**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 «Прокат автомобилей»**

Выполнил студент гр. ИСП-21 /\_\_\_\_\_\_\_\_\_/Абилов Эльмар Тайяр оглы /

(подпись) (Ф.И.О.)

**ОЦЕНКА:**\_\_\_\_\_\_\_\_\_\_\_\_\_

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

**ПРОВЕРИЛ**

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

(подпись) (Ф.И.О.)

Воркута

2025

**СОДЕРЖАНИЕ**

[**ВВЕДЕНИЕ** 3](#_Toc189727963)

[**ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ** 4](#_Toc189727964)

[**1.1** **СУБД SQL Server** 4](#_Toc189727965)

[**1.2 Microsoft SQL Server Management Studio** 5](#_Toc189727966)

[**ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ** 7](#_Toc189727967)

[**2.1 Разработка диаграммы ERD** 7](#_Toc189727968)

[**2.2 Словарь данных** 9](#_Toc189727969)

[**ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ** 13](#_Toc189727970)

[**3.1 Разработка базы данных** 13](#_Toc189727971)

[**3.2 Введение в безопасность SQL Server** 17](#_Toc189727972)

[**3.4 Управление участниками уровня базы данных** 22](#_Toc189727973)

[**3.5 Управление разрешениями уровня базы данных** 22](#_Toc189727974)

[**3.6 Резервное копирование баз данных** 24](#_Toc189727975)

[**3.7 Шифрование данных баз данных** 25](#_Toc189727976)

[**ЗАКЛЮЧЕНИЕ** 25](#_Toc189727977)

[**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ** 30](#_Toc189727978)

# **ВВЕДЕНИЕ**

В современном мире клиенты ожидают быстрого и удобного сервиса, что делает необходимым внедрение автоматизированных систем, способных обеспечить высокую степень эффективности и прозрачности всех операций. АИС проката автомобилей обеспечивает автоматизацию таких процессов, как бронирование, учет наличия автомобилей, обработка платежей, управление договорами аренды и мониторинг состояния транспортных средств.

Прокат автомобилей – представляет собой комплексное решение, которое позволяет упростить и оптимизировать все аспекты аренды транспортных средств — от управления парком автомобилей до взаимодействия с клиентами.

**Объект**: база данных «Прокат автомобилей

**Предмет**: анализ бизнес-процесса «Прокат автомобилей».

**Цель работы**: разработать базу данных «Прокат автомобилей».

**Задачи:**

* ﻿﻿выбрать инструментарий;
* ﻿﻿спроектировать базу данных;
* разработать базу данных.

# **ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ**

## **СУБД SQL Server**

SQL SERVER — это система управления базами данных, в работе с которой используется язык программирования SQL. О принципах работы и основных видах серверов баз данных SQL SERVER будет рассказано в представленной статье.

Особенности функционирования SQL SERVER

СУБД SQL SERVER используются для создания, размещения, хранения и управления реляционными (табличными) базами данных на специальных серверах или в облаке. Они работают через настольные приложения и WEB-сайты. К основным преимуществам их функционирования относятся:

* высокоскоростной доступ к данным, обеспечиваемый надежной клиент-серверной архитектурой СУБД;
* простота работы и администрирования, обусловленные понятной структурой языка программирования SQL;
* безопасность хранения информации в БД - благодаря возможности шифрования данных и резервного копирования.

Специфика работы сервера базы данных SQL SERVER заключается в транзакционной обработке данных. Это означает, что по каждому запросу от СУБД обрабатывается и сохраняется небольшое количество информации.

Применение SQL SERVER позволяет автоматизировать решение различных бизнес-задач, поддерживать проведение аналитики данных в режиме онлайн, отслеживать направление ресурсов СУБД, управлять транзакциями (операциями по обработке данных).

Одним из ключевых компонентов SQL Server является его архитектура, которая включает в себя несколько слоев. На самом нижнем уровне находится система хранения данных, которая отвечает за физическое хранение информации на диске. Над этим уровнем располагается система управления памятью, обеспечивающая эффективное использование ресурсов. Также в архитектуре SQL Server присутствует компонент обработки запросов, который интерпретирует SQL-запросы и выполняет необходимые операции над данными.

SQL Server предлагает множество инструментов для администрирования и разработки баз данных. Одним из наиболее популярных инструментов является SQL Server Management Studio (SSMS), который предоставляет пользователям удобный интерфейс для работы с базами данных. SSMS позволяет создавать и изменять структуры баз данных, выполнять запросы, а также управлять пользователями и правами доступа.

Безопасность данных является одной из главных задач любой СУБД, и SQL Server предлагает множество функций для обеспечения защиты информации. Это включает в себя шифрование данных, управление доступом на основе ролей и аудит действий пользователей. Эти механизмы помогают предотвратить несанкционированный доступ к данным и обеспечивают их целостность.

## **1.2 Microsoft SQL Server Management Studio**

SQL Server Management Studio (SSMS) — это интегрированная среда для управления любой инфраструктурой SQL. Используйте SSMS для доступа, настройки, администрирования и разработки всех компонентов SQL Server, [база данных SQL Azure, управляемый экземпляр SQL Azure](https://learn.microsoft.com/ru-ru/azure/azure-sql/database/sql-database-paas-overview)[,](https://learn.microsoft.com/ru-ru/azure/azure-sql/managed-instance/sql-managed-instance-paas-overview)[SQL Server на виртуальной машине](https://learn.microsoft.com/ru-ru/azure/azure-sql/virtual-machines/windows/sql-server-on-azure-vm-iaas-what-is-overview) Azure и [Azure Synapse Analytics](https://learn.microsoft.com/ru-ru/azure/synapse-analytics/sql-data-warehouse/sql-data-warehouse-overview-what-is/). Среда SSMS предоставляет единую комплексную служебную программу, которая сочетает в себе обширную группу графических инструментов с рядом многофункциональных редакторов скриптов для доступа к SQL Server для разработчиков и администраторов баз данных всех профессиональных уровней.

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

Одной из ключевых особенностей SSMS является возможность работы с различными версиями SQL Server. Пользователи могут подключаться к локальным и удаленным серверам, управлять базами данных, выполнять запросы и анализировать результаты. Интерфейс SSMS интуитивно понятен и предоставляет доступ ко всем основным функциям СУБД через меню и панели инструментов.

SSMS поддерживает язык SQL, что позволяет пользователям писать и выполнять запросы к базе данных. В редакторе запросов можно использовать подсветку синтаксиса, автозаполнение и другие функции, которые делают процесс написания кода более удобным и эффективным. Кроме того, SSMS позволяет сохранять запросы в виде скриптов, что облегчает повторное использование кода в будущем.

SSMS предлагает различные механизмы для управления доступом к данным. Администраторы могут создавать роли и назначать им права доступа, что позволяет гибко настраивать уровень доступа для различных пользователей. Это помогает защитить конфиденциальную информацию и предотвратить несанкционированный доступ к данным.

Инструмент также поддерживает аудит действий пользователей, что позволяет отслеживать изменения в базе данных и выявлять потенциальные угрозы безопасности. Аудит может быть настроен на уровне серверов или отдельных баз данных, что обеспечивает дополнительный уровень защиты.

SSMS интегрируется с другими продуктами Microsoft, такими как Visual Studio и Azure Data Studio. Это позволяет разработчикам использовать знакомые инструменты для создания приложений, работающих с SQL Server. Кроме того, SSMS поддерживает различные форматы экспорта данных, что облегчает интеграцию с другими системами и приложениями.

С помощью SSMS можно также создавать резервные копии баз данных и восстанавливать их в случае необходимости. Это важная функция для обеспечения надежности хранения данных и защиты от потерь информации.

# **ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**

## **2.1 Разработка диаграммы ERD**

Характеристика диаграмм «сущность — связь». Данная 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. Структурой данных называют совокупность правил и ограничений, которые отражают связи, существующие между отдельными частями (элементами) данных.

Каждая сущность должна иметь:

* ﻿﻿уникальное имя;
* ﻿﻿один или несколько атрибутов, которые либо принадлежат сущности, либо наследуются через связь;
* ﻿﻿один или несколько атрибутов, которые однозначно идентифицируют каждый экземпляр сущности.

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. Экземпляр атрибута — определенная характеристика конкретного экземпляра сущности.

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

Первичный ключ — это атрибут или совокупность атрибутов и (или) связей, предназначенная для уникальной идентификации каждого экземпляра сущности (совокупность признаков, позволяющих идентифицировать объект). Ключевые атрибуты помещают в начало списка и помечают символом «#».

Описательные атрибуты могут быть обязательными или необязательными.

Обязательные атрибуты для каждой сущности всегда имеют конкретное значение, необязательные могут быть не определены. Обязательные и необязательные описательные атрибуты помечают символами «\*» и «°» соответственно.

Связь – это, отношение одной сущности к другой или к самой себе. Каждая связь может иметь одну из двух модальностей связей. 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.



Рис.2.1.1. ER-диаграмма

Связь может иметь разную модальность с разных концов. Каждая сущность может обладать любым количеством связей с другими сущностями модели. Различают три типа отношений «один-к-одному»; «один-ко-многим»; «многие-ко-многим».

На рис. 2.1.1 показана ER-диаграмма информационной системы «Прокат автомобилей». В данной диаграмме используются такие сущности как: Автомобиль (Car), Rental (Аренда), Покупатель (Customer), Payment (Платёж), Employee (Работник). Каждая из сущностей имеет свои атрибуты.

## **2.2 Словарь данных**

Словарь данных — это структура данных, которая представляет собой набор пар ключ-значение. Каждый ключ уникален в словаре данных, и ему соответствует определенное значение. Словари данных используются для хранения и организации информации, а также для быстрого доступа к данным по ключу. Они широко применяются в программировании для решения различных задач, таких как хранение конфигурационных данных, управление ресурсами и т.д. Словари данных также удобны для хранения и передачи структурированных данных между различными частями программы.

Таблица 1. Car

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Car** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Car\_id** | INT | Y | Уникальный идентификатор автомобиля |
|  | **Model** | VARCHAR(100) | Y | Модель автомобиля |
|  | **Year** | VARCHAR(4) | Y | Год автомобиля |
|  | **LicensePlate** | VARCHAR(100) | Y | Номерной знак автомобиля |
|  | **Color** | VARCHAR(50) | Y | Цвет автомобиля |
|  | **DailyRate** | DECIMAL(10,2) | N | Суточная цена аренды автомобиля |
|  | **Status(Available, Rented, Maintenance)** | ENUM | Y | Статус автомобиля |

Таблица 2. Rental

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Rental** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Rental\_id** | INT | Y | Уникальный идентификатор аренды |
| FK | **Car\_id** | INT | Y | Уникальный идентификатор автомобиля |
| FK | **Customer\_id** | INT | Y | Уникальный идентификатор покупателя |
|  | **Rental\_date** | DATE | Y | Дата аренды |
|  | **Return\_date** | DATE | Y | Дата, до которой автомобиль нужно вернуть. |
|  | **Total Amount** | DECIMAL(10,2) | Y | Общая сумма проката |
|  | **Status(Available, Rented, Maintenance)** | VARCHAR(50) | Y | Статус автомобиля |

Таблица 3. Customer

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Customer** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Customer\_id** | INT | Y | Уникальный идентификатор покупателя |
|  | **FirstName** | VARCHAR(50) | Y | Имя покупателя |
|  | **LastName** | VARCHAR(50) | Y | Фамилия покупателя |
|  | **Email** | VARCHAR(255) | N | Почта покупателя |
|  | **Phone** | VARCHAR(15) | N | Номер телефона покупателя |
|  | **DriverLicenseNumber** | VARCHAR(20) | N | Номер лицензии покупателя |
|  | **Address** | VARCHAR(255) | N | Адрес покупателя |

Таблица 4. Payment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Payment** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Payment\_id** | INT | Y | Уникальный идентификатор платежа |
| FK | **Rental\_id** | INT | Y | Уникальный идентификатор аренды |
|  | **PaymentDate** | DATE | Y | Дата платежа |
|  | **Amount** | DECIMAL(10,2) | N | Сумма платежа |
|  | **PaymentMethod (Credit Card, Cash, etc.)** | VARCHAR(50) | N | Метод платежа |

Таблица 5. Employee

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Employee** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Employee\_id** | INT | Y | Уникальный идентификатор работника |
|  | **FirstName** | VARCHAR(50) | Y | Имя работника |
|  | **LastName** | VARCHAR(50) | Y | Фамилия работника |
|  | **Email** | VARCHAR(100) | N | Почта работника |
|  | **Phone** | VARCHAR(15) | N | Номер телефона работника |

# **ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ**

## **3.1 Разработка базы данных**

Согласно техническому заданию, разработаем базу данных «Прокат автомобилей», которая содержит информацию об автомобильном фонде, арендаторах, менеджерах.

Согласно техническому заданию, создадим таблицу «Аренда».

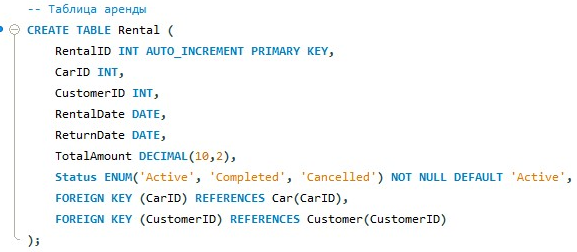


Рис.3.1.2 Создание таблицы Rental

Согласно техническому заданию, создадим таблицу «Платеж».

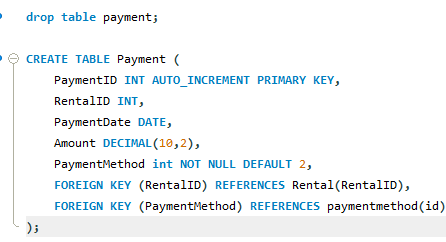


Рис.3.1.3 Создание таблицы Payment

Согласно техническому заданию, создадим таблицу «Автомобиль».

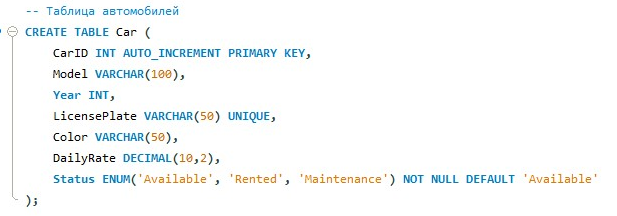
****

Рис.3.1.4 Создание таблицы Car

Согласно техническому заданию, создадим таблицу «Клиент».

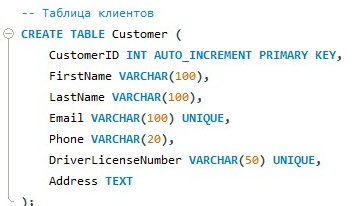


Рис.3.1.5 Создание таблицы Customer

Согласно техническому заданию, создадим таблицу «Сотрудник».

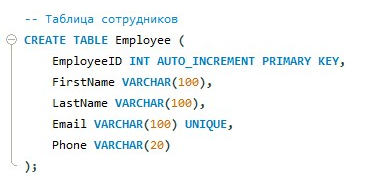


Рис.3.1.6 Создание таблицы Employee

Согласно техническому заданию, выдадим справку об автомобильном фонде.

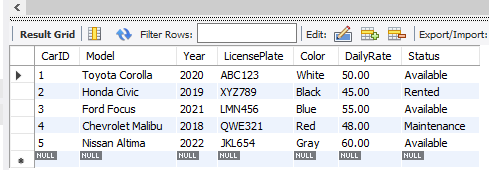
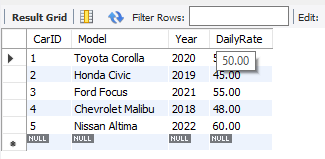


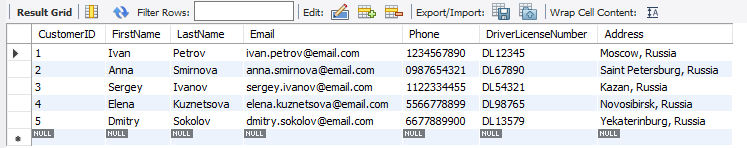
Рис.3.1.7 Вывод всех автомобилей с их характеристиками

Согласно техническому заданию, **выдадим справку об услугах компании и их стоимости.**



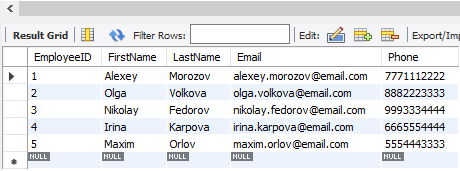
*Рис.3.1.8 Вывод всех автомобилей с их дневной стоимостью аренды*

Согласно техническому заданию, выведем список арендаторов с персональными данными

****

*Рис.3.1.9 Вывод всех клиентов с их контактной информацией*

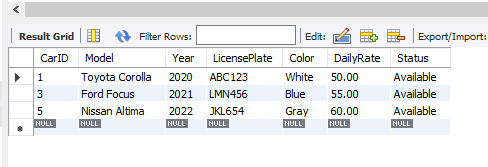
Согласно техническому заданию, выведем список менеджеров с персональными данными



*Рис.3.1.10 Вывод всех сотрудников*

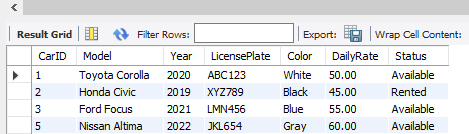
Согласно техническому заданию, **сформировать список свободных**

**автомобилей**



*Рис.3.1.11 Вывод автомобилей, доступных для аренды*

Согласно техническому заданию, выдадим список **автомобилей, которые будут свободны в заданные дни**



*Рис.3.1.12 Автомобили, аренда которых заканчивается до указанной даты*

## **3.2 Введение в безопасность SQL Server**

Безопасность SQL Server — это комплекс мер и механизмов, направленных на защиту данных и предотвращение несанкционированного доступа к базе данных. Введение в безопасность SQL Server включает в себя несколько ключевых аспектов:

**1. Аутентификация**

Аутентификация — это процесс проверки подлинности пользователя, который пытается получить доступ к SQL Server. В SQL Server существуют два основных метода аутентификации:

• Windows Authentication: Использует учетные записи Windows для аутентификации пользователей. Это более безопасный метод, так как он интегрируется с Active Directory и использует существующие политики безопасности Windows.

• SQL Server Authentication: Пользователи создают учетные записи непосредственно в SQL Server. Этот метод менее безопасен, так как пароли хранятся в базе данных.

**2. Авторизация**

После успешной аутентификации необходимо определить, какие действия может выполнять пользователь. SQL Server использует модели безопасности на основе ролей:

• Роли: группы пользователей с определенными правами. Существует несколько встроенных ролей, таких как db\_owner, db\_datareader, db\_datawriter и другие.

• Права доступа: определяют, какие операции (например, SELECT, INSERT, UPDATE, DELETE) могут выполнять пользователи или роли на объектах базы данных.

**3. Шифрование**

Шифрование данных помогает защитить конфиденциальную информацию от несанкционированного доступа:

• Transparent Data Encryption (TDE): Позволяет шифровать файлы базы данных на уровне файловой системы.

• Column-level encryption: Позволяет шифровать отдельные столбцы таблиц.

• Always Encrypted: Обеспечивает шифрование данных в состоянии покоя и при передаче, позволяя только клиентскому приложению расшифровывать данные.

**4. Защита от SQL-инъекций**

SQL-инъекции — это распространенный вид атак, при котором злоумышленник внедряет вредоносный SQL-код в запросы к базе данных. Для защиты от таких атак рекомендуется:

• Использовать параметризованные запросы и подготовленные выражения.

• Ограничивать привилегии пользователей.

• Проверять и фильтровать входные данные.

**5. Мониторинг и аудит**

Мониторинг активности пользователей и аудирование действий в базе данных помогают выявлять подозрительное поведение и нарушения безопасности:

• SQL Server Audit: Позволяет отслеживать действия пользователей и изменения в базе данных.

• Extended Events: Инструмент для мониторинга событий и производительности SQL Server.

**6. Обновления и патчи**

Регулярное обновление SQL Server и применение патчей безопасности помогают защищать систему от известных уязвимостей.

**7. Резервное копирование и восстановление**

Создание резервных копий базы данных и регулярное тестирование восстановления данных обеспечивают защиту от потери данных в случае инцидентов или атак.

**3.3 Управление безопасностью уровня сервера**

1. Создание логинов и пользователей

Откройте SSMS и подключитесь к серверу.

Перейдите в раздел Security> Logins:

* + - * Щелкните правой кнопкой мыши на Logins и выберите New Login....
      * Укажите имя пользователя (воспользуюсь именованием, Employee, Customer).
      * В разделе Authentication выберите SQL Server Authentication и установите пароль.
  + Отключите опцию Enforce password policy (если тестовая среда, иначе оставьте включенной для безопасности).

1. Перейдите в раздел User Mapping:

Поставьте галочку напротив вашей базы данных. “carRental”

Назначьте роли для пользователя:

* + - Для работника: роль db\_owner.
    - Для клиента: роль db\_datareader.

Нажмите OK для сохранения настроек.

1. Настройка ролей и прав доступа

**Для работника:**

1. В SSMS создайте нового логина EmployeeUser:
   * В меню New Login введите имя EmployeeUser и пароль.
2. В разделе User Mapping:
   * Выберите вашу базу данных.
   * Установите роль db\_owner (полный доступ к базе).
   * Login-EmployeeUser
   * Pass-Employee

**Для клиента:**

1. Создайте нового логина ClientUser:
   * + - Укажите имя ClientUser и пароль.
2. В разделе-User Mapping:
   * Выберите вашу базу данных.
   * Установите только роль db\_datareader (только чтение).
   * Login-ClientUser
   * Pass-123

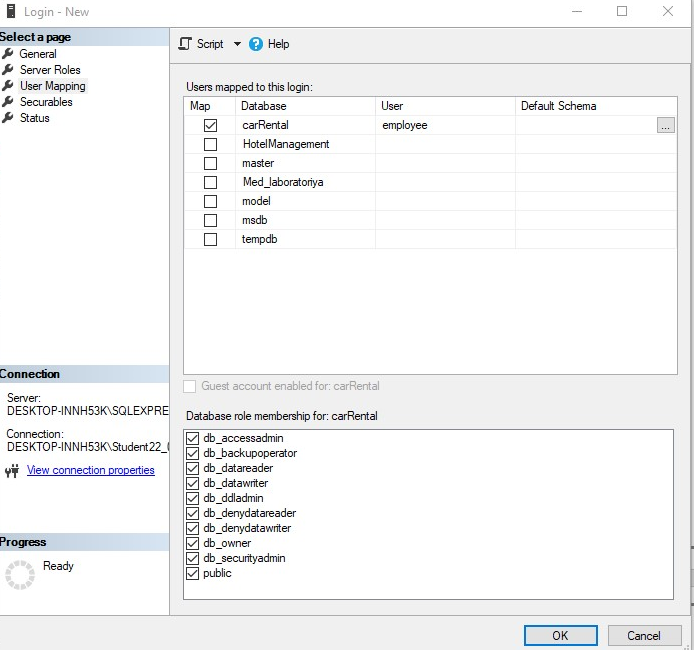


Рисунок 3.2.1 Пример работы с добавлением пользователей (Работник)

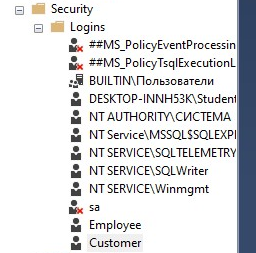
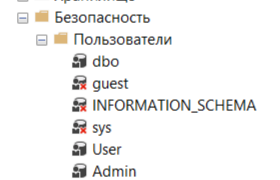


Рисунок 3.2.2 Успешное добавление пользователей

## **3.4 Управление участниками уровня базы данных**

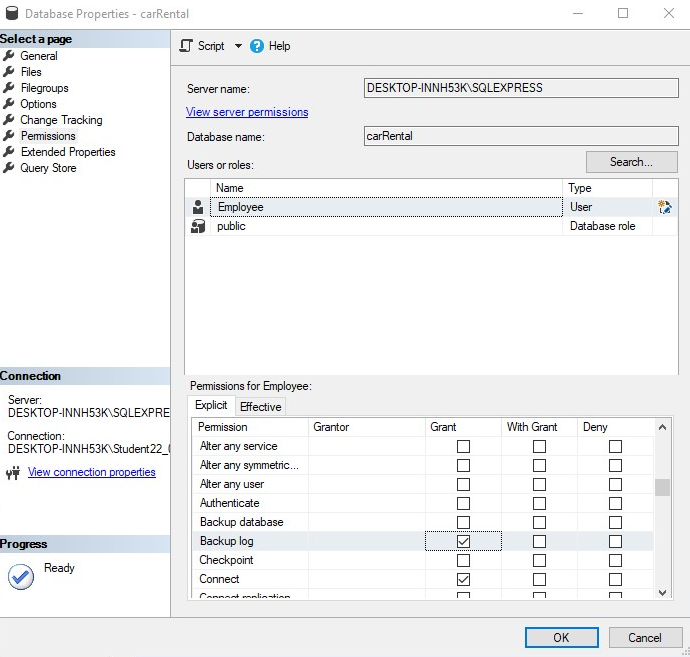
Создание двух пользователей Employee и Customer на уровне сервера, для дальнейшего подключения их к базе данных.



*Рис.3.4.1 Созданные пользователи в базе данных carRental*

## **3.5 Управление разрешениями уровня базы данных**

SQL Server использует системную базу данных для поддержки внутренних метаданных. Администратор базы данных сервера должен уметь управлять ими, а пользователь правильно ими пользоваться. В базе данных выдали права доступа Employee и Customer к таблицам и права пользования функциями в данных таблицах.

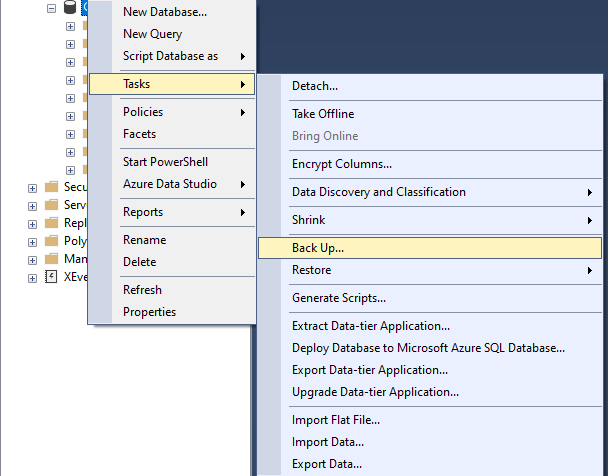


*Рис.3.5.1 Выдача прав пользователю Employee*

## **3.6 Резервное копирование баз данных**

BackUp (резервное копирование) – это процесс создания копии данных с целью их сохранения и возможного восстановления в случае их потери, повреждения или удаления. Резервное копирование может быть выполнено как на физических носителях, таких как внешние жесткие диски, USB-накопители, так и в различных облачных хранилищах. Резервное копирование играет важную роль в обеспечении безопасности данных и защите от потери информации.

Создание Backup copy для базы данных Прокат автомобилей. В Object Explorer нажав “ПКМ” на свою базу данных появится меню взаимодействия с вашей базой данных там находим “Task” &” Backup”



*Рисунок 3.6.1 Выполнение резервной копии через использование интерфейс*

После открывается окно ничего не меняем и нажимает “ОК”.

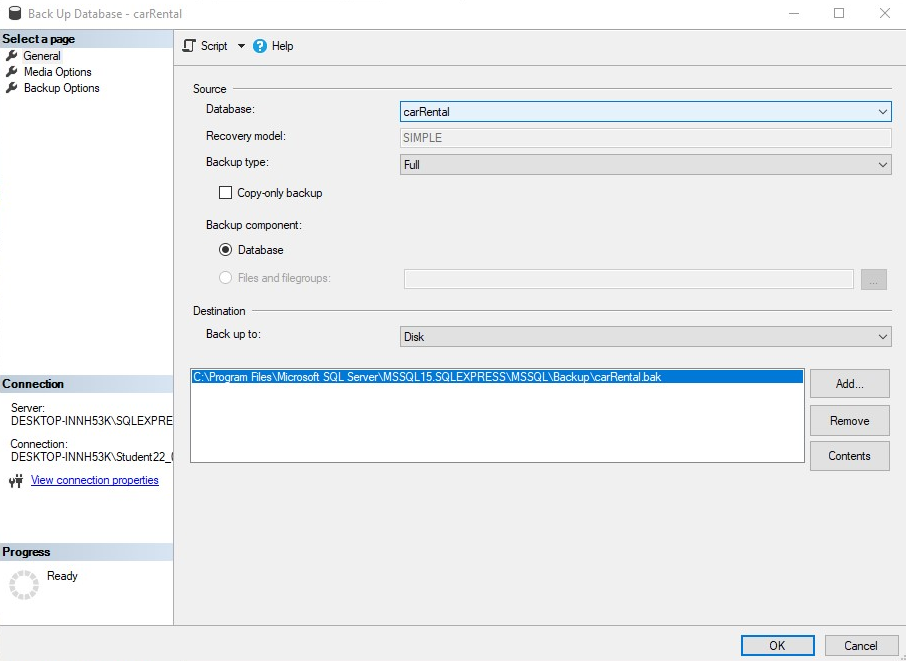


Рисунок 3.6.2 Выполнение резервной копии через использование интерфейс

Далее переходим по пути и копируем в удобное место для хранения.



Рисунок 3.6.2 Файл резервной копии

Создание Backup для базы данных Club.

## **3.7 Шифрование данных баз данных**

Шифрование данных в базе данных играет ключевую роль в обеспечении безопасности и конфиденциальности информации. Оно защищает данные от несанкционированного доступа, даже если злоумышленник получает физический доступ к базе данных. Шифрование представляет собой процесс преобразования данных в нечитабельный формат, который можно расшифровать только с использованием определённого ключа или метода.

Защита конфиденциальности данных

Шифрование позволяет скрыть конфиденциальную информацию, такую как пароли пользователей, личные данные, финансовая информация и т. д. Даже если кто-то получит доступ к файлам базы данных или их резервным копиям, данные останутся защищёнными и нечитаемыми без ключа дешифровки.

Пример: В базе данных могут храниться номера кредитных карт, медицинские записи или логины и пароли пользователей. Шифрование этих данных гарантирует, что никто, кроме авторизованных лиц, не сможет их прочитать.

Соответствие нормативным требованиям и стандартам безопасности

Многие отрасли и организации обязаны соблюдать требования безопасности данных, установленные государственными или международными стандартами. Например:

* GDPR (Общий регламент защиты данных) в Европе требует защиты личных данных.
* PCI DSS (Стандарт безопасности данных индустрии платежных карт) регулирует, как должны храниться и обрабатываться данные о кредитных картах.

Шифрование данных помогает организациям соответствовать таким требованиям, гарантируя защиту конфиденциальных данных.

3. Защита данных при утечке или краже

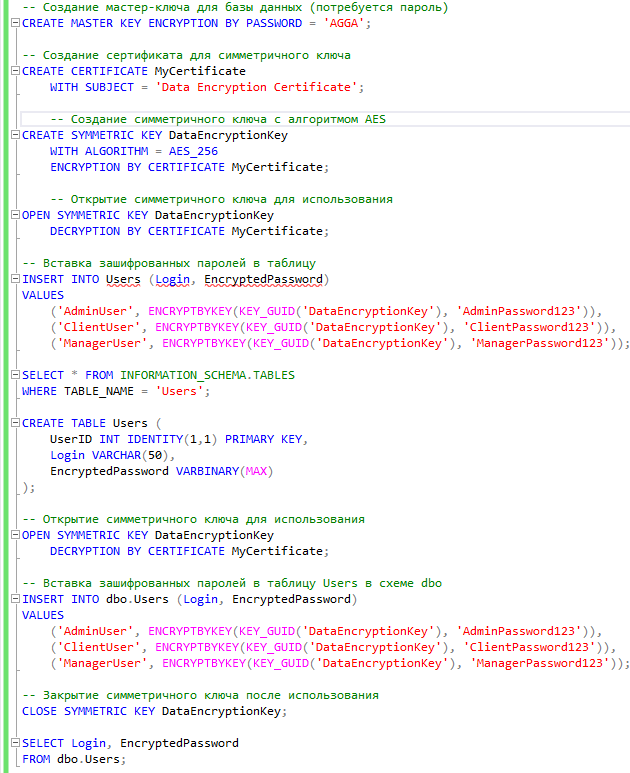
В случае кражи или утечки данных, шифрование снижает риски, так как даже если данные попадут в руки злоумышленников, они не смогут их использовать без соответствующего ключа для расшифровки. Это особенно важно в случае физического доступа к серверам базы данных (например, при краже жесткого диска).

4. Минимизация ущерба при нарушении безопасности

Если система безопасности базы данных будет скомпрометирована (например, атакующий получит доступ к данным на уровне приложения или сервера), зашифрованные данные будут оставаться защищёнными. Даже если атакующий получит доступ к файлам базы данных или резервным копиям, шифрование может предотвратить полный доступ к данным.

5. Защита данных в процессе передачи

Шифрование данных также важно для защиты информации, передаваемой через сети (например, между сервером базы данных и клиентами). Использование SSL/TLS или других методов шифрования сетевого трафика помогает предотвратить утечку данных в процессе их передачи между клиентом и сервером.



# **ЗАКЛЮЧЕНИЕ**

В результате проведенного анализа и проектирования информационной системы для проката автомобилей, была разработана комплексная база данных, которая отвечает всем необходимым требованиям для эффективного управления процессорами аренды автомобилей.

Создание данной информационной системы позволит автоматизировать процессы учета автомобилей, управления клиентами, обработки выдач и возвратов, а также учета задолженности клиента. База данных будет включать в себя информацию о доступных автомобилях, клиентах и выданных автомобилях, что обеспечит быстрый доступ к актуальным данным и упростит их обработку.

В данной курсовой работе был определен необходимый инструментарий и спроектирована база данных «Прокат автомобилей» в интегрированной среде SSMS (SQL Server Management Studio). База данных была разработана для решения проблем, связанных с управлением автопарком и клиентами. Это ценный инструмент для компаний по прокату автомобилей всех размеров, который повышает точность, эффективность и прозрачность их операций, а также обеспечивает удобство для пользователей.

Ссылка на репозиторий:

# https://github.com/Elmar011/carrental

# **СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ**

Электронные ресурсы

1. Сервер баз данных SQL SERVER // Высшая школа бизнеса – Режим доступа: <https://hsbi.hse.ru/articles/server-baz-dannykh-sql-server/>
2. Что такое SQL Server Management Studio (SSMS)? // Microsoft Build – Режим доступа: <https://learn.microsoft.com/ru-ru/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver16>

Видео – ресурсы

1. Создание ER – диаграммы // Wordskills <https://nationalteam.worldskills.ru/skills/proektirovanie-er-diagrammy/>
2. Администрирование SQL Server ч.1 // SQL Developer <https://youtu.be/X508KcPidyk?si=OYzfUpwM34gAuKS1>
3. BackUp/Restore баз данных. Администрирование SQL Server ч.2 // SQL Developer <https://www.youtube.com/watch?v=rGTTDC0psr8&list=PLhhjwMYxzolgZyb5R3ZnTbVPG4_gHHQJh&index=2>
4. Администрирование SQL Server ч.3. Login, User. // SQL Developer
5. <https://www.youtube.com/watch?v=ETX3OXonDrg&list=PLhhjwMYxzolgZyb5R3ZnTbVPG4_gHHQJh&index=3>
6. Администрирование SQL Server ч.4. Роли. // SQL Developer <https://www.youtube.com/watch?v=_JmzlbA3cp8&list=PLhhjwMYxzolgZyb5R3ZnTbVPG4_gHHQJh&index=4>
7. Импорт/Экспорт данных SQL. Администрирование SQL Server ч.5 // SQL Developer <https://www.youtube.com/watch?v=i4BLUNYObP4&list=PLhhjwMYxzolgZyb5R3ZnTbVPG4_gHHQJh&index=5>
8. Установка SQL Server. Администрирование SQL Server ч.6 // SQL Developer <https://www.youtube.com/watch?v=F8nG5P3ErIo&list=PLhhjwMYxzolgZyb5R3ZnTbVPG4_gHHQJh&index=6>
9. SQL Profiler. Администрирование SQL Server ч.7 // SQL Developer <https://www.youtube.com/watch?v=EMooNYEf7is&list=PLhhjwMYxzolgZyb5R3ZnTbVPG4_gHHQJh&index=7>