**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 «Автосалон»**

Выполнил студент гр. ИСП-21/\_\_\_­\_\_/Тельтевский Николай Владимирович/

(подпись) (Ф.И.О.)

**ОЦЕНКА:\_\_\_\_\_\_\_\_\_\_\_\_\_**

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

**ПРОВЕРИЛ**

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

(подпись) (Ф.И.О.)

Воркута

2024

**Оглавление**

[**ВВЕДЕНИЕ** 3](#_Toc185357629)

[**ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ** 4](#_Toc185357630)

[**1.1 СУБД SQL server** 4](#_Toc185357631)

[**1.2 Microsoft SQL Server Management Studio** 6](#_Toc185357632)

[**ГЛАВА 2. ПРОЕТИРОВАНИЕ БАЗЫ ДАННЫХ** 8](#_Toc185357633)

[**2.1 Разработка диаграммы ERD** 8](#_Toc185357634)

[**2.2 Словарь данных** 10](#_Toc185357635)

[**ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ** 13](#_Toc185357636)

[**3.1 Разработка базы данных** 13](#_Toc185357637)

[**3.2 Введение в безопасность SQL Server** 18](#_Toc185357638)

[**3.3 Управление безопасностью уровня сервера** 20](#_Toc185357639)

[**3.4 Управление участниками уровня базы данных** 21](#_Toc185357640)

[**3.5 Управление разрешениями уровня базы данных** 22](#_Toc185357641)

[**3.6 Резервное копирование баз данных** 23](#_Toc185357642)

[**3.7 Шифрование данных баз данных** 24](#_Toc185357643)

# **ВВЕДЕНИЕ**

В современном мире автомобильная отрасль является одной из ключевых. Создание информационной системы для автосалона позволит улучшить управление клиентской базой, автоматизировать процессы продажи и обслуживания автомобилей, а также повысить уровень обслуживания клиентов. Это будет стратегически важно для автосалонов в условиях активной конкуренции и постоянно меняющихся технологий.

Данная информационная система поможет управлять информацией о доступных автомобилях, заказах, клиентах и других аспектах, связанных с продажей автомобилей. Она обеспечит эффективное управление бизнесом, улучшит сервис для клиентов и повысит общую производительность.

**Объект:** информационная система «Автосалон».

**Предмет:** автоматизация бизнес-процесса работы автосалона.

**Цель работы -** разработать базу данных и информационную систему «Автосалон».

**Задачи:**

* выбрать инструментарий;
* спроектировать базу данных;
* сделать разграничение прав доступа базы данных;
* создать информационную систему.

# **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.

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;
* Транзакционная обработка данных. По каждому запросу от СУБД обрабатывается и сохраняется небольшое количество информации.

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).

Для реализации функций СУБД на сегодняшний день чаще всего используются следующие SQL-серверы:

MS SQL server - многопользовательский программный продукт, разработанный компанией Microsoft, обладающий высокой производительностью и отказоустойчивостью, тесно интегрированный с ОС Windows. Этот сервер поддерживает удаленные подключения, работает с многими популярными типами данных, дает возможность создавать триггеры и хранимые данные, имеет практичные и удобные утилиты для настройки;

Oracle Database server - СУБД, предназначенная для создания, консолидации и управления базами данных в облачной среде. Используя этот сервер, можно как автоматизировать обычные бизнес-операции, так и выполнять динамический многомерный анализ данных (OLAP), проводить операции с документами xml-формата и управлять разделенной и локальной информацией;

IBM DB2 - семейство СУБД для работы с реляционными базами данных, признанное самым производительным, имеющим высокие технические показатели и возможности масштабирования. SQL-серверы этой группы характеризуются мультиплатформенностью, способностью к мгновенному созданию резервных копий и восстановлению БД, реорганизации таблиц в онлайн-режиме, разбиению баз данных, определению пользователями новых типов данных;

MySQL - СУБД, разработанная и поддерживаемая компанией Oracle. В основном она используется локальными или удаленными клиентами, позволяя им работать с таблицами разных типов, поддерживающих полнотекстовый поиск или выполняющих транзакции на уровне отдельных записей;

PostgreSQL - СУБД с открытым исходным кодом, работающая с объектно-реляционными (поддерживающими пользовательские объекты) базами данных. Также PostgreSQL предназначена для создания, хранения и извлечения сложных структур данных. Она поддерживает самые различные типы данных (среди них - числовые, текстовые, булевы, денежные, бинарные данные, сетевые адреса, xml и другие).

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

Microsoft SQL Server Management Studio (SSMS) — это программное приложение, разработанное корпорацией Майкрософт, которое используется для настройки, управления и администрирования всех компонентов Microsoft SQL Server. Впервые запущенный в Microsoft SQL Server 2005, он является преемником Enterprise Manager в SQL 2000 или более ранних версиях. Инструмент включает в себя как редакторы скриптов, так и графические инструменты, которые работают с объектами и функциями сервера.

Центральной функцией SSMS является обозреватель объектов, который позволяет пользователю просматривать, выбирать и работать с любыми объектами на сервере. Он также поставлял отдельное экспресс-издание, которое можно было свободно загрузить; однако последние версии SSMS полностью поддерживают подключение к любому экземпляру SQL Server Express и управление им. Корпорация Майкрософт также включила обратную совместимость для более старых версий SQL Server, что позволило более новой версии SSMS подключаться к более старым версиям экземпляров SQL Server. Он также поставляется с Microsoft SQL Server Express 2012, или пользователи могут загрузить его отдельно.

В среде SQL Server Management Studio доступны следующие функции:

* Создание баз данных SQL Server, сценариев SQL и заданий агента SQL Server с помощью SQL Server;
* Подключение к экземплярам базы данных SQL — объекты базы данных SQL, такие как таблицы, индексы и схемы, можно создавать, изменять и удалять;
* Экземплярами базы данных SQL можно управлять и отслеживать.

Он также используется для создания запросов и скриптов, а также для развертывания, мониторинга и обновления компонентов уровня данных, используемых приложениями.

SQL Server Management Studio — это веб-программа и клиентская программа, которая позволяет управлять базами данных и хранилищами данных на локальном компьютере или в облаке. Он также позволяет управлять средой SQL Server путем создания, изменения и удаления объектов, администрирования баз данных и мониторинга среды SQL Server.

SSMS также предоставляет платформу, которая позволяет администраторам и разработчикам SQL использовать различные мощные редакторы скриптов. В двух словах, это упрощает обслуживание и администрирование SQL Server и связанных с ним экземпляров, чтобы вы могли сосредоточиться на своих приложениях.

SSMS позволяет администраторам выполнять следующие задачи:

* Подключение к базам данных, данным и экземплярам SQL Server;
* Конфигурацию экземпляров SQL Server можно просматривать и изменять;
* Управление подключениями и разрешениями в SQL Server;
* Создание новых баз данных, пользователей и ролей, а также выполнение других административных обязанностей;
* Доступ к журналам ошибок, системным базам данных и двоичным файлам экземпляра SQL Server.

# **ГЛАВА 2. ПРОЕТИРОВАНИЕ БАЗЫ ДАННЫХ**

## **2.1 Разработка диаграммы ERD**

Описание:

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».

Понятия «ER-модель» и «ER-диаграмма» часто не различают, хотя для визуализации ER-моделей могут быть использованы и другие графические нотации, либо визуализация может вообще не применяться (например, использоваться текстовое описание).

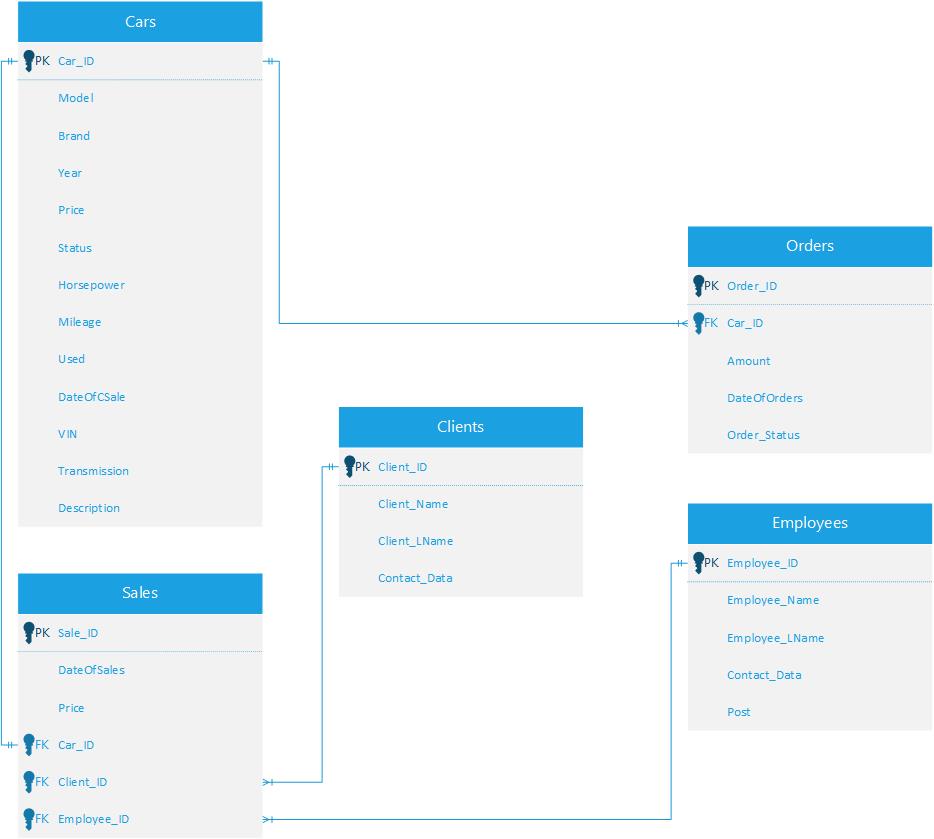
Элементы:

Сущность — представляет объект или сущность, которая будет храниться в базе данных как таблица. Обычно изображается в виде прямоугольника с названием сущности внутри.

Атрибут — представляет информацию о сущности. Он хранится в виде столбца в таблице базы данных. Обычно изображается как овал, присоединенный к сущности линией.

Связь — показывает связь между двумя сущностями. Она может быть однонаправленной или двунаправленной и иметь атрибуты, определяющие характер связи. Обычно изображается линией, соединяющей две сущности, с названием связи и типом связи.

Кардинальность — определяет количество связанных сущностей в отношении. Например, один ко многим, многие ко многим и т.д. Он обычно отображается в виде числовых значений рядом с линиями связей.

Ключ — определяет уникальный идентификатор сущности. Обычно обозначается подчеркнутым атрибутом.

*Рисунок 2.1.1 ER-диаграмма*

В данной диаграмме используются такие сущности как Клиенты(Clients), Заказы(Orders), Автомобили(Cars), Sales(Продажи), Сотрудники(Employees).

Сущность Клиент(Client) имеет следующие атрибуты: Client\_ID(Уникальный ключ), Сlient\_Name(Имя клиента), Client\_FName(Фамилия клиента), Contact\_Data(Контактные данные)

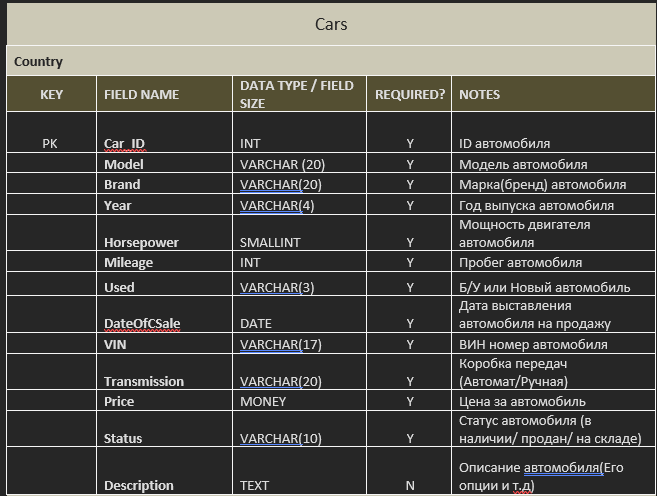
Сущность Автомобили(Cars) имеет следующие атрибуты: Car\_ID(Уникальный ключ), Model(Модель), Brand(Бренд), Year(Год выпуска), Price(Цена), Status(Статус), Horsepower(Мощность), Mileage(Пробег), Used(Новая или Б/У), DateOfCSale(Дата выставления на продажу), VIN(ВИН номер), Transmission(Варианты коробки передач), Description(Описание).

## **2.2 Словарь данных**

Словарь данных (или дата-словарь) — это структурированный набор информации, который описывает данные, используемые в системе или приложении. Он служит для управления данными и их метаданными, обеспечивая ясность и согласованность в понимании данных. Вот основные аспекты словаря данных:

1. **Определение данных**: Словарь данных содержит определения всех данных, используемых в системе, включая типы данных, форматы и ограничения.
2. **Структура**: Он описывает структуру данных, включая таблицы, поля, связи между ними и иерархию.
3. **Метаданные**: Словарь данных включает метаданные, которые предоставляют дополнительную информацию о данных, такие как автор, дата создания, правила обработки и т.д.
4. **Управление данными**: Словарь данных помогает в управлении данными, обеспечивая единообразие и стандарты для их использования и обработки.
5. **Документация**: Он служит важным инструментом для документирования данных, что облегчает понимание и использование данных как для разработчиков, так и для пользователей.

Словари данных часто используются в базах данных, системах управления данными и в процессе разработки программного обеспечения для обеспечения четкого понимания и управления данными.

Таблица 1. Cars

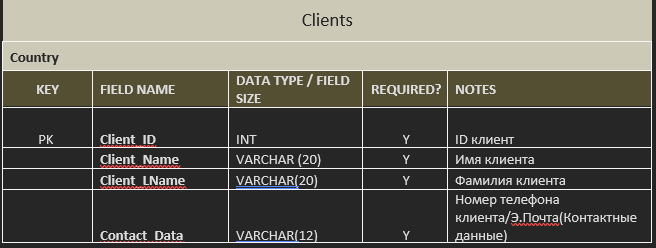
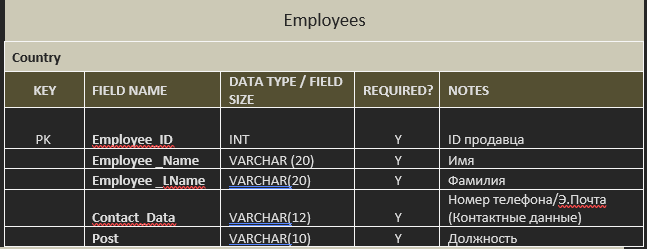
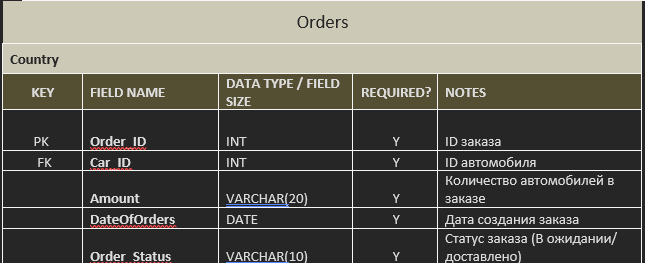
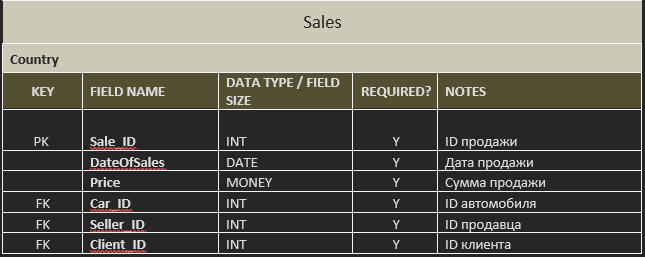
Таблица 2. Clients

Таблица 3. Employees

Таблица 4. Orders

Таблица 5. Sales

# **ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ**

## **3.1 Разработка базы данных**

Создание базы данных:

CREATE DATABASE CarShow;

CREATE TABLE Cars(

Car\_ID INT PRIMARY KEY AUTO\_INCREMENT,

Brand VARCHAR(20) NOT NULL,

Model VARCHAR(20) NOT NULL,

Years VARCHAR(4) NOT NULL,

Horsepower SMALLINT NOT NULL,

Mileage INT NOT NULL,

Used VARCHAR(3) NOT NULL,

DateOfCSale DATE,

VIN VARCHAR(17) NOT NULL,

Transmission VARCHAR(20) NOT NULL,

Price DECIMAL(19,3) NOT NULL,

Car\_status VARCHAR(10) NOT NULL,

Description TEXT

);

CREATE TABLE Clients(

Client\_ID INT PRIMARY KEY AUTO\_INCREMENT,

Client\_Name VARCHAR(20) NOT NULL,

CLient\_LName VARCHAR(20) NOT NULL,

Contact\_Data VARCHAR(35) NOT NULL

);

CREATE TABLE Employees(

Employee\_ID INT PRIMARY KEY AUTO\_INCREMENT,

Employee\_Name VARCHAR(20) NOT NULL,

Employee\_LName VARCHAR(20) NOT NULL,

Contact\_data VARCHAR(20) NOT NULL,

Post VARCHAR(25) NOT NULL

);

CREATE TABLE Orders(

Order\_ID INT PRIMARY KEY AUTO\_INCREMENT,

Amount SMALLINT NOT NULL,

DateOfOrders DATE NOT NULL,

Order\_status VARCHAR(20) NOT NULL,

Car\_ID INT NOT NULL,

FOREIGN KEY (Car\_ID) REFERENCES Cars (Car\_ID)

);

CREATE TABLE Sales(

Sale\_ID INT PRIMARY KEY AUTO\_INCREMENT,

DateOfSales DATE NOT NULL,

Price DECIMAL(19,3) NOT NULL,

Car\_ID INT NOT NULL,

Client\_ID INT NOT NULL,

Employee\_ID INT NOT NULL,

FOREIGN KEY (Car\_ID) REFERENCES Cars (Car\_ID),

FOREIGN KEY (Client\_ID) REFERENCES Clients (Client\_ID),

FOREIGN KEY (Employee\_ID) REFERENCES Employees (Employee\_ID)

);

Внесение данных в базу данных:

INSERT INTO Cars (Brand, Model, Years, Horsepower, Mileage, Used, DateOfCSale, VIN, Transmission, Price, Car\_status, Description) VALUES

('Tesla', 'Model S', '2020', 670, 15000, 'Да', '2022-10-12', '5YJSA1E26LF123456', 'Автоматическая', 79999.999, 'Доступна', 'Электрический седан с большой мощностью.'),

('Honda', 'Civic', '2019', 158, 25000, 'Нет','2023-01-22', '19XFC2F59KE123456', 'Автоматическая', 21999.500, 'Продана', 'Компактная машина с маленьким расходом бензина.'),

('Ford', 'Mustang', '2021', 450, 5000, 'Нет','2024-12-01', '1FA6P8CF5M5101234', 'Механическая', 55999.000, 'Доступна', 'Спортивное купе с мощным двигателем.'),

('Toyota', 'Camry', '2018', 203, 30000, 'Да','2021-06-09', '4T1B11HK8JU123456', 'Автоматическая', 23999.750, 'Доступна', 'Надежный седан с просторным салоном.');

INSERT INTO Clients (Client\_Name, CLient\_LName, Contact\_data) VALUES

('Джон', 'Дое', 'john.doe@example.com'),

('Джейн', 'Смит', 'jane.smith@example.com'),

('Майкл', 'Джордан', 'michael.jordan@example.com');

INSERT INTO Employees (Employee\_Name, Employee\_LName, Contact\_data, Post) VALUES

('Иван', 'Иванов', '1234567890', 'Менеджер'),

('Петр', 'Петров', '0987654321', 'Разработчик'),

('Светлана', 'Сидорова', '1122334455', 'Дизайнер');

INSERT INTO Orders (Amount, DateOfOrders, Order\_status, Car\_ID) VALUES

(150, '2024-10-28', 'В обработке', 1),

(200, '2024-11-02', 'Доставленно', 2),

(300, '2024-11-07', 'Доставленно', 3);

INSERT INTO Sales (DateOfSales, Price, Car\_ID, Client\_ID, Employee\_ID) VALUES

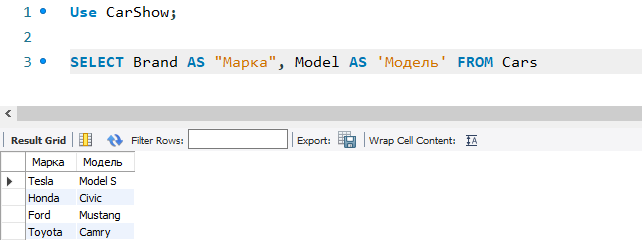
('2024-10-15', 25000.000, 1, 1, 1),

('2024-02-16', 30000.500, 2, 2, 1),

('2024-05-17', 15000.750, 3, 3, 2),

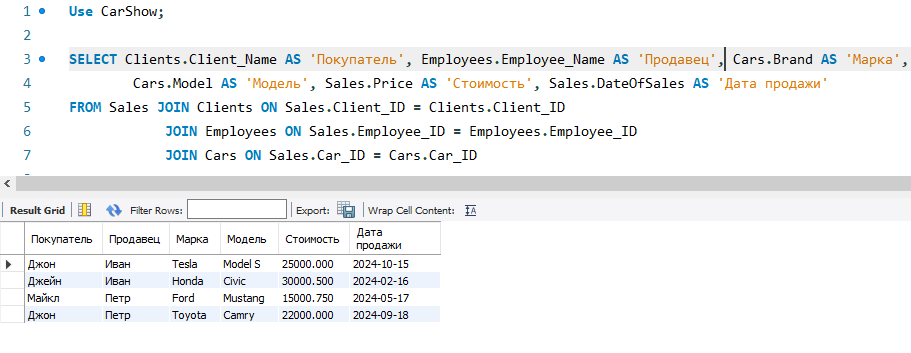
('2024-09-18', 22000.000, 4, 1, 2);

Согласно техническому заданию, необходимо вывести следующие запросы:

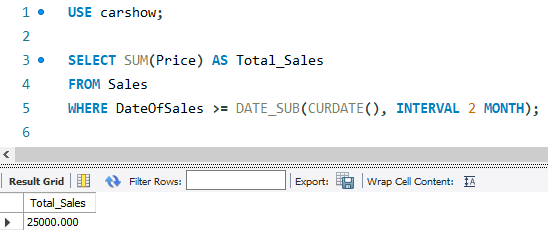
1. Выдать каталог автомобилей.

*Рисунок 3.1.1 Запрос вывода каталога автомобилей*

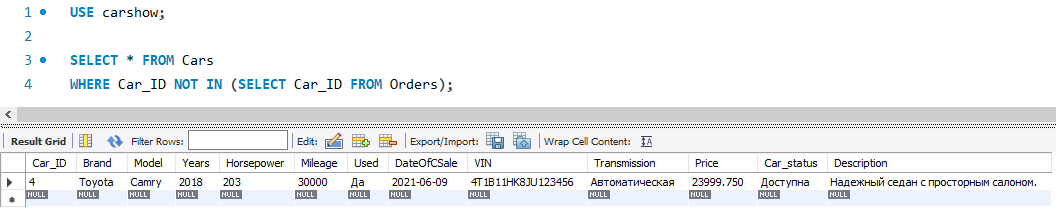
1. Cформировать список всех продаж, с указанием покупателя, продавца, автомобиля и стоимости.



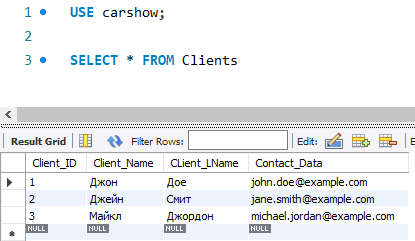
*Рисунок 3.1.2 Запрос вывода списка всех продаж*

1. Рассчитать стоимость всех продаж за последний месяц.

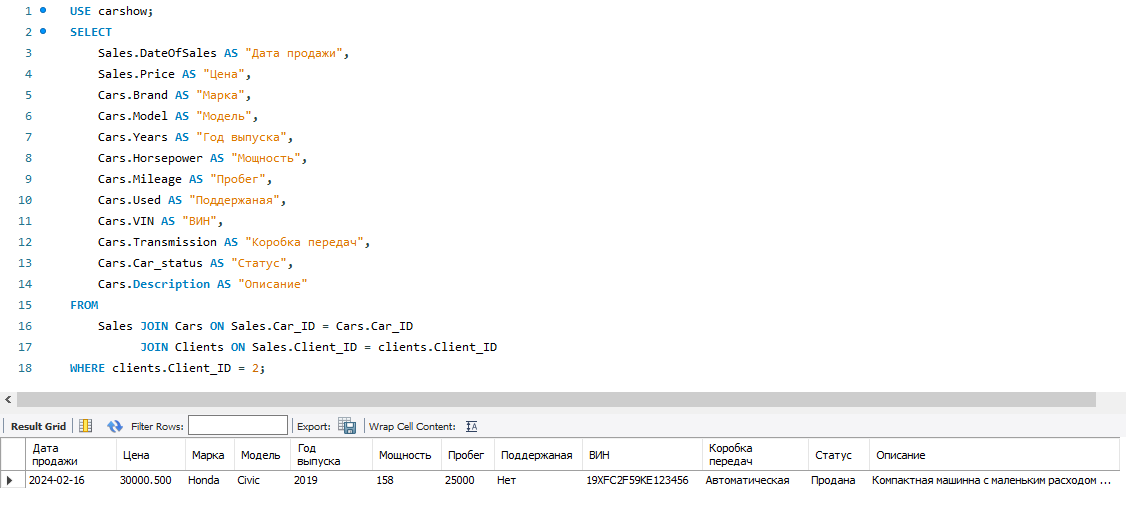
*Рисунок 3.1.3 Запрос вывода стоимости всех продаж*

1. Сформировать заказ на доставку автомобилей для салона, критерий доставки отсутствие автомобиля из каталога в салоне.

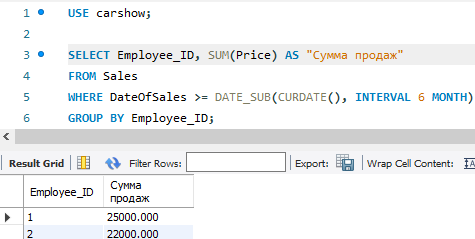
*Рисунок 3.1.4 Запрос вывода заказа на доставку*

1. Выдать список всех покупателей.

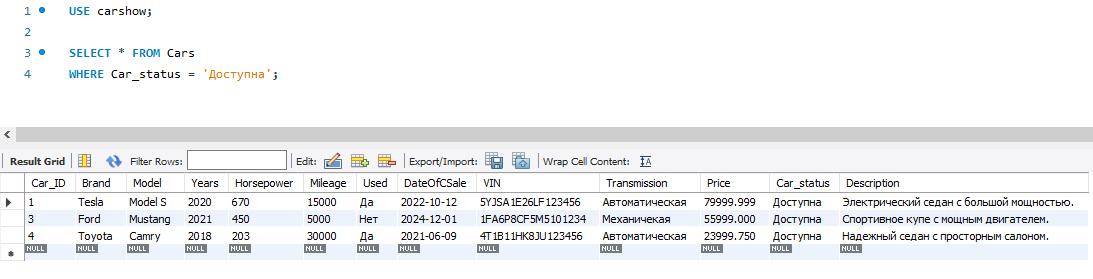
*Рисунок 3.1.5 Запрос вывода списка покупателей*

1. Для заданного покупателя вывести все его покупки.

*Рисунок 3.1.6 Запрос вывода покупок покупателя*

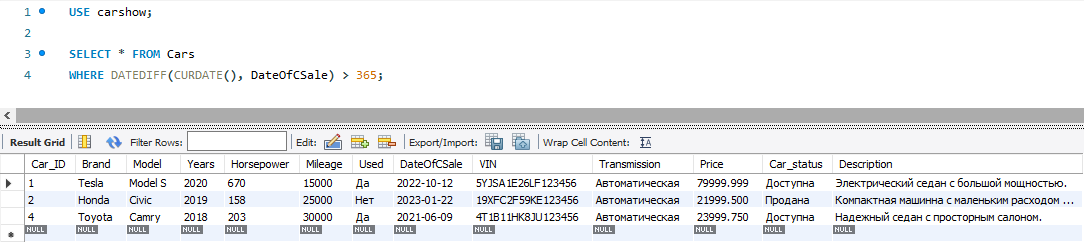
1. Для продавцов рассчитать стоимость их продаж за последний месяц.

*Рисунок 3.1.7 Запрос вывода продаж продавцов*

1. Выдать список автомобилей, имеющихся в салоне.

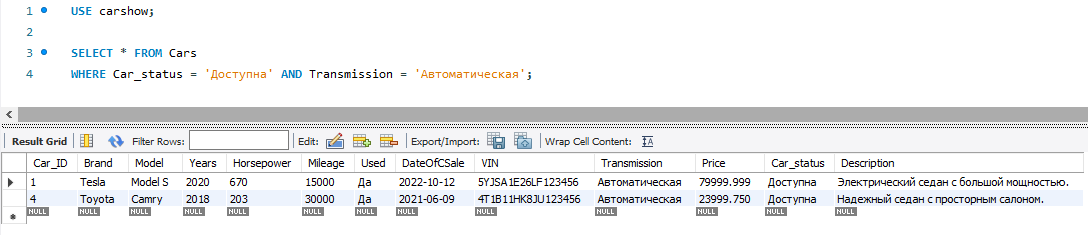
*Рисунок 3.1.8 Запрос вывода имеющихся автомобилей*

1. Выдать список автомобилей, находящихся в продаже более года.



*Рисунок 3.1.9 Запрос вывода автомобиля, находящегося в продаже более года*

1. Выдать список автомобилей заданного класса, имеющихся в наличии в салоне.



*Рисунок 3.1.10 Запрос вывода автомобиля заданного класса*

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

1. Введение в безопасность SQL Server

Безопасность SQL Server включает в себя набор мер и практик, направленных на защиту данных и ресурсов базы данных от несанкционированного доступа, утечек и других угроз. Основные компоненты безопасности включают аутентификацию, авторизацию, шифрование и аудит.

2. Управление безопасностью уровня сервера

Управление безопасностью на уровне сервера включает в себя:

**Аутентификация**: Определение, кто может подключаться к серверу. SQL Server поддерживает два режима аутентификации: Windows Authentication и SQL Server Authentication.

**Роли сервера**: Назначение ролей, таких как sysadmin, serveradmin и другие, для управления правами доступа на уровне сервера.

**Шифрование соединений**: Использование SSL для защиты данных, передаваемых между клиентом и сервером.

3. Управление участниками уровня базы данных

Управление участниками на уровне базы данных включает в себя:

**Создание пользователей и ролей**: Определение пользователей и групп, которые могут получать доступ к базе данных.

**Назначение ролей**: Использование встроенных ролей (например, db\_owner, db\_datareader) для управления доступом к объектам базы данных.

**Управление схемами**: Организация объектов базы данных в схемы для более гибкого управления доступом.

4. Управление разрешениями уровня базы данных

Управление разрешениями включает в себя:

**Назначение разрешений**: Определение, какие действия могут выполнять пользователи (например, SELECT, INSERT, UPDATE, DELETE) на уровне таблиц, представлений и других объектов.

**Использование GRANT, REVOKE и DENY**: Команды для управления разрешениями на доступ к объектам базы данных.

**Аудит разрешений**: Мониторинг и проверка разрешений для обеспечения соответствия политике безопасности.

5. Резервное копирование баз данных

Резервное копирование является критически важным для обеспечения безопасности данных:

**Типы резервного копирования**: Полное, дифференциальное и транзакционное резервное копирование.

**Планирование резервного копирования**: Настройка регулярных резервных копий для минимизации потерь данных.

**Восстановление данных**: Процессы восстановления данных из резервных копий в случае сбоя или потери данных.

6. Шифрование данных баз данных

Шифрование данных помогает защитить конфиденциальную информацию:

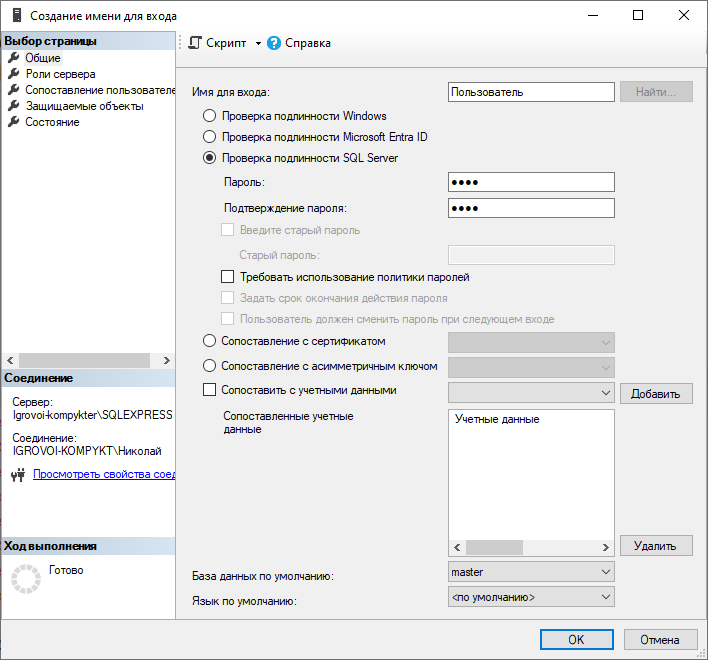
**Transparent Data Encryption (TDE)**: Шифрование данных на уровне файловой системы, что защищает данные в состоянии покоя.

**Column-level encryption**: Шифрование отдельных столбцов в таблицах для защиты чувствительных данных.

**Always Encrypted**: Позволяет шифровать данные на стороне клиента, так что даже администраторы базы данных не могут видеть нешифрованные данные.

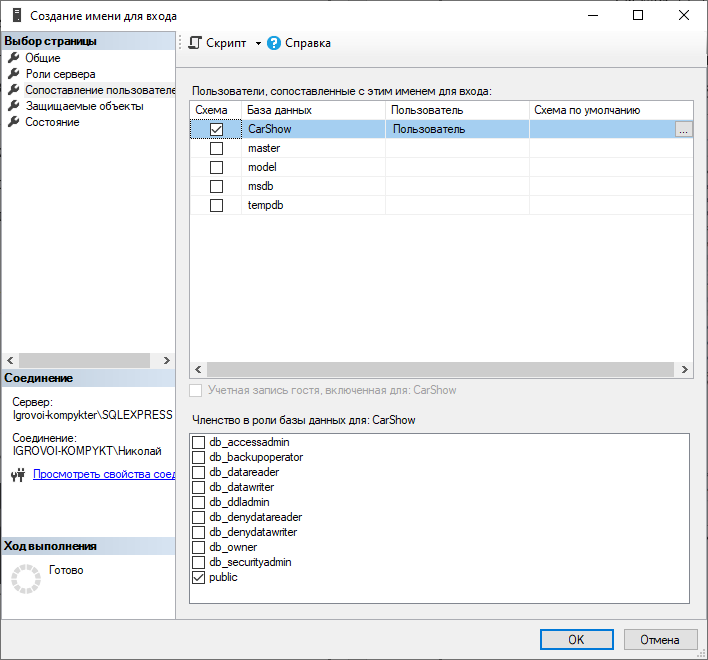
Эти аспекты безопасности SQL Server помогают обеспечить защиту данных и соответствие требованиям безопасности в организациях.

## **3.3 Управление безопасностью уровня сервера**

Создание пользователя для базы данных и присвоение ему роли.

*Рисунок 3.3.1 Добавление пользователя*

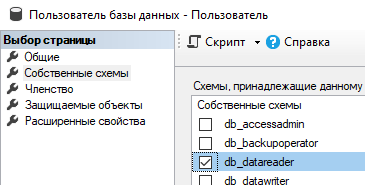
Подключение пользователя к базе данных CarShow.



*Рисунок 3.3.2 Разграничение прав доступа к базе данных*

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

Создание двух пользователей «Администратор» и «Пользователь» на уровне сервера, для дальнейшего подключения их к базе данных.



*Рисунок 3.4.1 Создание пользователей и их уровень в базе данных*

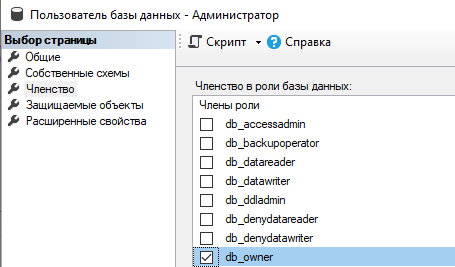
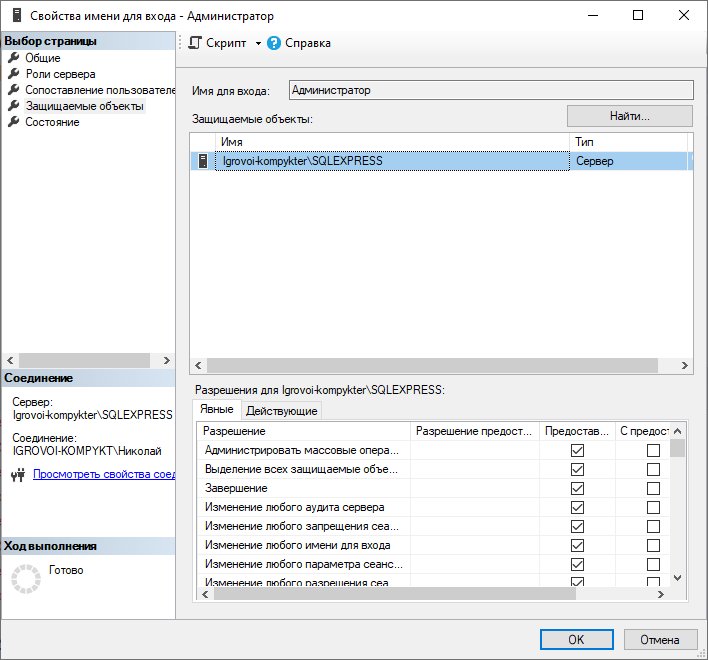
**

Рисунок 3.4.2 1 Создание пользователей и их уровень в базе данных

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

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



*Рисунок 3.5.1 Выдача прав Администратору*

## **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 B различных облачных хранилищах. Резервное копирование играет важную роль в обеспечении безопасности данных и защите от потери информации.

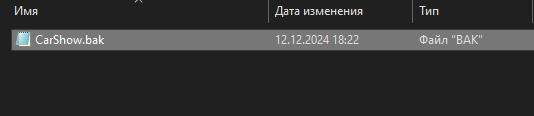
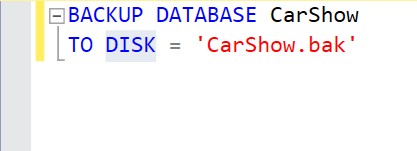
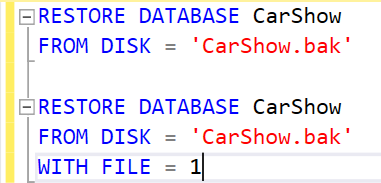
Создание BackUp для базы данных CarShow в файл CarShow.bak.

Рисунок 3.6.1 Код для создания файла восстановления

Рисунок 3.6.2 Файла восстановления

Запрос восстановления файла с BackUp базы данных

*Рисунок 3.6.3 Код для восстановления базы данных*

## **3.7 Шифрование данных баз данных**

Шифрование данных в базах данных представляет собой важный аспект безопасности, который обеспечивает защиту конфиденциальной информации от несанкционированного доступа. Этот процесс включает преобразование открытых данных в зашифрованный формат, который может быть прочитан только с использованием соответствующего ключа или пароля.

-- Создание мастер-ключа для базы данных (потребуется пароль)

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'UWU';

-- Создание сертификата для симметричного ключа

CREATE CERTIFICATE MyCertificate

WITH SUBJECT = 'Data Encryption Certificate';

-- Создание симметричного ключа с алгоритмом AES

CREATE SYMMETRIC KEY DataEncryptionKey

WITH ALGORITHM = AES\_256

ENCRYPTION BY CERTIFICATE MyCertificate;

-- Открытие симметричного ключа для использования

OPEN SYMMETRIC KEY DataEncryptionKey

DECRYPTION BY CERTIFICATE MyCertificate;

CREATE TABLE ED (

UserID INT IDENTITY(1,1) PRIMARY KEY,

Login VARCHAR(50),

EncryptedPassword VARBINARY(MAX)

);

-- Открытие симметричного ключа для использования

OPEN SYMMETRIC KEY DataEncryptionKey

DECRYPTION BY CERTIFICATE MyCertificate;

-- Вставка зашифрованных паролей в таблицу

INSERT INTO ED (Login, EncryptedPassword)

VALUES

('Администратор', ENCRYPTBYKEY(KEY\_GUID('DataEncryptionKey'), 'AdminPassword')),

('Пользователь', ENCRYPTBYKEY(KEY\_GUID('DataEncryptionKey'), 'UserPassword'));

-- Закрытие симметричного ключа после использования

CLOSE SYMMETRIC KEY DataEncryptionKey;

SELECT Login, EncryptedPassword

FROM dbo.ED;

-- Вставка зашифрованных паролей в таблицу Users в схеме dbo

INSERT INTO dbo.ED (Login, EncryptedPassword)

VALUES

('Администратор', ENCRYPTBYKEY(KEY\_GUID('DataEncryptionKey'), 'AdminPassword')),

('Пользователь', ENCRYPTBYKEY(KEY\_GUID('DataEncryptionKey'), 'UserPassword'));

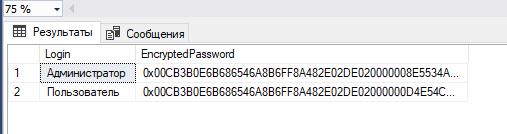


Рисунок 3.7.1 Результат шифрования