc Shelters;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| ShelterID | int          | NO   | PRI | NULL    |       |
| Name      | varchar(100) | YES  |     | NULL    |       |
| Location  | varchar(150) | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

TABLE NAME: Donations

-> create table Donations (DonationID int primary key, DonorName varchar(100), DonationType varchar(50), DonationAmount decimal(10,2), DonationItem varchar(100), DonationDate datetime);
Query OK, 0 rows affected (0.06 sec)

mysql> desc Donations;

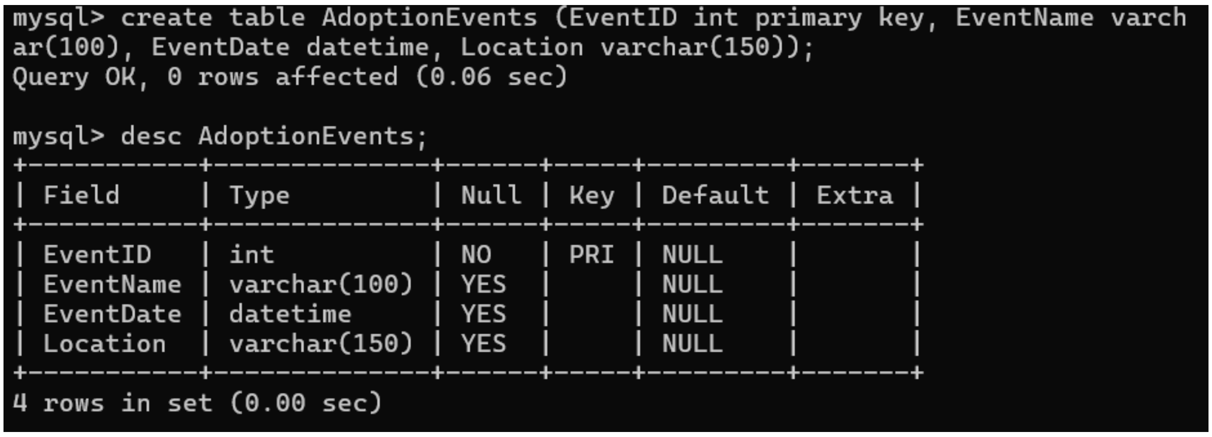


```
mysql> create table Donations (DonationID int primary key, DonorName varchar
(100), DonationType varchar(50), DonationAmount decimal(10,2), DonationItem
varchar(100), DonationDate datetime);
Query OK, 0 rows affected (0.06 sec)

mysql> desc Donations;
+----------------+---------------+------+-----+---------+-------+
| Field          | Type          | Null | Key | Default | Extra |
+----------------+---------------+------+-----+---------+-------+
| DonationID     | int           | NO   | PRI | NULL    |       |
| DonorName      | varchar(100)  | YES  |     | NULL    |       |
| DonationType   | varchar(50)   | YES  |     | NULL    |       |
| DonationAmount | decimal(10,2) | YES  |     | NULL    |       |
| DonationItem   | varchar(100)  | YES  |     | NULL    |       |
| DonationDate   | datetime      | YES  |     | NULL    |       |
+----------------+---------------+------+-----+---------+-------+
6 rows in set (0.01 sec)
```

TABLE NAME: AdoptionEvents

-> create table AdoptionEvents (EventID int primary key, EventName varchar(100), EventDate datetime, Location varchar(150));
Query OK, 0 rows affected (0.06 sec)

mysql> desc AdoptionEvents;

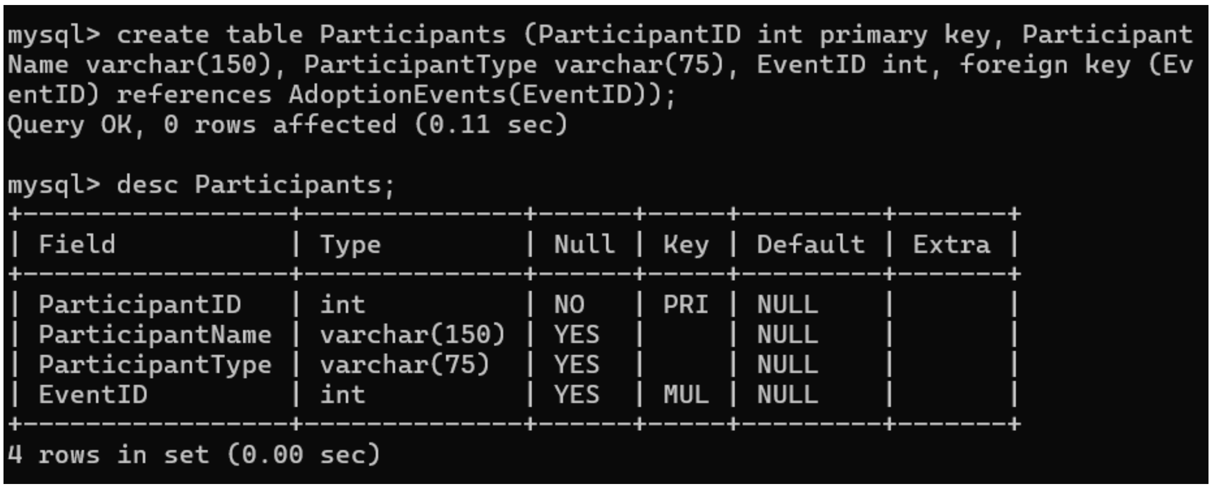


```
mysql> create table AdoptionEvents (EventID int primary key, EventName varch
ar(100), EventDate datetime, Location varchar(150));
Query OK, 0 rows affected (0.06 sec)

mysql> desc AdoptionEvents;
+-----------+--------------+------+-----+---------+-------+
| Field     | Type         | Null | Key | Default | Extra |
+-----------+--------------+------+-----+---------+-------+
| EventID   | int          | NO   | PRI | NULL    |       |
| EventName | varchar(100) | YES  |     | NULL    |       |
| EventDate | datetime     | YES  |     | NULL    |       |
| Location  | varchar(150) | YES  |     | NULL    |       |
+-----------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

TABLE NAME: Participants

-> create table Participants (ParticipantID int primary key, ParticipantName varchar(150), ParticipantType varchar(75), EventID int, foreign key (EventID) references AdoptionEvents(EventID));
Query OK, 0 rows affected (0.11 sec)

mysql> desc Participants;



```
mysql> create table Participants (ParticipantID int primary key, Participant
Name varchar(150), ParticipantType varchar(75), EventID int, foreign key (Ev
entID) references AdoptionEvents(EventID));
Query OK, 0 rows affected (0.11 sec)

mysql> desc Participants;
+-----------------+--------------+------+-----+---------+-------+
| Field           | Type         | Null | Key | Default | Extra |
+-----------------+--------------+------+-----+---------+-------+
| ParticipantID   | int          | NO   | PRI | NULL    |       |
| ParticipantName | varchar(150) | YES  |     | NULL    |       |
| ParticipantType | varchar(75)  | YES  |     | NULL    |       |
| EventID         | int          | YES  | MUL | NULL    |       |
+-----------------+--------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

3. Define appropriate primary keys, foreign keys, and constraints.

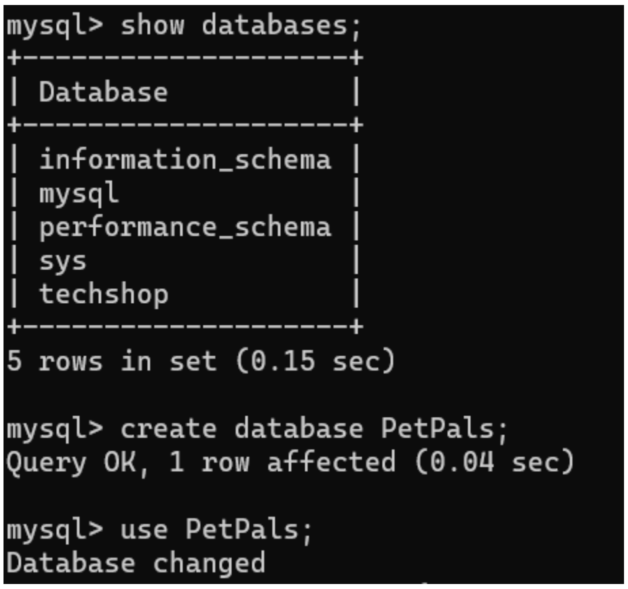
Primary Keys:

- PetID → Primary Key of the Pets table
- ShelterID → Primary Key of the Shelters table
- DonationID → Primary Key of the Donations table
- EventID → Primary Key of the AdoptionEvents table
- ParticipantID → Primary Key of the Participants table

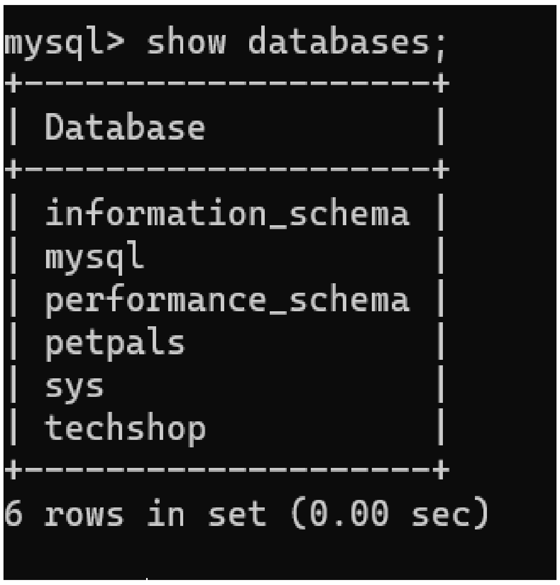
Foreign Keys

- EventID in the Participants table → references EventID in the AdoptionEvents table

4. Ensure the script handles potential errors, such as if the database or tables already exist.



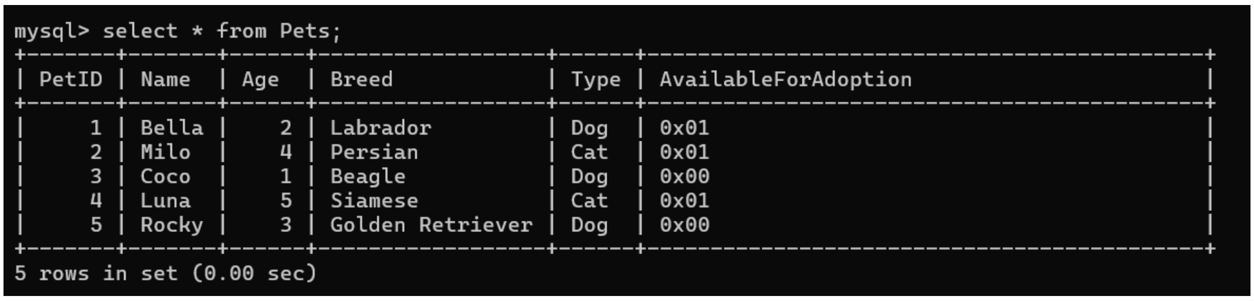
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| techshop           |
+--------------------+
5 rows in set (0.15 sec)

mysql> create database PetPals;
Query OK, 1 row affected (0.04 sec)

mysql> use PetPals;
Database changed
```



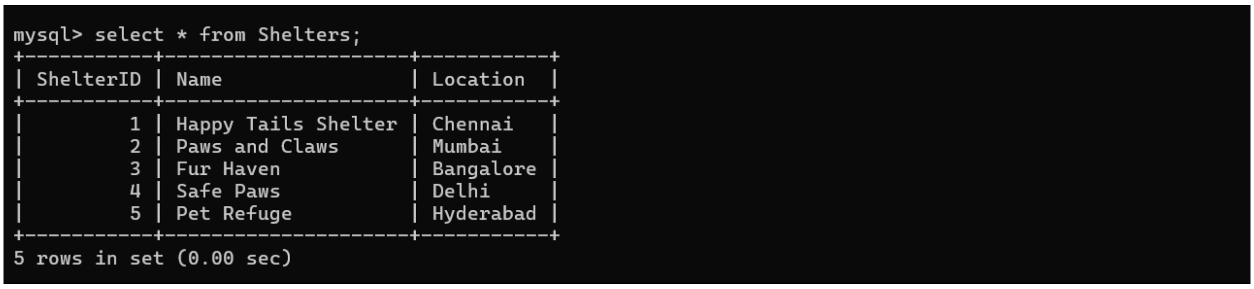
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| petpals            |
| sys                |
| techshop           |
+--------------------+
6 rows in set (0.00 sec)
```

Before creating the PetPals database and its tables, I first checked the list of existing databases to make sure there's no duplicate. Since the PetPals database didn't exist, it was safe to go ahead and create everything from scratch. This way, I avoided using DROP IF EXISTS because I made sure there were no conflicts. All the tables (Pets, Shelters, Donations, AdoptionEvents, and Participants) were created fresh with their respective fields, so there was no risk of errors or duplication.

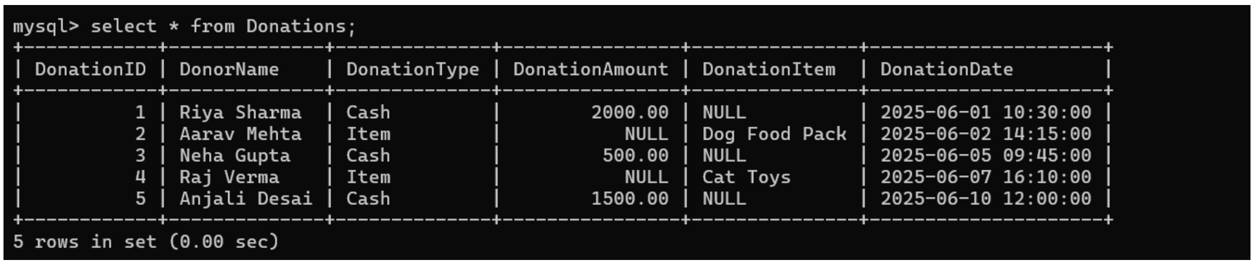
NOTE: *To test the structure and check that the tables are working properly, I added 5 sample records to each table. This helped me make sure that the schema is correct and all the constraints are working as expected. The insert queries were written manually to include different types of data and cover various cases in the PetPals system.*



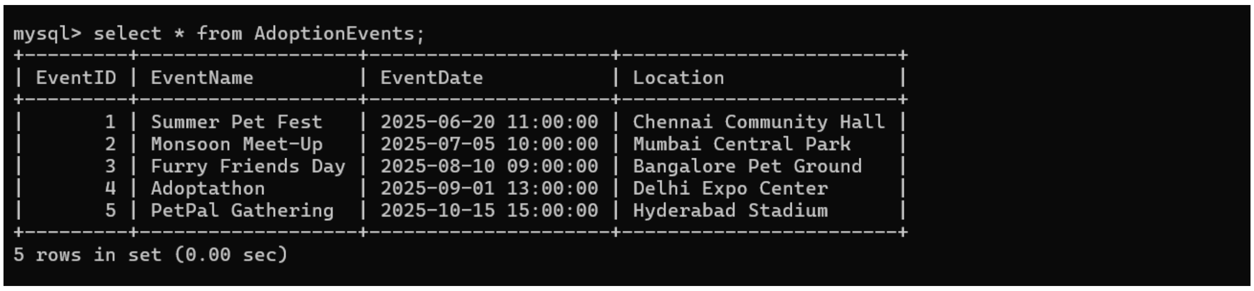
```
mysql> select * from Pets;
+-------+-------+------+------------------+------+------------------------------------------+
| PetID | Name  | Age  | Breed            | Type | AvailableForAdoption                     |
+-------+-------+------+------------------+------+------------------------------------------+
|     1 | Bella |    2 | Labrador         | Dog  | 0x01                                     |
|     2 | Milo  |    4 | Persian          | Cat  | 0x01                                     |
|     3 | Coco  |    1 | Beagle           | Dog  | 0x00                                     |
|     4 | Luna  |    5 | Siamese          | Cat  | 0x01                                     |
|     5 | Rocky |    3 | Golden Retriever | Dog  | 0x00                                     |
+-------+-------+------+------------------+------+------------------------------------------+
5 rows in set (0.00 sec)
```



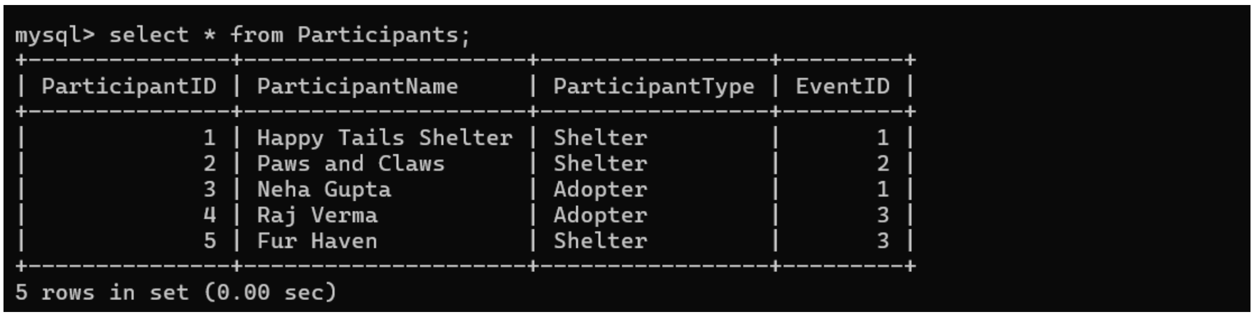
```
mysql> select * from Shelters;
+-----------+---------------------+-----------+
| ShelterID | Name                | Location  |
+-----------+---------------------+-----------+
|         1 | Happy Tails Shelter | Chennai   |
|         2 | Paws and Claws      | Mumbai    |
|         3 | Fur Haven           | Bangalore |
|         4 | Safe Paws           | Delhi     |
|         5 | Pet Refuge          | Hyderabad |
+-----------+---------------------+-----------+
5 rows in set (0.00 sec)
```



```
mysql> select * from Donations;
+------------+--------------+--------------+----------------+---------------+---------------------+
| DonationID | DonorName    | DonationType | DonationAmount | DonationItem  | DonationDate        |
+------------+--------------+--------------+----------------+---------------+---------------------+
|          1 | Riya Sharma  | Cash         |        2000.00 | NULL          | 2025-06-01 10:30:00 |
|          2 | Aarav Mehta  | Item         |           NULL | Dog Food Pack | 2025-06-02 14:15:00 |
|          3 | Neha Gupta   | Cash         |         500.00 | NULL          | 2025-06-05 09:45:00 |
|          4 | Raj Verma    | Item         |           NULL | Cat Toys      | 2025-06-07 16:10:00 |
|          5 | Anjali Desai | Cash         |        1500.00 | NULL          | 2025-06-10 12:00:00 |
+------------+--------------+--------------+----------------+---------------+---------------------+
5 rows in set (0.00 sec)
```



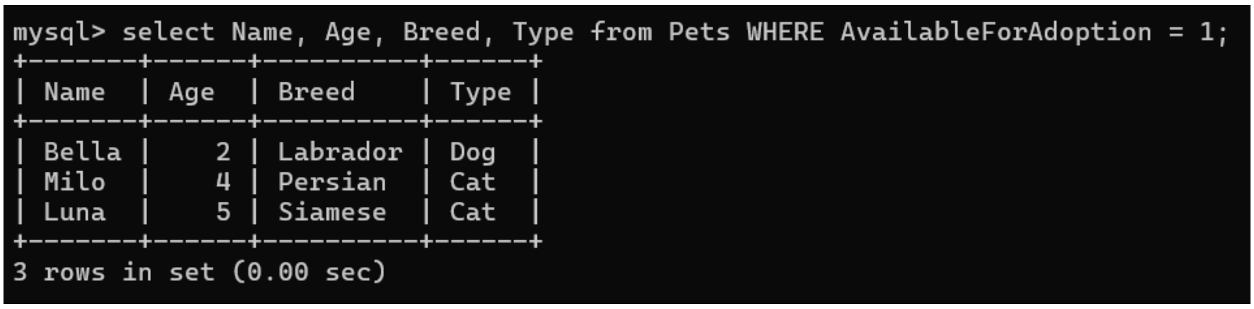
```
mysql> select * from AdoptionEvents;
+---------+-------------------+---------------------+------------------------+
| EventID | EventName         | EventDate           | Location               |
+---------+-------------------+---------------------+------------------------+
|       1 | Summer Pet Fest   | 2025-06-20 11:00:00 | Chennai Community Hall |
|       2 | Monsoon Meet-Up   | 2025-07-05 10:00:00 | Mumbai Central Park    |
|       3 | Furry Friends Day | 2025-08-10 09:00:00 | Bangalore Pet Ground   |
|       4 | Adoptathon        | 2025-09-01 13:00:00 | Delhi Expo Center      |
|       5 | PetPal Gathering  | 2025-10-15 15:00:00 | Hyderabad Stadium      |
+---------+-------------------+---------------------+------------------------+
5 rows in set (0.00 sec)
```



```
mysql> select * from Participants;
+---------------+---------------------+-----------------+---------+
| ParticipantID | ParticipantName     | ParticipantType | EventID |
+---------------+---------------------+-----------------+---------+
|             1 | Happy Tails Shelter | Shelter         |       1 |
|             2 | Paws and Claws      | Shelter         |       2 |
|             3 | Neha Gupta          | Adopter         |       1 |
|             4 | Raj Verma           | Adopter         |       3 |
|             5 | Fur Haven           | Shelter         |       3 |
+---------------+---------------------+-----------------+---------+
5 rows in set (0.00 sec)
```

5. Write an SQL query that retrieves a list of available pets (those marked as available for adoption) from the "Pets" table. Include the pet's name, age, breed, and type in the result set. Ensure that the query filters out pets that are not available for adoption.

Query:
-> select Name, Age, Breed, Type from Pets WHERE AvailableForAdoption = 1;

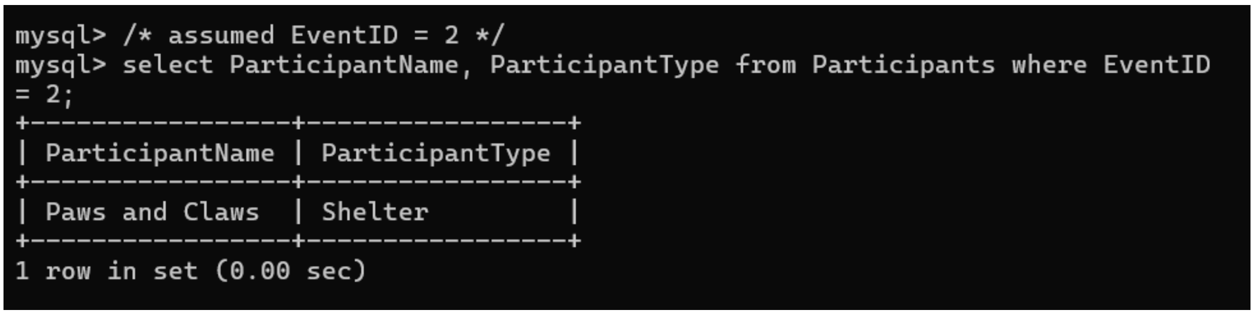


```
mysql> select Name, Age, Breed, Type from Pets WHERE AvailableForAdoption = 1;
+-------+------+----------+------+
| Name  | Age  | Breed    | Type |
+-------+------+----------+------+
| Bella |    2 | Labrador | Dog  |
| Milo  |    4 | Persian  | Cat  |
| Luna  |    5 | Siamese  | Cat  |
+-------+------+----------+------+
3 rows in set (0.00 sec)
```

6. Write an SQL query that retrieves the names of participants (shelters and adopters) registered for a specific adoption event. Use a parameter to specify the event ID. Ensure that the query joins the necessary tables to retrieve the participant names and types.

Query:
-> select ParticipantName, ParticipantType from Participants where EventID = @EventID;

select ParticipantName, ParticipantType from Participants WHERE EventID = 2;

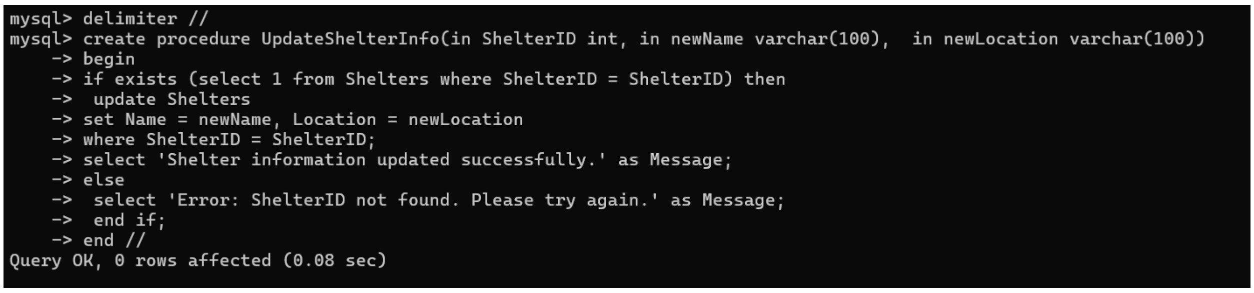


```
mysql> /* assumed EventID = 2 */
mysql> select ParticipantName, ParticipantType from Participants where EventID
= 2;
+-----------------+-----------------+
| ParticipantName | ParticipantType |
+-----------------+-----------------+
| Paws and Claws  | Shelter         |
+-----------------+-----------------+
1 row in set (0.00 sec)
```

7. Create a stored procedure in SQL that allows a shelter to update its information (name and location) in the "Shelters" table. Use parameters to pass the shelter ID and the new information. Ensure that the procedure performs the update and handles potential errors, such as an invalid shelter ID.

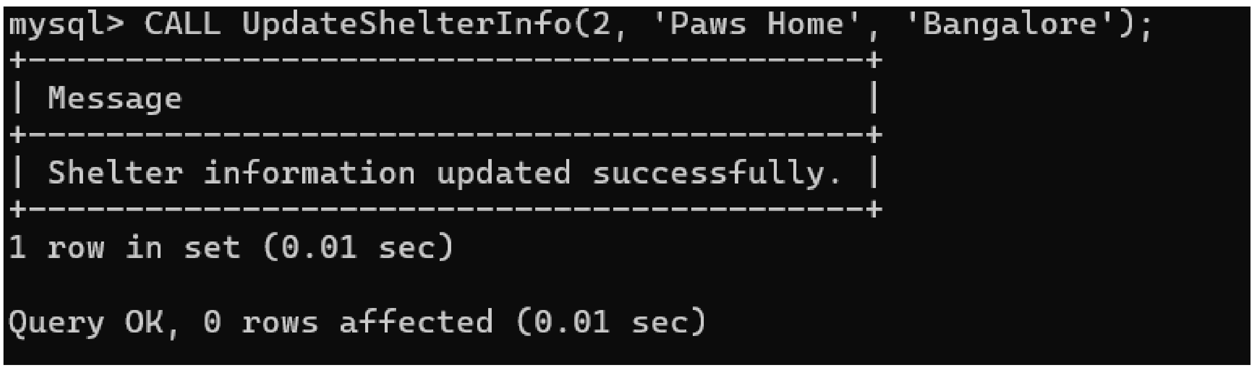
Query:

```
->  delimiter //
mysql> create procedure UpdateShelterInfo(in ShelterID int, in newName varchar(100), in newLocation varchar(100))
    -> begin
    -> if exists (select 1 from Shelters where ShelterID = ShelterID) then
    ->  update Shelters
    -> set Name = newName, Location = newLocation
    -> where ShelterID = ShelterID;
    -> select 'Shelter information updated successfully.' as Message;
    -> else
    ->  select 'Error: ShelterID not found. Please try again.' as Message;
    ->  end if;
    -> end //
Query OK, 0 rows affected (0.08 sec)

mysql> delimiter;
```



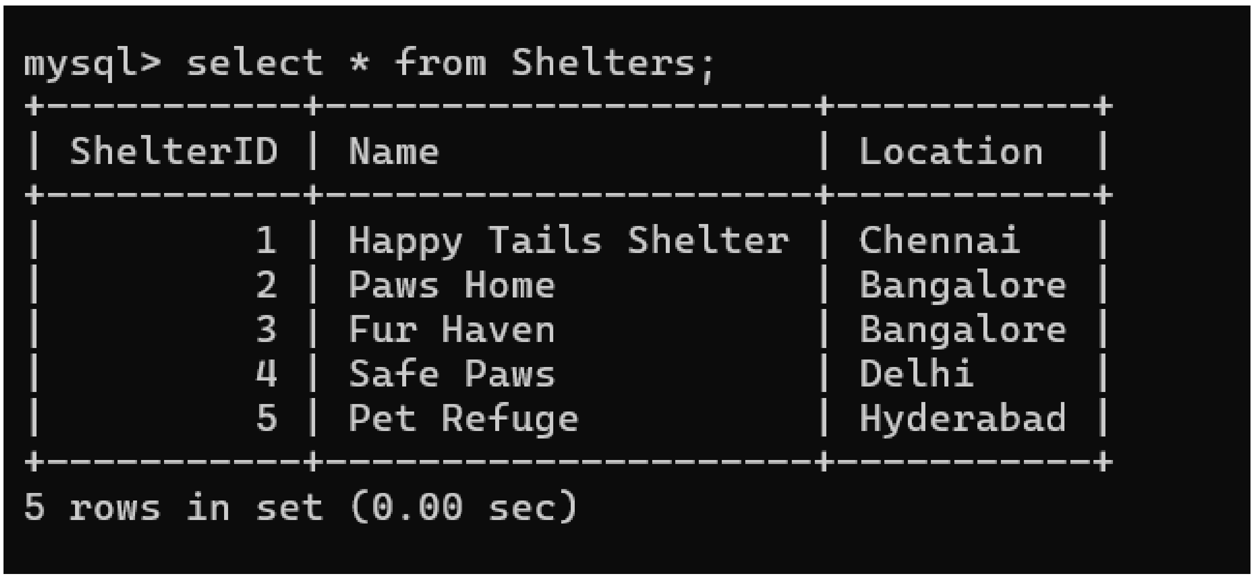
```
mysql> delimiter //
mysql> create procedure UpdateShelterInfo(in ShelterID int, in newName varchar(100),  in newLocation varchar(100))
    -> begin
    -> if exists (select 1 from Shelters where ShelterID = ShelterID) then
    ->  update Shelters
    -> set Name = newName, Location = newLocation
    -> where ShelterID = ShelterID;
    -> select 'Shelter information updated successfully.' as Message;
    -> else
    ->  select 'Error: ShelterID not found. Please try again.' as Message;
    ->  end if;
    -> end //
Query OK, 0 rows affected (0.08 sec)
```

CALL UpdateShelterInfo(2, 'Paws Home', 'Bangalore');



```
mysql> CALL UpdateShelterInfo(2, 'Paws Home', 'Bangalore');
+-------------------------------------------+
| Message                                   |
+-------------------------------------------+
| Shelter information updated successfully. |
+-------------------------------------------+
1 row in set (0.01 sec)

Query OK, 0 rows affected (0.01 sec)
```

select * from Shelters;


```
mysql> select * from Shelters;
+-----------+---------------------+-----------+
| ShelterID | Name                | Location  |
+-----------+---------------------+-----------+
|         1 | Happy Tails Shelter | Chennai   |
|         2 | Paws Home           | Bangalore |
|         3 | Fur Haven           | Bangalore |
|         4 | Safe Paws           | Delhi     |
|         5 | Pet Refuge          | Hyderabad |
+-----------+---------------------+-----------+
5 rows in set (0.00 sec)
```

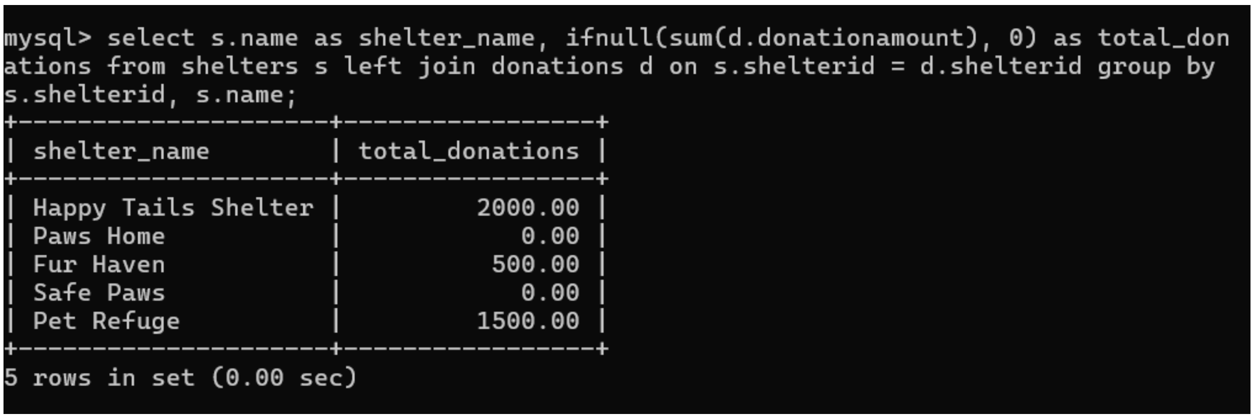
8. Write an SQL query that calculates and retrieves the total donation amount for each shelter (by shelter name) from the "Donations" table. The result should include the shelter name and the total donation amount. Ensure that the query handles cases where a shelter has received no donations.

Query:

-> alter table donations add column shelterid int;
Query OK, 0 rows affected (0.06 sec)
Records: 0 Duplicates: 0 Warnings: 0

-> select s.name as shelter_name, ifnull(sum(d.donationamount), 0) as total_donations from shelters s left join donations d on s.shelterid = d.shelterid group by s.shelterid, s.name;

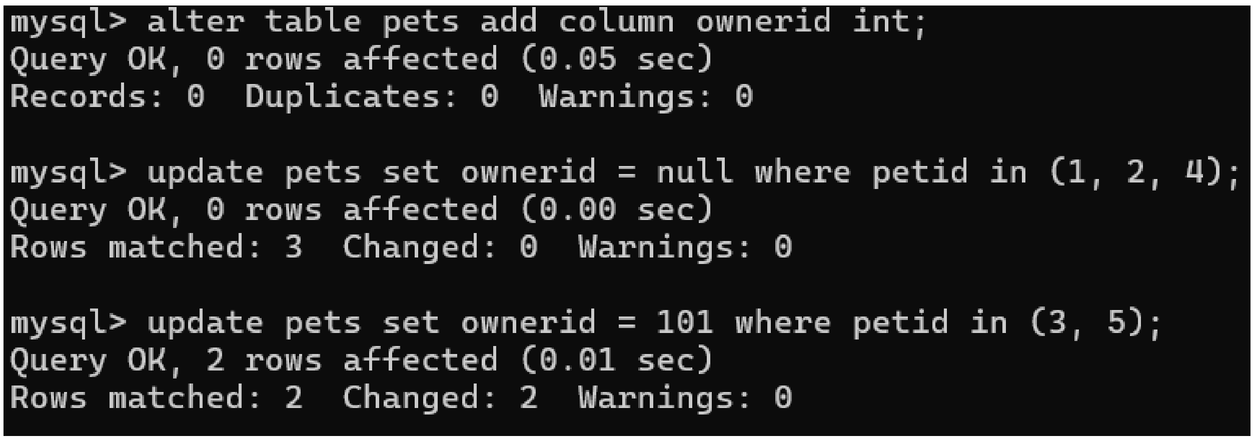
NOTE: *Since the original Donations table doesn't include a ShelterID, I added that column to create a proper link between donations and shelters. This allows me to calculate total donations per shelter using a left join.*



```
mysql> select s.name as shelter_name, ifnull(sum(d.donationamount), 0) as total_don
ations from shelters s left join donations d on s.shelterid = d.shelterid group by
s.shelterid, s.name;
+---------------------+-----------------+
| shelter_name        | total_donations |
+---------------------+-----------------+
| Happy Tails Shelter |         2000.00 |
| Paws Home           |            0.00 |
| Fur Haven           |          500.00 |
| Safe Paws           |            0.00 |
| Pet Refuge          |         1500.00 |
+---------------------+-----------------+
5 rows in set (0.00 sec)
```

9. Write an SQL query that retrieves the names of pets from the "Pets" table that do not have an owner (i.e., where "OwnerID" is null). Include the pet's name, age, breed, and type in the result set.

```
->  alter table pets add column ownerid int;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> update pets set ownerid = null where petid in (1, 2, 4);
Query OK, 0 rows affected (0.00 sec)
Rows matched: 3  Changed: 0  Warnings: 0

mysql> update pets set ownerid = 101 where petid in (3, 5);
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0
```

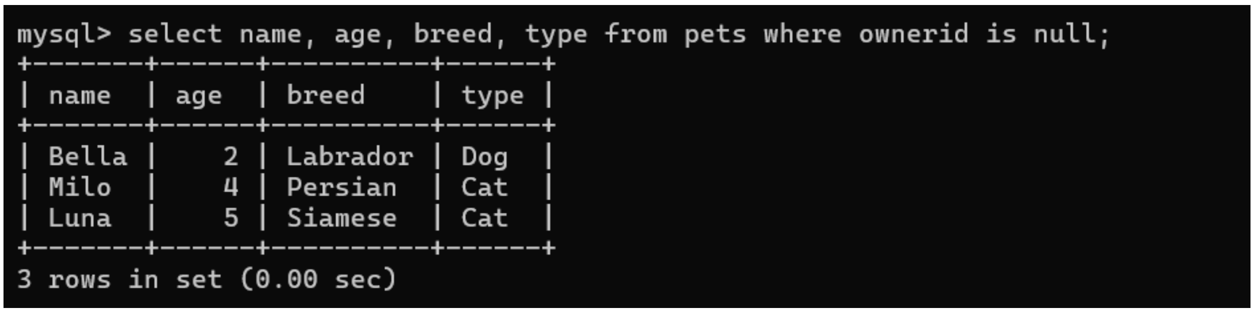


```
mysql> alter table pets add column ownerid int;
Query OK, 0 rows affected (0.05 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> update pets set ownerid = null where petid in (1, 2, 4);
Query OK, 0 rows affected (0.00 sec)
Rows matched: 3  Changed: 0  Warnings: 0

mysql> update pets set ownerid = 101 where petid in (3, 5);
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0
```

NOTE: *I added an ownerid column to the pets table to track adoption status. Then I wrote a query to retrieve pets where the owner ID is null — meaning those pets are not adopted yet.*

Query:

-> select name, age, breed, type from pets where ownerid is null;

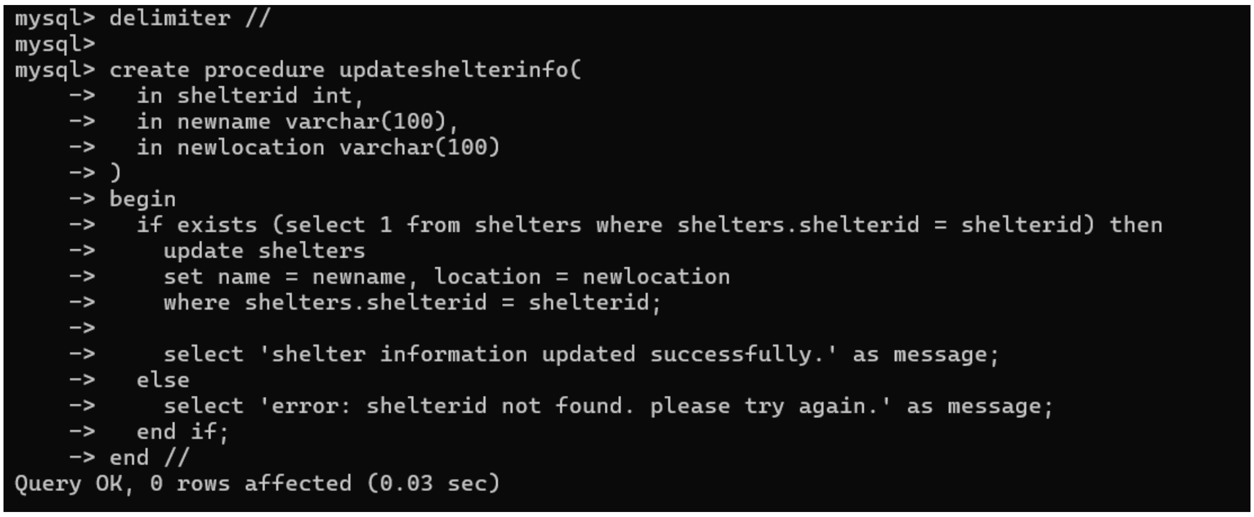


```
mysql> select name, age, breed, type from pets where ownerid is null;
+-------+------+----------+------+
| name  | age  | breed    | type |
+-------+------+----------+------+
| Bella |    2 | Labrador | Dog  |
| Milo  |    4 | Persian  | Cat  |
| Luna  |    5 | Siamese  | Cat  |
+-------+------+----------+------+
3 rows in set (0.00 sec)
```

10. Write an SQL query that retrieves the total donation amount for each month and year (e.g., January 2023) from the "Donations" table. The result should include the month-year and the corresponding total donation amount. Ensure that the query handles cases where no donations were made in a specific month-year.

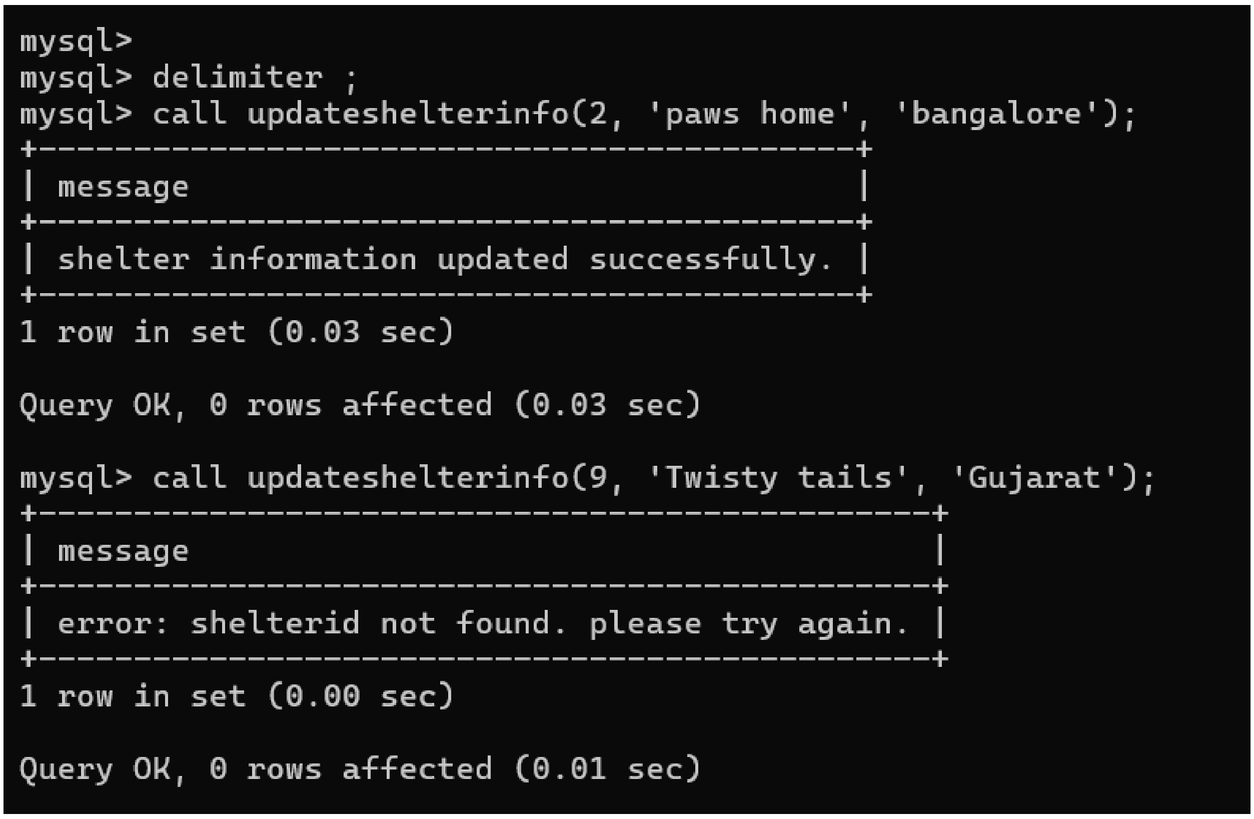
Query:

```
->  delimiter //
mysql>
mysql> create procedure updateshelterinfo(
   ->   in shelterid int,
   ->   in newname varchar(100),
   ->   in newlocation varchar(100)
   -> )
   -> begin
   ->   if exists (select 1 from shelters where shelters.shelterid = shelterid) then
   ->     update shelters
   ->     set name = newname, location = newlocation
   ->     where shelters.shelterid = shelterid;
   ->
   ->     select 'shelter information updated successfully.' as message;
   ->   else
   ->     select 'error: shelterid not found. please try again.' as message;
   ->   end if;
   -> end //
Query OK, 0 rows affected (0.03 sec)
```



```
mysql> delimiter //
mysql>
mysql> create procedure updateshelterinfo(
    ->   in shelterid int,
    ->   in newname varchar(100),
    ->   in newlocation varchar(100)
    -> )
    -> begin
    ->   if exists (select 1 from shelters where shelters.shelterid = shelterid) then
    ->     update shelters
    ->     set name = newname, location = newlocation
    ->     where shelters.shelterid = shelterid;
    ->
    ->     select 'shelter information updated successfully.' as message;
    ->   else
    ->     select 'error: shelterid not found. please try again.' as message;
    ->   end if;
    -> end //
Query OK, 0 rows affected (0.03 sec)
```

delimiter ;
-> call updateshelterinfo(2, 'paws home', 'bangalore');
-> call updateshelterinfo(9, 'Twisty tails', 'Gujarat');

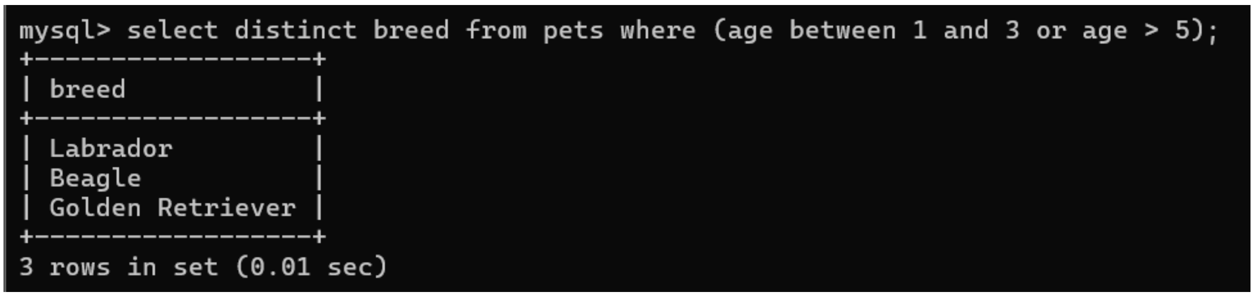


```
mysql>
mysql> delimiter ;
mysql> call updateshelterinfo(2, 'paws home', 'bangalore');
+--------------------------------------------+
| message                                    |
+--------------------------------------------+
| shelter information updated successfully. |
+--------------------------------------------+
1 row in set (0.03 sec)

Query OK, 0 rows affected (0.03 sec)

mysql> call updateshelterinfo(9, 'Twisty tails', 'Gujarat');
+------------------------------------------------+
| message                                        |
+------------------------------------------------+
| error: shelterid not found. please try again. |
+------------------------------------------------+
1 row in set (0.00 sec)

Query OK, 0 rows affected (0.01 sec)
```

NOTE: *I created a stored procedure called updateshelterinfo that allows a shelter's name and location to be updated using its shelter ID. The procedure first checks if the shelter exists. If it does, it updates the details and shows a success message. If not, it shows a custom error message saying the update was skipped. I tested the procedure using the call statement with both valid and invalid IDs to confirm that it works correctly in both cases.*

11. Retrieve a list of distinct breeds for all pets that are either aged between 1 and 3 years or older than 5 years.

Query:
-> select distinct breed from pets where (age between 1 and 3 or age > 5);



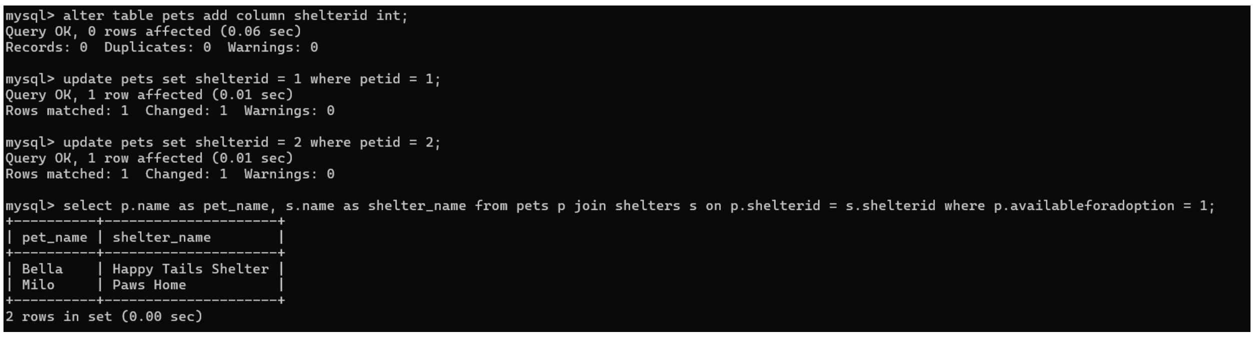
```
mysql> select distinct breed from pets where (age between 1 and 3 or age > 5);
+------------------+
| breed            |
+------------------+
| Labrador         |
| Beagle           |
| Golden Retriever |
+------------------+
3 rows in set (0.01 sec)
```

12. Retrieve a list of pets and their respective shelters where the pets are currently available for adoption.

Query:
-> alter table pets add column shelterid int;

-> update pets set shelterid = 1 where petid = 1;
update pets set shelterid = 2 where petid = 2;

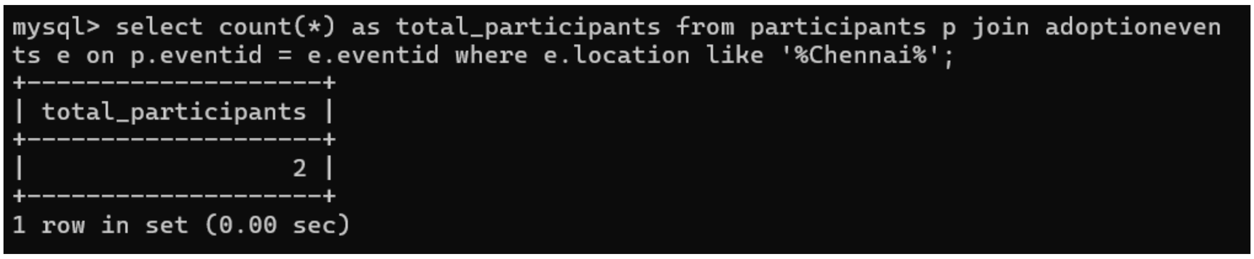
->select p.name as pet_name, s.name as shelter_name from pets p join shelters s on p.shelterid = s.shelterid where p.availableforadoption = 1;



```
mysql> alter table pets add column shelterid int;
Query OK, 0 rows affected (0.06 sec)
Records: 0  Duplicates: 0  Warnings: 0

mysql> update pets set shelterid = 1 where petid = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> update pets set shelterid = 2 where petid = 2;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1  Changed: 1  Warnings: 0

mysql> select p.name as pet_name, s.name as shelter_name from pets p join shelters s on p.shelterid = s.shelterid where p.availableforadoption = 1;
+----------+--------------------+
| pet_name | shelter_name       |
+----------+--------------------+
| Bella    | Happy Tails Shelter |
| Milo     | Paws Home          |
+----------+--------------------+
2 rows in set (0.00 sec)
```

13. Find the total number of participants in events organized by shelters located in specific city. Example: City=Chennai

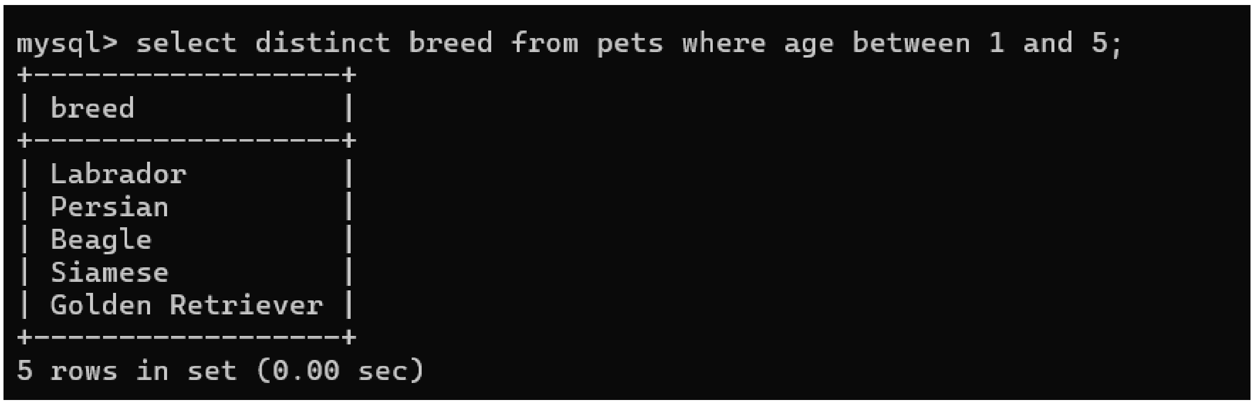
Query:
-> select count(*) as total_participants from participants p join adoptionevents e on p.eventid = e.eventid where e.location like '%Chennai%';



```
mysql> select count(*) as total_participants from participants p join adoptioneven
ts e on p.eventid = e.eventid where e.location like '%Chennai%';
+--------------------+
| total_participants |
+--------------------+
|                  2 |
+--------------------+
1 row in set (0.00 sec)
```

14. Retrieve a list of unique breeds for pets with ages between 1 and 5 years.

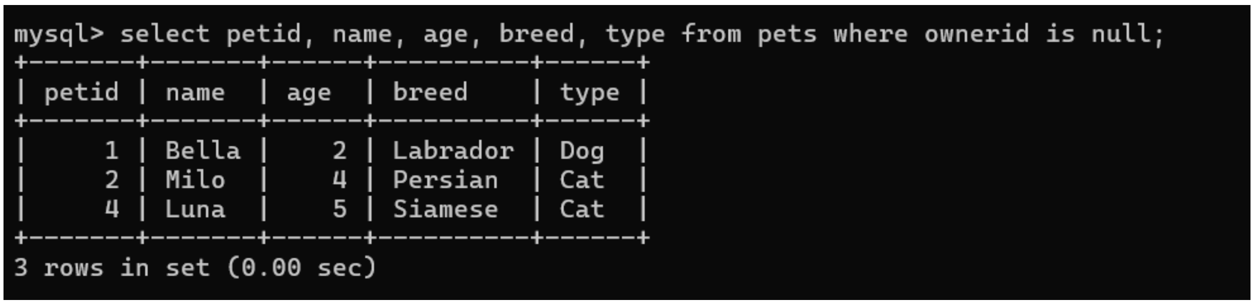
Query:
->select distinct breed from pets where age between 1 and 5;



```
mysql> select distinct breed from pets where age between 1 and 5;
+------------------+
| breed            |
+------------------+
| Labrador         |
| Persian          |
| Beagle           |
| Siamese          |
| Golden Retriever |
+------------------+
5 rows in set (0.00 sec)
```

15. Find the pets that have not been adopted by selecting their information from the 'Pet' table.

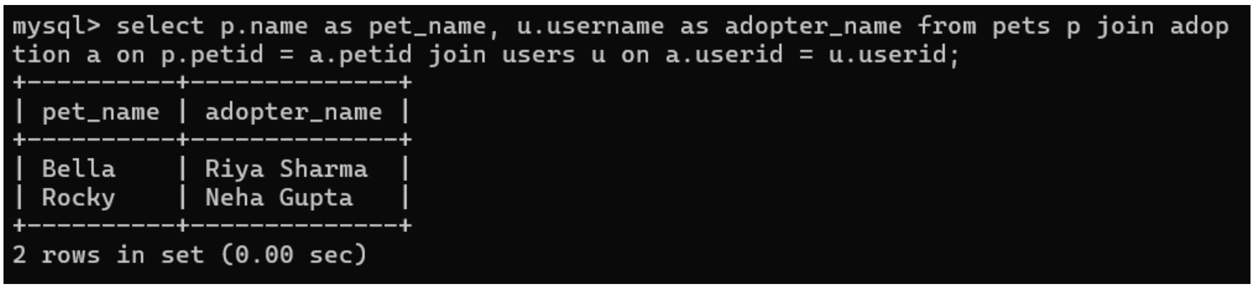
Query:
-> select petid, name, age, breed, type from pets where ownerid is null;



```
mysql> select petid, name, age, breed, type from pets where ownerid is null;
+-------+-------+------+----------+------+
| petid | name  | age  | breed    | type |
+-------+-------+------+----------+------+
|     1 | Bella |    2 | Labrador | Dog  |
|     2 | Milo  |    4 | Persian  | Cat  |
|     4 | Luna  |    5 | Siamese  | Cat  |
+-------+-------+------+----------+------+
3 rows in set (0.00 sec)
```

16. Retrieve the names of all adopted pets along with the adopter's name from the 'Adoption' and 'User' tables.

Query:
-> select p.name as pet_name, u.username as adopter_name from pets p join adoption a on p.petid = a.petid join users u on a.userid = u.userid;

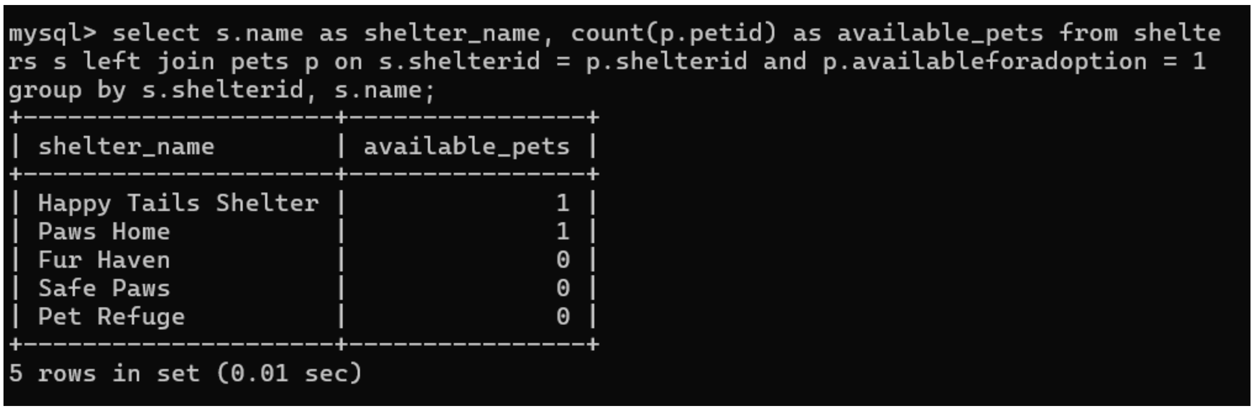


```
mysql> select p.name as pet_name, u.username as adopter_name from pets p join adop
tion a on p.petid = a.petid join users u on a.userid = u.userid;
+----------+--------------+
| pet_name | adopter_name |
+----------+--------------+
| Bella    | Riya Sharma  |
| Rocky    | Neha Gupta   |
+----------+--------------+
2 rows in set (0.00 sec)
```

NOTE: *Since the schema didn't include Adoption and Users tables by default, I created them with the necessary foreign key relationships. This allowed me to link each adopted pet to its adopter and retrieve their names using a multi-table join.*

17. Retrieve a list of all shelters along with the count of pets currently available for adoption in each shelter.

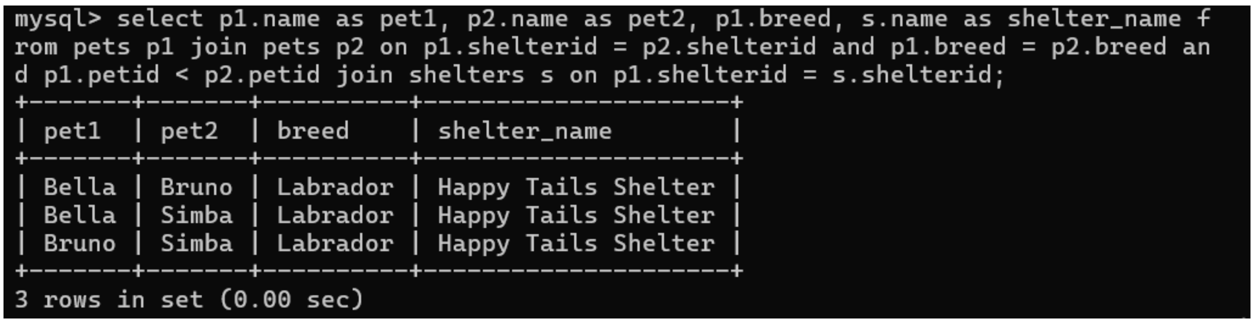
Query:
-> select s.name as shelter_name, count(p.petid) as available_pets from shelters s left join pets p on s.shelterid = p.shelterid and p.availableforadoption = 1 group by s.shelterid, s.name;



```
mysql> select s.name as shelter_name, count(p.petid) as available_pets from shelte
rs s left join pets p on s.shelterid = p.shelterid and p.availableforadoption = 1
group by s.shelterid, s.name;
+---------------------+----------------+
| shelter_name        | available_pets |
+---------------------+----------------+
| Happy Tails Shelter |              1 |
| Paws Home           |              1 |
| Fur Haven           |              0 |
| Safe Paws           |              0 |
| Pet Refuge          |              0 |
+---------------------+----------------+
5 rows in set (0.01 sec)
```

18. Find pairs of pets from the same shelter that have the same breed.

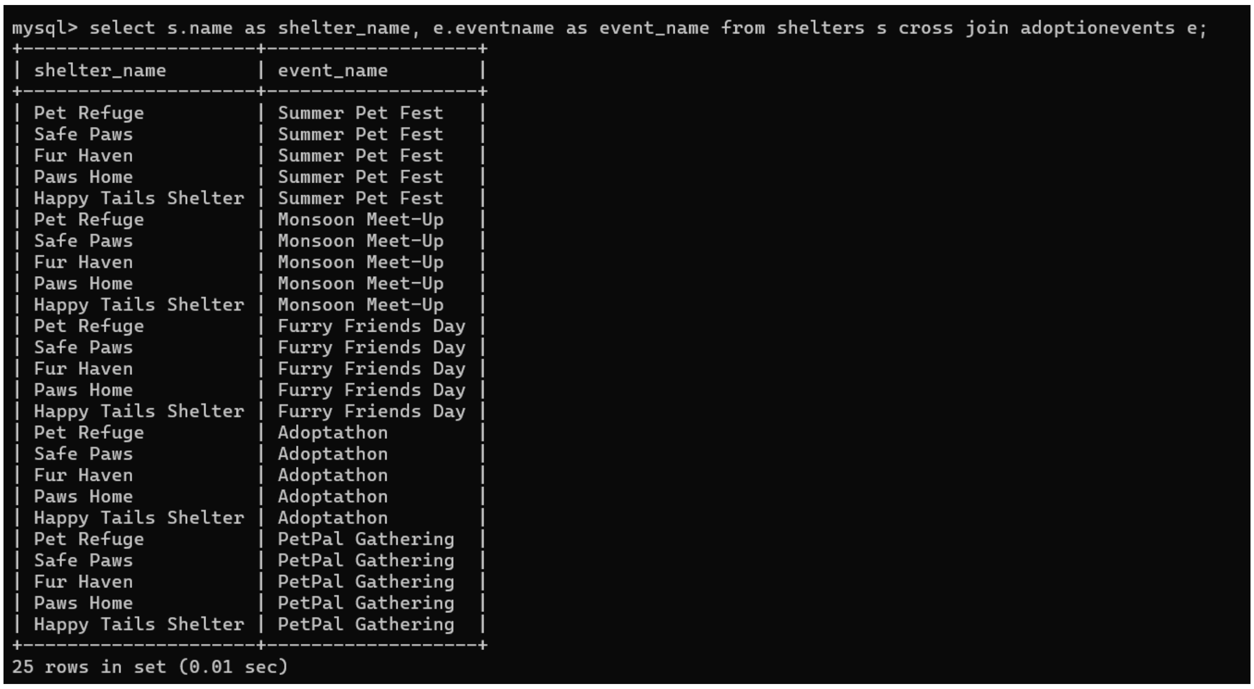
Query:
-> select p1.name as pet1, p2.name as pet2, p1.breed, s.name as shelter_name from pets p1 join pets p2 on p1.shelterid = p2.shelterid and p1.breed = p2.breed and p1.petid < p2.petid join shelters s on p1.shelterid = s.shelterid;



```
mysql> select p1.name as pet1, p2.name as pet2, p1.breed, s.name as shelter_name f
rom pets p1 join pets p2 on p1.shelterid = p2.shelterid and p1.breed = p2.breed an
d p1.petid < p2.petid join shelters s on p1.shelterid = s.shelterid;
+-------+-------+----------+---------------------+
| pet1  | pet2  | breed    | shelter_name        |
+-------+-------+----------+---------------------+
| Bella | Bruno | Labrador | Happy Tails Shelter |
| Bella | Simba | Labrador | Happy Tails Shelter |
| Bruno | Simba | Labrador | Happy Tails Shelter |
+-------+-------+----------+---------------------+
3 rows in set (0.00 sec)
```

19. List all possible combinations of shelters and adoption events.

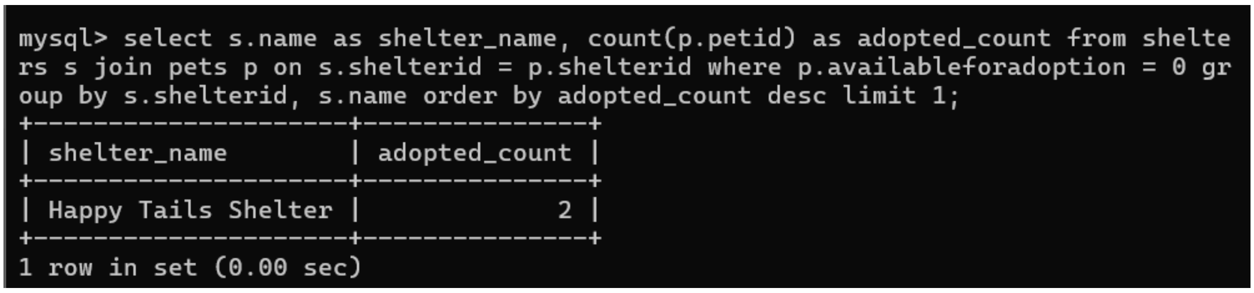
Query:
-> select s.name as shelter_name, e.eventname as event_name from shelters s cross join adoptionevents e;



```
mysql> select s.name as shelter_name, e.eventname as event_name from shelters s cross join adoptionevents e;
+---------------------+-------------------+
| shelter_name        | event_name        |
+---------------------+-------------------+
| Pet Refuge          | Summer Pet Fest   |
| Safe Paws           | Summer Pet Fest   |
| Fur Haven           | Summer Pet Fest   |
| Paws Home           | Summer Pet Fest   |
| Happy Tails Shelter | Summer Pet Fest   |
| Pet Refuge          | Monsoon Meet-Up   |
| Safe Paws           | Monsoon Meet-Up   |
| Fur Haven           | Monsoon Meet-Up   |
| Paws Home           | Monsoon Meet-Up   |
| Happy Tails Shelter | Monsoon Meet-Up   |
| Pet Refuge          | Furry Friends Day |
| Safe Paws           | Furry Friends Day |
| Fur Haven           | Furry Friends Day |
| Paws Home           | Furry Friends Day |
| Happy Tails Shelter | Furry Friends Day |
| Pet Refuge          | Adoptathon        |
| Safe Paws           | Adoptathon        |
| Fur Haven           | Adoptathon        |
| Paws Home           | Adoptathon        |
| Happy Tails Shelter | Adoptathon        |
| Pet Refuge          | PetPal Gathering  |
| Safe Paws           | PetPal Gathering  |
| Fur Haven           | PetPal Gathering  |
| Paws Home           | PetPal Gathering  |
| Happy Tails Shelter | PetPal Gathering  |
+---------------------+-------------------+
25 rows in set (0.01 sec)
```

20. Determine the shelter that has the highest number of adopted pets.

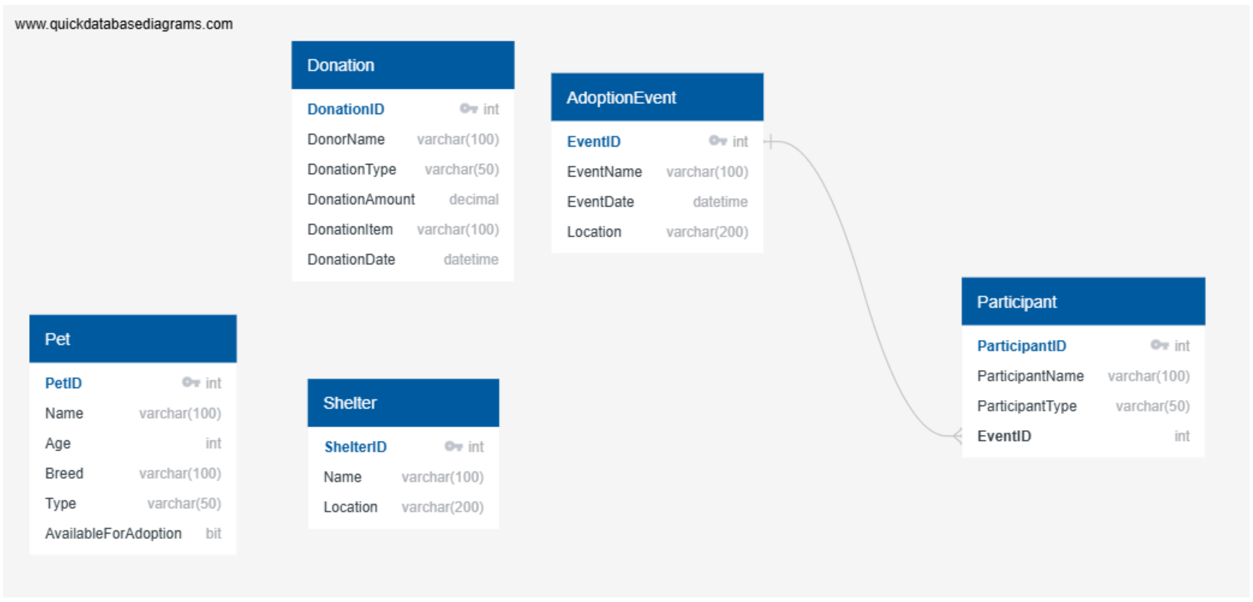
Query:
-> select s.name as shelter_name, count(p.petid) as adopted_count from shelters s join pets p on s.shelterid = p.shelterid where p.availableforadoption = 0 group by s.shelterid, s.name order by adopted_count desc limit 1;



```
mysql> select s.name as shelter_name, count(p.petid) as adopted_count from shelte
rs s join pets p on s.shelterid = p.shelterid where p.availableforadoption = 0 gr
oup by s.shelterid, s.name order by adopted_count desc limit 1;
+---------------------+---------------+
| shelter_name        | adopted_count |
+---------------------+---------------+
| Happy Tails Shelter |             2 |
+---------------------+---------------+
1 row in set (0.00 sec)
```

NOTE: *I used a join between shelters and pets, filtered pets that have availableforadoption = 0, and grouped the results by shelter. Then, I selected the shelter with the highest adoption count using order by limit 1.*

ENTITY RELATIONSHIP DIAGRAM:


www.quickdatabasediagrams.com
Donation
DonationID int
DonorName varchar(100)
DonationType varchar(50)
DonationAmount decimal
DonationItem varchar(100)
DonationDate datetime
AdoptionEvent
EventID int
EventName varchar(100)
EventDate datetime
Location varchar(200)
Participant
ParticipantID int
ParticipantName varchar(100)
ParticipantType varchar(50)
EventID int
Pet
PetID int
Name varchar(100)
Age int
Breed varchar(100)
Type varchar(50)
AvailableForAdoption bit
Shelter
ShelterID int
Name varchar(100)
Location varchar(200)