**Министерство транспорта Российской Федерации**

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«РУТ (МИИТ)»**

**Институт транспортной техники и систем управления**

**Кафедра «Управление и защита информации»**

КУРСОВОЙ ПРОЕКТ

по дисциплине

**«Основы построения защищенных баз данных»**

**на тему  
«Поликлиника»**

Выполнил: ст. гр. ТКИ-441

Куминов В. П.

Проверил: доц., к.т.н.

Васильева М. А.

Москва 2023

Оглавление

[ЦЕЛЬ КУРСОВОГО ПРОЕКТА \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5](#_Toc135052254)

[ЗАДАНИЕ НА КУРСОВОВОЙ ПРОЕКТ\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_5](#_Toc135052255)

[ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ\_\_\_\_\_\_\_\_\_6](#_Toc135052256)

[1 Инфологическое проектирование 6](#_Toc135052257)

[1.1 Анализ предметной области 6](#_Toc135052258)

[1.2 Анализ информационных задач и круга пользователей системы…………………………………………………………………………..7](#_Toc135052259)

[Логическое проектирование реляционной БД 7](#_Toc135052260)

[2.1 Составление реляционных отношений 7](#_Toc135052261)

[2.2 Нормализация полученных отношений 12](#_Toc135052262)

[2 Физическое проектирование БД 13](#_Toc135052263)

[3.1 Разработка скриптов на создание базы данных и таблиц 13](#_Toc135052264)

[Разработка скриптов на добавление данных в таблицы 15](#_Toc135052265)

[3.2 Разработка необходимых запросов: 17](#_Toc135052266)

[Список литературы\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_20](#_Toc135052267)

# ЦЕЛЬ КУРСОВОГО ПРОЕКТА

Целью курсового проекта является изучение методов и закрепление знаний в проектировании реляционных баз данных (РБД) в системе управления базами данных (СУБД) Система управления базами данных Postgres.

# ЗАДАНИЕ НА КУРСОВОВОЙ ПРОЕКТ

В данном курсовом проекте ставится задача разработать РБД в СУБД DBeaver по заданной теме «Поликлиника». Проектирование РБД проводится с помощью метода «Сущность-связь». Проверка построенной модели РБД осуществляется с помощью метода нормализации отношений.

Описание предметной области.

База данных создаётся для информационного обслуживания регистрационного отдела поликлиники. БД должна содержать информацию о врачах, ведущих прием, расписании приема, и пациентах, льготах пациентах, кабинете врача.

Готовые запросы:

1. Выдавать сводную информацию обо всех врачах поликлиники.
2. Выдавать сводную информацию о пациентах.
3. Выдавать информацию о записи пациента к врачу.
4. Выдавать информацию о приеме врачей на указанную дату.
5. Выдавать информацию о пациентах, имеющих льготы на приобретение лекарств.

# ПРОЕКТИРОВАНИЕ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ

## Инфологическое проектирование

### Анализ предметной области

База данных создаётся для информационного обслуживания некой поликлиники

Выделим базовые сущности этой предметной области:

* Doctor
* Patient
* Schedule

ER-диаграмма БД, приведена на (Рисунке 1). На данной ER-диаграмме присутствуют связи многие - ко - многим и один - ко - многим.

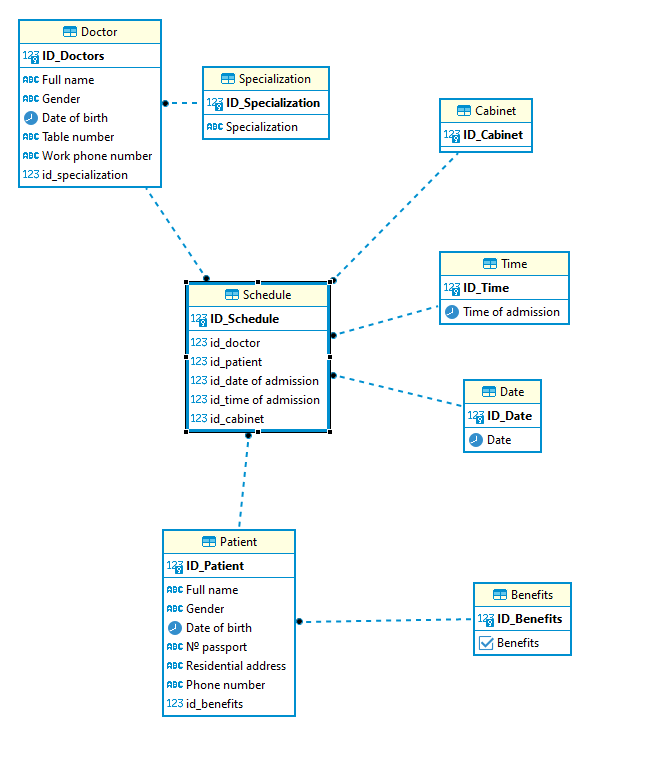


Рисунок 1 – ER-диаграмма

### Анализ информационных задач и круга пользователей системы

Система создаётся для обслуживания следующих групп пользователей:

* Врач
* Пациент

Определим границы информационной поддержки пользователей:

**1) Функциональные возможности:**

* ведение БД (запись, чтение, модификация, удаление);
* обеспечение логической непротиворечивости БД;
* реализация наиболее часто встречающихся запросов в готовом виде.

**2) Готовые запросы:**

* Выдавать сводную информацию обо всех врачах поликлиники.
* Выдавать сводную информацию о пациентах.
* Выдавать информацию о записи пациента к врачу.
* Выдавать информацию о приеме врачей на указанную дату.
* Выдавать информацию о пациентах, имеющих льготы на приобретение лекарств.

## Логическое проектирование реляционной БД

### Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности (объекту ПрО), и в него вносятся все атрибуты сущности. Для каждого отношения необходимо определить первичный ключ и внешние ключи (если они есть). В том случае, если базовое отношение не имеет потенциальных ключей, вводится *суррогатный первичный ключ*, который не несёт смысловой нагрузки и служит только для идентификации записей.

Были созданы следующие сущности:

1. Таблица "Doctor" - содержит информацию о врачах, такую как ID\_Doctors, полное имя, пол, дата рождения, номер кабинета, рабочий телефон и идентификатор специализации.
2. Таблица "Time" - содержит время приема пациентов.
3. Таблица "Specialization" - содержит информацию о специализации врачей.
4. Таблица "Schedule" - содержит расписание приемов пациентов, включая идентификатор врача, пациента, дату и время приема и номер кабинета.
5. Таблица "Patient" - содержит информацию о пациентах, такую как ID\_Patient, полное имя, пол, дата рождения, номер паспорта, адрес проживания, телефон и идентификатор льгот.
6. Таблица "Date" - содержит даты приемов пациентов.
7. Таблица "Cabinet" - содержит номера кабинетов, где проходят приемы пациентов.
8. Таблица "Benefits" - содержит информацию о льготах, доступных пациентам.

**Таблица 1 – Схема отношения** **Доктора** (Doctors)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Длина** | **Примечания** |
| Код доктора | ID\_Doctors | Целое число | - | Первичный ключ, генерируемый автоматически |
| ФИО | Full name | Текст | - | Обязательное поле |
| Пол | Gender | Символ | 3 | Обязательное поле |
| Дата рождения | Date of birth | Дата | - | Обязательное поле |
| Табельный номер | Table number | Символ | 6 | Обязательное поле |
| Рабочий телефон | Work phone number | Символ | 5 | Обязательное поле |
| Код специализации | id\_specialization | Целое число | - | Внешний ключ, ссылается на таблицу "Specialization" |

Таблица 2 – Схема отношения Специализация (Специальность)

|  |  |  |  |
| --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип, длина** | **Примечания** |
| ID специализации | ID\_Specialization | Целое число | первичный ключ, автоинкремент |
| Специализация | Specialization | Текст | не может быть NULL |

Таблица 3 – Схема отношения Пациент (Patient)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Длина** | **Примечания** |
| Код пациента | ID\_Patient | Целое число |  | первичный ключ |
| ФИО | Full name | Текст |  |  |
| Пол | Gender | Текст |  |  |
| Дата рождения | Date of birth | Дата |  | обязательное поле |
| Номер паспорта | № passport | Текст | 10 | уникальное, обязательное поле |
| Адрес проживания | Residential address | Текст | 250 | обязательное поле |
| Номер телефона | Phone number | Текст | 11 | уникальное, обязательное поле, без пробелов и дефисов |
| Код льготы | id\_benefits | Целое число |  | обязательное поле, внешний ключ на таблицу "Льготы" |

Таблица 4 – Схема отношения Льгота (Benefits)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Длина** | **Примечания** |
| Идентификатор | ID\_Benefits | Integer |  | Уникальный идентификатор |
| Льгота | Benefits | Boolean |  | Наличие льгот у пациента |

Таблица 5 – Схема отношения Кабинет (Cabinet)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Длина** | **Примечания** |
| Идентификатор | ID\_Cabinet | SERIAL |  | Первичный ключ, автоинкремент |

Таблица 6 – Схема отношения Дата (Date)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Длина** | **Примечания** |
| Идентификатор | ID\_Date | integer |  | PRIMARY KEY |
| Дата | Date | date |  | Не может быть NULL |

Таблица 7 – Схема отношения Время (Time)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Длина** | **Примечания** |
| Идентификатор | ID\_Time | integer |  | not null, generated always as identity, PRIMARY KEY |
| Время приема | Time of admission | time |  | not null |

Таблица 8– Схема отношения Расписание (Schedule)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Содержание поля** | **Имя поля** | **Тип** | **Длина** | **Примечания** |
| Идентификатор записи | ID\_Schedule | integer |  | Ключевой индекс |
| Идентификатор доктора | id\_doctor | integer |  | Внешний ключ, ссылающийся на таблицу "Doctor" |
| Идентификатор пациента | id\_patient | integer |  | Внешний ключ, ссылающийся на таблицу "Patient" |
| Идентификатор даты приёма | id\_date of admission | integer |  | Внешний ключ, ссылающийся на таблицу "Date" |
| Идентификатор времени приёма | id\_time of admission | integer |  | Внешний ключ, ссылающийся на таблицу "Time" |
| Идентификатор кабинета | id\_cabinet | integer |  | Внешний ключ, ссылающийся на таблицу "Cabinet" |

### Нормализация полученных отношений

**1НФ.** Для приведения таблиц к 1НФ требуется составить таблицы (один атрибут – один столбец) и разбить сложные атрибуты на простые, а многозначные атрибуты вынести в отдельные отношения.  
В наших таблицах все атрибуты простые.

**2НФ.** Все отношения находятся во 2НФ, так как для каждого из них введен уникальный ключ.

**3НФ**. Все неключевые атрибуты зависят только от первичного ключа и не зависят друг от друга. Поэтому спроектированная **база данных находится в третьей нормальной форме**.

## Физическое проектирование БД

### Разработка скриптов на создание базы данных и таблиц

БД «Компания» проектируется в среде Система управления базами данных PostgresSQL.

*Создание базы данных* «Поликлиника»:

USE public;

*Создание таблицы Специализация*

create TABLE "Specialization" (

"ID\_Specialization" integer not null generated always as identity PRIMARY key,

"Specialization" text not null

);

*Создание таблицы Доктор*

*CREATE TABLE "Doctor" (*

*"ID\_Doctors" integer not null generated always as identity PRIMARY KEY,*

*"Full name" text not null,*

*"Gender" char(3) not null,*

*"Date of birth" date not null,*

*"Table number" char(6) not null,*

*"Work phone number" char(5) not null,*

*"id\_specialization" integer not null,*

*FOREIGN KEY (id\_specialization) REFERENCES "Specialization" ("ID\_Specialization")*

*);*

*Создание таблицы Кабинет*

*create TABLE "Cabinet" (*

*"ID\_Cabinet" SERIAL PRIMARY KEY*

*);*

*Создание таблицы Дата*

*CREATE TABLE "Date" (*

*"ID\_Date" integer not null generated always as identity PRIMARY key,*

*"Date" date not null*

*);*

*Создание таблицы Время*

*CREATE TABLE "Time" (*

*"ID\_Time" integer not null generated always as identity PRIMARY key,*

*"Time of admission" time not null*

*);*

*Создание таблицы Льгота*

*CREATE TABLE "Benefits" (*

*"ID\_Benefits" integer not null generated always as identity PRIMARY key,*

*"Benefits" boolean*

*);*

*Создание таблицы Пациент*

*CREATE TABLE "Patient" (*

*"ID\_Patient" integer not null generated always as identity PRIMARY KEY,*

*"Full name" text,*

*"Gender" text,*

*"Date of birth" date not null,*

*"№ passport" char(10) not null,*

*"Residential address" varchar(250) not null,*

*"Phone number" char(11) not null,*

*"id\_benefits" integer not null,*

*FOREIGN KEY (id\_benefits) REFERENCES "Benefits" ("ID\_Benefits")*

*Создание таблицы Расписание*

*CREATE TABLE "Schedule" (*

*"ID\_Schedule" integer not null generated always as identity PRIMARY key,*

*"id\_doctor" integer not null,*

*"id\_patient" integer not null,*

*"id\_date of admission" integer not null,*

*"id\_time of admission" integer not null,*

*"id\_cabinet" integer not null,*

*FOREIGN KEY (id\_doctor) REFERENCES "Doctor" ("ID\_Doctors"),*

*FOREIGN KEY (id\_patient) REFERENCES "Patient" ("ID\_Patient"),*

*FOREIGN KEY ("id\_date of admission") REFERENCES "Date" ("ID\_Date"),*

*FOREIGN KEY ("id\_time of admission") REFERENCES "Time" ("ID\_Time"),*

*FOREIGN KEY (id\_cabinet) REFERENCES "Cabinet" ("ID\_Cabinet")*

*);*

### Разработка скриптов на добавление данных в таблицы

*Заполнение таблицы* *Специализация:*

insert into "Specialization" ("Specialization") values

('Терапевт'),

('Офтальмолог'),

('Отоларинголог'),

('Гинеколог'),

('Стоматолог'),

('Врач общей практики'),

('Хирург'),

('Гастроэнтеролог'),

('Проктолог'),

('Уролог')

*Заполнение таблицы* *Доктор:*

*insert into "Doctor" ("Full name", "id\_specialization", "Gender", "Date of birth", "Table number", "Work phone number") values*

*('Петраков Андрей Васильевич',1, 'МУЖ', '12.01.1987', 152041, 24325),*

*('Никифоров Никита Евгеньевич',2, 'МУЖ', '22.11.1958', 324512, 87657),*

*('Киракосян Анна Феликсовна',3, 'ЖЕН', '14.09.1974', 191631, 32567),*

*Заполнение таблицы* *Кабинет:*

*INSERT into "Cabinet" VALUES*

*(101),*

*(102),*

*(103),*

*(104),*

*(105),*

*Заполнение таблицы* *Дата:*

*insert into "Date" ("Date") values*

*('05.05.2023'),*

*('06.05.2023'),*

*('07.05.2023'),*

*('08.05.2023'),*

*('09.05.2023'),*

*Заполнение таблицы* *Время:*

*insert into "Time" ("Time of admission") values*

*('07:00'),*

*('07:15'),*

*('07:30'),*

*('07:45'),*

*('08:00'),*

*Заполнение таблицы* *Льгота:*

*insert into "Benefits" ("Benefits") values*

*('No'),*

*('Yes')*

*Заполнение таблицы* *Пациент:*

*insert into "Patient" ("Full name", "Gender", "Date of birth", "№ passport" , "Residential address","Phone number",id\_benefits) values*

*('Смирнова Елена Петровна', 'ЖЕН', '12.10.1985', 5314682487, 'г. Москва, ул. Ленина, дом 25, кв. 18', 89991234567, 1),*

*('Иванов Игорь Владимирович', 'МУЖ', '23.05.1978', 1234567890, 'г. Москва, ул. Красная, дом 5, кв. 47', 89054561234, 1),*

*('Петров Петр Петрович', 'МУЖ', '03.12.1990', 9876543210, 'г. Москва, ул. Гагарина, дом 12, кв. 56', 89101122334, 2),*

*Заполнение таблицы* *Расписание:*

*insert into "Schedule" ("id\_doctor", "id\_patient", "id\_date of admission", "id\_time of admission", "id\_cabinet") values*

*(1, 2, 1, 1, 101),*

*(3, 4, 2, 4, 205),*

*(5, 6, 3, 7, 315),*

*(2, 3, 4, 13, 110),*

### Разработка необходимых запросов:

1. Выдавать сводную информацию обо всех врачах поликлиники.

*SELECT "Doctor"."ID\_Doctors", "Doctor"."Full name", "Specialization"."Specialization"*

*FROM "Doctor"*

*JOIN "Specialization" ON "Doctor"."id\_specialization" = "Specialization"."ID\_Specialization";*

2. Выдавать сводную информацию о пациентах.

SELECT \*

FROM "Patient" AS a;

3. Выдавать информацию о записи пациента к врачу.

*SELECT*

*"Schedule"."ID\_Schedule",*

*"Doctor"."Full name" as "Doctor name",*

*"Patient"."Full name" as "Patient name",*

*"Date"."Date",*

*"Time"."Time of admission",*

*"Cabinet"*

*FROM*

*"Schedule"*

*JOIN "Doctor" ON "Schedule"."id\_doctor" = "Doctor"."ID\_Doctors"*

*JOIN "Patient" ON "Schedule"."id\_patient" = "Patient"."ID\_Patient"*

*JOIN "Date" ON "Schedule"."id\_date of admission" = "Date"."ID\_Date"*

*JOIN "Time" ON "Schedule"."id\_time of admission" = "Time"."ID\_Time"*

*JOIN "Cabinet" ON "Schedule"."id\_cabinet" = "Cabinet"."ID\_Cabinet"*

4. Выдавать информацию о приеме врачей на указанную дату.

*SELECT "Doctor"."Full name" AS "Doctor's name", "Patient"."Full name" AS "Patient's name",*

*"Date"."Date" AS "Date of admission", "Time"."Time of admission" AS "Time of admission",*

*"Cabinet" AS "Number cabinet"*

*FROM "Schedule"*

*JOIN "Doctor" ON "Doctor"."ID\_Doctors" = "Schedule"."id\_doctor"*

*JOIN "Patient" ON "Patient"."ID\_Patient" = "Schedule"."id\_patient"*

*JOIN "Date" ON "Date"."ID\_Date" = "Schedule"."id\_date of admission"*

*JOIN "Time" ON "Time"."ID\_Time" = "Schedule"."id\_time of admission"*

*JOIN "Cabinet" ON "Cabinet"."ID\_Cabinet" = "Schedule"."id\_cabinet"*

*WHERE "Date"."Date" = '2023-05-08';*

1. Выдавать информацию о пациентах, имеющих льготы на приобретение лекарств.

*SELECT \**

*FROM "Patient"*

*JOIN "Benefits" ON "Patient"."id\_benefits" = "Benefits"."ID\_Benefits"*

*WHERE "Benefits"."Benefits" = true;*

# Список литературы

**Visual Paradigm** Entity Relation [Online] // Visual Paradigm Online. - @2020 by Visual Paradigm, 2020. - 2 2, 2020. - https://online.visual-paradigm.com/app/diagrams/#diagram:proj=0&type=ERDiagram.

ГОСТ 7.32-2017. Межгосударственный стандарт. Система стандартов по информации, библиотечному и издательскому делу. Отчет о научно-исследовательской работе. Структура и правила оформления [В Интернете] // Официальный сайт компании «КонсультантПлюс». - 24 Март 2020 г.. - http://www.consultant.ru/document/cons\_doc\_LAW\_292293/.