



Софийски университет „св. Климент Охридски“

Факултет по Математика и Информатика

ДОКУМЕНТАЦИЯ

към проект по „Системи за управление на бази
от данни-практикум“ на тема:

„Ветеринарна клиника“

Изготвил:

Боянна-Ди Милчева Маринова ф.н.72047

Специалност: Информационни системи

Курс: III

Група: 2

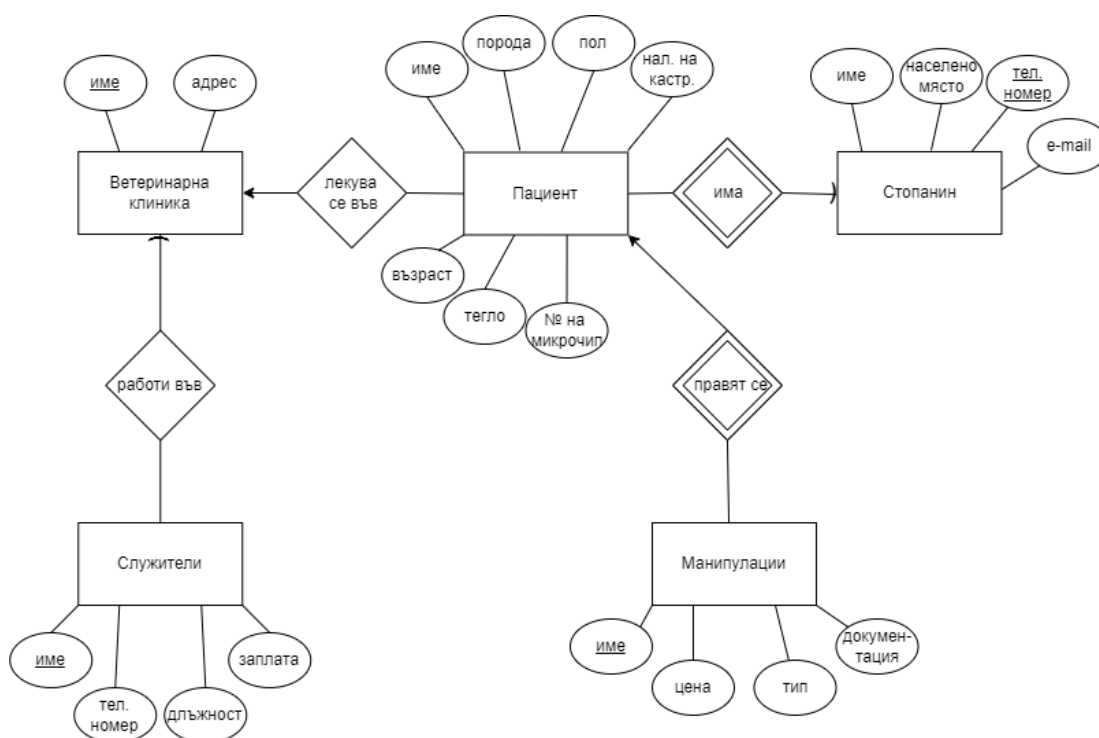
1. Обхват на модела. Дефиниране на задачата

Базата от данни за Ветеринарна клиника, ще съхранява информация за данните на пациентите и техните стопани. Достъп до нея имат главните лекари, лекарите и рецепционистите. В базата данни на клиниката всяко животно има профил, в който пише име, порода, пол, наличие на кастрация, възраст, тегло и номер на микрочипа. В нея има и данните на неговия стопанин като име, населено място, телефонен номер и e-mail. На пациента му се правят манипулации, които се характеризират с име, цена, тип и документация. Във ветеринарната клиника работят около десет човека, от които има главни лекари, лекари, рецепционисти и обслужващ персонал. В базата данни за всеки един от тях пише име, телефонен номер, длъжност и заплата. В една клиника работят много служители. Служителите работят точно в една клиника.

2. Множества от същности и техните атрибути

- Клиника - име и адрес
- Пациент - име, порода, пол, наличие на кастрация, възраст, тегло и номер на микрочипа
- Служители - име, телефонен номер, длъжност и заплата
- Стопанин - име, телефонен номер, населено място и e-mail
- Манипулации – име, цена, тип и документация

3. E/R модел на данни



4. Релационен модел на данни

Първо преобразуваме isa-йерархиите, после множествата от същности, после връзките много-много и накрая връзките много-един. При isa-йерархия може да използваме три подхода: ER, ООП и , и след това да се разглеждат останалите елементи. Тъй като нямам дефинирани isa-йерархии, както и връзките много-много, започвам работа по останалите елементи.

В E/R модела на база данни за ветеринарна клиника имам дефинирани четири пълни множества и едно непълно множество със съответните им допълващи връзки. Първо разглеждам и оформям пълните множества, а след това спрямо техните елементи определям и слабото множество посредство поддържащите връзки.

При преобразуването в релационни схеми полетата, които еднозначно определят множествата се превръщат в колона за първичен ключ (primary key), а останалите атрибути се добавят в последващите колони на таблицата. Релационните схеми на множествата за ветеринарната клиника ще са следните:

- Ветеринарна клиника (име, адрес)
- Служители (име, тел.номер, длъжност, заплата, имеВетеринарнаКлиника)
- Стопанин (тел.номер, име, населено място, e-mail)
- Манипулации (име, цена, тип, документация, имеПациент, тел.номерСтопанин)
- Работи във (имеВетеринарнаКлиника, имеСлужител) X (оптимизираме)
- Правят се (имеМанипулация, имеПациент) X (оптимизираме)

След като оправих пълните множества от същности сега трябва да направим и непълното множество от същности Пациент:

- Пациент (име, тел.номерСтопанин, порода, пол, нал. на кастр., възраст, тегло, № на микрочипа, имеВетеринарнаКлиника)
- Има (имеПациент, тел.номерСтопанин) X (оптимизираме)
- Лекува се във (имеВетеринарнаКлиника, имеПациент) X (оптимизираме)

Окончателно схемата на базата от данни се състои от следните релационни схеми:

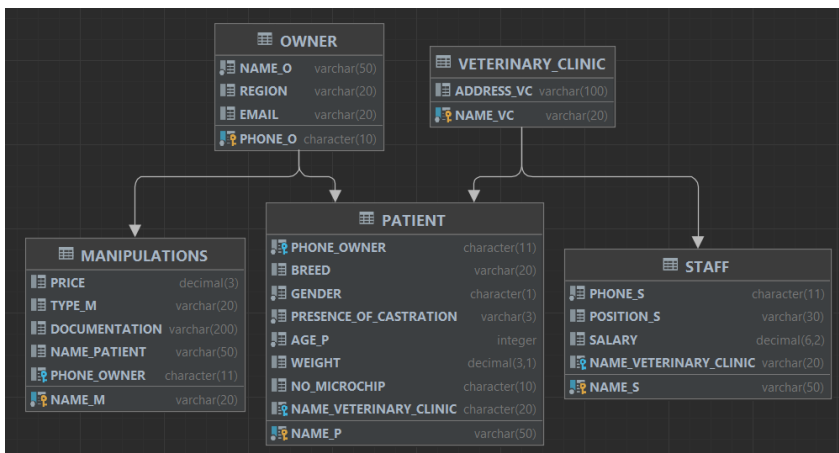
- Ветеринарна клиника (име, адрес)
- Служители (име, тел.номер, длъжност, заплата, имеВетеринарнаКлиника)
- Стопанин (тел.номер, име, населено място, e-mail)
- Манипулации(име, цена, тип, документация, имеПациент, тел.номерСтопанин)
- Пациент (име, тел.номерСтопанин, порода, пол, нал. на кастр., възраст, тегло, № на микрочипа, имеВетеринарнаКлиника)

5. Реляционен модел на данни (реляционни схеми)

Схемата на базата от данни се състои от следните реляционни схеми:

- Ветеринарна клиника (име, адрес)
- Служители (име, тел.номер, длъжност, заплата, имеВетеринарнаКлиника)
- Стопанин (тел.номер, име, населено място, е-mail)
- Манипулации (име, цена, тип, документация, имеПациент, тел.номерСтопанин)
- Пациент (име, тел.номерСтопанин, порода, пол, нал. на кастр., възраст, тегло, № на микрочипа, имеВетеринарнаКлиника)

6. Схема на базата от данни



7. Функции в проекта

Функцията е обект на база данни в SQL. Набор е от SQL изрази, които приемат само входни параметри и произвеждат изход във форма с една стойност или таблична форма.

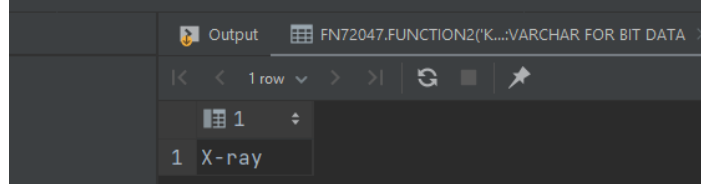
В първата функция, взимам телефонния номер на служителя и проверявам дали съвпада с телефонния номер на стопанина. Ако съвпадат, изкарвам името на пациента.

```
1 CREATE FUNCTION FUNCTION1(PHONE_S VARCHAR(11))
2 RETURNS VARCHAR(50)
3 BEGIN
4     RETURN
5     (SELECT NAME_P
6      FROM PATIENT
7      WHERE PHONE_S = PHONE_OWNER);
8 END;
9
10 VALUES FN72047.FUNCTION1( PHONE_S: '0881234567');
11
```

Output: 1 Kai

Във втората функция, проверяваме дали въведеното име съвпада с името на пациента и ако да връщаме името на манипулацията, която му се прави.

```
12 CREATE FUNCTION FUNCTION2(NAME_P VARCHAR(50))
13 RETURNS VARCHAR(20)
14 BEGIN
15 RETURN
16 (SELECT NAME_M
17 FROM MANIPULATIONS
18 WHERE NAME_P = NAME_PATIENT);
19 END;
20
21 VALUES FN72047.FUNCTION2( NAME_P: 'Kai');
```

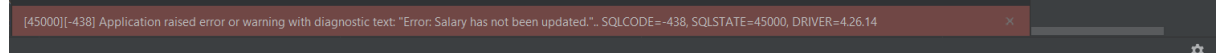


8. Тригери в проекта

Тригерът се извиква всеки път, когато в таблицата възникне събитие като INSERT, DELETE или UPDATE. Тригерът помага за автоматичното изпълнение на действие.

В първия тригер преди да ъпдейтнем заплата на служител проверяваме дали съвпадат, ако да, изписва следното съобщение за грешка:

```
12 UPDATE STAFF
13 SET SALARY = 1500
14 WHERE NAME_S = 'Valya Mitova';
```



Във втория тригер след изтриване на служител, изписва, кои са останалите служители.

9. Изгледи в проекта

Изгледът е виртуална таблица, базирана на резултатния набор от SQL оператор. Той съдържа редове и колони, точно като истинска таблица. Полетата в изглед са полета от една или повече реални таблици в базата данни.

В първи изглед, проверявам дали телефонния номер на служителя - съвпада с телефонния номер на стопанина. Ако да изкарвам името стопанина, на пациента и породата.

```

3 CREATE VIEW INFO
4 AS
5 SELECT S.NAME_S, P.NAME_P, P.BREED
6 FROM STAFF S, PATIENT P
7 WHERE S.PHONE_S = P.PHONE_OWNER;
8
9 SELECT * FROM INFO;

```

	NAME_S	NAME_P	BREED
1	Vladimir Yordanov	Kai	Husky
2	Valya Mitova	Bagira	European shorthair

Във втория изглед, проверявам дали телефонния номер на стопанина съответства в друга таблица с телефонния номер на стопанина. Ако да изкарвам името стопанина, на пациента и породата.

```

11 CREATE VIEW INFO2
12 AS
13 SELECT O.NAME_O, P.NAME_P, P.BREED
14 FROM OWNER O, PATIENT P
15 WHERE O.PHONE_O = P.PHONE_OWNER;
16
17 SELECT * FROM INFO2;

```

	NAME_O	NAME_P	BREED
1	Vladimir Yordanov	Kai	Husky
2	Valya Mitova	Bagira	European shorthair
3	Ivan Petrov	Lady	Sphinx
4	Ivan Petrov	Caesar	German shepherd

10. Процедури в проекта

Процедурата помага да се изпълни определена задача, когато бъде извикана. Може да осъществява достъп или да променя данни в база данни.

В първата процедура взима име на манипулация и връща цената ѝ.

```
MANIPULATION X-ray HAVE PRICE 50
```

Във втората процедура взима име на служител и бонус и връща новата заплата на служителя, ако бонуса е нула, връща грешка.

```
53 BEGIN
54 DECLARE NAME VARCHAR(50) DEFAULT 'Valya Mitova';
55 DECLARE BONUS DECIMAL(6,2) DEFAULT 50;
56 DECLARE NEW_SALARY DECIMAL(6,2);
57 CALL FN72047.BONUS_STAFF( NAME: NAME, V_BONUS: BONUS, NEW_SALARY: NEW_SALARY);
58 CALL DBMS_OUTPUT.PUT_LINE('New salary is = ' || NEW_SALARY);
59 END;
```

New salary is = 1550.00

В третата процедура изписва на дисплея, кои са служителите, каква им е позицията и къде работят.

```
91 CALL FN72047.DISPLAY_STAFF();
92
```

Vladimir Yordanov is a doctor, orthopedist and work in SANTE VITA-VRACA
Ivailo Vasilev is a doctor and work in SANTE VITA-KOZLODUY
Joseph Elenkov is a doctor, cardiologist and work in SANTE VITA-KOZLODUY
Valya Mitova is a receptionist and work in SANTE VITA-VRACA