

# Documentație Proiect

Poenaru Iulian

## 1. Descrierea proiectului

Proiectul are rolul de a centraliza și gestiona informațiile esențiale pentru un cabinet veterinar, referitoare atât la stăpânii animalelor, cât și la acestea din urmă. Fiecare stăpân poate fi asociat cu mai multe animale, iar pentru fiecare animal sunt înregistrate una sau mai multe date medicale relevante. Aceste informații sunt esențiale pentru medicul veterinar, furnizându-i datele necesare pentru a-și desfășura activitatea într-un mod eficient și precis. Evident, acest proiect reprezintă o abordare mai restrânsă în comparație cu necesitățile reale ale unui cabinet veterinar, însă acoperă o parte din datele vitale folosite în această activitate.

## 2. Dependențe funcționale identificate și Forma Normală Boyce-Codd

PetOwners:  $\{CNP\} \rightarrow \{first\_name, last\_name, address, phone, email\}$

Pet:  $\{ID\} \rightarrow \{CNP\_owner, name, species, breed, age, gender\}$

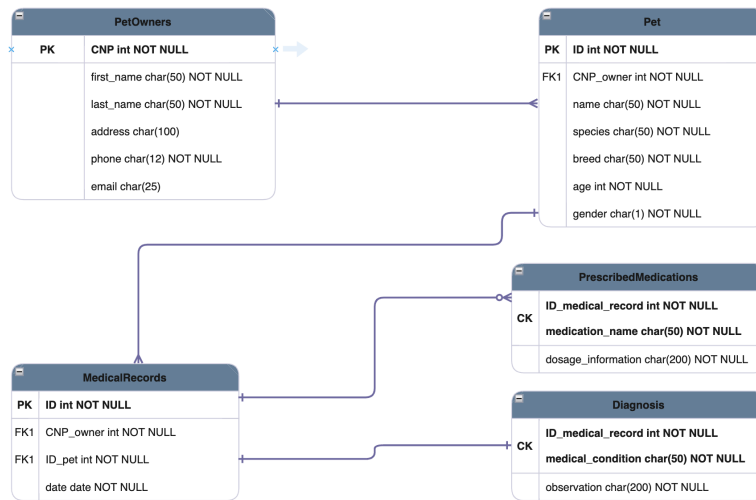
MedicalRecords:  $\{ID\} \rightarrow \{CNP\_owner, ID\_pet, date\}$

PrescribedMedications:  $\{ID\_medical\_record, medication\_name\} \rightarrow \{dosage\_information\}$

Diagnosis:  $\{ID\_medical\_record, medical\_condition\} \rightarrow \{observation\}$

Absența dependențelor funcționale non-triviale, precum și faptul că fiecare atribut al tabelor depinde direct de cheia primară, confirmă conformitatea cu cerințele BCNF. De asemenea, menționarea creării de tabele separate pentru grupurile de obiecte subliniază respectarea principiului de a evita stocarea mai multor tipuri de informații într-un singur rând, consolidând astfel structura bazei de date conform BCNF.

### 3. Diagrama ERD



### 4. & 5. Definirea tabelelor și exemple de operații

Definirea tabelelor poate fi exemplificată prin operația de creare a acestora, cât și prin afișarea structurii. Mai jos sunt prezentate operațiile *create table* și *describe*.

```
create table PetOwner(  
    cnp int not null primary key,  
    first_name char(50) not null,  
    last_name char(50) not null,  
    address char(50),  
    phone char(12) not null,  
    email char(25)  
);
```

name	type	notnull	dfilt_value	pk
Filter...	Filter...	Filter...	Filter...	Filter...
cnp	INT	1	NULL	1
first_name	char(50)	1	NULL	0
last_name	char(50)	1	NULL	0
address	char(50)	0	NULL	0
phone	char(12)	1	NULL	0
email	char(25)	0	NULL	0

```
create table Pet(  
    id int not null primary key,  
    cnp_owner int not null ,  
    name char(50) not null,  
    species char(50) not null,  
    breed char(50) not null,  
    age int not null,  
    gender char(1) not null  
  
    foreign key (cnp_owner) references PetOwner(cnp)  
);
```

name	type	notnull	dfilt_value	pk
Filter...	Filter...	Filter...	Filter...	Filter...
id	INT	1	NULL	1
cnp_owner	INT	1	NULL	0
name	char(50)	1	NULL	0
species	char(50)	1	NULL	0
breed	char(50)	1	NULL	0
age	INT	1	NULL	0
gender	char(1)	1	NULL	0

```
create table MedicalRecords(  
    id int not null primary key,  
    cnp_owner int not null,  
    id_pet int not null,  
  
    foreign key (cnp_owner) references PerOwner(cnp),  
    foreign key (id_pet) references Pet(id)  
);
```

name	type	notnull	dfit_value	pk
id	INT	1	NULL	1
cnp_owner	INT	1	NULL	0
id_pet	INT	1	NULL	0

```
create table PrescribedMedications(  
    id_medical_record int not null,  
    medication_name char(50) not null,  
    dosage_infomation char(200) not null,  
  
    primary key(id_medical_record, medication_name),  
    foreign key (id_medical_record) references MedicalRecord(id)  
);
```

name	type	notnull	dfit_value	pk
id_medical_record	INT	1	NULL	1
medication_name	char(50)	1	NULL	2
dosage_infomation	char(200)	1	NULL	0

```
create table Diagnosis(  
    id_medical_record int not null,  
    medical_condition char(50) not null,  
    observation char(200) not null,  
  
    primary key(id_medical_record, medical_condition),  
    foreign key (id_medical_record) references MedicalRecord(id)  
);
```

name	type	notnull	dfit_value	pk
id_medical_record	INT	1	NULL	1
medical_condition	char(50)	1	NULL	2
observation	char(200)	1	NULL	0

Tabela **PetOwner** reține date referitoare la stăpânii animalelor ce vizitează cabinetul, iar fiecare stăpân poate avea mai multe animale. Datele animalelor de companie sunt reținute în tabela **Pet**, fiecare animal având o înregistrare medicală prezentă în tabela **MedicalRecords**. Fiecarei înregistrări medicale îi este asociat un diagnostic și o prescripție medicală, prezente în tabela **Diagnosis** și **PrescribedMedications**.

Pentru a exemplifica operația de modificare a structurii unei tabele putem adăga coloane, redenumi coloane și șterge coloane. Mai jos vom exemplifica aceste operații prin adăugarea unei noi coloane *color* în tabela **Pet**, o vom redenumi, ca mai apoi să o ștergem.

name	type	notnull	dfit_value	pk
id	INT	1	NULL	1
cnp_owner	INT	1	NULL	0
name	char(50)	1	NULL	0
species	char(50)	1	NULL	0
breed	char(50)	1	NULL	0
age	INT	1	NULL	0
gender	char(1)	1	NULL	0
color	char(20)	0	NULL	0

```
alter table Pet  
add column color char(20);
```

name	type	notnull	dfit_value	pk
aBc Filter...	aBc Filter...	aBc Filter...	aBc Filter...	aBc
id	INT	1	NULL	1
cnp_owner	INT	1	NULL	0
name	char(50)	1	NULL	0
species	char(50)	1	NULL	0
breed	char(50)	1	NULL	0
age	INT	1	NULL	0
gender	char(1)	1	NULL	0
color_of_pet	char(20)	0	NULL	0

```
alter table Pet
rename column color to color_of_pet
```

name	type	notnull	dfit_value	pk
aBc Filter...	aBc Filter...	aBc Filter...	aBc Filter...	aBc
id	INT	1	NULL	1
cnp_owner	INT	1	NULL	0
name	char(50)	1	NULL	0
species	char(50)	1	NULL	0
breed	char(50)	1	NULL	0
age	INT	1	NULL	0
gender	char(1)	1	NULL	0

```
alter table Pet
drop column color_of_pet
```

## 6. Definirea de obiecte ale bazei de date

Pentru a exemplifica crearea de obiecte pentru baza noastră de date vom crea două view-uri.

id	cnp_owner	name	species
aBc Filter...	aBc Filter...	aBc Filter...	aBc Filter...
1	111111111	Buddy	Dog
3	333333333	Max	Dog
5	555555555	Rocky	Dog
7	777777777	Bella	Dog
8	888888888	Oscar	Dog
10	123456789	Teddy	Dog
12	999999999	Sophie	Dog
13	111111111	Chloe	Dog
15	666666666	Ruby	Dog

```
create view DogsFromCabinet as
select * from Pet
where species like "Dog"
```

```
select * FROM DogsFromCabinet
```

cnp	first_name	last_name	address
abc Filter...	abc Filter...	abc Filter...	abc Filter...
222222222	Bob	Johnson	456 Oak St
999999999	Isabella	Taylor	321 Oak St

```

create view OakStreet as
select * from PetOwner
where address like '%Oak%'

select * from OakStreet

```

În prima imagine creăm un view care conține toți câinii din tabela Pet, iar în a doua imagine selectăm toți proprietarii care au menționat în adresă strada Oak.

## 7. Prelucrarea datelor din tabele

Pentru a exemplifica prelucrarea datelor din tabel este nevoie să inserăm date în acestea, lucru pe care îl obținem executând comanda *insert*. Mai jos sunt prezentate exemple de inserare în tabela PetOwner, Pet și PrescribedMedications.

```

INSERT INTO PetOwner (cnp, first_name, last_name, address, phone, email)
VALUES
(1, 'John', 'Doe', '123 Main St', '1234567890', 'john.doe@example.com'),
(2, 'Jane', 'Smith', NULL, '9876543210', NULL),
(3, 'Bob', 'Johnson', '456 Oak St', '5551234567', 'bob.j@example.com'),
(4, 'Alice', 'Williams', NULL, '7778889999', 'alice.w@example.com'),
(5, 'Charlie', 'Brown', '789 Elm St', '4445556666', NULL),
(6, 'Eva', 'Miller', '101 Pine St', '1112223333', 'eva.m@example.com'),
(7, 'David', 'Lee', NULL, '9990001111', NULL),
(8, 'Grace', 'Clark', '202 Cedar St', '3334445555', 'grace.c@example.com'),
(9, 'Sam', 'Taylor', '303 Maple St', '6667778888', NULL),
(10, 'Sophia', 'Moore', '404 Birch St', '2223334444', 'sophia.m@example.com');

```

```
INSERT INTO PrescribedMedications (id_medical_record, medication_name, dosage_information)
VALUES
(1, 'PainKiller', 'Take twice a day with food.'),
(2, 'Antihistamine', 'Take once daily.'),
(3, 'Antibiotic', 'Take as prescribed by the vet.'),
(4, 'Vitamin Supplement', 'Administer with meals.'),
(5, 'Heartworm Preventative', 'Monthly dosage.'),
(6, 'Flea and Tick Medication', 'Apply once a month.'),
(7, 'Anti-inflammatory', 'Take with meals.'),
(8, 'Dewormer', 'Administer as directed by the vet.'),
(9, 'Joint Supplement', 'Administer daily.'),
(10, 'Calming Medication', 'As needed.'),
(11, 'PainKiller', 'Take twice a day with food.'),
(12, 'Vitamin Supplement', 'Administer with meals.'),
(13, 'Antibiotic', 'Take as prescribed by the vet.'),
(14, 'Anti-inflammatory', 'Take with meals.'),
(15, 'Heartworm Preventative', 'Monthly dosage.'),
(16, 'Flea and Tick Medication', 'Apply once a month.'),
(17, 'Dewormer', 'Administer as directed by the vet.'),
(18, 'Joint Supplement', 'Administer daily.'),
(19, 'Calming Medication', 'As needed.'),
(20, 'Antihistamine', 'Take once daily.');
```

```
INSERT INTO Pet (id, cnp_owner, name, species, breed, age, gender)
VALUES
(1, 123456789, 'Buddy', 'Dog', 'Golden Retriever', 3, 'M'),
(2, 234567890, 'Whiskers', 'Cat', 'Siamese', 2, 'F'),
(3, 345678901, 'Max', 'Dog', 'German Shepherd', 4, 'M'),
(4, 456789012, 'Luna', 'Cat', 'Persian', 1, 'F'),
(5, 567890123, 'Rocky', 'Dog', 'Boxer', 2, 'M'),
(6, 678901234, 'Mittens', 'Cat', 'Tabby', 3, 'F'),
(7, 789012345, 'Charlie', 'Dog', 'Beagle', 5, 'M'),
(8, 890123456, 'Oliver', 'Cat', 'Maine Coon', 2, 'M'),
(9, 901234567, 'Bella', 'Dog', 'Labrador Retriever', 1, 'F'),
(10, 123045678, 'Chloe', 'Cat', 'Ragdoll', 4, 'F'),
(11, 123456789, 'Cooper', 'Dog', 'Poodle', 2, 'M'),
(12, 123456789, 'Lucy', 'Dog', 'Labrador Retriever', 1, 'F'),
(13, 345678901, 'Whiskey', 'Dog', 'Dachshund', 3, 'M'),
(14, 345678901, 'Misty', 'Cat', 'Calico', 2, 'F'),
(15, 567890123, 'Baxter', 'Dog', 'Bulldog', 4, 'M'),
(16, 567890123, 'Smokey', 'Cat', 'Russian Blue', 2, 'M'),
(17, 789012345, 'Milo', 'Dog', 'Shih Tzu', 3, 'M'),
(18, 789012345, 'Lily', 'Cat', 'Sphynx', 1, 'F'),
(19, 901234567, 'Zeus', 'Dog', 'Rottweiler', 2, 'M'),
(20, 901234567, 'Lola', 'Cat', 'Persian', 3, 'F');
```

Deoarece fiecare proprietar poate avea mai multe animale inserăm mai multe animale in tabela corespunzătoare, iar pentru ca o înregistrare medicală poate să nu aibă asociat vreo medicamentație, iar altele pot avea mai multe, in tabela PrescribedMecations avem unele intrări cu id\_medical\_record duplicate, iar altele nici nu există.

Pentru a exemplifica comanda *update* vom afișa întreaga tabelă nemodificată ca mai apoi, după execuția comenzii exemplificate să afișăm tabela rezultată. Afișarea tabelului o vom face folosindu-ne de comanda *select \* from table\_name*.

id	cnp_owner	name	species	breed	id	cnp_owner	name	species	breed
1	111111111	Buddy	Dog	Labrador Retriever	1	111111111	Buddy	Dog	Labrador Retriever
2	222222222	Whiskers	Cat	Siamese	2	222222222	Whiskers	Cat	Siamese
3	333333333	Max	Dog	Golden Retriever	3	333333333	Max	Dog	Golden Retriever
4	444444444	Fluffy	Cat	Persian	4	444444444	Fluffy	Cat	Persian
5	555555555	Rocky	Dog	German Shepherd	5	555555555	Rocky	Dog	German Shepherd
6	666666666	Snowball	Cat	Maine Coon	6	666666666	Snowball	Cat	Maine Coon
7	777777777	Bella	Dog	Poodle	7	777777777	Bella	Dog	YorkShire
8	888888888	Oscar	Dog	Bulldog	8	888888888	Oscar	Dog	Bulldog
9	999999999	Luna	Cat	Ragdoll	9	999999999	Luna	Cat	Ragdoll
10	123456789	Teddy	Dog	Dachshund	10	123456789	Teddy	Dog	Dachshund
11	888888888	Milo	Cat	British Shorthair	11	888888888	Milo	Cat	British Shorthair
12	999999999	Sophie	Dog	Beagle	12	999999999	Sophie	Dog	YorkShire
13	111111111	Chloe	Dog	Siberian Husky	13	111111111	Chloe	Dog	Siberian Husky
14	333333333	Leo	Cat	Sphynx	14	333333333	Leo	Cat	Sphynx
15	666666666	Ruby	Dog	Rottweiler	15	666666666	Ruby	Dog	Rottweiler

```

update Pet
set breed = "YorkShire"
where
    breed in ('Poodle', 'Beagle') and
    species is not 'Cat'

```

Pentru fiecare câine care face parte din rasa Poodle sau Beagle îi schimbăm rasa în YorkShire.

id	cnp_owner	id_pet	id	cnp_owner	id_pet	date	id	cnp_owner	id_pet	date
1	111111111	1	1	111111111	1	2024-01-02	1	111111111	1	2024-01-02
2	222222222	2	2	222222222	2	2024-01-02	2	222222222	2	2023-12-15
3	333333333	3	3	333333333	3	2024-01-02	3	333333333	3	2024-01-02
4	444444444	4	4	444444444	4	2024-01-02	4	444444444	4	2023-12-15
5	555555555	5	5	555555555	5	2024-01-02	5	555555555	5	2024-01-02
6	666666666	6	6	666666666	6	2024-01-02	6	666666666	6	2023-12-15
7	777777777	7	7	777777777	7	2024-01-02	7	777777777	7	2024-01-02
8	888888888	8	8	888888888	8	2024-01-02	8	888888888	8	2023-12-15
9	999999999	9	9	999999999	9	2024-01-02	9	999999999	9	2024-01-02
10	123456789	10	10	123456789	10	2024-01-02	10	123456789	10	2023-12-15
11	888888888	11	11	888888888	11	2024-01-02	11	888888888	11	2024-01-02
12	999999999	12	12	999999999	12	2024-01-02	12	999999999	12	2023-12-15
13	111111111	13	13	111111111	13	2024-01-02	13	111111111	13	2024-01-02
14	333333333	14	14	333333333	14	2024-01-02	14	333333333	14	2023-12-15
15	666666666	15	15	666666666	15	2024-01-02	15	666666666	15	2024-01-02

```







alter table MedicalRecords
add column date date

update MedicalRecords
set date = DATE('now')
where date is null

update MedicalRecords
set date = '2023-12-15'
where date is date('now') and id%2==0

```

Am uitat sa adăugăm o dată pentru înregistrările medicale, așa ca adăugăm o coloană *date* și inițializem fiecare coloană cu data curentă, ca mai apoi sa modificăm toate înregistrările care au id-ul un număr par.

id_medical_rec...	medication_name	dosage_infomation	id_medical_rec...	medication_name	dosage_infomation
 Filter...	 Filter...	 Filter...	 Filter...	 Filter...	 Filter...
1	PawPainRelief	Apply on pet paws twice daily for pain relief	1	HeartWormPrevent	Administer once a month to prevent heart
1	HeartWormPrevent	Administer once a month to prevent heartworms	2	FleaTickControl	Apply on pet back every three months for
2	FleaTickControl	Apply on pet back every three months for flea and tick	4	AntibioticX	Give two tablets twice daily for infection t
4	AntibioticX	Give two tablets twice daily for infection treatment	5	JointHealthSupplement	Administer as directed for improving joint
5	JointHealthSupplement	Administer as directed for improving joint health	5	Antinflammatory	Use for reducing inflammation in pets
5	Antinflammatory	Use for reducing inflammation in pets	6	EyeDrops	Apply drops to the affected pet eye every
6	EyeDrops	Apply drops to the affected pet eye every 4 hours	8	VitaminBoost	Give once daily with pet meal for essential
8	VitaminBoost	Give once daily with pet meal for essential vitamins	8	DigestiveCare	Administer if pet shows signs of digestive
10	DigestiveCare	Administer if pet shows signs of digestive issues	10	SkinHygieneCream	Apply on pet affected skin area twice daily
12	SkinHygieneCream	Apply on pet affected skin area twice daily	12	ThyroidRegulator	Give once a day for regulating pet thyroid
13	ThyroidRegulator	Give once a day for regulating pet thyroid function	13	EarCleaner	Use weekly for maintaining pet ear hygiene
13	EarCleaner	Use weekly for maintaining pet ear hygiene			

```
delete from PrescribedMedications
where medication_name like '_a%'
```

Ștergem toate rândurile din tabela PrescribedMedications pentru care *medication\_name* are pe poziția a doua litera *a*.



## 8. Selecția datelor folosind interogari complexe

1. Selectăm toate înregistrările medicale ale animalelor ale căror proprietari dețin doar un singur animal.

```
create view OwnerWithOnePet as
select po1.* from PetOwner po1
join (
    select p.cnp_owner, count(*) as pet_count from Pet p
    group by p.cnp_owner
) pet_counts
on pet_counts.cnp_owner = po1.cnp
where pet_counts.pet_count = 1

select * from MedicalRecords
where cnp_owner in (SELECT cnp from OwnerWithOnePet)
```

2. Selectăm diagnosticul fiecărui câine al cărui proprietar locuiește pe strada Oak.

```
create view MRFromDogsAroundOak as
select * from MedicalRecords
where
    cnp_owner in (select cnp from OakStreet) and
    id_pet in (SELECT id from DogsFromCabinet)

select * from Diagnosis
where id_medical_record
    in (select id from MRFromDogsAroundOak)
```

3. Selectăm medicamentația pisicilor care au în nume cel puțin 2 vocale.

```
create view CatWithLotsOfVowels as
select * from Pet
where
    species like 'Cat' and
    length(name) - length(replace(lower(name), 'a', '')) +
    length(name) - length(replace(lower(name), 'e', '')) +
    length(name) - length(replace(lower(name), 'i', '')) +
    length(name) - length(replace(lower(name), 'o', '')) +
    length(name) - length(replace(lower(name), 'u', '')) > 1

select * from PrescribedMedications
where id_medical_record in (
    select id from MedicalRecords
    where id_pet in (select id from CatWithLotsOfVowels)
)
```

4. Selectăm numele și vârsta animalelor care au fost diagnosticate având o infecție.

```
create view Infections as
select * from Diagnosis
where instr(lower(medical_condition),'infection')>0

select name, age from Pet
where
    id in (
        select id_pet from MedicalRecords
        where id in (
            select id_medical_record from Infections
        )
    )
```

5. Selectăm animalele ale căror înregistrări datează din 15-12-2023 și le grupăm pe vârste, numărându-le.

```
select count(*) as 'Number of pets', age
from Pet p join MedicalRecords
on p.id=id_pet
where date='2023-12-15'
group by age
```

6. Selectăm observațiile scrise în diagnostic pentru pisicile de peste 3 ani ale căror condiție se termină în “itis”.

```
select * from Pet
where
    species = 'Cat' and
    age >= 3 and
    id in (
        select id_pet from MedicalRecords
        join (
            select observation, id_medical_record
            from Diagnosis
            where medical_condition like '%itis'
        ) diagnosis
        on id=diagnosis.id_medical_record
    )
```

## Observații:

Proiectul a fost realizat folosind baza de date SQLite, iar mai jos sunt diferențele întâlnite de mine în realizarea proiectului:

- Afișarea structurii tabelului - se face cu ajutorul comenzii: *pragma table\_info(TableName)*
- Dacă am dori să folosim *generate as identity* pentru câmpul ID al unei tabeli, SQLite nu ar recunoaște această comandă, echivalentul fiind *autoincrement*