**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 «Автотранспортное предприятие»**

Выполнил студент гр. ИСП-20 / /Погожев Александр Сергеевич/

(подпись) (Ф.И.О.)

**ОЦЕНКА:\_\_\_\_\_\_\_\_\_\_\_\_\_**

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

**ПРОВЕРИЛ**

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

(подпись) (Ф.И.О.)

**СОДЕРЖАНИЕ**

[ВВЕДЕНИЕ 3](#_Toc152277106)

[ГЛАВА 1. ВЫБОР ИНСТРУМЕНТАРИЯ 5](#_Toc152277107)

[1.1 СУБД SQL Server 5](#_Toc152277108)

[1.2 Microsoft SQL Server Management Studio 7](#_Toc152277109)

[1.3 Entity Framework 8](#_Toc152277110)

[ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ 12](#_Toc152277111)

[2.1 Разработка диаграммы ERD 12](#_Toc152277112)

[2.2 Разработка базы данных 14](#_Toc152277113)

[2.3 Словарь данных 20](#_Toc152277114)

[ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ 22](#_Toc152277115)

[3.1 Введение в безопасность 22](#_Toc152277116)

[3.2 Управление безопасностью уровня сервера 22](#_Toc152277117)

[3.3 Управление участниками уровня базы данных 24](#_Toc152277118)

[3.4 Управление разрешениями уровня базы данных 25](#_Toc152277119)

[3.5 Шифрование данных базы данных 26](#_Toc152277120)

[ЗАКЛЮЧЕНИЕ 27](#_Toc152277121)

[СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ 29](#_Toc152277122)

# ВВЕДЕНИЕ

**Автотранспортное предприятие (АТП)** - организация, осуществляющая перевозки автомобильным транспортом, а также хранение, техническое обслуживание и ремонт подвижного состава. АТП решает вопросы, связанные с доставкой, т.е. выбор вида транспорта, методы организации перевозок, тип транспортных средств, участие в организации погрузочно-разгрузочных работ, использование современных технологий в организации размещения, учета товаров и запасов на складах и терминалах.

Грузоперевозки всегда были и будут актуальны так как перевозки на большие расстояния и с большим объёмом грузов требуют специальный транспорт, способный перевозить большие объёмы грузов. Так же грузоперевозки являются очень важной частью экономики страны, без которых многие товары, привезённые из других стран, не доставлялись бы в дальние уголки нашей страны. Ранее все операции записывались в путевые журналы и журналы учёта заказов. Автоматизированная информационная система необходима для ускорения получения, обработки и выполнения заказов.

**Объект** – база данных информационной системы “Автотранспортное предприятие”.

**Предмет** – автоматизация бизнес-процессов автотранспортного предприятия.

**Цель** – разработать базу данных информационной системы автотранспортного предприятия.

**Задачи:**

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

При написании курсового проекта использовались методы: теоретический (анализ документации), эмпирический метод (изучение сред разработки).

Практическая значимость заключается в том, что данное приложение можно использовать для внедрения в информационную систему автотранспортного предприятия.

# 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.

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**

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, Базы данных SQL Azure, Управляемого экземпляра SQL Azure, SQL Server на виртуальной машине Azure и Azure Synapse Analytics. 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.

Среда SQL Server Management Studio предназначена для доступа к службам Службы Analysis Services, Integration Services и Reporting Services, а также для их настройки, администрирования и управления ими. Хотя все три технологии бизнес-аналитики полагаются на среду SQL Server Management Studio, административные задачи, связанные с каждой из этих технологий, несколько отличаются.

Среда SQL Server Management Studio позволяет управлять объектами Службы Analysis Services, например, выполнять их резервное копирование и обработку.

Среда Management Studio позволяет создавать проекты скриптов служб Службы Analysis Services, в которых выполняются разработка и сохранение скриптов с использованием многомерных выражений (MDX), расширений интеллектуального анализа данных (DMX) и XML для аналитики (XMLA). Проекты скриптов служб Службы Analysis Services используются для выполнения задач управления или повторного создания баз данных, кубов и других объектов в экземплярах служб Службы Analysis Services. Например, можно разработать скрипт XMLA в проекте скрипта служб Службы Analysis Services, который создает объекты непосредственно в существующем экземпляре служб Службы Analysis Services. Проекты скриптов служб Службы Analysis Services могут быть сохранены в составе решения и интегрироваться с контролем исходного кода.

SQL Server Management Studio (SSMS) — утилита из Microsoft SQL Server 2005 и более поздних версий для конфигурирования, управления и администрирования всех компонентов Microsoft SQL Server. Утилита включает скриптовый редактор и графическую программу, которая работает с объектами и настройками сервера.

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

## **1.3 Entity Framework**

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

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

Многолетним и общим подходом к разработке является подход, при котором построение приложения или службы представляет собой его разделение на три части: модель домена, логическую модель и физическую модель. Модель домена определяет сущности и связи в моделируемой системе. Логическая модель для реляционной базы данных обеспечивает нормализацию сущностей и связей в целях создания таблиц с ограничениями внешнего ключа. В физической модели учитываются возможности конкретной системы обработки данных, путем определения зависящих от ядра базы данных подробных сведений о хранении данных, которые касаются секционирования и индексирование.

Физическая модель совершенствуется администраторами базы данных в целях повышения производительности, но программисты, которые разрабатывают код приложения, в основном вынуждены ограничиваться работой с логической моделью, подготавливая SQL-запросы и вызывая хранимые процедуры. Модели домена в основном используются как инструмент для представления и обмена мнениями о требованиях к приложению, поэтому чаще всего служат в качестве практически не изменяющихся схем, которые рассматриваются и обсуждаются на ранних стадиях проекта, после чего выходят из сферы внимания. Во многих коллективах разработчиков принято пропускать этап создания концептуальной модели и начинать с определения таблиц, столбцов и ключей в реляционной базе данных.

Entity Framework дает жизнь модели, позволяя разработчикам запрашивать сущности и связи в модели предметной области (называемой концептуальной моделью в Entity Framework), в то же время полагаться на Entity Framework преобразования этих операций в команды, относящиеся к источнику данных. Это позволяет отказаться от применения в приложениях жестко заданных зависимостей от конкретного источника данных.

При работе в режиме Code First концептуальная модель сопоставлена с режимом хранения в коде. Entity Framework может вычислять концептуальную модель на основе типов объектов и дополнительных конфигураций, которые вы определяете. Метаданные сопоставления формируются во время выполнения на основе сочетания определений типов домена и дополнительной информации о конфигурации, которая указана в коде. Entity Framework создает базу данных по мере необходимости на основе метаданных.

При работе со средствами работы с моделью EDM концептуальная модель, модель хранения и сопоставление между ними выражены в схемах на основе XML и определены в файлах с именами с соответствующими расширениями.

* Язык CSDL определяет концептуальную модель. Язык CSDL — это реализация EDMEntity Framework. Расширение файла - CSDL.
* Язык SSDL определяет модель хранения данных, которая также называется логической моделью. Расширение файла - SSDL.
* Язык MSL определяет сопоставление модели хранения и концептуальной модели. Расширение файла - MSL.

Модель хранения и сопоставления при необходимости могут быть изменены без изменения концептуальной модели, классов данных и кода приложения. Модели хранения зависят от поставщика, поэтому можно работать с согласованной концептуальной моделью через различные источники данных.

Entity Framework использует эти файлы модели и сопоставления для создания, чтения, обновления и удаления операций с сущностями и связями в концептуальной модели в эквивалентных операциях в источнике данных. Entity Framework даже поддерживает сопоставление сущностей в концептуальной модели с хранимыми процедурами в источнике данных.

# ГЛАВА 2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

## **2.1 Разработка диаграммы ERD**

Схема «сущность-связь» (также ERD или ER-диаграмма) — это разновидность блок-схемы, где показано, как разные «сущности» (люди, объекты, концепции и так далее) связаны между собой внутри системы. ER-диаграммы чаще всего применяются для проектирования и отладки реляционных баз данных в сфере образования, исследования и разработки программного обеспечения и информационных систем для бизнеса. ER-диаграммы (или ER-модели) полагаются на стандартный набор символов, включая прямоугольники, ромбы, овалы и соединительные линии, для отображения сущностей, их атрибутов и связей. Эти диаграммы устроены по тому же принципу, что и грамматические структуры: сущности выполняют роль существительных, а связи — глаголов.

Под понятием «сущности» подразумеваются объекты или понятия, несущие важную информацию. С точки зрения грамматики, они, как правило, обозначаются существительными, например, «товар», «клиент», «заведение» или «промоакция».

Связи используются в схемах «сущность-связь» для обозначения взаимодействия между двумя сущностями. Грамматически связи, как правило, выражаются глаголами, например, «назначить», «закрепить», «отследить», и несут полезную информацию, которую невозможно получить, опираясь только на типы сущностей.

ERD-атрибуты характеризуют сущности, позволяя пользователям лучше разобраться в устройстве базы данных. Атрибуты содержат информацию о сущностях, выделенных в концептуальной ER-диаграмме.



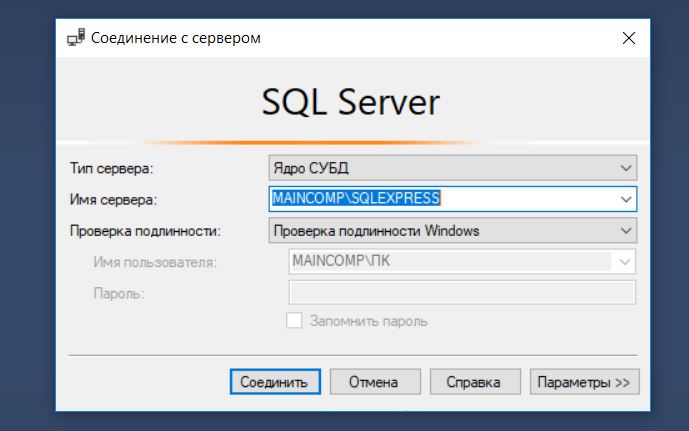
*Рисунок 2.1. ERD-Диаграмма*

На данной диаграмме представлены сущности Customer, Order, Client Manager и Car Manager, которые взаимодействуют между собой. В сущностях имеются различные атрибуты: Id, FirstName, SecondName, MiddleName, Phone, Password, Name, DateOfOrder, Weight и DateOfBirthday. Между сущностями проставлены отношения: Многий к многим, а также Один ко Многим.

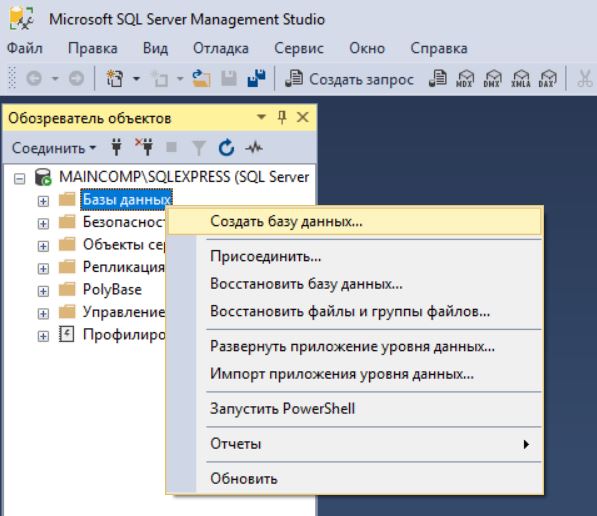
## **2.2 Разработка базы данных**

SQL Server Management Studio (SSMS) — это интегрированная среда для управления любой инфраструктурой SQL. Используйте SSMS для доступа, настройки, администрирования, администрирования и разработки всех компонентов SQL Server, базы данных Azure SQL, Управляемый экземпляр SQL Azure, SQL Server на виртуальной машине Azure и Azure Synapse Analytics. 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.

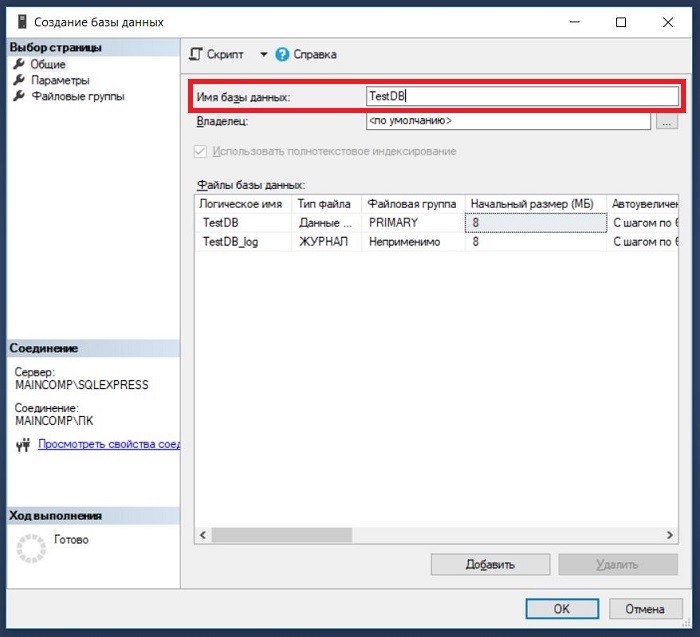
Создание базы данных в SQL Server Management Studio:

1. Запустить среду SQL Server Management Studio и подключиться к серверу.

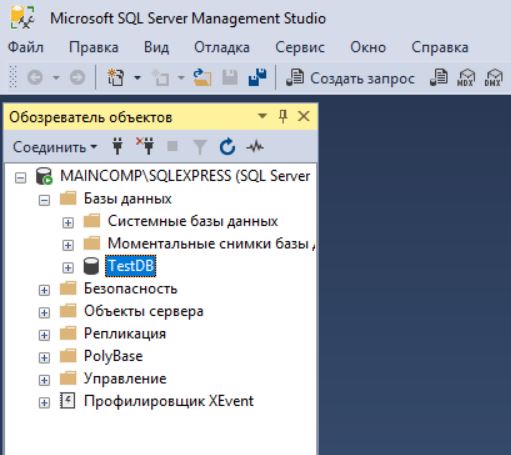
*Рисунок 2.2. Окно входа в БД*

2. В обозревателе объектов щелкнуть ПКМ по контейнеру «База данных» и создать базу данных

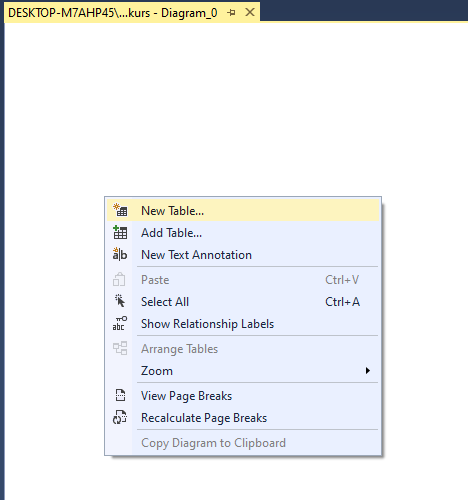
*Рисунок 2.3. Создание БД через обозреватель объектов*

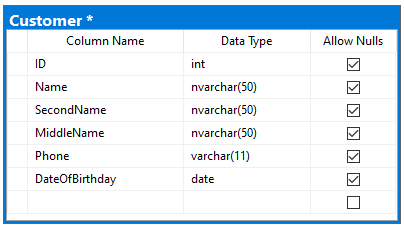
1. Открывается окно «Создание базы данных», где необходимо написать название базы данных т.к. остальные параметры необходимо настраивать уже после создания базы данных.

*Рисунок 2.4. Создание БД*

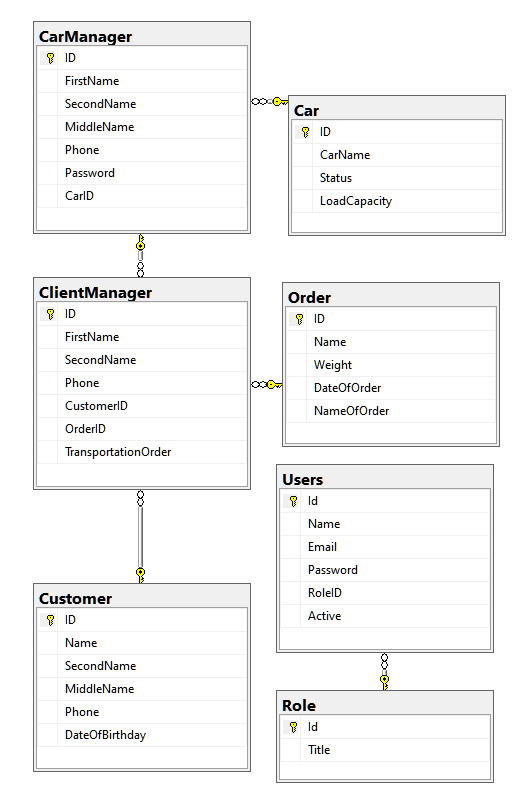
1. База данных создана и видна в обозревателе объектов

*Рисунок 2.5. БД в обозревателе объектов*

*Рисунок 2.6. Создание «Сущности» в БД*

****

*Рисунок 2.7. Созданная «Сущность» с атрибутами и типами данных*



*Рисунок 2.8. Созданная диаграмма Базы Данных с внесенными в нее данными*

# 2.3 Словарь данных

Словарь данных содержит описание сущностей (таблиц), включающее в себя определение атрибутов, а также дополнительные сведения, например, единицы измерения и диапазоны изменения атрибутов, цель определения такого объекта, сведения о его разработчике и времени создания и т.д.

Таблица 1. Словарь данных

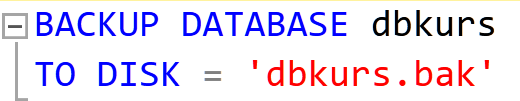
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Data Dictionary | | | | |
| **Car** | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
|  | **ID** | int | Y | ID номер автомобиля |
|  | **CarName** | nvarchar (MAX) | Y | Название автомобиля |
|  | **Status** | nvarchar (50) | N | Статус автомобиля |
|  | **LoadCapacity** | float | N | Грузоподъемность авто |
| СarManager | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
|  | **ID** | int | Y | ID номер менеджера по работе с автомобилями |
|  | **FirstName** | nvarchar (50) | Y | Имя менеджера |
|  | **SecondName** | nvarchar (50) | Y | Фамилия менеджера |
|  | **MiddleName** | nvarchar (50) | N | Отчество менеджера |
|  | **Phone** | nvarchar (11) | Y | Номер телефона менеджера |
|  | **Password** | nvarchar (MAX) | Y | Пароль менеджера |
|  | **CarID** | int | Y | ID номер автомобиля |
| ClientManager | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
|  | **ID** | int | Y | ID номер менеджера по работе с клиентами |
|  | **FirstName** | nvarchar (50) | Y | Имя менеджера |
|  | **SecondName** | nvarchar (50) | Y | Фамилия менеджера |
|  | **Phone** | nvarchar (11) | Y | Номер телефона менеджера |
|  | **CustomerID** | int | N | Номер заказчика |
|  | **OrderID** | int | N | Номер заказа |
|  | **TransportationOrder** | int | N | Номер заказа на перевоз |
| Customer | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
|  | **ID** | int | Y | ID номер покупателя |
|  | **Name** | nvarchar (50) | Y | Имя покупателя |
|  | **SecondName** | nvarchar (50) | Y | Фамилия покупателя |
|  | **MiddleName** | nvarchar (50) | Y | Отчество покупателя |
|  | **Phone** | nvarchar (11) | Y | Номер телефона покупателя |
|  | **DateOfBirthday** | Date | N | Дата рождения покупателя |
| Users | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
|  | **ID** | Int | Y | ID номер пользователя |
|  | **Name** | varchar (50) | Y | Имя пользователя |
|  | **Email** | nvarchar (50) | Y | Email адрес пользователя |
|  | **Password** | varchar (50) | Y | Пароль пользователя |
|  | **RoleID** | int | N | ID номер роли пользователя |
|  | **Active** | bit | N | Работает ли пользователь в Базе Данных |
| Role | | | | |
| **KEY** | **FIELD NAME** | **DATA TYPE / FIELD SIZE** | **REQUIRED?** | **NOTES** |
|  | **ID** | Int | Y | ID номер роли |
|  | **Title** | varchar (50) | Y | Название роли |

# ГЛАВА 3. РАЗРАБОТКА БАЗЫ ДАННЫХ

## **3.1 Введение в безопасность**

Создает резервную копию полной базы данных SQL Server для создания резервной копии базы данных или одного, или нескольких файловых групп базы данных для создания резервной копии файлов (BACKUP DATABASE). Кроме того, при использовании модели полного восстановления или модели восстановления с неполным протоколированием создается резервная копия журнала транзакций (BACKUP LOG).

Указывает, что должна быть создана резервная копия всей базы данных. Если указан список файлов и файловых групп, то только они включаются в резервную копию. Во время полной или разностной резервной копии базы данных SQL Server создает достаточно резервных копий журнала транзакций, чтобы создать согласованную базу данных при восстановлении резервной копии.

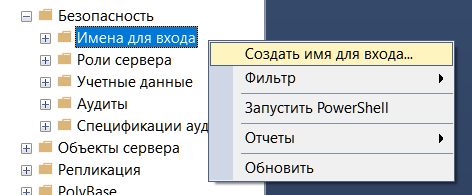
Чтобы сделать BACKUP базы данных необходимо:

*Рисунок 3.1 Создание бэкапа БД с помощью запроса*

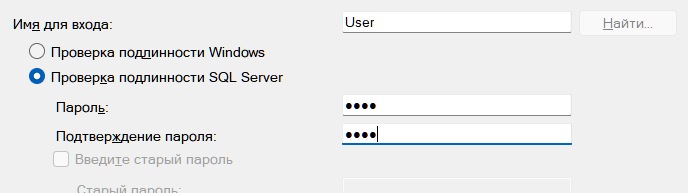
# 3.2 Управление безопасностью уровня сервера

Пользователь является субъектом безопасности уровня сервера. Для соединения с базой данных имя входа должно быть сопоставлено с пользователем базы данных. Имя входа может быть сопоставлено с различными базами данных в качестве разных пользователей, но в каждой базе данных ему может быть сопоставлен только один пользователь.

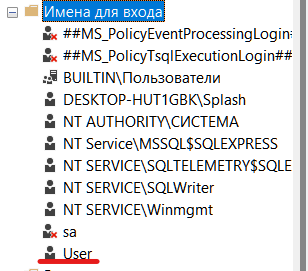
Как субъектам безопасности, пользователям могут предоставляться разрешения. Областью действия пользователя является база данных. Чтобы подключиться к определенной базе данных в экземпляре SQL Server, необходимо сопоставить имя входа с пользователем базы данных. Разрешения внутри базы данных предоставляются и запрещаются для пользователя базы данных, а не имени входа.

Для создания пользователя необходимо:

*Рисунок 3.2 Создание логина пользователя в БД*

1. 

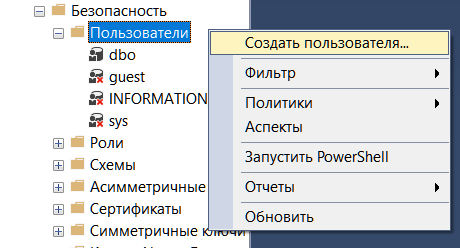
*Рисунок 3.3 Создание логина и пароля пользователя БД*

1. 

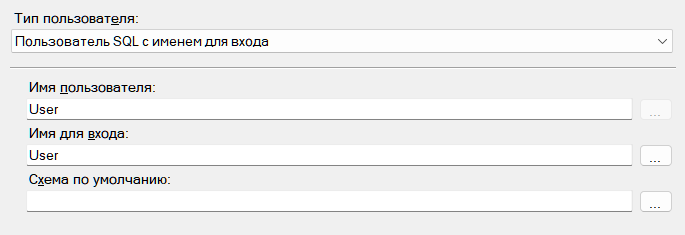
*Рисунок 3.4 Созданный пользователь в БД*

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

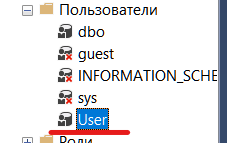
Управление участниками уровня базы данных является важной частью обеспечения безопасности и эффективного функционирования базы данных. Управление участниками уровня базы данных имеет решающее значение для обеспечения безопасности данных и предотвращения угроз безопасности. Важно тщательно контролировать доступ к базе данных и постоянно обновлять политики и процедуры управления участниками для предотвращения потенциальных уязвимостей.

1.

*Рисунок 3.5 Создание пользователя в БД*

2.

*Рисунок 3.6 Данные пользователя “User”*

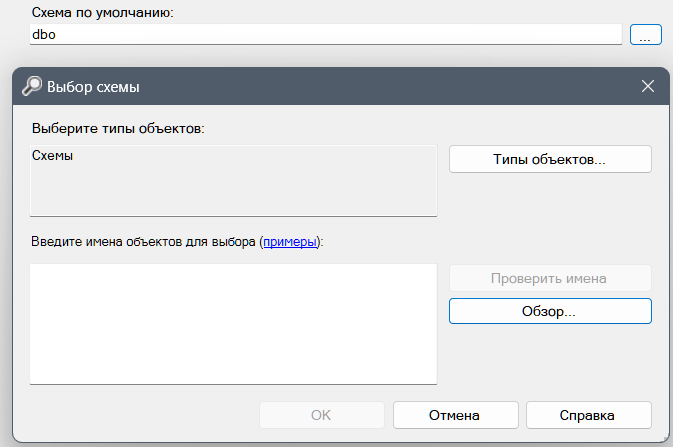
3.

*Рисунок 3.7 Пользователь созданный на уровне БД*

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

Каждая защищаемая среда SQL Server имеет связанные разрешения, которые можно предоставить субъекту. Разрешения в ядро СУБД управляются на уровне сервера, назначенном ролям входа и серверам, а также на уровне базы данных, назначенной пользователям базы данных и ролям базы данных. Модель для База данных SQL Azure имеет ту же систему для разрешений базы данных, но разрешения на уровне сервера недоступны.

Как установить разрешения пользователю БД:

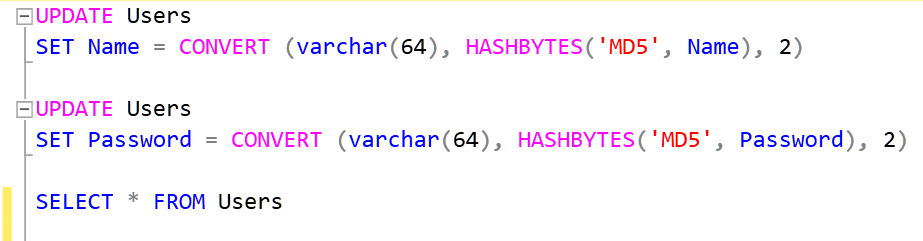
1. 

*Рисунок 3.8 Окно выбора роли пользователя*

## **3.5 Шифрование данных базы данных**

Шифрование обеспечивает безопасность информации на уровне базы данных и операционной системы. Кроме того, оно снижает вероятность несанкционированного раскрытия конфиденциальных сведений, даже если поражены инфраструктура или база данных SQL Server.

Вот как сделать шифрование логина и пароля в БД:

1. 

*Рисунок 3.9 Запрос для шифрования пароля и логина в БД*

# ЗАКЛЮЧЕНИЕ

Необходимо сказать, что в настоящее время не все современные предприятия работают на базе различных автоматизированных систем, внедрение АИС в производственном цикле предприятий не распространено повсеместно. Большинство предприятий не применяют автоматизированные системы управления, поскольку: автоматизация всего комплекса деятельности предприятия недостаточно развита; владельцы предприятий не понимают очевидные преимущества автоматизированного управления; недостаточное количество специалистов в области обеспечения работоспособности систем управления предприятий; стоимость оборудования; большинство сотрудников боятся сокращения в виду того, что машина может заменить человека. Между тем, внедрение АИС на предприятиях позволяет увеличить показатели таких производственных потоков, как:

* составление и контроль исполнения проектов;
* управление загрузкой мощностей предприятия;
* финансовый анализ и бухгалтерский учет;
* управление ресурсами предприятия;
* оптимизация движения различных производственных потоков;
* создание новых изделий, включая техническую документацию;
* оформление заказов и контроль современного исполнения;
* анализ внутренних и внешних изменений, предупреждение о чрезвычайных и внештатных ситуациях и прочее.

Разработана база данных автоматизированной информационной системы «Автотранспортного предприятия». Благодаря этой информационной системе упростилось создание, реализация и доставка заказов клиентам. Информационная система “Автотранспортное предприятие” снижает время, затраченное на создание, оформление, заказ, доставку и получение заказа. Автоматизированная информационная система “Автотранспортное предприятие” позволяет использовать меньше ручного труда, заменяя его автоматизированными функциями.

# СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Троелсен Э. Язык программирования C# 7 и платформы .NET и .NET Core 8-е издание: книга/ Э. Троелсен, Ф. Джепикс. СПб. 2018 - 1328 с.
2. Троелсен и Джепикс. Язык программирования C# 7 и платформы .NET и .NET Core
3. Стивен С. Скиена. Алгоритмы. Руководство по разработке
4. Ицик Бен-Ган. Microsoft SQL Server 2018. Основы T-SQL

Интернет-ресурсы

1. Обзор Entity Framework – Режим доступа: https://learn.microsoft.com/ru-ru/dotnet/framework/data/adonet/ef/overview.
2. Обзор платформы .NET – Режим доступа: https://learn.microsoft.com/ru-ru/dotnet/framework/get-started/overview.
3. Обзор функционала SQL Server Management Studio (SSMS) – Режим доступа: https://info-comp.ru/ssms-feature-overview?ysclid=lenvbogeq8366784796.
4. Общие сведения о WPF – Режим доступа: https://learn.microsoft.com/ru-ru/dotnet/desktop/wpf/introduction-to-wpf?view=netframeworkdesktop-4.8.
5. Руководство по WPF – Режим доступа: https://metanit.com/sharp/wpf/?ysclid=lenvi5n1bh823160038.
6. Руководство по классическим приложениям (WPF .NET) – Режим доступа:
7. Что такое MS SQL Server и чем она отличается от других СУБД – Режим доступа: https://timeweb.com/ru/community/articles/osnovy-raboty-s-mssqlserver.
8. Язык С#: где используют, что пишут, как появился и чем хорош – Режим доступа: https://skillbox.ru/media/code/yazyk-s-gde-ispolzuyut-chto-pishut-kak-poyavilsya-i-chem-khorosh/?ysclid=lenvmtelk1292224169.