**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КОМИ**

**Государственное профессиональное образовательное учреждение**

**«Воркутинский арктический горно-политехнический колледж»**

**КУРСОВОЙ ПРОЕКТ**

По дисциплине МДК.07.01 Управление и автоматизация баз данных

**Разработка базы данных  
“Поликлиника”**

Выполнил студент гр. ИСП-20 / / Райков Степан Владимирович /

(подпись) (Ф.И.О.)

**ОЦЕНКА:\_\_\_\_\_\_\_\_\_\_\_\_\_**

Дата:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**ПРОВЕРИЛ**

Научный руководитель\_\_\_\_\_\_\_\_\_\_\_\_\_/ Егоров Данил Павлович/

(подпись) (Ф.И.О.)

Воркута

2023

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc153145167)

[ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ 5](#_Toc153145168)

[1.1 СУБД SQL server 5](#_Toc153145169)

[1.2 Microsoft SQL Server Management Studio 8](#_Toc153145170)

[1.3 Entity Framework 10](#_Toc153145171)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 12](#_Toc153145172)

[2.1 Разработка диаграммы ERD 12](#_Toc153145173)

[2.2 Разработка базы данных 13](#_Toc153145174)

[2.3 Словарь данных 19](#_Toc153145175)

[ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ 23](#_Toc153145176)

[3.1 Введение в безопасность SQL Server 23](#_Toc153145177)

[3.2 Управление безопасностью уровня сервера 24](#_Toc153145178)

[3.3 Управление участниками уровня базы данных 28](#_Toc153145179)

[3.4 Управление разрешениями уровня базы данных 30](#_Toc153145180)

[3.5 Шифрование данных баз данных 32](#_Toc153145181)

[ЗАКЛЮЧЕНИЕ 34](#_Toc153145182)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 36](#_Toc153145183)

# ВВЕДЕНИЕ

Сегодня управление предприятием без компьютера просто немыслимо. Компьютеры давно и прочно вошли в такие области управления, как бухгалтерский учет, управление складом, ассортиментом, а также и медициной. Однако современный бизнес, как и многие другие направления, требует гораздо более широкого применения информационных технологий в управлении предприятием. Жизнеспособность и развитие информационных технологий объясняется тем, что современный бизнес крайне чувствителен к ошибкам в управлении. Интуиции, личного опыта руководителя и размеров капитала уже мало для того, чтобы быть первым. Для принятия любого грамотного управленческого решения в условиях неопределенности и риска необходимо постоянно держать под контролем различные аспекты финансово-хозяйственной деятельности, будь то: торговля, медицина или предоставление каких-либо услуг. И чем крупнее предприятие, тем серьезнее должны быть подобные вложения сил и средств. Они являются жизненной необходимостью — в жесткой конкурентной борьбе одержать победу сможет лишь тот, кто лучше оснащен и наиболее эффективно организован.

Автоматизированная информационная система «Поликлиника» включает в себя данные о врачах и пациента, которые необходимые для работы поликлиники. База данных позволяет осуществлять добавление, изменение, поиск и удаление данных, а также просматривать эти данные.

Актуальность данной темы в том, что в наш век информационных технологий, стало реально все документы преобразовывать в электронный вид и регистратура в считанные минуты может найти сведения о принятых пациентах, врачах и истории болезней.

**Объект:** база данных для информационной системы “поликлиника”.

**Предмет:** автоматизация бизнес - процессов работы поликлиники.

**Цель работы:** разработать базу данных для автоматизированной информационной системы работы поликлиники.

**Задачи:**

* Выбрать инструментарий базы данных для информационной системы ”поликлиника”;
* Спроектировать базу данных к информационной системе ”поликлиника”;
* Сделать разграничение ролей для базы данных;
* Сделать шифрование данных для базы данных;
* Сделать резервное копирование для базы данных.

# ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ

# СУБД SQL server

Microsoft SQL Server является одной из наиболее популярных систем управления базами данных (СУБД) в мире. Данная СУБД подходит для самых различных проектов: от небольших приложений до больших высоконагруженных проектов.

SQL Server был создан компанией Microsoft. Первая версия вышла в 1987 году. А текущей версией является версия 2022, которая вышла в ноябре 2022 году и которая будет использоваться в текущем руководстве.

SQL Server долгое время был исключительно системой управления базами данных для Windows, однако начиная с версии 16 эта система доступна и на Linux.

SQL Server характеризуется такими особенностями как:

Производительность. SQL Server работает очень быстро.

Надежность и безопасность. SQL Server предоставляет шифрование данных.

Простота. С данной СУБД относительно легко работать и вести администрирование.

Центральным аспектом в MS SQL Server, как и в любой СУБД, является база данных. База данных представляет хранилище данных, организованных определенным способом. Нередко физически база данных представляет файл на жестком диске, хотя такое соответствие необязательно. Для хранения и администрирования баз данных применяются системы управления базами данных (database management system) или СУБД (DBMS). И как раз MS SQL Server является одной из такой СУБД.

Для организации баз данных MS SQL Server использует реляционную модель. Эта модель баз данных была разработана еще в 1970 году Эдгаром Коддом. А на сегодняшний день она фактически является стандартом для организации баз данных.

Реляционная модель предполагает хранение данных в виде таблиц, каждая из которых состоит из строк и столбцов. Каждая строка хранит отдельный объект, а в столбцах размещаются атрибуты этого объекта.

Для идентификации каждой строки в рамках таблицы применяется первичный ключ (primary key). В качестве первичного ключа может выступать один или несколько столбцов. Используя первичный ключ, мы можем ссылаться на определенную строку в таблице. Соответственно две строки не могут иметь один и тот же первичный ключ.

Через ключи одна таблица может быть связана с другой, то есть между двумя таблицами могут быть организованы связи. А сама таблица может быть представлена в виде отношения ("relation").

Для взаимодействия с базой данных применяется язык SQL (Structured Query Language). Клиент (например, внешняя программа) отправляет запрос на языке SQL посредством специального API. СУБД должным образом интерпретирует и выполняет запрос, а затем посылает клиенту результат выполнения.

Изначально язык SQL был разработан в компании IBM для системы баз данных, которая называлась System/R. При этом сам язык назывался SEQUEL (Structured English Query Language). Хотя в итоге ни база данных, ни сам язык не были впоследствии официально опубликованы, по традиции сам термин SQL нередко произносят как "сиквел".

В 1979 году компания Relational Software Inc. разработала первую систему управления баз данных, которая называлась Oracle и которая использовала язык SQL. В связи с успехом данного продукта компания была переименована в Oracle.

Впоследствии стали появляться другие системы баз данных, которые использовали SQL. В итоге в 1989 году Американский Национальный Институт Стандартов (ANSI) кодифицировал язык и опубликовал его первый стандарт. После этого стандарт периодически обновлялся и дополнялся. Последнее его обновление состоялось в 2011 году. Но несмотря на наличие стандарта нередко производители СУБД используют свои собственные реализации языка SQL, которые немного отличаются друг от друга.

Выделяются две разновидности языка SQL: PL-SQL и T-SQL. PL-SQL используется в таких СУБД как Oracle и MySQL. T-SQL (Transact-SQL) применяется в SQL Server. Собственно поэтому, в рамках текущего руководства будет рассматриваться именно T-SQL.

В зависимости от задачи, которую выполняет команда T-SQL, он может принадлежать к одному из следующих типов:

DDL (Data Definition Language / Язык определения данных). К этому типу относятся различные команды, которые создают базу данных, таблицы, индексы, хранимые процедуры и т.д. В общем определяют данные.

В частности, к этому типу мы можем отнести следующие команды:

CREATE: создает объекты базы данных (саму базу даных, таблицы, индексы и т.д.)

ALTER: изменяет объекты базы данных

DROP: удаляет объекты базы данных

TRUNCATE: удаляет все данные из таблиц

DML (Data Manipulation Language / Язык манипуляции данными). К этому типу относят команды на выбор данных, их обновление, добавление, удаление - в общем все те команды, с помощью которыми мы можем управлять данными.

К этому типу относятся следующие команды:

SELECT: извлекает данные из БД

UPDATE: обновляет данные

INSERT: добавляет новые данные

DELETE: удаляет данные

DCL (Data Control Language / Язык управления доступа к данным). К этому типу относят команды, которые управляют правами по доступу к данным. В частности, это следующие команды:

GRANT: предоставляет права для доступа к данным

REVOKE: отзывает права на доступ к данным

# Microsoft SQL Server Management Studio

SQL Server Management Studio (SSMS) — утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера.

Главным инструментом SQL Server Management Studio является Object Explorer, который позволяет пользователю просматривать, извлекать объекты сервера, а также полностью ими управлять.

Также есть SQL Server Management Studio Express для Express версии сервера, которая является бесплатной. Однако в ней нет поддержки ряда компонентов (Analysis Services, Integration Services, Notification Services, Reporting Services) и SQL Server 2005 Mobile Edition.

Начиная с версии 16.5.3 пакет SSMS выделен в отдельный обновляемый продукт, доступный для скачивания на сайте Microsoft. Текущая доступная версия SSMS 18.4 (15.0.18206.0) (general availability) поддерживает MS SQL server начиная с версии 2008 по 2019.

Среда SQL Server Management Studio – это основной, стандартный и полнофункциональный инструмент для работы с Microsoft SQL Server, разработанный компанией Microsoft, который предназначен как для разработчиков, так и для администраторов SQL Server.

С помощью SSMS Вы можете разрабатывать базы данных, выполнять инструкции T-SQL, а также администрировать Microsoft SQL Server.

Если в Ваши задачи входит полное сопровождение Microsoft SQL Server, начиная от создания баз данных, написания SQL запросов, создания хранимых процедур и функций, и заканчивая администрированием SQL Server, включая управление безопасностью, то основным Вашим инструментом будет как раз среда SQL Server Management Studio.

Среда SQL Server Management Studio реализована только под Windows, поэтому если Вам нужен инструмент для работы с Microsoft SQL Server, который будет работать на других платформах, например, на Linux или macOS, то Вам следует использовать инструмент Azure Data Studio, который также является официальным инструментом, разработанным компанией Microsoft.

Основной функционал SQL Server Management Studio

Теперь давайте рассмотрим функционал и возможности среды SQL Server Management Studio, иными словами, какие именно действия и операции мы можем выполнять, используя данный инструмент.

Сначала давайте посмотрим на общий перечень возможностей, которые нам предоставляет среда SQL Server Management Studio, а затем более подробно рассмотрим каждый пункт из этого перечня.

Основной функционал SSMS

* Подключение к службам SQL Server
* Обозреватель объектов
* Обозреватель шаблонов
* Редактор SQL кода
* Просмотр плана выполнения запроса
* Обозреватель решений
* Конструктор таблиц
* Конструктор баз данных (Диаграммы баз данных)
* Конструктор запросов и представлений
* Просмотр свойств объектов
* Мастер создания скриптов
* Управление безопасностью SQL Server
* Присоединение и отсоединение баз данных
* Создание резервных копий баз данных и восстановление баз данных из архива
* Создание связанных серверов (Linked Server)
* Монитор активности SQL Server
* Профилировщик XEvent

# Entity Framework

ADO.NET Entity Framework (EF) — объектно-ориентированная технология доступа к данным, является object-relational mapping (ORM) решением для .NET Framework от Microsoft. Предоставляет возможность взаимодействия с объектами как посредством LINQ в виде LINQ to Entities, так и с использованием Entity SQL. Для облегчения построения web-решений используется как ADO.NET Data Services (Astoria), так и связка из Windows Communication Foundation и Windows Presentation Foundation, позволяющая строить многоуровневые приложения, реализуя один из шаблонов проектирования MVC, MVP или MVVM.

**Entity SQL** представляет собой язык, подобный языку [SQL](https://ru.wikipedia.org/wiki/SQL), который позволяет выполнять запросы к концептуальным моделям в Entity Framework.

**LINQ to Entities –** это альтернативный интерфейс LINQ API, используемый для обращения к базе данных. Он отделяет сущностную объектную модель данных от физической базы данных, вводя логическое отображение между ними. Так, например, схемы реляционных баз данных не всегда подходят для построения объектно-ориентированных приложений и в результате мы имеем объектную модель приложения, существенно отличающуюся от логической модели данных, в этом случае используется LINQ to Entities, который использует модель EDM (Entity Data Model). То есть, если вам нужно ослабить связь между вашей сущностной объектной моделью данных и физической моделью данных, например, если ваши сущностные объекты конструируются из нескольких таблиц или вам нужна большая гибкость в моделировании ваших сущностных объектов используйте LINQ to Entities

Релиз ADO.NET Entity Framework состоялся 11 августа 2008 года в составе .NET Framework 3.5 Service Pack 1 и Visual Studio 2008 Service Pack 1. В VS 2008 вошёл EDM Wizard для реверс-инжиниринга существующих баз данных и EDM Designer для редактирования сгенерированных моделей или создания их с нуля.

23 июня 2008 года, ещё до релиза первой версии, на стадии финальной доводки Entity Framework V1, начался процесс разработки Entity Framework V2.0. По словам англ. Tim Mallalieu, менеджера программы LINQ to SQL и EF, в .NET Framework 4.0 именно Entity Framework станет рекомендуемой технологией доступа к реляционным СУБД посредством LINQ.

12 апреля 2010 года в составе релиза Visual Studio 2010 и .NET Framework 4.0 был представлена Entity Framework 4.0. Позже уже отдельно от фреймворка были представлены версии: 4.1 (апрель 2011), 4.2 (октябрь 2011), 4.3 (февраль 2012).

11 августа 2012 года была представлена версия 5.0.0, которая была предназначена для .NET Framework 4.5. А 17 октября 2013 года была представлена версия 6.0, которая вышла под лицензией Apache License v2, тем самым став open-source проектом.

Версия 6.0 была выпущена 17 октября 2013 года и сейчас это проект с открытым исходным кодом под лицензией Apache License v2. В версии 6.0 был сделан ряд улучшений в поддержке метода работы Code First.

Изначально с самой первой версии Entity Framework поддерживал подход Database First, который позволял по готовой базе данных сгенерировать модель edmx. Затем эта модель использовалась для подключения к базе данных. Позже был добавлен подход Model First. Он позволял создать вручную с помощью визуального редактора модель edmx, и по ней создать базу данных. Начиная с 5.0 предпочтительным подходом становится Code First. Его суть - сначала пишется код модели на C#, а затем по нему генерируется база данных. При этом модель edmx уже не используется.

# ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

# 2.1 Разработка диаграммы ERD

Диаграмма — (ER-модель данных) обеспечивает стандартный способ определения данных и отношений между ними. Она включает сущности и взаимосвязи, отражающие основные бизнес-правила предметной области. Диаграммы «сущность— связь» в отличие от функциональных диаграмм определяют спецификации структур данных программного обеспечения.

Базовыми понятиями ER-модели данных (ER — Entity— Relationship) являются сущность, атрибут и связь.

Сущность — это класс однотипных реальных или абстрактных объектов (людей, событий, состояний, предметов и т.п.), информация о которых имеет существенное значение для рассматриваемой предметной области.

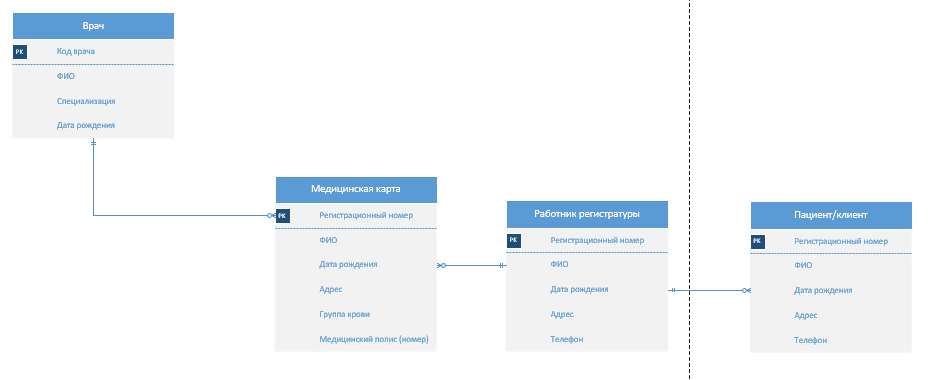
Атрибут — любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Атрибут, таким образом, представляет собой некоторый тип характеристик или свойств, ассоциированных с множеством реальных или абстрактных объектов.

Атрибуты делятся на ключевые, т. е. входящие в состав уникального идентификатора ключа, и описательные — прочие.

Первичный ключ — это атрибут или совокупность атрибутов и связей, предназначенная для уникальной идентификации каждого экземпляра сущности.

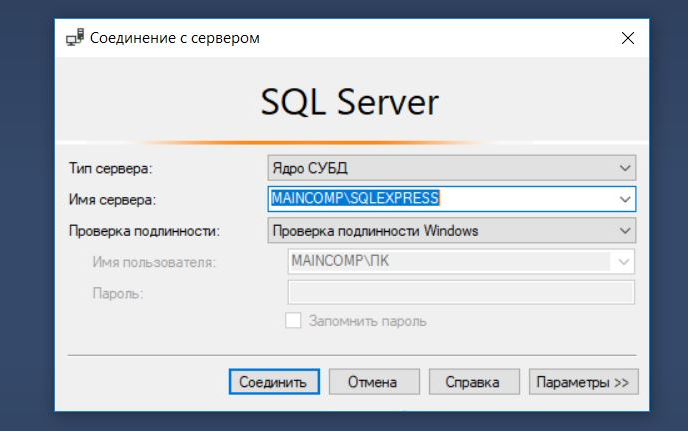
Описательные атрибуты могут быть обязательными или необязательными. Обязательные атрибуты для каждой сущности всегда имеют конкретное значение, необязательные могут быть не определены.

Связь — это отношение одной сущности к другой или к самой себе. Если любой экземпляр одной сущности связан хотя бы с одним экземпляром другой сущности, то связь является обязательной. Необязательная связь представляет собой условное отношение между сущностями. Каждая сущность может обладать любым количеством связей с другими сущностями модели. Различают три типа отношений «один-к-одному»; «один-ко-многим»; «многие-ко-многим».

*Рисунок 2.1 Диаграмма ER*

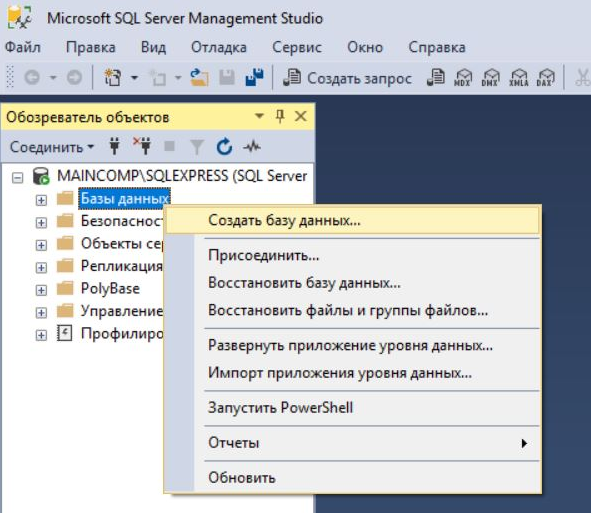
* Основные сущности для данной диаграммы это: клиент/пациент, врач, администратор и относящиеся сущности к системе. Второстепенными сущностями являются: работник регистратуры, медицинская карта. Возникают связи между сущностями — клиент, который может подать много заявок, а работник регистратуры один. Такое отношение “многие к одному”. Также у сущностей есть атрибуты, например у клиента есть ФИО, дата рождения и т.д. Каждый запрос содержит уникальный идентификатор (личный номер).

# 2.2 Разработка базы данных

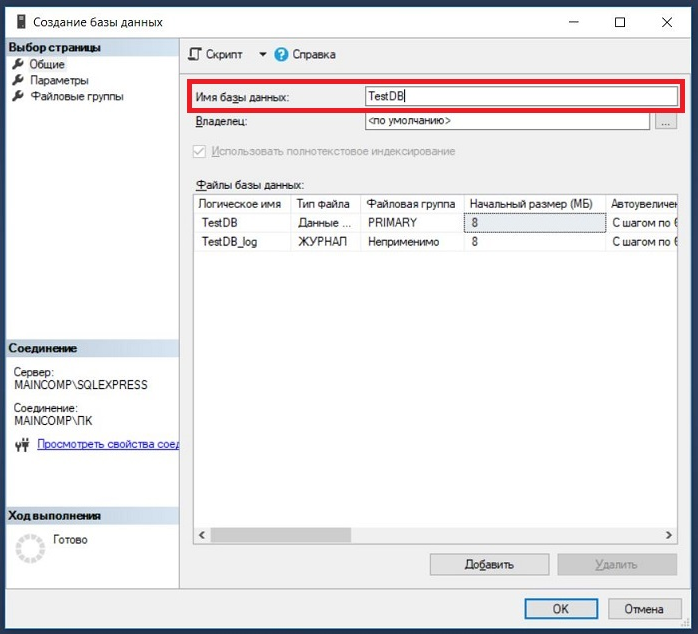
Первое, что нужно сделать, это запустить среду SQL Server Management Studio и подключиться к SQL серверу.

*Рисунок 2.2 Подключение к серверу базы данных*

Затем в обозревателе объектов щелкнуть по контейнеру «Базы данных» правой кнопкой мыши и выбрать пункт «Создать базу данных».

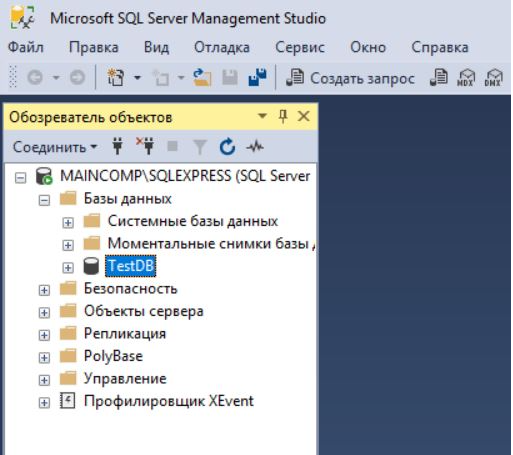
*Рисунок 2.3 Окно создание базы данных*

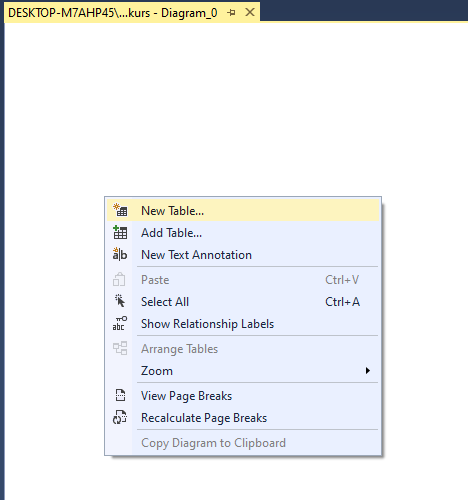
В результате откроется окно «Создание базы данных». Здесь обязательно нужно заполнить только поле «Имя базы данных», остальные параметры настраиваются по необходимости. После того, как Вы ввели имя БД, нажимайте «ОК».



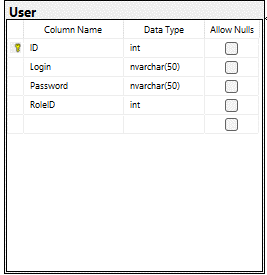
*Рис. 2.4 Окно создания базы данных*

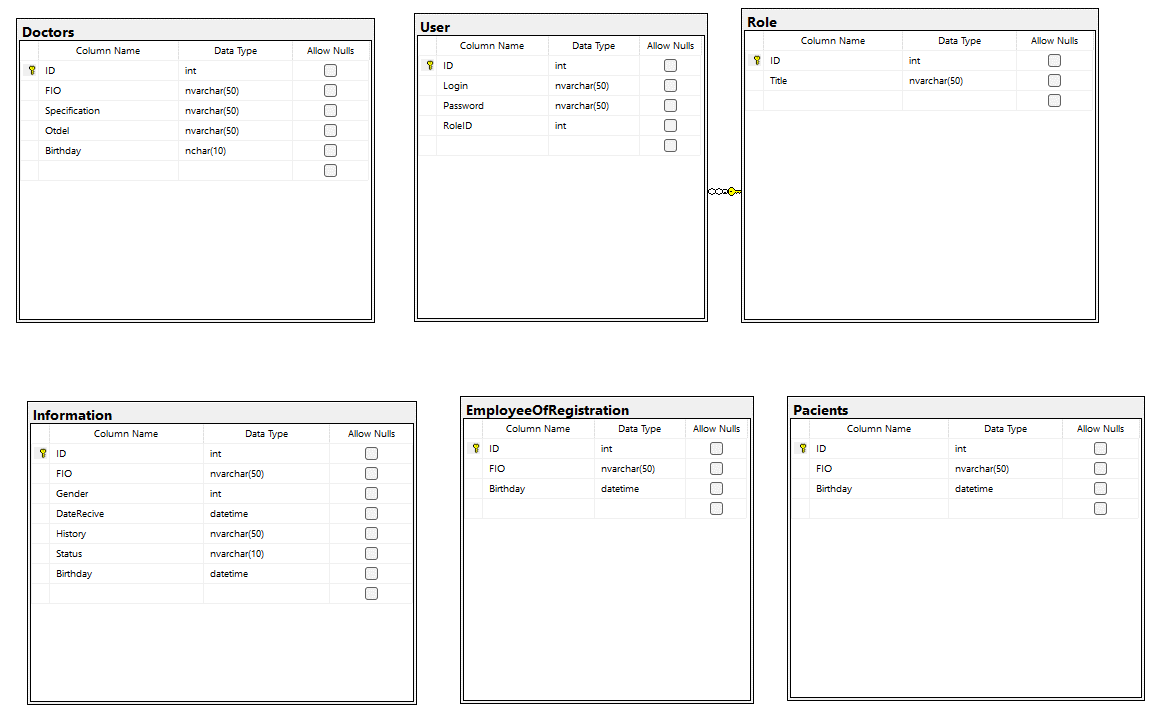
Если БД с таким именем на сервере еще нет, то она будет создана, в обозревателе объектов она сразу отобразится.



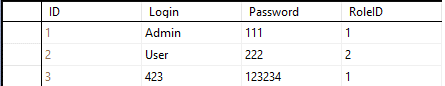
*Рисунок 2.5 Создание БД*

*Рисунок 2.6 создание таблицы в SSMS*

*Рис. 2.7 Созданная «сущность» пользователя в БД*



*Рис. 2.8 Созданная таблица информационной системы “поликлиника”*



*Рис. 2.9 Сущность “пользователь” базы данных с внесёнными данными*

# 2.3 Словарь данных

Словарь данных (Data Dictionary) – сервис, который предоставляет описание данных в бизнес-терминах, и дополнительно может содержать другие сведения о данных, например информацию о типах форматов данных, детализацию структур данных и нормативно-справочной информации, ограничений по безопасности. Таким образом, словарь данных является одним из способов ведения метаданных.

Словари данных связаны с хранилищем данных (Data Warehouses) или другими информационными ресурсами, описывая возможности их использования. Структура словаря данных, как правило, берет свое начало из логической модели данных информационных ресурсов. Словарь данных помогает организовать ведение метаданных с более высоким качеством по сравнению с неунифицированным документированием, благодаря дисциплинированному и систематическому подходу к управлению определениями и семантикой (смысловым содержанием) информационного пространства организации.

Несмотря на то, что инструменты типа “Словарь данных” имеют различный вид и формат, все они предоставляют компоненты и функции, необходимые для создания соответствующих описаний реляционных баз данных. Международная организация по стандартизации (ISO) выделяет 3 категории для элементов словаря данных:

1. Бизнес-концепты (Business Concepts): бизнес-компоненты с   семантическим (смысловым) значением: связанные с ними бизнес-ассоциации и функциональные роли. К примеру, организация должна определить термин “клиент”, описать характеристики в виде атрибутов и назначить ответственных за их ведение.
2. Типы данных (Data Types): технические характеристики допустимых значений определенного бизнес-элемента. Например, формат данных числовой, текстовый и др.
3. Концепты сообщений (Message Concepts): ограничения и элементы, которые делают сообщения уникальными. Например, данные они могут быть ограничены фиксированными промежутками и областью значений. Значения, находящиеся за пределами указанного диапазона, не будут приниматься базой данных.

Таблица 2.1 Словарь данных сущности пользователь

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **User** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ID** | INT | Y | ID пользователя |
|  | **Login** | NVARCHAR (50) | Y | Логин пользователя |
|  | **Password** | NVARCHAR (50) | Y | Пароль пользователя |
| FK | **RoleID** | INT | Y | Роль в системе |

Таблица 2.2 Словарь данных сущности роль

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Role** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| FK | **ID** | INT | Y | ID роли пользователя |
|  | **Title** | NVARCHAR (50) | Y | Описание роли пользователя |

Таблица 2.3 Словарь данных сущности информация

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Information** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| FK | **ID** | int | Y | Id клиента |
|  | **FIO** | NVARCHAR (50) | Y | ФИО клиента |
|  | **Gender** | int | Y | Пол, где 1 – м, 2 - ж |
|  | **DateRecive** | datetime | Y | Дата обращения |
|  | **History** | NVARCHAR (50) | Y | Краткая история обращений |
|  | **Status** | NVARCHAR (10) | Y | Статус клиента |
|  | **Birthday** | datetime | Y | День рождения клиента |

Таблица 2.4 Словарь данных сущности работник регистратуры

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Employee of registration** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| FK | **ID** | int | Y | Id работника регистратуры |
|  | **FIO** | NVARCHAR (50) | Y | ФИО работника регистратуры |
|  | **Birthday** | Datetime | Y | День рождения |

Таблица 2.5 Словарь данных сущности врачи

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Doctors** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ID** | int | Y | Id работника мед. персонала |
|  | **FIO** | NVARCHAR (50) | Y | ФИО врача |
|  | **Specification** | NVARCHAR (10) | Y | Спецификация врача |
|  | **Otdel** | NVARCHAR (50) | Y | Отдел работы врача |
|  | **Birthday** | Datetime | Y | День рождения врача |

Таблица 2.6 Словарь данных сущности пациенты

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Pacients** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **ID** | int | Y | Id пациента |
|  | **FIO** | NVARCHAR (50) | Y | ФИО пациента |
|  | **Birthday** | Datetime | Y | День рождения пациента |

# ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ

# 3.1 Введение в безопасность SQL Server

Введение в безопасность SQL Server включает в себя ряд важных мер, которые необходимо предпринять для защиты данных и обеспечения безопасности операций. Ниже приведены основные аспекты, которые обычно рассматриваются при обсуждении безопасности SQL Server.

1. Аутентификация и авторизация: SQL Server предоставляет средства аутентификации и авторизации для управления доступом к данным и функциональным возможностям сервера. Это включает управление пользователями, ролями, уровнями доступа и создание безопасных паролей.

2. Шифрование данных: SQL Server поддерживает шифрование данных в хранилище, используя различные методы, такие как прозрачное шифрование столбцов, Transparent Data Encryption (TDE) и шифрование резервных копий.

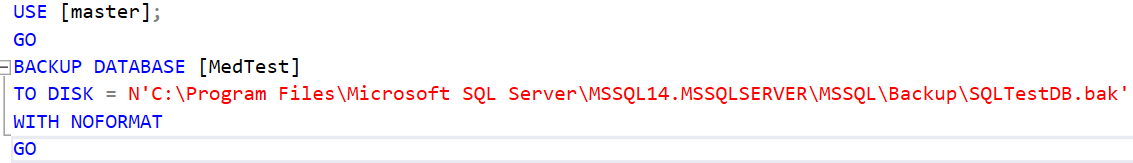
3. Аудит и мониторинг: SQL Server предоставляет средства для мониторинга и аудита доступа к данным и операциям с сервером. Это включает возможности отслеживания изменений, журналирования событий и анализа безопасности.

4. Управление ролями и полномочиями: В SQL Server можно определять различные роли и полномочия для пользователей и групп, что позволяет более точно управлять доступом к различным частям базы данных.

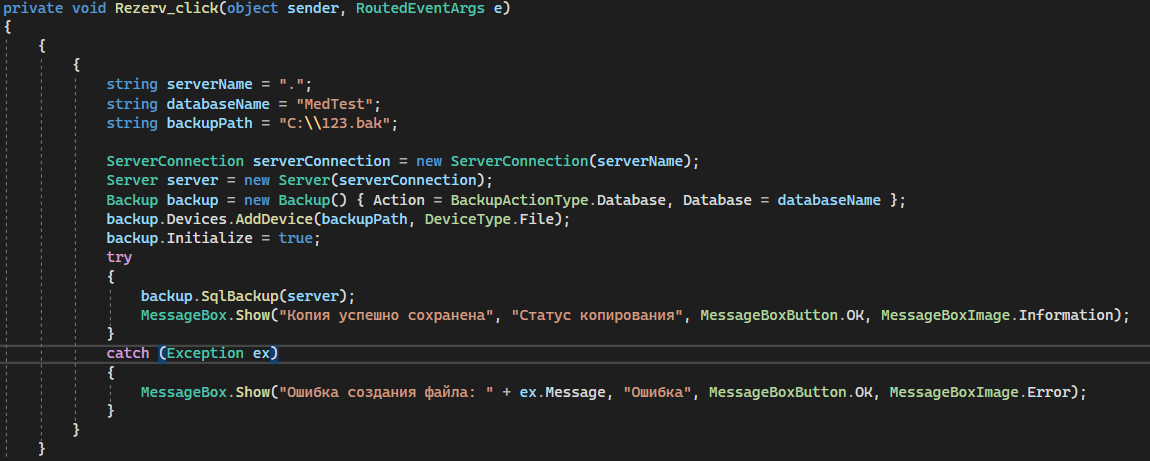
5. Защита от инъекций: SQL Server обеспечивает механизмы защиты от различных форм атак, таких как SQL-инъекции, предотвращая возможные уязвимости.

6. Обновление и патчи: Регулярное обновление и установка патчей являются важными аспектами обеспечения безопасности SQL Server, поскольку это помогает в устранении обнаруженных уязвимостей.

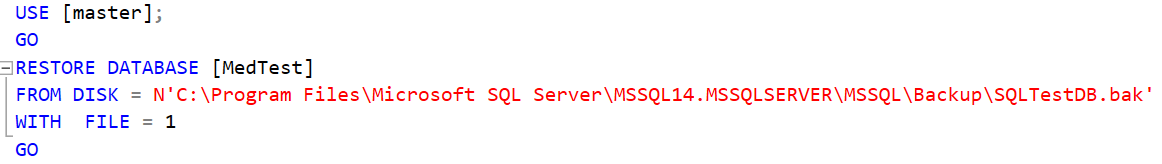
Введение в безопасность SQL Server состоит из умения применять вышеперечисленные меры, а также из понимания угроз безопасности и способов их предотвращения.

Для создания резервной копии базы данных необходимо использовать команду, изображённую на рисунке 3.1 или же использовать кнопку в программе, а код этой кнопки на рисунке 3.2.

*Рисунок 3.1 Создание резервной копии для базы данных*



*Рисунок 3.2 Код кнопки резервного копирования*

Также для восстановления резервной копии базы данных необходимо использовать команду, изображённую на рисунке 3.3 или же воспользоваться интерфейсом программы и выполнить восстановление через саму программу.

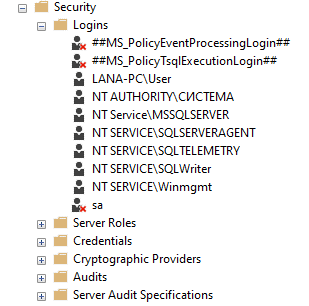
*Рисунок 3.3 Восстановление из резервной копии базы данных*

# 3.2 Управление безопасностью уровня сервера

Для управления безопасностью уровня сервера необходимо создать логины и пароли, которые позволят ограничить пользователей в доступе к базам данных и действиям их редактирования.

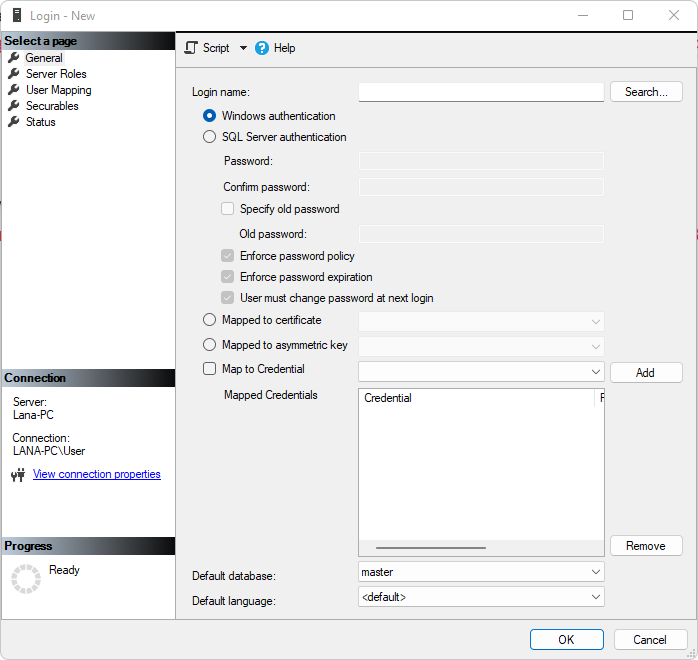
Чтобы создать пользователей глобально в программе SQL Server Management Studio (SSMS), необходимо выполнить следующие шаги:

1. Откройте SQL Server Management Studio: запустите программу SQL Server Management Studio и подключитесь к серверу баз данных, к которому вы хотите создать пользователя.

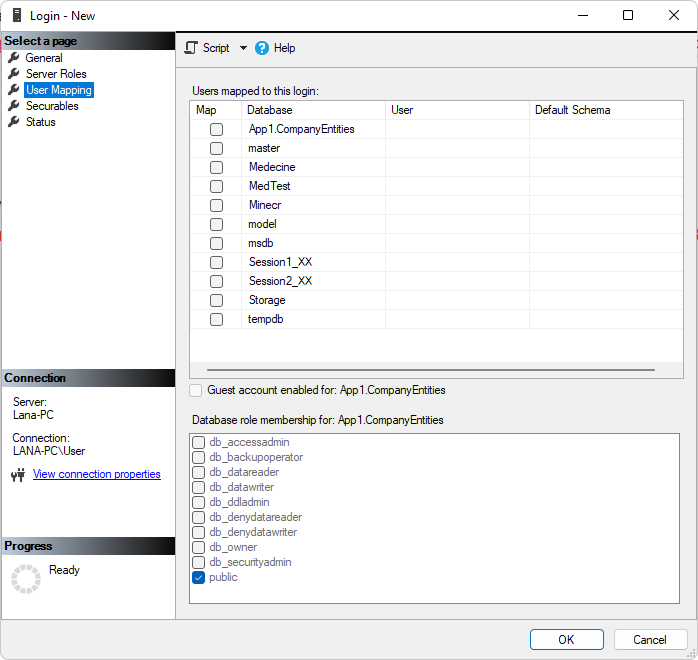
**2. Откройте папку безопасности: в обозревателе объектов раскройте папку "Безопасность" и перейдите к папке "Пользователи", изображено на рисунке 3.4.

*Рисунок 3.4 Открытие безопасности в SSMS*

3. Щелкните правой кнопкой мыши по папке "Пользователи" и выберите "Новый пользователь": это откроет всплывающее окно, в котором вы сможете указать имя нового пользователя, тип аутентификации (Windows или SQL Server), а также другие настройки, такие как разрешения и роль, изображено на рисунке 3.5.

 *Рисунок 3.5 Добавление пользователя*

4. Заполните данные пользователя: в появившемся окне укажите имя нового пользователя, его тип аутентификации (Windows или SQL Server), а также другие необходимые параметры.

5. Укажите разрешения и роль пользователя (при необходимости): выберите необходимые разрешения и роль для нового пользователя, учитывая требования безопасности и цели использования этого пользователя.

*Рисунок 3.6 Выдача пользователю прав на базу(ы) данных в системе*

6. Нажмите "ОК" для создания пользователя: после заполнения всех необходимых данных нажмите кнопку "ОК" для создания нового пользователя.

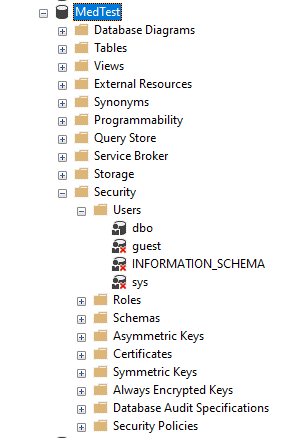
Этот процесс позволяет создавать пользователей глобально в SQL Server Management Studio, что позволяет управлять доступом пользователей к серверу и базам данных.

# 3.3 Управление участниками уровня базы данных

Для управления участниками уровня базы данных необходимо создать пользователей или группу пользователей в конкретной базе данных и дать им определённые возможности, в зависимости от их роли в базе данных.

Для создания пользователей локально для базы данных в программе SQL Server Management Studio (SSMS) необходимо выполнить следующие шаги:

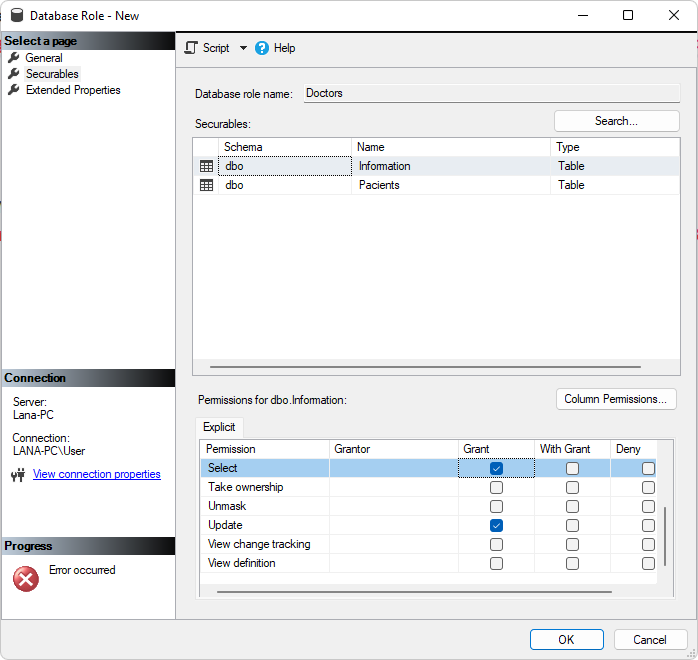
1. Откройте SQL Server Management Studio: запустите программу SQL Server Management Studio и подключитесь к серверу баз данных.

2. Раскройте папку вашей базы данных: в обозревателе объектов найдите свою базу данных, щелкните на нее правой кнопкой мыши и выберите "Свойства".

*Рисунок 3.7 Добавление нового пользователя локально для базы данных*

3. Перейдите к папке "Безопасность": в окне "Свойства базы данных" выберите папку "Безопасность".

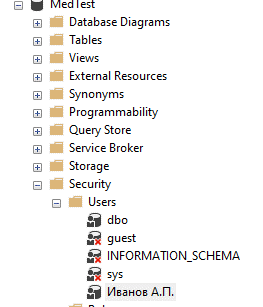
4. Перейдите к папке "Users": в папке security выберите папку "Users".

5. Щелкните правой кнопкой мыши по папке "Пользователи" и выберите "Новый пользователь": это откроет всплывающее окно, в котором вы сможете указать имя нового пользователя и его параметры.

*Рисунок 3.8 Выдача разрешений для пользователя*

6. Задайте параметры нового пользователя: укажите имя нового пользователя, его тип аутентификации (Windows или SQL Server), а также другие параметры, такие как разрешения и роль.

7. Укажите разрешения и роль пользователя (при необходимости): выберите необходимые разрешения и роль для нового пользователя, учитывая требования безопасности и цели использования этого пользователя.

8. Нажмите "ОК" для создания пользователя: после заполнения всех необходимых данных нажмите кнопку "ОК" для создания нового пользователя в вашей базе данных.

*Рисунок 3.9 Пример созданного пользователя с его ограничениями*

Этот процесс позволяет создавать пользователей локально для конкретной базы данных в SQL Server Management Studio, обеспечивая более тесное управление доступом пользователей к базе данных и ее объектам.

# 3.4 Управление разрешениями уровня базы данных

Управление разрешениями уровня базы данных осуществляется путём выдачи разрешений пользователям на какие – либо действия над таблицами в базе данных.

Для создания разрешений для локальных пользователей базы данных в SQL Server Management Studio (SSMS) выполните следующие шаги:

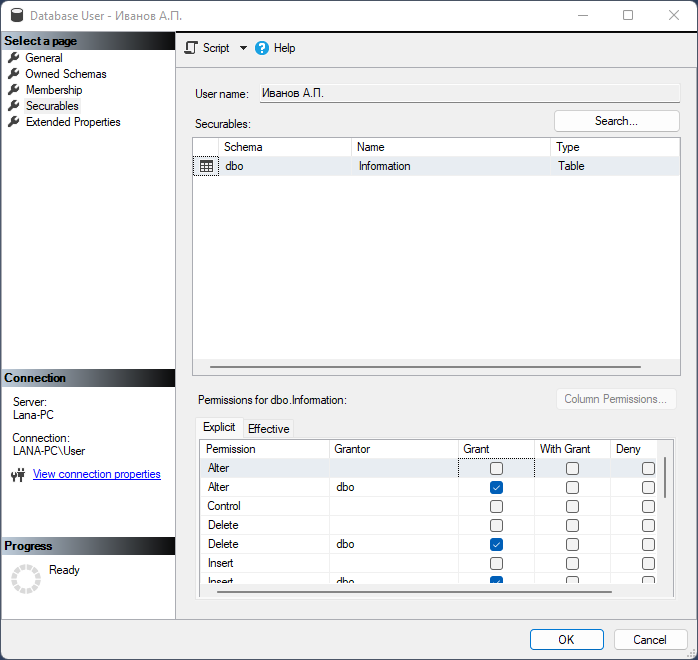
1. Откройте SQL Server Management Studio: запустите программу SQL Server Management Studio и подключитесь к серверу баз данных.

2. Раскройте папку вашей базы данных: в обозревателе объектов найдите вашу базу данных, щелкните на нее правой кнопкой мыши и выберите "Свойства".

3. Перейдите к папке "Безопасность": в окне "Свойства базы данных" выберите папку "Безопасность".

4. Выберите "Роли базы данных" или "Схемы": в зависимости от того, какие разрешения вы хотите предоставить, выберите либо "Роли базы данных", либо "Схемы", и выберите нужный элемент.

5. Щелкните правой кнопкой мыши по выбранному элементу и выберите "Новая роль" или "Новая схема": это откроет окно создания новой роли или схемы.

6. Задайте параметры для новой роли или схемы: укажите имя новой роли или схемы, затем установите соответствующие разрешения для этой роли или схемы, такие как SELECT, INSERT, UPDATE, DELETE и т.д.

*Рисунок 3.10 Изменение разрешений для пользователя базы данных*

7. Добавьте локальных пользователей в созданную роль или схему: после создания роли или схемы, добавьте нужных локальных пользователей базы данных в эту роль или схему, чтобы предоставить им соответствующие разрешения.

Этот процесс позволяет создавать и управлять разрешениями для локальных пользователей базы данных в SQL Server Management Studio, обеспечивая более гранулированный контроль над доступом к данным и объектам базы данных.

# 3.5 Шифрование данных баз данных

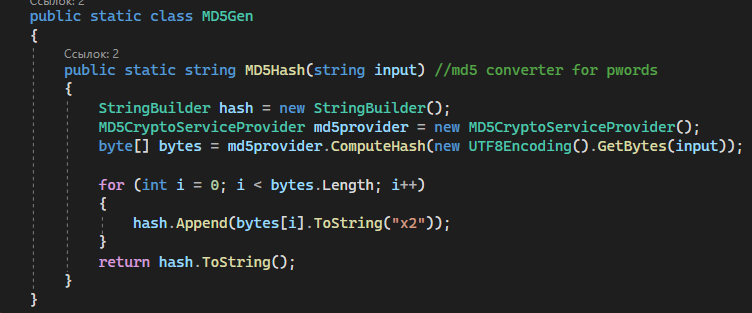
Шифрование данных в базе данных используется для защиты данных, которые могут украсть

Для создания шифрования данных в базе данных с помощью SQL Server Management Studio (SSMS) следует выполнить следующие шаги:

1. Зашифровать в самой базе данных в SQL Server Management Studio (SSMS) нужные поля с помощью команды, показанной на рисунке 3.11.



*Рисунок 3.11 Шифрование в базе данных*

1. В программе создать новый класс, в который вставить код, изображённый на рисунке 3.12.

*Рисунок 3.12 Класс шифрования в программе*

1. В программе в окне авторизации добавить строки, отсылающие на созданный ранее класс для расшифровки зашифрованных полей в базе данных, изображено на рисунке 3.13.

*Рисунок 3.13 Использование класса в программе*

Эти шаги позволяют создать и применить шифрование данных в базе данных, обеспечивая защиту данных от несанкционированного доступа.

# ЗАКЛЮЧЕНИЕ

По итогу проделанной работы можно сказать, что получилось хорошая база данных для информационной системы «Поликлиника», которая обеспечивает пользователю удобство при работе с приложением и в самой базе данных.

Поставленные цели и задачи при проектировании информационной системы были выполнены в полной мере, а именно выполнены следующие задачи:

* создана база данных информационной системы “поликлиника” для регистрации клиентов/больных в базу данных поликлиники;
* созданы пользователи для работы с базой данных;
* реализовано создание резервных копий для базы данных;
* сформирована физическая структура базы данных;
* реализовано шифрование базы данных.

Работая над курсовым проектом, были сделаны следующие основные выводы, что каждая база данных должна иметь:

* идейную ценность, в виде решения какой-либо проблемы;
* представление о том, как будет выглядеть конечный продукт;
* технические требования к реализации;
* план разработки, в котором описаны все ступени разработки проекта;
* основные инструменты разработки;

В первой части были рассмотрены:

* инструменты разработки базы данных для автоматизированной системы;
* инструменты внедрения базы данных в приложение;

Во второй части были рассмотрены:

* диаграмма ER – сущность связь;
* процесс разработки базы данных по модели диаграммы ER;

В третьей части были рассмотрены:

* процесс создания резервных копий базы данных;
* процесс создания пользователей для базы данных;
* процесс создания шифрования для базы данных;

Ссылка на репозиторий на GitHub: https://github.com/Kot-Patriot/Kursovoy\_project2-Raykov

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Эндрю Троелсен. "Язык программирования C#. 5.0 и платформа .NET 4.5": учебник для вузов/ Эндрю Троелсен. - Питер, 2021. – С. 1250-1270.
2. Уильям Р. Станек. "SQL Server 2019. Базовый курс": книга по программированию/ Уильям Р. Станек. - Питер, 2020. – С. 230-340.
3. Microsoft Tutors // Помощь по ошибкам и инструкции по написанию кода от “Майкрософт”. [Электронный ресурс] – Режим доступа: [https://learn.microsoft.com/en-us/dotnet/tutorials/c#](https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/tutorials/%20)
4. SQL Server Management Studio (SSMS) // работа с базами данных от “Майкрософт”. [Электронный ресурс] – Режим доступа: [https://learn.microsoft.com/en-us/sql/ssms](https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16%20)
5. Visual Studio IDE // среда разработки от “Майкрософт”. [Электронный ресурс] – Режим доступа: <https://visualstudio.microsoft.com/ru/visualstudio> (дата обращения 23.10.2022)
6. Wikipedia C# // Объяснение принципов разработки на языке программирования C#. [Электронный ресурс] – Режим доступа: <https://en.wikipedia.org/wiki/C#/help>
7. Wikipedia SQL Server // Помощь и объяснение с примерами работы с базами данных с помощью Microsoft SQL Server. [Электронный ресурс] – Режим доступа: [https://en.wikipedia.org/wiki/SQLServer](https://en.wikipedia.org/wiki/Microsoft_SQL_Server%20)
8. w3schools // Подробные учебные материалы по C# и SQL. [Электронный ресурс] – Режим доступа: <https://www.w3schools.com/>
9. Microsoft Learn // Официальный сайт Microsoft с бесплатными учебными материалами по C# и SQL Server. [Электронный ресурс] – Режим доступа: <https://learn.microsoft.com/>
10. Tutorialspoint // Бесплатные онлайн-уроки по C# и SQL Server. [Электронный ресурс] – Режим доступа: https://www.tutorialspoint.com/