

1. ОБХВАТ НА МОДЕЛА. ДЕФИНИРАНЕ НА ЗАДАЧАТА.

Базата от данни на ветеринарната клиника ще обслужва вътрешната система на клиниката, както и системата им за планиране на ресурси на предприятието. На практика това означава, че служителите ще имат достъп до информацията, която се съхранява в базата данни, за по лесно и ефикасно планиране и взимане на решения.

Клиниката има нужда да съхранява информация за различни обекти, количества консумативи, различни животни (пациенти), различни клиенти (собственици на животните) и документация на всеки преглед.

За всеки обект се пази информация за име и адрес на обекта.

За консумативите се пази информация за име на консуматива, цена и количество в даден обект.

За всеки клиент се пази информация за име, телефон и адрес.

За всяко животно се пази информация за име, вид животно, порода, възраст и собственик.

За всеки преглед се пази информация за прегледаното животно, описание на прегледа, цена на прегледа и обекта в който е извършен.

2. МНОЖЕСТВА ОТ СЪЩНОСТИ, ТЕХНИТЕ АТРИБУТИ И ДОМЕЙН

- Обекти
 - Идентификатор - естество число
 - Име - низ
 - Адрес - низ
- Консумативи
 - Идентификатор - естество число
 - Име - низ
 - Обект (идентификатор на обект) - естество число
 - Количество - реално число
 - Цена - реално число
- Клиенти
 - Идентификатор - естество число
 - Име - низ
 - Телефонен номер - низ
 - Адрес - низ
- Животни
 - Идентификатор - естество число
 - Име - низ
 - Вид - низ
 - Порода - ния
 - Възраст - естество число
 - Собственик (идентификатор на клиент) - естество число
- Прегледи
 - Идентификатор - естество число
 - Животно (идентификатор на животно) - естество число
 - Обект (идентификатор на обект) - естество число
 - Описание - низ
 - Цена - реално число

4. ВРЪЗКИ

- В един обект има много консумативи и един консуматив го има в много обекти.
- Всяко животно има точно един собственик и всеки клиент може да има много животни.
- Всеки преглед е на точно едно животно и е в точно един обект, във всеки обект може да има много прегледи и всяко животно може да има много прегледи в различни обекти.

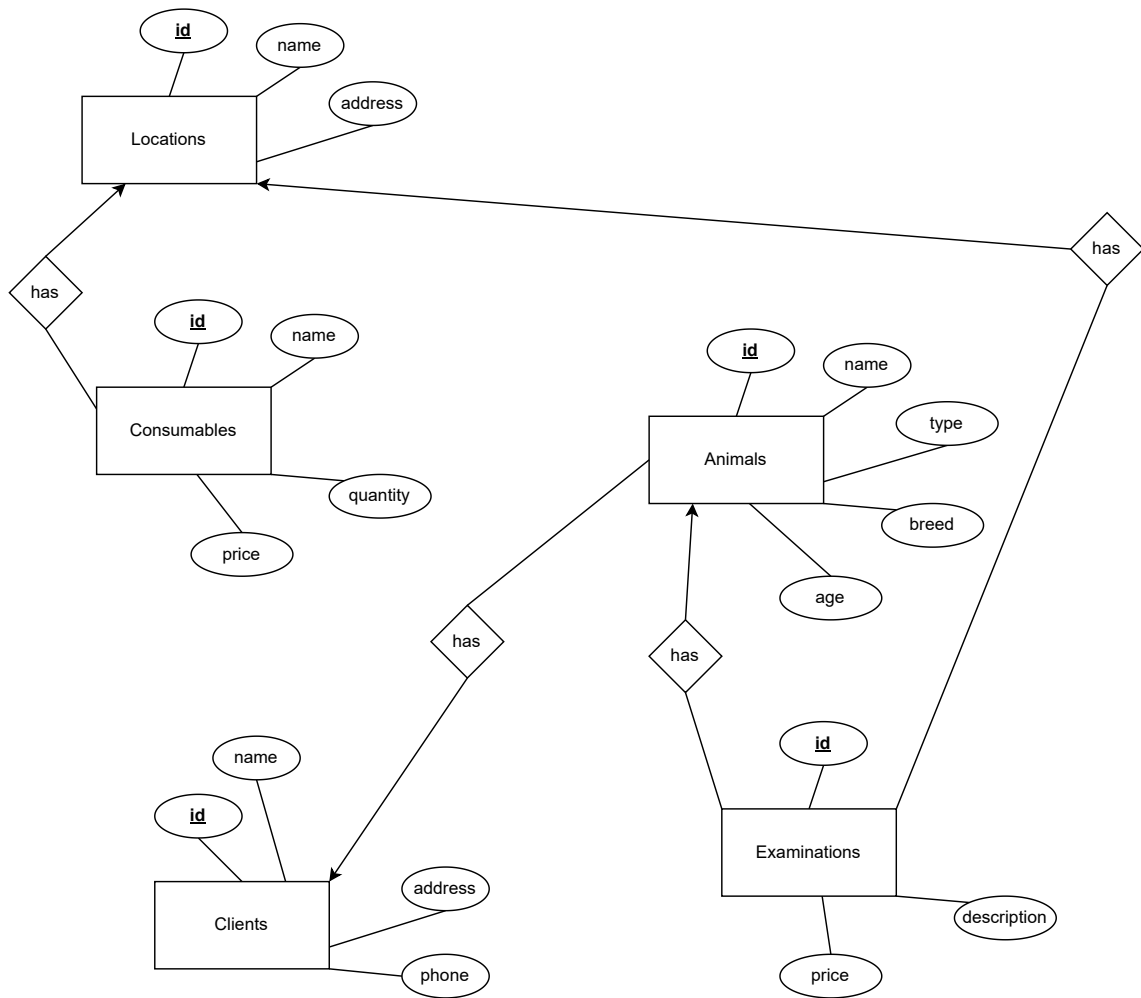
5. КЛЮЧОВЕ

Всеки обект има уникален идентификатор, който се използва за еднозначно достъпване.

6. ПРАВИЛА И ПРОВЕРКИ

Посочените полета трябва да имат уникални стойности:

- Клиенти
 - Идентификатор
 - Име
 - Адрес
- Консумативи
 - Идентификатор
 - Обект
- Клиенти
 - Идентификатор
 - Телефонен номер
- Животни
 - Идентификатор
- Прегледи
 - Идентификатор



SET SCHEMA FN72029;

```
CREATE TABLE Locations (  
    id integer NOT NULL GENERATED ALWAYS AS  
IDENTITY (START WITH 1 INCREMENT BY 1) PRIMARY  
KEY,  
    name VARCHAR(25),  
    address VARCHAR(25)  
);
```

```
CREATE TABLE Consumables (  
    id integer NOT NULL GENERATED ALWAYS AS  
IDENTITY (START WITH 1 INCREMENT BY 1) PRIMARY  
KEY,  
    name VARCHAR(25),  
    quantity double,  
    price double,  
    location integer NOT NULL,  
    FOREIGN KEY (location) references  
Locations(id)  
);
```

```
CREATE TABLE Clients (  
    id integer NOT NULL GENERATED ALWAYS AS  
IDENTITY (START WITH 1 INCREMENT BY 1) PRIMARY  
KEY,  
    name VARCHAR(25),  
    phone_number varchar(13),  
    address VARCHAR(25)  
);
```

```
CREATE TABLE Animals (  
    id integer NOT NULL GENERATED ALWAYS AS  
IDENTITY (START WITH 1 INCREMENT BY 1) PRIMARY  
KEY,  
    name VARCHAR(25),  
    type VARCHAR(25),  
    breed VARCHAR(25),  
    age integer,  
    owner integer NOT NULL,  
    FOREIGN KEY (owner) references Clients(id)  
);
```

```
CREATE TABLE Examinations (  
    id integer NOT NULL GENERATED ALWAYS AS  
IDENTITY (START WITH 1 INCREMENT BY 1) PRIMARY  
KEY,  
    animal integer NOT NULL,  
    location integer NOT NULL,  
    description VARCHAR(100),  
    price double,  
    FOREIGN KEY (animal) references  
Animals(id),  
    FOREIGN KEY (location) references  
Locations(id)  
);
```

```
SET SCHEMA FN72029;
```

```
INSERT INTO FN72029.Locations (name, address)
VALUES ('Sofia', 'addr1');
INSERT INTO FN72029.Locations (name, address)
VALUES ('Plovdiv', 'addr2');
INSERT INTO FN72029.Locations (name, address)
VALUES ('Varna', 'addr3');
INSERT INTO FN72029.Locations (name, address)
VALUES ('Pernik', 'addr4');
```

```
INSERT INTO FN72029.Consumables (name,
quantity, price, location) VALUES ('dog food',
5, 2, 1);
INSERT INTO FN72029.Consumables (name,
quantity, price, location) VALUES ('dog food',
10, 2, 2);
INSERT INTO FN72029.Consumables (name,
quantity, price, location) VALUES ('dog food',
2, 2, 3);
INSERT INTO FN72029.Consumables (name,
quantity, price, location) VALUES ('cat food',
6, 3, 1);
INSERT INTO FN72029.Consumables (name,
quantity, price, location) VALUES ('cat food',
14, 3, 2);
INSERT INTO FN72029.Consumables (name,
quantity, price, location) VALUES ('cat food',
7, 3, 3);
```

```
INSERT INTO FN72029.Consumables (name,  
quantity, price, location) VALUES ('cat food',  
12, 3, 4);  
INSERT INTO FN72029.Consumables (name,  
quantity, price, location) VALUES ('dog  
pills', 8, 15, 1);  
INSERT INTO FN72029.Consumables (name,  
quantity, price, location) VALUES ('dog  
pills', 5, 15, 2);  
INSERT INTO FN72029.Consumables (name,  
quantity, price, location) VALUES ('dog  
pills', 7, 15, 3);  
INSERT INTO FN72029.Consumables (name,  
quantity, price, location) VALUES ('dog  
pills', 5, 15, 4);  
  
INSERT INTO FN72029.Clients (name,  
phone_number, address) VALUES ('Ivan',  
'088888888', 'addr1');  
INSERT INTO FN72029.Clients (name,  
phone_number, address) VALUES ('Georgi',  
'089999999', 'addr2');  
INSERT INTO FN72029.Clients (name,  
phone_number, address) VALUES ('Boyan',  
'077777777', 'addr3');  
  
INSERT INTO FN72029.Animals (name, type,  
breed, age, owner) VALUES ('dogo', 'dog',  
'golden', 7, 1);
```

```
INSERT INTO FN72029.Animals (name, type,
breed, age, owner) VALUES ('dari', 'dog',
'street', 1, 3);
INSERT INTO FN72029.Animals (name, type,
breed, age, owner) VALUES ('lori', 'cat',
'street', 9, 2);

INSERT INTO FN72029.Examinations (animal,
location, description, price) VALUES
(1,1,'burz pregled', 50);
INSERT INTO FN72029.Examinations (animal,
location, description, price) VALUES
(2,1,'golyam pregled', 100);
INSERT INTO FN72029.Examinations (animal,
location, description, price) VALUES
(3,3,'sreden pregled', 80);

CREATE VIEW ClientAnimals
AS
    SELECT c.name as Client, a.name as
animal
    FROM Clients c, Animals a
    WHERE c.id = a.owner;

CREATE VIEW AnimalExamination
AS
    SELECT e.description as Examination,
a.name as animal
    FROM Examinations e, Animals a
```



```
WHERE e.animal = a.id;
```

Select заявка

```
1  const ibmdb = require('ibm_db');
2  const connStr = "DATABASE=SAMPLE;HOSTNAME=62.44.108.24;
   UID=db2admin;PWD=db2admin;PORT=50000;";
3
4  let query = (query) => {
5    return ibmdb.open(connStr).then(
6      conn => {
7        return conn.query(query).then(data => {
8          conn.closeSync();
9          return data;
10         }, err => {
11           console.log(err);
12         });
13       }, err => {
14         console.log(err);
15       })
16    )
17  }
18
19  module.exports.getAnimals = () => {
20    return query("SELECT * FROM FN72029.Animals;");
21  }
22
```

Insert заявка

```
38
39 module.exports.createLocation = (name, address) => {
40   return query(`INSERT INTO FN72029.Locations (name,
41     address) VALUES ('${name}', '${address}')`);
42 }
```

Express server

```
1  const express = require('express')
2  const app = express()
3  const port = 3000
4  const bodyParser = require('body-parser')
5  const dao = require('./dao');
6  let cors = require("cors");
7
8  app.use(cors());
9  app.use(bodyParser.json());
10
11 app.get('/animals', (req, res) => {
12   dao.getAnimals().then(response => {
13     console.log("GET /animals")
14     console.log(response)
15     console.log('--')
16     if(response) {
17       res.json(response)
18     } else {
19       res.status(400).send("Database error")
20     }
21   })
22 })
```

Frontend

```
1  async function getClients() {
2    let response = await fetch("http://127.0.0.1:3000/
  clients")
3
4    return response.json();
5  }
```

```
31  function loadClients() {
32    getClients().then(res => {
33      let table = document.getElementById("clients");
34
35      res.forEach(el => {
36        let row = document.createElement("tr");
37
38        let id = document.createElement("td");
39        let name = document.createElement("td");
40        let phone = document.createElement("td");
41        let address = document.createElement("td");
42
43        id.appendChild(document.createTextNode(el.ID))
44        name.appendChild(document.createTextNode(el.NAME))
45        phone.appendChild(document.createTextNode(el.
  PHONE_NUMBER))
46        address.appendChild(document.createTextNode(el.
  ADDRESS))
47
48        row.appendChild(id)
49        row.appendChild(name)
50        row.appendChild(phone)
51        row.appendChild(address)
52      })
53      table.appendChild(row)
54    })
55  }
56
59  module.exports.deleteExamination = (id) => {
60    return query(`DELETE FROM FN72029.Examinations WHERE id
  = ${id}`);
61  }
```