**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 «Архив»**

Выполнил студент гр.ИСП20/\_\_\_\_\_\_\_\_\_\_\_\_\_/ Черчел Анна Алексеевна/

(подпись) (Ф.И.О.)

**ОЦЕНКА:\_\_\_\_\_\_\_\_\_\_\_\_\_**

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

**ПРОВЕРИЛ**

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

(подпись) (Ф.И.О.)

Воркута

2023

# **СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc151119728)

[ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ 5](#_Toc151119729)

[1.1 СУБД SQL server 5](#_Toc151119730)

[1.2 Microsoft SQL Server Management Studio 6](#_Toc151119731)

[1.3 Entity Framework 7](#_Toc151119732)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 10](#_Toc151119733)

[2.1 Разработка диаграммы ERD 10](#_Toc151119734)

[2.2 Разработка базы данных 12](#_Toc151119735)

[2.3 Словарь данных 15](#_Toc151119736)

[ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ 18](#_Toc151119737)

[3.1 Введение в безопасность sql server 18](#_Toc151119738)

[3.2 Управление безопасностью уровня сервера 19](#_Toc151119739)

[3.3 Управление участниками уровня базы данных 20](#_Toc151119740)

[3.4 Управление разрешениями уровня базы данных 22](#_Toc151119741)

[3.5 Шифрование данных баз данных 23](#_Toc151119742)

[ЗАКЛЮЧЕНИЕ 25](#_Toc151119743)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 26](#_Toc151119744)

**ВВЕДЕНИЕ**

Постоянный рост объемов информации, многообразие и противоречивость позволяет автоматизировать процесс управления документооборотом и делопроизводством в организации. Информационная система дает возможность создать единую базу документов, категоризировать их, настроить поиск, оптимизировать учет и контроль сроков, задач, поручений, повысить эффективность.

Основная идея автоматизации архивов состоит в том, что таким образом, отдавая документы в архив, вы заботитесь о том, чтобы компания заняла надлежащее место в финансово-государственном обществе. Архивное хранение документов гарантирует, что компания всегда сможет ответить на запросы государства, что она не будет создавать помехи в более высоких системах документооборота, а также тот факт, что в случае, если документы понадобятся частному лицу или представителю той или иной организации, не будет возникать масса ненужных спорных ситуаций. Современные технологии применяются и в архивном хранении. Информация дублируется и сохраняется в электронном виде, что позволяет легче хранить и добывать информацию.

**Объект:** информационно-справочная система «Архив».

**Предмет:** автоматизация архивной организации.

**Цель работы:** разработать базу данных для информационной системы «Архив».

**Задачи:**

* Проведение предпроектного обследования (изучение предметной области предприятия, описание основных бизнес-процессов на предприятии);
* проектирование информационной системы;
* разработать базу данных;
* сделать разграничение прав доступа базы данных;
* делать шифрование данных.

**ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ**

**1.1 СУБД SQL server**

Microsoft SQL Server — система управления реляционными базами данных (РСУБД), разработанная корпорацией Microsoft. Основной используемый язык запросов — Transact-SQL, создан совместно Microsoft и Sybase. Transact-SQL является реализацией стандарта ANSI/ISO по структурированному языку запросов (SQL) с расширениями. Используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

SQL Server 2012 (2010—2012)

Основная статья: Microsoft SQL Server 2012

См. также: Microsoft SQL Server Compact 4.0

Разработка

Параллельно с этим, 30 июня 2010 года Скотт Гатри в своём блоге анонсировал выход новой мобильной версии SQL Server — SQL Server Compact 4.0, ориентированной в первую очередь на веб-приложения, созданные на основе технологии ASP.NET 4. В качестве достоинств новой версии Гатри выделил отсутствие необходимости в установке программы, а также совместимость с API .NET Framework (поддержка технологий ADO.NET, Entity Framework, NHibernate и др., возможность работы с Visual Studio 2010 и Visual Web Developer 2010 Express) и последними на тот момент версиями SQL Server и SQL Azure.

7 июля 2010 года Амбриш Мишра, менеджер проекта SQL Server Compact, в официальном блоге команды разработчиков SQL Server CE представил версию CTP1 нового SQL Server Compact 4.0. В качестве нововведений (помимо указанных Скоттом Гатри) указывались повышенная надёжность, улучшение алгоритма шифрования SHA 2, совместимость с файлами БД версии Compact 3.5, упрощение установки (в том числе и поддержка режимов WOW64 и 64-битных естественных приложений), снижение использования виртуальной памяти, технология делегирования полномочий Allow Partially Trusted Caller’s Attribute (APTCA), поддержка WebMatrix Beta и Visual Studio 2010, поддержка Paging Queries в языке T-SQL. При этом версия CTP1 обладала определёнными проблемами (некорректная работа деинсталляции через командную строку, проблемы с совместимостью с актуальной на тот момент версией ADO.NET Entity Framework CTP3 и др.).

Бета-версии

Во время конференции PASS Summit, проходившей с 8 по 11 ноября 2010 года в Сиэтле, её участникам (а также подписчикам MSDN и TechNet) раздавались копии CTP-варианта Denali (через некоторое время эта версия была выложена на официальном сайте Microsoft). На самой конференции Тед Каммерт и Квентин Кларк, генеральный менеджер подразделения Microsoft Database Systems Group, представляя новую версию, рассказали о новой функции AlwaysOn и технологии VertiPac (входящей в состав служб аналитики SQL Server и хранилищ данных). Также упор был сделан на развитие инструментов бизнес-аналитики в новой версии, интерактивных средств виртуализации на базе Web (проект Crescent), а также инструментов для разработчиков под кодовым названием Juneau.

22 декабря 2010 года Амбриш Мишра в официальном блоге команды разработчиков объявил о выходе версии SQL Server Compact 4.0 CTP2 и наборе инструментов Visual Studio 2010 для работы с этой версией SQL Sever CE.

Финальная версия SQL Server Compact 4.0 официально была выложена на сайте Microsoft 12 января 2011 года, завершив таким образом этап разработки длиной примерно год.

11 июля 2011 года команда разработчиков SQL Server в своём официальном блоге объявила о выходе версии Community Technology Preview 3 (сокр. CTP3) и первом сервис-паке для SQL Server 2008 R2[21]. В качестве самых существенных нововведений (относительно SQL Server 2008 R2), реализованных в CTP3-версии нового продукта, аналитики отмечали компонент SQL Server AlwaysOn для создания резервных копий БД, возможность устанавливать SQL Server в среде Windows Server Core, столбцовую организацию хранения данных для ускорения выполнения запросов, усовершенствования языка T-SQL (введение объектов Sequence и оконных функций), возможность отслеживать изменение данных (CDC) для СУБД Oracle, возможность определения пользователем ролей сервера (ранее они были жёстко закреплены), службы управления качеством данных Data Quality Services (базы знаний, определяющие правила метаданных), новый инструмент визуализации данных под названием «Проект Crescent», поддержку автономных баз данных (для перемещения между локальными экземплярами SQL Server и SQL Azure) и новую среду разработки SQL Server Developer Tools под кодовым названием Juneau.

SQL Server 2008 R2 SP1 содержал исправления ошибок, на которые Microsoft получила жалобы от клиентов через службу Windows Error Reporting, а также некоторые улучшения функциональности (динамические административные представления (англ. Dynamic Management Views), повышение скорости выполнения запросов при помощи технологии ForceSeek, технология Data-tier Application Component Framework (сокр. DAC Fx) для упрощения обновления БД, контроль доступного места на жёстком диске для PowerPivot).

SQL Server 2014

В конце 2010 года (то есть до выхода SQL Server 2012) вице-президент подразделения Microsoft Business Platform Division Тед Каммерт в интервью рассказал о планах по дальнейшему развитию продукта (как версии SQL Server 2012, так и будущих версий). В частности, Каммерт рассказал о том, что работа над SQL Server идёт в контексте идей Information Platform Vision, представляющей собой набор разнообразных возможностей, который ложится в основу платформы. SQL Server будет по-прежнему представлять собой единый продукт, реализуемый в настольных системах, в центрах обработки данных и в «облаке» (как в 32-битном, так и 64-битном варианте). Одним из приоритетных направлений по-прежнему останется бизнес-аналитика (англ. business intelligence, BI). С точки зрения Microsoft приоритетом в области бизнес-аналитики останется разработка средств BI, реализующих принцип самообслуживания, а также развитие экосистемы «облачных» вычислений. Кроме того, Microsoft при переносе средств бизнес-аналитики в «облака» работает над реализацией принципа согласованности относительно реализуемых моделей программирования и инструментальных средств (это подразумевает, в частности, наращивание возможностей работы SQL Server Management Studio со средой SQL Azure). Большое внимание уделено вопросам масштабирования СУБД (при этом лимит системы SQL Server должен быть увеличен до порога в несколько сот терабайтов), виртуализации приложений в баз данных, а также пространственному представлению данных.

**1.5 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.

Немного теории: о СУБД и БД

Само по себе представление о БД (базах данных) различается с учётом того, какая это точка зрения — пользовательская либо системная. В первом случае под БД понимается некая таблица с нужными данными (информацией), отображённая на мониторе либо даже листке бумаги. Во втором случае мы имеем системную, то есть администраторскую точку зрения, исходя из которой БД — это набор файлов, в которых реализованы таблицы и связи между таблицами, и к которым надо организовать доступ со стороны пользователей.

В настоящее время этот самый доступ организуется централизованно. В основе современной технологии лежит взаимодействие сервера и клиента. Клиенты — персональные компьютеры пользователей, серверы баз данных — мощные и высокопроизводительные компьютеры, обеспечивающие одновременный доступ к данным сотен и тысяч пользователей.

Сервер предоставляет данные, а клиенты их запрашивают, что происходит путём отправки клиентом SQL-запроса серверу. Сервер, в свою очередь, выполняет обработку запроса, возвращая запрошенные данные клиенту. Т. к. запрос реализуется на специальном языке SQL, он хорошо понятен и серверу, и пользователю.

Тут следует заметить, что серверы бывают разные и различаются тем, как они:

• предоставляют доступ;

• хранят данные; • предоставляют данные пользователю.

Соответственно, для каждой из вышеперечисленных целей существует отдельный программный компонент:

• прикладные программы — предоставляют доступ и обеспечивают визуализацию; • интерфейс — служит для предоставления данных;

• программы взаимодействия с файловой системой — обеспечивают хранение данных.

И вот как раз для взаимодействия программных компонентов и необходим ещё один компонент, называемый СУБД, то есть речь идёт о системе управления базами данных. По сути, СУБД представляет собой программную прослойку между сервером и пользователем. СУБД даёт возможность абстрагировать пользователя от системного видения базы данных, самой же системе предоставляется способ непосредственного взаимодействия с пользователем.

Задачи СУБД:

• извлечение данных с диска и сохранение данных на диск;

• работа с данными в оперативной памяти сервера;

• логирование (журналирование, запись истории изменения БД);

• поддержка запросов пользователя в целях определения и манипуляции данными на структурированном языке SQL.

Вышеописанные задачи по взаимодействию с БД решаются с помощью разной реализации компонентов, в результате чего различаются и СУБД, которые бывают иерархические, сетевые, реляционные, объектно-реляционные, объектно-ориентированные.

Кроме того, по способу доступа к БД системы бывают клиент-серверные, файл-серверные и встраиваемые.

Что касается СУБД SQL Server, то она является реляционной и клиент-серверной. Эта СУБД довольно популярна на системах, построенных на базе ОС Windows. Давайте установим и настроим её, взяв для примера SQL Server 2016 Enterprise.

Устанавливаем ПО для работы

В первую очередь, нужно скачать установочный пакет SQL Server 2016 Enterprise, перейдя на официальную страницу. Скачанная вами версия будет бесплатна в течении 180 дней. Также, если вы имеете подписку MSDN, можете использовать SQL Server 2016 Developer Edition.

Итак, софт скачан. Однако прежде, чем запускать установщик, надо создать учётную запись — она позволит авторизоваться на сервере с клиентского ПК. Так как в нашем случае это будет один и тот же компьютер, SQL Server будет авторизовываться через Management Studio, однако его скачаем несколько позже.

**1.6 Entity Framework**

Entity Framework — это решение для работы с базами данных, которое используется в программировании на языках семейства .NET. Оно позволяет взаимодействовать с СУБД с помощью сущностей (entity), а не таблиц. Также код с использованием EF пишется гораздо быстрее.

Например, работая с базами данных напрямую, разработчик должен беспокоиться о подключении, подготовке SQL и параметров, отправке запросов и транзакций. На Entity Framework всё это делается автоматически — программист же работает непосредственно с сущностями и только говорит EF, что нужно сохранить изменения.

В этой статье будут разобраны основы применения Entity Framework, для понимания которых нужно владеть ADO.NET — пользоваться базами данных, писать SQL-запросы и работать с подключениями.

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.

История

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.[1] 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.

Entity SQL

Entity SQL представляет собой язык, подобный языку SQL, который позволяет выполнять запросы к концептуальным моделям в Entity Framework.

LINQ to Entities

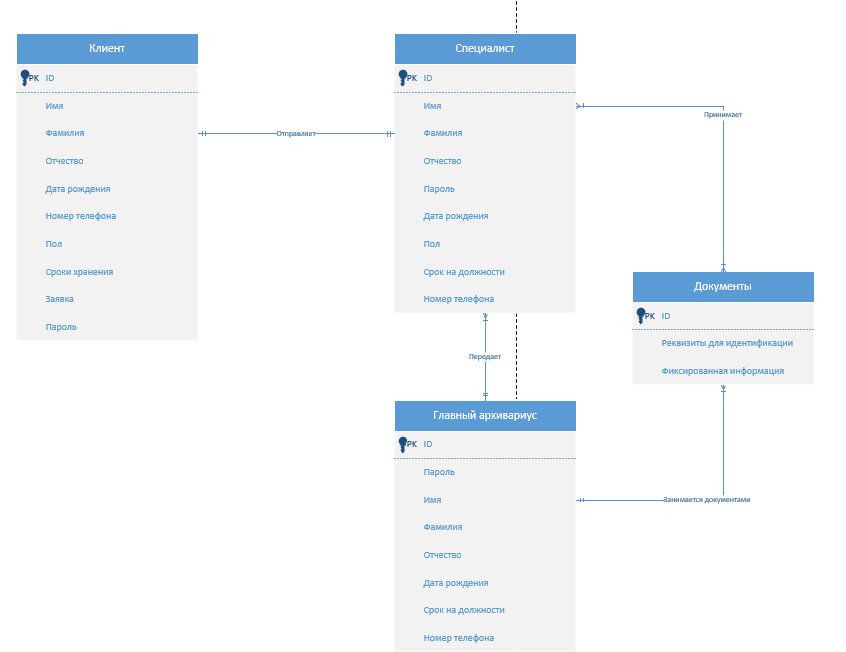
Это альтернативный интерфейс LINQ API, используемый для обращения к базе данных. Он отделяет сущностную объектную модель данных от физической базы данных, вводя логическое отображение между ними. Так, например, схемы реляционных баз данных не всегда подходят для построения объектно-ориентированных приложений 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 Архивная копия от 27 октября 2009 на Wayback Machine.

Подходы в EF

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**



*Рисунок 2.1 Диаграмма ER*

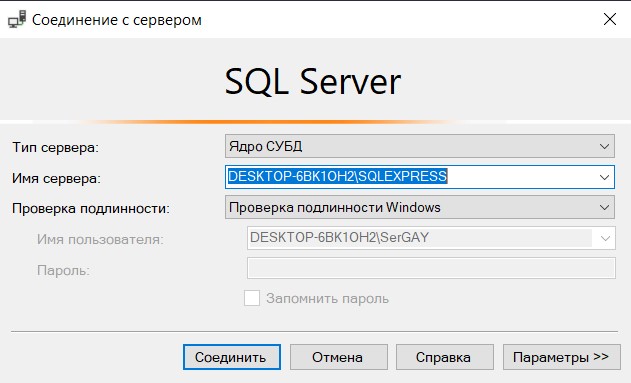
ER-модель — модель данных, позволяющая описывать концептуальные схемы предметной области. ER-модель используется при высокоуровневом проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

В данной модели диаграммы ERD представлены четыре сущности: Клиент, специалист, главный архивариус и документы.

Один клиент может обратиться к одному специалисту, один и более специалист могут передать запрос одному главному архивариусу, один главный архивариус занимается одним и более документом, один и более документ передается одному или более специалисту.

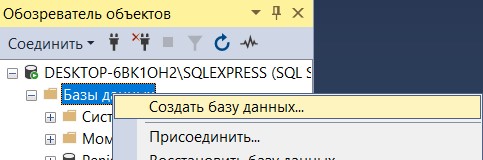
**2.2 Разработка базы данных**

Что бы создать базу данных, нужно запустить “Microsoft SQL Server Management Studio” и выполнить подключение к серверу.

****

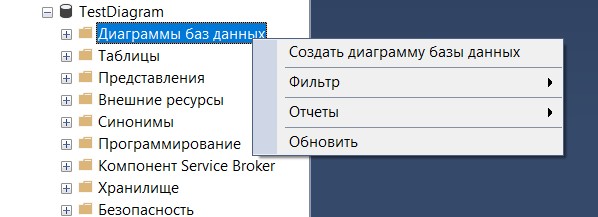
*Рисунок 2.2 Подключение к серверу*

Следующим шагом будет нажатие “Обозреватель объектов” на панели в левой части экрана, нажимаем правой кнопкой мыши по “Базы данных” и выбираем “Создать базу данных”.

****

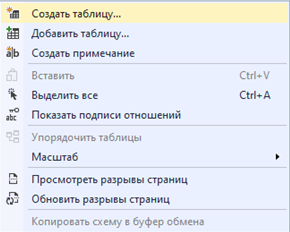
*Рисунок 2.3 Создание БД*

В обозревателе объектов появилась созданная нами база данных, открываем её и нажимаем на “Диаграммы баз данных” и выбираем “Создать диаграмму базы данных”.



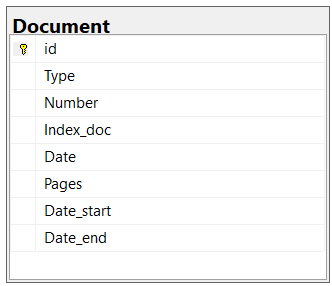
*Рисунок 2.4 Создание диаграммы*

Для того что бы создавать таблицы нашей базы данных, нужно нажать правой кнопкой мыши по пустому месту нашей диаграммы и выбрать пункт “Создать таблицу” и вводим название нашей таблицы.

****

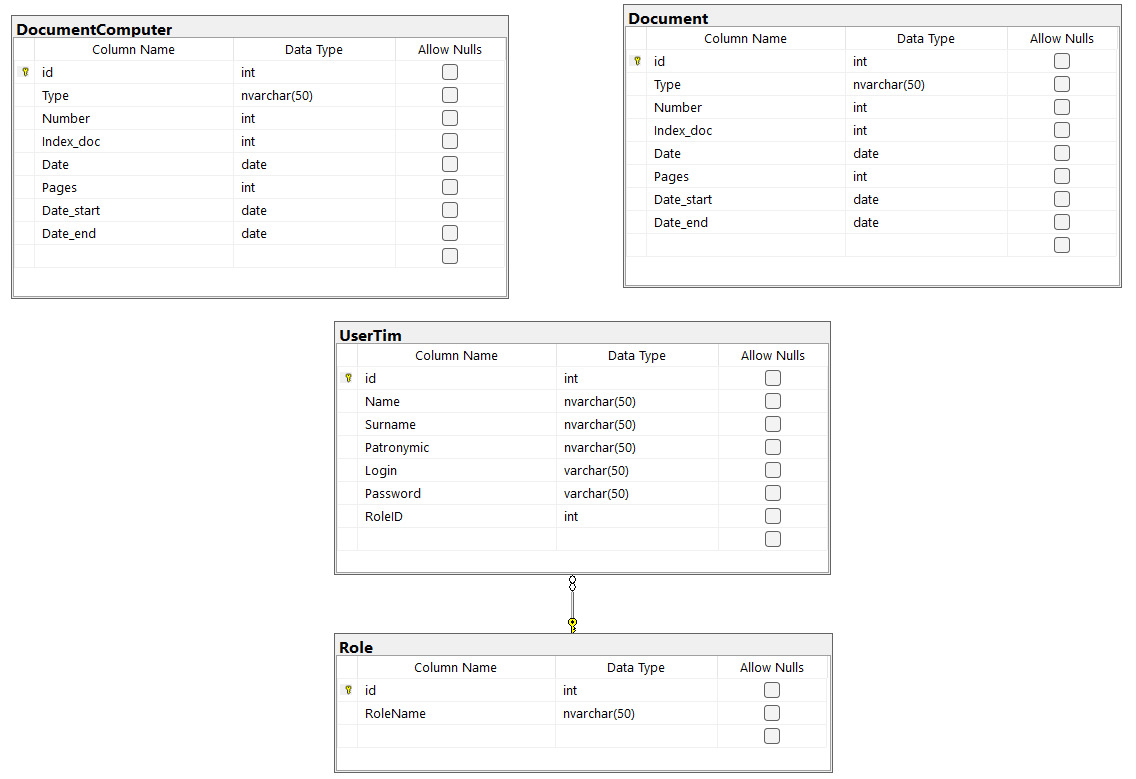
*Рисунок 2.5 Создание таблиц*

Таблица создалась, в которую мы можем писать атрибуты и уникальные идентификаторы. Вводим атрибуты и присваиваем им типы данных.



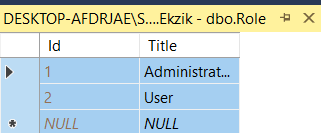
*Рисунок 2.6 Типы данных и атрибуты*

После создания и заполнения таблиц, соединяем их связями. Для этого зажимаем ключик и ведём к нужному полю в другой таблице.



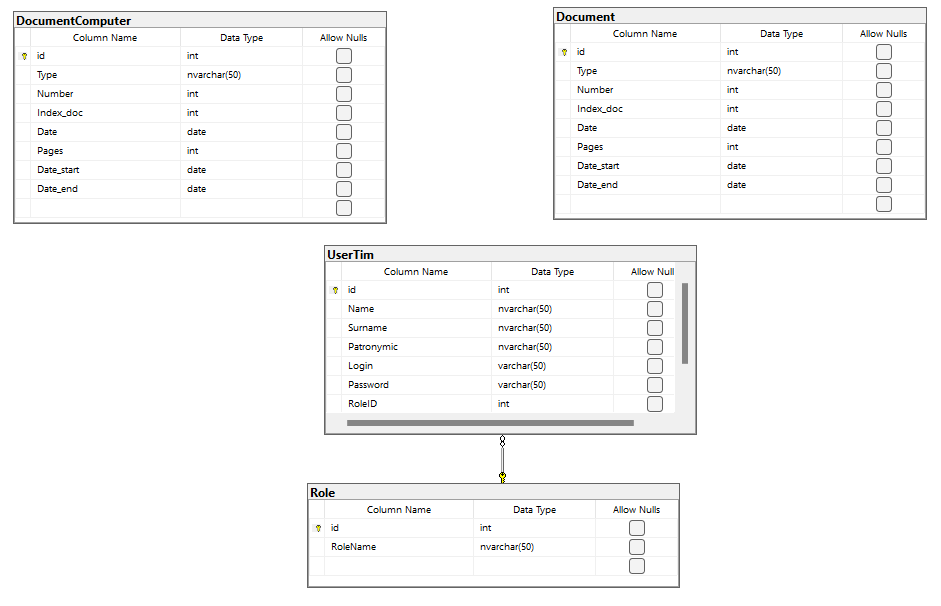
*Рисунок 2.7 Связь*

Последним шагом будет сохранить диаграмму и выбрать во вкладке “таблицы” табли чку и нажать 200 rows.

****

*Рисунок 2.8 Заполнение таблиц*

Финальная база данных ИС выглядит следующим образом:



*Рисунок 2.9 Финальная БД*

**2.3 Словарь данных**

Словарь данных представляет собой подсистему банка данных, предназначенную для централизованного хранения информации о структурах данных, взаимосвязях файлов БД друг с другом, типах данных и форматах их представления, принадлежности данных пользователям, когда защиты и разграничения доступа.

Одно из главных назначений словаря данных состоит в документировании данных. Так как база данных обслуживает многих пользователей, крайне необходимо, чтобы они правильно понимали, что представляют собой данные.

Таблица 1

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data Dictionary | | | | | |
| **UserTim** | | | | | |
| **KEY** | | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PRIMARY | | **ID** | int | Y | Уникальный идентификатор |
| FOREIGN | | **Name** | nvarchar(50) | Y | Имя пользователя |
|  | | **Surname** | nvarchar(50) | Y | Фамилия пользователя |
|  | | **Patronymic** | nvarchar(50) | Y | Отчество пользователя |
|  | | **Login** | varchar(50) | Y | Логин |
|  | | **Password** | varchar(50) | Y | Пароль |
|  | | **RoleID** | int | Y | Уникальный идентификатор роли |
| **Role** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PRIMARY | **ID** | int | Y | Уникальный идентификатор |
|  | **Name** | nvarchar(50) | Y | Название роли |

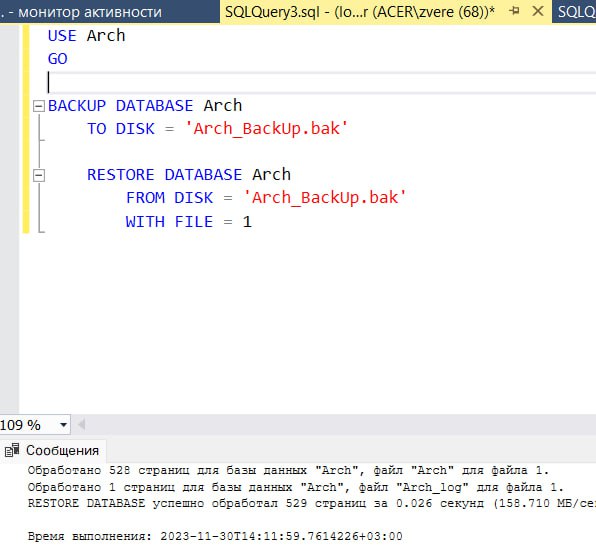
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PRIMARY | **ID** | int | Y | Уникальный идентификатор |
| FOREIGN | **Type** | nvarchar(50) | Y | Тип документа |
|  | **Number** | int | Y | Номер документа |
|  | **Index\_doc** | int | Y | Индекс документа |
|  | **Date** | date | Y | Дата |
|  | **Pages** | int | Y | Страниц в документе |
|  | **Date\_start** | date | Y | Дата начала заявки |
|  | **Date\_end** | date | Y | Дата окончания заявки |
| **DocumentComputer** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PRIMARY | **ID** | int | Y | Уникальный идентификатор |
| FOREIGN | **Type** | nvarchar(50) | Y | Тип документа |
|  | **Number** | int | Y | Номер документа |
|  | **Index\_doc** | int | Y | Индекс документа |
|  | **Date** | date | Y | Дата |
|  | **Pages** | int | Y | Страниц в документе |
|  | **Date\_start** | date | Y | Дата начала заявки |
|  | **Date\_end** | date | Y | Дата окончания заявки |

**ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ**

**3.1 Введение в безопасность sql server**

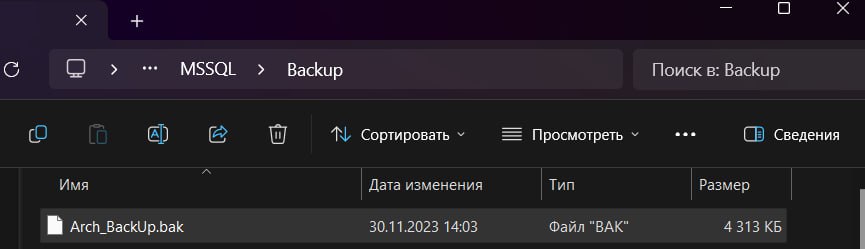
Безопасность SQL Server — стандартная комбинация имени пользователя для регистрации и пароля, а доверенная проверка подлинности предполагает, что устройство, которое пытается подключиться к экземпляру SQL Server, одобрено процедурой проверки подлинности домена, и результаты этой проверки переданы экземпляру SQL Server: считается, что домен, в котором размещен экземпляр SQL Server, доверяет учетной записи пользователя — проверка выполнена ранее.

Для обеспечения безопасности используем функцию “Backup” для создания файла восстановления.



*Рисунок 3.1. Код для создания файла восстановления*

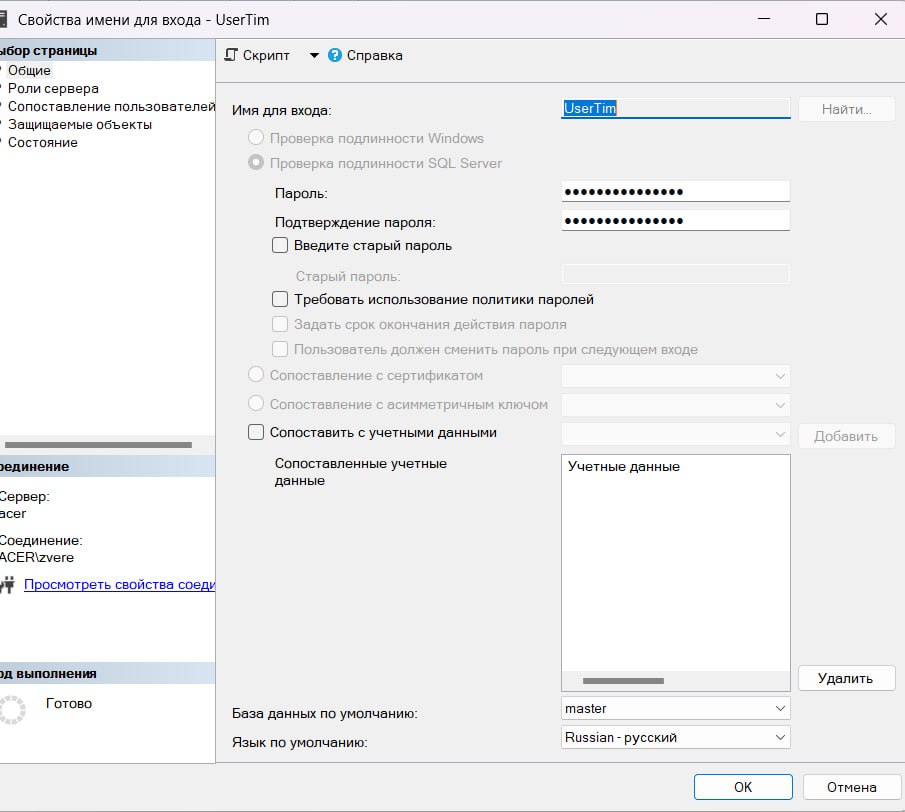
Создали файл резервного восстановления базы данных.



*Рисунок 3.2. Файл восстановления*

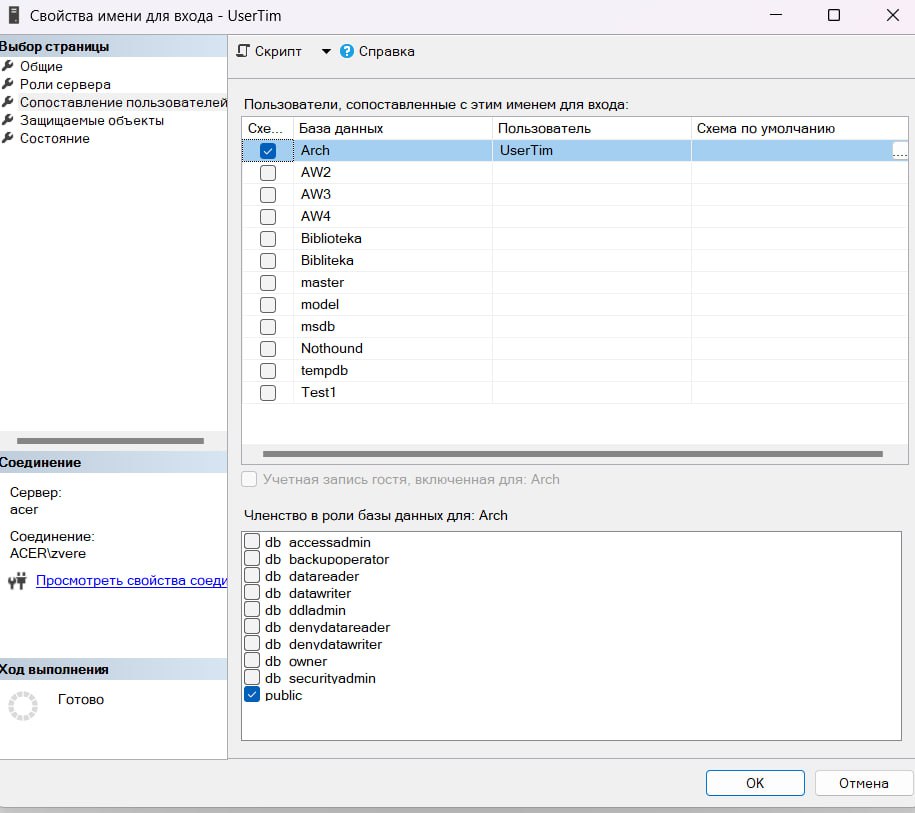
**3.2 Управление безопасностью уровня сервера**

Создание пользователя, присвоение ему пароля.



*Рисунок 3.4. Добавление пользователя*

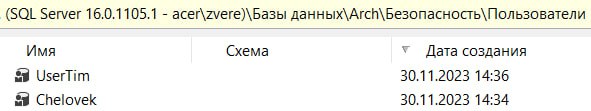
Подключение пользователя к базе данных.



*Рисунок 3.5. Разграничение прав доступа к базе данных*

**3.3 Управление участниками уровня базы данных**

Создали два пользователя на уровне сервера, для дальнейшего подключение их к базе данных.

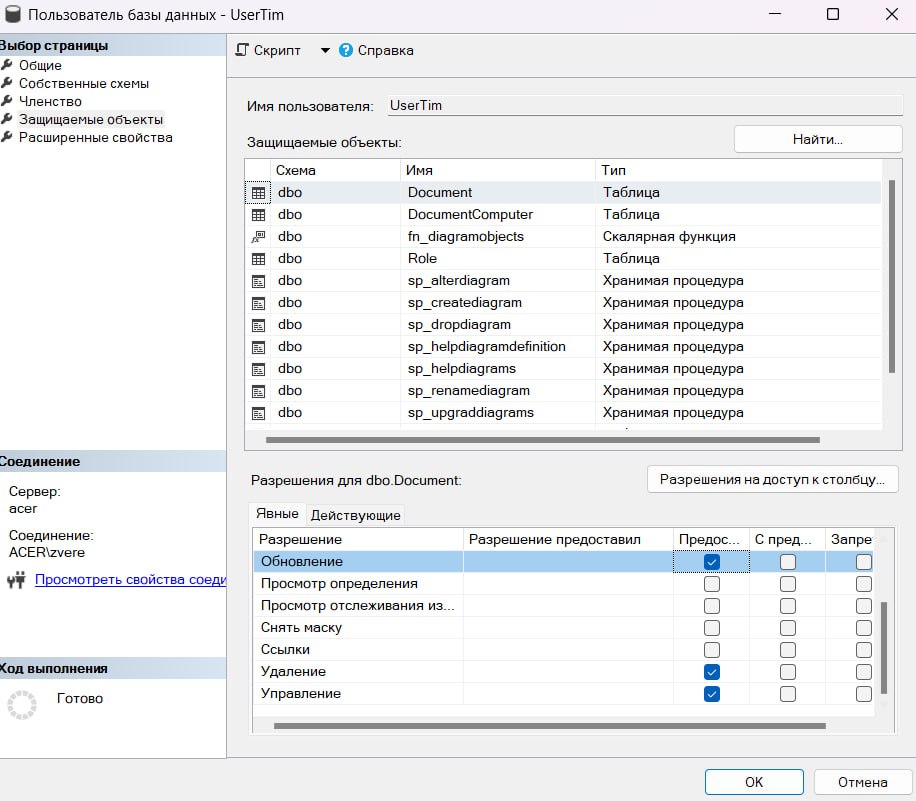
**

*Рисунок 3.6 Созданные пользователи на сервере*

**3.4 Управление разрешениями уровня базы данных**

Управление разрешениями дает возможность администраторам баз данных контролировать, какие данные доступны пользователю и какие операции они могут выполнять с этими данными. Например, администратор может разрешить определенным пользователям только чтение данных из таблицы, но не разрешать им изменение или удаление данных.

SQL Server использует системную базу данных для поддержания внутренних метаданных. Администратор базы данных сервера должен данных уметь управлять ими. В базе данных выдали права доступа к таблицам и права пользования функциями в данных таблицах.

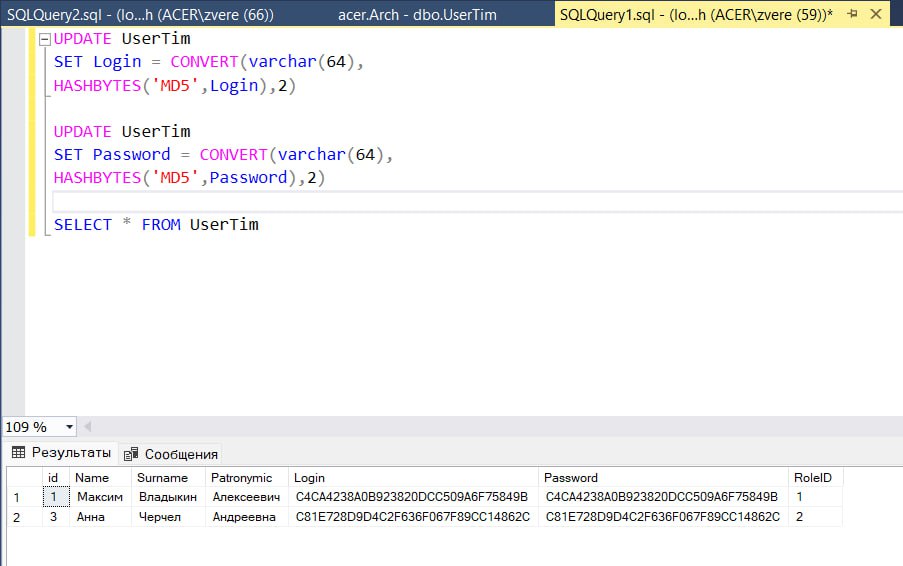


*Рисунок 3.8 Выдача прав пользователю*

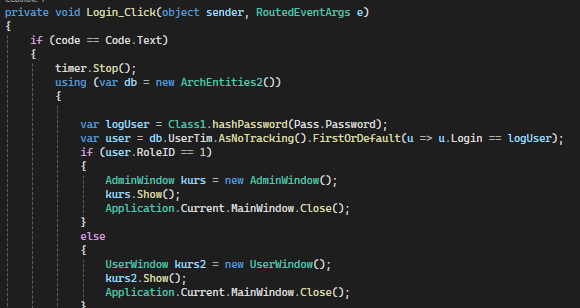
**3.5 Шифрование данных баз данных**

Шифрование — это процесс кодирования информации с целью предотвращения несанкционированного доступа. В случае кражи или утечки зашифрованные данные будут недоступны для прочтения без соответствующего ключа.

Шифрование может быть использовано для защиты конфиденциальных данных, таких как личная информация клиентов, финансовые данные или коммерческая информация. При использовании шифрования, данные хранятся в зашифрованном виде и могут быть прочитаны только с помощью правильного ключа или пароля.

****

*Рисунок 3.9. Запрос на шифрование данных и сами данные*

****

*Рисунок 3.10 Код шифрования в программе*

**ЗАКЛЮЧЕНИЕ**

Все большую актуальность и широкое распространение получают базы данных и системы управления базами данных, использующиеся для обработки больших объемов разного рода информации, в том числе экономической. БД способны хранить информацию о десятках, сотнях тысяч и миллионах различных объектов. Создание базы данных является одним из самых важных частей предприятия в целом.

Базы данных дают возможность множеству пользователей в одно и то же время быстро и безопасно получать доступ к данным и запрашивать их, используя развитую логику и язык запросов.

Подводя итоги работы, можно сказать, что было сделана качественная автоматизация базы данных «Архив». Указанная база данных и автоматизированная информационная система была разработана для персонала и пользователей архива. Благодаря шифрованию, работники и клиенты могут не переживать за безопасность входа.

Автоматизированная информационная система «Архив» позволяет сформировать отчеты по документам, их срокам пребывания в архиве.

**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. Мохова, А. С. Особенности применения баз данных и систем управления базами данных в экономической сфере / А. С. Мохова, М. Ю. Модулева.
2. Эрик Редмонд, Джим Р. Уилсон "Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL" (2015)
3. Алекс Кригель и др. "SQL. Библия пользователя" (2010)

Интернет-ресурсы

1. Полное руководство по языку программирования С# и платформе базы данных. // C# - .NET– Режим доступа: <https://metanit.com/sharp/tutorial/>
2. Справочник 'WPF+C#'// Stack Overflow. – Режим доступа: <https://itproger.com/course/csharp-wpf/4>
3. Учебник ‘Базы данных для начинающих’. – Режим доступа: <https://lesmatveev.narod.ru/knigi_proect/ci_sharp.pdf>
4. Документация ‘Enity Framework’ – Режим доступа: <https://learn.microsoft.com/ru-ru/ef/>
5. Справочник ‘WpfHost’ – Режим доступа: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/app-development/hosting-wpf-applications?view=netframeworkdesktop-4.8>