**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 “Автокредитование”**

Выполнил студент гр. ИСП-20 /\_\_\_\_\_\_\_\_\_\_\_\_\_/Скрипак Сергей Андреевич/

(подпись) (Ф.И.О.)

**ОЦЕНКА:\_\_\_\_\_\_\_\_\_\_\_\_\_**

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

**ПРОВЕРИЛ**

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

(подпись) (Ф.И.О.)

Воркута

2023

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc153109517)

[**ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ** 5](#_Toc153109518)

[**1.1** **СУБД SQL server** 5](#_Toc153109519)

[**1.2** **Microsoft SQL Server Management Studio** 6](#_Toc153109520)

[**1.3** **Entity Framework** 8](#_Toc153109521)

[**ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ** 9](#_Toc153109522)

[**2.1 Разработка диаграммы ERD** 9](#_Toc153109523)

[**2.2 Разработка базы данных** 10](#_Toc153109524)

[**2.3 Словарь данных** 16](#_Toc153109525)

[**ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ** 19](#_Toc153109526)

[**3.1 Введение в безопасность SQL Server** 19](#_Toc153109527)

[**3.2 Управление безопасностью уровня сервера** 20](#_Toc153109528)

[**3.3 Управление участниками уровня базы данных** 22](#_Toc153109529)

[**3.4 Управление разрешениями уровня базы данных** 23](#_Toc153109530)

[**3.5 Шифрование данных баз данных** 25](#_Toc153109531)

[**ЗАКЛЮЧЕНИЕ** 27](#_Toc153109532)

[**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ** 28](#_Toc153109533)

# **ВВЕДЕНИЕ**

Актуальность темы курсового проекта обусловлена тем, что кредитование в настоящее время стало наиболее популярным. Поэтому появилось много банков, которые предлагают различные виды кредитования, а также различные кредитные линии. Одним из самых популярных видов кредитования является автокредитование. Приобрести автомобиль хочется многим, однако не многие могут позволить себе приобретение авто за наличные. То есть автомобиль Вы приобретаете за деньги банка, а после выплачиваете банку эти деньги с процентами.

Многие банки и автосалоны выбирают автокредитование приоритетным направлением развития, а значит, растет конкуренция в этом направлении. С расширением данного рынка услуг - условия между банками по услуге автокредитование становятся все более выгодными, процентные ставки ниже, сроки принятия решения по автокредиту - короче (так называемые экспресс - кредиты). Как правило, банки выдают кредиты на покупку автомобилей, а залогом являются сами автомобили. Это делается для того, чтобы банк мог окупить свои расходы, продав автомобиль в случае невыплаты клиентом кредита. Авто обязательно страхуется клиентом, что уменьшает риск для банка и, следовательно, понижает процентную ставку при автокредитовании.

Одним из факторов, способствующих увеличению числа покупателей, является изменение условий получения автомобиля. Раньше для получения машины в кредит нужно было довольно долго ждать, и первоначальный взнос на автомобиль был значительно выше. Сейчас же этот срок сократился до нескольких дней, а первоначальные взносы стали ниже.

**Объект:** база данных "Автокредитование".

**Предмет:** автоматизация бизнес-процесса "Автокредитование".

**Цель**: разработать базу данных для информационной системы "Автокредитование".

**Задачи**:

* выбрать инструментарий;
* спроектировать базу данных;
* разработать базу данных для информационной системы.

# **ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ**

## **СУБД SQL server**

Microsoft SQL Server — система управления реляционными базами данных (СУБД), 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 небольшими и средними по размеру базами данных до крупных баз данных масштаба предприятия; конкурирует с другими СУБД в этом сегменте рынка.

SQL является общепринятым интерфейсом к базам данных. «Все промышленные базы — Oracle, Microsoft SQL Server, PostgreSQL, MySQL — работают на SQL.

Преимущества решения:

* тесная интеграция с операционной системой Windows;
* высокая производительность, отказоустойчивость;
* поддержка многопользовательской среды;
* расширенные функции резервирования данных;
* работа с удаленным подключением.

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.

## **Microsoft SQL Server Management Studio**

SQL Server Management Studio (SSMS) – это бесплатная графическая среда, включающая набор инструментов для разработки сценариев на T-SQL и управления инфраструктурой Microsoft SQL Server.

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**

Перечень возможностей, которые предоставляет среда SQL Server Management Studio:

* 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;
* настройка репликации баз данных;
* профилировщик XEvent.

## **Entity Framework**

Entity Framework — это набор технологий в ADO.NET, которые поддерживают разработку программных приложений, ориентированных на данные. Архитекторам и разработчикам приложений, ориентированных на обработку данных, приходится учитывать необходимость достижения двух совершенно различных целей. Они должны моделировать сущности, связи и логику решаемых бизнес-задач, а также работать с ядрами СУБД, используемыми для сохранения и получения данных. Данные могут распределяться по нескольким системам хранения данных, в каждой из которых применяются свои протоколы, но даже в приложениях, работающих с одной системой хранения данных, необходимо поддерживать баланс между требованиями системы хранения данных и требованиями написания эффективного и удобного для обслуживания кода приложения.

Платформа Entity Framework позволяет работать с данными в форме специфических для домена объектов и свойств (например, с клиентами и их адресами) без необходимости учитывать формат базовых таблиц и столбцов базы данных, где хранятся эти данные. Entity Framework дает разработчикам возможность работать с данными на более высоком уровне абстракции, создавать и сопровождать приложения, ориентированные на работу с данными, одновременно с этим сокращая объем кода по сравнению с традиционными приложениями. поскольку Entity Framework является компонентом платформа .NET Framework, Entity Framework приложения могут работать на любом компьютере, на котором установлена платформа .NET Framework с пакетом обновления 1 (SP1) версии 3,5.

# **ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

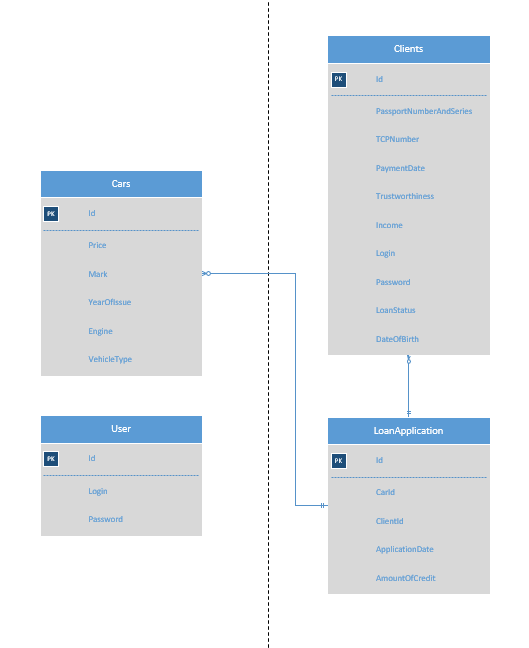
## **2.1 Разработка диаграммы ERD**

**ER-модель** (от англ. *Entity-Relationship model*, *модель «сущность — связь»*) — модель данных, позволяющая описывать концептуальные схемы предметной области.

ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Во время проектирования баз данных происходит преобразование схемы, созданной на основе ER-модели, в конкретную схему базы данных на основе выбранной модели данных (реляционной, объектной, сетевой или др.).

ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена *диаграмма «сущность-связь»* (англ. *Entity-Relationship diagram, ERD*, *ER-диаграмма*).



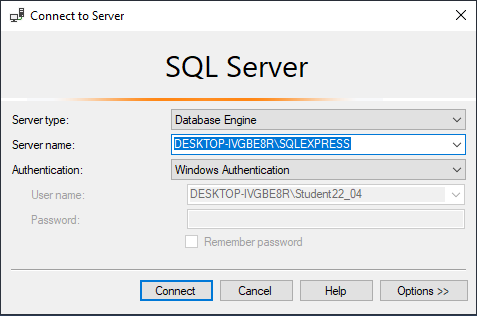
*Рисунок 2.1 Диаграмма ER (Entity-Relationship Diagram)*

Основными сущностями для решения указанной задачи являются: машины, клиенты, заявка на кредит и пользователь. Сразу возникает очевидная связь между сущностями – связь между клиентом и заявками, связь между машинами и заявками.

## **2.2 Разработка базы данных**

Для разработки базы данных в данном случае используется ранее разработанная ER-диаграмма. Также для создания базы данных используется программа Microsoft SQL Server Management Studio.

При открытии программы появляется окно (рис 2.2):

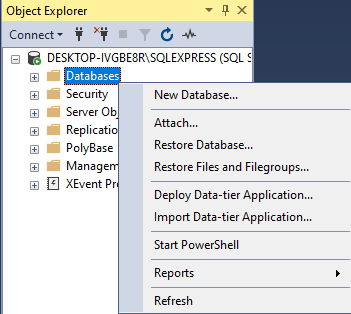


*Рисунок 2.2 Окно подключения к локальному серверу*

При нажатии кнопки “Connect” программа подключается к локальному серверу.

После подключения к локальному серверу станет доступна меню “Object Explorer”.

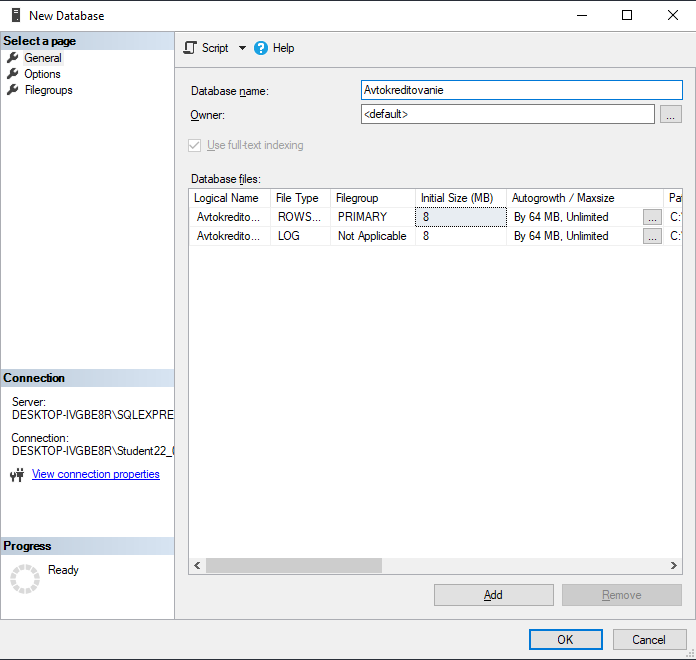
В данном меню есть возможность создать базу данных нажатием правой кнопки мыши на папку “Databases” и в появившимся меню выбором пункта “New Database…” (рис 2.3)



*Рисунок 2.3 Окно с файлами локальной базы данных и меню с созданием базы данных*

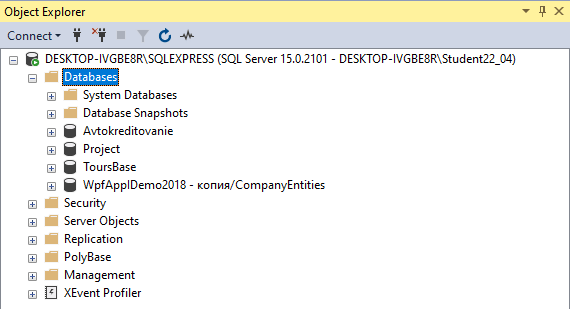
После нажатия на пункт “New Database…” появляется окно с созданием базы данных.

В данном окне нужно ввести название диаграммы и нажать кнопку “Ok” (рис. 2.4)



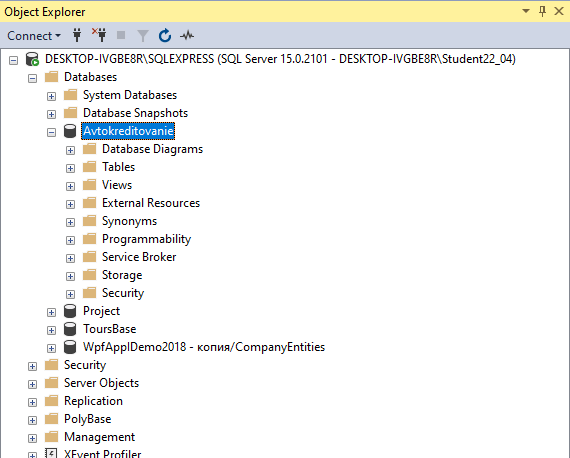
*Рисунок 2.4 Окно создания базы данных*

После создания базы данных, она появляется в меню “Object Explorer” в папке “Databases” (рис. 2.5)

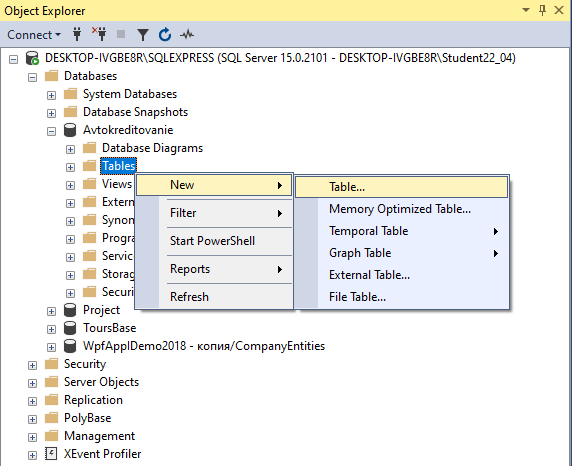


*Рисунок 2.5 Созданная база данных в папке “Databases”*

Далее для работы нужно создать таблицы(“Tables”). Для этого нужно открыть базу данных “Avtokreditovanie” (рис. 2.6), правой кнопкой мыши нажать на “Tables” и выбрать пункт “New => Table…”(рис 2.7):

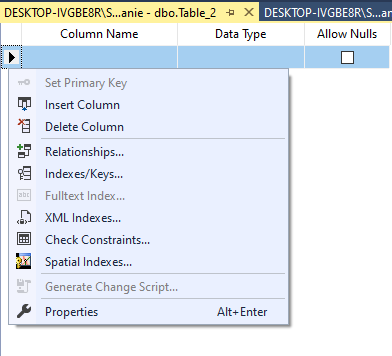


*Рисунок 2.6 Содержание базы данных*

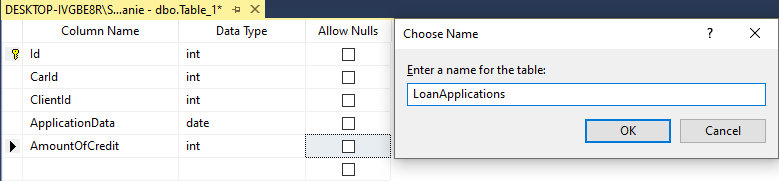


*Рисунок 2.7 Создание таблиц в базе данных*

После нажатия на пункт “Table…” открывается окно с таблицей в которую вносятся данные: название колонки, тип данных, возможность оставить пустое значение, возможность расставить первичные ключи(рис 2.8). После внесения данных в таблицу, она сохраняется и ей дается название(рис 2.9).

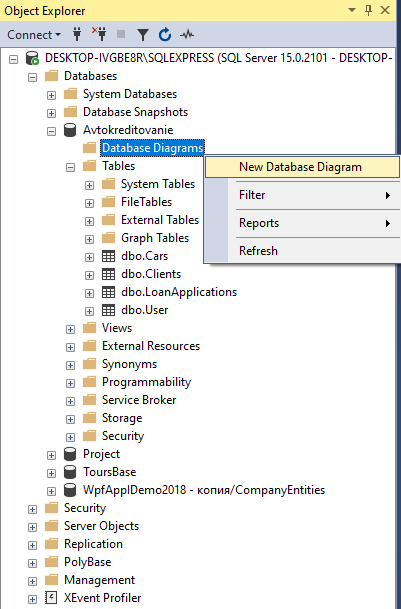


*Рисунок 2.8 Окно с таблицей*

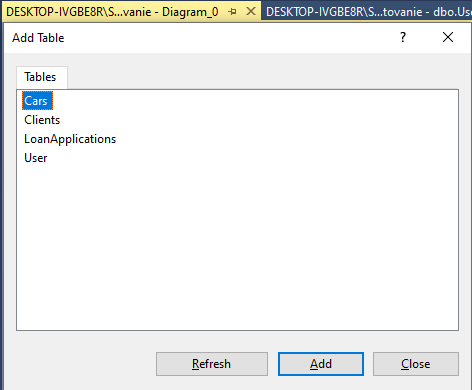


*Рисунок 2.9 Окно сохранения и присваивания названия таблице*

После создания таблиц создается диаграмма. Для того чтобы создать диаграмму базы данных нужно нажать правой кнопкой мыши на “Database Diagrams” и выбрать пункт “New Diagrams” (рис. 2.10), после этого открывается окно “Add Table” в котором можно добавить таблицы в диаграмму базы данных нажатием на кнопку “Add” (рис. 2.11)

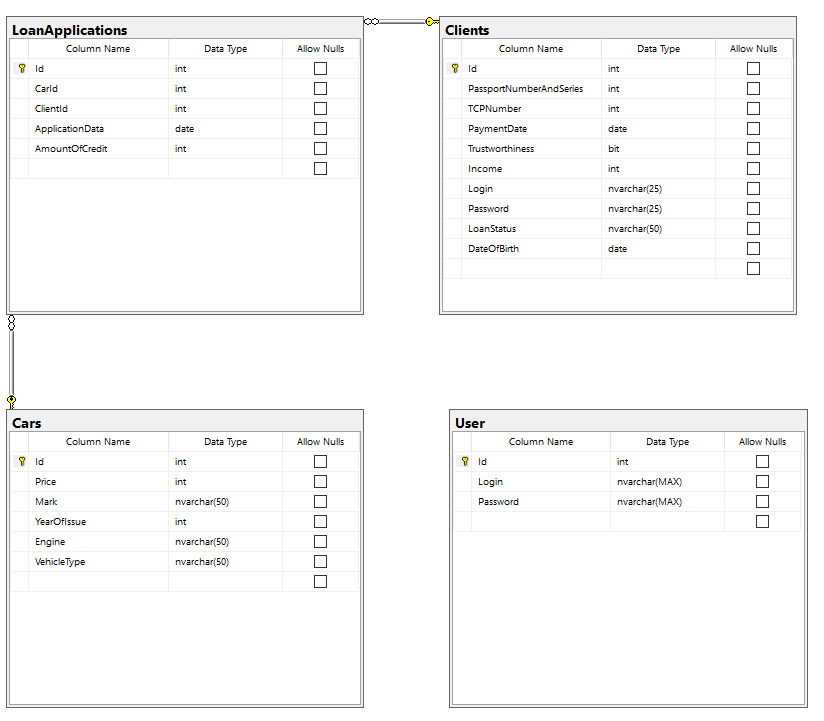


*Рисунок 2.10 Пункт меню с созданием диаграммы базы данных*



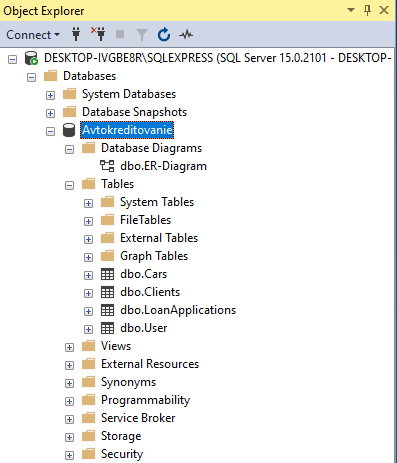
*Рисунок 2.11 Окно с добавлением таблиц в диаграмму базы данных*

После создания диаграммы базы данных и добавления в неё таблиц и расстановки связей её необходимо сохранить кнопкой или сочетанием клавиш “Ctrl+S”.



*Рисунок 2.12 Диаграмма базы данных для системы “Автокредитование”*

После сохранения диаграммы базы данных и таблиц они сохраняются в меню “Object Explorer” в созданной базе данных (рис. 2.13)

***Рисунок 2.13 Созданные таблицы и диаграмма базы данных для системы*

## **2.3 Словарь данных**

Словарь данных (Data Dictionary) - это структура данных, которая хранит информацию о базе данных и ее объектах. Словарь позволяет получить метаданные, такие как названия таблиц, столбцов, типы данных, индексы и т.д.

Таблица 1. Словарь данных

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Data Dictionary | | | | | | | | |
| **ClientTrustworthiness** | | | | | | | | |
| **KEY** | **FIELD NAME** | | | **DATA TYPE / FIELD SIZE** | | **REQUIRED?** | | **NOTES** |
| P.K | **Id** | | | Int | | N | | Идентификатор |
|  | **Trustworthiness** | | | nvarchar(50) | | N | | Надежность |
| **User** | | | | | | | | | |
| **KEY** | | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | | **REQUIRED?** | | **NOTES** | | |
| P.K | | **Id** | Int | | N | | Идентификатор | | |

Таблица 1. Продолжение

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | **Login** | | | | nvarchar(50) | | | | | N | | | | Логин пользователя | | | |
|  | | | **Password** | | | | nvarchar(50) | | | | | N | | | | Пароль пользователя | | | |
| F.K | | | **RoleId** | | | | Int | | | | | N | | | | Идентификатор роли | | | |
| **Role** | | | | | | | | | | | | | | | | | | |
| **KEY** | | | | **FIELD NAME** | | **DATA TYPE / FIELD SIZE** | | | | | **REQUIRED?** | | | | **NOTES** | | | |
| P.K | | | | **Id** | | Int | | | | | N | | | | Идентификатор | | | |
|  | | | | **Title** | | nvarchar(50) | | | | | N | | | | Название роли | | | |
| **Clients** | | | | | | | | | | | | | | | | | | |
| **KEY** | **FIELD NAME** | | | | | | | | | **DATA TYPE / FIELD SIZE** | | | | **REQUIRED?** | | | **NOTES** | |
| P.K | **Id** | | | | | | | | | Int | | | | N | | | Идентификатор | |
|  | **FIO** | | | | | | | | | nvarchar(50) | | | | N | | | Ф.И.О. | |
|  | **PaymentDate** | | | | | | | | | Date | | | | N | | | Дата платежа | |
|  | **Trustworthiness** | | | | | | | | | nvarchar(50) | | | | N | | | Надежность | |
|  | **LoanStatus** | | | | | | | | | nvarchar(50) | | | | N | | | Статус кредита | |
|  | **Car** | | | | | | | | | nvarchar(50) | | | | N | | | Автомобиль | |
| F.K | **CarId** | | | | | | | | | Int | | | | N | | | Идентификатор автомобиля | |
|  | **AmountOfCredit** | | | | | | | | | nvarchar(50) | | | | N | | | Размер кредита | |
|  | **Comment** | | | | | | | | | nvarchar(MAX) | | | | Y | | | Комментарий | |
| **Cars** | | | | | | | | | | | | | | | | | | |
| **KEY** | | **FIELD NAME** | | | | | | **DATA TYPE / FIELD SIZE** | | | | **REQUIRED?** | | | | **NOTES** | | |
| P.K | | **Id** | | | | | | Int | | | | N | | | | Идентификатор | | |
|  | | **ModelName** | | | | | | nvarchar(50) | | | | N | | | | Название модели | | |
|  | | **Price** | | | | | | nvarchar(50) | | | | N | | | | Цена | | |
|  | | **YearOfIssue** | | | | | | Int | | | | N | | | | Год производства | | |
|  | | **Engine** | | | | | | nvarchar(50) | | | | N | | | | Двигатель | | |
|  | | **VehicleType** | | | | | | nvarchar(50) | | | | N | | | | Тип автомобиля | | |
| **LoanApplications** | | | | | | | | | | | | | | | | | |
| **KEY** | | | | | **FIELD NAME** | | | | **DATA TYPE / FIELD SIZE** | | | | **REQUIRED?** | | | **NOTES** | |

Таблица 1. Продолжение

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| P.K | **Id** | int | N | Идентификатор |
| F.K | **CarId** | int | Y | Идентификатор автомобиля |
| F.K | **ClientId** | int | Y | Идентификатор клиента |
|  | **ApplicationDate** | date | N | Дата поданной заявки |
|  | **AmountOfCredit** | nvarchar(100) | N | Размер кредита |
|  | **TCPNumber** | nvarchar(10) | N | ПТС номер |
|  | **Income** | nvarchar(50) | N | Доход |
|  | **FIO** | nvarchar(50) | N | Ф.И.О. |
|  | **DateOfBirth** | date | N | Дата рождения |
|  | **Car** | nvarchar(50) | N | Автомобиль |

# **ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ**

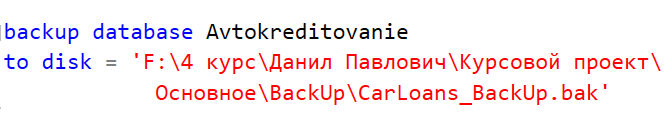
## **3.1 Введение в безопасность SQL Server**

Microsoft SQL Server предоставляет несколько встроенных функций, обеспечивающих безопасность, включая зашифрованную связь по протоколу SSL/TLS, Windows Data Protection API (DPAPI), используемый для шифрования данных в состоянии покоя, аутентификации и авторизации. Каждый администратор базы данных должен настроить эти функции или использовать дополнительные меры безопасности по мере необходимости для удовлетворения требований безопасности и соответствия требованиям своих данных и приложений.

**Резервная копия SQL Server:**

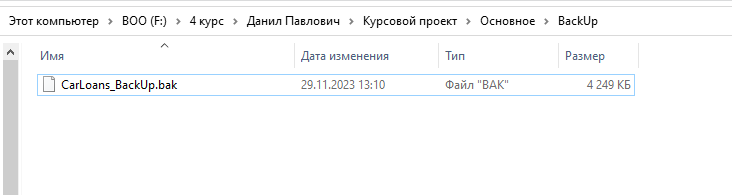
Копия данных SQL Server, которую можно использовать для восстановления данных после сбоя. Резервная копия данных SQL Server создается на уровне базы данных или одного или нескольких ее файлов или файловых групп.

Для создания файла резервной копии базы данных необходимо написать код с указанием места сохранения файла и его имени (Рисунок 3.1)



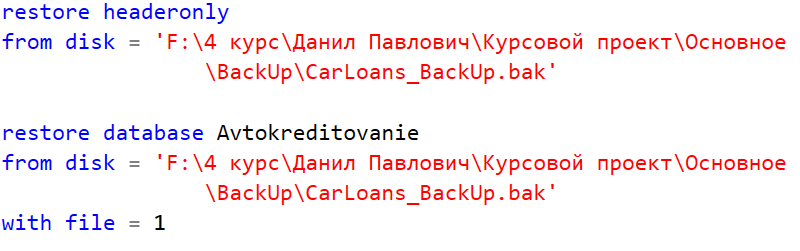
*Рисунок 3.1 Код для создания файла восстановления*

После выполнения данного действия создается файл резервной копии базы данных (Рисунок 3.2)



*Рисунок 3.2 Созданный файл восстановления*

Для восстановления базы данных используется код указанный на рисунке 3.3.



*Рисунок 3.3 Код для восстановления базы данных*

## **3.2 Управление безопасностью уровня сервера**

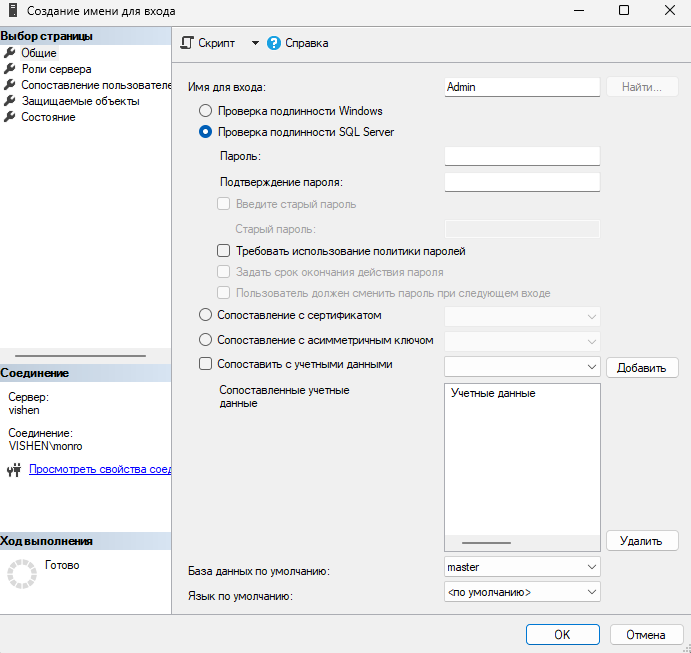
Роли уровня безопасности SQL-сервера используются для помощи в управлении разрешениями на сервере. Роли сервера фиксированы и могут быть назначены пользователям индивидуально.

Роли, которые можно использовать, приведены ниже, с кратким описанием разрешений, назначенных каждой:

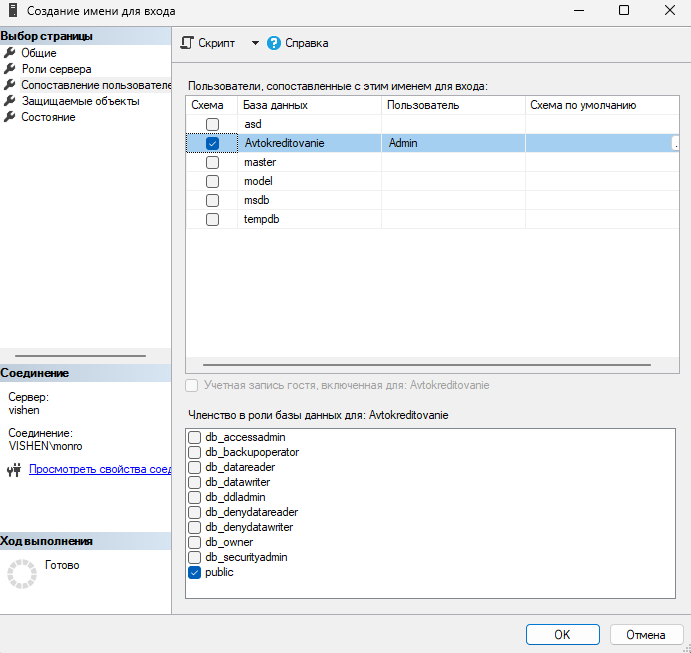
* sysadmin — может выполнять любые действия;
* serveradmin — действия, специфичные для сервера;
* securityadmin — разрешения и пароли, связанные с безопасностью;
* processadmin — Может завершать любые процессы;
* setupadmin — Обслуживание серверов;
* bulkadmin — может выполнять массовые вставки;
* diskadmin — управление файлами на диске;
* dbcreator — разрешения на создание базы данных;
* public — общий доступ пользователей.

Для создания нового пользователя, необходимо:

1. В “Object Explore” открыть папку “Security”;
2. Нажать ПКМ на папку “Logins”;
3. Выбрать пункт “New Login…”;
4. В появившимся окне “Login - New” (Рисунок 3.4) необходимо поставить курсор на строчку “SQL Server authentication”;
5. В поле “Login Name:” ввести логин, а в поле “Password” ввести пароль и подтвердить его в поле ниже;
6. Далее в левой панели “Select a page” выбрать вкладку “User Mapping” (Рисунок 3.5);
7. В появившейся вкладке выдать доступ к необходимой базе данных и нажать на кнопку “OK”.

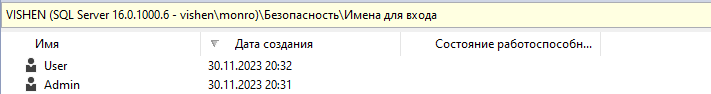


*Рисунок 3.4 Добавление пользователя*



*Рисунок 3.5 Разграничение прав доступа к базе данных*

После выполнения всех действий создаются два пользователя “Admin” и “User”.



*Рисунок 3.6 Созданные пользователи на сервере*

Увидеть всех пользователей можно в “Logins” (Рисунок 3.6).

## **3.3 Управление участниками уровня базы данных**

Пользователь базы данных участник - уровня базы данных, который обычно сопоставляется с именем входа на уровне сервера.

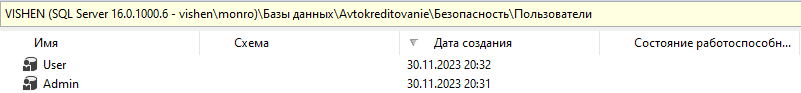
Наличие доступа к серверу не означает, что имя входа имеет доступ к пользовательским базам данных на сервере.

Чтобы логин имел доступ к базам данных, надо сопоставить имя входа (login) и пользователя базы данных (database user) в базе данных.

Можно добавить пользователей базы данных к роли уровня базы данных (database-level roles) для упрощения управления разрешениями.

Наилучшим способом считается сопоставление имен входа именам пользователей базы данных.

После создания пользователей на серверном уровне, данные пользователи также появляются на уровне базы данных (Рисунок 3.7).



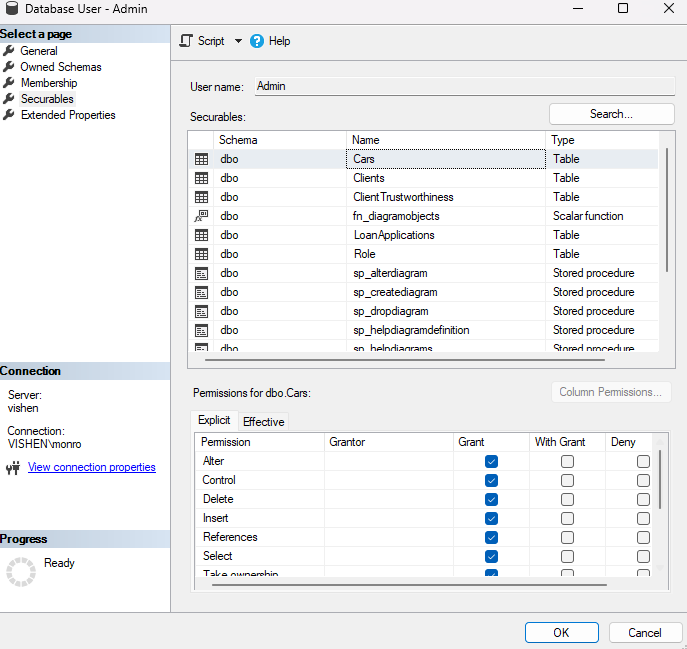
*Рисунок 3.7 Созданные пользователи в базе данных Avtokreditovanie*

## **3.4 Управление разрешениями уровня базы данных**

Разрешения — это типы доступа, предоставляемые конкретным защищаемым объектам. На уровне сервера разрешения назначаются логинам и ролям сервера SQL Server. На уровне базы данных они назначаются пользователям базы данных и ролям базы данных.

Для выдачи разрешений пользователю в базе данных небходимо:

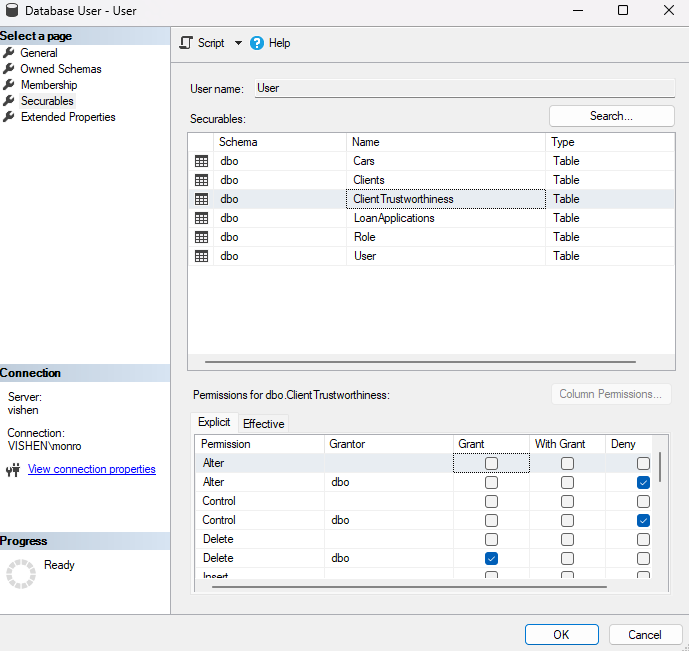
1. В “Object Explore” открыть необходимую базу данных и в ней выбрать папку “Security” и в ней открыть папку “User”;
2. В открывшейся папке “User” выбрать необходимого пользователя и нажать на него ПКМ, в появившемся списке выбрать вкладку “Properties”;
3. После этого откроется окно “Database User – ‘Имя пользователя’” (Рисунок 3.8) и в левой части данного окна необходимо выбрать вкладку “Securables”;
4. В появившейся вкладке выбрать необходимые таблицу в которой будут выдаваться разрешения;
5. Выдать необходимые разрешения и нажать кнопку “OK”. (Рисунок 3.8)



*Рисунок 3.8 Выдача прав администратору*

Для пользователя “Admin” выданы все разрешения во всех таблицах (Рисунок 3.8).

Для пользователя “User” в таблицах “Cars”, “Clients”, “ClientTrustworthiness”, “LoanApplications” выданы разрешения на удаление, добавление, выводить. В таблицах “Role” и “User” у данного пользователя нет разрешений (Рисунок 3.9).



*Рисунок 3.9 Выдача прав пользователю*

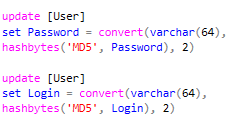
## **3.5 Шифрование данных баз данных**

Шифрование — это процесс сокрытия данных с помощью ключа или пароля. Этот процесс может сделать данные бесполезными без соответствующего ключа дешифрования или пароля.

Шифрование не решает проблемы контроля доступа. Однако это повышает безопасность, ограничивая потерю данных даже в случае обхода контроля доступа. Например, если главный компьютер базы данных настроен неправильно и хакер получает конфиденциальные данные, украденная информация может оказаться бесполезной, если она зашифрована.

SQL Server включает в себя два способа шифрования данных: прозрачное шифрование данных (Transparent Data Encryption, TDE) и расширенное управление ключами (Extensible Key Management EKM).

В данном случае для шифрования используется алгоритм “MD5”. Чтобы зашифровать логин и пароль пользователя, используется код и рисунка 3.10.



*Рисунок 3.10 Код для зашифрофки данных*

# **ЗАКЛЮЧЕНИЕ**

В данный момент автоматизированные информационные системы являются актуальной темой.

При выполнении данной работы, а именно построения базы данных «автокредитование», была спроектирована диаграмма: Entity Relationship. Данная автоматизированная информационная система была разработана для сотрудников банка. С её помощью можно быстро производить добавление заявок на кредит автомобилей и их сопровождение. Базы данных имеют важную роль в современных информационных системах.

Базы данных представляют собой структурированный набор данных, который обеспечивает возможность хранения, управления и извлечения информации. Они широко используются в различных областях, включая бизнес, образование, здравоохранение, научные исследования, государственное управление и другие сферы деятельности.

Экономический эффект от внедрения автоматизированной информационной системы «автокредитование» ожидается за счет сокращения затрат времени на выполняемые менеджером и работником банка задачи, исключения ошибок при формировании отчетов, увеличения времени на анализ. При ручном создании отчетов могут быть допущены ошибки, правильно составленный алгоритм разрабатываемой программы экономит время и исключает ошибки.

Исходя из общего положения и опираясь на совокупность всех ранее вышеперечисленных и упомянутых фактов можно сделать вывод, что поставленные цель и задачи при проектировании информационной системы были выполнены в полной мере.

Github.com – <https://github.com/monroepls/CarLoans>

# **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

1. Долженко, А. И., Глушенко, С. А. Разработка и сопровождение программных систем. Технологии Microsoft.NET для разработки приложений: Лабораторный практикум – Ростов-на-Дону: издательство РГЭУ (РИНХ), 2019. – 133 с.
2. Албахари Джозеф, Албахари Бен. C# 7.0. Карманный справочник: Пер. с англ. Пер. с англ. - СПБ: издательство ООО "Альфа-книга", 2017. — 224 с.

ИНТЕРНЕТ-РЕСУРСЫ

1. Entity–relationship model. – Режим доступа: <https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model>
2. SQL Server technical documentation. – Режим доступа: <https://learn.microsoft.com/en-us/sql/sql-server/?view=sql-server-ver16>
3. АО «РОСБИЗНЕСКОНСАЛТИНГ». Как получить автокредит: пошаговая инструкция. – Режим доступа: <https://www.autonews.ru/news/5bfee33d9a7947e02548e46f>
4. ГК Авто Премиум. Как работает автокредит: полная информация по теме. – Режим доступа: <https://www.inomarkispb.ru/blog/kak-rabotaet-avtocredit>
5. ИА «Банки.ру». Автокредиты. – Режим доступа: <https://www.banki.ru/products/autocredits/>
6. ООО «Сравни.ру». Что такое автокредит? – Режим доступа: <https://www.sravni.ru/avtokredity/info/chto-takoe-avtokredit/>
7. РИА "БанкИнформСервис". Автокредиты. – Режим доступа: <https://bankinform.ru/services/credits/avtokredity>