ANIMAL SHELTER

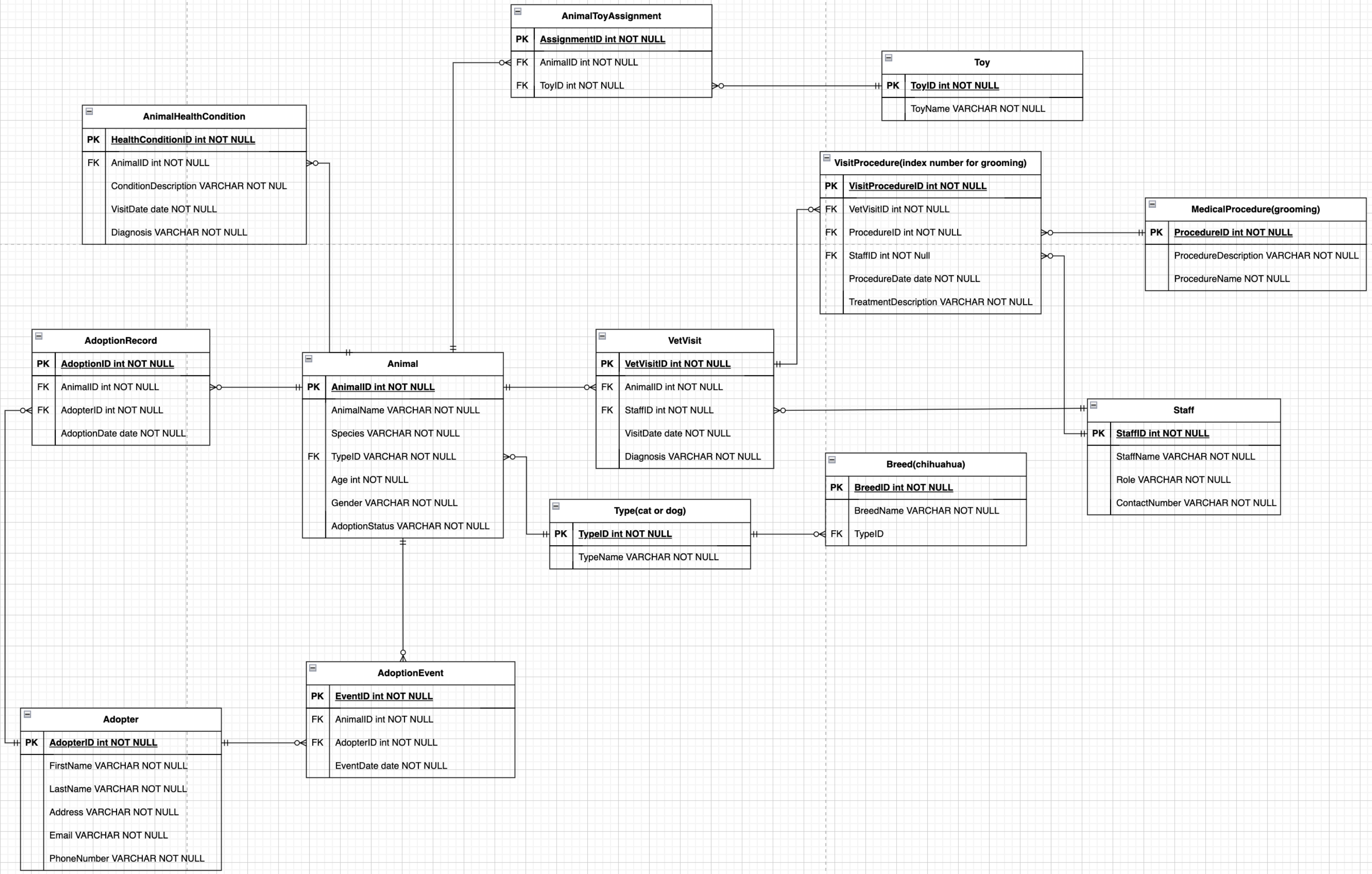
IMT 543
Final Project (Presentation)

BUSINESS PROBLEM SPACE

Shelters play a crucial role in providing a safe haven for animals

Challenges: Animal Profiles, Adoption Status, Adopter Details, and Medical History can become overwhelming.

Animal Shelter Database is needed

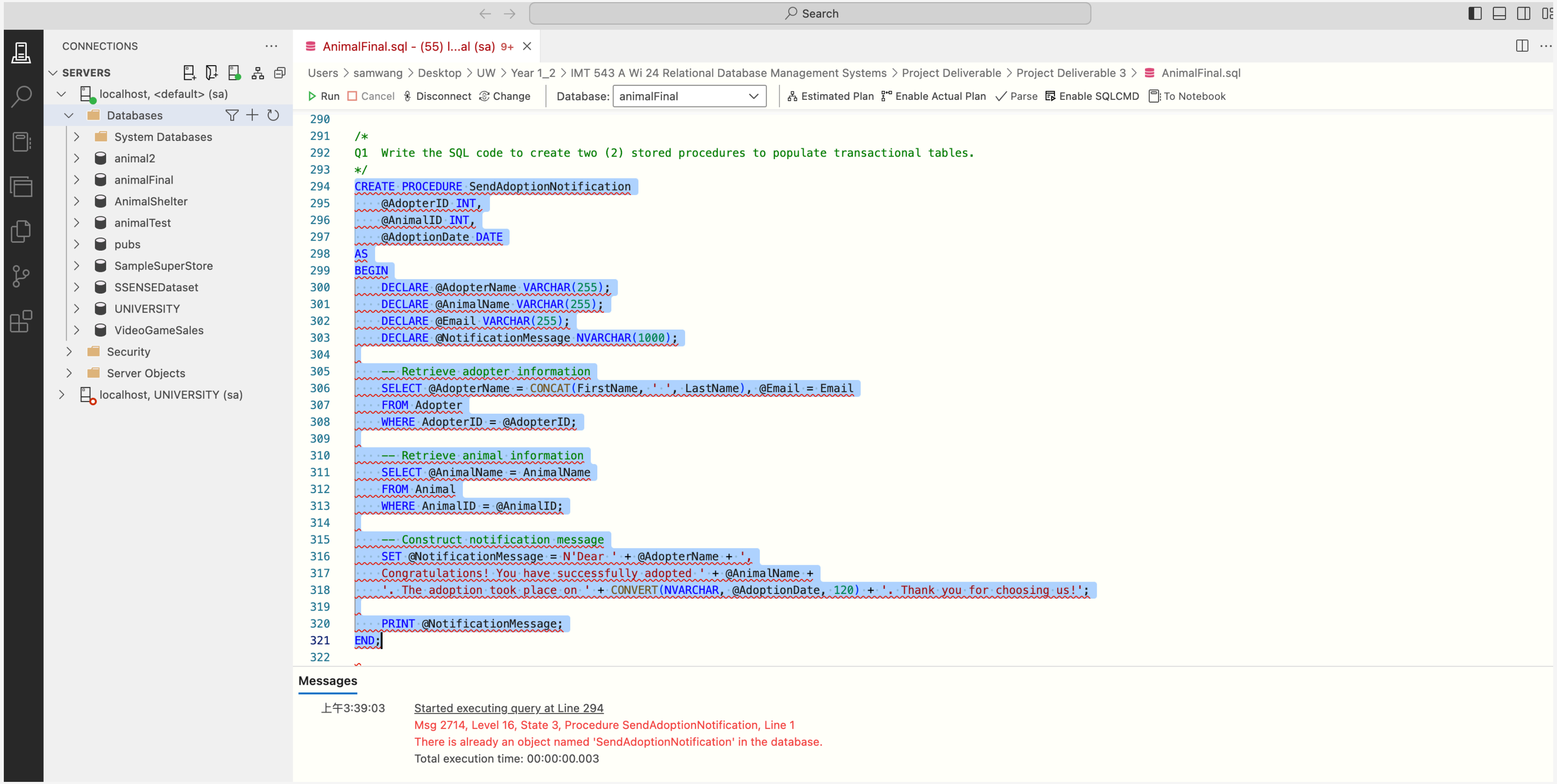

AnimalToyAssignment
PK AssignmentID int NOT NULL
FK AnimalID int NOT NULL
FK ToyID int NOT NULL
Toy
PK ToyID int NOT NULL
ToyName VARCHAR NOT NULL
AnimalHealthCondition
PK HealthConditionID int NOT NULL
FK AnimalID int NOT NULL
ConditionDescription VARCHAR NOT NUL
VisitDate date NOT NULL
Diagnosis VARCHAR NOT NULL
VisitProcedure(index number for grooming)
PK VisitProcedureID int NOT NULL
FK VetVisitID int NOT NULL
FK ProcedureID int NOT NULL
FK StaffID int NOT Null
ProcedureDate date NOT NULL
TreatmentDescription VARCHAR NOT NULL
MedicalProcedure(grooming)
PK ProcedureID int NOT NULL
ProcedureDescription VARCHAR NOT NULL
ProcedureName NOT NULL
AdoptionRecord
PK AdoptionID int NOT NULL
FK AnimalID int NOT NULL
FK AdopterID int NOT NULL
AdoptionDate date NOT NULL
Animal
PK AnimalID int NOT NULL
AnimalName VARCHAR NOT NULL
Species VARCHAR NOT NULL
FK TypeID VARCHAR NOT NULL
Age int NOT NULL
Gender VARCHAR NOT NULL
AdoptionStatus VARCHAR NOT NULL
VetVisit
PK VetVisitID int NOT NULL
FK AnimalID int NOT NULL
FK StaffID int NOT NULL
VisitDate date NOT NULL
Diagnosis VARCHAR NOT NULL
Staff
PK StaffID int NOT NULL
StaffName VARCHAR NOT NULL
Role VARCHAR NOT NULL
ContactNumber VARCHAR NOT NULL
Breed(chihuahua)
PK BreedID int NOT NULL
BreedName VARCHAR NOT NULL
FK TypeID
Type(cat or dog)
PK TypeID int NOT NULL
TypeName VARCHAR NOT NULL
AdoptionEvent
PK EventID int NOT NULL
FK AnimalID int NOT NULL
FK AdopterID int NOT NULL
EventDate date NOT NULL
Adopter
PK AdopterID int NOT NULL
FirstName VARCHAR NOT NULL
LastName VARCHAR NOT NULL
Address VARCHAR NOT NULL
Email VARCHAR NOT NULL
PhoneNumber VARCHAR NOT NULL

Why did I choose this topic?

Love animals

I aim to assist animal shelters making their data managing smoother and enhancing the overall adoption experience, both the shelter side for their operations efficiently and the adopter side

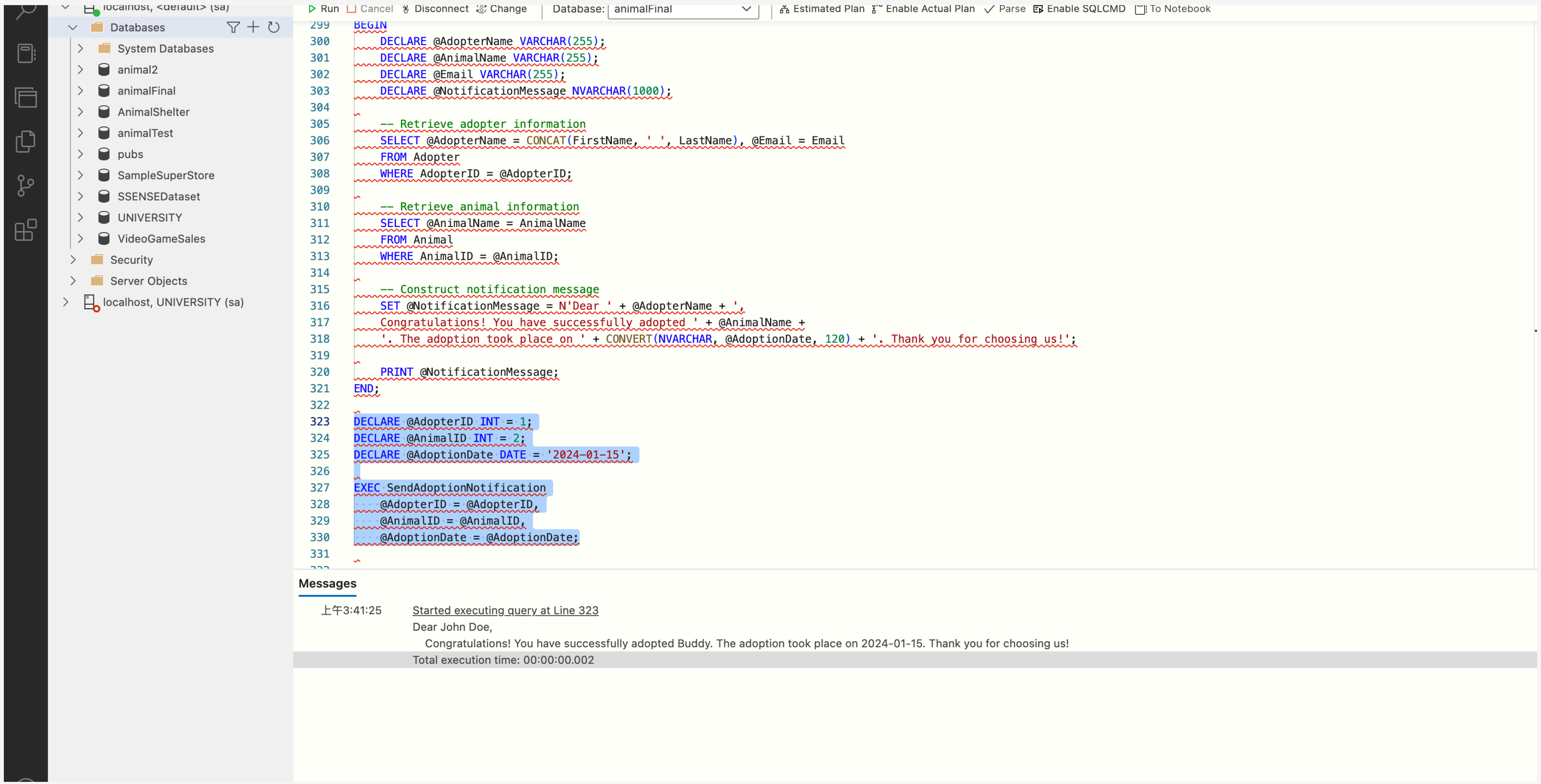
2. DDL/physical database design highlights: Procedure



```sql
/*
Q1  Write the SQL code to create two (2) stored procedures to populate transactional tables.
*/
CREATE PROCEDURE SendAdoptionNotification
    @AdopterID INT,
    @AnimalID INT,
    @AdoptionDate DATE
AS
BEGIN
    DECLARE @AdopterName VARCHAR(255);
    DECLARE @AnimalName VARCHAR(255);
    DECLARE @Email VARCHAR(255);
    DECLARE @NotificationMessage NVARCHAR(1000);

    -- Retrieve adopter information
    SELECT @AdopterName = CONCAT(FirstName, ' ', LastName), @Email = Email
    FROM Adopter
    WHERE AdopterID = @AdopterID;

    -- Retrieve animal information
    SELECT @AnimalName = AnimalName
    FROM Animal
    WHERE AnimalID = @AnimalID;

    -- Construct notification message
    SET @NotificationMessage = N'Dear ' + @AdopterName + ',
    Congratulations! You have successfully adopted ' + @AnimalName +
    '. The adoption took place on ' + CONVERT(NVARCHAR, @AdoptionDate, 120) + '. Thank you for choosing us!';

    PRINT @NotificationMessage;
END;
```

```
上午3:39:03    Started executing query at Line 294
               Msg 2714, Level 16, State 3, Procedure SendAdoptionNotification, Line 1
               There is already an object named 'SendAdoptionNotification' in the database.
               Total execution time: 00:00:00.003
```

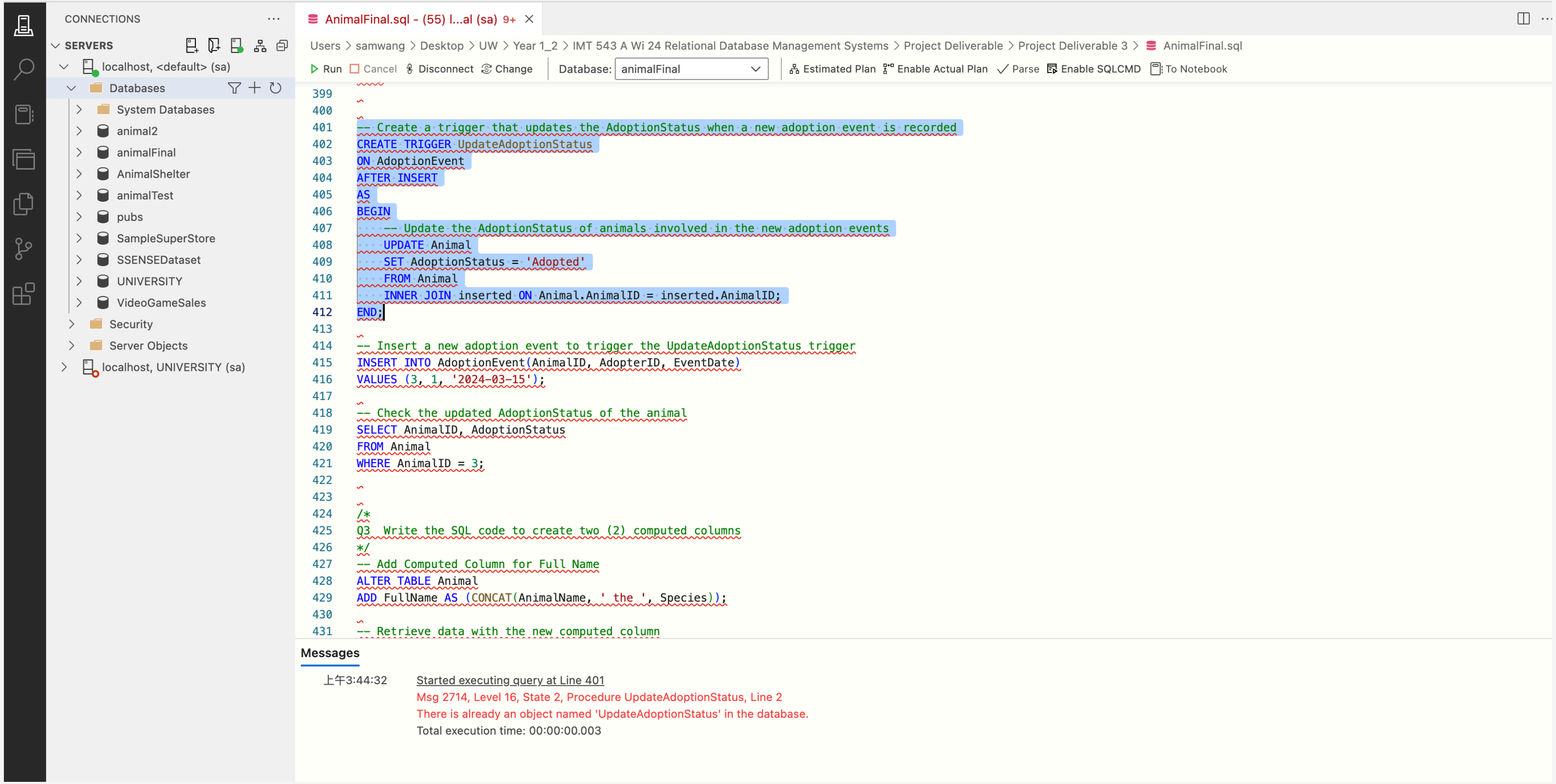
2. DDL/physical database design highlights: Procedure



```sql
BEGIN
    DECLARE @AdopterName VARCHAR(255);
    DECLARE @AnimalName VARCHAR(255);
    DECLARE @Email VARCHAR(255);
    DECLARE @NotificationMessage NVARCHAR(1000);

    -- Retrieve adopter information
    SELECT @AdopterName = CONCAT(FirstName, ' ', LastName), @Email = Email
    FROM Adopter
    WHERE AdopterID = @AdopterID;

    -- Retrieve animal information
    SELECT @AnimalName = AnimalName
    FROM Animal
    WHERE AnimalID = @AnimalID;

    -- Construct notification message
    SET @NotificationMessage = N'Dear ' + @AdopterName + ',
    Congratulations! You have successfully adopted ' + @AnimalName +
    '. The adoption took place on ' + CONVERT(NVARCHAR, @AdoptionDate, 120) + '. Thank you for choosing us!';

    PRINT @NotificationMessage;
END;

DECLARE @AdopterID INT = 1;
DECLARE @AnimalID INT = 2;
DECLARE @AdoptionDate DATE = '2024-01-15';

EXEC SendAdoptionNotification
    @AdopterID = @AdopterID,
    @AnimalID = @AnimalID,
    @AdoptionDate = @AdoptionDate;
```

```
上午3:41:25    Started executing query at Line 323
               Dear John Doe,
                 Congratulations! You have successfully adopted Buddy. The adoption took place on 2024-01-15. Thank you for choosing us!
               Total execution time: 00:00:00.002
```

2. DDL/physical database design highlights: Trigger

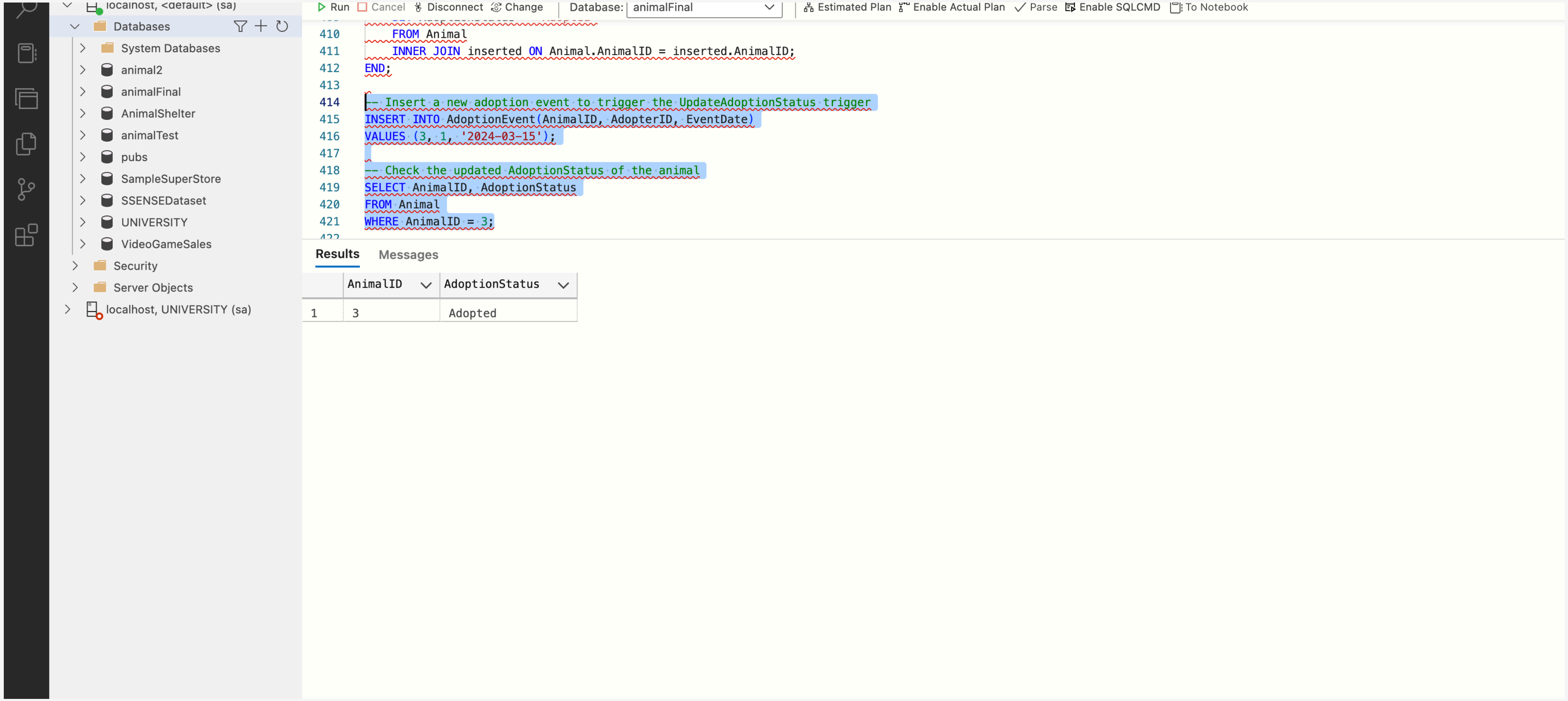


```sql
-- Create a trigger that updates the AdoptionStatus when a new adoption event is recorded
CREATE TRIGGER UpdateAdoptionStatus
ON AdoptionEvent
AFTER INSERT
AS
BEGIN
    -- Update the AdoptionStatus of animals involved in the new adoption events
    UPDATE Animal
    SET AdoptionStatus = 'Adopted'
    FROM Animal
    INNER JOIN inserted ON Animal.AnimalID = inserted.AnimalID;
END;

-- Insert a new adoption event to trigger the UpdateAdoptionStatus trigger
INSERT INTO AdoptionEvent(AnimalID, AdopterID, EventDate)
VALUES (3, 1, '2024-03-15');

-- Check the updated AdoptionStatus of the animal
SELECT AnimalID, AdoptionStatus
FROM Animal
WHERE AnimalID = 3;


/*
Q3  Write the SQL code to create two (2) computed columns
*/
-- Add Computed Column for Full Name
ALTER TABLE Animal
ADD FullName AS (CONCAT(AnimalName, ' the ', Species));

-- Retrieve data with the new computed column
```

Messages

```
上午3:44:32    Started executing query at Line 401
               Msg 2714, Level 16, State 2, Procedure UpdateAdoptionStatus, Line 2
               There is already an object named 'UpdateAdoptionStatus' in the database.
               Total execution time: 00:00:00.003
```

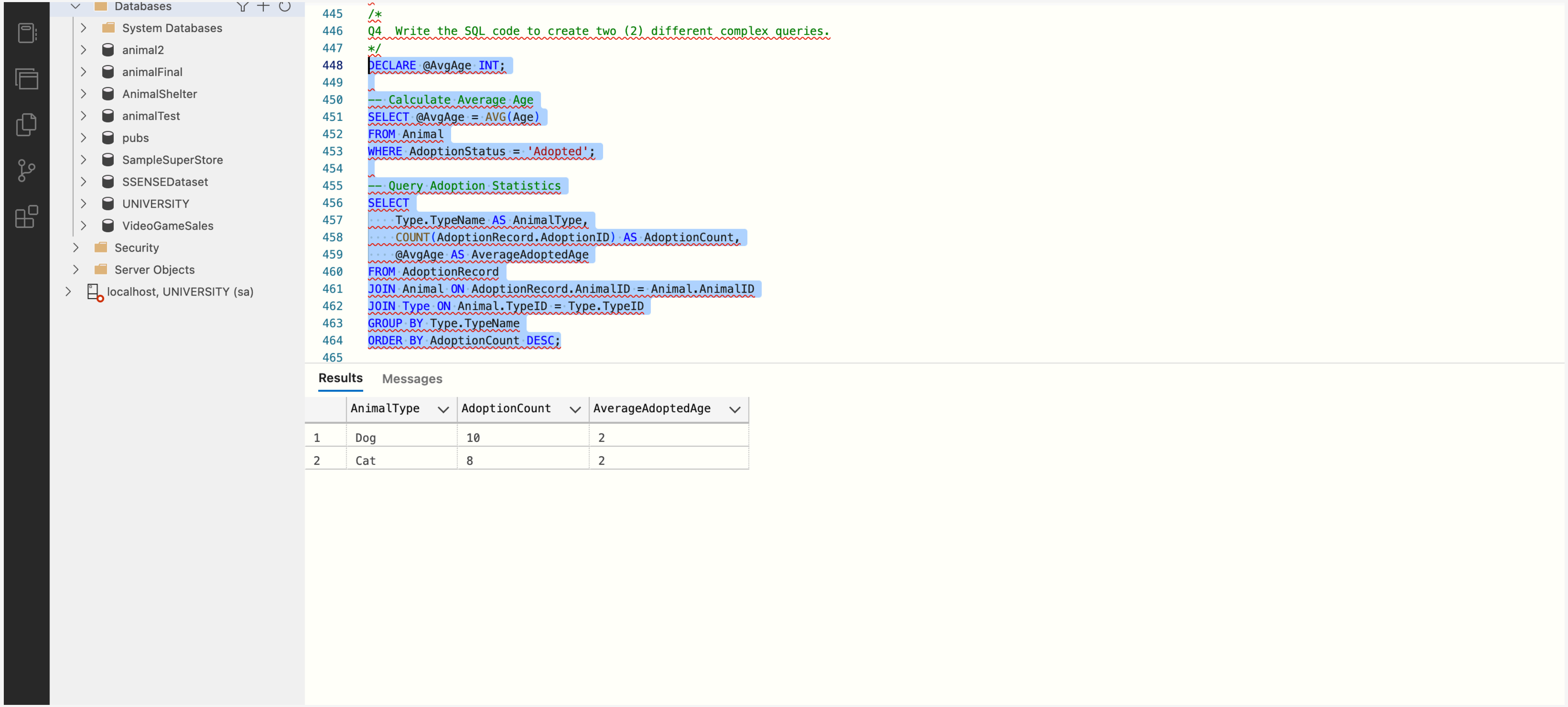
2. DDL/physical database design highlights: Trigger



```sql
    FROM Animal
    INNER JOIN inserted ON Animal.AnimalID = inserted.AnimalID;
END;

-- Insert a new adoption event to trigger the UpdateAdoptionStatus trigger
INSERT INTO AdoptionEvent(AnimalID, AdopterID, EventDate)
VALUES (3, 1, '2024-03-15');

-- Check the updated AdoptionStatus of the animal
SELECT AnimalID, AdoptionStatus
FROM Animal
WHERE AnimalID = 3;
```

	AnimalID	AdoptionStatus
1	3	Adopted

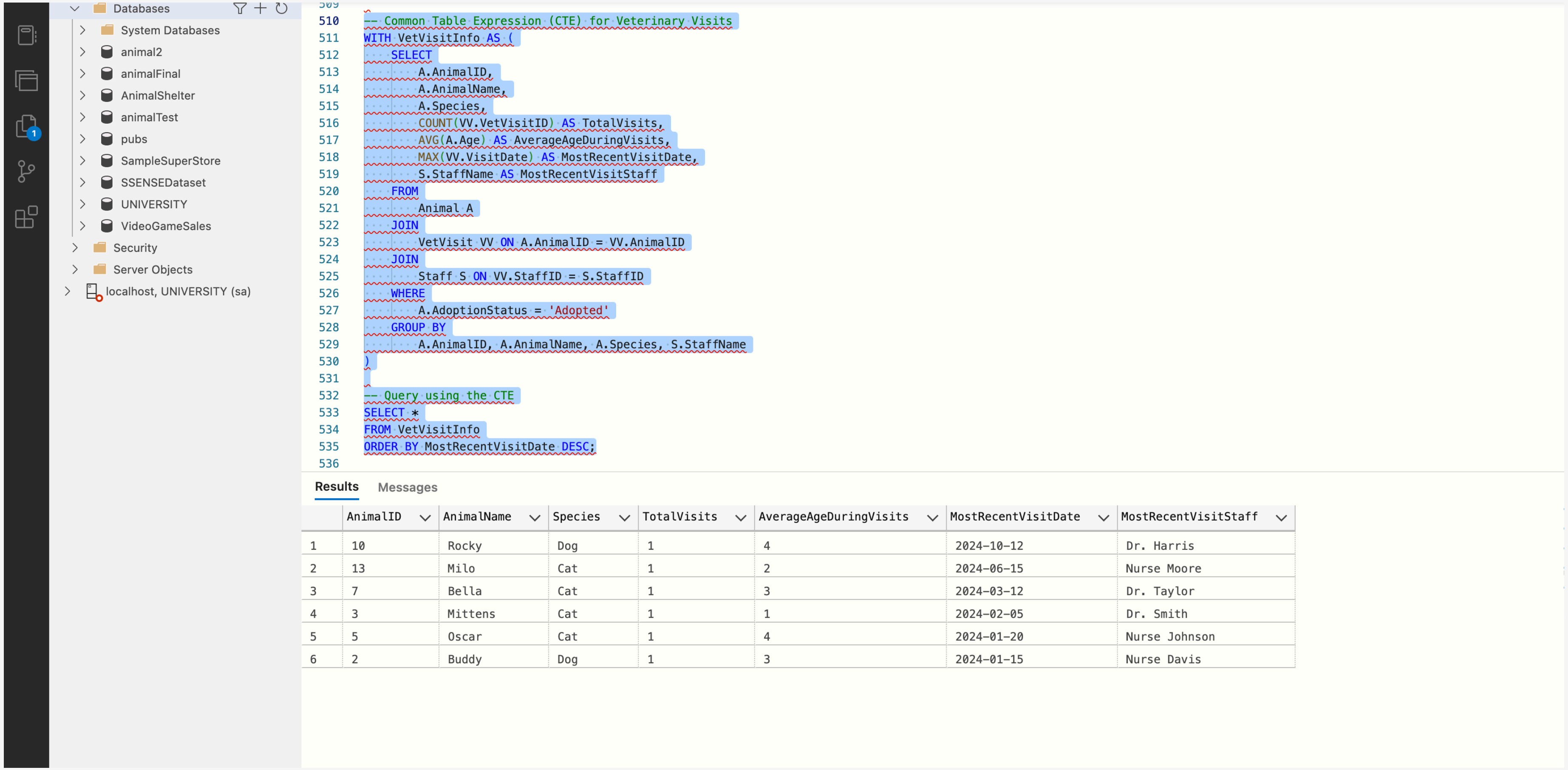
3. Retrieving data from the database (SQL)



```sql
/*
Q4  Write the SQL code to create two (2) different complex queries.
*/
DECLARE @AvgAge INT;

-- Calculate Average Age
SELECT @AvgAge = AVG(Age)
FROM Animal
WHERE AdoptionStatus = 'Adopted';

-- Query Adoption Statistics
SELECT
    Type.TypeName AS AnimalType,
    COUNT(AdoptionRecord.AdoptionID) AS AdoptionCount,
    @AvgAge AS AverageAdoptedAge
FROM AdoptionRecord
JOIN Animal ON AdoptionRecord.AnimalID = Animal.AnimalID
JOIN Type ON Animal.TypeID = Type.TypeID
GROUP BY Type.TypeName
ORDER BY AdoptionCount DESC;
```

	AnimalType	AdoptionCount	AverageAdoptedAge
1	Dog	10	2
2	Cat	8	2

3. Retrieving data from the database (SQL)



```sql
-- Common Table Expression (CTE) for Veterinary Visits
WITH VetVisitInfo AS (
    SELECT
        A.AnimalID,
        A.AnimalName,
        A.Species,
        COUNT(VV.VetVisitID) AS TotalVisits,
        AVG(A.Age) AS AverageAgeDuringVisits,
        MAX(VV.VisitDate) AS MostRecentVisitDate,
        S.StaffName AS MostRecentVisitStaff
    FROM
        Animal A
    JOIN
        VetVisit VV ON A.AnimalID = VV.AnimalID
    JOIN
        Staff S ON VV.StaffID = S.StaffID
    WHERE
        A.AdoptionStatus = 'Adopted'
    GROUP BY
        A.AnimalID, A.AnimalName, A.Species, S.StaffName
)

-- Query using the CTE
SELECT *
FROM VetVisitInfo
ORDER BY MostRecentVisitDate DESC;
```

	AnimalID	AnimalName	Species	TotalVisits	AverageAgeDuringVisits	MostRecentVisitDate	MostRecentVisitStaff
1	10	Rocky	Dog	1	4	2024-10-12	Dr. Harris
2	13	Milo	Cat	1	2	2024-06-15	Nurse Moore
3	7	Bella	Cat	1	3	2024-03-12	Dr. Taylor
4	3	Mittens	Cat	1	1	2024-02-05	Dr. Smith
5	5	Oscar	Cat	1	4	2024-01-20	Nurse Johnson
6	2	Buddy	Dog	1	3	2024-01-15	Nurse Davis

Takeaways

- Database Design: Successfully designed a relational database schema with tables for animals, adoptions, health conditions, staff, and more.
- SQL Skills: SQL commands, including CREATE TABLE, INSERT INTO, and foreign key relationships
- Trigger/Procedure Implementation: Created a trigger to automatically update animal adoption status when corresponding adoption records are inserted.
- Complex Queries and CTE: Formulated complex queries involving table joins, aggregate functions, and a Common Table Expression (CTE) to retrieve insightful information about adopted animals and their veterinary visits.