Trandafir Laura

341C3

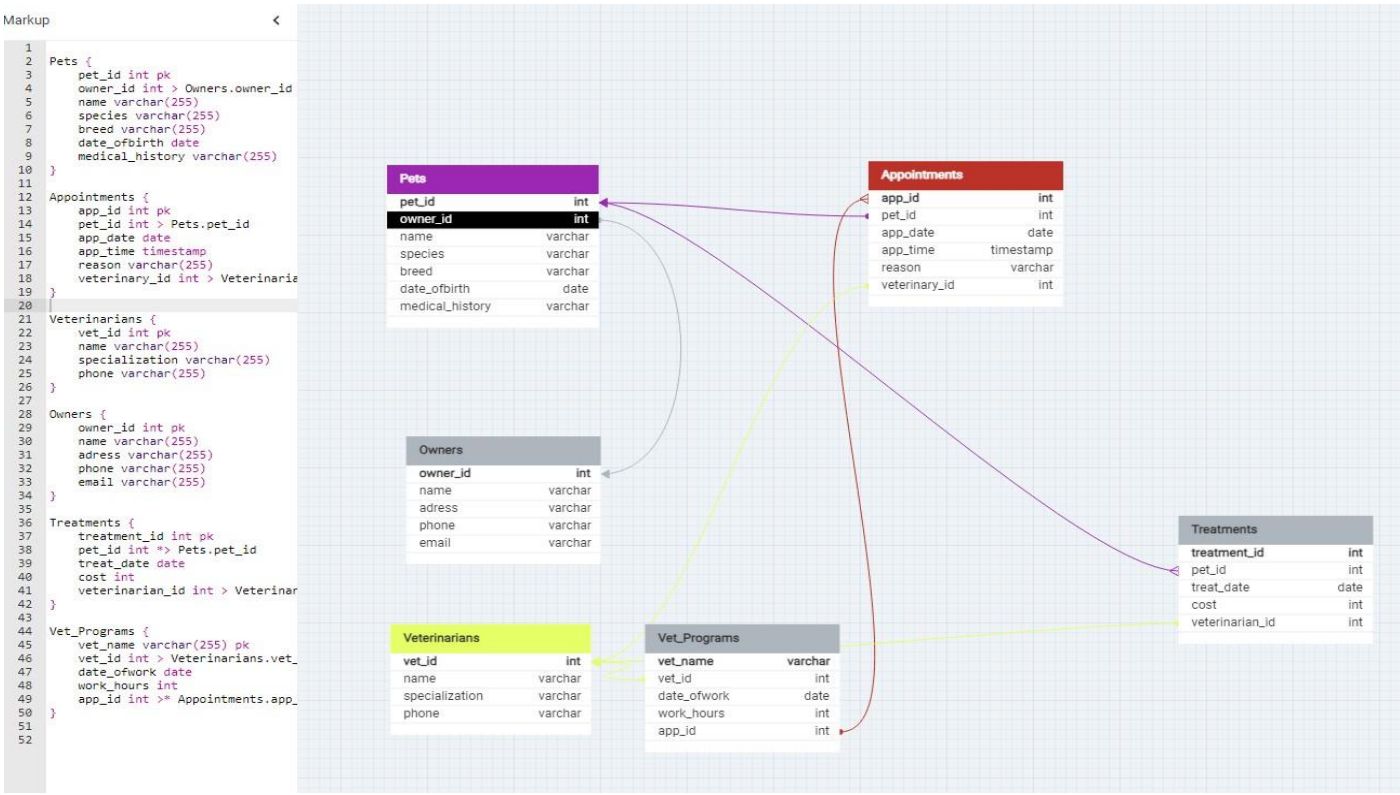
Proiect Baze de Date 2

Am ales sa proiectez baza de date pentru un cabinet veterinar ce contine informatii administrative si ajuta la sincronizarea si pastrarea datelor in sistem. Baza de date contine informatii relevante despre animel, proprietari, veterinari, programul acestora dar si despre programari si tratamente.

1.Descrierea Bazei de Date

a. Diagrama Bazei de date

Pentru implementarea diagramei bazei de date am folosit DB Deisnger, unde am definit cele 6 rabele utilizate si relatiile dintre ele.



Am folosit 6 tabele :

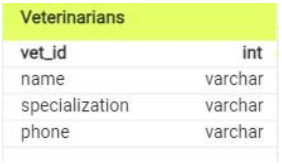
- “Pets” – unde salvez informatiile relevante pentru fiecare animalut;
- “Owners” – unde salvaez informatiile relevante despre stapanii animalelor;
- “Veterinarians” – unde salvez informatiile importante despre medicii veterinary
- “Vet_programs” - care pastreaza programul veterinarilor
- “Appointments” – pastreaza infromatiile programarilor
- “Treatments” – salvez informatii despre fiecare tratament administrat animalelor

1.2 Structura tabelelor:

Tabelul “Pets” continue:
pet_id – id unic de identificare al animalului;
owner_id – id unic pentru proprietar
name – numele animalului
species – specia animalului
breed – Rasa animalului
date_of_birth – data de nastere a micutului
medical_history – istoricul medical al animalului

Tabelul “Owners” continue:
owner_id - id unic de identificare al proprietarului
name – numele proprietarului
address – adresa proprietarului
phone – telefonul proprietarului
email – mailul proprietarului

Tabelul “Veterinarians” continue:
vet_id – id unic de identificare al veterinarului
name – numele medicului veterinary
specialization – specializarea veterinarului
phone – numarul de telefon



Tabelul Vet_programs continue:
vet_name - nume
vet_id- id medic veterinary
date_ofwork- zilele in care lucreaza
app_id- id unic al programarii pe care o are

Vet_Programs	
vet_name	varchar
vet_id	int
date_ofwork	date
work_hours	int
app_id	int

Tabelul “Treatments” continue:
treatment_id- identificator unic pentru tratament
pet_id- identificatory unic pentru pet
treat_date-data tratamentului la care a fost prescris
cost- costul tratamentului
veterinarian_id – identificator unic pentru veterinary

Treatments	
treatment_id	int
pet_id	int
treat_date	date
cost	int
veterinarian_id	int

Tabelul “Appointments” continue:
app_id – identificator unic
pet_id – identificatory unic pentru animale
app_date – data programarii
reason- motivul programarii
veterinary_id – id medic veterinary ce se va ocupa de programare

Appointments	
app_id	int
pet_id	int
app_date	date
app_time	timestamp
reason	varchar
veterinary_id	int

1.3 Descrierea constrangerilor de integritate:

Din punct de vedere al constrangerilor, la crearea tabelelor am pus drept constrangere „NOT NULL” pentru fiecare camp din interiorul unui tabel, facand astfel obligatorie completarea fiecaruia dintre acestea, astfel:

```sql
CREATE TABLE Pets (
    pet_id int NOT NULL,
    owner_id int NOT NULL,
    name varchar(255) NOT NULL,
    species varchar(255) NOT NULL,
    breed varchar(255) NOT NULL,
    date_ofbirth DATE NOT NULL,
    medical_history varchar(255) NOT NULL,
    PRIMARY KEY (pet_id)
);
CREATE TABLE Appointments (
    app_id int NOT NULL,
    pet_id int NOT NULL,
    app_date DATE NOT NULL,
    app_time TIMESTAMP NOT NULL,
    reason varchar(255) NOT NULL,
    veterinary_id int NOT NULL,
    PRIMARY KEY (app_id)
);
CREATE TABLE Veterinarians (
    vet_id int NOT NULL,
    name varchar(255) NOT NULL,
    specialization varchar(255) NOT NULL,
    phone varchar(255) NOT NULL,
    PRIMARY KEY (vet_id)
);
CREATE TABLE Owners (
    owner_id int NOT NULL,
    name varchar(255) NOT NULL,
    adress varchar(255) NOT NULL,
    phone varchar(255) NOT NULL,
    email varchar(255) NOT NULL,
    PRIMARY KEY (owner_id)
);
CREATE TABLE Treatments (
    treatment_id int NOT NULL,
    pet_id int NOT NULL,
    treat_date DATE NOT NULL,
    cost int NOT NULL,
    veterinarian_id int NOT NULL,
    PRIMARY KEY (treatment_id)
);
CREATE TABLE Vet_Programs (
    vet_name varchar(255) NOT NULL,
    vet_id int NOT NULL,
    date_ofwork DATE NOT NULL,
    work_hours int NOT NULL,
    app_id int NOT NULL,
    PRIMARY KEY (vet_name)
);
```

Iar din punct de vedere al primary si foreign keys am urmatoarele constrangeri:

```sql
ALTER TABLE Pets ADD CONSTRAINT Pets_fk0 FOREIGN KEY (owner_id) REFERENCES Owners(owner_id);

ALTER TABLE Appointments ADD CONSTRAINT Appointments_fk0 FOREIGN KEY (pet_id) REFERENCES Pets(pet_id);

ALTER TABLE Appointments ADD CONSTRAINT Appointments_fk1 FOREIGN KEY (veterinary_id) REFERENCES Veterinarians(vet_id);

ALTER TABLE Treatments ADD CONSTRAINT Treatments_fk0 FOREIGN KEY (pet_id) REFERENCES Pets(pet_id);

ALTER TABLE Treatments ADD CONSTRAINT Treatments_fk1 FOREIGN KEY (veterinarian_id) REFERENCES Veterinarians(vet_id);

ALTER TABLE Vet_Programs ADD CONSTRAINT Vet_Programs_fk0 FOREIGN KEY (vet_id) REFERENCES Veterinarians(vet_id);

ALTER TABLE Vet_Programs ADD CONSTRAINT Vet_Programs_fk1 FOREIGN KEY (app_id) REFERENCES Appointments(app_id);
```

Pentru relatiile din tabele am folosit relatii de tip One to One(exemplu intre Vet_programs - vet_id si Veterinary tabl- vet_id) si One to Many(exemplu in Vet_programs app_id pentru tabelul Appointments cate appointment_id).

1.4 Descrierea procedurilor si a functiilor

Am folosit mai multe tipuri de proceduri pentru a putea extrage date utile din baza de date si a testa si vizualiza corectitudinea din punct de vedere functional al bazei de date.

Prima procedura cu complexitate 4 afiseaza toate programarile de dupa o anumita data, ce ramane la latitudinea userului.

```
CREATE PROCEDURE GetAppointmentss

AS
BEGIN
    SELECT
        P.name AS PetName,
        P.species,
        P.breed,
        A.app_date,
        A.app_time,
        A.reason AS AppointmentReason,
        V.name AS VeterinarianName,
        V.specialization AS VeterinarianSpecialization,
        O.name AS OwnerName,
        O.adress AS OwnerAddress
    FROM
        Pets P
    JOIN
        Appointments A ON P.pet_id = A.pet_id
    JOIN
        Veterinarians V ON A.veterinary_id = V.vet_id
    JOIN
        Owners O ON P.owner_id = O.owner_id
    WHERE
        A.app_date >= '2024-01-11';
END
EXEC GetAppointmentss;
```

Messages

PetName	species	breed	app_date	app_time	AppointmentReason	VeterinarianName	VeterinarianSpecialization	OwnerName	OwnerAddress
Charlie	Dog	Dachshund	2024-02-20	0x0000000000007...	Dental Cleaning	Dr. Smith	Canine Specialist	Lucas White	8901 Cedar Blvd
Daisy	Cat	Maine Coon	2024-02-25	0x0000000000007...	Vaccination	Dr. Jones	Feline Specialist	Sara Miller	4567 Spruce Dr
Rocky	Dog	Boxer	2024-03-05	0x0000000000007...	Checkup	Dr. Smith	Canine Specialist	David Wilson	9012 Willow Way
Milo	Cat	Ragdoll	2024-03-10	0x0000000000007...	Vaccination	Dr. Jones	Feline Specialist	Emma Taylor	5678 Aspen Court
Ruby	Dog	German Sheph...	2024-03-15	0x0000000000007DF	Dental Cleaning	Dr. Smith	Canine Specialist	Jack Davis	6789 Redwood St
Oscar	Cat	Sphinx	2024-03-20	0x0000000000007E0	Grooming	Dr. Smith	Canine Specialist	Olivia Lee	7890 Magnolia P...
Ginger	Dog	Cocker Spaniel	2024-03-25	0x0000000000007E1	Checkup	Dr. Jones	Feline Specialist	James Martin	8901 Sycamore ...
Bella	Dog	Labrador	2024-03-20	0x0000000000007E2	Checkup	Dr. Smith	Canine Specialist	Rachel Green	3456 Pine Lane
Bella	Dog	Labrador	2024-03-22	0x0000000000007E3	Vaccination	Dr. Jones	Feline Specialist	Rachel Green	3456 Pine Lane
Bella	Dog	Labrador	2024-03-23	0x0000000000007E4	Dental Cleaning	Dr. Smith	Canine Specialist	Rachel Green	3456 Pine Lane
Molly	Dog	Poodle	2024-03-20	0x0000000000007E5	Checkup	Dr. Smith	Canine Specialist	Ethan Davis	1234 Cedar Road
Molly	Dog	Poodle	2024-03-22	0x0000000000007E6	Vaccination	Dr. Jones	Feline Specialist	Ethan Davis	1234 Cedar Road
Molly	Dog	Poodle	2024-03-23	0x0000000000007E7	Dental Cleaning	Dr. Smith	Canine Specialist	Ethan Davis	1234 Cedar Road
Ruby	Dog	German Sheph...	2024-03-20	0x0000000000007E8	Checkup	Dr. Smith	Canine Specialist	Jack Davis	6789 Redwood St
Ruby	Dog	German Sheph...	2024-03-22	0x0000000000007E9	Vaccination	Dr. Jones	Feline Specialist	Jack Davis	6789 Redwood St
Ruby	Dog	German Sheph...	2024-03-23	0x0000000000007EA	Dental Cleaning	Dr. Smith	Canine Specialist	Jack Davis	6789 Redwood St

Am implementat, de asemenea si o procedura de complexitate 6 care sa filtreze dupa motivul programarii, acesta fiind „regular checkup”:

```
CREATE PROCEDURE GetRegularCheckupAppointments
AS
BEGIN
    -- Definiți instrucțiunea SELECT pentru a obține rezultatele
    SELECT
        P.name AS PetName,
        P.species,
        P.breed,
        P.date_ofbirth,
        A.app_date,
        A.app_time,
        V.name AS VeterinarianName
    FROM
        Pets P
    JOIN
        Appointments A ON P.pet_id = A.pet_id
    JOIN
        Veterinarians V ON A.veterinary_id = V.vet_id
    WHERE
        A.reason = 'Regular Checkup';
END;

EXEC GetRegularCheckupAppointments;


/*Acest raport va arăta numele animalului de companie, specia, rasa, numele și adresa
```

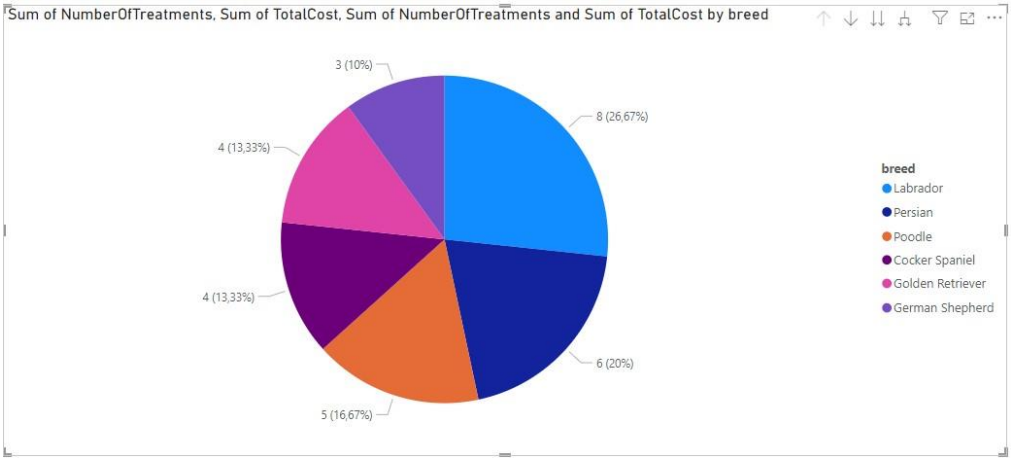
Messages

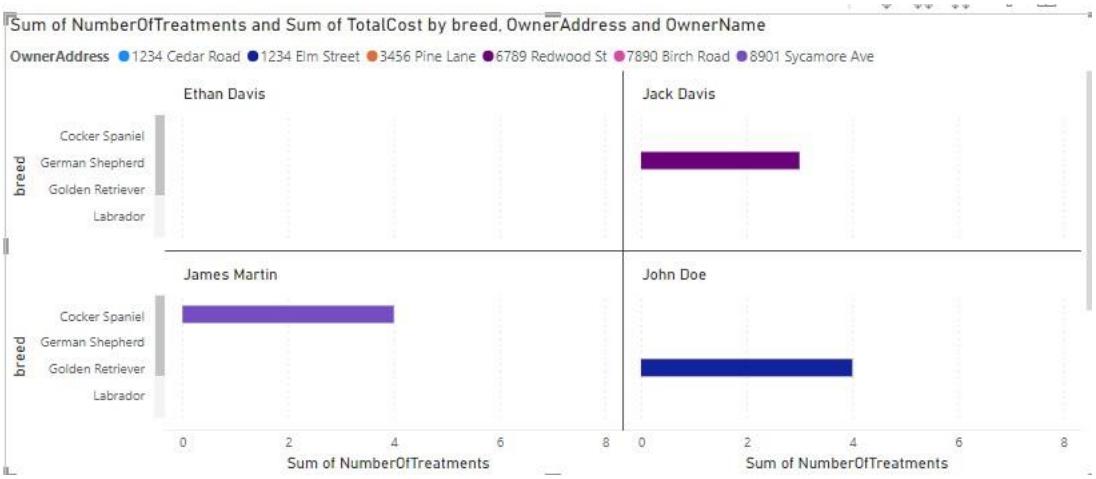
PetName	species	breed	date_ofbirth	app_date	app_time	VeterinarianName
Buddy	Dog	Golden Retriever	2019-05-01	2024-01-10	0x00000000000007D1	Dr. Smith

O alta procedura de complexitate 4 pe care am implementat-o este `GetPetTreatmentSummary`. Aceasta arăta numele animalului de companie, specia, rasa, numele și adresa proprietarului, numărul total de tratamente și costul total
al acestor tratamente. Procedura se concentrează pe animalele de companie care au avut mai mult de două tratamente

```
CREATE PROCEDURE GetPetTreatmentSummary
AS
BEGIN
    SELECT
        P.name AS PetName,
        P.species,
        P.breed,
        O.name AS OwnerName,
        O.adress AS OwnerAddress,
        COUNT(T.treatment_id) AS NumberOfTreatments,
        SUM(T.cost) AS TotalCost
    FROM
        Pets P
    JOIN
        Treatments T ON P.pet_id = T.pet_id
    JOIN
        Owners O ON P.owner_id = O.owner_id
    GROUP BY
        P.pet_id, P.name, P.species, P.breed, O.name, O.adress
    HAVING
        COUNT(T.treatment_id) > 2;
END;
EXEC GetPetTreatmentSummary;
```

Pentru aceasta procedura, m-am conectat la toolul PowerBI de vizualizare a datelor si am creat grafic si un pie chart:



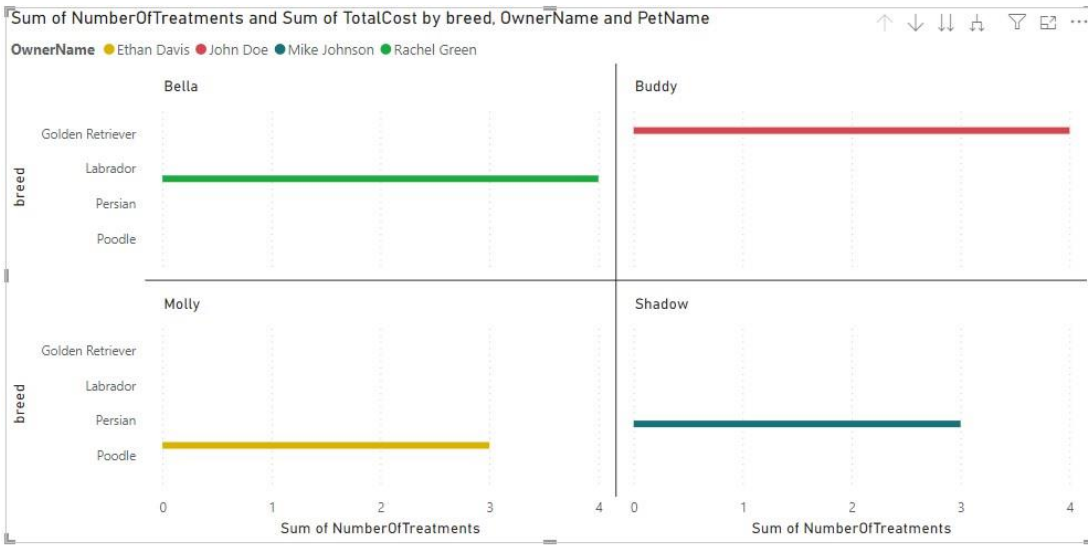


Procedura „GetPetDetails”, cu complexitate 6, include și numele și specializarea veterinarului care a efectuat tratamentele, împreună cu informațiile despre animalul de companie și proprietarul acestuia. Interogarea afișează doar acele animale care au avut mai mult de două tratamente efectuate de același veterinar (cu veterinarian_id = 1) în ultimul an.

```sql
/*Procedura include și numele și specializarea veterinarului care a efectuat tratamentele, împreună cu informațiile despre animalul de
companie și proprietarul acestuia. Interogarea afișează doar acele animale care au avut mai mult de două tratamente efectuate de același
veterinar (cu veterinarian_id = 1) în ultimul an.*/
CREATE PROCEDURE GetPetDetails
AS
BEGIN
    SELECT
        P.name AS PetName,
        P.species,
        P.breed,
        O.name AS OwnerName,
        O.adress AS OwnerAddress,
        V.name AS VeterinarianName,
        V.specialization AS VeterinarianSpecialization,
        COUNT(T.treatment_id) AS NumberOfTreatments,
        SUM(T.cost) AS TotalCost
    FROM
        Pets P
    JOIN
        Treatments T ON P.pet_id = T.pet_id
    JOIN
        Owners O ON P.owner_id = O.owner_id
    JOIN
        Veterinarians V ON T.veterinarian_id = V.vet_id
    WHERE
        T.treat_date > DATEADD(year, -1, SYSDATETIME())
        AND T.veterinarian_id = 1
    GROUP BY
        P.pet_id, P.name, P.species, P.breed, O.name, O.adress, V.name, V.specialization
    HAVING
        COUNT(T.treatment_id) > 2;
END
EXEC GetPetDetails
```

Results | Messages

PetName	species	breed	OwnerName	OwnerAddress	VeterinarianName	VeterinarianSpecialization	NumberOfTreatments	TotalCost
Buddy	Dog	Golden Retriever	John Doe	1234 Elm Street	Dr. Smith	Canine Specialist	4	850
Shadow	Cat	Persian	Mike Johnson	7890 Birch Road	Dr. Smith	Canine Specialist	3	310
Bella	Dog	Labrador	Rachel Green	3456 Pine Lane	Dr. Smith	Canine Specialist	4	440
Molly	Dog	Poodle	Ethan Davis	1234 Cedar Road	Dr. Smith	Canine Specialist	3	450

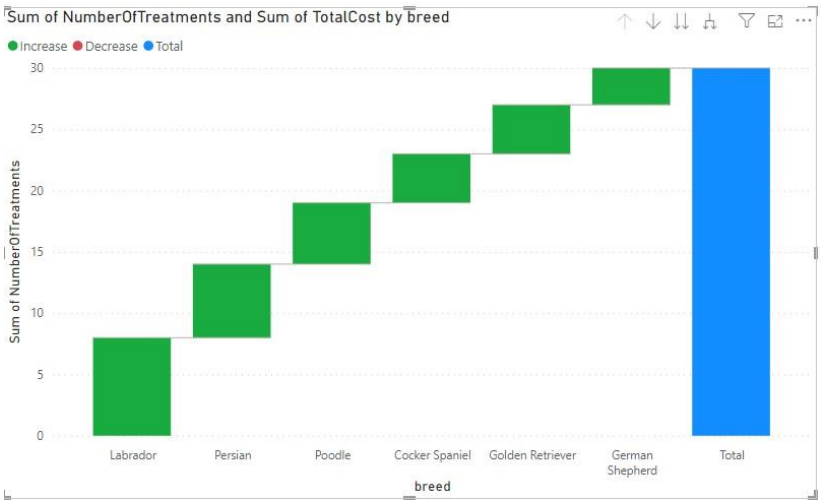


Pentru ultima procedura, de complexitate 7, numita „`GetPetTreatmentAppointmentSumm`” ofer o privire de ansamblu asupra tratamentelor și programărilor veterinare pentru fiecare animal de companie în ultimul an, inclusiv numărul de tratamente, costul total, și ultima dată de tratament, focalizându-se pe animalele care au fost tratate de mai mult de un veterinar.

```sql
CREATE PROCEDURE GetPetTreatmentAppointmentSumm
AS
BEGIN
    SELECT
        P.name AS PetName,
        P.species,
        P.breed,
        O.name AS OwnerName,
        O.adress AS OwnerAddress,
        V.name AS VeterinarianName,
        COUNT(DISTINCT T.treatment_id) AS NumberOfTreatments,
        COUNT(DISTINCT A.app_id) AS NumberOfAppointments,
        MAX(T.treat_date) AS LastTreatmentDate,
        SUM(T.cost) AS TotalCost
    FROM
        Pets P
    JOIN
        Treatments T ON P.pet_id = T.pet_id
    JOIN
        Owners O ON P.owner_id = O.owner_id
    JOIN
        Veterinarians V ON T.veterinarian_id = V.vet_id
    JOIN
        Appointments A ON P.pet_id = A.pet_id
    WHERE
        T.treat_date > DATEADD(year, -1, SYSDATETIME())
    GROUP BY
        P.pet_id, P.name, P.species, P.breed, O.name, O.adress, V.name
    HAVING
        COUNT( T.veterinarian_id) > 1;
END;
EXEC GetPetTreatmentAppointmentSumm;
```

Results | Messages

PetName	species	breed	OwnerName	OwnerAddress	VeterinarianName	NumberOfTreatments	NumberOfAppointments	LastTreatmentDate	TotalCost
Buddy	Dog	Golden Retriever	John Doe	1234 Elm Street	Dr. Smith	4	3	2024-02-15	2550
Whiskers	Cat	Siamese	Jane Smith	5678 Oak Avenue	Dr. Jones	2	1	2024-02-20	450
Shadow	Cat	Persian	Mike Johnson	7890 Birch Road	Dr. Jones	3	1	2024-06-05	310
Shadow	Cat	Persian	Mike Johnson	7890 Birch Road	Dr. Smith	3	1	2024-06-04	310
Bella	Dog	Labrador	Rachel Green	3456 Pine Lane	Dr. Jones	4	4	2024-07-08	2240
Bella	Dog	Labrador	Rachel Green	3456 Pine Lane	Dr. Smith	4	4	2024-07-08	1760
Ruby	Dog	German Shepherd	Jack Davis	6789 Redwood St	Dr. Jones	2	4	2024-07-08	1000
Ruby	Dog	German Shepherd	Jack Davis	6789 Redwood St	Dr. Smith	1	4	2024-04-15	1000
Ginger	Dog	Cocker Spaniel	James Martin	8901 Sycamore Ave	Dr. Jones	3	1	2024-05-05	410
Molly	Dog	Poodle	Ethan Davis	1234 Cedar Road	Dr. Jones	2	3	2024-07-08	900
Molly	Dog	Poodle	Ethan Davis	1234 Cedar Road	Dr. Smith	3	3	2024-04-10	1350



2.0 Aplicatie

Pentru aceasta baza de date, nu am folosit o aplicatie. Pentru partea de backend am utilizat SQL Server unde am creat o baza de date si m-am conectat cu ajutorul dockerului, iar vizualizarea datelor am facut-o conform laboratorului cu PowerBI.

3.0 Concluzii

A fost un proiect interesant, care mi-a cauzat probleme doar la partea de conexiune intre SQL Server si PowerBI, insa m-a ajutat sa inteleg mai bine cum functioneaza.

4.0 Bibliografie

https://ocw.cs.pub.ro/courses/bd2

https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16

https://learn.microsoft.com/en-us/power-bi/