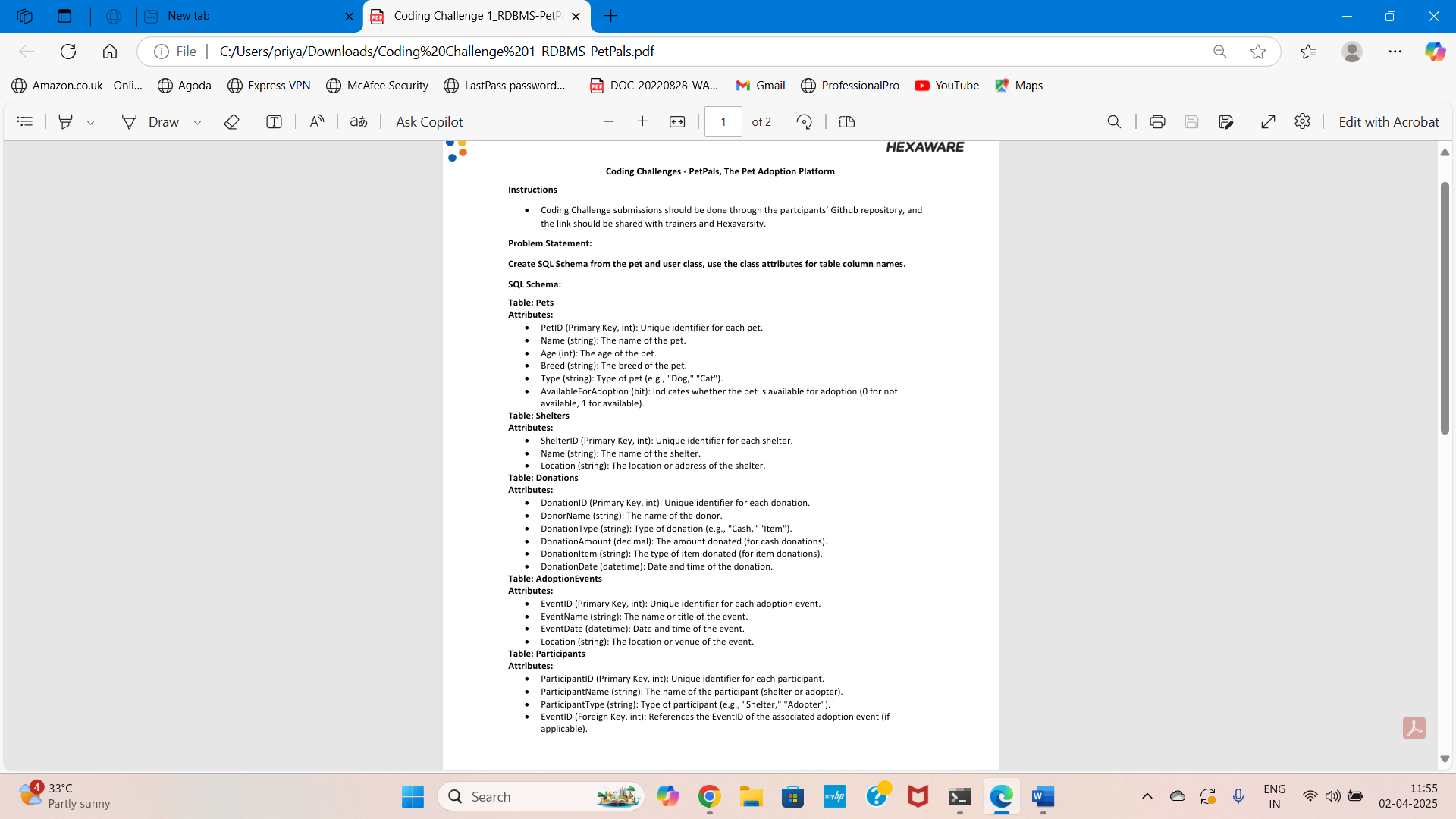
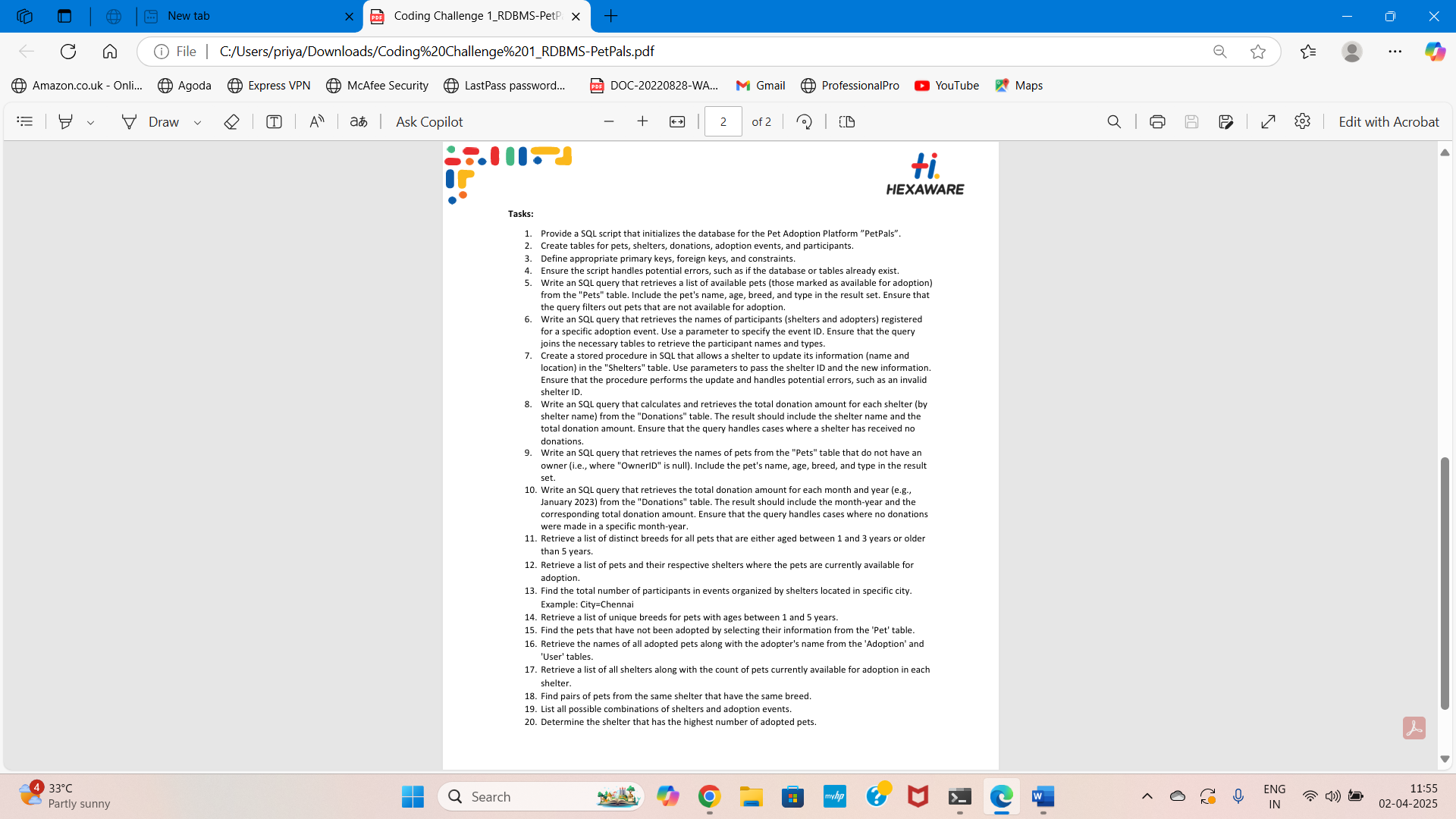
Coding Challenges - PetPals, The Pet Adoption Platform

Priyadarshini S





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

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

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

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

mysql> Create Database Petpals;

Query OK, 1 row affected (0.06 sec)

mysql> Use PetPals;

Database changed

mysql> CREATE TABLE Pets (

-> PetID INT PRIMARY KEY AUTO\_INCREMENT,

-> Name VARCHAR(255) NOT NULL,

-> Age INT,

-> Breed VARCHAR(255),

-> Type VARCHAR(50),

-> AvailableForAdoption BIT DEFAULT 1

-> );

Query OK, 0 rows affected (0.13 sec)

mysql> desc Pets;

+----------------------+--------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+----------------------+--------------+------+-----+---------+----------------+

| PetID | int | NO | PRI | NULL | auto\_increment |

| Name | varchar(255) | NO | | NULL | |

| Age | int | YES | | NULL | |

| Breed | varchar(255) | YES | | NULL | |

| Type | varchar(50) | YES | | NULL | |

| AvailableForAdoption | bit(1) | YES | | b'1' | |

+----------------------+--------------+------+-----+---------+----------------+

6 rows in set (0.03 sec)

mysql> CREATE TABLE Shelters (

-> ShelterID INT PRIMARY KEY AUTO\_INCREMENT,

-> Name VARCHAR(255) NOT NULL,

-> Location VARCHAR(255) NOT NULL

-> );

Query OK, 0 rows affected (0.03 sec)

mysql> desc Shelters;

+-----------+--------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-----------+--------------+------+-----+---------+----------------+

| ShelterID | int | NO | PRI | NULL | auto\_increment |

| Name | varchar(255) | NO | | NULL | |

| Location | varchar(255) | NO | | NULL | |

+-----------+--------------+------+-----+---------+----------------+

3 rows in set (0.00 sec)

mysql> CREATE TABLE Donations (

-> DonationID INT PRIMARY KEY AUTO\_INCREMENT,

-> DonorName VARCHAR(255) NOT NULL,

-> DonationType VARCHAR(50),

-> DonationAmount DECIMAL(10,2),

-> DonationItem VARCHAR(255),

-> DonationDate DATETIME DEFAULT CURRENT\_TIMESTAMP

-> );

Query OK, 0 rows affected (0.03 sec)

mysql> desc Donations;

+----------------+---------------+------+-----+-------------------+-------------------+

| Field | Type | Null | Key | Default | Extra |

+----------------+---------------+------+-----+-------------------+-------------------+

| DonationID | int | NO | PRI | NULL | auto\_increment |

| DonorName | varchar(255) | NO | | NULL | |

| DonationType | varchar(50) | YES | | NULL | |

| DonationAmount | decimal(10,2) | YES | | NULL | |

| DonationItem | varchar(255) | YES | | NULL | |

| DonationDate | datetime | YES | | CURRENT\_TIMESTAMP | DEFAULT\_GENERATED |

+----------------+---------------+------+-----+-------------------+-------------------+

6 rows in set (0.00 sec)

mysql> CREATE TABLE AdoptionEvents (

-> EventID INT PRIMARY KEY AUTO\_INCREMENT,

-> EventName VARCHAR(255) NOT NULL,

-> EventDate DATETIME NOT NULL,

-> Location VARCHAR(255) NOT NULL

-> );

Query OK, 0 rows affected (0.03 sec)

mysql> desc AdoptionEvents;

+-----------+--------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-----------+--------------+------+-----+---------+----------------+

| EventID | int | NO | PRI | NULL | auto\_increment |

| EventName | varchar(255) | NO | | NULL | |

| EventDate | datetime | NO | | NULL | |

| Location | varchar(255) | NO | | NULL | |

+-----------+--------------+------+-----+---------+----------------+

4 rows in set (0.00 sec)

mysql> CREATE TABLE Participants (

-> ParticipantID INT PRIMARY KEY AUTO\_INCREMENT,

-> ParticipantName VARCHAR(255) NOT NULL,

-> ParticipantType ENUM('Shelter', 'Adopter') NOT NULL,

-> EventID INT,

-> FOREIGN KEY (EventID) REFERENCES AdoptionEvents(EventID)

-> );

Query OK, 0 rows affected (0.06 sec)

mysql> desc Participants;

+-----------------+---------------------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-----------------+---------------------------+------+-----+---------+----------------+

| ParticipantID | int | NO | PRI | NULL | auto\_increment |

| ParticipantName | varchar(255) | NO | | NULL | |

| ParticipantType | enum('Shelter','Adopter') | NO | | NULL | |

| EventID | int | YES | MUL | NULL | |

+-----------------+---------------------------+------+-----+---------+----------------+

4 rows in set (0.00 sec)

mysql> INSERT INTO Pets (Name, Age, Breed, Type, AvailableForAdoption) VALUES

-> ('Jimmy', 2, 'Labrador', 'Dog', 1),

-> ('Tommy', 4, 'Persian', 'Cat', 1),

-> ('Lisa', 3, 'Beagle', 'Dog', 0);

Query OK, 3 rows affected (0.02 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql> select \* from Pets;

+-------+-------+------+----------+------+--------------------------------------------+

| PetID | Name | Age | Breed | Type | AvailableForAdoption |

+-------+-------+------+----------+------+--------------------------------------------+

| 1 | Jimmy | 2 | Labrador | Dog | 0x01 |

| 2 | Tommy | 4 | Persian | Cat | 0x01 |

| 3 | Lisa | 3 | Beagle | Dog | 0x00 |

+-------+-------+------+----------+------+--------------------------------------------+

3 rows in set (0.00 sec)

mysql> INSERT INTO Shelters (Name, Location) VALUES

-> ('Happy Paws Shelter', 'Mumbai'),

-> ('Safe Haven', 'Chennai'),

-> ('Animal Care Center', 'Delhi');

Query OK, 3 rows affected (0.01 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql> select \* from Shelters;

+-----------+--------------------+----------+

| ShelterID | Name | Location |

+-----------+--------------------+----------+

| 1 | Happy Paws Shelter | Mumbai |

| 2 | Safe Haven | Chennai |

| 3 | Animal Care Center | Delhi |

+-----------+--------------------+----------+

3 rows in set (0.00 sec)

mysql> INSERT INTO Donations (DonorName, DonationType, DonationAmount, DonationItem, DonationDate) VALUES

-> ('Ram', 'Cash', 5000.00, NULL, NOW()),

-> ('Janu', 'Item', NULL, 'Dog Food', NOW()),

-> ('Vikram', 'Cash', 3000.00, NULL, NOW());

Query OK, 3 rows affected (0.01 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql> Select \* from Donations;

+------------+-----------+--------------+----------------+--------------+---------------------+

| DonationID | DonorName | DonationType | DonationAmount | DonationItem | DonationDate |

+------------+-----------+--------------+----------------+--------------+---------------------+

| 1 | Ram | Cash | 5000.00 | NULL | 2025-04-02 10:27:06 |

| 2 | Janu | Item | NULL | Dog Food | 2025-04-02 10:27:06 |

| 3 | Vikram | Cash | 3000.00 | NULL | 2025-04-02 10:27:06 |

+------------+-----------+--------------+----------------+--------------+---------------------+

3 rows in set (0.00 sec)

mysql> INSERT INTO AdoptionEvents (EventName, EventDate, Location) VALUES

-> ('Mega Adoption Drive', '2025-04-15 10:00:00', 'Delhi'),

-> ('Pet Love Fest', '2025-05-10 11:00:00', 'Bangalore'),

-> ('Rescue Meet', '2025-06-20 09:30:00', 'Chennai');

Query OK, 3 rows affected (0.01 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql> select \* from AdoptionEvents;

+---------+---------------------+---------------------+-----------+

| EventID | EventName | EventDate | Location |

+---------+---------------------+---------------------+-----------+

| 1 | Mega Adoption Drive | 2025-04-15 10:00:00 | Delhi |

| 2 | Pet Love Fest | 2025-05-10 11:00:00 | Bangalore |

| 3 | Rescue Meet | 2025-06-20 09:30:00 | Chennai |

+---------+---------------------+---------------------+-----------+

3 rows in set (0.00 sec)

mysql> INSERT INTO Participants (ParticipantName, ParticipantType, EventID) VALUES

-> ('Rohan', 'Adopter', 1),

-> ('Priya', 'Adopter', 2),

-> ('Safe Haven', 'Shelter', 3);

Query OK, 3 rows affected (0.01 sec)

Records: 3 Duplicates: 0 Warnings: 0

mysql> select \* from Participants;

+---------------+-----------------+-----------------+---------+

| ParticipantID | ParticipantName | ParticipantType | EventID |

+---------------+-----------------+-----------------+---------+

| 1 | Rohan | Adopter | 1 |

| 2 | Priya | Adopter | 2 |

| 3 | Safe Haven | Shelter | 3 |

+---------------+-----------------+-----------------+---------+

3 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.

mysql> SELECT Name, Age, Breed, Type FROM Pets WHERE AvailableForAdoption = 1;

+-------+------+----------+------+

| Name | Age | Breed | Type |

+-------+------+----------+------+

| Jimmy | 2 | Labrador | Dog |

| Tommy | 4 | Persian | Cat |

+-------+------+----------+------+

2 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.

mysql> SELECT p.ParticipantName, p.ParticipantType

-> FROM Participants p

-> JOIN AdoptionEvents e ON p.EventID = e.EventID

-> WHERE p.EventID = 1;

+-----------------+-----------------+

| ParticipantName | ParticipantType |

+-----------------+-----------------+

| Rohan | Adopter |

+-----------------+-----------------+

1 row in set (0.01 sec)

mysql> select p.participantname, p.participanttype

-> from participants p

-> join adoptionevents e on p.eventid = e.eventid

-> where p.eventid = 2;

+-----------------+-----------------+

| participantname | participanttype |

+-----------------+-----------------+

| Priya | Adopter |

+-----------------+-----------------+

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.

mysql> update shelters

-> set name = 'happy tails shelter', location = 'hyderabad'

-> where shelterid = 1;

Query OK, 1 row affected (0.01 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> select \* from Shelters;

+-----------+---------------------+-----------+

| ShelterID | Name | Location |

+-----------+---------------------+-----------+

| 1 | happy tails shelter | hyderabad |

| 2 | Safe Haven | Chennai |

| 3 | Animal Care Center | Delhi |

+-----------+---------------------+-----------+

3 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.

mysql> select s.name, coalesce(sum(d.donationamount), 0) as total\_donations

-> from shelters s

-> left join donations d on s.shelterid = d.donationid

-> group by s.name;

+---------------------+-----------------+

| name | total\_donations |

+---------------------+-----------------+

| happy tails shelter | 5000.00 |

| Safe Haven | 0.00 |

| Animal Care Center | 3000.00 |

+---------------------+-----------------+

3 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.

mysql> select name, age, breed, type

-> from pets

-> where ownerid is null;

+-------+------+---------+------+

| name | age | breed | type |

+-------+------+---------+------+

| Tommy | 4 | Persian | Cat |

| Lisa | 3 | Beagle | Dog |

+-------+------+---------+------+

2 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.

mysql> select date\_format(donationdate, '%M %Y') as month\_year,

-> coalesce(sum(donationamount), 0) as total\_donations

-> from donations

-> group by month\_year

-> order by min(donationdate);

+------------+-----------------+

| month\_year | total\_donations |

+------------+-----------------+

| April 2025 | 8000.00 |

+------------+-----------------+

1 row in set (0.00 sec)

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

mysql> SELECT DISTINCT Breed

-> FROM Pets

-> WHERE (Age BETWEEN 1 AND 3) OR (Age > 5);

+----------+

| Breed |

+----------+

| Labrador |

| Beagle |

+----------+

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

mysql> SELECT e.location, COUNT(\*) AS total\_participants

-> FROM participants p

-> JOIN adoptionevents e ON p.eventid = e.eventid

-> WHERE e.location = 'Chennai'

-> GROUP BY e.location;

+----------+--------------------+

| location | total\_participants |

+----------+--------------------+

| Chennai | 1 |

+----------+--------------------+

1 row in set (0.00 sec)

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

mysql> SELECT DISTINCT breed, age

-> FROM pets

-> WHERE age BETWEEN 1 AND 5;

+----------+------+

| breed | age |

+----------+------+

| Labrador | 2 |

| Persian | 4 |

| Beagle | 3 |

+----------+------+

3 rows in set (0.00 sec)

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

mysql> SELECT \*

-> FROM pets

-> WHERE OwnerID IS NULL;

+-------+-------+------+---------+------+--------------------------------------------+---------+

| PetID | Name | Age | Breed | Type | AvailableForAdoption | OwnerID |

+-------+-------+------+---------+------+--------------------------------------------+---------+

| 2 | Tommy | 4 | Persian | Cat | 0x01 | NULL |

| 3 | Lisa | 3 | Beagle | Dog | 0x00 | NULL |

+-------+-------+------+---------+------+--------------------------------------------+---------+

2 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.

mysql> SELECT p.Name AS pet\_name, pa.ParticipantName AS adopter\_name

-> FROM pets p

-> JOIN participants pa ON p.OwnerID = pa.ParticipantID;

Empty set (0.00 sec)

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

mysql> SELECT p1.Name AS Pet1, p2.Name AS Pet2, p1.Breed

-> FROM pets p1

-> JOIN pets p2

-> ON p1.Breed = p2.Breed

-> AND p1.PetID < p2.PetID;

Empty set (0.00 sec)

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

mysql> SELECT s.Name AS ShelterName, e.EventName AS AdoptionEvent

-> FROM shelters s

-> CROSS JOIN adoptionevents e;

+---------------------+---------------------+

| ShelterName | AdoptionEvent |

+---------------------+---------------------+

| Animal Care Center | Mega Adoption Drive |

| Safe Haven | Mega Adoption Drive |

| happy tails shelter | Mega Adoption Drive |

| Animal Care Center | Pet Love Fest |

| Safe Haven | Pet Love Fest |

| happy tails shelter | Pet Love Fest |

| Animal Care Center | Rescue Meet |

| Safe Haven | Rescue Meet |

| happy tails shelter | Rescue Meet |

+---------------------+---------------------+

9 rows in set (0.00 sec)

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

mysql> SELECT s.Name AS ShelterName, COUNT(p.PetID) AS AdoptedPets

-> FROM shelters s

-> JOIN pets p ON s.ShelterID = p.ShelterID

-> WHERE p.OwnerID IS NOT NULL

-> GROUP BY s.Name

-> ORDER BY AdoptedPets DESC

-> LIMIT 1;

+---------------------+-------------+

| ShelterName | AdoptedPets |

+---------------------+-------------+

| happy tails shelter | 1 |

+---------------------+-------------+

1 row in set (0.00 sec)