Санкт-Петербургский политехнический университет Петра Великого

Институт компьютерных наук и технологий

**Высшая школа интеллектуальных систем и суперкомпьютерных**

**технологий**

**Практическая работа**

**АВТОМАТИЗИРВАННОЕ ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

**ПОЛИКЛИНИКИ**

по дисциплине «Базы данных»

Выполнил

студент гр.3530203/10001 А. В. Банникова

<подпись>

Руководитель

доцент, к.т.н. Е. Е. Андрианова

<подпись>

Санкт-Петербург

2023

**Цель**:

научиться проектировать базы данных, поработать с программами

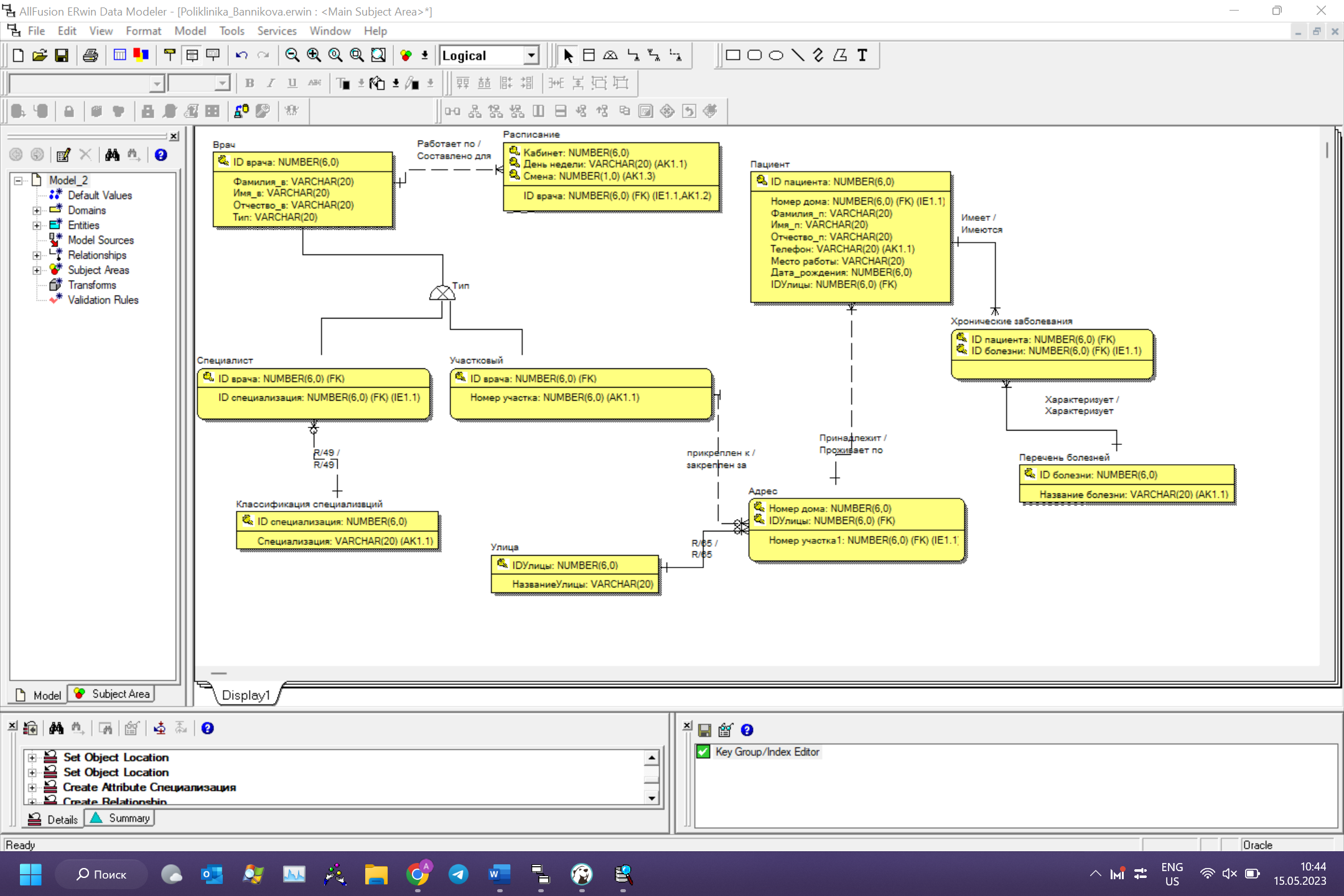
AllFusion Erwin Data Modeler и CA AllFusion Data Modeler Validator.

**Задачи:**

1. Создать логическую модель базы данных поликлиники.
2. Создать физическую модель базы данных. Отобразить ключевые поля.
3. Создать представления для отображения
4. Проверить разработанную модель средствами Validator.
5. Устранить ошибки модели, которые выявил Validator.
6. Провести нормализацию таблиц до пятой нормальной формы. Внести
7. коррекцию в модель.
8. Провести прямое проектирование – создать объекты базы данных в
9. Oracle (PostgreSQL).
10. Проверить работоспособность базы данных в Oracle (PostgreSQL)
11. Провести обратное проектирование базы данных из Oracle (PostgreSQL).

**Ход работы:**

1. Логическая модель в Erwin



Физическая модель в Erwin

Изображение выглядит как диаграмма

Автоматически созданное описание

1. Обоснование построения логической модели

Сущности, описанные в задании: **Участковый врач**, **Специалист**, **Врач**, **Адрес**, **Специализация**, **Расписание**, **Пациент**, **Перечень болезней**

Так как Пациент и Перечень болезней имеют связь М:М, добавим сущность **Хронические заболевания**(ассоциативная таблица)

Атрибуты Участкового врача: ID врача, номер участка

Атрибуты Специалиста: ID врача, ID специалиста

Атрибуты Врача: ID врача, Фамилия\_в, Имя\_в, Отчество\_в, Тип

Атрибуты Адреса: Номер дома, улица, номер участка

Атрибуты Специализации: ID специализация, специализация

Атрибуты Расписания: кабинет, день недели, смена, ID врача

Атрибуты Пациента: ID пациента, номер дома, улица, фамилия\_п, имя\_п, отчество\_п, телефон, место работы, дата рождения

Атрибуты перечня болезней: ID пациента, ID болезни

Логическая модель удовлетворяет всем требованиям:

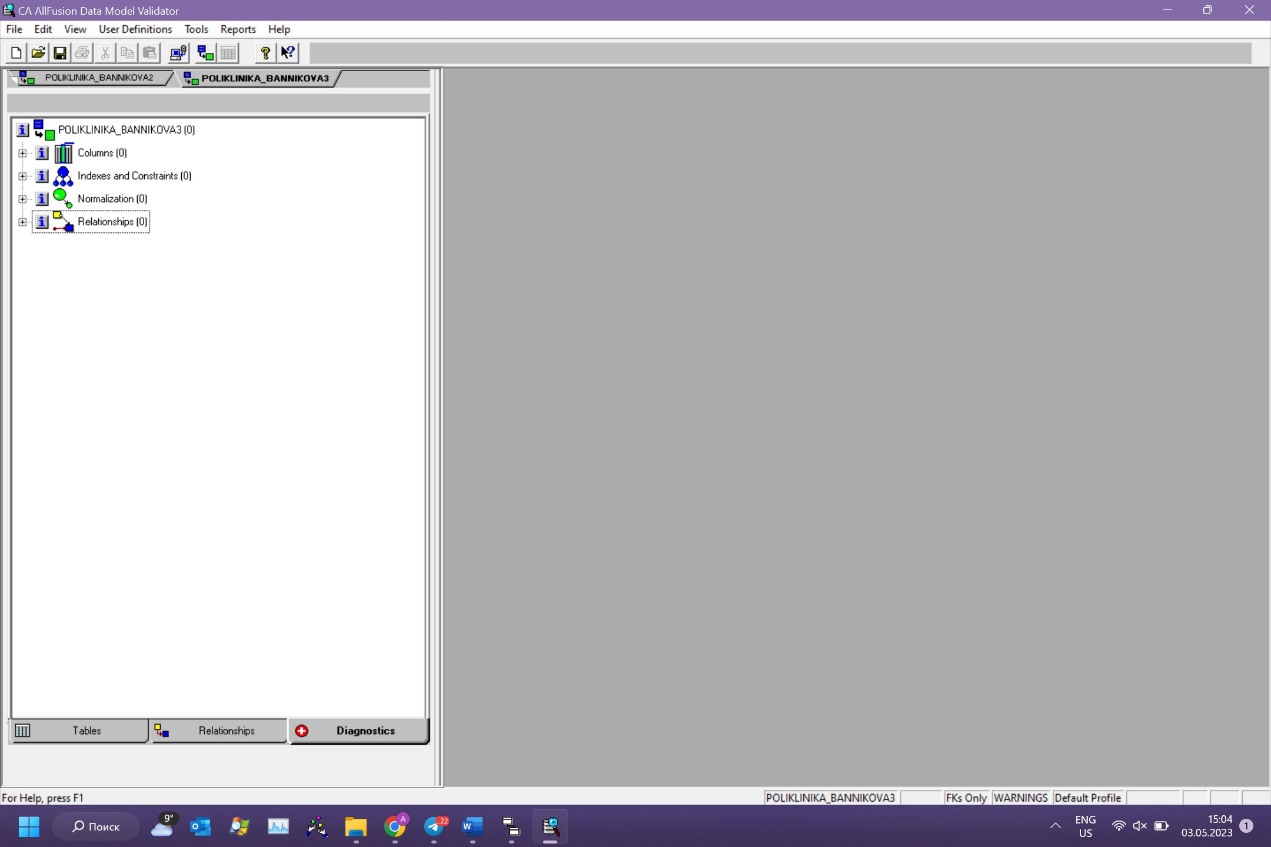
* В поликлинике работают участковые врачи и специалисты.
* Каждый врач может быть только специалистом или только участковым врачом.
* Каждый участковый врач обслуживает только один участок.
* Один участок обслуживает только один участковый врач.
* Каждый врач-специалист имеет определенную специализацию – хирург, ЛОР и т.д.
* Участок характеризуется перечнем улиц с указанием домов, которые соответствуют данному участку.
* Расписания врачей стабильны и заданы на неделю.
* В расписании указаны часы приема (смена), кабинет и день недели.
* В одном кабинете могут работать разные врачи, но только в разное время (смену).
* При записи к врачу пациент должен указать свой адрес – улицу и дом, по адресу определяется его участок.
* При записи к врачу пациент указывает также фамилию, имя, отчество, дату рождения, телефон, место работы, перечень хронических заболеваний – гепатит, например.

1. Результаты проверки модели на Validator – до устранения ошибок

* Изображение выглядит как текст, программное обеспечение, компьютер, Мультимедийное программное обеспечение

  Автоматически созданное описание

1. Результаты проверки модели на Validator – после устранения ошибок.



1. Прямое проектирование в Oracle (генерация SQL кода)

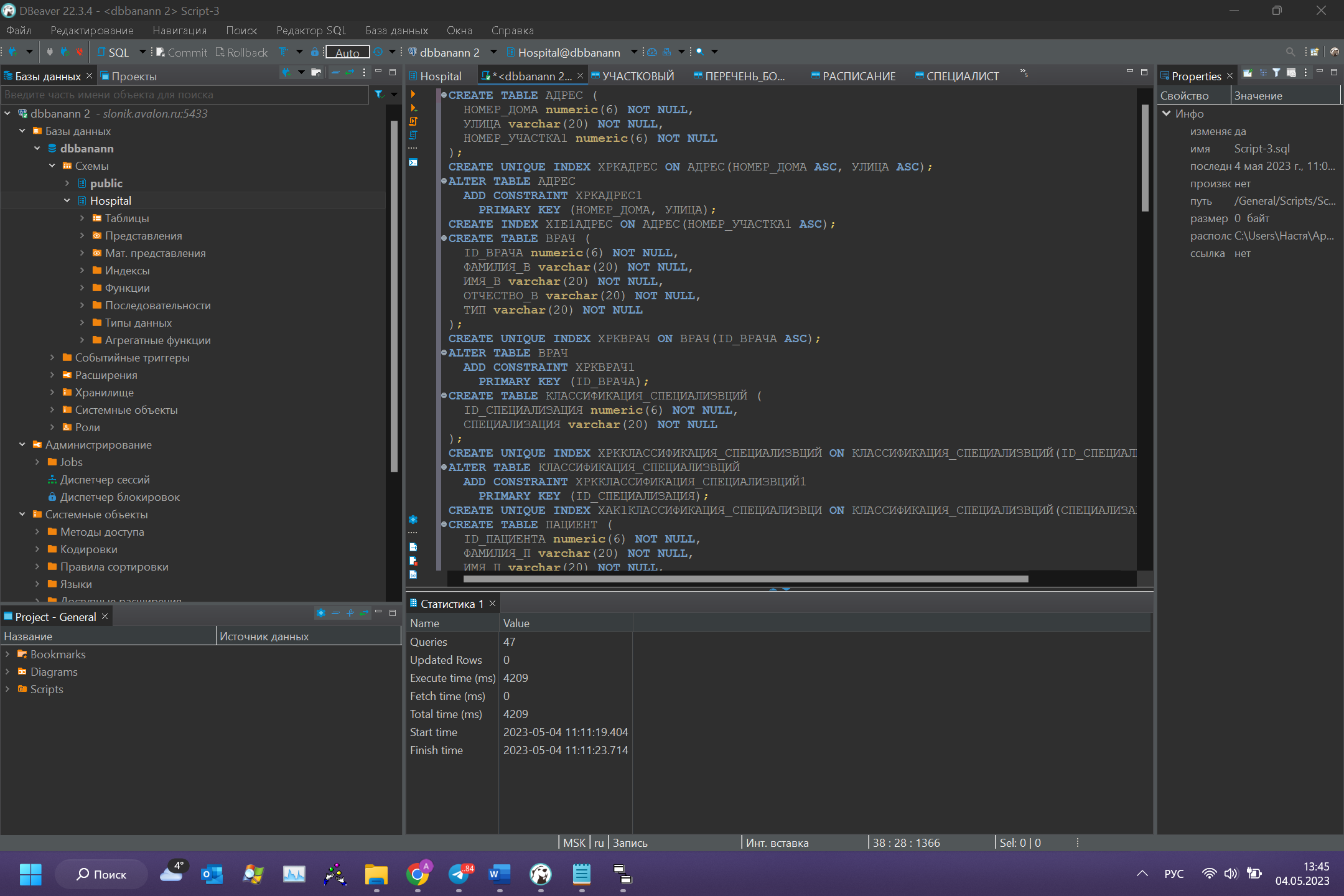
Изображение выглядит как текст

Автоматически созданное описание

1. Перенос кода SQL Oracle в SQL PostgreSQL с помощью JOOQ (в отчете приложить коды для двух СУБД)

Изображение выглядит как текст, в помещении, снимок экрана, монитор

Автоматически созданное описание

1. Результаты проверки работоспособности базы данных 
2. Результаты обратного проектирования (модель в DBeaver)

Диаграмма

Изображение выглядит как текст, монитор, снимок экрана, в помещении

Автоматически созданное описание

Схема Hospital

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Адрес

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Врач

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Классификация специализаций

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Пациент

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Перечень болезней

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Расписание

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Специалист

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Участковый

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Хронические заболевания

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

1) Нельзя добавить двух врачей в одно время, один кабинет и один день недели

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

2) Нельзя добавить двух врачей на один участок

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

1. Отчет о проведении нормализации

1) R1 **Врач**

Отношение находится в 1НФ, т.к. все значения атомарны, составных атрибутов

нет, порядок строк и столбцов не несет в себе никакой информации, все

столбцы имеют уникальные имена, отсутствуют повторяющиеся строки.

Потенциальный (и первичный) ключ в отношении: {ID врача}.

Неключевые атрибуты: {Фамилия\_в},{Имя\_в},{Отчество\_в},{Тип}

Отношение R1 находится в 2НФ, так как неключевые атрибуты не зависят от части ключа.

Отношение R1 находится в 3НФ, так как не содержит транзитивных зависимостей.

Отношение R1 находится в НФБК, так как в нём нет пересекающихся потенциальных ключей.

Отношение R1 находится в 4НФ, так как не содержит многозначных зависимостей.

Отношение R1 находится в 5НФ, так как не содержит зависимостей соединения.

2) R2 **Участковый**

Отношение находится в 1НФ, т.к. все значения атомарны, составных атрибутов

нет, порядок строк и столбцов не несет в себе никакой информации, все

столбцы имеют уникальные имена, отсутствуют повторяющиеся строки.

Потенциальный (и первичный) ключ в отношении: {ID врача}.

Неключевые атрибуты: {Номер участка}

Отношение R2 находится в 2НФ, так как неключевой атрибут находится в полной функциональной зависимости от ключа.

Отношение R2 находится в 3НФ, так как неключевой атрибут один.

Отношение R2 находится в НФБК, так как в нём нет пересекающихся потенциальных ключей.

Отношение R2 находится в 4НФ, так как оно состоит из менее трех атрибутов.

Отношение R2 находится в 5НФ, так как оно состоит из менее трех атрибутов.

3) R3 **Специалист**

Отношение находится в 1НФ, т.к. все значения атомарны, составных атрибутов

нет, порядок строк и столбцов не несет в себе никакой информации, все

столбцы имеют уникальные имена, отсутствуют повторяющиеся строки.

Потенциальный (и первичный) ключ в отношении: {ID Врача}.

Неключевые атрибуты: { ID Специализации}

Отношение R3 находится в 2НФ, так как неключевой атрибут находится в полной функциональной зависимости от ключа.

Отношение R3 находится в 3НФ, так как неключевой атрибут один.

Отношение R3 находится в НФБК, так как в нём нет пересекающихся потенциальных ключей.

Отношение R3 находится в 4НФ, так как оно состоит из менее трех атрибутов.

Отношение R3 находится в 5НФ, так как оно состоит из менее трех атрибутов.

4) R4 **Классификация специализаций**

Отношение находится в 1НФ, т.к. все значения атомарны, составных атрибутов

нет, порядок строк и столбцов не несет в себе никакой информации, все

столбцы имеют уникальные имена, отсутствуют повторяющиеся строки.

Потенциальный (и первичный) ключ в отношении: {ID Специализации}.

Альтернативный ключ в отношении: {Название Специализации}

Неключевые атрибуты: {Название Специализации}

Отношение R4 находится в 2НФ, так как неключевой атрибут находится в полной функциональной зависимости от ключа.

Отношение R4 находится в 3НФ, так как неключевой атрибут один.

Отношение R4 находится в НФБК, так как в нём нет пересекающихся потенциальных ключей.

Отношение R4 находится в 4НФ, так как оно состоит из менее трех атрибутов.

Отношение R4 находится в 5НФ, так как оно состоит из менее трех атрибутов.

5) R5 **Расписание**

Отношение находится в 1НФ, т.к. все значения атомарны, составных атрибутов

нет, порядок строк и столбцов не несет в себе никакой информации, все

столбцы имеют уникальные имена, отсутствуют повторяющиеся строки.

Первичный ключ в отношении: {Кабинет, День недели, Смена}.

Альтернативный ключ в отношении: {День недели, Смена, ID врача}

Неключевые атрибуты: {ID врача}

Отношение R5 находится в 2НФ, так как неключевые атрибуты не зависят от части ключа.

Отношение R5 находится в 3НФ, так как не содержит транзитивных зависимостей.

Отношение R5 находится в НФБК, так как в нём нет пересекающихся потенциальных ключей.

Отношение R5 находится в 4НФ, так как не содержит многозначных зависимостей.

Отношение R5 находится в 5НФ, так как не содержит зависимостей соединения.

6)R6 **Адрес**

Отношение находится в 1НФ, т.к. все значения атомарны, составных атрибутов

нет, порядок строк и столбцов не несет в себе никакой информации, все

столбцы имеют уникальные имена, отсутствуют повторяющиеся строки.

Первичный ключ в отношении: {Номер дома, Улица}.

Неключевые атрибуты: {Номер участка}

Отношение R6 находится в 2НФ, так как неключевые атрибуты не зависят от части ключа.

Отношение R6 находится в 3НФ, так как не содержит транзитивных зависимостей.

Отношение R6 находится в НФБК, так как в нём нет пересекающихся потенциальных ключей.

Отношение R6 находится в 4НФ, так как не содержит многозначных зависимостей.

Отношение R6 находится в 5НФ, так как не содержит зависимостей соединения.

7) R7 **Пациент**

Отношение находится в 1НФ, т.к. все значения атомарны, составных атрибутов

нет, порядок строк и столбцов не несет в себе никакой информации, все

столбцы имеют уникальные имена, отсутствуют повторяющиеся строки.

Первичный ключ в отношении: {ID пациента}.

Альтернативный ключ в отношении: {Телефон}

Неключевые атрибуты: {Фамилия\_п}, {Имя\_п},{Отчество\_п},{Дата рождения},{Место работы},{Телефон},{Улица},{Номер дома}

Отношение R7 находится в 2НФ, так как неключевые атрибуты не зависят от части ключа.

Отношение R7 находится в 3НФ, так как не содержит транзитивных зависимостей.

Отношение R7 находится в НФБК, так как в нём нет пересекающихся потенциальных ключей.

Отношение R7 находится в 4НФ, так как не содержит многозначных зависимостей.

Отношение R7 находится в 5НФ, так как не содержит зависимостей соединения.

8)R8 **Перечень болезней**

Отношение находится в 1НФ, т.к. все значения атомарны, составных атрибутов

нет, порядок строк и столбцов не несет в себе никакой информации, все

столбцы имеют уникальные имена, отсутствуют повторяющиеся строки.

Потенциальный (и первичный) ключ в отношении: {ID болезни }.

Альтернативный ключ в отношении: {Название Болезни}

Неключевые атрибуты: {Название Болезни}

Отношение R8 находится в 2НФ, так как неключевой атрибут находится в полной функциональной зависимости от ключа.

Отношение R8 находится в 3НФ, так как неключевой атрибут один.

Отношение R8 находится в НФБК, так как в нём нет пересекающихся потенциальных ключей.

Отношение R8 находится в 4НФ, так как оно состоит из менее трех атрибутов.

Отношение R8 находится в 5НФ, так как оно состоит из менее трех атрибутов.

9) R9 **Хронические заболевания**

Отношение находится в 1НФ, т.к. все значения атомарны, составных атрибутов

нет, порядок строк и столбцов не несет в себе никакой информации, все

столбцы имеют уникальные имена, отсутствуют повторяющиеся строки.

Потенциальный (и первичный) ключ в отношении: {ID Пациента, ID болезни}.

Неключевые атрибуты: нет

Отношение R9 находится в 2НФ, так как неключевой атрибутов нет

Отношение R9 находится в 3НФ, так как неключевой атрибутов нет

Отношение R9 находится в НФБК, так как в нём нет пересекающихся потенциальных ключей.

Отношение R9 находится в 4НФ, так как оно состоит из менее трех атрибутов.

Отношение R9 находится в 5НФ, так как оно состоит из менее трех атрибутов.

**Приложение:**

**SQL код для Oracle:**

CREATE TABLE Адрес

(

Номер\_дома NUMBER(6,0) NOT NULL ,

Улица VARCHAR2(20) NOT NULL ,

Номер\_участка1 NUMBER(6,0) NOT NULL

);

CREATE UNIQUE INDEX XPKАдрес ON Адрес

(

Номер\_дома ASC,

Улица ASC

);

ALTER TABLE Адрес

ADD CONSTRAINT XPKАдрес PRIMARY KEY (Номер\_дома,Улица);

CREATE INDEX XIE1Адрес ON Адрес

(

Номер\_участка1 ASC

);

CREATE TABLE Врач

(

ID\_врача NUMBER(6,0) NOT NULL ,

Фамилия\_в VARCHAR2(20) NOT NULL ,

Имя\_в VARCHAR2(20) NOT NULL ,

Отчество\_в VARCHAR2(20) NOT NULL ,

Тип VARCHAR2(20) NOT NULL

);

CREATE UNIQUE INDEX XPKВрач ON Врач

(

ID\_врача ASC

);

ALTER TABLE Врач

ADD CONSTRAINT XPKВрач PRIMARY KEY (ID\_врача);

CREATE TABLE Классификация\_специализвций

(

ID\_специализация NUMBER(6,0) NOT NULL ,

Специализация VARCHAR2(20) NOT NULL

);

CREATE UNIQUE INDEX XPKКлассификация\_специализвций ON Классификация\_специализвций

(

ID\_специализация ASC

);

ALTER TABLE Классификация\_специализвций

ADD CONSTRAINT XPKКлассификация\_специализвций PRIMARY KEY (ID\_специализация);

CREATE UNIQUE INDEX XAK1Классификация\_специализвци ON Классификация\_специализвций

(

Специализация ASC

);

CREATE TABLE Пациент

(

ID\_пациента NUMBER(6,0) NOT NULL ,

Фамилия\_п VARCHAR2(20) NOT NULL ,

Имя\_п VARCHAR2(20) NOT NULL ,

Отчество\_п VARCHAR2(20) NULL ,

Телефон VARCHAR2(20) NOT NULL ,

Место\_работы VARCHAR2(20) NULL ,

Дата\_рождения NUMBER(6,0) NULL ,

Номер\_дома NUMBER(6,0) NOT NULL ,

Улица VARCHAR2(20) NOT NULL

);

CREATE UNIQUE INDEX XPKПациент ON Пациент

(

ID\_пациента ASC

);

ALTER TABLE Пациент

ADD CONSTRAINT XPKПациент PRIMARY KEY (ID\_пациента);

CREATE UNIQUE INDEX XAK1Пациент ON Пациент

(

Телефон ASC

);

CREATE INDEX XIE1Пациент ON Пациент

(

Номер\_дома ASC,

Улица ASC

);

CREATE TABLE Перечень\_болезней

(

ID\_болезни NUMBER(6,0) NOT NULL ,

Название\_болезни VARCHAR2(20) NOT NULL

);

CREATE UNIQUE INDEX XPKПеречень\_болезней ON Перечень\_болезней

(

ID\_болезни ASC

);

ALTER TABLE Перечень\_болезней

ADD CONSTRAINT XPKПеречень\_болезней PRIMARY KEY (ID\_болезни);

CREATE UNIQUE INDEX XAK1Перечень\_болезней ON Перечень\_болезней

(

Название\_болезни ASC

);

CREATE TABLE Расписание

(

Кабинет NUMBER(6,0) NOT NULL ,

День\_недели VARCHAR2(20) NOT NULL ,

Смена NUMBER(1,0) NOT NULL ,

ID\_врача NUMBER(6,0) NOT NULL

);

CREATE UNIQUE INDEX XPKРасписание ON Расписание

(

Кабинет ASC,

День\_недели ASC,

Смена ASC

);

ALTER TABLE Расписание

ADD CONSTRAINT XPKРасписание PRIMARY KEY (Кабинет,День\_недели,Смена);

CREATE UNIQUE INDEX XAK1Расписание ON Расписание

(

День\_недели ASC,

ID\_врача ASC,

Смена ASC

);

CREATE INDEX XIE1Расписание ON Расписание

(

ID\_врача ASC

);

CREATE TABLE Специалист

(

ID\_врача NUMBER(6,0) NOT NULL ,

ID\_специализация NUMBER(6,0) NOT NULL

);

CREATE UNIQUE INDEX XPKСпециалист ON Специалист

(

ID\_врача ASC

);

ALTER TABLE Специалист

ADD CONSTRAINT XPKСпециалист PRIMARY KEY (ID\_врача);

CREATE INDEX XIE1Специалист ON Специалист

(

ID\_специализация ASC

);

CREATE TABLE Участковый

(

ID\_врача NUMBER(6,0) NOT NULL ,

Номер\_участка NUMBER(6,0) NOT NULL

);

CREATE UNIQUE INDEX XPKУчастковый ON Участковый

(

ID\_врача ASC

);

ALTER TABLE Участковый

ADD CONSTRAINT XPKУчастковый PRIMARY KEY (ID\_врача);

CREATE UNIQUE INDEX XAK1Участковый ON Участковый

(

Номер\_участка ASC

);

CREATE TABLE Хронические\_заболевания

(

ID\_пациента NUMBER(6,0) NOT NULL ,

ID\_болезни NUMBER(6,0) NOT NULL

);

CREATE UNIQUE INDEX XPKХронические\_заболевания ON Хронические\_заболевания

(

ID\_пациента ASC,

ID\_болезни ASC

);

ALTER TABLE Хронические\_заболевания

ADD CONSTRAINT XPKХронические\_заболевания PRIMARY KEY (ID\_пациента,ID\_болезни);

CREATE INDEX XIE1Хронические\_заболевания ON Хронические\_заболевания

(

ID\_болезни ASC

);

CREATE VIEW ПациентУчасковый

AS SELECT Пациент.Фамилия\_п,Врач.Фамилия\_в

FROM Пациент,Участковый,Врач,Адрес

WHERE Пациент.ID\_улицы = Адрес.ID\_улицы AND Пациент.Номер\_дома = Адрес.Номер\_дома AND Адрес.Номер\_участка = Участковый.Номер\_участка AND Участковый.ID\_врача = Врач.ID\_врача;

CREATE VIEW СпециалистСпециализация

AS SELECT Врач.Фамилия\_в

FROM Специалист,Врач

WHERE Специалист.ID\_врача = Врач.ID\_врача ;

ALTER TABLE Адрес

ADD (CONSTRAINT прикреплен\_к FOREIGN KEY (Номер\_участка1) REFERENCES Участковый(Номер\_участка) ON DELETE SET NULL);

ALTER TABLE Пациент

ADD (CONSTRAINT Принадлежит FOREIGN KEY (Номер\_дома,Улица) REFERENCES Адрес(Номер\_дома,Улица) ON DELETE SET NULL);

ALTER TABLE Расписание

ADD (CONSTRAINT Работает\_по FOREIGN KEY (ID\_врача) REFERENCES Врач(ID\_врача));

ALTER TABLE Специалист

ADD (CONSTRAINT Is\_a\_Врач FOREIGN KEY (ID\_врача) REFERENCES Врач(ID\_врача) ON DELETE CASCADE);

ALTER TABLE Специалист

ADD (CONSTRAINT R\_49 FOREIGN KEY (ID\_специализация) REFERENCES Классификация\_специализвций(ID\_специализация));

ALTER TABLE Участковый

ADD (CONSTRAINT Is\_a\_Врач2 FOREIGN KEY (ID\_врача) REFERENCES Врач(ID\_врача) ON DELETE CASCADE);

ALTER TABLE Хронические\_заболевания

ADD (CONSTRAINT Имеет FOREIGN KEY (ID\_пациента) REFERENCES Пациент(ID\_пациента));

ALTER TABLE Хронические\_заболевания

ADD (CONSTRAINT Характеризует FOREIGN KEY (ID\_болезни) REFERENCES Перечень\_болезней(ID\_болезни));

**SQL код для PostgreSQL:**

CREATE TABLE АДРЕС (

НОМЕР\_ДОМА numeric(6) NOT NULL,

УЛИЦА varchar(20) NOT NULL,

НОМЕР\_УЧАСТКА1 numeric(6) NOT NULL

);

CREATE UNIQUE INDEX XPKАДРЕС ON АДРЕС(НОМЕР\_ДОМА ASC, УЛИЦА ASC);

ALTER TABLE АДРЕС

ADD CONSTRAINT XPKАДРЕС

PRIMARY KEY (НОМЕР\_ДОМА, УЛИЦА);

CREATE INDEX XIE1АДРЕС ON АДРЕС(НОМЕР\_УЧАСТКА1 ASC);

CREATE TABLE ВРАЧ (

ID\_ВРАЧА numeric(6) NOT NULL,

ФАМИЛИЯ\_В varchar(20) NOT NULL,

ИМЯ\_В varchar(20) NOT NULL,

ОТЧЕСТВО\_В varchar(20) NOT NULL,

ТИП varchar(20) NOT NULL

);

CREATE UNIQUE INDEX XPKВРАЧ ON ВРАЧ(ID\_ВРАЧА ASC);

ALTER TABLE ВРАЧ

ADD CONSTRAINT XPKВРАЧ

PRIMARY KEY (ID\_ВРАЧА);

CREATE TABLE КЛАССИФИКАЦИЯ\_СПЕЦИАЛИЗВЦИЙ (

ID\_СПЕЦИАЛИЗАЦИЯ numeric(6) NOT NULL,

СПЕЦИАЛИЗАЦИЯ varchar(20) NOT NULL

);

CREATE UNIQUE INDEX XPKКЛАССИФИКАЦИЯ\_СПЕЦИАЛИЗВЦИЙ ON КЛАССИФИКАЦИЯ\_СПЕЦИАЛИЗВЦИЙ(ID\_СПЕЦИАЛИЗАЦИЯ ASC);

ALTER TABLE КЛАССИФИКАЦИЯ\_СПЕЦИАЛИЗВЦИЙ

ADD CONSTRAINT XPKКЛАССИФИКАЦИЯ\_СПЕЦИАЛИЗВЦИЙ

PRIMARY KEY (ID\_СПЕЦИАЛИЗАЦИЯ);

CREATE UNIQUE INDEX XAK1КЛАССИФИКАЦИЯ\_СПЕЦИАЛИЗВЦИ ON КЛАССИФИКАЦИЯ\_СПЕЦИАЛИЗВЦИЙ(СПЕЦИАЛИЗАЦИЯ ASC);

CREATE TABLE ПАЦИЕНТ (

ID\_ПАЦИЕНТА numeric(6) NOT NULL,

ФАМИЛИЯ\_П varchar(20) NOT NULL,

ИМЯ\_П varchar(20) NOT NULL,

ОТЧЕСТВО\_П varchar(20) NULL,

ТЕЛЕФОН varchar(20) NOT NULL,

МЕСТО\_РАБОТЫ varchar(20) NULL,

ДАТА\_РОЖДЕНИЯ numeric(6) NULL,

НОМЕР\_ДОМА numeric(6) NOT NULL,

УЛИЦА varchar(20) NOT NULL

);

CREATE UNIQUE INDEX XPKПАЦИЕНТ ON ПАЦИЕНТ(ID\_ПАЦИЕНТА ASC);

ALTER TABLE ПАЦИЕНТ

ADD CONSTRAINT XPKПАЦИЕНТ

PRIMARY KEY (ID\_ПАЦИЕНТА);

CREATE UNIQUE INDEX XAK1ПАЦИЕНТ ON ПАЦИЕНТ(ТЕЛЕФОН ASC);

CREATE INDEX XIE1ПАЦИЕНТ ON ПАЦИЕНТ(НОМЕР\_ДОМА ASC, УЛИЦА ASC);

CREATE TABLE ПЕРЕЧЕНЬ\_БОЛЕЗНЕЙ (

ID\_БОЛЕЗНИ numeric(6) NOT NULL,

НАЗВАНИЕ\_БОЛЕЗНИ varchar(20) NOT NULL

);

CREATE UNIQUE INDEX XPKПЕРЕЧЕНЬ\_БОЛЕЗНЕЙ ON ПЕРЕЧЕНЬ\_БОЛЕЗНЕЙ(ID\_БОЛЕЗНИ ASC);

ALTER TABLE ПЕРЕЧЕНЬ\_БОЛЕЗНЕЙ

ADD CONSTRAINT XPKПЕРЕЧЕНЬ\_БОЛЕЗНЕЙ

PRIMARY KEY (ID\_БОЛЕЗНИ);

CREATE UNIQUE INDEX XAK1ПЕРЕЧЕНЬ\_БОЛЕЗНЕЙ ON ПЕРЕЧЕНЬ\_БОЛЕЗНЕЙ(НАЗВАНИЕ\_БОЛЕЗНИ ASC);

CREATE TABLE РАСПИСАНИЕ (

КАБИНЕТ numeric(6) NOT NULL,

ДЕНЬ\_НЕДЕЛИ varchar(20) NOT NULL,

СМЕНА numeric(1) NOT NULL,

ID\_ВРАЧА numeric(6) NOT NULL

);

CREATE UNIQUE INDEX XPKРАСПИСАНИЕ ON РАСПИСАНИЕ(КАБИНЕТ ASC, ДЕНЬ\_НЕДЕЛИ ASC, СМЕНА ASC);

ALTER TABLE РАСПИСАНИЕ

ADD CONSTRAINT XPKРАСПИСАНИЕ

PRIMARY KEY (КАБИНЕТ, ДЕНЬ\_НЕДЕЛИ, СМЕНА);

CREATE UNIQUE INDEX XAK1РАСПИСАНИЕ ON РАСПИСАНИЕ(ДЕНЬ\_НЕДЕЛИ ASC, ID\_ВРАЧА ASC, СМЕНА ASC);

CREATE INDEX XIE1РАСПИСАНИЕ ON РАСПИСАНИЕ(ID\_ВРАЧА ASC);

CREATE TABLE СПЕЦИАЛИСТ (

ID\_ВРАЧА numeric(6) NOT NULL,

ID\_СПЕЦИАЛИЗАЦИЯ numeric(6) NOT NULL

);

CREATE UNIQUE INDEX XPKСПЕЦИАЛИСТ ON СПЕЦИАЛИСТ(ID\_ВРАЧА ASC);

ALTER TABLE СПЕЦИАЛИСТ

ADD CONSTRAINT XPKСПЕЦИАЛИСТ

PRIMARY KEY (ID\_ВРАЧА);

CREATE INDEX XIE1СПЕЦИАЛИСТ ON СПЕЦИАЛИСТ(ID\_СПЕЦИАЛИЗАЦИЯ ASC);

CREATE TABLE УЧАСТКОВЫЙ (

ID\_ВРАЧА numeric(6) NOT NULL,

НОМЕР\_УЧАСТКА numeric(6) NOT NULL

);

CREATE UNIQUE INDEX XPKУЧАСТКОВЫЙ ON УЧАСТКОВЫЙ(ID\_ВРАЧА ASC);

ALTER TABLE УЧАСТКОВЫЙ

ADD CONSTRAINT XPKУЧАСТКОВЫЙ

PRIMARY KEY (ID\_ВРАЧА);

CREATE UNIQUE INDEX XAK1УЧАСТКОВЫЙ ON УЧАСТКОВЫЙ(НОМЕР\_УЧАСТКА ASC);

CREATE TABLE ХРОНИЧЕСКИЕ\_ЗАБОЛЕВАНИЯ (

ID\_ПАЦИЕНТА numeric(6) NOT NULL,

ID\_БОЛЕЗНИ numeric(6) NOT NULL

);

CREATE UNIQUE INDEX XPKХРОНИЧЕСКИЕ\_ЗАБОЛЕВАНИЯ ON ХРОНИЧЕСКИЕ\_ЗАБОЛЕВАНИЯ(ID\_ПАЦИЕНТА ASC, ID\_БОЛЕЗНИ ASC);

ALTER TABLE ХРОНИЧЕСКИЕ\_ЗАБОЛЕВАНИЯ

ADD CONSTRAINT XPKХРОНИЧЕСКИЕ\_ЗАБОЛЕВАНИЯ

PRIMARY KEY (ID\_ПАЦИЕНТА, ID\_БОЛЕЗНИ);

CREATE INDEX XIE1ХРОНИЧЕСКИЕ\_ЗАБОЛЕВАНИЯ ON ХРОНИЧЕСКИЕ\_ЗАБОЛЕВАНИЯ(ID\_БОЛЕЗНИ ASC);

CREATE VIEW ПАЦИЕНТУЧАСКОВЫЙ

AS

SELECT ПАЦИЕНТ.ФАМИЛИЯ\_П, ВРАЧ.ФАМИЛИЯ\_В

FROM ПАЦИЕНТ, УЧАСТКОВЫЙ, ВРАЧ, АДРЕС

WHERE (

ПАЦИЕНТ.ID\_УЛИЦЫ = АДРЕС.ID\_УЛИЦЫ

AND ПАЦИЕНТ.НОМЕР\_ДОМА = АДРЕС.НОМЕР\_ДОМА

AND АДРЕС.НОМЕР\_УЧАСТКА = УЧАСТКОВЫЙ.НОМЕР\_УЧАСТКА

AND УЧАСТКОВЫЙ.ID\_ВРАЧА = ВРАЧ.ID\_ВРАЧА

);

CREATE VIEW СПЕЦИАЛИСТСПЕЦИАЛИЗАЦИЯ

AS

SELECT ВРАЧ.ФАМИЛИЯ\_В

FROM СПЕЦИАЛИСТ, ВРАЧ

WHERE СПЕЦИАЛИСТ.ID\_ВРАЧА = ВРАЧ.ID\_ВРАЧА;

ALTER TABLE АДРЕС

ADD CONSTRAINT ПРИКРЕПЛЕН\_К

FOREIGN KEY (НОМЕР\_УЧАСТКА1)

REFERENCES УЧАСТКОВЫЙ (НОМЕР\_УЧАСТКА) ON DELETE SET NULL;

ALTER TABLE ПАЦИЕНТ

ADD CONSTRAINT ПРИНАДЛЕЖИТ

FOREIGN KEY (НОМЕР\_ДОМА, УЛИЦА)

REFERENCES АДРЕС (НОМЕР\_ДОМА, УЛИЦА) ON DELETE SET NULL;

ALTER TABLE РАСПИСАНИЕ

ADD CONSTRAINT РАБОТАЕТ\_ПО

FOREIGN KEY (ID\_ВРАЧА)

REFERENCES ВРАЧ (ID\_ВРАЧА);

ALTER TABLE СПЕЦИАЛИСТ

ADD CONSTRAINT IS\_A\_ВРАЧ

FOREIGN KEY (ID\_ВРАЧА)

REFERENCES ВРАЧ (ID\_ВРАЧА) ON DELETE CASCADE;

ALTER TABLE СПЕЦИАЛИСТ

ADD CONSTRAINT R\_49

FOREIGN KEY (ID\_СПЕЦИАЛИЗАЦИЯ)

REFERENCES КЛАССИФИКАЦИЯ\_СПЕЦИАЛИЗВЦИЙ (ID\_СПЕЦИАЛИЗАЦИЯ);

ALTER TABLE УЧАСТКОВЫЙ

ADD CONSTRAINT IS\_A\_ВРАЧ2

FOREIGN KEY (ID\_ВРАЧА)

REFERENCES ВРАЧ (ID\_ВРАЧА) ON DELETE CASCADE;

ALTER TABLE ХРОНИЧЕСКИЕ\_ЗАБОЛЕВАНИЯ

ADD CONSTRAINT ИМЕЕТ

FOREIGN KEY (ID\_ПАЦИЕНТА)

REFERENCES ПАЦИЕНТ (ID\_ПАЦИЕНТА);

ALTER TABLE ХРОНИЧЕСКИЕ\_ЗАБОЛЕВАНИЯ

ADD CONSTRAINT ХАРАКТЕРИЗУЕТ

FOREIGN KEY (ID\_БОЛЕЗНИ)

REFERENCES ПЕРЕЧЕНЬ\_БОЛЕЗНЕЙ (ID\_БОЛЕЗНИ);