**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 «Станция технического обслуживания»**

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](#_Toc199286580)

[**ГЛАВА 1. ПРЕДПРОЕКТНАЯ ЧАСТЬ ТРЕБОВАНИЙ К ИНФОРМАЦИОННОЙ СИСТЕМЕ** 4](#_Toc199286581)

[**1.1** **Анализ предметной области** 4](#_Toc199286582)

[**1.2** **Разработка технического задания** 5](#_Toc199286583)

[**1.3** **Моделирование бизнес-процессов (IDEF0 is / to be)** 9](#_Toc199286584)

[**ГЛАВА 2. ВЫБОР ИНСТРУМЕНТАРИЯ** 12](#_Toc199286585)

[**2.1 Разработка диаграммы ERD** 12](#_Toc199286586)

[**2.2 Словарь данных** 14](#_Toc199286587)

[**2.3 СУБД SQL Server** 18](#_Toc199286588)

[**2.4 Microsoft SQL Server Management Studio** 19](#_Toc199286589)

[**ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ** 21](#_Toc199286590)

[**3.1 Разработка базы данных** 21](#_Toc199286591)

[**3.2 Введение в безопасность SQL Server** 27](#_Toc199286592)

[**3.5 Управление разрешениями уровня базы данных** 36](#_Toc199286593)

[**3.6 Резервное копирование баз данных** 39](#_Toc199286594)

[**3.7 Шифрование данных баз данных** 41](#_Toc199286595)

[**ЗАКЛЮЧЕНИЕ** 43](#_Toc199286596)

[**СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ** 44](#_Toc199286597)

# **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. Автоматизированная информационная система (АИС) станции технического обслуживания (СТО) обеспечивает автоматизацию таких процессов, как запись на обслуживание, учет наличия услуг, обработка платежей, управление договорами на выполнение работ и мониторинг состояния транспортных средств.

Станция технического обслуживания представляет собой комплексное решение, которое позволяет упростить и оптимизировать все аспекты обслуживания автомобилей — от управления процессами ремонта и технического обслуживания до взаимодействия с клиентами. АИС СТО обеспечивает интеграцию всех необходимых функций, включая управление записями клиентов, учет выполненных работ, планирование графиков механиков и анализ качества предоставляемых услуг, что в конечном итоге повышает уровень удовлетворенности клиентов и эффективность работы станции.

**Объект**: база данных «Станция технического обслуживания»

**Предмет**: анализ бизнес-процесса «Станция технического обслуживания».

**Цель работы**: разработать базу данных «Станция технического обслуживания».

**Задачи:**

* ﻿﻿провести анализ предметной области;
* выбрать инструментарий;
* ﻿﻿спроектировать базу данных;
* разработать базу данных.

# **ГЛАВА 1. ПРЕДПРОЕКТНАЯ ЧАСТЬ ТРЕБОВАНИЙ К ИНФОРМАЦИОННОЙ СИСТЕМЕ**

## **Анализ предметной области**

Станция технического обслуживания — это организация, предоставляющая услуги по ремонту и обслуживанию автомобилей. Основные услуги могут включать:

* технический осмотр;
* ремонт двигателей;
* замена масла и фильтров;
* ремонт трансмиссий;
* диагностика электрооборудования;
* шиномонтаж и балансировка.

Процессы, происходящие на СТО, можно разбить на несколько этапов:

* **Запись на обслуживание**: Клиенты могут записываться на услуги через телефон, сайт или непосредственно на месте.
* **Прием автомобиля**: Администратор принимает автомобиль, фиксирует его состояние и запрашивает информацию о необходимых услугах.
* **Диагностика**: Механики проводят диагностику автомобиля для выявления проблем.
* **Выполнение работ**: На основе диагностики выполняются необходимые работы.
* **Оплата и выдача автомобиля**: После завершения работ клиент оплачивает услуги и получает свой автомобиль.

В этой области можно выделить несколько ключевых ролей:

* **Клиенты**: Владельцы автомобилей, которые обращаются за услугами.
* **Механики**: Работники, выполняющие техническое обслуживание и ремонт.
* **Администраторы**: Персонал, занимающийся записью клиентов, управлением расписанием и документооборотом.
* **Менеджеры**: Лица, ответственные за стратегическое управление СТО и взаимодействие с поставщиками.

Требования к системе:

* простота использования интерфейса для всех категорий пользователей;
* быстрая и стабильная работа при большом количестве данных;
* разграничение доступа в зависимости от роли пользователя;
* возможность быстрого поиска и фильтрации данных.

## **Разработка технического задания**

**1.Наименование работы**: Разработка баз данных «Станция технического обслуживания».

**2. Назначение разработки**

Разработка баз данных «Станция технического обслуживания» предназначена для оптимизации процессов обслуживания автомобилей и управления ресурсами станции.

Пользователями программы выступают: клиенты, механики, администраторы, менеджеры.

Основные функции системы:

1. **Управление клиентами**:
   * Регистрация клиентов и их автомобилей;
   * Хранение истории обслуживания и контактов.
2. **Запись на обслуживание**:
   * Онлайн-запись клиентов на услуги;
   * Уведомления о предстоящих записях и напоминания.
3. **Управление услугами**:
   * Каталог доступных услуг с описаниями и ценами;
   * Возможность добавления, редактирования и удаления услуг.
4. **Учет запасов**:
   * Автоматизированный учет запчастей и расходных материалов;
   * Уведомления о необходимости пополнения запасов.
5. **Финансовый учет**:
   * Учет доходов и расходов;
   * Формирование отчетов по финансовым показателям.
6. **Отчеты и аналитика**:
   * Генерация отчетов по выполненным работам, финансовым показателям, популярности услуг и т.д;
   * Анализ данных для принятия управленческих решений.

**3. Требования к программе**

**3.1. Требования к функциональным характеристикам**

Разработка баз данных «Станция технического обслуживания» должна обеспечивать выполнение следующих функций:

■ ввод, хранение, поиск и обработка информации о транспортных средствах, клиентах и выполненных услугах;

**■** ведение журнала технического обслуживания и ремонта автомобилей;

**■**формирование отчетов, необходимых для работы сотрудников, содержащих данные о выполненных услугах, клиентах и истории обслуживания.

Нормативно-справочная информация для разработки баз данных «Станция технического обслуживания» включает в себя:

**3.2. Требования к надежности**

Разрабатываемое программное обеспечение должно иметь:

■ возможность самовосстановления после сбоев (отключения электропитания, сбои в операционной системе и т.д.);

■ ограничение несанкционированного доступа к данным;

■ возможность резервного копирования информационной базы;

■ разграничение пользовательских прав;

■ данные в базе данных должны иметь шифрование.

Предусмотреть контроль вводимой информации и блокировку некорректных действий пользователя при работе с системой.

**3.3. Требования к составу и параметрам технических средств**

Системные требования для работы программного продукта должны быть следующими: тактовая частота процессора - 1800 Гц, объем оперативной памяти 4 Гб; объем свободного дискового пространства 2 Гб; разрешение монитора 1920 х 1080.

**3.4. Требования к информационной и программной совместимости**

Программа должна работать в операционных системах Windows 10. Все формируемые отчеты должны иметь возможность экспортирования в редактор электронных таблиц MS Office Excel 2016/2021 и MS Office Word 2016/2019.

**3.5. Требования к транспортированию и хранению**

Программная документация поставляется в электронном и печатном виде.

**3.6. Специальные требования**

Программное обеспечение должно иметь дружественный интерфейс, рассчитанный на пользователя средней квалификации (с точки зрения компьютерной грамотности).

Ввиду объемности проекта задачи предполагается решать поэтапно. При этом модули программного обеспечения (ПО), созданные в разное время, должны предполагать возможность наращивания системы и быть совместимы друг с другом; поэтому документация на принятое эксплуатационное ПО должна содержать полную информацию, необходимую для работы с ним программистов. Язык программирования определяется выбором исполнителя, при этом он должен обеспечивать возможность интеграции программного обеспечения с пакетом MS Office 2010/2019.

**4. Требования к программной документации**

При выполнении операций по регистрации и учету обслуживания автомобилей, а также их диагностике и ремонту используется ручной труд, в частности, ведутся журналы учета клиентов и выполненных работ. Очевидно, что использование баз данных значительно сократит время, затрачиваемое на регистрацию автомобилей и клиентов. Для каждого автомобиля существует инвентарная карточка, в которую заносятся сведения о проведенных работах, состоянии автомобиля и его текущем местонахождении. Для получения этих сведений по конкретному автомобилю требуется не менее 7-10 минут. С использованием программы затраты времени сокращаются до 2-5 минут.

В конце каждого месяца (или другого отчетного периода) ответственный работник компании составляет отчеты о выполненных работах, состоянии автомобилей и задолженности клиентов. На эту операцию уходит 1-2 дня, то есть 6-12 часов. Формирование отчетной ведомости в компьютере займет всего 2-3 минуты. Кроме того, предполагается возможность получения отчетов за любой период времени. При ручном создании отчетов могут быть допущены ошибки; правильно составленный алгоритм разрабатываемой программы исключает вероятность ошибок.

Таким образом, автоматизация процессов учета и отчетности в системе «Станция технического обслуживания» не только значительно ускорит работу сотрудников, но и повысит точность данных, что, в свою очередь, улучшит управление процессом обслуживания автомобилей и клиентами.

Внедрение баз данных «Станция технического обслуживания» может значительно повысить эффективность работы компании. Это достигается за счет ускорения обработки заявок на обслуживание, оптимизации управления автопарком и клиентами, что снижает затраты на труд и минимизирует ошибки. Система предоставляет доступ к аналитическим данным, позволяя лучше планировать закупки запчастей, отслеживать популярность услуг и управлять клиентской базой.

Улучшение сервиса и более точное отслеживание задолженностей способствуют увеличению удовлетворенности клиентов и снижению потерь от невыполненных услуг. В результате работа компании становится более прозрачной и контролируемой, что также способствует снижению рисков и улучшению показателей работы учреждения.

## **Моделирование бизнес-процессов (IDEF0 is / to be)**

IDEF0 — методология функционального моделирования (англ. function modeling) и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является ее акцент на соподчиненность объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность (поток работ.

 Данная модель позволяет описать все основные виды процессов, как административные, так и организационные.

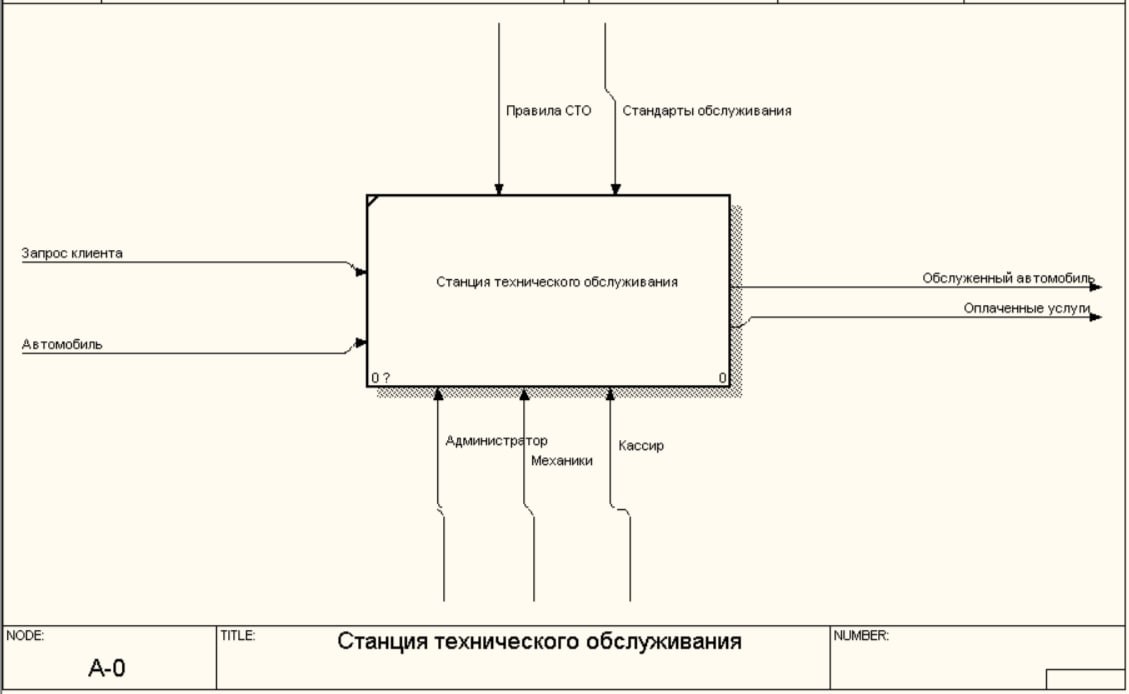
Стрелки могут быть:

* входящие – вводящие, которые ставят определенную задачу;
* исходящие – выводящие результат деятельности;
* контроль (сверху вниз) – механизмы управления (положения, инструкции и пр);
* механизмы (снизу вверх) – что используется для того, чтобы произвести необходимую работу.

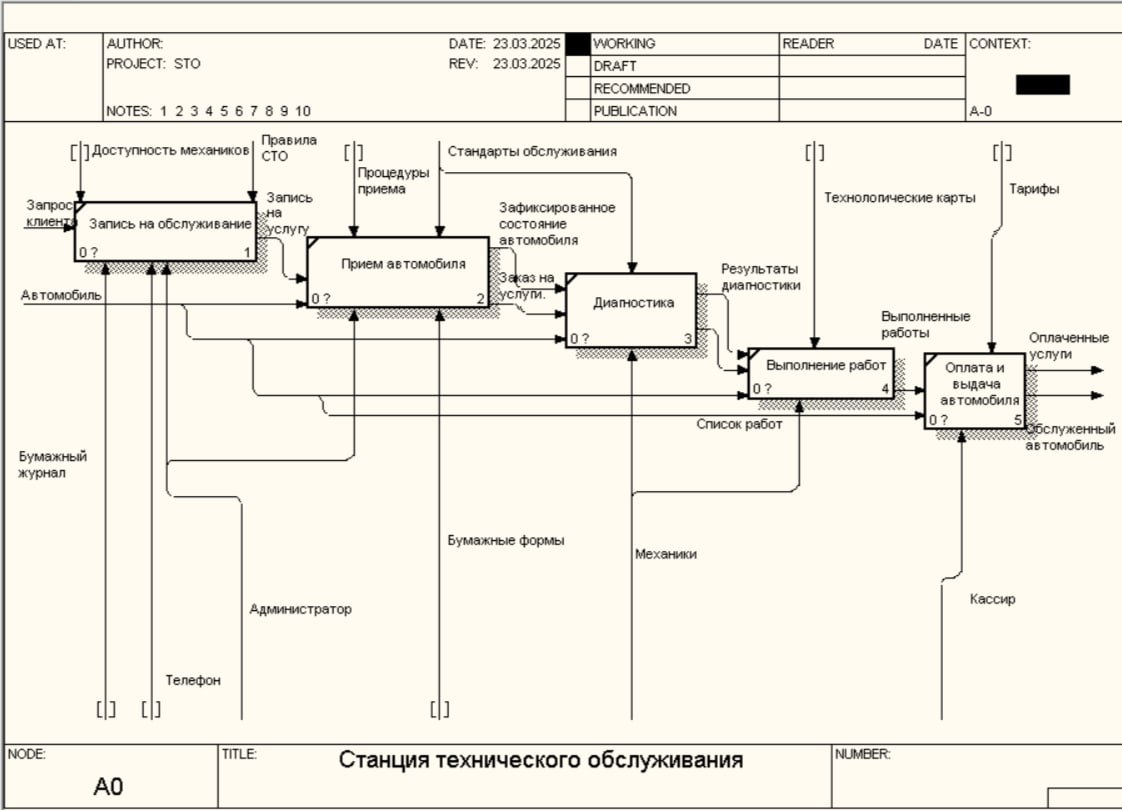
**Модель AS IS** («Как есть») — это описание процессов в том состоянии, в котором они находятся в данный момент. Она фиксирует, как сейчас устроены процессы в организации. Чтобы создать AS IS, изучают документы и регламенты, наблюдают за процессом в реальном времени и фиксируют это в схемах, проводят интервью со всеми, кто участвует в процессе.

**Модель TO BE** («Как должно быть») — это модель для описания бизнес-процессов, которая помогает представить идеальное или улучшенное состояние организации и работы всех её подразделений. С её помощью отражают, как устранить недостатки, оптимизировать процессы и внедрить улучшения. Для построения TO BE нужны данные из AS IS — определяют «зоны роста», выбирают те, которых нужно достичь в первую очередь.

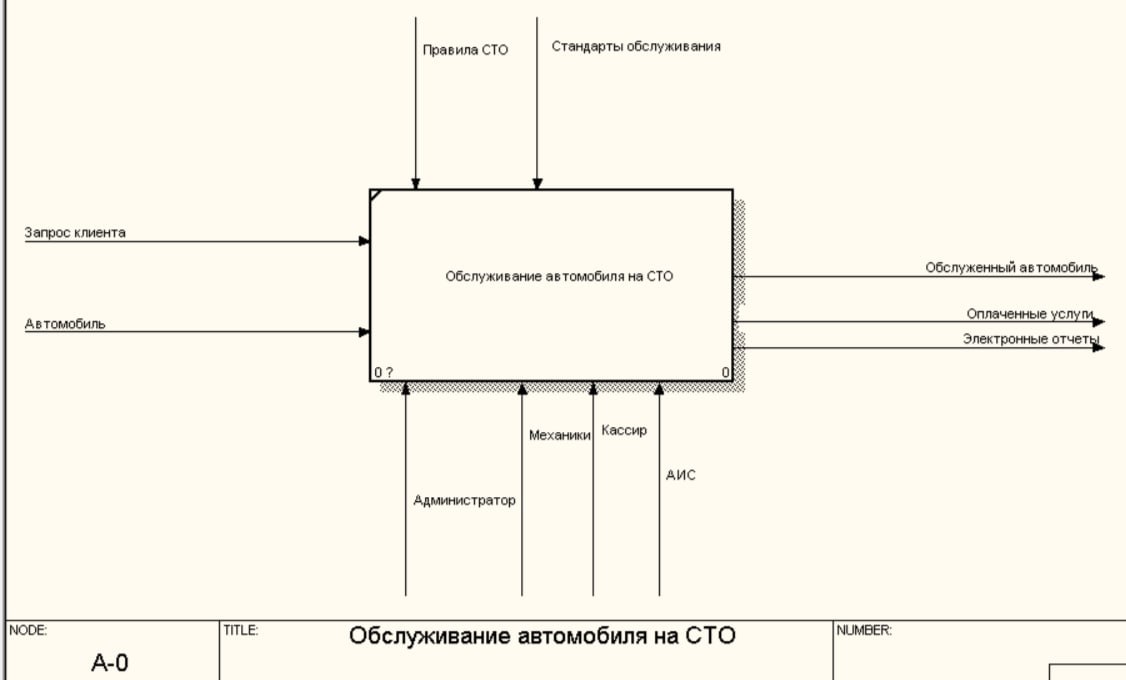
Чаще всего эти модели применяют для оптимизации бизнес-процессов в компании.

****

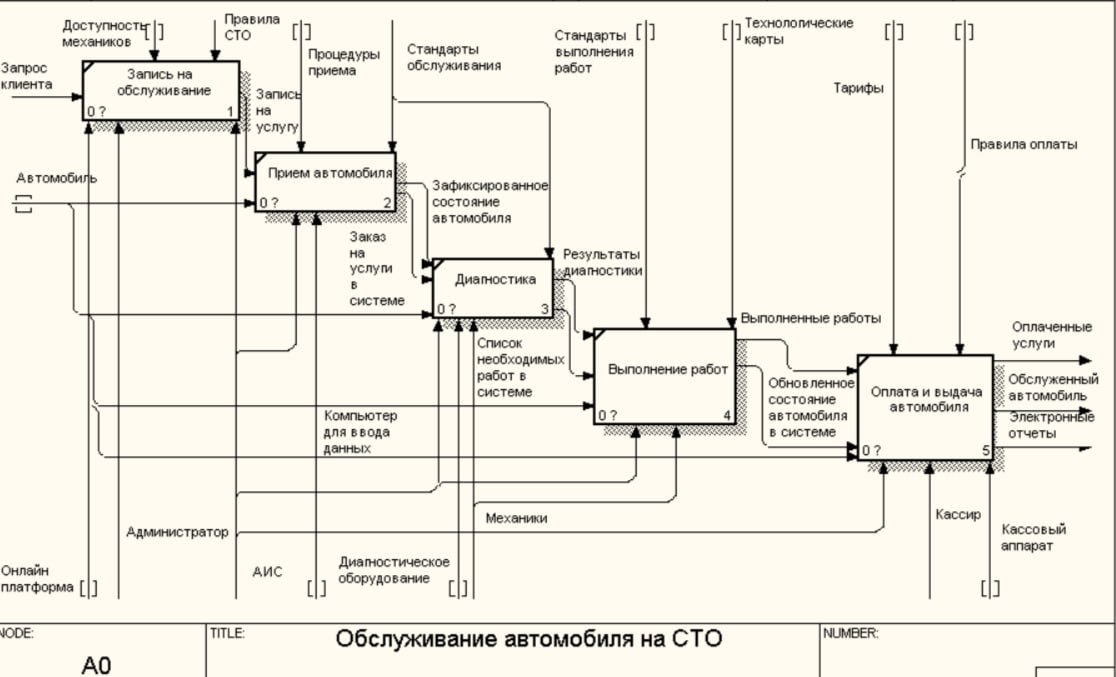
*Рис.1.3.1. IDEF(0) AS IS*

**

*Рис.1.3.2. IDEF(0) AS IS*

**

*Рис.1.3.3. IDEF(0) TO BE*

**

*Рис.1.3.4. IDEF(0) TO BE*

# **ГЛАВА 2. ВЫБОР ИНСТРУМЕНТАРИЯ**

## **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 cвязь представляет собой условное отношение между сущностями.



*Рис.2.1.1. ER-диаграмма*

## **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 |
| FK | **Client\_id** | INT | Y | Уникальный идентификатор клиента |
|  | **Model** | VARCHAR(100) | Y | Модель автомобиля |
|  | **VIN** | VARCHAR(100) | Y | ВИН номер автомобиля |
|  | **Year** | VARCHAR(4) | Y | Год автомобиля |

Таблица 2. Client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Client** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Client\_id** | INT | Y | Уникальный идентификатор клиента |
|  | **Last Name** | VARCHAR(50) | Y | Имя клиента |
|  | **First Name** | VARCHAR(50) | Y | Фамилия клиента |
|  | **Phone** | VARCHAR(15) | N | Номер телефона клиента |
|  | **Email** | VARCHAR(255) | N | Почта клиента |
|  | **Address** | VARCHAR(255) | N | Адрес клиента |

Таблица 3. Service

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Service** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Service\_id** | INT | Y | Уникальный идентификатор услуги |
|  | **Title** | VARCHAR(50) | Y | Название услуги |
|  | **Description** | VARCHAR(50) | Y | Описание услуги |
|  | **Price** | DECIMAL(10,2) | Y | Цена услуги |

Таблица 4. Mechanic

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Mechanic** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Mechanic\_id** | INT | Y | Уникальный идентификатор механика |
|  | **Last Name** | VARCHAR(50) | Y | Имя механика |
|  | **First Name** | VARCHAR(50) | Y | Фамилия механика |
|  | **Specialization** | VARCHAR(100) | Y | Специализация механика |

Таблица 5. Record For Service

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Record For Service** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Record\_id** | INT | Y | Уникальный идентификатор записи |
| FK | **Client id** | INT | Y | Уникальный идентификатор клиента |
| FK | **Car id** | INT | Y | Уникальный идентификатор автомобиля |
| FK | **Service id** | INT | N | Уникальный идентификатор услуги |
| FK | **Mechanic id** | INT | N | Уникальный идентификатор механика |
|  | **Date** | DATE | Y | Дата обслуживания |
|  | **Time** | TIME | Y | Время обслуживания |
|  | **Status** | ENUM | Y | Статус обслуживания |

Таблица 6. Work

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Work** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
| PK | **Work\_id** | INT | Y | Уникальный идентификатор работы |
| FK | **Record id** | INT | N | Уникальный идентификатор записи |
| FK | **Mechanic id** | INT | N | Уникальный идентификатор механика |
|  | **Start Time** | DATETIME | N | Начало работы |
|  | **End Time** | DATETIME | N | Окончание работы |
|  | **Description** | ENUM | Y | Описание работы |

## **2.3 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.

## **2.4 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 3. РАЗРАБОТКА БАЗЫ ДАННЫХ**

## **3.1 Разработка базы данных**

Согласно техническому заданию, разработаем базу данных «Станция технического обслуживания», которая содержит информацию о клиентах, механики, администраторы, менеджеры.

## Создание базы данных

CREATE DATABASE ServiceStation;

## Используем созданную базу данных

USE ServiceStation;

## Таблица для клиентов

CREATE TABLE Clients (

ClientID INT AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Phone VARCHAR(15),

Email VARCHAR(255),

Address VARCHAR(255),

);

## Таблица для механиков

CREATE TABLE Mechanics (

MechanicID INT AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Specialization VARCHAR(100), NOT NULL,

);

## Таблица для администраторов

CREATE TABLE Administrators (

AdminID INT AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Phone VARCHAR(15),

Email VARCHAR(100),

HireDate DATETIME DEFAULT CURRENT\_TIMESTAMP

);

## Таблица для менеджеров

CREATE TABLE Managers (

ManagerID INT AUTO\_INCREMENT PRIMARY KEY,

FirstName VARCHAR(50) NOT NULL,

LastName VARCHAR(50) NOT NULL,

Phone VARCHAR(15),

Email VARCHAR(100),

HireDate DATETIME DEFAULT CURRENT\_TIMESTAMP

);

## Таблица для записей о ремонтах

CREATE TABLE Repairs (

RepairID INT AUTO\_INCREMENT PRIMARY KEY,

ClientID INT,

MechanicID INT,

RepairDate DATETIME DEFAULT CURRENT\_TIMESTAMP,

Description TEXT,

Cost DECIMAL(10, 2),

FOREIGN KEY (ClientID) REFERENCES Clients(ClientID),

FOREIGN KEY (MechanicID) REFERENCES Mechanics(MechanicID)

);

Согласно по техническому заданию, внесли данные в таблицы:

## Вставка данных в таблицу Clients

INSERT INTO Clients (FirstName, LastName, Phone, Email, Address) VALUES

('Иван', 'Иванов', '1234567890', 'ivan@example.com', 'Москва, ул. Ленина, 1'),

('Петр', 'Петров', '0987654321', 'petr@example.com', 'Москва, ул. Пушкина, 2'),

('Сидор', 'Сидоров', '1122334455', 'sidor@example.com', 'Москва, ул. Гоголя, 3');

## Вставка данных в таблицу Mechanics

INSERT INTO Mechanics (FirstName, LastName, Specialization) VALUES

('Алексей', 'Смирнов', 'Двигатели'),

('Олег', 'Кузнецов', 'Тормоза'),

('Дмитрий', 'Попов', 'Электрика');

## Вставка данных в таблицу Administrators

INSERT INTO Administrators (FirstName, LastName, Phone, Email, Hire Date) VALUES

('Анна', 'Сидорова', '2233445566', 'anna@example.com', '2025-02-12'),

('Мария', 'Петрова', '3344556677', 'maria@example.com', '2023-04-16');

## Вставка данных в таблицу Managers

INSERT INTO Managers (FirstName, LastName, Phone, Email, Hire Date) VALUES

('Сергей', 'Иванов', '4455667788', 'sergey@example.com', '2019-08-29'),

('Елена', 'Кузнецова', '5566778899', 'elena@example.com', ‘2017-05-27’);

## Вставка данных в таблицу Repairs

INSERT INTO Repairs (ClientID, MechanicID, RepairDate, Description, Cost) VALUES

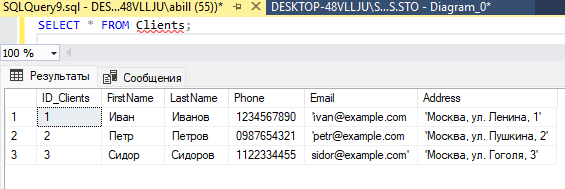
(1, 1, (13.10.2025), 'Замена масла', 1500.00),

(2, 2, (14.07.2025), 'Замена тормозов', 3000.00),

(3, 3, (13.03.2025), 'Ремонт электрики', 2000.00);

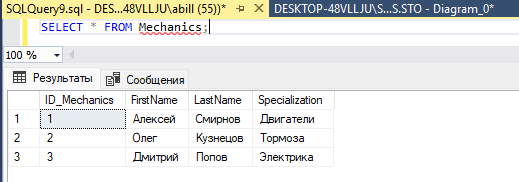
Согласно по техническому заданию, вывели следующие запросы:

1. Выборка всех клиентов.



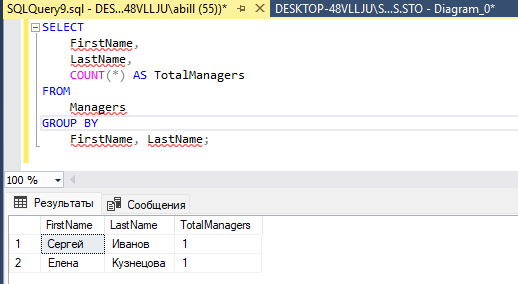
*Рисунок 3.1.1 Запрос выборки всех клиентов*

1. Выборка всех механиков с их специализациями.



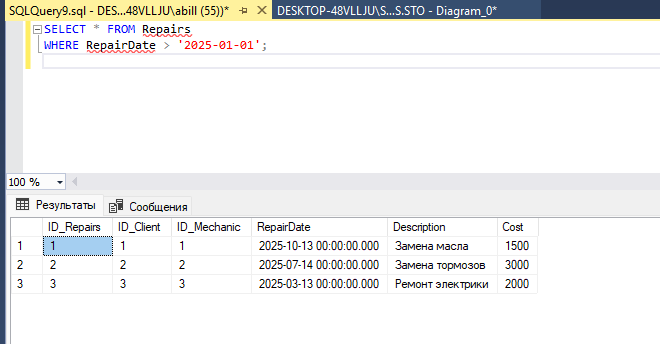
*Рисунок 3.1.2 Запрос выборки всех механиков с их специализациями.*

3. Получение информации о менеджерах и их количеств.



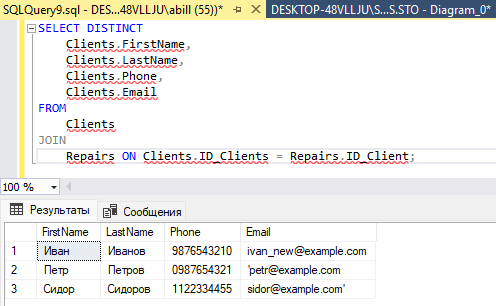
*Рисунок 3.1.3 Запрос на получение информации о менеджерах и их количеств.*

4. Получение всех ремонтов, проведенных после определенной даты.

****

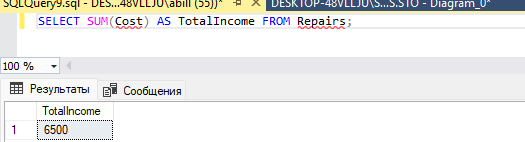
*Рисунок 3.1.4 Запрос на получение всех ремонтов, проведенных после определенной даты*

5. Получение информации о клиентах, у которых были ремонты

****

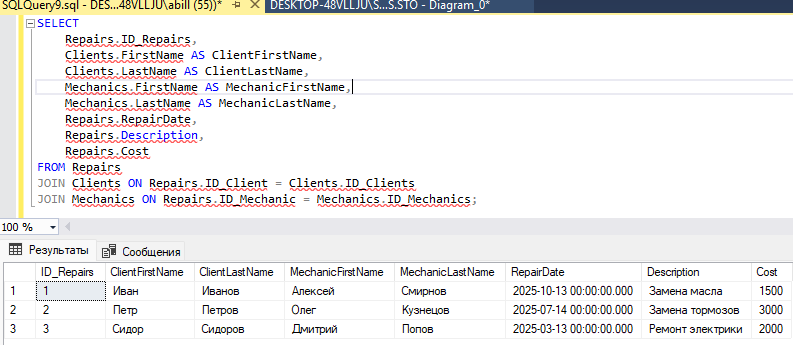
*Рисунок 3.1.5 Запрос на получение информации о клиентах, у которых были ремонты*

* 1. Получение общего дохода от всех ремонтов

****

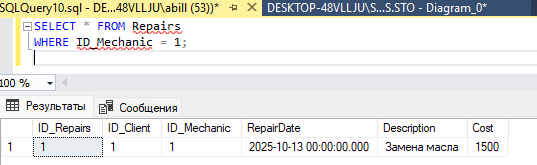
*Рисунок 3.1.6 Запрос на получение общего дохода от всех ремонтов*

* 1. Получение всех записей о ремонтах с информацией о клиентах и механиках

**

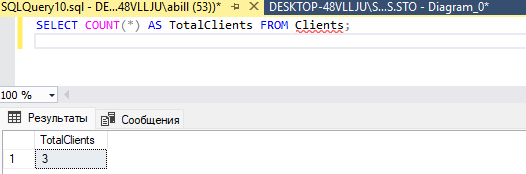
*Рисунок 3.1.7 Запрос на получение всех записей о ремонтах с информацией о клиентах и механиках*

* 1. Получение всех ремонтов, выполненных механиком с ID = 1



*Рисунок 3.1.8 Запрос на получение всех ремонтов, выполненных механиком с ID = 1*

* 1. Подсчет количества клиентов



*Рисунок 3.1.9 Запрос на получение количества клиентов*

## **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**

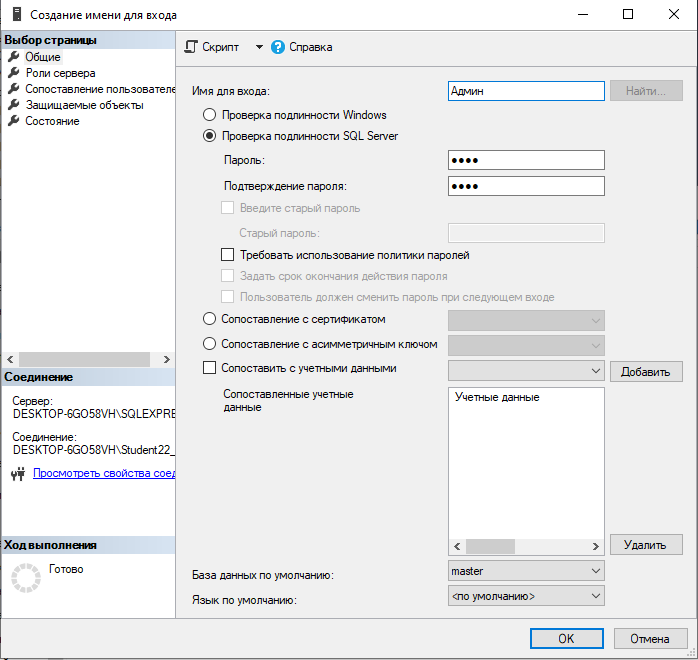
Регулярное обновление SQL Server и применение патчей безопасности помогают защищать систему от известных уязвимостей.

**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.

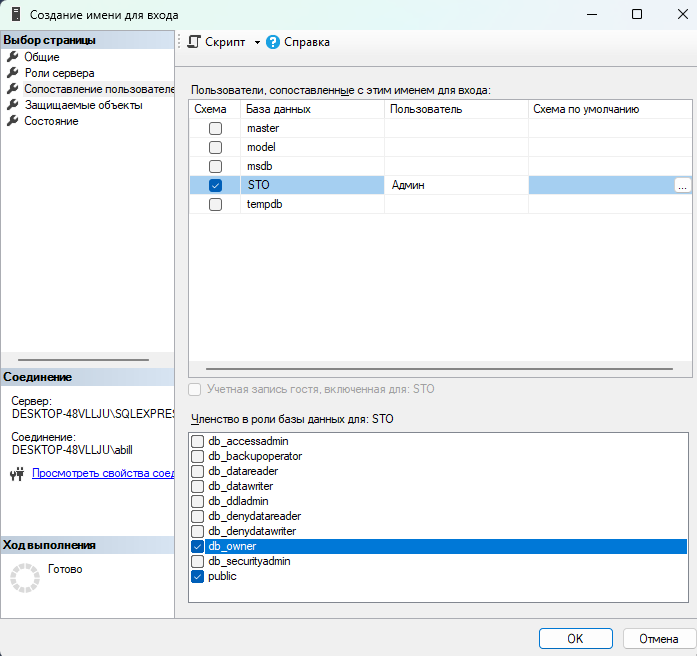
**3.3 Управление безопасностью уровня сервера**

Создание пользователя для базы данных и присвоение ему роли.

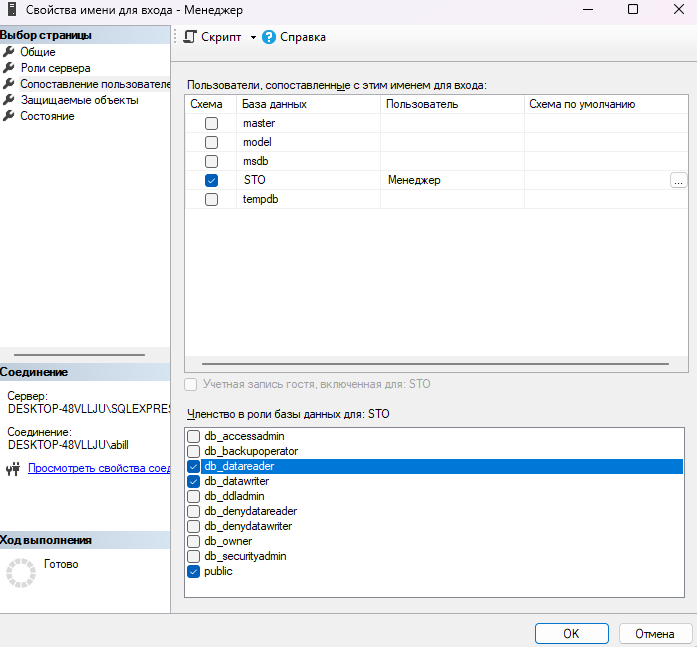


*Рисунок 3.3.1 Добавление пользователя*

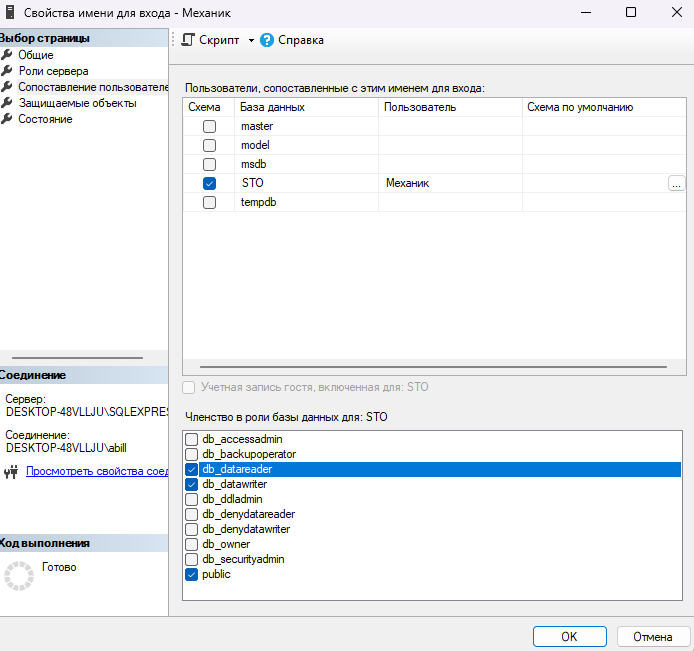
Подключение пользователя к базе данных STO.



*Рисунок 3.3.2 Разграничение прав доступа к базе данных для Администратора*

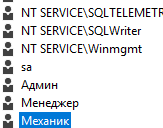


*Рисунок 3.3.3 Разграничение прав доступа к базе данных для Менеджера*



*Рисунок 3.3.4 Разграничение прав доступа к базе данных для Механика*

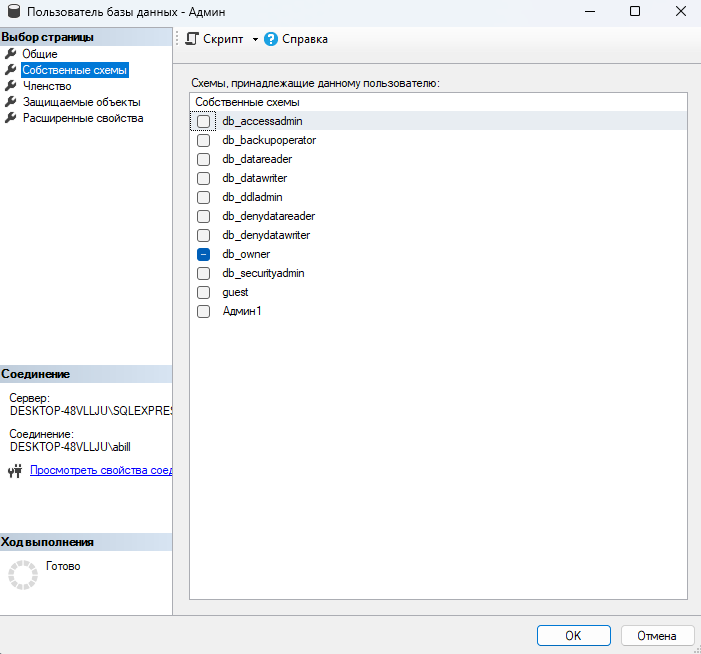
Добавили трех пользователей (Админ, Менеджер, Механик):



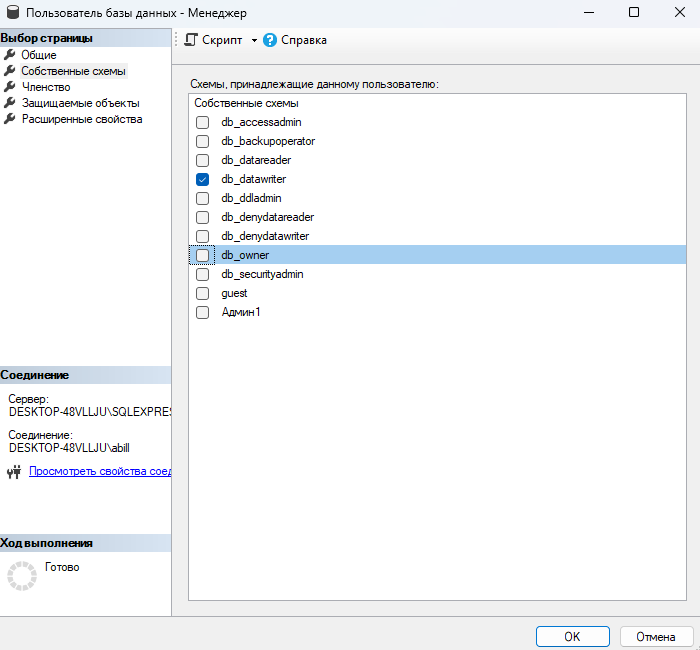
*Рисунок 3.3.5 Пользователи базы данных*

**3.4 Управление участниками уровня базы данных**

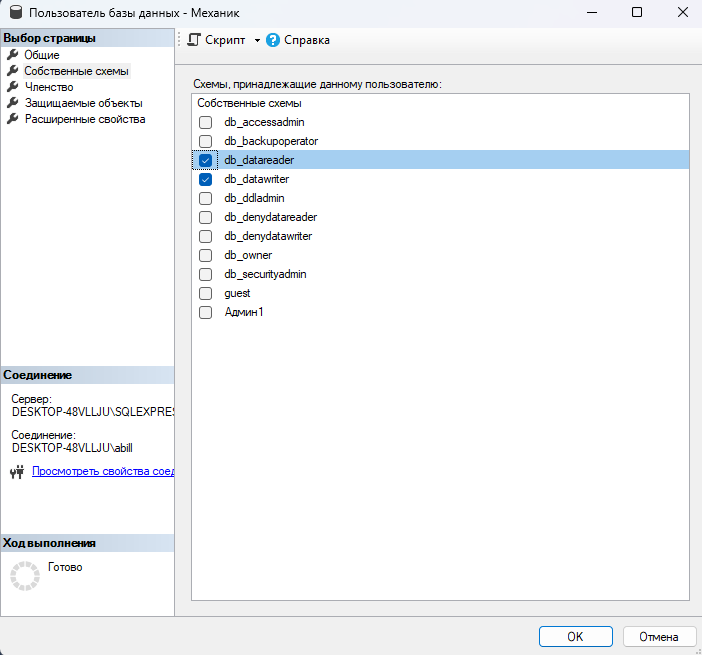
Создание трёх пользователей Админ, Менеджер и Механик на уровне сервера, для дальнейшего подключения их к базе данных.



*Рисунок 3.4.1 Уровень в базе данных Администратора*

****

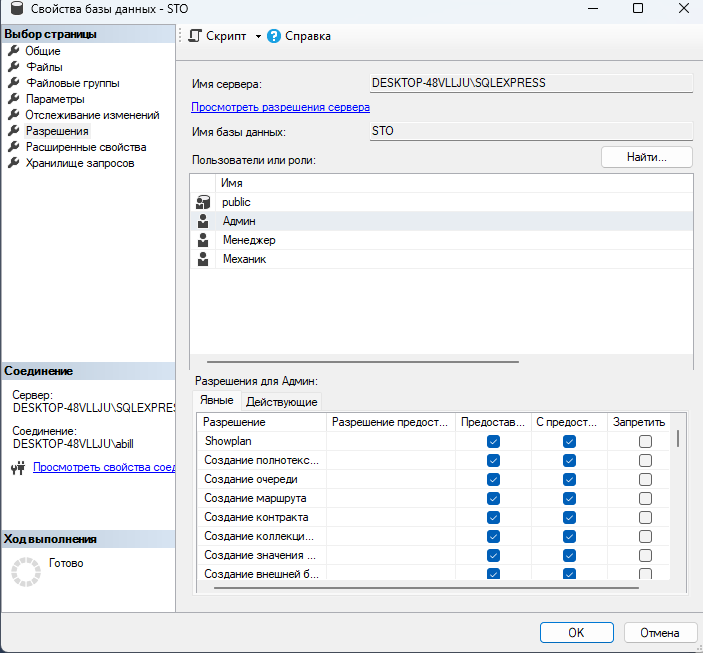
*Рисунок 3.4.2 Уровень в базе данных Менеджера*



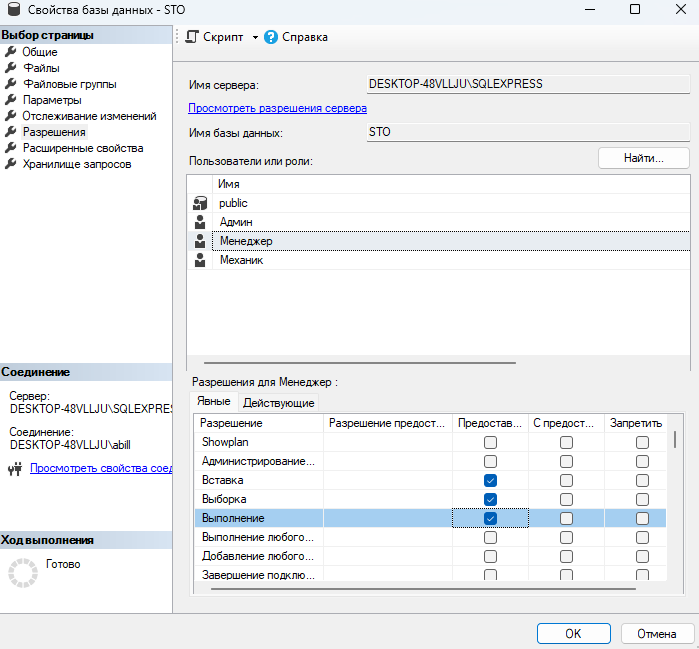
*Рисунок 3.4.2 Уровень в базе данных Механика*

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

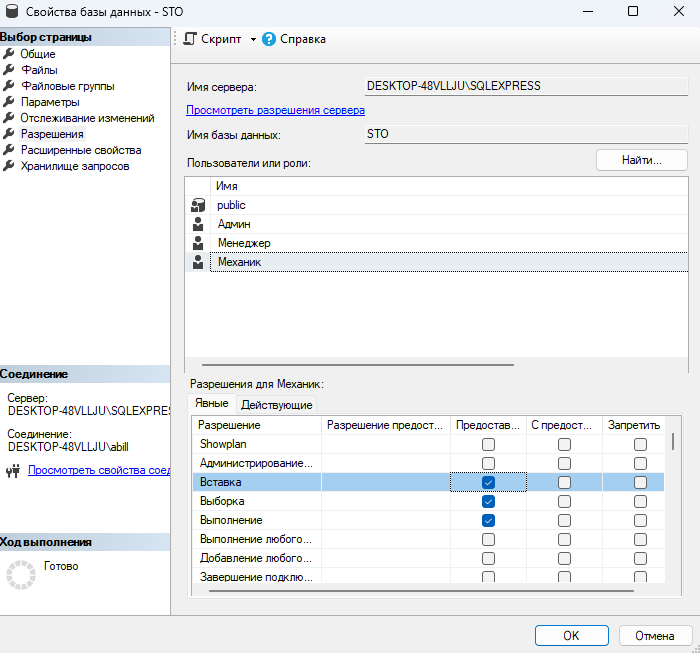
SQL Server использует системную базу данных для поддержания внутренних метаданных. Администратор базы данных сервера должен данных уметь управлять ими, а пользователь правильно ими пользоваться. В базе данных выдали права доступа Админу, Менеджеру и Механику к таблицам и права пользования функциями в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



*Рисунок 3.5.1 Выдача прав Администратору*

**

*Рисунок 3.5.3 Выдача прав Менеджеру*

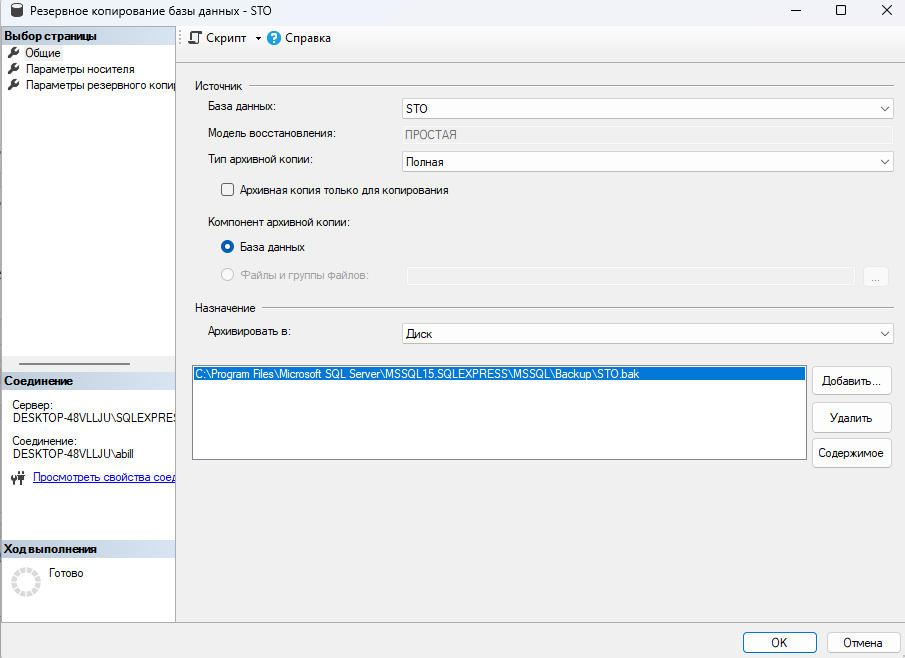
****

*Рисунок 3.5.3 Выдача прав Механику*

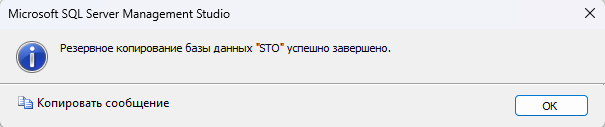
## **3.6 Резервное копирование баз данных**

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.

Резервное копирование базы данных STO:

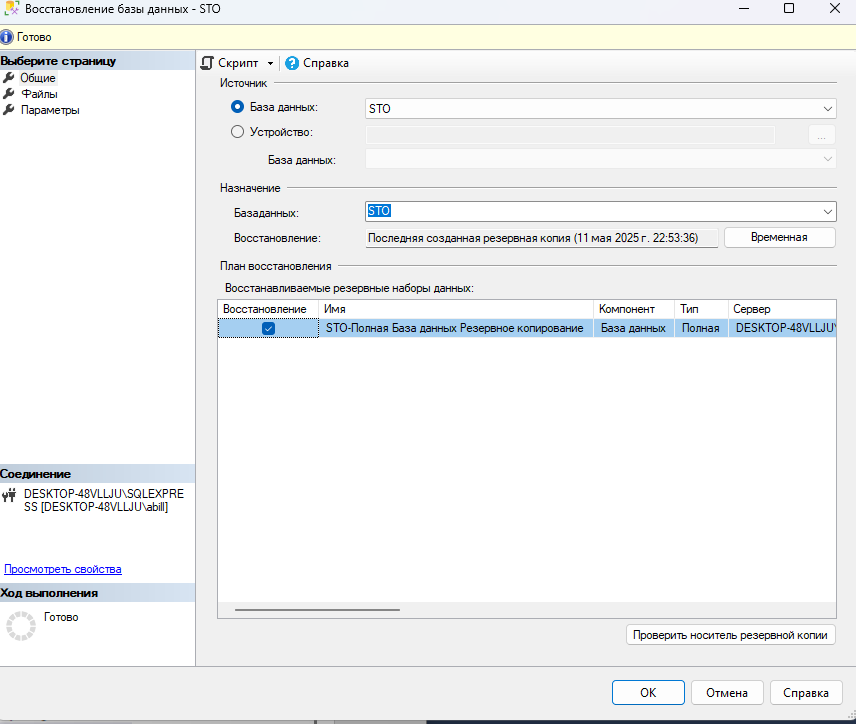


*Рисунок 3.6.1 Создание резервной копии базы данных*



*Рисунок 3.6.2 Успешное создание резервной копии базы данных*

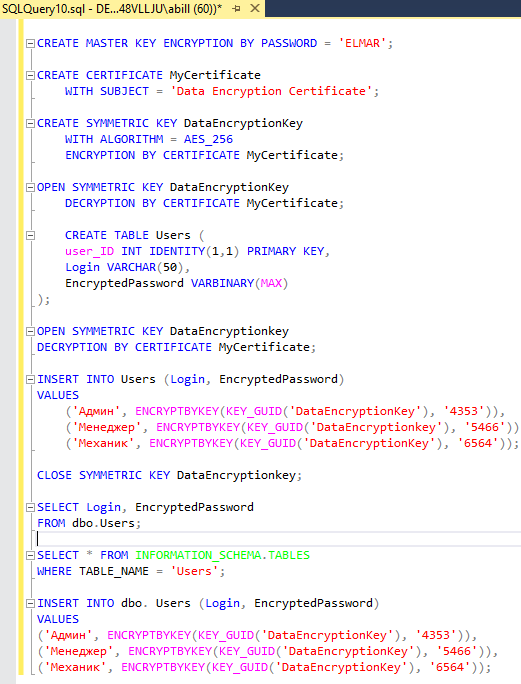
Восстановление резервной копии базы данных STO:



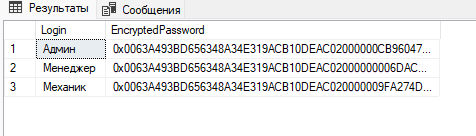
*Рисунок 3.6.4 Восстановление резервной копии базы данных*

## **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.



*Рисунок 3.7.1 Запрос на шифрование*



*Рисунок 3.7.2 Зашифрованные данные*

## **ЗАКЛЮЧЕНИЕ**

В результате выполнения данного курсового проекта будет разработана эффективная база данных для станции технического обслуживания, которая обеспечит надежное хранение и управление информацией о ремонте и обслуживании транспортных средств. Созданная структура базы данных позволит автоматизировать процессы учета, что, в свою очередь, приведет к улучшению качества предоставляемых услуг и оптимизации работы персонала СТО.

При выполнении данной работы, а именно базы данных «Станция технического обслуживания», были разработаны диаграммы Entity Relationship Diagram, была составлена таблица Data Dictionary, а также спроектирована база данных в приложении Microsoft SQL Server Management Studio. Это обеспечит удобный доступ к данным о клиентах, автомобилях, выполненных работах и запасных частях, что повысит эффективность работы станции.

Ссылка на репозиторий: <https://github.com/Elmar011/DIPLOM>

# **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/](https://hsbi.hse.ru/articles/server-baz-dannykh-sql-server/" \o "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>
10. Как устроен язык SQL и почему он так востребован // <https://practicum.yandex.ru/blog/chto-takoe-sql/?utm_source=yandex&utm_medium=cpc&utm_campaign=Yan_Perfmax_RF_other_Resk_b2c_Gener_Regular_Blog_tCPA_460&utm_content=sty_search:s_yandex.ru:cid_117415262:gid_5532382425:pid_54117132014:aid_16780971780:crid_0:rid_54117132014:p_2:pty_dynamic_places:mty_syn:mkw_:dty_desktop:cgcid_0:rn_Республика%20Коми:rid_10939&utm_term=&yclid=6867757352264400895>