**Федеральное государственное автономное образовательное учреждение высшего образования**

**"Национальный исследовательский университет**

**"Высшая школа экономики"**

**Московский институт электроники и математики им. А.Н.Тихонова**

Департамент компьютерной инженерии

**Курс:** **Базы данных**

**Отчет по домашнему заданию**

Выполнили:

Студенты группы БИВ-152

Дубинина Мария Сергеевна  
Жалкова Наталия Евгеньевна

Москва 2017

**1. Инфологическое проектирование**

### 1.1. Анализ предметной области

База данных создаётся для информационного обслуживания руководства ветеринарной клиники, сотрудников клиники, клиентов клиники. БД должна содержать данные об отделениях клиники, врачах, диагнозах, предоставляемых услугах, животных, клиентах (хозяева животных).

В соответствии с предметной областью система строится с учётом следующих особенностей:

* Каждый врач работает в определённом отделении, в каждом отделении могут работать несколько врачей.
* Каждая услуга относится к определённому отделению, каждое отделение может специализироваться на оказании нескольких услуг.
* У каждого хозяина может быть несколько животных, у каждого животного один хозяин.
* Каждому животному могут быть поставлены несколько диагнозов, каждый диагноз может быть поставлен нескольким животным.
* Каждый врач может оказывать несколько услуг из перечня услуг отделения, к которому он относится, каждая услуга может быть оказана несколькими врачами отделения, которое на ней специализируется.
* Каждый врач может принять несколько животных, каждое животное может быть записано на прием к нескольким врачам.
* Каждое животное может получить несколько услуг, каждая услуга может быть оказана нескольким животным.
* Диагноз ставится животному после оказания всех необходимых услуг для его постановки (осмотр, проведение анализов).
* Врач не может проводить прием разных пациентов в один и тот же промежуток времени.
* Степень тяжести диагноза: легкая, средняя, тяжелая.
* Роль в отделении врача: руководитель, заместитель руководителя, сотрудник, стажер.

Для создания ER-модели необходимо выделить сущности предметной области:

1. **Клиенты.** Атрибуты: ФИО, контактный телефон, адрес, пол.
2. **Животные.** Атрибуты: кличка, вид, порода, вес, высота, ширина, длина, пол, дата рождения.
3. **Врачи.** Атрибуты: паспортные данные, ФИО, пол, ИНН, телефоны, образование, оклад, должность, дата рождения, адреса, логин, роль в отделении, специальность.
4. **Отделения.** Атрибуты: номер, название, кабинеты, телефоны.
5. **Диагнозы.** Атрибуты: название, симптомы, степень тяжести.
6. **Услуги.** Атрибуты: название, требуемое оборудование, лекарства, стоимость.

Исходя из выявленных сущностей, построим ER–диаграмму (рис. 1).

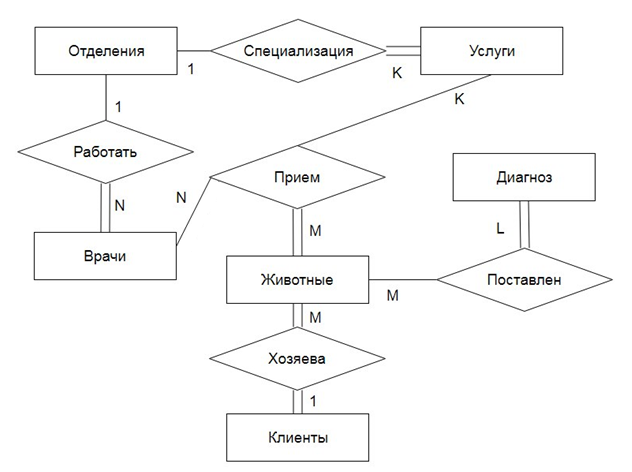


Рис. 1. ER–диаграмма ПрО «Ветеринарная клиника»

### 1.2. Анализ информационных задач и круга пользователей системы

Определим группы пользователей, их основные задачи и запросы к БД:

1. **Руководство ветеринарной клиники**:

* получение полной информации о пациентах, клиентах, врачах;
* внесение изменений в предоставляемые услуги;
* добавление/удаление услуги из перечня;
* назначение услуги отделению;
* назначение врача в отделение;
* изменение должностных окладов и штатного расписания врачей;
* создание/роспуск отделений;
* назначение руководителей отделений.

1. **Руководители отделений:**

* изменение, добавление, удаление контактной информации об отделении;
* назначение заместителя.

1. **Клиенты клиники**:

* получение списка услуг и их стоимости;
* получение полной информации о животном, хозяином которого он является;
* просмотр определенной информации о врачах;
* получение контактной информации отделений.

1. **Врачи**:

* добавление животных;
* изменение, удаление информации о животном;
* просмотр информации о животных;
* получение контактной информации о клиенте;
* постановка диагнозов животным;
* добавление диагнозов;
* изменение, добавление, удаление информации о диагнозах;
* получение и изменение информации о приемах, проводимых данным врачом.

1. **Сотрудники регистрации:**

* добавление клиентов;
* изменение, удаление информации о клиентах.

1. **Сотрудники отдела кадров**:

* приём/увольнение врачей;
* внесение изменений в данные о врачах.

1. **Бухгалтеры**:

* получение ведомости на выплату зарплаты.

**2. Логическое проектирование реляционной БД**

### 2.1. Преобразование ER–диаграммы в схему базы данных

Связь типа 1:n (один-ко-многим) между отношениями реализуется через внешний ключ. Ключ вводится для дочернего отношения. Внешнему ключу должен соответствовать первичный или уникальный ключ основного (родительского) отношения.

Связь типа n:m реализуется через вспомогательное отношение, которое содержит комбинации первичных ключей соответствующих исходных отношений.

Бинарная связь между отношениями не может быть обязательной для обоих отношений. Поэтому для такой связи необходимо снять с одной стороны условие обязательности.

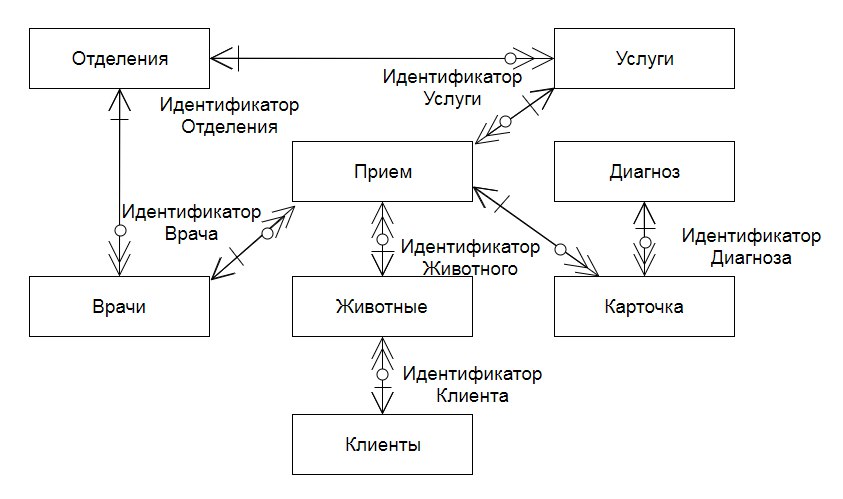


Рис. 2. Схема РБД

### 2.2. Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности. Для каждого отношения определяются первичный ключ и внешние ключи. Отношения приведены в табл. 1-8. Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной. Типы данных обозначаются так: N – числовой, C – символьный тип фиксированной длины, V – символьный тип переменной длины, D – дата (этот тип имеет стандартную длину, зависящую от СУБД, поэтому она не указывается).

Потенциальными ключами отношения ОТДЕЛЕНИЕ являются атрибуты *Номер отделения* и *Название отделения*. Первый занимает меньше места и имеет числовой тип, поэтому мы выбираем его в качестве первичного ключа.

Таблица 1. Схема отношения ОТДЕЛЕНИЯ (Departs)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Номер отделения | Number | N(4) | **Первичный ключ** |
| Название отделения | Name | V(100) | Обязательное поле |
| Кабинеты | Rooms | V(20) | Обязательное многозначное поле |
| Телефоны | Phone | V(40) | Обязательное многозначное поле |

Потенциальным ключом отношения КЛИЕНТЫ являются только атрибут *ФИО*. Но ФИО может совпасть у двух клиентов, поэтому в качестве первичного ключа вводим суррогатный первичный ключ - идентификатор.

Таблица 2. Схема отношения КЛИЕНТЫ (Clients)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(7) | **Суррогатный первичный ключ** |
| ФИО | Name | V(100) | Обязательное поле |
| Пол | Sex | C(1) | Обязательное поле, 'м' или 'ж' |
| Контактный телефон | Phone | V(10) | Обязательное поле |
| Адрес | Address | V(100) | Обязательное поле |

В отношении ЖИВОТНЫЕ нет потенциального первичного ключа, поэтому в качестве первичного ключа вводим суррогатный первичный ключ - идентификатор.

Таблица 3. Схема отношения ЖИВОТНЫЕ (Pets)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(7) | **Суррогатный первичный ключ** |
| Кличка | Name | V(30) | Обязательное поле |
| Хозяин | Owner | N(7) | Внешний ключ (к Clients) |
| Вид | Species | V(30) | Обязательное поле |
| Порода | Breed | V(50) | Необязательное поле |
| Вес | Weight | N(5,5) | Необязательное поле |
| Высота | Height | N(5,5) | Необязательное поле |
| Ширина | Width | N(5,5) | Необязательное поле |
| Длина | Length | N(5,5) | Необязательное поле |
| Пол | Sex | C(1) | Необязательное поле |
| Дата рождения | Birth | D | Необязательное поле |

Потенциальными ключами отношения ВРАЧИ являются поля *Паспортные данные* и *ИНН*. Все они занимают достаточно много места, а паспортные данные кроме того могут меняться. Введём суррогатный первичный ключ *идентификатор врача*.

Таблица 4. Схема отношения ВРАЧИ (Doctors)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(4) | **Суррогатный первичный ключ** |
| ФИО | Name | V(100) | Обязательное поле |
| Пол | Sex | C(1) | Обязательное поле, 'м' или 'ж' |
| Дата рождения | Birth | D | Обязательное поле |
| Паспортные данные | Passport | V(50) | Обязательное уникальное поле |
| ИНН | Inn | C(12) | Обязательное уникальное поле |
| Отделение | Depart | N(4) | Внешний ключ (к Departs) |
| Образование | Education | V(200) | Обязательное многозначное поле |
| Специальность | Specialization | V(100) | Обязательное поле |
| Должность | Position | V(50) | Обязательное поле |
| Оклад | Salary | N(7) | Обязательное поле, >15000 руб. |
| Телефоны | Phone | V(40) | Обязательное многозначное поле |
| Адреса | Address | V(200) | Обязательное многозначное поле |
| Роль в отделении | Role | V(25) | Обязательное поле |
| Логин | Login | V(30) | Обязательное поле |

Потенциальным ключом отношения ДИАГНОЗЫ является поле *Название*. Но оно занимает достаточно много места. Введём суррогатный первичный ключ *идентификатор диагноза.*

Таблица 5. Схема отношения ДИАГНОЗЫ (Diagnoses)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(4) | **Суррогатный первичный ключ** |
| Название | Name | V(100) | Обязательное уникальное поле |
| Симптомы | Symptoms | V(300) | Обязательное многозначное поле |
| Степень тяжести | Severity | N(1) | Обязательное поле, 1 - легкая, 2 - средняя, 3 - тяжелая |

Потенциальным ключом отношения УСЛУГИ является поле *Название*. Но оно занимает достаточно много места. Введём суррогатный первичный ключ *идентификатор услуги.*

Таблица 6. Схема отношения УСЛУГИ (Services)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(2) | **Суррогатный первичный ключ** |
| Название | Name | V(100) | Обязательное уникальное поле |
| Требуемое оборудование | Equipment | V(200) | Обязательное многозначное поле |
| Лекарства | Medicine | V(150) | Обязательное многозначное поле |
| Стоимость | Cost | N(6) | Обязательное поле |
| Отделение | Depart | N(4) | Внешний ключ (к Depart) |

В качестве потенциального ключа в отношении ПРИЕМ может быть выбрана комбинация поля *Врач* и *Дата и время*, так как один и тот же врач не может проводить несколько приемов в одно и тоже время, но использования типа datatime не удобно в качестве первичного ключа, поэтому введем суррогатный первичный ключ.

Таблица 7. Схема отношения ПРИЕМ (Appointment)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(5) | **Суррогатный первичный ключ** |
| Врач | Doctor | N(4) | Внешний ключ (к Doctors) |
| Дата и время | Date\_Time | D | Обязательное поле |
| Животное | Pet | N(7) | Внешний ключ (к Pets) |
| Услуга | Service | N(2) | Внешний ключ (к Services) |
| Кабинет | Room | N(3) | Обязательное поле |

В качестве первичного ключа в отношении КАРТОЧКА может быть выбрана комбинация полей Животное, диагноз, дата постановки. Но так как такой первичный ключ слишком большой, то введём суррогатный первичный ключ номер карточки.

Таблица 8. Схема отношения КАРТОЧКА (Card)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Номер | Id | N(5) | **Суррогатный первичный ключ** |
| Диагноз | Diagnose | N(4) | Внешний ключ (к Diagnoses) |
| Прием | Appoint | N(5) | Внешний ключ (к Appointment) |
| Дата выздоровления | Rec\_date | D | Необязательное поле |

### 2.3. Нормализация полученных отношений

На первом этапе вынесем многозначные атрибуты в отдельные отношения.

**1НФ.** Для приведения таблиц к 1НФ требуется составить прямоугольные таблицы (одно значение атрибута – одна ячейка таблицы) и разбить сложные атрибуты на простые.

***Отношение ОТДЕЛЕНИЯ.***Многозначные атрибуты *Кабинеты* и *Телефоны* из отношения вынесем в отдельное отношение КАБИНЕТЫ. Так как поле *Номер кабинета* является уникальным атрибутом, сделаем его первичным ключом. Так как в отношении ПРИЕМ есть атрибут *Кабинет,* то его необходимо сделать внешним ключом к отношению КАБИНЕТЫ

***Отношение КЛИЕНТЫ.*** Разделим атрибут *ФИО* на два атрибута *Фамилия* и *Имя*, *отчество*.

***Отношение ВРАЧИ.*** Разделим атрибут *ФИО* на два атрибута *Фамилия* и *Имя, отчество*, *Паспортные данные* на *Номер паспорта* (уникальный), *Дата выдачи* и *Кем выдан*, а *Образование* – на *Вид образования*, *Специальность*, *Номер диплома* и *Год окончания* учебного заведения. Данные об образовании выделим в отдельное отношение ОБРАЗОВАНИЕ. *Телефоны* и *адреса* врачей разделим в отношение АДРЕСА-ТЕЛЕФОНЫ. В отношениях АДРЕСА-ТЕЛЕФОНЫ и ОБРАЗОВАНИЕ нельзя выделить первичные ключи, но эти отношения можно оставить без первичного ключа, т.к. на них никто не ссылается.

***Отношение ДИАГНОЗЫ.*** Выделим поле *Симптомы* в отдельное отношение СИМПТОМЫ, а связь между отношениями ДИАГНОЗЫ и СИМПТОМЫ в отношении СИМПТОМ-ДИАГНОЗ.

***Отношение УСЛУГИ.*** Выделим поле *Требуемое оборудование* в отдельное отношение ОБОРУДОВАНИЕ, а связь между отношениями УСЛУГИ и ОБОРУДОВАНИЕ в отношении УСЛУГА-ОБОРУДОВАНИЕ. Аналогично поступим с полем *Лекарства,* выделив его в отношение ЛЕКАРСТВА, связь между отношениями УСЛУГИ и ЛЕКАРСТВА определим в отношении УСЛУГА-ЛЕКАРСТВО.

**2НФ.** В нашем случае составные первичные ключи имеет только отношение ПРИЕМ. Неключевые атрибуты этого отношения функционально полно зависят от составного первичного ключа.

**3НФ.** В отношении ВРАЧИ атрибут *Оклад* зависит от атрибута *Должность*. Поэтому выделим его в отношение ДОЛЖНОСТИ, перенесём в него атрибуты *Должность* и *Оклад*, а первичным ключом сделаем название должности.

В отношениях ВРАЧИ атрибуты *Дата выдачи* и *Кем выдан*, а также в отношении ОБРАЗОВАНИЕ *Номер диплома* и *Год окончания учебного заведения* зависят не от первичного ключа, а от атрибутов соответственно *Номер паспорта* и *Специальность*. Но если мы выделим их в отдельное отношение, то получим связи типа 1:1. Следовательно, здесь декомпозиция нецелесообразна.

**4НФ.** Отношение АДРЕСА-ТЕЛЕФОНЫ нарушают 4НФ, т.к. не всякий телефон привязан к конкретному адресу (т.е. мы имеем две многозначных зависимости в одном отношении). Но выделять *Телефоны* в отдельное отношение не стоит, т.к. эти сведения носят справочный характер и не требуется их автоматическая обработка.

Отношения, полученные после нормализации, приведены в табл. 8-26.

Таблица 9. Схема отношения ОТДЕЛЕНИЯ (Departs)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Номер отделения | Number | N(4) | **Первичный ключ** |
| Название отделения | Name | V(100) | Обязательное поле |

Таблица 10. Схема отношения КАБИНЕТЫ (Rooms)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Номер кабинета | Room\_Num | N(3) | **Первичный ключ** |
| Отделение | Depart | N(4) | Внешний ключ (к Departs) |
| Телефон | Phone | V(20) | Необязательное поле |

Таблица 11. Схема отношения КЛИЕНТЫ (Clients)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(7) | **Суррогатный первичный ключ** |
| Фамилия | Surname | V(50) | Обязательное поле |
| Имя, Отчество | Name | V(50) | Обязательное поле |
| Пол | Sex | C(1) | Обязательное поле, 'м' или 'ж' |
| Контактный телефон | Phone | V(10) | Обязательное поле |
| Адрес | Address | V(100) | Обязательное поле |

Таблица 12. Схема отношения ДОЛЖНОСТИ (Posts)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Должность | Post | V(50) | **Первичный ключ** |
| Оклад | Salary | N(7) | Обязательное поле, > 15000 руб. |

Таблица 13. Схема отношения ВРАЧИ (Doctors)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(4) | **Суррогатный первичный ключ** |
| Фамилия | Surname | V(50) | Обязательное поле |
| Имя, Отчество | Name | V(50) | Обязательное поле |
| Пол | Sex | C(1) | Обязательное поле, 'м' или 'ж' |
| Дата рождения | Birth | D | Обязательное поле |
| Серия и номер паспорта | Pass\_Num | С(10) | Обязательное уникальное поле |
| Когда выдан паспорт | Pass\_Date | D | Обязательное поле |
| Кем выдан паспорт | Pass\_Given | V(50) | Обязательное поле |
| ИНН | Inn | C(12) | Обязательное уникальное поле |
| Отделение | Depart | N(4) | Внешний ключ (к Departs) |
| Должность | Position | V(50) | Внешний ключ (к Posts) |
| Роль в отделении | Role | V(25) | Обязательное поле |
| Логин | Login | V(30) | Обязательное поле |

Таблица 14. Схема отношения ОБРАЗОВАНИЕ (Education)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор врача | Id | N(4) | Внешний ключ (к Doctors) |
| Вид образования | Type | V(20) | Обязательное поле |
| Специальность | Specialization | V(40) | Необязательное поле |
| Номер диплома | Diploma | V(15) | Необязательное поле |
| Год окончания учебного заведения | Grad\_year | N(4) | Обязательное поле |

Таблица 15. Схема отношения АДРЕСА-ТЕЛЕФОНЫ (Address-Phone)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор врача | Id | N(4) | Внешний ключ (к Doctors) |
| Адрес | Address | V(50) | Необязательное поле |
| Телефон | Phone | V(30) | Необязательное поле |

Таблица 16. Схема отношения ДИАГНОЗЫ (Diagnoses)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(4) | **Суррогатный первичный ключ** |
| Название | Name | V(100) | Обязательное уникальное поле |
| Степень тяжести | Severity | N(1) | Обязательное поле, 1 - легкая, 2 - средняя, 3 - тяжелая |

Таблица 17. Схема отношения СИМПТОМЫ (Symptoms)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(5) | **Первичный ключ** |
| Название | Name | V(100) | Обязательное уникальное поле |

Таблица 18. Схема отношения СИМПТОМЫ-ДИАГНОЗЫ (Diag-Symp)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Симптом | Symptom | N(5) | Внешний ключ (к Symptoms) |
| Диагноз | Diagnose | N(4) | Внешний ключ (к Diagnoses) |

Таблица 19. Схема отношения УСЛУГИ (Services)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(2) | **Суррогатный первичный ключ** |
| Название | Name | V(100) | Обязательное уникальное поле |
| Стоимость | Cost | N(6) | Обязательное поле |
| Отделение | Depart | N(4) | Внешний ключ (к Depart) |

Таблица 20. Схема отношения ЛЕКАРСТВА (Medicines)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(4) | **Первичный ключ** |
| Название | Name | V(50) | Обязательное уникальное поле |

Таблица 21. Схема отношения УСЛУГИ-ЛЕКАРСТВА (Serv-Med)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Услуга | Service | N(5) | Внешний ключ (к Services) |
| Лекарство | Medicine | N(4) | Внешний ключ (к Medicines) |

Таблица 22. Схема отношения ОБОРУДОВАНИЕ (Equipment)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(2) | **Первичный ключ** |
| Название | Name | V(100) | Обязательное уникальное поле |

Таблица 23. Схема отношения УСЛУГИ-ОБОРУДОВАНИЕ (Serv-Equip)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Услуга | Service | N(5) | Внешний ключ (к Services) |
| Оборудование | Equipment | N(2) | Внешний ключ (к Equipment) |

Таблица 24. Схема отношения ПРИЕМ (Appointment)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(5) | **Суррогатный первичный ключ** |
| Врач | Doctor | N(4) | Внешний ключ (к Doctors) |
| Дата и время | Date\_Time | D | Обязательное поле |
| Животное | Pet | N(7) | Внешний ключ (к Pets) |
| Услуга | Service | N(2) | Внешний ключ (к Services) |
| Кабинет | Room | N(3) | Внешний ключ (к Rooms) |

Таблица 25. Схема отношения КАРТОЧКА (Card)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Номер | Id | N(5) | **Суррогатный первичный ключ** |
| Диагноз | Diagnose | N(4) | Внешний ключ (к Diagnoses) |
| Прием | Appoint | N(5) | Внешний ключ (к Appointment) |
| Дата выздоровления | Rec\_date | D | Необязательное поле |

Таблица 26. Схема отношения ЖИВОТНЫЕ (Pets)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| Идентификатор | Id | N(7) | **Суррогатный первичный ключ** |
| Кличка | Name | V(30) | Обязательное поле |
| Хозяин | Owner | N(7) | Внешний ключ (к Clients) |
| Вид | Species | V(30) | Обязательное поле |
| Порода | Breed | V(50) | Необязательное поле |
| Вес | Weight | N(5,5) | Необязательное поле |
| Высота | Height | N(5,5) | Необязательное поле |
| Ширина | Width | N(5,5) | Необязательное поле |
| Длина | Length | N(5,5) | Необязательное поле |
| Пол | Sex | C(1) | Необязательное поле |
| Дата рождения | Birth | D | Необязательное поле |

Схема базы данных после нормализации приведена на рис. 3.

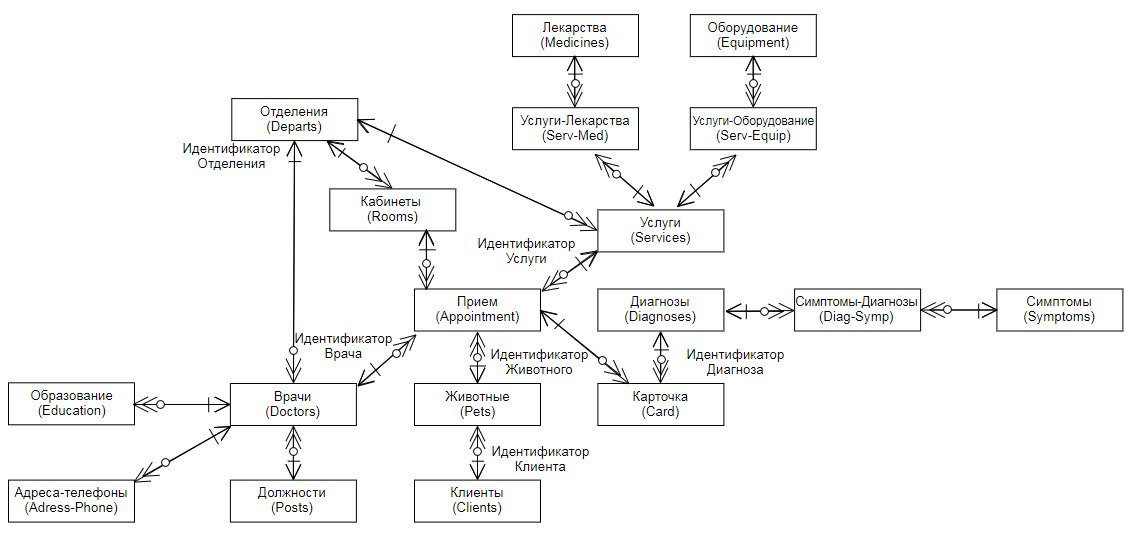


Рис. 3. Окончательная схема БД

### 2.4. Определение дополнительных ограничений целостности

Перечислим ограничения целостности, которые не указаны в табл. 8–26:

1. Атрибут *Вид образования* может принимать одно из следующих значений: 'начальное', 'среднее', 'средне-специальное', 'высшее'.
2. Атрибут *Роль* *в отделении* может принимать одно из четырех значений: 'руководитель', 'заместитель руководителя', 'сотрудник', 'стажер'.
3. *Дата рождения, Дата выдачи паспорта,* *Дата приема* и *Дата постановки* диагнозане может быть больше текущей даты.
4. Значение атрибута *Дата выздоровления* не может быть меньше чем *Дата постановки* диагноза.
5. Атрибут *Стоимость* не может быть отрицательным.
6. Ни один идентификатор не может быть отрицательным.

### 2.5. Описание групп пользователей и прав доступа

Опишем для каждой группы пользователей права доступа к каждой таблице. Права доступа должны быть распределены так, чтобы для каждого объекта БД был хотя бы один пользователь, который имеет право добавлять и удалять данные из объекта. Права приведены в табл. 27. Используются следующие сокращения:

s – чтение данных (select);

u – модификация данных (update);

i – добавление данных (insert);

d – удаление данных(delete).

Таблица 27. Права доступа к таблицам для групп пользователей

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Таблицы** | **Группы пользователей (роли)** | | | | | | |
| **Руководство ветеринарной клиники** | **Руководители отделений** | **Клиенты клиники** | **Врачи** | **Сотрудники регистрации** | **Сотрудники отдела кадров** | **Бухгалтеры** |
| **Отделения** | SUID | SUID | S | S | S | S | S |
| **Кабинеты** | SUID | SUID | S | S | S | S | S |
| **Животные** | S | S | S | SUID | S |  |  |
| **Клиенты** | S | S |  | S | SUID |  |  |
| **Должности** | SUID | S |  |  | S | S | S |
| **Врачи** | SU | SU | S |  | S | SUID | S |
| **Образование** | S | S |  |  |  | SUID | S |
| **Адреса-телефоны** | S | S |  |  |  | SUID | S |
| **Диагнозы** | S | S |  | SUID |  |  |  |
| **Симптомы** | S | S |  | SUID |  |  |  |
| **Симптомы-диагнозы** | S | S |  | SUID |  |  |  |
| **Услуги** | SUID | SU | S | S |  |  |  |
| **Лекарства** | SUID | S |  | S |  |  |  |
| **Услуги-лекарства** | SUID | S |  | S | S |  |  |
| **Оборудование** | SUID | S |  | S |  |  |  |
| **Услуги-оборудование** | SUID | S |  | S | S |  |  |
| **Прием** | S | S | S | S | SUID |  |  |
| **Карточка** | S | S | S | SUID |  |  |  |

## 3. Реализация проекта базы данных

### 3.1. Создание таблиц

1. Отношение Departs (отделы):

CREATE TABLE Departs (

DNumber numeric(4) Primary key,

Name varchar(100) Not null);

1. Отношение Rooms (кабинеты):

CREATE TABLE Rooms (

Room\_Num numeric(3) primary key,

Depart numeric(4) REFERENCES Departs(DNumber),

Phone varchar(20) not null);

1. Отношение Clients (Клиенты):

CREATE TABLE Clients (

Id numeric(7) primary key,

Surname varchar(50) not null,

Name varchar(50) not null,

Sex char(1) not null,

Phone varchar(10) not null,

Address varchar(100) not null);

1. Отношение Pets (животные):

CREATE TABLE Pets (

Id numeric(7) primary key,

Name varchar(30) not null,

Owner numeric(7) references Clients(Id),

Species varchar(30) not null,

Breed varchar(50),

Weight numeric(5,5),

Height numeric(5,5),

Width numeric(5,5),

Length numeric(5,5),

Sex char(1),

Birth date);

1. Отношение Posts (должности):

CREATE TABLE Posts (

Post varchar(50) primary key,

Salary numeric(7) not null,

check(p\_salary>=15000)));

1. Отношение Doctors (доктора):

CREATE TABLE Doctors (

Id numeric(4) primary key,

Surname varchar(50) not null,

Name varchar(50) not null,

Sex char(1) not null,

Birth date not null,

Pass\_Num char(10) not null unique,

Pass\_Date date not null,

Pass\_Given varchar(50) not null,

Inn char(12) not null unique,

Depart numeric(4) references Departs(Number),

Position varchar(50) references Posts(Post),

Role varchar(25) not null,

Login varchar(30) not null,

check(Sex in ('ж','м')),

check(Pass\_Date >= curdate())));

1. Отношение Education (образование):

CREATE TABLE Education (

Id numeric(4) references Doctors(Id),

Type varchar(20) not null,

Specialization varchar(40),

Diploma varchar(15),

Grad\_year numeric(4) not null,

check(Specialization in ('начальное', 'среднее', 'высшее','средне-специальное'))));

1. Отношение Address\_Phone (адреса-телефоны):

CREATE TABLE Address\_Phone (

Id numeric(4) references Doctors(Id),

Address varchar(50),

Phone varchar(30));

1. Отношение Diagnoses (диагнозы):

CREATE TABLE Diagnoses (

Id numeric(4) primary key,

Name varchar(100) not null unique,

Severity numeric(1) not null);

1. Отношение Symptoms (симптомы):

CREATE TABLE Symptoms (

Id numeric(5) primary key,

Name varchar(100) not null unique);

1. Отношение Diag\_Symp (диагнозы-симптомы):

CREATE TABLE Diag\_Symp (

Symptom numeric(5) references Symptoms(Id),

Diagnose numeric(4) references Diagnoses(Id));

1. Отношение Services (услуги):

CREATE TABLE Services (

Id numeric(2) primary key,

Name varchar(100) not null unique,

Cost numeric(6) not null,

Depart numeric(4) references Departs(Number));

1. Отношение Medicines (лекарства):

CREATE TABLE Medicines (

Id numeric(4) primary key,

Name varchar(50) not null unique);

1. Отношение Serv\_Med (услуги-лекарства):

CREATE TABLE Serv\_Med (

Service numeric(5) references Services(Id),

Medicine numeric(4) references Medicines(Id));

1. Отношение Equipment (оборудование):

CREATE TABLE Equipment (

Id numeric(2) primary key,

Name varchar(100) not null unique);

1. Отношение Serv-Equip (услуги-оборудование):

CREATE TABLE Serv\_Equip (

Service numeric(5) references Services(Id),

Equipment numeric(2) references Equipment(Id));

1. Отношение Appointment (прием):

CREATE TABLE Appointment (

Id numeric(5) primary key,

Doctor numeric(4) references Doctors(Id),

Date\_Time datetime,

Pet numeric(7) references Pets(Id),

Service numeric(2) references Services(Id),

Room numeric(3) references Rooms(Rum\_Num));

18.Отношение Card (карточка):

CREATE TABLE Card (

Id numeric(5) primary key,

Diagnose numeric(4) references Diagnoses(Id),

Appoint numeric(5) references Appointment(Id),

Doctor numeric(4) not null,

Rec\_Date date);

### 3.2. Создание представлений

1) Будущее расписание врача (начиная с сегодняшней даты) c идентификатором 1:

create or replace view Doc\_Sch

as select a.Date\_Time, a.Room as 'Room', p.name as 'Pet', s.Name as 'Service'

from Appointment a, Doctors d, Pets p, Services s

where ((d.Id = 1) AND (a.Doctor = d.Id) AND (a.Date\_Time > curdate()) AND (a.Pet = p.Id) AND (a.Service = s.Id));

2) Перечень услуг и их стоимость:

create or replace view Serv\_Cost

as select s.name, s.cost

from Services s;

3) Данные об отделениях для руководителя отделения:

create or replace view Depart\_data

as select dp.\*, r.Room\_Num, r.Phone

from Departs as dp, Rooms as r, Doctors as d

where d.Login = user and d.Depart = dp.Number

and d.Role = 'Руководитель'

4) Данные о сотрудниках отделения для руководителя:

create or replace Dep\_work

as select dc.\*

from Departs d, Doctors dc, Doctors dhead

where ((dhead.role = 'руководитель') AND (dhead.login = user)AND (dhead.Depart = d.Number) AND (dc.Depart = d.Number));

5) Выписка на получение зарплаты:

create or replace Salary

as select d.Surname, d.Name, p.Salary from Doctors as d, Posts as p WHERE p.Post = d.Position;

6) Текущие больные:

Create or replace view Cur\_Sick

as select distinct cl.name as 'Owner name', p.\*, d.name as 'Diagnose'

from Pets p, Card c, Appointment a, Clients cl, Diagnoses d where ((p.Owner = cl.Id) AND (c.Rec\_Date IS NULL) AND (c.Appoint = a.Id) AND (c.Diagnose = d.id) AND (a.Pet = p.Id));

7) Информация о животных, которым был поставлен определенный диагноз (Здесь Перелом):

create or replace view Animal\_Ill

as select p.\*, a.Date\_Time

from Pets p, Card c, Appointment a, Diagnoses d

where (c.Appoint = a.Id) AND (a.Pet = p.Id) AND(c.Diagnose = d.id) AND (d.name = 'Перелом');

8) Свободные сегодня врачи определенного отделения (Здесь №1):

Create or replace Depart\_Doc

as select d.\*, dc.surname as 'Doctor surname', dc.name as 'Doctor name'

from Departs d, Doctors dc

where ((d.Number = 1) AND (dc.Depart = d.Number) AND dc.Id NOT IN

(select Doctor from Appointment where year(Date\_Time) = year(curdate())

AND month(Date\_Time) = month(curdate())

AND day(Date\_Time) = day(curdate())));

9) Личный кабинет клиента (Здесь с идентификатором 1). Информация о всех питомцах данного клиента, дате приемов, врачах, комнатах, где проводятся данные приемы.

create or replace Cli\_Acc

as select p.name, p.species, a.Date\_Time, d.surname as 'Doctor surname',

d.name as 'Doctor name', a.Room as 'Room', s.Name, s.cost

from Pets p, Appointment a, Services s, Doctors d

where ((p.Owner = 1) AND (a.Pet = p.Id) AND (a.Service = s.Id) AND a.Doctor = d.Id));

10) Возможные для постановки пациенту диагнозы на определенный симптом

Create or replace Dia\_Symp

as select d.\*

from Diagnoses d, Symptoms s, Diag\_Symp ds

where ((ds.Symptom = s.Id) AND (ds.Diagnose = d.Id)

AND (s.Name ='тошнота'));

Для того чтобы можно было работать с этими представлениями, соответствующим пользователям нужно назначить права доступа к представлениям. Эти права перечислены в табл. 28.

Таблица 28. Права доступа к представлениям

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Представления** | **Группы пользователей (роли)** | | | | | |
| **Руководство ветеринарной клиники** | **Руководители отделений** | **Клиенты клиники** | **Врачи** | **Сотрудники регистрации** | **Бухгалтеры** |
| **Расписание врача (Doc\_Sch)** | S | S | S | S | S |  |
| **Перечень услуг (Serv\_Cost)** | SUID | S | S | S | S |  |
| **Отделения для руководителей (Depart\_data)** | S | S |  |  |  |  |
| **Сотрудники отделения (Dep\_work)** | S | S |  |  |  |  |
| **Выписка по зарплате (Salary)** |  |  |  |  |  | S |
| **Текущие больные (Cur\_Sick)** | S | S |  | S |  |  |
| **Животные по диагнозу (Animal\_Ill)** | S | S |  | S |  |  |
| **Свободные на текущий день врачи отделения (Depart\_Doc)** | S | S | S | S | S |  |
| **Личный кабинет клиента (Cli\_Acc)** |  |  | S |  |  |  |
| **Диагнозы по симптому (Dia\_Symp)** |  | S | S |  |  |  |

### 3.3. Назначение прав доступа

**Примеры:**

Права доступа на таблицу Отделы для руководства клиники(ruk\_user):

grant select, insert, update, delete on Departs to ruk\_user;

Права доступа на таблицу Врачи для клиентов(client):

grant select on Doctors to ruk\_user;

Права доступа на представление Диагнозы по симптому для руководителей отделений(rd\_user):

grant select on Dia\_Symp to ruk\_user;

Права доступа на представление Расписание врача для сотрудников регистрации (reg\_user):

grant select on Doc\_Sch to ruk\_user;

### 3.4. Создание индексов

**Примеры:**

create index D\_Position on Doctors(Position);   
create index S\_Depart on Services(Depart);   
create index A\_Pet on Appointment(Pet);   
create index A\_Service on Appointment(Service);   
create index A\_Room on Appointment(Room); 