**Coding Challenges - PetPals, The Pet Adoption Platform**  
  
1. Provide a SQL script that initializes the database for the Pet Adoption Platform ”PetPals”.

--create database

**create database PetPals**

use 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.

create table tbl\_pets (

petid int primary key identity(1,1),

name varchar(200) not null,

age int not null check (age >= 0),

breed varchar(200) not null,

type varchar(100) not null,

availableforadoption int not null

);

create table tbl\_shelters (

shelterid int primary key identity(1,1),

name varchar(200) not null unique,

location varchar(500) not null

);

create table tbl\_donations (

donationid int primary key identity(1,1),

donorname varchar(200) not null,

donationtype varchar(100) not null check (donationtype in ('Cash', 'Item')),

donationamount decimal(10,2) check (donationamount >= 0),

donationitem varchar(200),

donationdate datetime not null

);

create table tbl\_adoptionevents (

eventid int primary key identity(1,1),

eventname varchar(200) not null unique,

eventdate datetime not null,

location varchar(500) not null

);

create table tbl\_participants (

participantid int primary key identity(1,1),

participantname varchar(200) not null,

participanttype varchar(100) not null check (participanttype in ('Shelter', 'Adopter')),

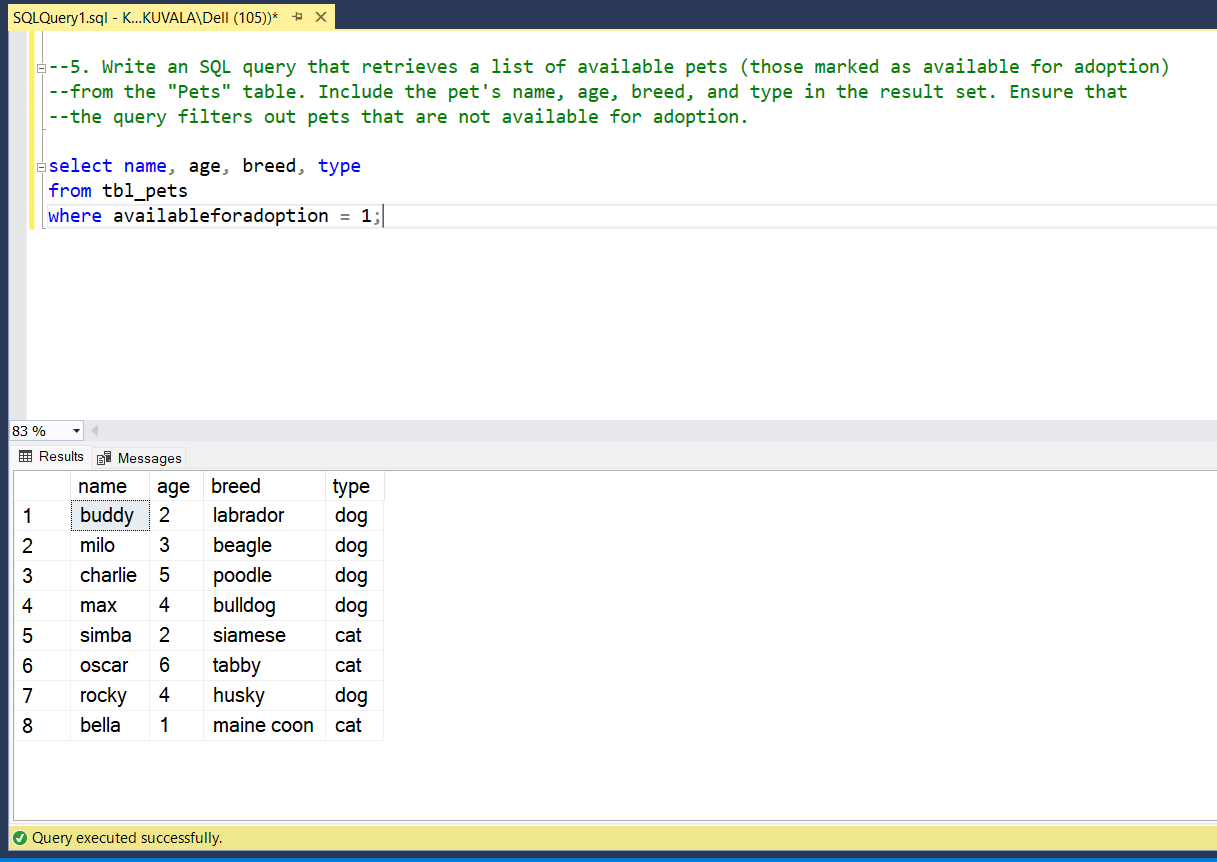
eventid int,

constraint fk\_participant\_event foreign key (eventid) references tbl\_adoptionevents(eventid)

);

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.

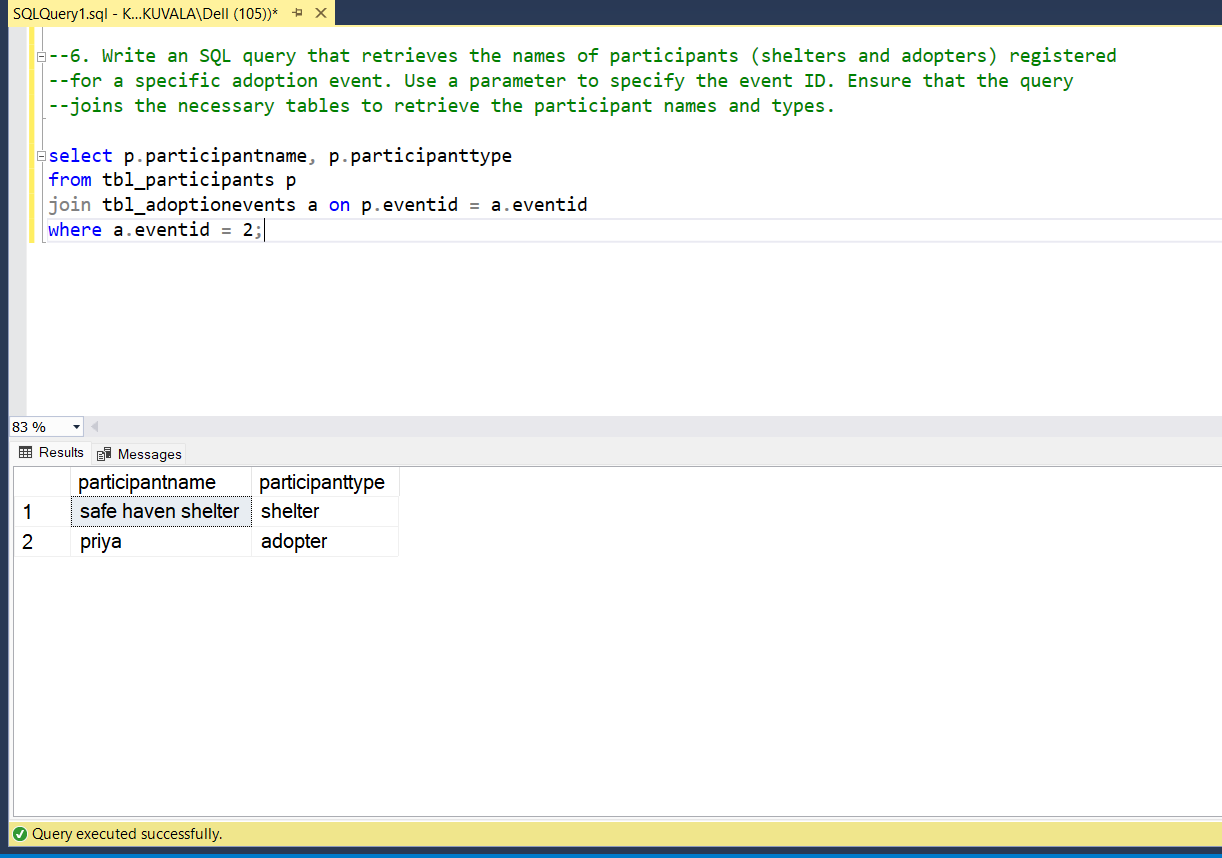
**select name, age, breed, type from tbl\_pets where availableforadoption = 1;**



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.

**select p.participantname, p.participanttype from tbl\_participants p**

**join tbl\_adoptionevents a on p.eventid = a.eventid where a.eventid = 2;**

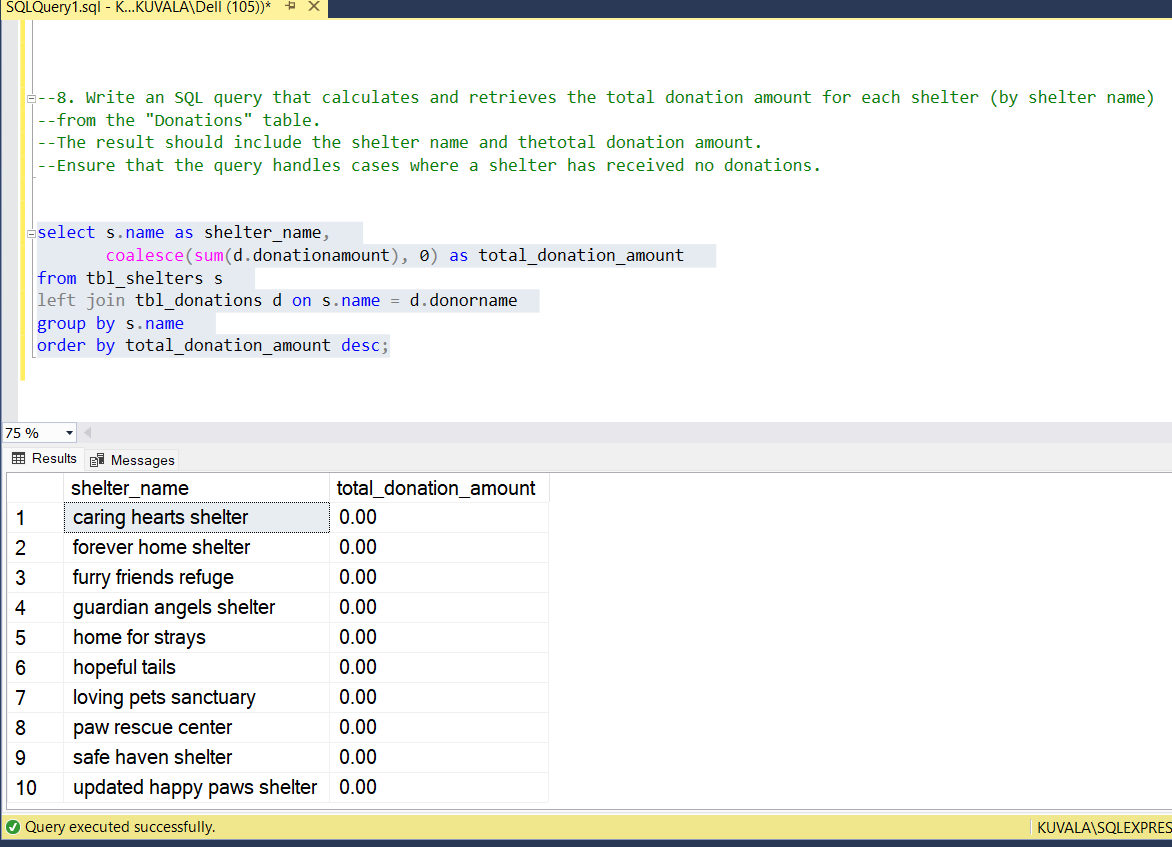


8. Write an SQL query that calculates and retrieves the total donation amount for each shelter (by shelter name) from the "Donations" table. The result should include the shelter name and thetotal donation amount. Ensure that the query handles cases where a shelter has received no donations.

**select s.name as shelter\_name, coalesce(sum(d.donationamount), 0) as total\_donation\_amount**

**from tbl\_shelters s left join tbl\_donations d on s.name = d.donorname**

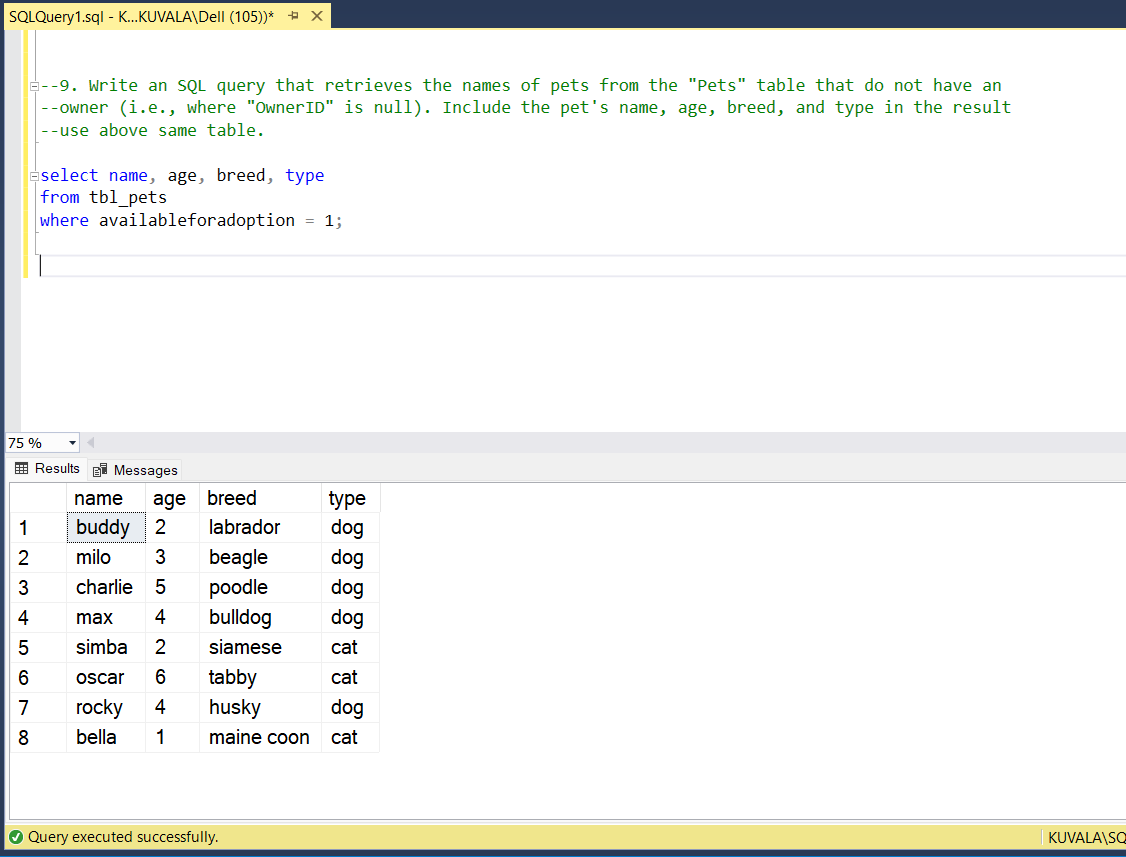
**group by s.name order by total\_donation\_amount desc;**



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 resultuse above same table.

[As there no separate adoption table with owner\_Id ,I assume that when availableforadoption = 1 which don’t have an owner and query goes like this below]

**select name, age, breed, type from tbl\_pets where availableforadoption = 1;**

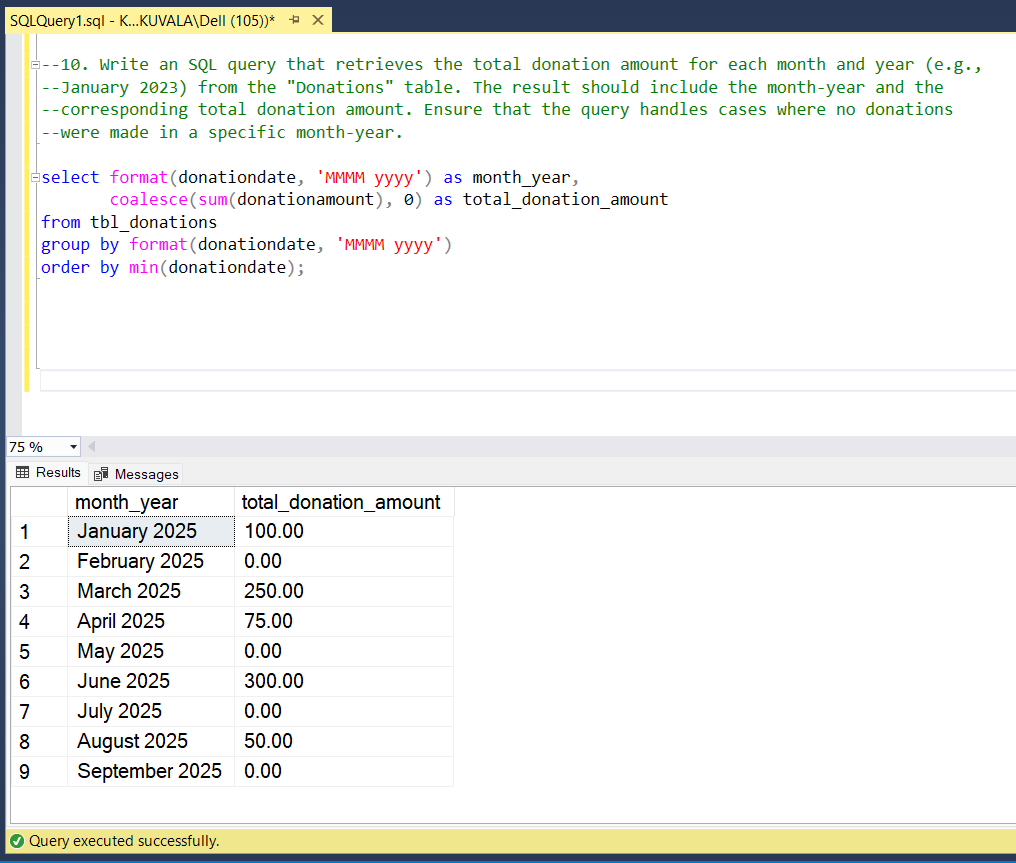
****

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.

**select format(donationdate, 'MMMM yyyy') as month\_year, coalesce(sum(donationamount), 0) as**

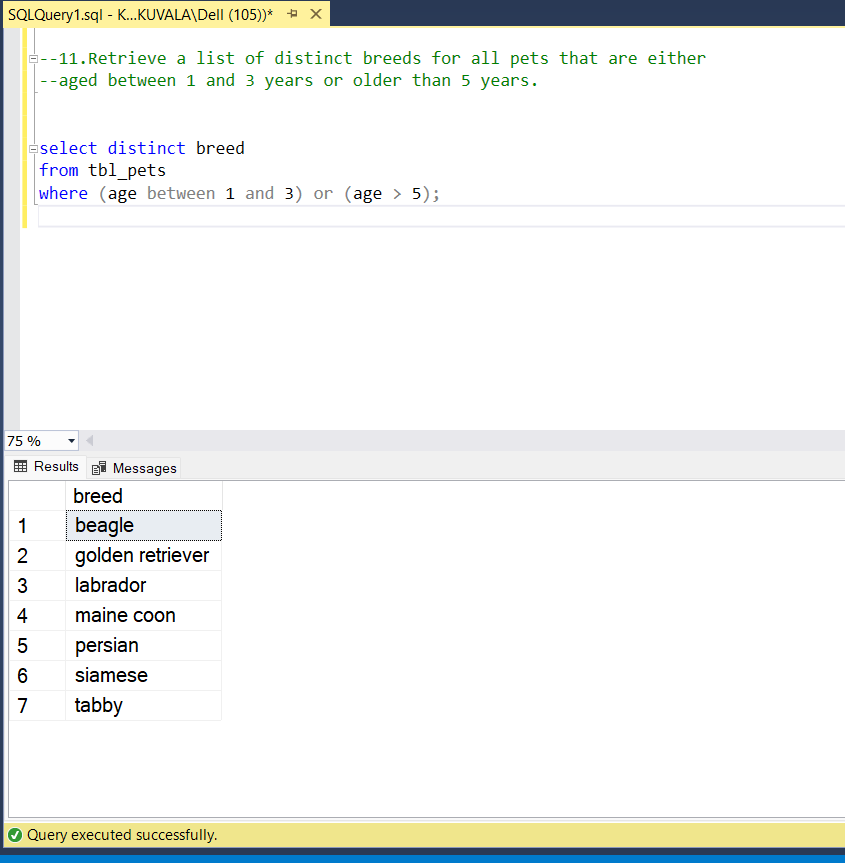
**total\_donation\_amount from tbl\_donations group by format(donationdate, 'MMMM yyyy')**

**order by min(donationdate);**



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

**select distinct breed from tbl\_pets where (age between 1 and 3) or (age > 5);**

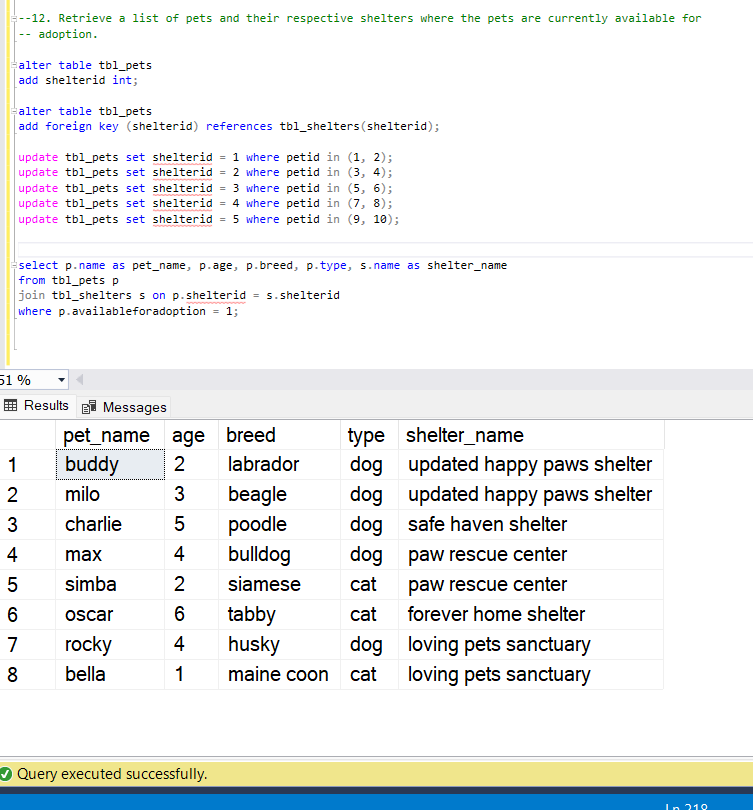


12. Retrieve a list of pets and their respective shelters where the pets are currently available for adoption.

[As currect schema doesn’t have **link** between tbl\_pets and tbl\_shelters,I modify the table then written the query]

**select p.name as pet\_name, p.age, p.breed, p.type, s.name as shelter\_name from tbl\_pets p**

**join tbl\_shelters s on p.shelterid = s.shelterid where p.availableforadoption = 1;**

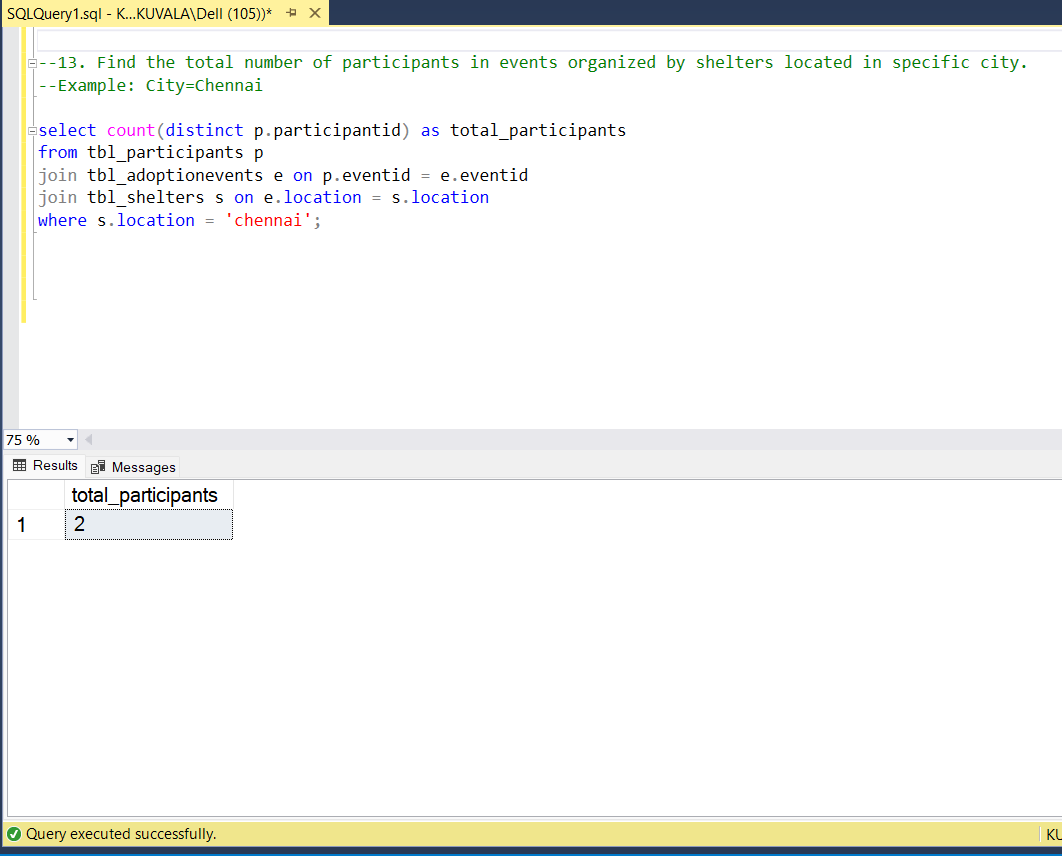


13. Find the total number of participants in events organized by shelters located in specific city.Example: City=Chennai.

**select count(distinct p.participantid) as total\_participants**

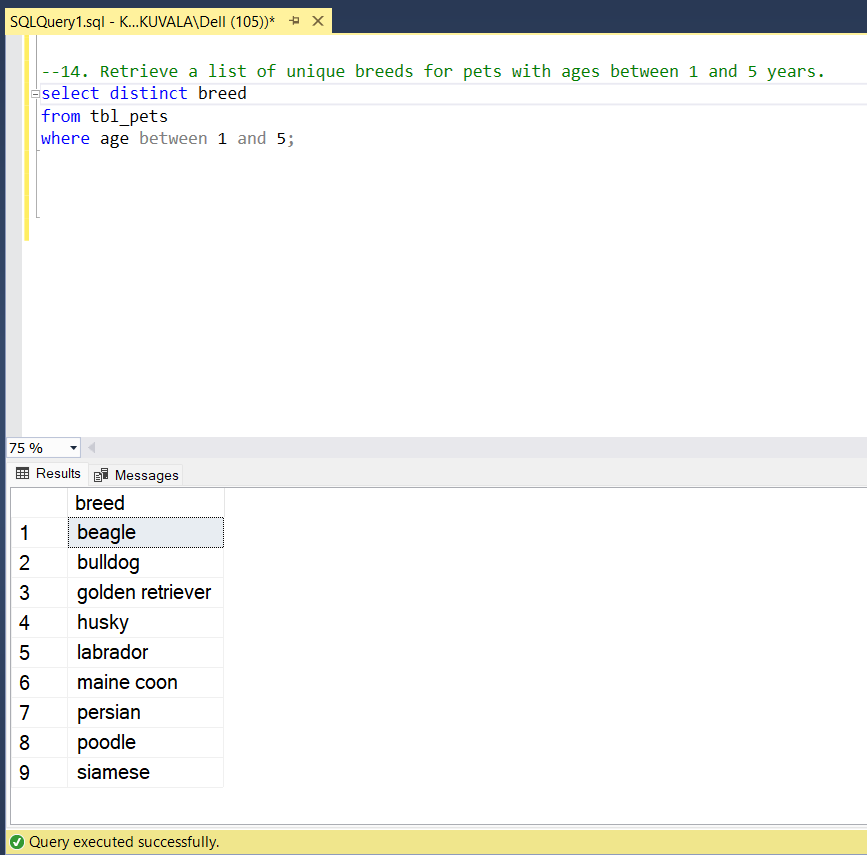
**from tbl\_participants p join tbl\_adoptionevents e on p.eventid = e.eventid**

**join tbl\_shelters s on e.location = s.location where s.location = 'chennai';**

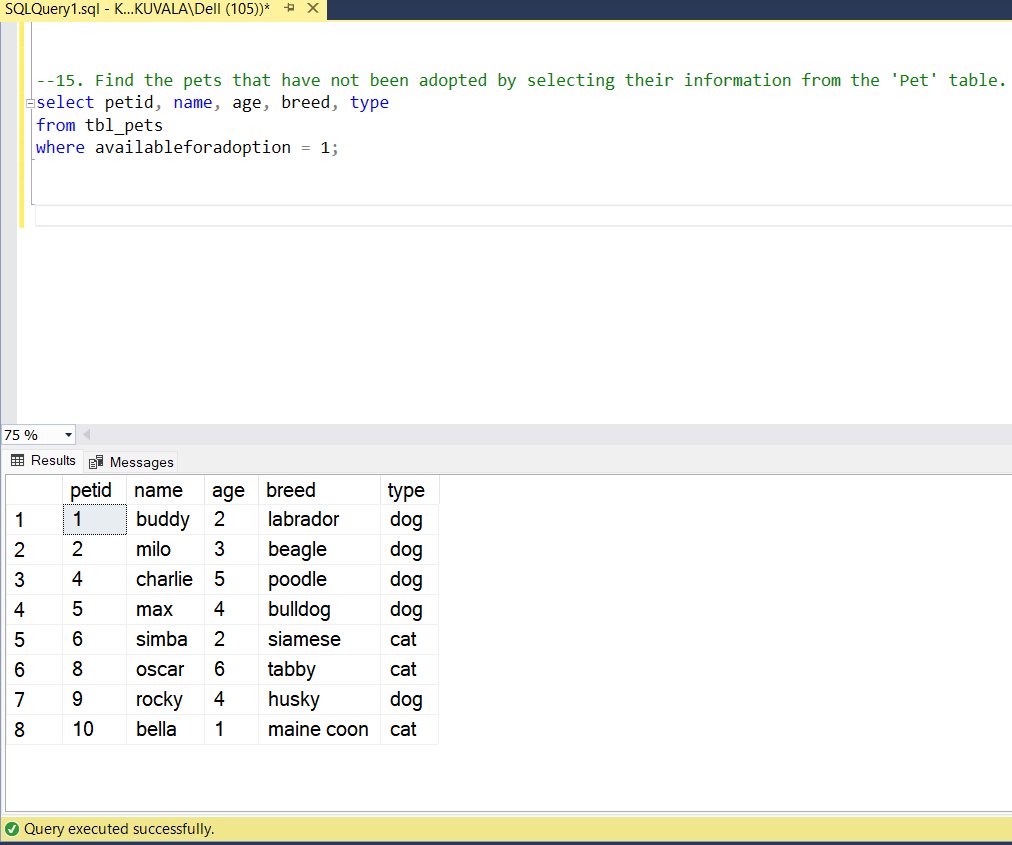


14. Retrieve a list of unique breeds for pets with ages between 1 and 5 years.

**select distinct breed from tbl\_pets where age between 1 and 5;**



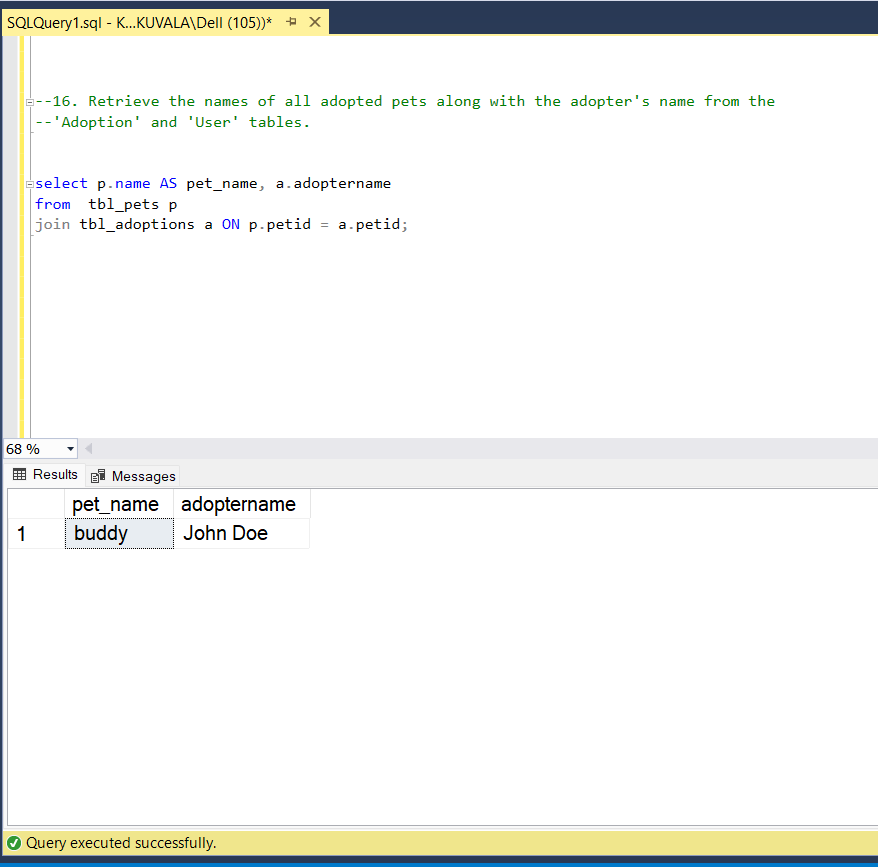
**15. Find the pets that have not been adopted by selecting their information from the 'Pet' table.**

select petid, name, age, breed, type from tbl\_pets where availableforadoption = 1;  


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

[Since schema doesn’t include **adoption records** or a **user table** in one table (which store adopter details) so to perform this query I have created an table linking both as tlb\_adoption then query goes like this ..]

**select p.name as pet\_name, a.adoptername from tbl\_pets p join tbl\_adoptions a on p.petid = a.petid;**

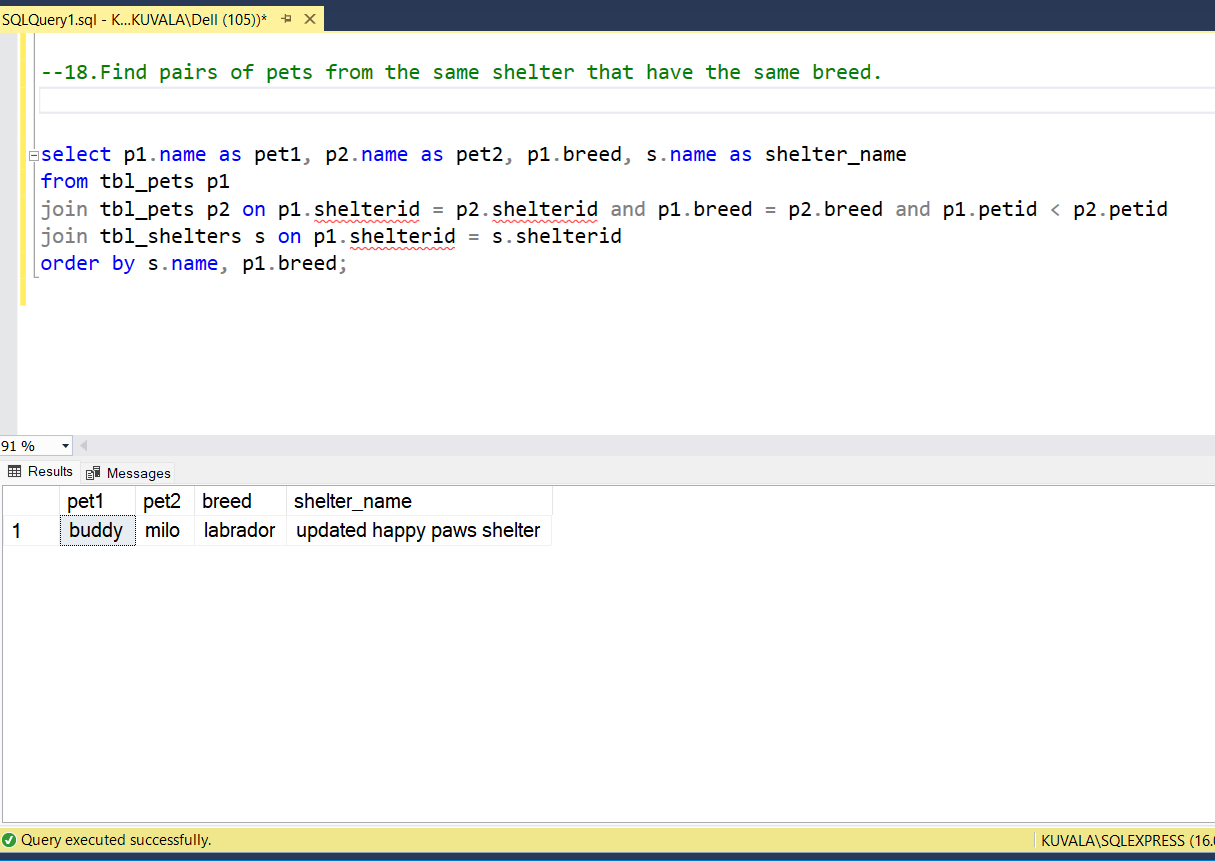
****

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

**select p1.name as pet1, p2.name as pet2, p1.breed, s.name as shelter\_name**

**from tbl\_pets p1 join tbl\_pets p2 on p1.shelterid = p2.shelterid and p1.breed = p2.breed and p1.petid < p2.petid**

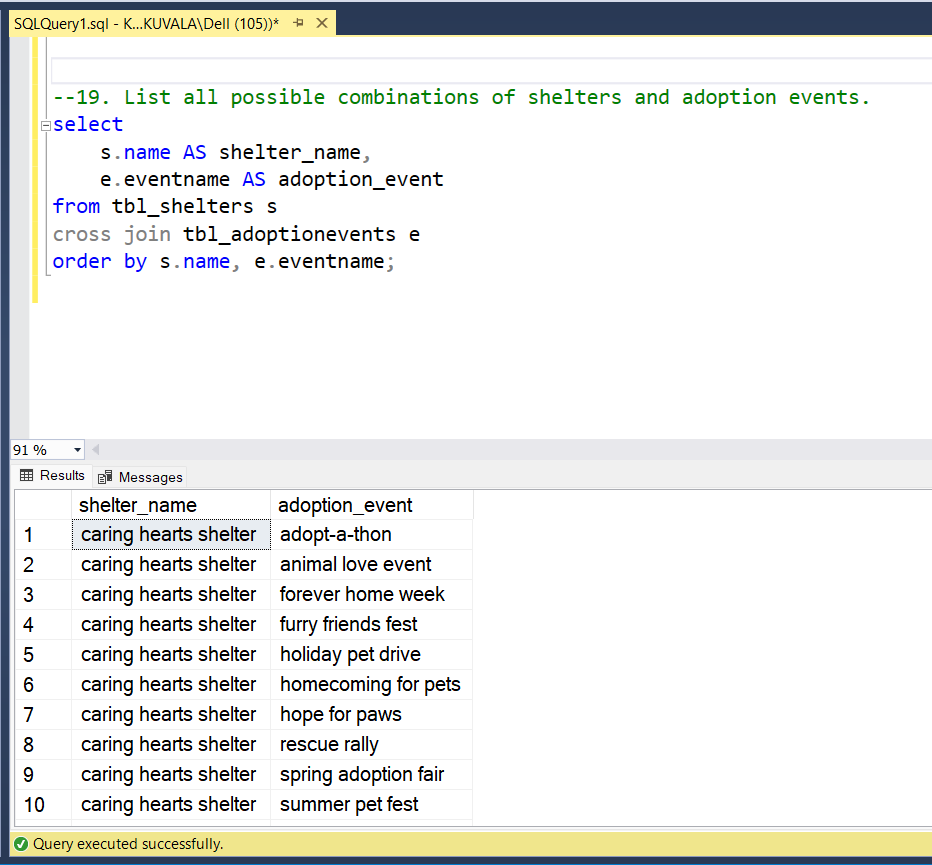
**join tbl\_shelters s on p1.shelterid = s.shelterid order by s.name, p1.breed;**



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

**select s.name AS shelter\_name, e.eventname AS adoption\_event**

**from tbl\_shelters s cross join tbl\_adoptionevents e order by s.name, e.eventname;**

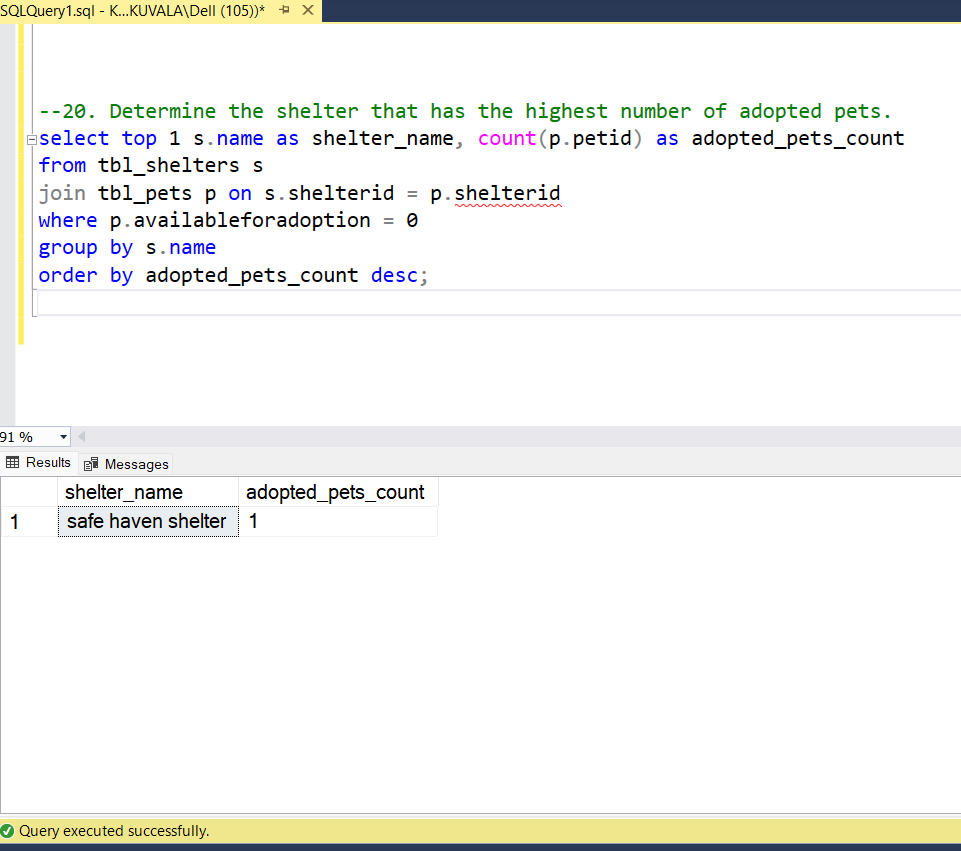


20. Determine the shelter that has the highest number of adopted pets.

**select top 1 s.name as shelter\_name, count(p.petid) as adopted\_pets\_count from tbl\_shelters s**

**join tbl\_pets p on s.shelterid = p.shelterid where p.availableforadoption = 0**

**group by s.name order by adopted\_pets\_count desc;**



End.