**Министерство науки и высшего образования Российской Федерации**

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Владимирский государственный университет имени Александра Григорьевича и Николая Григорьевича Столетовых**»

(ВлГУ)

Колледж инновационных технологий и предпринимательства

**КАФЕДРА ФИЗИКИ И ПРИКЛАДНОЙ МАТЕМАТИКИ**

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К КУРСОВОЙ РАБОТЕ**

по дисциплине «Технология разработки и защиты баз данных»

Тема: «Медицинские услуги. Больной, болезнь, история: разработка и администрирование базы данных, разработка клиентского приложения»

Выполнил студент

группы ПКсп-120

Владимиров А.И.

Приняла Кабанова М.Ю.

Владимир, 2023

Содержание

[Введение 3](#_Toc127483329)

[1 Проектирование базы данных 4](#_Toc127483330)

[1.1 ER-диаграмма 4](#_Toc127483331)

[2. Подготовка базы данных 6](#_Toc127483332)

[2.1 Словарь данных 6](#_Toc127483333)

[2.2 Нормализация полученных отношений 10](#_Toc127483334)

[2.3 Процедуры и функции 11](#_Toc127483335)

[3. Резервное копирование и восстановление 12](#_Toc127483336)

[4 Разработка клиентского приложения 14](#_Toc127483337)

[4.1 Структура приложения 14](#_Toc127483338)

[4.2 Пользовательский интерфейс 14](#_Toc127483339)

[Заключение 15](#_Toc127483340)

[Список использованных источников 16](#_Toc127483341)

[Приложение А 17](#_Toc127483342)

[Приложение Б 19](#_Toc127483343)

[Приложение В 25](#_Toc127483344)

[Приложение Г 26](#_Toc127483345)

# 

# Введение

SQL Server – это система, созданная для управления набора данных с предопределенными связями между ними, которые организованы в виде таблиц, состоящих из строк и столбцов. SQL Server используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятий, выполняющие функции постоянного хранения данных, поиска по ключевым критериям, чтение и редактирования реляционными базами данных.

В настоящее время базы данных используются для хранения, обработки и структурирования информации. Так же существует возможность множеству людей одновременно использовать базу данных, что делает работу с данными более быстрее и безопасней.

С развитием компьютерной техники появилась возможность автоматизировать многие процессы, в связи со стремительным развитием строительства увеличился объем обрабатываемой информации, и поэтому возникла объективная причина автоматизировать систему заказа материалов, хранение информации о заключённых договоров между клиентом и застройщиком, информацию о постройке, так же информацию, когда началась стройка, когда закончится, кто на ней работает и многое другое. Если хранить данную информацию в базе данных, то вся необходимая информация хранится в структурированном виде. База данных может использоваться в любой больнице для эффективного хранения и обработки большого количества информации.

# 1 Проектирование базы данных

## 1.1 ER-диаграмма

В базе данных «Медицинские услуги. Больной, болезнь, история.» содержится следующий набор сущностей с присущими им атрибутами:

* + Врач (doctor): Должность; Фамилия; Имя; Отчество;
  + История болезни (medical\_history): Мед. карта; Номер электронной мед. карты
  + Пациент (patient): Фамилия; Имя; Отчество; Пол; Возраст; Номер телефона; Серия и номер паспорта;
  + Заболевание (disease): Дата заболевания; Температура;
  + Полис (policy): Страховая компания;
  + Поликлиника (hospital): Улица, Номер дома;

Между этими сущностями организованы связи типа «один-ко-многим», «многие-ко-многим» через ассоциативные таблицы.

Диаграмма Чена представлена на рисунке 1.

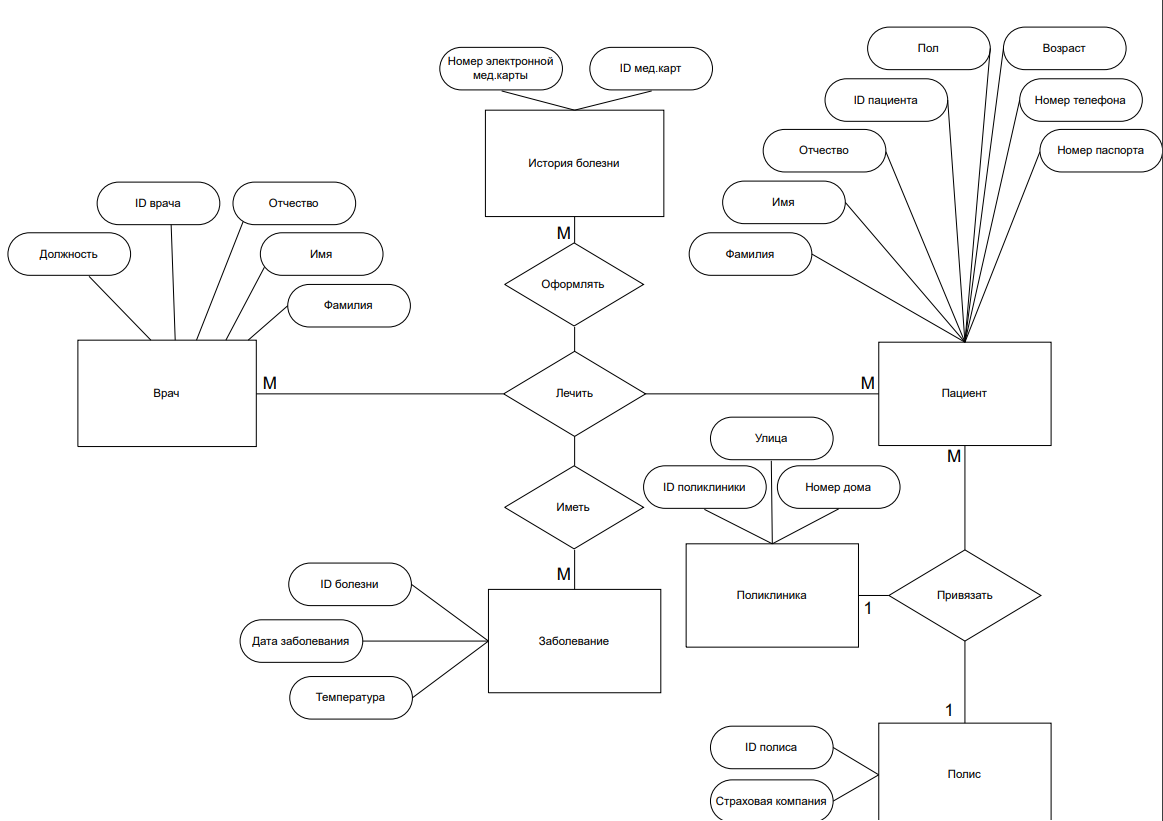


Рисунок 1. Диаграмма Чена.

Диаграмма Мартина представлена на рисунке 2.

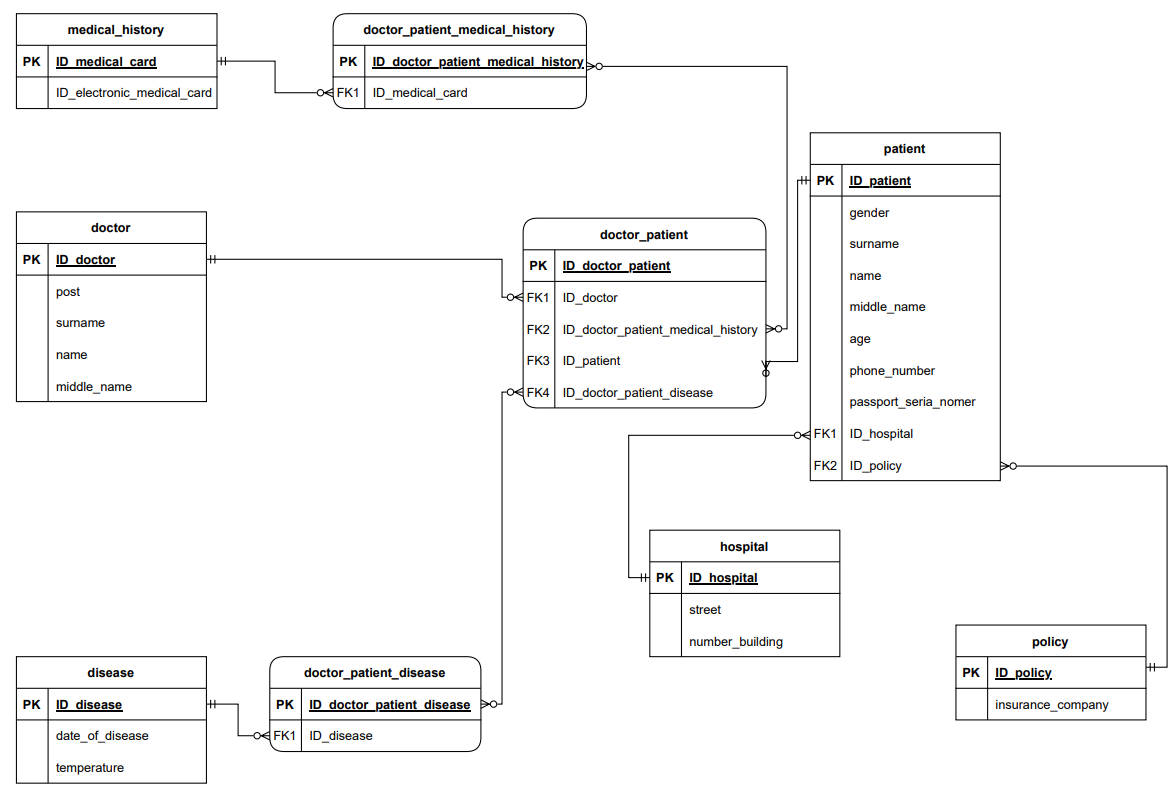


Рисунок 2. Диаграмма Мартина.

Исходя из этих таблиц можно составить словарь данных

# 2. Подготовка базы данных

## 2.1 Словарь данных

Словарь данных для создания базы данных представлен в таблице 1.

Таблица 1 – Словарь данных

|  |  |  |  |
| --- | --- | --- | --- |
| Ключ | Поле | Обязательное (да/нет) | Примечание |
| doctor | | | |
| Первичный ключ | id\_doctor | Да | Id врача |
|  | surname | Да | Фамилия врача |
|  | name | Да | Имя врача |
|  | middle\_name | Нет | Отчество врача |
|  | post | Да | Должность врача |
| doctor\_patient | | | |
| Первичный ключ | id\_doctor\_patient | Да | Id врача и пациента |
| Внешний ключ | id\_doctor | Да | Id врача |
| Внешний ключ | id\_doctor\_patient\_medical\_history | Да | Внешний ключ на ассоциативную таблицу |
| Внешний ключ | id\_patient | Да | Id пациента |
| Внешний ключ | id\_doctor\_patient\_disease | Да | Внешний ключ на ассоциативную таблицу |
| medical\_history | | | |
| Первичный ключ | id\_medical\_card | Да | Id мед. карты |
|  | number\_electronic\_medical\_card | Да | Номер электронной мед. карты |
| doctor\_patient\_medical\_history | | | |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| Первичный ключ | id\_ doctor\_patient\_medical\_history | Да | Id врача, пациента и истории болезни |
| Внешний ключ | id\_medical\_card | Да | Id мед. карты.  Внешний ключ на таблицу medical\_history |
| disease | | | |
| Первичный ключ | id\_disease | Да | Id болезни |
|  | date\_of\_disease | Да | Дата заболевания |
|  | temperature | Нет | Значение температуры |
| doctor\_patient\_disease | | | |
| Первичный ключ | ID\_doctor\_patient\_disease | Да | Id врача, пациента и болезни |
| Внешний ключ | id\_disease | Да | Id болезни.  Внешний ключ на таблицу disease |
| patient | | | |
| Первичный ключ | ID\_patient | Да | Id пациента |
|  | gender | Да | Пол пациента |
|  | surname | Да | Фамилия |
|  | name | Да | Имя |
|  | middle\_name | Нет | Отчество |
|  | age | Да | Возраст |
|  | phone\_number | Да | Номер телефона |
|  | passport\_seria\_nomer | Да | Серия и номер паспорта |
| Внешний ключ | ID\_hospital | Да | Id поликлиники.  Внешний ключ на таблицу hospital |
| Внешний ключ | ID\_policy | Да | Id полиса.  Внешний ключ на таблицу policy |

Продолжение таблицы 1

|  |  |  |  |
| --- | --- | --- | --- |
| hospital | | | |
| Первичный ключ | ID\_hospital | Да | Id поликлиники |
|  | street | Да | Улица |
|  | number\_building | Да | Номер дома |
| policy | | | |
| Первичный ключ | ID\_policy | Да | Id полиса |
|  | insurance\_company | Да | Страховая компания полиса |

Итоговая диаграмма базы данных, сгенерированная после создания всех таблиц представлена на рисунке 3.

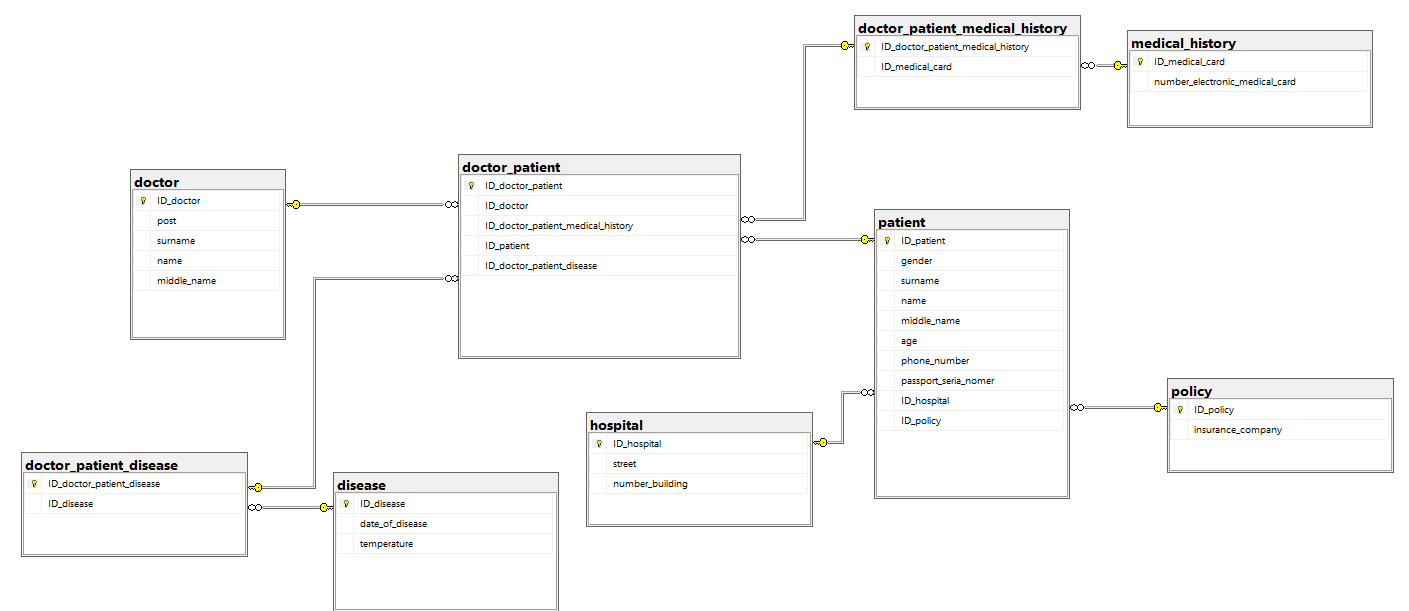


Рисунок 3. Диаграмма базы данных

Код создания таблиц представлен в Приложении А. Код заполнения базы данных представлен в приложении Б.

## 2.2 Нормализация полученных отношений

Разработанная схема базы данных находится в:

* Первой нормальной форме, так как в качестве доменов выступают только скалярные значения, и информация в таблицах не дублируется. Во всех таблицах есть идентификатор id в качестве первичного ключа
* Второй и третьей нормальных формах, каждый не ключевой атрибут не приводимо и не транзитивно зависит от первичного ключа. Схема базы данных находится в нормальной форме Бойса-Кодда.

## 2.3 Процедуры и функции

В базе данных реализованы процедура, хранимая функция, триггеры. Хранимая процедура имеет название count\_patient. Она предназначена для того, чтобы вывести в одну таблицу информацию о количестве всех пациентов, содержащиеся в базе данных. На выходе из хранимой процедуры формируется выборка в которую будет входить такое поле как:

* Количество пациентов мужского пола. В ней пишется количество пациентов мужского пола
* В базе данных реализована скалярная функция fio\_doctor, которая выводит фамилию, имя и отчество докторов

В базе данных разработаны триггеры:

* insert\_trig. Служит для того, чтобы вместо добавления новых строк в таблицу Equipment добавлять запись в таблицу Updates\_equipment.
* del\_rollback. Нужна для отката удалений и вывода сообщение.
* upd\_equipment. Используется для добавления записи в таблицу Update\_equipment об обновлений таблицы Equipment.

Коды создания хранимой процедуры, функции, триггеров представлены в приложении В.

# 3. Резервное копирование и восстановление

Чтобы выполнить резервное копирование базы данных с использованием командного файла, нужно написать SQL-скрипт, который реализует резервное копирование базы, затем сохранить данный скрипт в файле с расширением .sql. Скрипт для резервного копирования представлен в листинге 1.

Листинг 1

BACKUP DATABASE [Med\_Yslugi]

TO DISK = N'C:\SQL Buckup\Med\_Yslugi.bak' WITH NOFORMAT, NOINIT, NAME = N'Med\_Yslugi-Full Database Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10

GO

Для запуска данного скрипта необходимо с помощью любого текстового редактора создать скрипт файл с расширением .bat, внутри которого написать код, представленный в листинге 2.

Листинг 2

sqlcmd -s laptop-clvpr4q1 -i C:\SQL Buckup\back.bat

Теперь при запуске файла будет создаваться резервная копия базы данных в указанной папке.

Так же был создан SQL-скрипт и командный файл для запуска восстановления базы данных из резервной копии. SQL-скрипт для восстановления представлен в листинге 3.

Листинг 3

USE [master]

BACKUP LOG [Med\_Yslugi] TO DISK = N'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\Backup\Med\_Yslugi\_LogBackup\_2023-03-09\_11-13-42.bak' WITH NOFORMAT, NOINIT, NAME = N'Med\_Yslugi\_LogBackup\_2023-03-09\_11-13-42', NOSKIP, NOREWIND, NOUNLOAD, NORECOVERY , STATS = 5

RESTORE DATABASE [Med\_Yslugi] FROM DISK = N'C:\SQL Buckup\Med\_Yslugi.bak' WITH FILE = 1, NOUNLOAD, STATS = 5

GO

Код командного файла представлен в листинге 4.

Листинг 4

sqlcmd -s laptop-clvpr4q1 -i C:\SQL Buckup\res.sql

Теперь база данных содержит резервную копию, которой можно воспользоваться в любой момент и открыть на других компьютерах.

# 4 УПРАВЛЕНИЕ ДОСТУПОМ

Для управления доступом в базе банных с помощью sql скриптов были созданы 2 роли:

* admin, которая не имеет ограничений в работе с данными в базе данных.
* visiter, которая может лишь просматривать данные в таблицах базы данных, соответственно для этой роли ограничены возможности добавления, удаления и редактирования данных.

Для каждой роли были созданы пользователи

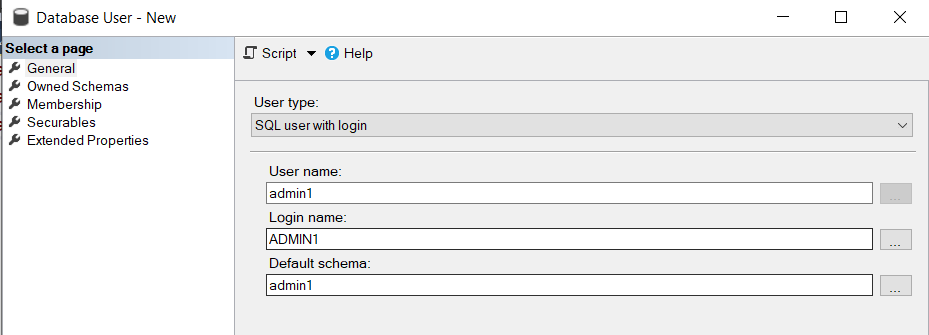


Рисунок 4 - пользователь без ограничений

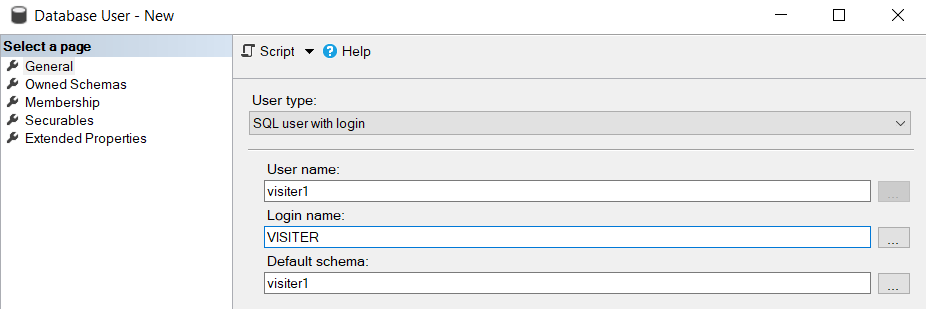


Рисунок 5 - пользователь с ограничениями

SQL скрипт выполняющий создание ролей и пользователей представлены в приложении В

# 4 Разработка клиентского приложения

## 4.1 Структура приложения

Текст раздела

Конец раздела

## 4.2 Пользовательский интерфейс

Текст раздела

# Заключение

Текст раздела

# Список использованных источников

1)

# Приложение А

Создание таблицы doctor

CREATE TABLE doctor

(

ID\_doctor INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

post VARCHAR(40) NOT NULL,

surname VARCHAR(40) NOT NULL,

[name] VARCHAR(30) NOT NULL ,

middle\_name VARCHAR(50)

)

Создание таблицы policy

CREATE TABLE [policy]

(

ID\_policy INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

insurance\_company VARCHAR(50) NOT NULL,

)

Создание таблицы patient

CREATE TABLE patient

(

ID\_patient INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

gender varchar (1) NOT NULL check (gender in ('м', 'М', 'ж', 'Ж')),

surname VARCHAR(40) NOT NULL,

[name] VARCHAR(30) NOT NULL ,

middle\_name VARCHAR(50)

age int NOT NULL,

phone\_number char (13) NOT NULL check (phone\_number like '[1-9][0-9][0-9]-[0-9][0-9][0-9]-[0-9][0-9]-[0-9][0-9]'),

passport\_seria\_nomer char (12) NOT NULL check (passport\_seria\_nomer like '[0-9][0-9]-[0-9][0-9]-[0-9][0-9][0-9][0-9][0-9][0-9]'),

ID\_hospital int not null

CONSTRAINT fk\_ID\_hospital FOREIGN KEY (ID\_hospital) REFERENCES hospital,

ID\_policy int not null

CONSTRAINT fk\_ID\_policy FOREIGN KEY (ID\_policy) REFERENCES [policy]

)

Создание таблицы medical\_history

CREATE TABLE medical\_history

(

ID\_medical\_card INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

number\_electronic\_medical\_card INT NOT NULL,

)

Создание таблицы hospital

CREATE TABLE hospital

(

ID\_hospital INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

street VARCHAR(50) NOT NULL,

number\_building VARCHAR(50) NOT NULL,

)

Создание таблицы doctor\_patient\_medical\_history

CREATE TABLE doctor\_patient\_medical\_history

(

ID\_doctor\_patient\_medical\_history INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

ID\_medical\_card INT NOT NULL,

CONSTRAINT fk\_ID\_medical\_card FOREIGN KEY (ID\_medical\_card) REFERENCES medical\_history

)

Создание таблицы doctor\_patient\_disease

CREATE TABLE doctor\_patient\_disease

(

ID\_doctor\_patient\_disease INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

ID\_disease INT NOT NULL,

CONSTRAINT fk\_ID\_disease FOREIGN KEY (ID\_disease) REFERENCES disease

)

Создание таблицы doctor\_patient

CREATE TABLE doctor\_patient

(

ID\_doctor\_patient INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

ID\_doctor INT NOT NULL,

CONSTRAINT fk\_ID\_doctor FOREIGN KEY (ID\_doctor) REFERENCES doctor,

ID\_doctor\_patient\_medical\_history INT NOT NULL,

CONSTRAINT fk\_ID\_doctor\_patient\_medical\_history FOREIGN KEY (ID\_doctor\_patient\_medical\_history) REFERENCES doctor\_patient\_medical\_history,

ID\_patient INT NOT NULL,

CONSTRAINT fk\_ID\_patient FOREIGN KEY (ID\_patient) REFERENCES patient,

ID\_doctor\_patient\_disease INT NOT NULL,

CONSTRAINT fk\_ID\_doctor\_patient\_disease FOREIGN KEY (ID\_doctor\_patient\_disease) REFERENCES doctor\_patient\_disease

)

Создание таблицы desease

CREATE TABLE disease

(

ID\_disease INT NOT NULL IDENTITY(1,1) PRIMARY KEY,

date\_of\_disease date NOT NULL,

temperature decimal (3,1)f

)

# Приложение Б

Заполнение таблицы doctor

insert into doctor (post, surname, [name], middle\_name)

values ('Хирург', 'Ширяев', 'Арнольд', 'Тимурович')

insert into doctor (post, surname, [name], middle\_name)

values ('Хирург', 'Евсеев', 'Мстислав', 'Евсеевич')

insert into doctor (post, surname, [name], middle\_name)

values ('Терапевт', 'Муравьёв', 'Вадим', 'Антонович')

insert into doctor (post, surname, [name], middle\_name)

values ('Терапевт', 'Исаев ', 'Георгий ', 'Иванович')

insert into doctor (post, surname, [name], middle\_name)

values ('Терапевт', 'Суханов', 'Глеб ', 'Геннадьевич')

insert into doctor (post, surname, [name], middle\_name)

values ('Терапевт', 'Шашков ', 'Василий ', 'Георгьевич')

insert into doctor (post, surname, [name], middle\_name)

values ('Терапевт', 'Крюков', 'Касьян ', 'Вадимович')

insert into doctor (post, surname, [name], middle\_name)

values ('Терапевт', 'Аксёнов', 'Альберт', 'Тихонович')

insert into doctor (post, surname, [name], middle\_name)

values ('Терапевт', 'Карпов', 'Адриан ', 'Лаврентьевич')

insert into doctor (post, surname, [name], middle\_name)

values ('Терапевт', 'Агафонов ', 'Михаил ', 'Сергеевич')

Заполнение таблицы policy

insert into [policy] (insurance\_company)

values ('Ингосстрах')

insert into [policy] (insurance\_company)

values ('Ингосстрах')

insert into [policy] (insurance\_company)

values ('Ингосстрах')

insert into [policy] (insurance\_company)

values ('Ингосстрах')

insert into [policy] (insurance\_company)

values ('Ингосстрах')

insert into [policy] (insurance\_company)

values ('Ингосстрах')

insert into [policy] (insurance\_company)

values ('Согаз')

insert into [policy] (insurance\_company)

values ('Согаз')

insert into [policy] (insurance\_company)

values ('Согаз')

insert into [policy] (insurance\_company)

values ('Согаз')

Заполнение таблицы patient

insert into patient (gender, surname, [name], middle\_name, age, phone\_number, passport\_seria\_nomer, ID\_hospital, ID\_policy)

values ('м', 'Савельев', 'Гордей', 'Петрович', '32', '919-002-39-93', '39-24-754893', 1, 1)

insert into patient (gender, surname, [name], middle\_name, age, phone\_number, passport\_seria\_nomer, ID\_hospital, ID\_policy)

values ('м', 'Николаев ', 'Ярослав ', 'Кириллович', '39', '915-142-29-42', '25-64-789893', 1, 1)

insert into patient (gender, surname, [name], middle\_name, age, phone\_number, passport\_seria\_nomer, ID\_hospital, ID\_policy)

values ('м', 'Голубев ', 'Андрей ', 'Васильевич', '29', '910-742-29-43', '69-34-798753', 1, 1)

insert into patient (gender, surname, [name], middle\_name, age, phone\_number, passport\_seria\_nomer, ID\_hospital, ID\_policy)

values ('м', 'Зайцев ', 'Панкратий ', 'Христофорович', '53', '915-773-29-58', '23-77-249191', 1, 1)

insert into patient (gender, surname, [name], middle\_name, age, phone\_number, passport\_seria\_nomer, ID\_hospital, ID\_policy)

values ('м', 'Ширяев ', 'Ярослав ', 'Альбертович', '33', '905-345-79-82', '45-24-787351', 1, 1)

insert into patient (gender, surname, [name], middle\_name, age, phone\_number, passport\_seria\_nomer, ID\_hospital, ID\_policy)

values ('м', 'Нестеров ', 'Лаврентий ', 'Федорович', '39', '915-987-49-74', '25-64-875392', 1, 1)

insert into patient (gender, surname, [name], middle\_name, age, phone\_number, passport\_seria\_nomer, ID\_hospital, ID\_policy)

values ('м', 'Рогов ', 'Анатолий ', 'Ярославович', '36', '905-334-49-22', '65-26-567288', 1, 1)

insert into patient (gender, surname, [name], middle\_name, age, phone\_number, passport\_seria\_nomer, ID\_hospital, ID\_policy)

values ('м', 'Воробьёв ', 'Андрей ', 'Васильевич', '63', '900-985-29-64', '73-63-876535', 1, 1)

insert into patient (gender, surname, [name], middle\_name, age, phone\_number, passport\_seria\_nomer, ID\_hospital, ID\_policy)

values ('м', 'Ильин ', 'Гордей ', 'Русланович', '32', '955-373-73-42', '62-86-656847', 1, 1)

insert into patient (gender, surname, [name], middle\_name, age, phone\_number, passport\_seria\_nomer, ID\_hospital, ID\_policy)

values ('м', 'Николаев ', 'Ибрагим', 'Русланович', '51', '900-935-39-21', '26-67-153315', 1, 1)

Заполнение таблицы medical\_history

insert into medical\_history (number\_electronic\_medical\_card)

values (1)

insert into medical\_history (number\_electronic\_medical\_card)

values (2)

insert into medical\_history (number\_electronic\_medical\_card)

values (3)

insert into medical\_history (number\_electronic\_medical\_card)

values (4)

insert into medical\_history (number\_electronic\_medical\_card)

values (5)

insert into medical\_history (number\_electronic\_medical\_card)

values (6)

insert into medical\_history (number\_electronic\_medical\_card)

values (7)

insert into medical\_history (number\_electronic\_medical\_card)

values (8)

insert into medical\_history (number\_electronic\_medical\_card)

values (9)

insert into medical\_history (number\_electronic\_medical\_card)

values (10)

Заполнение таблицы hospital

insert into hospital (street, number\_building)

values ('Оружейный переулок', '43')

insert into hospital (street, number\_building)

values ('Оружейный переулок', '43')

insert into hospital (street, number\_building)

values ('Оружейный переулок', '43')

insert into hospital (street, number\_building)

values ('Оружейный переулок', '43')

insert into hospital (street, number\_building)

values ('Оружейный переулок', '43')

insert into hospital (street, number\_building)

values ('Оружейный переулок', '43')

insert into hospital (street, number\_building)

values ('Оружейный переулок', '43')

insert into hospital (street, number\_building)

values ('Оружейный переулок', '43')

insert into hospital (street, number\_building)

values ('Оружейный переулок', '43')

insert into hospital (street, number\_building)

values ('Оружейный переулок', '43')

Заполнение таблицы doctor\_patient\_medical\_history

insert into doctor\_patient\_medical\_history (ID\_medical\_card)

values (1)

insert into doctor\_patient\_medical\_history (ID\_medical\_card)

values (2)

insert into doctor\_patient\_medical\_history (ID\_medical\_card)

values (3)

insert into doctor\_patient\_medical\_history (ID\_medical\_card)

values (4)

insert into doctor\_patient\_medical\_history (ID\_medical\_card)

values (5)

insert into doctor\_patient\_medical\_history (ID\_medical\_card)

values (6)

insert into doctor\_patient\_medical\_history (ID\_medical\_card)

values (7)

insert into doctor\_patient\_medical\_history (ID\_medical\_card)

values (8)

insert into doctor\_patient\_medical\_history (ID\_medical\_card)

values (9)

insert into doctor\_patient\_medical\_history (ID\_medical\_card)

values (10)

Заполнение таблицы doctor\_patient\_disease

insert into doctor\_patient\_disease (ID\_disease)

values (1)

insert into doctor\_patient\_disease (ID\_disease)

values (2)

insert into doctor\_patient\_disease (ID\_disease)

values (3)

insert into doctor\_patient\_disease (ID\_disease)

values (4)

insert into doctor\_patient\_disease (ID\_disease)

values (5)

insert into doctor\_patient\_disease (ID\_disease)

values (6)

insert into doctor\_patient\_disease (ID\_disease)

values (7)

insert into doctor\_patient\_disease (ID\_disease)

values (8)

insert into doctor\_patient\_disease (ID\_disease)

values (9)

insert into doctor\_patient\_disease (ID\_disease)

values (10)

Заполнение таблицы doctor\_patient

insert into doctor\_patient (ID\_doctor, ID\_doctor\_patient\_medical\_history, ID\_patient, ID\_doctor\_patient\_disease)

values (1, 1, 21, 31)

insert into doctor\_patient (ID\_doctor, ID\_doctor\_patient\_medical\_history, ID\_patient, ID\_doctor\_patient\_disease)

values (2, 2, 22, 32)

insert into doctor\_patient (ID\_doctor, ID\_doctor\_patient\_medical\_history, ID\_patient, ID\_doctor\_patient\_disease)

values (3, 3, 23, 33)

insert into doctor\_patient (ID\_doctor, ID\_doctor\_patient\_medical\_history, ID\_patient, ID\_doctor\_patient\_disease)

values (4, 4, 24, 34)

insert into doctor\_patient (ID\_doctor, ID\_doctor\_patient\_medical\_history, ID\_patient, ID\_doctor\_patient\_disease)

values (5, 5, 25, 35)

insert into doctor\_patient (ID\_doctor, ID\_doctor\_patient\_medical\_history, ID\_patient, ID\_doctor\_patient\_disease)

values (6, 6, 26, 36)

insert into doctor\_patient (ID\_doctor, ID\_doctor\_patient\_medical\_history, ID\_patient, ID\_doctor\_patient\_disease)

values (7, 7, 27, 37)

insert into doctor\_patient (ID\_doctor, ID\_doctor\_patient\_medical\_history, ID\_patient, ID\_doctor\_patient\_disease)

values (8, 8, 28, 38)

insert into doctor\_patient (ID\_doctor, ID\_doctor\_patient\_medical\_history, ID\_patient, ID\_doctor\_patient\_disease)

values (9, 9, 29, 39)

insert into doctor\_patient (ID\_doctor, ID\_doctor\_patient\_medical\_history, ID\_patient, ID\_doctor\_patient\_disease)

values (10, 10, 30, 40)

Заполнение таблицы desease

insert into disease (date\_of\_disease, temperature)

values ('2023-01-11', 36.5)

insert into disease (date\_of\_disease, temperature)

values ('2023-01-2', 36.9)

insert into disease (date\_of\_disease, temperature)

values ('2023-01-13', 38.4)

insert into disease (date\_of\_disease, temperature)

values ('2023-02-1', 37.9)

insert into disease (date\_of\_disease, temperature)

values ('2023-01-21', 36.5)

insert into disease (date\_of\_disease, temperature)

values ('2023-01-26', 38.9)

insert into disease (date\_of\_disease, temperature)

values ('2023-01-19', 38.1)

insert into disease (date\_of\_disease, temperature)

values ('2023-02-15', 40.0)

insert into disease (date\_of\_disease, temperature)

values ('2023-01-20', 38.2)

insert into disease (date\_of\_disease, temperature)

values ('2023-02-5', 39.8)

# Приложение В

Создание функции

alter PROC count\_patient AS

SELECT COUNT([name]) as 'Количество пациентов мужского пола'

FROM patient p

WHERE p.gender = 'м'

GO

Создание функции

alter FUNCTION fio\_doctor (@fio varchar(155))

RETURNS varchar(155)

BEGIN

DECLARE @f varchar(55)

SELECT @f = surname

FROM doctor

DECLARE @i varchar(55)

SELECT @i = [name]

FROM doctor

DECLARE @o varchar(55)

SELECT @o = middle\_name

FROM doctor

select @fio = @f + @i + @o

RETURN @fio

END

Триггер insert

create trigger insert\_trig on doctor instead of insert as

begin

insert into Updates\_doctors

select ID\_doctor, 'Попытка добавления доктора' + post, surname , [name] , middle\_name, GETDATE()

FROM inserted

end

Триггер delete

create trigger del\_rollback on doctor

after delete as

begin

rollback

print 'Удаление'

end

Триггер update

create trigger upd\_equipment on doctor after update as

begin

insert into Updates\_doctors

select ID\_doctor, post, surname , [name] , middle\_name, GETDATE()

FROM INSERTED

End

Создание роли visiter

exec sp\_addrole 'visiter'

GRANT select

to sysAdm with grant option

Создание роли sysAdm

exec sp\_addrole 'sysAdm'

GRANT select, update , insert , delete

to sysAdm with grant option

Создание пользователя admin1

exec sp\_addlogin 'ADMIN1', 'admin1' , 'Med\_Yslugi' , 'Russian'

exec sp\_adduser 'ADMIN1' , 'admin1'

exec sp\_addrolemember 'ADMIN1' , 'admin1'

Создание пользователя visiter1

exec sp\_addlogin 'VISITER', 'visiter1' , 'Med\_Yslugi' , 'Russian'

exec sp\_adduser 'VISITER' , 'visiter1'

exec sp\_addrolemember 'VISITER' , 'visiter1'

# Приложение Г

Код главной формы

Код формы добавления/редактирования