

Софийски университет "Св. Климент Охридски"

Факултет по математика и информатика

# Тема 4:

## База от данни за ветеринарна клиника

---

*Изготвил: Георги Станчев, трети курс, първа група, факултетен номер: 71959*

*Специалност: Информационни системи*

## Съдържание

1. Обхват на модела. Дефиниране на задачата .....	3
2. Множества от същности и техните атрибути .....	4
3. Домейн на атрибутите.....	5
4. Връзки.....	5
5. Ограничения по единствена стойност, референтна цялостност и друг тип ограничения ..	6
6. Правила и проверки .....	6
7. E/R модел на данни .....	7
8. Релационен модел на данни .....	8
9. Функционални зависимости и Нормализация.....	9
10. Схема на базата от данни.....	9
11. Изгледи.....	10
12. Функции .....	11
13. Процедури .....	12
14. Тригери.....	14
15. Източници .....	17

## 1. Обхват на модела. Дефиниране на задачата

Базата от данни за ветеринарна клиника ще съхранява информация за дейностите и данните в клиниката. Разработената база от данни ще обслужва вътрешна система на клиниката, тоест няма да се обслужва сайт на клиниката. Клиниката разполага с множество клонове, като всеки един от тях е разположен в различен град. В рамките на един и същи град не може да има повече от един клон. Всеки клон се определя еднозначно от своето име, което представлява името на града, в който се намира, като също така той ще има и информация за точния адрес, която ще бъде задължителна за попълване.

Клиниката може да обслужва само определени видове животни – кучета, котки, зайци, морски свинчета, хамстери, таралежи и папагали. За всяко животно се съхраняват неговите вид, име, рождена дата, ваксинации, последно посещение и 12-цифрен код, който се определя от неговия микрочип за идентификация, като от всички тези характеристики само полето за ваксинации може и да не се попълва, но за другите полета е задължително да се даде информация. Клиниката е посещавана от клиенти, за които се пази също определено количество информация. Съхраняват се двете имена на клиента, неговите електронна поща, мобилен номер и адрес, като единствено за електронната поща може да не се дава информация. Животните се определят еднозначно от 12-цифрения си код, а техните собственици, именно клиентите на клиниката, се определят еднозначно по мобилен номер. Клиниката може да обслужва и животни, които си нямат свой собственик, но това е изцяло на тяхна отговорност.

Клиниката предлага различен набор от услуги, които не биха били възможни за изпълнение без служителите, които работят в нея. Те се определят еднозначно по служебен номер на служител. За служителите, освен информация за служебния номер, са пазят и техните две имена, мобилен номер, ветеринарно образование, допълнителни квалификации и адрес, като за допълнителни квалификации може и да няма информация, но за другите характеристики е задължително. Що се отнася до услугите, ветеринарната клиника ще предоставя услуги в следните области: първоначален прием и консултация, терапевтични грижи, ваксинация, хирургични услуги и операции, стоматологични услуги, рентгенови изследвания и лабораторни изследвания. За всяка област си има определени служители, като само услугите, които са в първоначален прием и консултация биха могли да се изпълняват от всички. Една услуга принадлежи на точно една област, но една област може да съдържа множество услуги.

За една услуга се пази име на процедурата, област, в която се намира, подробно описание на дейностите, времетраене в минути и цена в левове, като услугите се определят еднозначно от името на процедурата. От гореспоменатите характеристики всички са задължителни за попълване освен подробното описание на дейността. Важно е и да се отбележи, че в цената не се включени непредвидени разходи от консумативи. Те се заплащат допълнително на касата на клона. Под консумативи се подразбират лекарства, кремове и ваксини. За консуматив се съхранява информация за неговите име, срок на годност, вид на субстанцията, производствен номер и налично количество. Видът на субстанцията може и да не се попълва, но другите характеристики са задължителни, като всеки консуматив се определя еднозначно от своя производствен номер.

## 2. Множества от същности и техните атрибути

- Клонове – име, адрес
- Животни – вид, име, рождена дата, ваксинации, последно посещение, 12-цифрен код
- Клиенти – първо и фамилно име, електронна поща, мобилен номер, адрес
- Служители – първо и фамилно име, служебен номер, мобилен номер, ветеринарно образование, допълнителни квалификации, адрес
- Услуги – име на процедурата, област, към която принадлежи услугата, подробно описание на дейностите, времетраене, цена
- Консумативи – име, срок на годност, вид на субстанцията, производствен номер, налично количество

### 3. Домейн на атрибутите

- Клонове – име: низ, адрес: низ
- Животни – вид: низ, име: низ, рождена дата: дата, ваксинации: низ или NULL, последно посещение: дата, 12-цифрен код: низ
- Клиенти – две имена: низ, електронна поща: низ или NULL, мобилен номер: низ, адрес: низ
- Служители – две имена: низ, служебен номер: низ, мобилен номер: низ, ветеринарно образование: низ, допълнителни квалификации: низ или NULL, адрес: низ
- Услуги – име на процедурата: низ, област, към която принадлежи услугата: низ, подробно описание на дейностите: низ или NULL, времетраене: цяло положително число, цена: реално положително число
- Консумативи – име: низ, срок на годност: дата, вид на субстанцията: низ или NULL, производствен номер: низ, налично количество: реално положително число

### 4. Връзки

- В един Клон работят много Служители. Един Служител работи точно в един Клон.
- Едно Животно може да бъде обслужвано в много Клонове. В един Клон може да се обслужват много Животни.
- Един Клиент може да бъде собственик на много Животни. Едно Животно си има точно един собственик (Клиент).
- Един Служител може да е квалифициран да изпълнява много Услуги. Една Услуга може да бъде изпълнявана от много Служители.
- Един Консуматив може да се използва от точно един Клон. Един Клон може да използва много Консумативи.

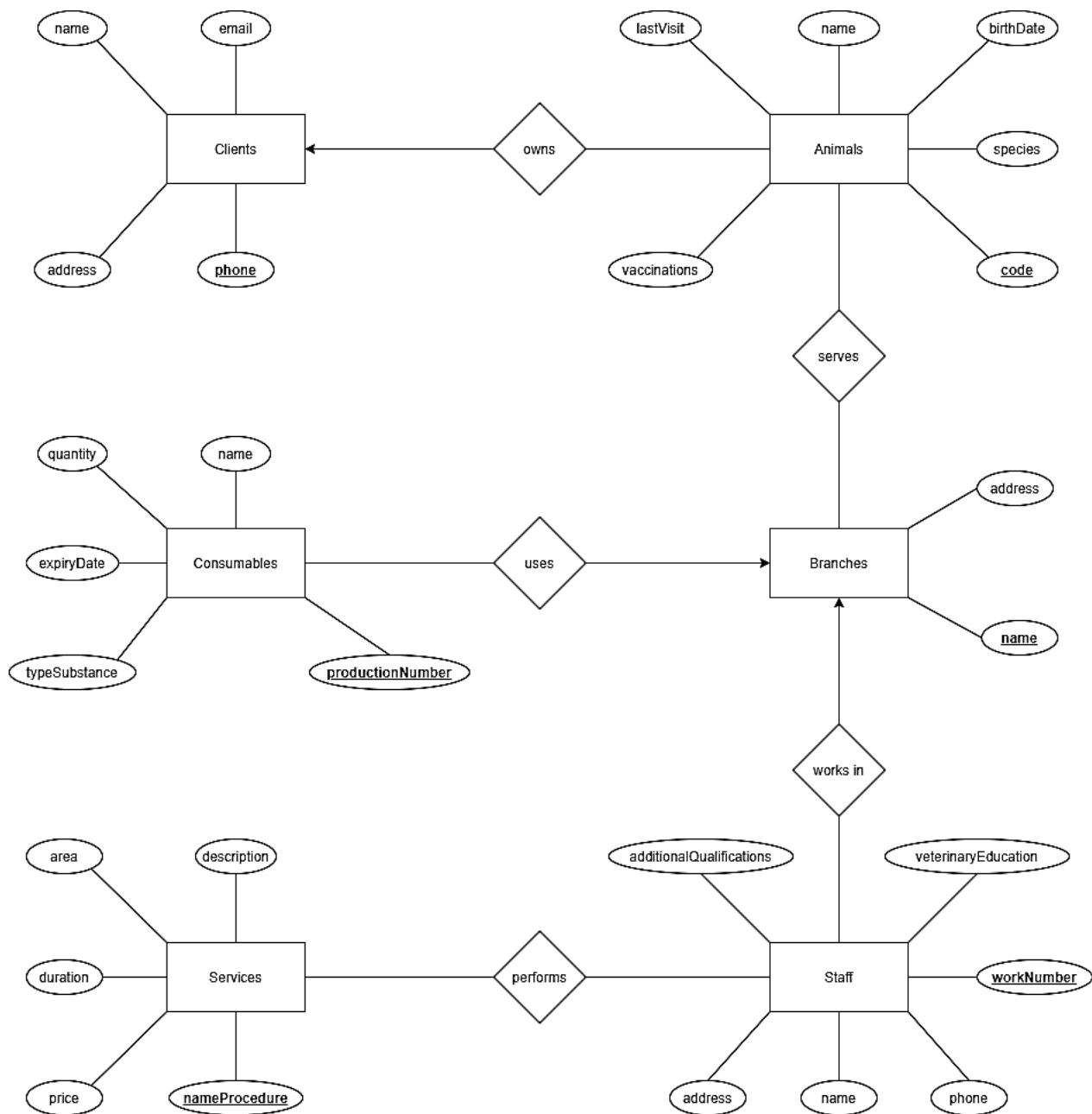
## 5. Ограничения по единствена стойност, референтна цялостност и друг тип ограничения

- Клонове – име: еднозначно определя клона
- Животни – 12-цифрен код: еднозначно определя животното
- Клиенти – мобилен номер: еднозначно определя клиента
- Служители – служебен номер: еднозначно определя служителя
- Услуги – име на процедурата: еднозначно определя услугата
- Консумативи – производствен номер: еднозначно определя консуматива

## 6. Правила и проверки

- Проверка за валиден електронен адрес
- Проверка на всички полета, където трябва да има положително число
- Проверка за валиден вид на животно – видът трябва да бъде един от следните:
  - кучета, котки, зайци, морски свинчета, хамстери, таралежи или папагали
- Проверка за валидна област на услуга – областта трябва да бъде една от следните:
  - първоначален прием и консултация, терапевтични грижи, ваксинация, хирургични услуги и операции, стоматологични услуги, рентгенови изследвания или лабораторни изследвания

## 7. E/R модел на данни



## 8. Релационен модел на данни

Първо преобразуваме множествата от същности, а след това връзките много към много и накрая връзките много към един. Връзките много към един оптимизираме.

Branches (name, address)

Staff (name, address, phone, workNumber, veterinaryEducation, additionalQualifications)

Services (nameProcedure, price, duration, area, description)

Consumables (name, productionNumber, quantity, typeSubstance, expiryDate)

Animals (name, birthDate, species, vaccinations, code, lastVisit)

Clients (name, address, phone, email)

Performs (serviceNameProcedure, staffWorkNumber)

Serves (animalCode, branchName)

WorksIn (staffWorkNumber, branchName) X (Оптимизираме)

Owns (animalCode, clientPhone) X (Оптимизираме)

Uses (branchName, consumableProductionNumber) X (Оптимизираме)

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

Branches (name, address)

Serves (animalCode, branchName)

Services (nameProcedure, price, duration, area, description)

Staff (name, address, phone, workNumber, veterinaryEducation, additionalQualifications, branchName)

Performs (serviceNameProcedure, staffWorkNumber)

Consumables (name, productionNumber, quantity, typeSubstance, expiryDate, branchName)

Animals (name, birthDate, species, vaccinations, code, lastVisit, clientPhone)

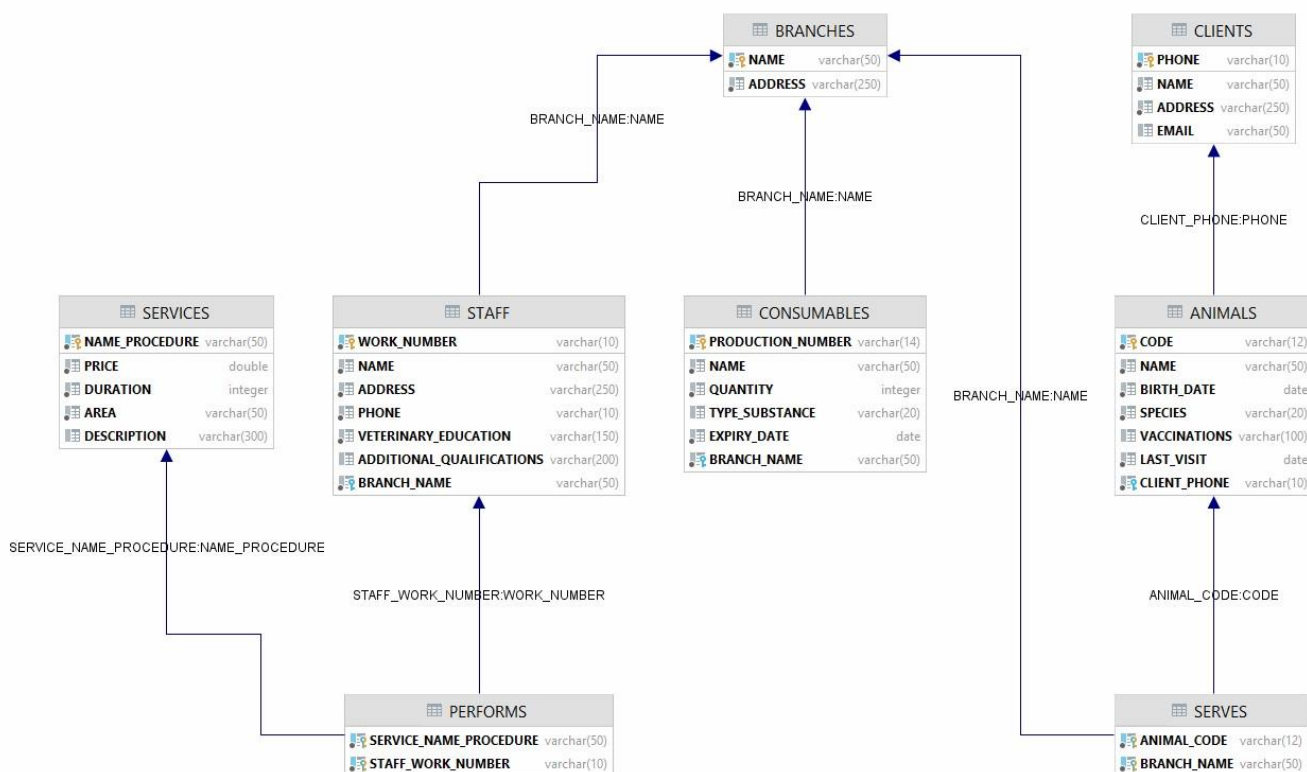
Clients (name, address, phone, email)



## 9. Функционални зависимости и Нормализация

- Ф3-1: name → address (Branches)
- Ф3-2: nameProcedure → price, duration, area, description (Services)
- Ф3-3: workNumber → name, address, phone, veterinaryEducation, additionalQualifications, branchName (Staff)
- Ф3-4: productionNumber → name, quantity, typeSubstance, expiryDate, branchName (Consumables)
- Ф3-5: code → name, birthDate, species, vaccinations, lastVisit, clientPhone (Animals)
- Ф3-6: phone → name, address, email (Clients)

## 10. Схема на базата от данни



## 11. Изгледи

**Изглед 1:** Представяне на информация за продуктите, които скоро биха могли да свършат, за да може администрацията да реагира навреме и да се поръча каквото е необходимо

```
CREATE VIEW DEPLETION_CONSUMABLES(CONSUMABLE, BRANCH, QUANTITY)
AS SELECT NAME, BRANCH_NAME, QUANTITY
FROM CONSUMABLES
WHERE QUANTITY < 15;
```

**Изглед 2:** Представяне на информация за продуктите, чийто срок на годност е изтекъл, за да може персоналът да ги отдели за изхвърляне

```
CREATE VIEW EXPIRED_CONSUMABLES
AS SELECT *
FROM CONSUMABLES
WHERE EXPIRY_DATE < (SELECT CURRENT_DATE FROM SYSIBM.SYSDUMMY1)
WITH CHECK OPTION;
```

**Изглед 3:** Представяне на информация за клиентите, които имат повече от едно животно

```
CREATE VIEW CLIENTS_WITH_MORE_THAN_ONE_ANIMAL (CLIENT, PHONE, NUMBER_OF_ANIMALS)
AS SELECT C.NAME, CLIENT_PHONE, COUNT(*)
FROM ANIMALS A, CLIENTS C
WHERE A.CLIENT_PHONE = C.PHONE
GROUP BY CLIENT_PHONE, C.NAME
HAVING COUNT(*) >= 2;
```

## 12. Функции

**Функция 1:** Приема като параметър код на животно и връща като резултат на колко години, месеци и дни е то.

```
CREATE FUNCTION GETAGEANIMAL(P_CODE VARCHAR(12))
RETURNS VARCHAR(64)
BEGIN
    DECLARE YEARS INT;
    DECLARE MONTHS INT;
    DECLARE DAYS INT;

    SET YEARS = YEAR(CURRENT_DATE - (SELECT BIRTH_DATE FROM ANIMALS WHERE CODE = P_CODE));
    SET MONTHS = MONTH(CURRENT_DATE - (SELECT BIRTH_DATE FROM ANIMALS WHERE CODE = P_CODE));
    SET DAYS = DAY(CURRENT_DATE - (SELECT BIRTH_DATE FROM ANIMALS WHERE CODE = P_CODE));

    RETURN 'Години: ' || YEARS || ', Месеци: ' || MONTHS || ', Дни: ' || DAYS;
END;
```

**Функция 2:** Приема като параметър телефонен номер на клиент и връща колко на брой животни притежава.

```
CREATE FUNCTION ANIMALSPERCLIENT(P_CLIENT_PHONE VARCHAR(10))
RETURNS INT
RETURN
    SELECT COUNT
    FROM (SELECT CLIENT_PHONE, COUNT(*) AS COUNT
        FROM ANIMALS
        GROUP BY CLIENT_PHONE)
    WHERE CLIENT_PHONE = P_CLIENT_PHONE;
```

**Функция 3:** Приема като параметър номер на животно и връща като резултат дали е ваксинирано против бяс.

```
CREATE FUNCTION ISVACCINATEDAGAINSTRABIES(P_CODE VARCHAR(12))
RETURNS VARCHAR(4)
BEGIN
    IF (SELECT VACCINATIONS FROM ANIMALS WHERE CODE = P_CODE
        AND VACCINATIONS LIKE '%Ваксинация против бяс%') IS NULL
    THEN RETURN 'Не';
    ELSE
    RETURN 'Да';
    END IF;
END;
```

## 13. Процедури

### Процедура 1:

Като параметри приема вид на животно и име на ветеринарната клиника.

Като резултат връща брой животни от подадения вид в оказаната ветеринарна клиника.

Ако видът или клиниката не съществуват, то чрез прихващане на изключение броя приема стойност 0.

```
CREATE OR REPLACE PROCEDURE FINDNUMBEROFSPECIES(IN P_SPECIES ANCHOR ANIMALS.SPECIES, IN P_BRANCH_NAME ANCHOR BRANCHES.NAME, OUT V_COUNT INT)
LANGUAGE SQL
BEGIN
    DECLARE COUNTCURSOR CURSOR FOR SELECT COUNT(*) FROM ANIMALS AN, SERVES SRV WHERE AN.CODE = SRV.ANIMAL_CODE
    AND AN.SPECIES = P_SPECIES AND SRV.BRANCH_NAME = P_BRANCH_NAME;

    DECLARE EXIT HANDLER FOR NOT FOUND
    BEGIN
        SET V_COUNT = 0;
    END;

    OPEN COUNTCURSOR;
    FETCH COUNTCURSOR INTO V_COUNT;
    CLOSE COUNTCURSOR;
END@
```

## Процедура 2:

Като параметър приема име на ветеринарната клиника.

Като резултат връща резултатно множество - информация за служителите в клиниката.

Ако клиниката не съществува, то чрез прихващане на изключение се изкарва грешка и съобщение, че няма такава клиника.

```
CREATE OR REPLACE PROCEDURE INFORMATIONEMPLOYEES(IN P_BRANCH_NAME ANCHOR BRANCHES.NAME)
RESULT SETS 1
LANGUAGE SQL
BEGIN

    DECLARE BRANCH_NAME_EXISTS INT;
    DECLARE OVERFLOW CONDITION FOR SQLSTATE '80000';
    DECLARE INFORMATIONCURSOR CURSOR WITH RETURN FOR SELECT * FROM STAFF WHERE BRANCH_NAME = P_BRANCH_NAME;

    DECLARE EXIT HANDLER FOR OVERFLOW
    BEGIN
        RESIGNAL SQLSTATE '80000' SET MESSAGE_TEXT='There is no branch with such a name!';
    END;

    SELECT COUNT(*) INTO BRANCH_NAME_EXISTS FROM BRANCHES WHERE NAME = P_BRANCH_NAME;

    IF BRANCH_NAME_EXISTS = 0 THEN
        SIGNAL OVERFLOW;
    END IF;

    OPEN INFORMATIONCURSOR;

END@
```

## Процедура 3:

Като параметър приема брой месеци.

Като резултат връща резултатно множество - цялата информация в предварително създадена таблица, в която се съхраняват данни за кодът на животното и телефонният номер на собственика му. В нея се вкарват записи за тези животни, които не са посещавали клиника повече от определения брой месеци, подаден като параметър на процедурата.

```
CREATE TABLE NECESSARYVISITS
(
    ANIMAL_CODE VARCHAR(12) NOT NULL,
    CLIENT_PHONE VARCHAR(10) NOT NULL
)@

CREATE OR REPLACE PROCEDURE CHECKFORNECESSARYVISITS(IN P_NUMBER_OF_MONTHS INT)
RESULT SETS 1
LANGUAGE SQL
BEGIN

    DECLARE SQLCODE INT;
    DECLARE RECORDINTABLE ANCHOR ROW ANIMALS;
    DECLARE ANIMALSCURSOR CURSOR FOR SELECT * FROM ANIMALS WHERE MONTH(CURRENT_DATE - ANIMALS.LAST_VISIT) >= P_NUMBER_OF_MONTHS;
    DECLARE OVERVIEW CURSOR WITH RETURN FOR SELECT * FROM NECESSARYVISITS;

    OPEN ANIMALSCURSOR;
    FETCH ANIMALSCURSOR INTO RECORDINTABLE;

    WHILE SQLCODE = 0 DO
        INSERT INTO NECESSARYVISITS(ANIMAL_CODE, CLIENT_PHONE) VALUES (RECORDINTABLE.CODE, RECORDINTABLE.CLIENT_PHONE);
        FETCH ANIMALSCURSOR INTO RECORDINTABLE;
    END WHILE;

    OPEN OVERVIEW;

END@
```

## 14. Тригери

**Тригер 1:** Преди вкарване на нов запис в консумативите се проверява дали има подадена информация за субстанцията. Ако няма, то тя ще приема стойност "НЕИЗВЕСТНО".

```
CREATE TRIGGER UNKNOWNSUBSTANCE
  BEFORE INSERT ON CONSUMABLES
  REFERENCING NEW AS N
  FOR EACH ROW
  WHEN (N.TYPE_SUBSTANCE = '')
  SET N.TYPE_SUBSTANCE = 'НЕИЗВЕСТНО';
```

**Тригер 2:** След промяна на последното посещение на животно се вкарва нов запис в предварително създадена таблица, в която се съхраняват данни за поредността, име, код и последно посещение на животното, телефонния номер на собственика и времето, в което е извършена промяната.

```
CREATE TABLE ANIMALLOGS
(
  ID INT GENERATED ALWAYS AS IDENTITY (START WITH 1 INCREMENT BY 1) NOT NULL PRIMARY KEY,
  NAME VARCHAR(100),
  CODE VARCHAR(12),
  LAST_VISIT DATE,
  CLIENT_PHONE VARCHAR(10),
  RECORD_TIME TIME
);
```

```
CREATE TRIGGER CREATELOGS
  AFTER UPDATE OF LAST_VISIT ON ANIMALS
  REFERENCING OLD AS O NEW AS N
  FOR EACH ROW
  INSERT INTO ANIMALLOGS(NAME, CODE, LAST_VISIT, CLIENT_PHONE, RECORD_TIME)
  VALUES (N.NAME, N.CODE, N.LAST_VISIT, N.CLIENT_PHONE, CURRENT_TIME);
```

**Тригер 3:** След промяна на ваксинациите на животно се проверява дали има разлика при ваксините. Ако да, то последното му посещение ще приеме за стойност датата, в която се прави промяната.

```
CREATE TRIGGER UPDATELASTVISIT
  AFTER UPDATE OF VACCINATIONS ON ANIMALS
  REFERENCING OLD AS O NEW AS N
  FOR EACH ROW
  WHEN (O.VACCINATIONS <> N.VACCINATIONS)
  UPDATE ANIMALS
  SET LAST_VISIT = CURRENT_DATE
  WHERE CODE = O.CODE;
```

**Тригер 4:** При вкарването на нов служител, изкарва информация за целия персонал във ветеринарната клиника, където е назначен.

```
CREATE TRIGGER SHOWINFOABOUTSTAFF
  AFTER INSERT ON STAFF
  REFERENCING NEW AS N
  FOR EACH ROW
  BEGIN
    DECLARE NUMBEROFEMPLOYEES INT DEFAULT 1;
    DECLARE COUNTER INT DEFAULT 0;
    DECLARE RECORD ANCHOR ROW STAFF;
    DECLARE loc RESULT_SET_LOCATOR VARYING;

    CALL FN71959.INFORMATIONEMPLOYEES( P_BRANCH_NAME: N.BRANCH_NAME);
    ASSOCIATE RESULT SET LOCATOR (loc) WITH PROCEDURE FN71959.INFORMATIONEMPLOYEES;
    ALLOCATE CURSORSTAFF CURSOR FOR RESULT SET loc;

    SELECT COUNT(*) INTO NUMBEROFEMPLOYEES FROM STAFF WHERE BRANCH_NAME = N.BRANCH_NAME;

    WHILE COUNTER < NUMBEROFEMPLOYEES DO
      FETCH FROM CURSORSTAFF INTO RECORD;
      CALL DBMS_OUTPUT.PUT_LINE(RECORD.NAME || ' | ' || RECORD.WORK_NUMBER
                                || ' | ' || RECORD.BRANCH_NAME
                                || ' | ' || RECORD.VETERINARY_EDUCATION);
      SET COUNTER = COUNTER + 1;
    END WHILE;

    CLOSE CURSORSTAFF;
  END;
```

**Тригер 5:** Тригер с процедура. При вкарване на ново животно, ще се проверява колко на брой животни от неговия вид вече са регистрирани във всички клонове на ветеринарната клиника. Ако бройката се дели целочислено на 5, то ще се изписва съобщение, гласящо, че собственикът на ново въведеното животно получава ваучер на стойност 15 лв. Извършва се и вкарване на нов запис в таблицата SERVES, като в зависимост от това къде живее клиента се определя ветеринарната клиника.

```
CREATE TRIGGER GETVOUCHER
AFTER INSERT ON ANIMALS
REFERENCING NEW AS N
FOR EACH ROW
BEGIN
    DECLARE V_SPECIES ANCHOR ANIMALS.SPECIES;
    DECLARE V_BRANCH_NAME ANCHOR BRANCHES.NAME;
    DECLARE V_ADDRESS ANCHOR CLIENTS.ADDRESS;
    DECLARE COUNT_SPECIES INT DEFAULT 0;
    DECLARE COUNT_SPECIES_PER_BRANCH INT DEFAULT 0;
    DECLARE COUNTER INT DEFAULT 0;
    DECLARE COUNT_ROWS INT DEFAULT 1;
    DECLARE CURSORBRANCHES CURSOR FOR SELECT NAME FROM BRANCHES;

    SET V_SPECIES = N.SPECIES;
    SELECT ADDRESS INTO V_ADDRESS FROM CLIENTS WHERE PHONE = N.CLIENT_PHONE;

    IF V_ADDRESS LIKE '%София%' THEN INSERT INTO SERVES (ANIMAL_CODE, BRANCH_NAME)
        VALUES (N.CODE, 'Пухкав свят - София');
    ELSEIF V_ADDRESS LIKE '%Варна%' THEN INSERT INTO SERVES (ANIMAL_CODE, BRANCH_NAME)
        VALUES (N.CODE, 'Пухкав свят - Варна');
    ELSEIF V_ADDRESS LIKE '%Велико Търново%' THEN INSERT INTO SERVES (ANIMAL_CODE, BRANCH_NAME)
        VALUES (N.CODE, 'Пухкав свят - Велико Търново');
    END IF;

    SELECT COUNT(*) INTO COUNT_ROWS FROM BRANCHES;
    OPEN CURSORBRANCHES;

    WHILE COUNTER < COUNT_ROWS DO
        FETCH FROM CURSORBRANCHES INTO V_BRANCH_NAME;
        CALL FN71959.FINDNUMBEROFSPECIES( P_SPECIES: V_SPECIES, P_BRANCH_NAME: V_BRANCH_NAME,
            V_COUNT: COUNT_SPECIES_PER_BRANCH);
        IF COUNT_SPECIES_PER_BRANCH = 1 THEN
            CALL DBMS_OUTPUT.PUT_LINE(V_BRANCH_NAME || ': ' || COUNT_SPECIES_PER_BRANCH
                || ' брой от вид ' || V_SPECIES);
        ELSE
            CALL DBMS_OUTPUT.PUT_LINE(V_BRANCH_NAME || ': ' || COUNT_SPECIES_PER_BRANCH
                || ' броя от вид ' || V_SPECIES);
        END IF;

        SET COUNT_SPECIES = COUNT_SPECIES + COUNT_SPECIES_PER_BRANCH;
        SET COUNTER = COUNTER + 1;
    END WHILE;
```



```
IF MOD(COUNT_SPECIES, 5) = 0 THEN
    CALL DBMS_OUTPUT.PUT_LINE('Собственикът, чийто телефон е ' || N.CLIENT_PHONE
                                || ', спечели ваучер на стойност 15 лв');
END IF;

CLOSE CURSORBRANCHES;
END;
```

## 15. Източници

- [dobrohrumvane.com](http://dobrohrumvane.com)
- [ovk-varna.com](http://ovk-varna.com)