

Курсова работа

Тема: База данни за приюти

Изготвена от:

Силвия Николова – 62542

Александра Радева – 62541

Михаела Илиева – 62550

Илиян Йорданов – 62546

Ивана Великова – 62554

Разпределение на задачите:

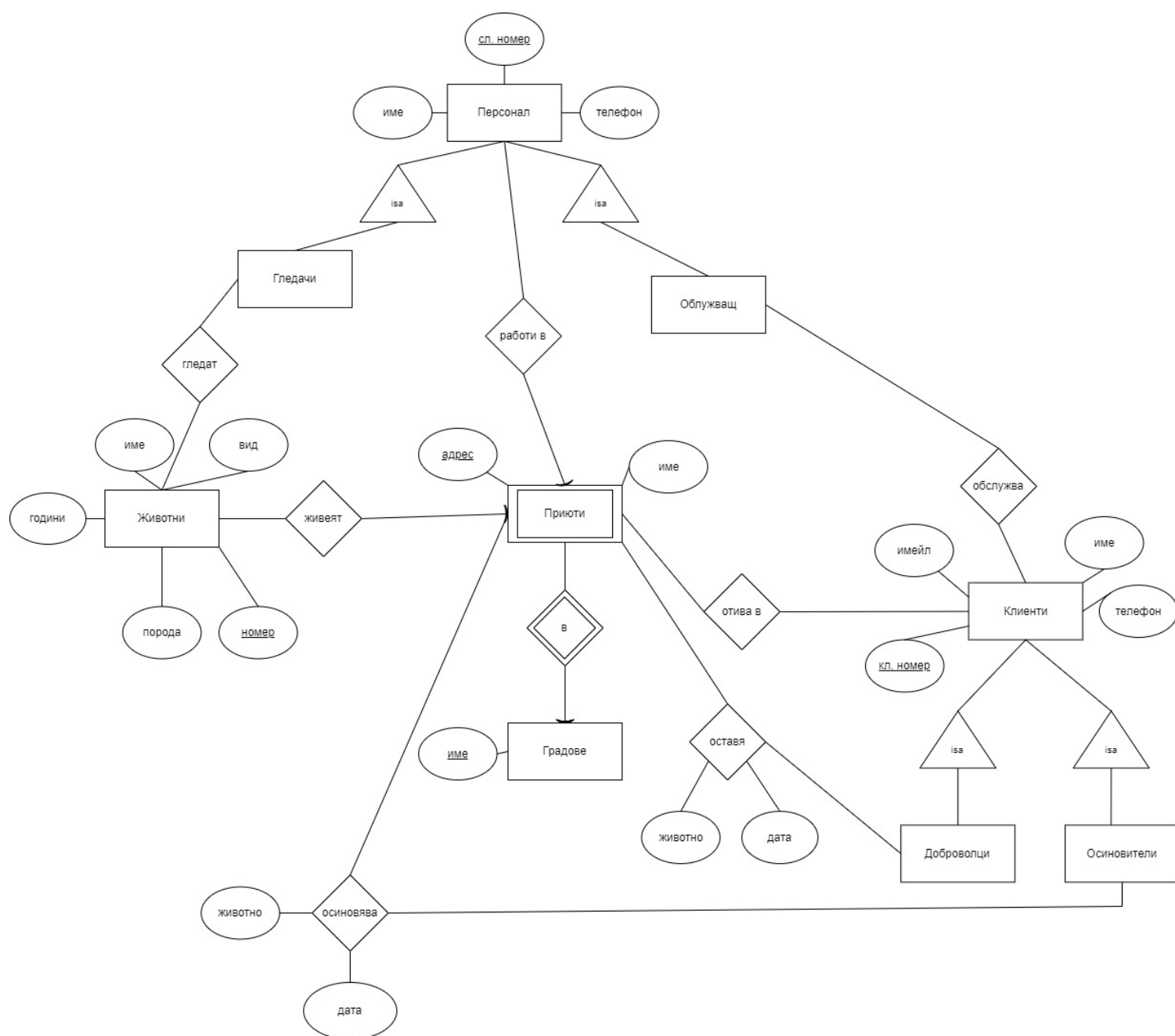
Задача	Фак. №
1. Създаване на схемите и релациите	62542
2. Добавяне на съдържание	62541
3. Примерни прости заявки и заявки върху две и повече релации	62550
4. Примери с подзаявки	62546
5. Примери със съединения	62554
6. Примери с групиране и агрегация	62542
7. Примери с ограничения	62541
8. Примери с изгледи и индекси	62550
9. Примери с тригери	62546
10. Подготовка на презентацията	62554

1. Описание на предметната област и заданието

Нашата идея е да се разработи база от данни, чрез която да се помага на бездомните животни и приютите на територията на България, като се подпомага процесът на осиновяване след това. Проблемът с бездомните животни е проблем на световно ниво. Скорошни проучвания показват, че на територията на България процентът на бездомните животни ежесечно се увеличава, което води до фатален край за голяма част от тях. Нашата идея е да намалим техния брой.

В обхвата на проекта ще бъде проектирана база от данни, от която ще бъдат показани релациите между приютите и хората, които оставят животни за осиновяване или осиновяват.

2. Диаграма на Модел-Същност на нашата система



3. Схема на релациите

- Animals (animalID, name, breed, species, age, shelterAddress, townName);
- Towns (name);

----- OO -----

Clients (clientID, name, email, phoneNumber);

- ClientsVolunteers (clientID, name, email, phoneNumber);
- ClientsAdopters (clientID, name, email, phoneNumber);
- ClientsVolunteersAdopters (clientID, name, email, phoneNumber);

----- E/R -----

- Personnel (privateID, name, phoneNumber, shelterAddress, townName);
- AnimalKeepers (privateID);
- Staff (privateID);
- Shelters (address, name, townName);

- takesCare (animalID, privateID);

~~inhabits (animalID, address, townName) X;~~

- adopts (address, townName, animal, date, clientID);

~~worksIN (privateID, address, townName) X;~~

visits (address, townName, clientID);

- givesForAdoption (address, townName, clientID, animal, date);

service (privateID, clientID);

Обосновка:

Подбраният от нас подход за реализация на наследяването при Клиенти е ООП подходът, защото един клиент от тип Доброволец, който остава за осиновяване, може да е и едновременно осиновител. Затова вместо да използваме друг подход, който ще усложни търсенето и правенето на заявки за намиране на клиенти, които са и от двата типа (Доброволци и Осиновители), сме въвели различни типове клиенти.

Друго, което забелязахме, е, че ако направим 4 различни таблици - Clients, ClientsVolunteers, ClientsAdopters, ClientsVolunteersAdopters (различните типове клиенти), таблицата Clients става безсмислена. По тази причина направихме оптимизация с добавянето на един атрибут Type, който ще показва типа на клиента. Така няма да се налага да правим 4 различни таблици, а само една, като таблиците adopts и givesForAdoption се връзват към Clients.

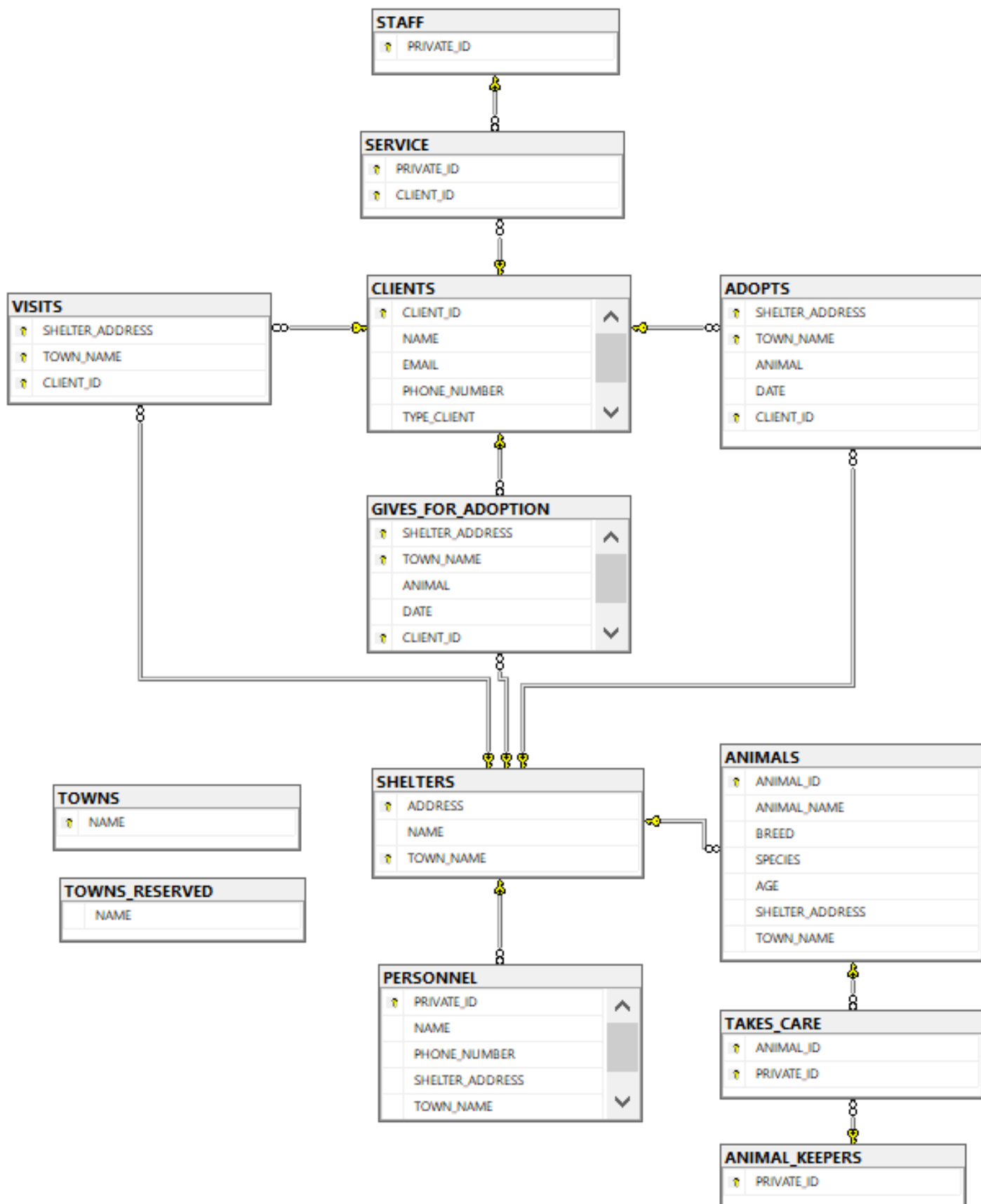
Clients (clientId, name, email, phoneNumber, Type);

Подходът за реализация на наследяването при Personnel е от тип E/R, тъй като един работник, който отговаря за животните, не може да работи в администрацията, както и обратно.

Shelter е слабо множество в нашата база данни, тъй като не е гарантирано, че два приюта в два различни града не са на една и съща улица. Един приют може да е само в един град, докато в един град може да има много приюти, затова връзката е много към едно.

Animals (животните) могат да бъдат само в един приют, докато в един приют може да има много животни, затова сме избрали отново връзка много към едно. Оттук отново направихме една оптимизация, като преценихме, че може да не създаваме таблица inhabits, тъй като се получава връзка много към едно и ключовете на таблицата Shelters могат да станат външни ключове на Animals.

Отново при Personnel можем да избегнем таблицата worksIn, понеже отново имаме връзка много към едно и ключовете на Shelters могат да станат външни на Personnel.



4. Примерни заявки и техният резултат

4.1. Примерни прости заявки

4.1.1. Заявка, която показва името на животното с идентификационен номер 12332:

```
SELECT a.ANIMAL_NAME  
FROM ANIMALS a  
WHERE a.ANIMAL_ID=12332
```

	ANIMAL_NAME
1	Doug

4.1.2. Заявка, която показва имената на приютите в Шумен:

```
SELECT s.NAME  
FROM SHELTERS s  
WHERE s.TOWN_NAME='Shumen'
```

	NAME
1	Animal Care

4.1.3. Заявка, която показва имената на кучетата от порода лабрадор:

```
SELECT a.ANIMAL_NAME  
FROM ANIMALS a  
WHERE a.SPECIES='Dog'  
AND a.BREED='Labrador'
```

	ANIMAL_NAME
1	Doug

4.1.4. Заявка, която показва в кои градове има приюти и ги сортира по азбучен ред:

```
SELECT DISTINCT s.TOWN_NAME  
FROM SHELTERS s  
ORDER BY s.TOWN_NAME
```

	TOWN_NAME
1	Pleven
2	Shumen
3	Sofia
4	Varna
5	Vidin

4.1.5. Заявка, която показва идентификационните номера и имената на персонала, работещи в градове, започващи с В (V):

```
SELECT p.PRIVATE_ID as PersonnelID, p.NAME as PersonnelName  
FROM PERSONNEL p  
WHERE p.TOWN_NAME LIKE 'V%'
```

	PersonnelID	PersonnelName
1	343743	Katerina
2	456789	Lina
3	567890	Plamen
4	879033	Viktoriq

4.2. Заявки върху две и повече релации

- 4.2.1. Заявка, която извежда имената на клиентите, които са дали животинче за осиновяване на приют в град София:

```
SELECT c.NAME
FROM CLIENTS c, GIVES_FOR_ADOPTION g, SHELTERS s
WHERE c.CLIENT_ID=g.CLIENT_ID
AND g.SHELTER_ADDRESS=s.ADDRESS
AND g.TOWN_NAME=s.TOWN_NAME
AND s.TOWN_NAME='Sofia'
```

	NAME
1	Ivan
2	Silvester

- 4.2.2. Заявка, която да изведе идентификационните номера на гледачите на животни с имена "Doug" (животните са с това име):

```
SELECT ak.PRIVATE_ID
FROM ANIMAL_KEEPERS ak, TAKES_CARE tc, ANIMALS a
WHERE ak.PRIVATE_ID=tc.PRIVATE_ID
AND tc.ANIMAL_ID=a.ANIMAL_ID
AND a.ANIMAL_NAME='Doug'
```

	PRIVATE_ID
1	209240

- 4.2.3. Заявка, която показва името, телефонния номер на хората и града им, работещи в приют с име "Animal Care":

```
SELECT p.NAME, p.PHONE_NUMBER
FROM PERSONNEL p, SHELTERS s
WHERE p.SHELTER_ADDRESS=s.ADDRESS
AND p.TOWN_NAME=s.TOWN_NAME
AND s.NAME='Animal Care'
```

	NAME	PHONE_NUMBER
1	Deniz	813125467
2	Gergana	820924099

- 4.2.4. Заявка, която показва името, породата и вида на животните, които са в приюта с име "PetMe" в София:

```
SELECT a.ANIMAL_NAME, a.BREED, a.SPECIES
FROM ANIMALS a, SHELTERS s
WHERE a.SHELTER_ADDRESS=s.ADDRESS
AND a.TOWN_NAME=s.TOWN_NAME
AND s.NAME='PetMe'
```

AND s.TOWN_NAME='Sofia'

	ANIMAL_NAME	BREED	SPECIES
1	Doug	Labrador	Dog

4.2.5. Заявка, която показва имената на папагалите, които са по-възрастни от животното с идентификационен номер 12332:

```
SELECT a.ANIMAL_NAME
FROM ANIMALS a, ANIMALS a1
WHERE a.SPECIES='Parrot'
AND a1.ANIMAL_ID=12332
AND a.AGE>a1.AGE
```

	ANIMAL_NAME
1	Pesho

4.3. Примери с подзаявки

4.3.1. Тази заявка извежда породите на животните от приютите извън град София:

```
select breed
from animals
where TOWN_NAME not in (select TOWN_NAME
                        from SHELTERS
                        where TOWN_NAME='Sofia')
```

	breed
1	Golden
2	Ara
3	Angora
4	Persian

4.3.2. Извежда имената на тези приюти, които са в градове, където адресът на приюта е на улица Сан Стефано:

```
select name
from Shelters
where TOWN_NAME in (select TOWN_NAME
                    from PERSONNEL
                    where SHELTER_ADDRESS='San Stefano')
```

	name
1	4 Paws

4.3.3. Тази заявка извежда имената само на тези градове, които биват посещавани от доброволци:

```
select TOWN_NAME
from visits
where CLIENT_ID in (select CLIENT_ID
                   from CLIENTS
                   where TYPE_CLIENT='volunteer')
```


	TOWN_NAME
1	Shumen
2	Sofia
3	Pleven
4	Varna
5	Vidin

4.3.4. Тази заявка извежда името на града, където се намира адресът, от който е осиновен домашният любимец Зузи:

```
select TOWN_NAME
from shelters
where ADDRESS in (select SHELTER_ADDRESS
                   from adopts
                   where ANIMAL='Zuzi')
```

	TOWN_NAME
1	Shumen

4.3.5. Тази заявка извежда адреса на приюта, в който Зузи е бил оставен за осиновяване:

```
select g.SHELTER_ADDRESS
from GIVES_FOR_ADOPTION g, SHELTERS s
where s.ADDRESS=g.SHELTER_ADDRESS and s.TOWN_NAME=g.TOWN_NAME and s.ADDRESS in
(select SHELTER_ADDRESS
                                     from adopts
                                     where ANIMAL='Zuzi')
```

	SHELTER_ADDRESS
1	Aleksander Todorov 15

4.4. Примери със съединения

4.4.1. Извежда клиентите с местонахождение София:

```
SELECT c.CLIENT_ID, c.NAME
FROM CLIENTS c
JOIN VISITS v ON c.CLIENT_ID = v.CLIENT_ID
WHERE TOWN_NAME = 'Sofia';
```

	CLIENT_ID	NAME
1	213542	Ivan
2	223443	Alex
3	6545344	Valentin
4	837462	Mihaela

4.4.2. Извежда клиентите, които са дали животни за осиновяване в София:

```
SELECT c.CLIENT_ID, c.NAME, gfa.ANIMAL, gfa.DATE
FROM CLIENTS c
JOIN GIVES_FOR_ADOPTION gfa ON c.CLIENT_ID = gfa.CLIENT_ID
WHERE gfa.TOWN_NAME = 'Sofia';
```

	CLIENT_ID	NAME	ANIMAL	DATE
1	213542	Ivan	Pesho	27.03.2015
2	8895426	Silvester	Doug	28.03.2021

4.4.3. Извежда приютите, които НЕ се грижат за хамстери и зайци:

```
SELECT s.NAME, s.ADDRESS, s.TOWN_NAME
FROM SHELTERS s
JOIN ANIMALS a ON s.ADDRESS = a.SHELTER_ADDRESS
WHERE a.SPECIES NOT IN ('Hamster', 'Rabbit');
```

	NAME	ADDRESS	TOWN_NAME
1	PetMe	James Boucher 14	Sofia
2	Animal Hope	Todor Kablev 10	Varna
3	Animal Care	Aleksander Todorov 15	Shumen

4.4.4. Извежда персонала, чийто приют се грижи за котки или кучета:

```
SELECT p.PRIVATE_ID, p.NAME, s.NAME
FROM PERSONNEL p
JOIN SHELTERS s ON p.SHELTER_ADDRESS = s.ADDRESS
JOIN ANIMALS a ON s.ADDRESS = a.SHELTER_ADDRESS
WHERE a.SPECIES = 'Dog' OR a.SPECIES = 'Cat';
```

	PRIVATE_ID	NAME	NAME
1	123456	Denitza	PetMe
2	874367	Kaloyan	PetMe
3	125467	Deniz	Animal Care
4	209240	Gergana	Animal Care

4.4.5. Извежда животните, които са осиновени от клиенти, чиито клиентски номера започват с 3:

```
SELECT a.ANIMAL_ID, a.ANIMAL_NAME, c.CLIENT_ID, c.NAME
FROM ANIMALS a
JOIN SHELTERS s ON a.SHELTER_ADDRESS = s.ADDRESS
JOIN ADOPTS ad ON ad.SHELTER_ADDRESS = s.ADDRESS
JOIN CLIENTS c ON c.CLIENT_ID = ad.CLIENT_ID
WHERE c.CLIENT_ID LIKE '3%';
```

	ANIMAL_ID	ANIMAL_NAME	CLIENT_ID	NAME
1	12332	Doug	343215	Silviya
2	45674	Gosho	343565	Iliyan

4.5. Примери с групиране и агрегация

4.5.1. Извежда броя и имената на всички котки, групирани по име:

```
Select a.ANIMAL_NAME, COUNT(a.SPECIES) as SPECIES_COUNT from ANIMALS a
where a.SPECIES='Cat'
group by a.ANIMAL_NAME
```

	ANIMAL_NAME	SPECIES_COUNT
1	Zuzi	1

4.5.2. Извежда броя на различните породи животни в приютите в София:

```
Select COUNT(a.BREED) as BREED_COUNT
from ANIMALS a join SHELTERS s on a.SHELTER_ADDRESS=s.ADDRESS and
a.TOWN_NAME=s.TOWN_NAME
where s.TOWN_NAME='Sofia'
```

	BREED_COUNT
1	1

4.5.3. Извежда имената и идентификационните номера на целия персонал, групирани по идентификационен номер и име:

```
Select p.PRIVATE_ID, p.NAME from PERSONNEL p
group by p.PRIVATE_ID,p.NAME
```

	PRIVATE_ID	NAME
1	123456	Denitza
2	125467	Deniz
3	209240	Gergana
4	234567	Viktor
5	343743	Katerina
6	345678	Siyana
7	456789	Lina
8	567890	Plamen
9	874367	Kaloyan
10	879033	Viktoriq

4.5.4. Извежда броя на приютите в град Варна:

```
Select count(ADDRESS) as ADDRESS_COUNT from SHELTERS s
where s.TOWN_NAME='Varna'
```

	ADDRESS_COUNT
1	1

4.5.5. Извежда броя на кучетата в приют с адрес София, Джеймс Баучер 14:

```
select s.ADDRESS, s.TOWN_NAME, COUNT(a.SPECIES) as SPECIES_COUNT from ANIMALS a
join SHELTERS s on s.ADDRESS=a.SHELTER_ADDRESS and s.TOWN_NAME=a.TOWN_NAME
where s.TOWN_NAME='Sofia' and s.ADDRESS='James Boucher 14' and a.SPECIES='dog'
group by s.ADDRESS, s.TOWN_NAME
```

	ADDRESS	TOWN_NAME	SPECIES_COUNT
1	James Boucher 14	Sofia	1

4.5.6. Извежда броя на хората от персонала в приюта с адрес Джеймс Баучер 14, София:

```
Select count(p.PRIVATE_ID) as PEOPLE_COUNT from PERSONNEL p
join SHELTERS s on p.SHELTER_ADDRESS=s.ADDRESS and p.TOWN_NAME=s.TOWN_NAME
where s.ADDRESS='James Boucher 14' and s.TOWN_NAME='Sofia'
```

	PEOPLE_COUNT
1	2

4.5.7. Извежда имената на осиновителите, които са осиновили поне 2 животни:

```
select c.NAME, c.CLIENT_ID from CLIENTS c
join ADOPTS ad on c.CLIENT_ID=ad.CLIENT_ID
group by c.NAME, c.CLIENT_ID
having count(ad.ANIMAL)>=2
```

	NAME	CLIENT_ID
1	Alex	223443
2	Mihaela	837462

4.5.8. Извежда адресите и броя на работещия персонал във всеки приют:

```
Select s.TOWN_NAME, s.ADDRESS, count(p.PRIVATE_ID) as PERSONEL_NUMBER from
PERSONNEL p
join SHELTERS s on p.SHELTER_ADDRESS=s.ADDRESS and p.TOWN_NAME=s.TOWN_NAME
group by s.ADDRESS, s.TOWN_NAME
```

	TOWN_NAME	ADDRESS	PERSONEL_NUMBER
1	Pleven	San Stefano	2
2	Shumen	Aleksander Todorov 15	2
3	Sofia	James Boucher 14	2
4	Varna	Todor Kablev 10	2
5	Vidin	Tsar Simeon Veliki	2

4.5.9. Извежда името на приюта и от колко човека е бил посетен:

```
Select s.NAME, COUNT(c.CLIENT_ID) as CLIENTS_NUMBER from SHELTERS s
join VISITS c on s.ADDRESS=c.SHELTER_ADDRESS and s.TOWN_NAME=c.TOWN_NAME
group by s.NAME
```

	NAME	CLIENTS_NUMBER
1	4 Paws	3
2	Animal Care	2
3	Animal Hope	2
4	PetMe	4
5	Save Animals	4

4.5.10. Извежда имената на най-посещаваните приюти:

```
select ss.NAME from SHELTERS ss
join VISITS v on ss.ADDRESS=v.SHELTER_ADDRESS and ss.TOWN_NAME=v.TOWN_NAME
group by name
having COUNT (v.CLIENT_ID)>=ALL(
select COUNT(c.CLIENT_ID)from SHELTERS s
join VISITS c on s.ADDRESS=c.SHELTER_ADDRESS and s.TOWN_NAME=c.TOWN_NAME
group by s.NAME)
```

	NAME
1	PetMe
2	Save Animals

4.5.11. Извежда имената на приютите с най-много животни в тях:

```
select ss.NAME from SHELTERS ss
join ANIMALS a on ss.ADDRESS=a.SHELTER_ADDRESS and ss.TOWN_NAME=a.TOWN_NAME
group by ss.NAME
```

```
having COUNT (a.ANIMAL_ID)>=ALL(
select COUNT(c.ANIMAL_ID)from SHELTERS s
join ANIMALS c on s.ADDRESS=c.SHELTER_ADDRESS and s.TOWN_NAME=c.TOWN_NAME
group by s.NAME)
```

	NAME
1	4 Paws
2	Animal Care
3	Animal Hope
4	PetMe
5	Save Animals

5. Заключение

5.1. Ограничения

- 5.1.1. Първичните и външните ключове за таблиците са съобразени спрямо правилата за тяхното използване и идейните нужди на нашата база от данни.
- 5.1.2. В таблицата CLIENTS с ограничението CHECK се реализира една от описаните по-горе от нас оптимизации, а именно добавянето на много типове клиенти с различен начин на изписване. По този начин се улеснява потребителят при въвеждането на нов тип клиент в базата от данни.

```
CREATE TABLE CLIENTS (
    CLIENT_ID CHAR(10) NOT NULL,
    NAME VARCHAR(20) NOT NULL,
    EMAIL VARCHAR(20),
    PHONE_NUMBER CHAR (15) NOT NULL,
    TYPE_CLIENT VARCHAR (20) NOT NULL CHECK (UPPER(TYPE_CLIENT) IN ('VOLUNTEER',
'ADOPTER', 'VOLUNTEER_ADOPTER', 'ADOPTER_VOLUNTEER', 'ADOPTERVOLUNTEER',
'VOLUNTEERADOPTER'))
);
```

5.2. Изгледи

- 5.2.1. Следният изглед изкарва имената за всеки работник от базата данни Staff:

```
GO
CREATE VIEW v_Names_Of_Staff_Members
AS
SELECT NAME FROM STAFF s JOIN PERSONNEL p
ON s.PRIVATE_ID = p.PRIVATE_ID
GO
```

- 5.2.2. Следният изглед изкарва имената за всеки работник от базата данни Animal_Keeper:

```
GO
CREATE VIEW v_Names_Of_Animal_Keeper_Members
```

```

AS
SELECT NAME FROM ANIMAL_KEEPERS s JOIN PERSONNEL p
ON s.PRIVATE_ID = p.PRIVATE_ID
GO

```

5.2.3. Следният изглед извежда имената и имейлите на клиентите, които са доброволци:

```

GO
CREATE VIEW v_Names_And_Emails_Of_Volunteers
AS
SELECT NAME, EMAIL
FROM CLIENTS
WHERE TYPE_CLIENT='volunteer';
GO

```

5.2.4. Изглед, който показва името на приюта, адреса, града, броя на животните в него:

```

GO
CREATE VIEW v_Number_Of_Animals
AS
SELECT s.NAME, s.ADDRESS, s.TOWN_NAME, COUNT(a.ANIMAL_ID) number_of_animals
FROM Shelters s, Animals a
WHERE s.ADDRESS=a.SHELTER_ADDRESS AND s.TOWN_NAME=a.TOWN_NAME
GROUP BY s.NAME, s.ADDRESS, s.TOWN_NAME
GO

```

5.2.5. Изглед, който показва броя на осиновените животни от всеки клиент:

```

GO
CREATE VIEW v_Clients_Adoptions_Count
AS
SELECT c.CLIENT_ID, c.NAME, count(ad.ANIMAL) AS "Adoptions count"
FROM CLIENTS c
JOIN ADOPTS ad ON ad.CLIENT_ID = c.CLIENT_ID
GROUP BY c.NAME, c.CLIENT_ID;
GO

```

5.2.6. Изглед, който показва идентификационния номер и имената на персонала, както и в кой приют работят:

```

GO
CREATE VIEW v_Shelters_Personnel
AS
SELECT p.PRIVATE_ID as PERSONNEL_ID, p.NAME as PERSONNEL_NAME, s.NAME as
SHELTER_NAME
FROM PERSONNEL p JOIN SHELTERS s
ON p.SHELTER_ADDRESS=s.ADDRESS
AND p.TOWN_NAME=s.TOWN_NAME
GO

```

5.3. Индекси

5.3.1. Индекс, който служи за оптимизация на търсенето в таблицата ANIMALS, тъй като колоната съдържа доста различни стойности и се използва доста често в заявки:

```
CREATE INDEX IDX_ANIMALS_SPECIES ON ANIMALS(SPECIES);
```

5.3.2. Отново индекс, който служи за оптимизация на търсенето в таблицата ANIMALS, тъй като колоната съдържа доста различни стойности и се използва доста често в заявки:

```
CREATE INDEX IDX_ANIMALS_BREED ON ANIMALS(BREED);
```

5.3.3. Индекс, който ще служи за оптимизация на търсенето на SHELTER_ADDRESS, в който работят служителите от таблицата PERSONNEL:

```
CREATE INDEX IDX_PERSONNEL_SHELTER_ADDRESS ON PERSONNEL(SHELTER_ADDRESS);
```

5.3.4. Понеже ключовете за SHELTERS са SHELTER_ADDRESS и TOWN_NAME, които правят всеки приют уникален, то сме създали индекс за тази колона, защото ще се използва доста често в заявки.

```
CREATE INDEX IDX_PERSONNEL_TOWN_NAME ON PERSONNEL(TOWN_NAME);
```

5.4. Тригери

5.4.1. Тригер, който при изтриване на данни от таблицата TOWNS, при създадена таблица TOWNS_RESERVED добавя изтритото в нея, с цел запазване на информацията:

```
CREATE TABLE TOWNS_RESERVED (  
    NAME VARCHAR(20) NOT NULL  
);  
  
GO  
CREATE TRIGGER TRG_TOWNS_RESERVED  
ON TOWNS  
FOR DELETE  
AS  
BEGIN  
    Declare @NAME VARCHAR(20)  
    SELECT @NAME = NAME FROM deleted  
    INSERT INTO TOWNS_RESERVED  
    VALUES (@NAME)  
END  
GO
```

5.4.2. Тригер, който при добавяне на нови кортежи, следи броят на посетителите на даден приют да не надвишава 10 клиенти:

```
GO  
CREATE TRIGGER TRG_CLIENTS_VISITS_COUNT  
ON VISITS  
FOR INSERT  
AS  
BEGIN  
    Declare @addr VARCHAR(30)  
    SELECT @addr = SHELTER_ADDRESS FROM inserted
```

```

Declare @townName VARCHAR(30)
SELECT @townName = TOWN_NAME FROM inserted

Declare @counter int
SELECT @counter = (SELECT COUNT(v.CLIENT_ID) FROM VISITS v WHERE
v.SHELTER_ADDRESS = @addr AND v.TOWN_NAME = @townName)

IF ( @counter > 10 )
    PRINT ('Shelter limit for clients reached!')
END
GO

```

5.4.3. Тригер, който следи при добавяне на клиент в GIVES_FOR_ADOPTION дали клиентското ID е на клиент, който е от тип Volunteer или Volunteer_Adopter:

```

GO
CREATE TRIGGER TRG_GIVES_FOR_ADOPTION_CHECK_ID
ON GIVES_FOR_ADOPTION
FOR INSERT
AS
BEGIN
    Declare @ID CHAR(10)
    SELECT @ID = CLIENT_ID FROM inserted
    IF (@ID NOT IN (SELECT CLIENT_ID FROM CLIENTS WHERE TYPE_CLIENT LIKE
'VOLUNTEER%' OR TYPE_CLIENT LIKE 'volunteer%'))
        PRINT ('Invalid client type ID')
END
GO

```

5.4.4. Тригер, който следи при добавяне на клиент в ADOPTS дали клиентското ID е на клиент, който е от тип Adopter или Adopter_Volunteer:

```

GO
CREATE TRIGGER TRG_ADOPTS_CHECK_ID
ON ADOPTS
FOR INSERT
AS
BEGIN
    Declare @ID CHAR(10)
    SELECT @ID = CLIENT_ID FROM inserted
    IF (@ID NOT IN (SELECT CLIENT_ID FROM CLIENTS WHERE TYPE_CLIENT LIKE
'ADOPTER%' OR TYPE_CLIENT LIKE 'adopter%'))
        PRINT ('Invalid client type ID')
END
GO

```