|  |  |  |
| --- | --- | --- |
| Департамент образования Ярославской области  Государственное профессиональное образовательное автономное учреждение  Ярославской области  «ЯРОСЛАВСКИЙ ПРОМЫШЛЕННО-ЭКОНОМИЧЕСКИЙ КОЛЛЕДЖ  им. Н.П. ПАСТУХОВА» | | |
| **Отчет По учебной практике**  **по профессиональному модулю ПМ.05**  **Проектирование и разработка информационных систем** | | |
| ОУП ПМ.05.09.02.07.18ИП1.13 | | |
|  | | |
|  | | |
|  | | |
|  | | |
|  | Студент  \_\_\_\_\_\_\_\_\_\_\_\_\_ А.Э. Кязимова  «28» сентябрь 2021 г. |
|  | Преподаватель  \_\_\_\_\_\_\_\_\_\_\_\_\_ Ю.В. Маянцева  «28» сентябрь 2021 г. |
| 2021 | | |

**Содержание**

[Введение 3](#_Toc83019495)

[Практическая работа №1 5](#_Toc83019496)

[Практическая работа №2 8](#_Toc83019497)

[Практическая работа №3 11](#_Toc83019498)

# **Введение**

Учебная практика направлена на:

* формирование у студентов практических профессиональных умений, приобретение первоначального практического опыта;
* закрепление теоретических знаний, полученными студентами в процессе обучения профессиональных модулей;
* углубление первоначального профессионального опыта студента, развития общих и профессиональных компетенций, проверку его готовности к самостоятельной трудовой деятельности;
* сбор, систематизация и обобщение практического материала (в том числе) для использования в выпускной квалификационной работе.

По окончании практики студент сдаёт отчет в соответствии с содержанием тематического плана практики и по форме, установленной в ГПОАУ ЯО «Ярославский промышленно-экономический колледж имени Н.П. Пастухова».

Итоговая аттестация проводится в форме - дифференцированного зачёта.

С целью овладения указанным видом профессиональной деятельности и соответствующими профессиональными компетенциями студент в ходе прохождения учебной практики должен:

иметь практический опыт:

* в управлении процессом разработки приложений с использованием инструментальных средств;
* обеспечении сбора данных для анализа использования и функционирования информационной системы;
* программировании в соответствии с требованиями технического задания;
* использовании критериев оценки качества и надежности функционирования информационной системы;
* применении методики тестирования разрабатываемых приложений;
* определении состава оборудования и программных средств разработки информационной системы;
* разработке документации по эксплуатации информационной системы;
* проведении оценки качества и экономической эффективности информационной системы в рамках своей компетенции;
* модификации отдельных модулей информационной системы.

уметь:

* осуществлять постановку задач по обработке информации;
* проводить анализ предметной области;
* осуществлять выбор модели и средства построения информационной системы и программных средств;
* использовать алгоритмы обработки информации для различных приложений;
* решать прикладные вопросы программирования и языка сценариев для создания программ;
* разрабатывать графический интерфейс приложения; создавать и управлять проектом по разработке приложения;
* проектировать и разрабатывать систему по заданным требованиям и спецификациям.

знать:

* основные виды и процедуры обработки информации, модели и методы решения задач обработки информации;
* основные платформы для создания, исполнения и управления информационной системой;
* основные процессы управления проектом разработки;
* основные модели построения информационных систем, их структуру, особенности и области применения;
* методы и средства проектирования, разработки и тестирования информационных систем;
* систему стандартизации, сертификации и систему обеспечения качества продукции

**Практическая работа №1**

**Тема**

Построение диаграммы вариантов использования.

**Цель работы**

Научиться строить диаграммы вариантов использования (use case diagram) в среде MS Visio.

**Теоретическая часть**

**Диаграмма вариантов использования** в UML— диаграмма, отражающая отношения между экторами и прецедентами и являющаяся составной частью *модели прецедентов*, позволяющей описать систему на концептуальном уровне

**Эктор -** это множество логически связанных ролей, исполняемых при взаимодействии с прецедентами или сущностями (система, подсистема или класс). Эктором может быть человек или другая система, подсистема или класс, которые представляют нечто вне сущности.

**Прецедент -** описание множества последовательных событий (включая варианты), выполняемых системой, которые приводят к наблюдаемому эктором результату. Прецедент представляет поведение сущности, описывая взаимодействие между экторами и системой. Прецедент не показывает, "как" достигается некоторый результат, а только "что" именно выполняется.

**Практическая часть**

С помощью MS Visio была построена диаграмма вариантов использования для предметной области «Грузоперевозки» (Рис. 1).

Единственным актером является диспетчер, к которому относятся несколько прецедентов с отношениями расширения и включением.

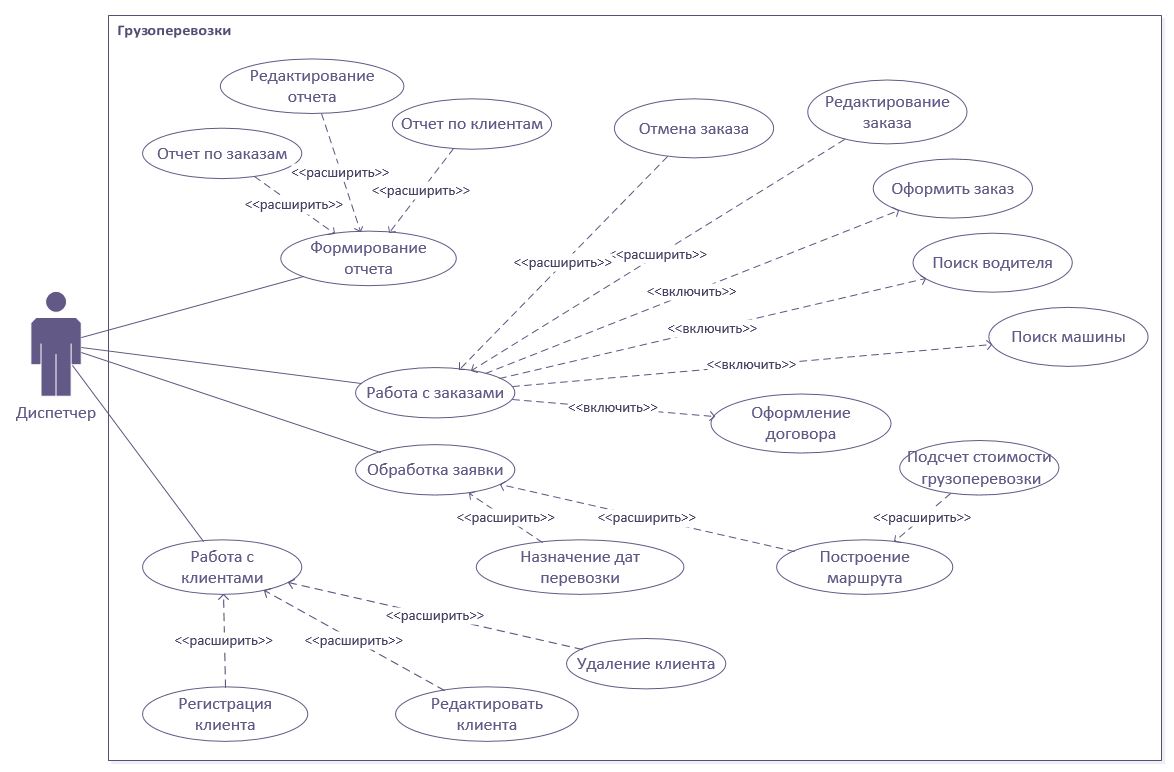


Рисунок 1 Диаграмма вариантов использования

**Спецификация**

1. **Название прецедента:** создание заказа

Действующее лицо: диспетчер

Цель: создать заказ по запросу клиента

Предусловия: диспетчер осуществил вход в систему

Главная последовательность: диспетчер, после входа в систему, выбирает в главном меню кнопку «создать заказ»; система открывает окно/форму для создания заказа, содержащее поля: номер заказа, ФИО клиента, номер телефона, объем груза и маршрут перевозки, а также в окне есть кнопки «далее» и «назад»; после заполнения полей, диспетчер нажимает кнопку «далее» и система добавляет заказ; диспетчеру открывается главное меню и в списке заказов появляется новый.

**Альтернативная последовательность** (возврат в главное меню без создания заказа):

Диспетчер выбирает в главном меню пункт «создать заказ»; система показывает диспетчеру окно создания заказа, содержащее поля для ввода номер заказа, ФИО клиента, номер телефона, объем груза и маршрут перевозки, а также в окне есть кнопки «далее» и «назад; диспетчер нажимает кнопку «назад»; диспетчеру открывается главное меню (при этом данные, введенные в формы окна создания заказа не сохраняются).

1. **Название прецедента:** регистрация клиента

Действующее лицо: диспетчер

Цель: зарегистрировать и добавить нового клиента в базу

Предусловия: диспетчер осуществил вход в систему

Главная последовательность: диспетчер, после входа в систему, выбирает в главном меню кнопку «добавить клиента»; система открывает окно добавления клиента с полями: ФИО клиента, номер телефона и номер заказа, а также в окне есть кнопки «далее» и «назад»; после заполнения полей, диспетчер нажимает кнопку «далее» и система добавляет клиента; диспетчеру в течении 5 секунд открывается главное меню и в списке клиентов появляется новый.

**Альтернативная последовательность** (возврат в главное меню без регистрации клиента):

Диспетчер выбирает в главном меню пункт «добавить клиента»; система показывает диспетчеру окно добавления клиента, содержащее поля для ввода ФИО клиента, номер телефона, объем груза и маршрут перевозки, а также кнопки «далее» и «назад»; диспетчер нажимает кнопку «назад»; диспетчеру открывается главное меню (при этом данные, введенные в формы окна добавления клиента не сохраняются).

**Контрольные вопросы**

1. Какие цели преследует разработка диаграммы использования?

* Определить общие границы и контекст моделируемой предметной области на начальных этапах проектирования системы;
* Сформулировать общие требования к функциональному поведению проектируемой системы;
* Разработать исходную концептуальную модель системы для ее последующей детализации в форме логических и физических моделей;
* Подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями.

1. Для чего нужна диаграмма вариантов использования?

Диаграммы вариантов использования предназначены для упрощения взаимодействия с будущими пользователями системы, с клиентами, и особенно пригодятся для определения необходимых характеристик системы.

1. Из чего состоит диаграмма вариантов использования?

Диаграмма вариантов использования состоит из: актера, прецедента, отношения и класса.

1. Виды взаимодействия, используемые в диаграмме вариантов использования?

Виды взаимодействия:

* Ассоциация;
* Обобщение;
* Расширение;
* Включение.

1. Из чего состоит созданная вами диаграмма?

Диаграмма по предметной области состоит из одного актера (диспетчера) который имеет несколько прецедентов и отношений.

**Вывод**

Получили практический опыт в построении диаграмм вариантов использования.

**Практическая работа №2**

**Тема**

Моделирование поведенческих аспектов предметной области на основе анализа описания предметной области, процессов. Построение диаграммы деятельности (Activity).

**Цель работы**

Закрепить навык построения диаграмм деятельности (activity diagram), представляющих узлы выполнения программных компонентов реального времени, а также процессов и объектов.

**Теоретическая часть**

Диаграмма деятельности (Activity diagram) – это диаграмма для демонстрации рабочего процесса некоторой деятельности, основанной на поэтапных действиях и действиях с поддержкой выбора и параллелизма

Диаграммы деятельности позволяют моделировать сложный жизненный цикл объекта, с переходами из одного состояния (деятельности) в другое.

**Практическая часть**

С помощью MS Visio была построена диаграмма деятельности для предметной области «Грузоперевозки» (Рис.2,3).

На диаграмме был расписан алгоритм взаимодействия между сотрудником системой и частично клиентом, возможные варианты исхода действий, их распараллеливание и слияние.

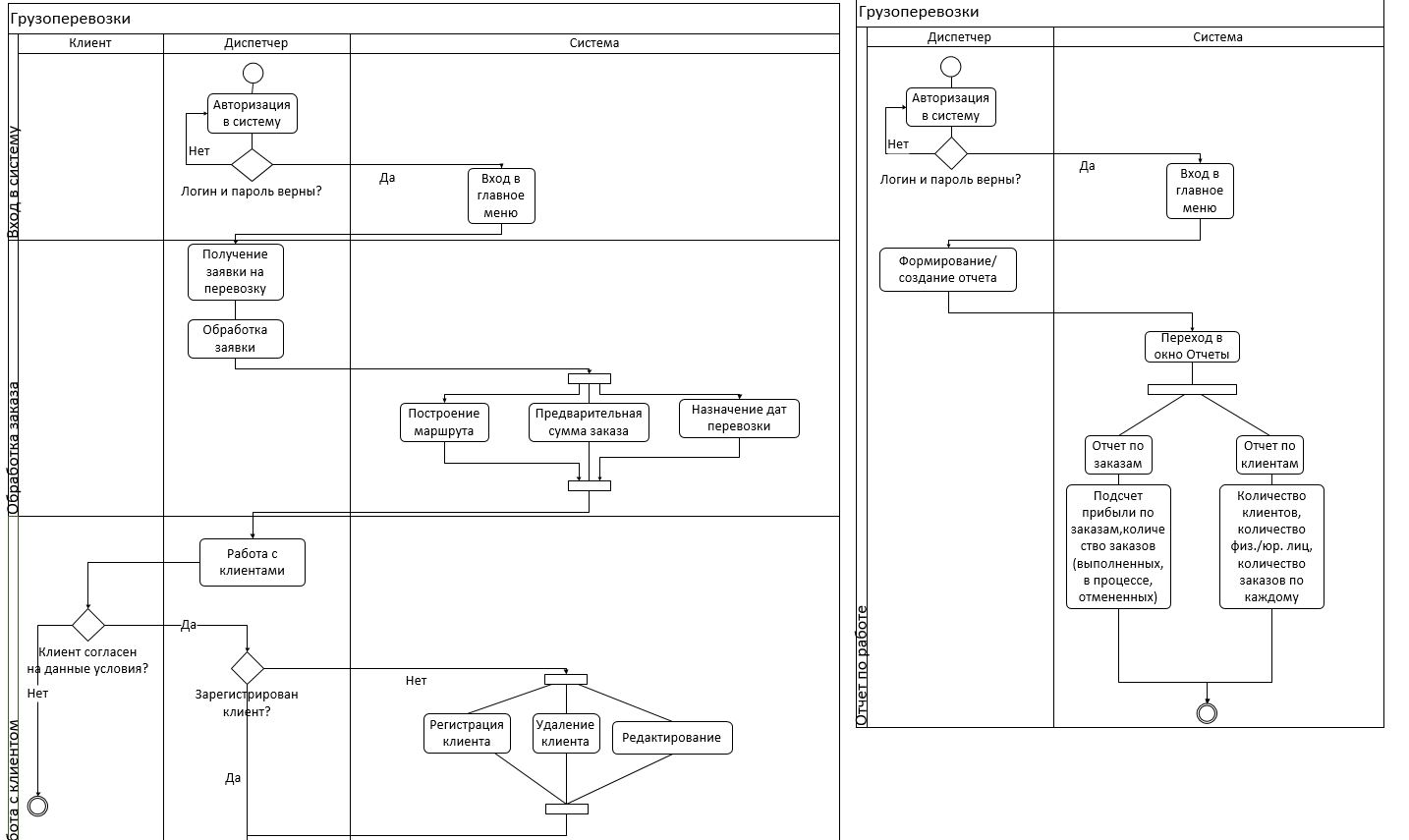


Рисунок 2 Диаграмма деятельности

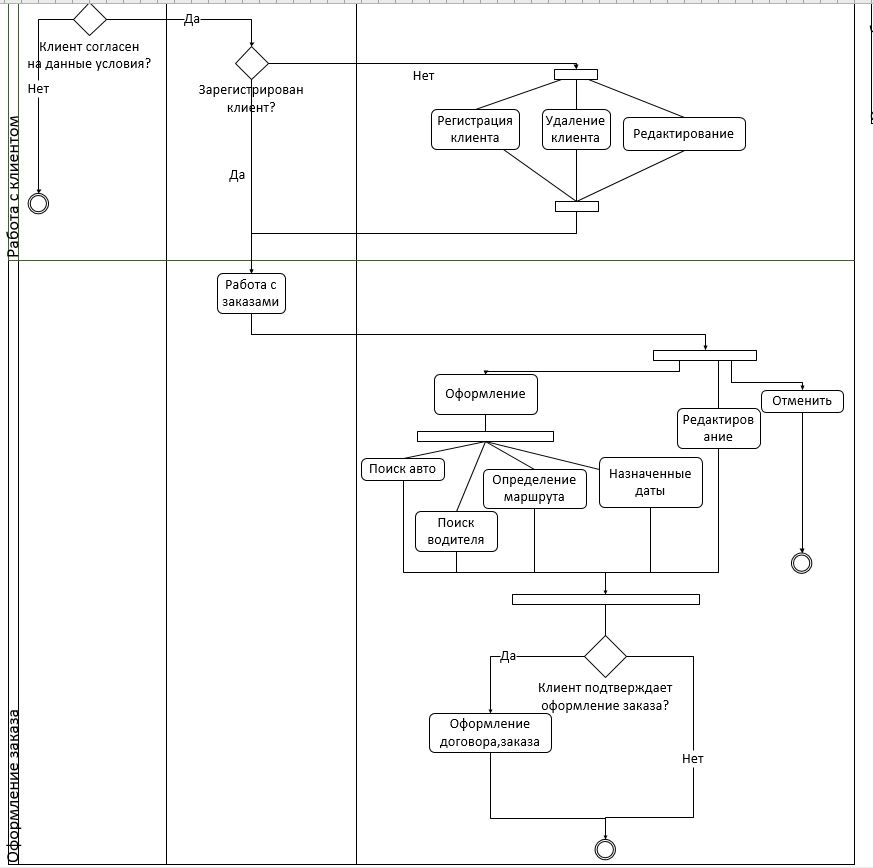


Рисунок 3 Диаграмма деятельности (2 часть)

**Контрольные вопросы**

1. Для чего строится диаграмма деятельности в проекте?

Диаграммы деятельности используются при моделировании бизнес-процессов, технологических процессов, последовательных и параллельных вычислений.

1. Расскажите об основных элементах диаграммы деятельности.

* Прямоугольники с закруглениями — действия (операция). Узел управления (control node) — это абстрактный узел действия, которое координирует потоки действий.
* Ромбы — решения. Узел решения предназначен для определения правила ветвления и различных вариантов дальнейшего развития сценария. В точку ветвления входит ровно один переход, а выходит — два или более.
* Широкие полосы — начало (разветвление) и окончание (схождение) ветвления действий. Узел объединения имеет два и более входящих узла и один исходящий.
* Чёрный круг — начало процесса (начальный узел). Начальный узел деятельности (или начальное состояние деятельности) является узлом управления, в котором начинается поток (или потоки) при вызове данной деятельности извне.
* Чёрный круг с обводкой — окончание процесса (финальный узел). Конечный узел деятельности (или конечное состояние деятельности) является узлом управления, который останавливает все потоки данной диаграммы деятельности. На диаграмме может быть более одного конечного узла.

**Вывод**

Получили практический опыт в построении диаграмм деятельности.

**Практическая работа №3**

**Тема**

Построение диаграммы состояний.

**Цель работы**

Научиться строить диаграммы состояний.

**Теоретическая часть**

Диаграмма состояний (Statechart Diagram) – это методология объектно-ориентированного проектирования, предназначенная для представления жизненного цикла объектов в реальном или абстрактном мире.

Диаграммы состояний применяются для того, чтобы объяснить, каким образом работают сложные объекты. Несмотря на то что смысл понятия «состояние» интуитивно понятен, все же приведем его определение в таком виде, в каком его дают классики и Zicom Mentor:

Состояние (state) - ситуация в жизненном цикле объекта, во время которой он удовлетворяет некоторому условию, выполняет определенную деятельность или ожидает какого-то события. Состояние объекта определяется значениями некоторых его атрибутов и присутствием или отсутствием связей с другими объектами.

**Практическая часть**

С помощью MS Visio была построена диаграмма состояний для предметной области «Грузоперевозки» (Рис. 4,5,6).

В каждом большом составном состоянии включены другие состояния (в большом окне включены окна меньше с ходом выполнения действий), которые выполняются при том или ином действии пользователя, т.е. описание того состояния, в котором находиться программа при том или ином действии. Каждое составное состояние имеет переход к другому состоянию, так же зависящее от того или иного действия

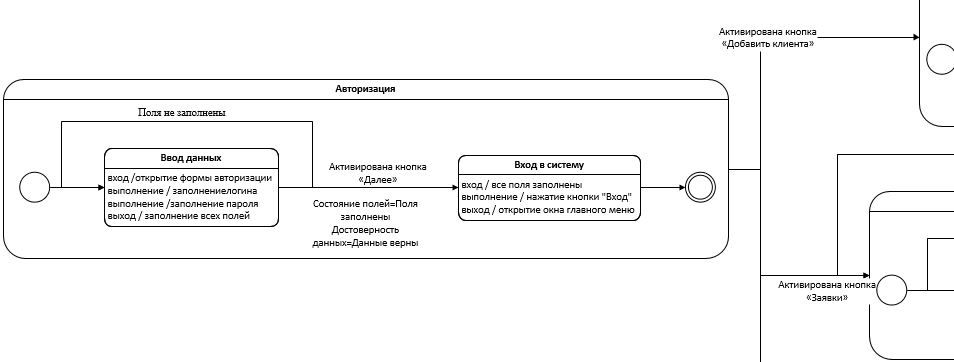


Рисунок 4 Диаграмма состояний

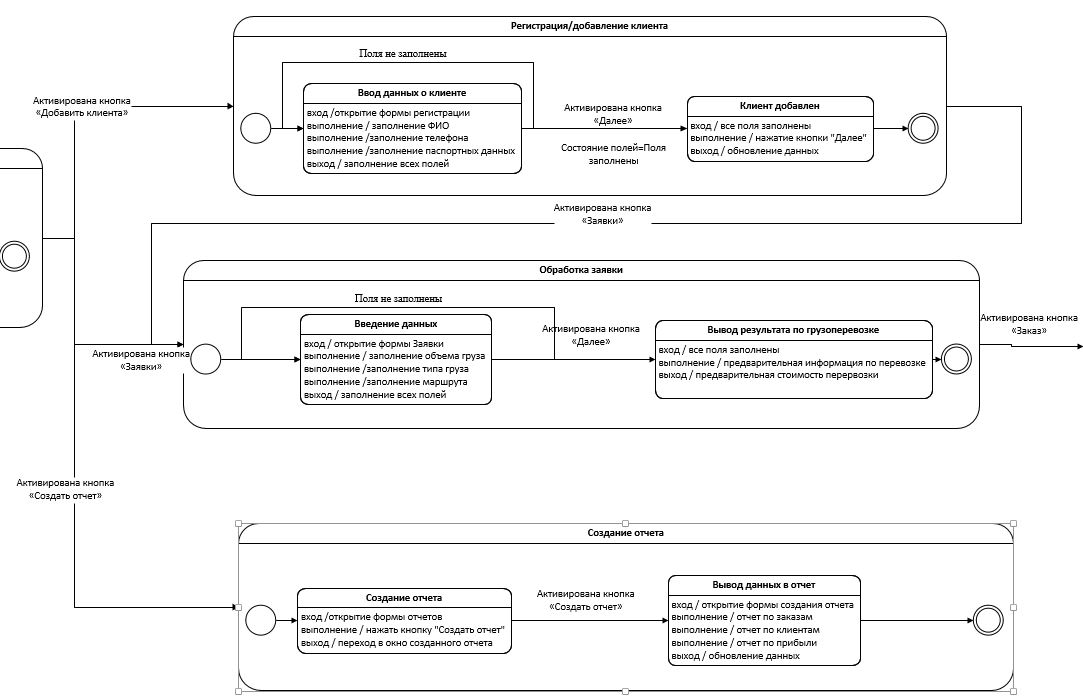


Рисунок 5 Диаграмма состояний (2 часть)



Рисунок 6 Диаграмма состояний (3 часть)

**Вывод**

Получили практический опыт в построении диаграмм состояний.

**Практическая работа №4**

**Тема**

Построение диаграммы последовательности

**Цель работы**

Закрепить навык построения диаграммы последовательности (sequence diagrams), отображающую взаимодействие объектов в динамике.

**Теоретическая часть**

Диаграмма последовательности *(*Sequence Diagram*)* — [UML-диаграмма](https://ru.wikipedia.org/wiki/%D0%94%D0%B8%D0%B0%D0%B3%D1%80%D0%B0%D0%BC%D0%BC%D0%B0_(UML)), на которой для некоторого набора объектов на единой временной оси показан жизненный цикл объекта (создание-деятельность-уничтожение некой сущности) и взаимодействие актеров (действующих лиц) информационной системы в рамках [прецедента](https://ru.wikipedia.org/wiki/%D0%9F%D1%80%D0%B5%D1%86%D0%B5%D0%B4%D0%B5%D0%BD%D1%82_(UML)).

Диаграмма последовательностей относится к диаграммам взаимодействия UML, описывающим поведенческие аспекты системы, но рассматривает взаимодействие объектов во времени. Другими словами, диаграмма последовательностей отображает временные особенности передачи и приема сообщений объектами.

На диаграмме последовательности объекты в основном представляю экземпляры класса или сущности, обладающие поведением. В качестве объектов могут выступать пользователи, инициирующие взаимодействие, классы, обладающие поведением в Системе или программные компоненты, а иногда и Системы в целом.

**Практическая часть**

С помощью MS Visio была построена диаграмма последовательности для предметной области «Грузоперевозки» (Рис. 7).

Диспетчер работает с системой

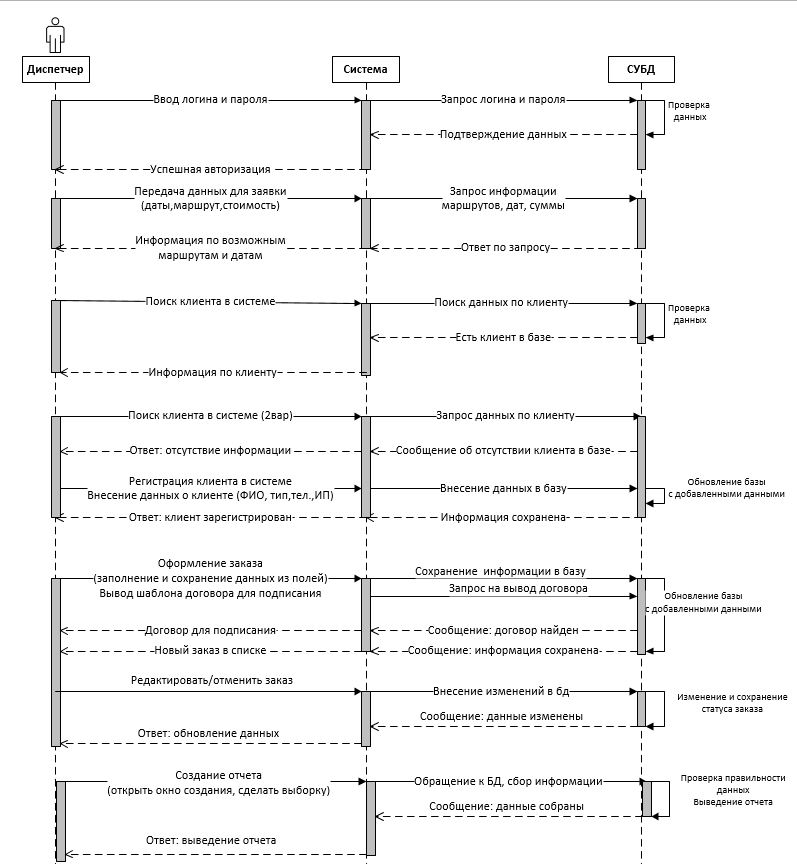


Рисунок 7 Диаграмма последовательности

**Контрольные вопросы**

1. Для каких целей строится диаграмма последовательности в проекте?

Главное предназначение этой диаграммы — описать возможные последовательности состояний и переходов, которые в совокупности характеризуют поведение элемента модели в течение его жизненного цикла. Диаграмма состояний представляет динамическое поведение сущностей, на основе спецификации их реакции на восприятие некоторых конкретных событий.

1. Расскажите об основных элементах диаграммы последовательности.

**Объекты** обозначаются прямоугольниками с подчеркнутыми именами (чтобы отличить их от классов).

**Сообщения (вызовы методов)** - линиями со стрелками.

**Возвращаемые результаты** - пунктирными линиями со стрелками.

**Вывод**

Получили практический опыт в построении диаграмм последовательности.

**Практическая работа №5**

**Тема**

Проектирование диаграммы сущность-связь. Создание словаря данных.

**Цель работы**

Закрепить навык построения диаграммы сущность-связь (ERD).

**Теоретическая часть**

ER диаграммаотображает отношения набора сущностей, хранящиеся в базе данных. Другими словами, ER-диаграммы помогают объяснить логическую структуру баз данных. На первый взгляд диаграмма ER выглядит очень похоже на блок-схему. Однако ER-диаграмма включает в себя множество специализированных символов, и ее значения делают эту модель уникальной. Цель ER Diagram — представить инфраструктуру структуры объекта.

**Практическая часть**

С помощью MS Visio была построена диаграмма сущность-связь для предметной области «Грузоперевозки» (Рис. 8).

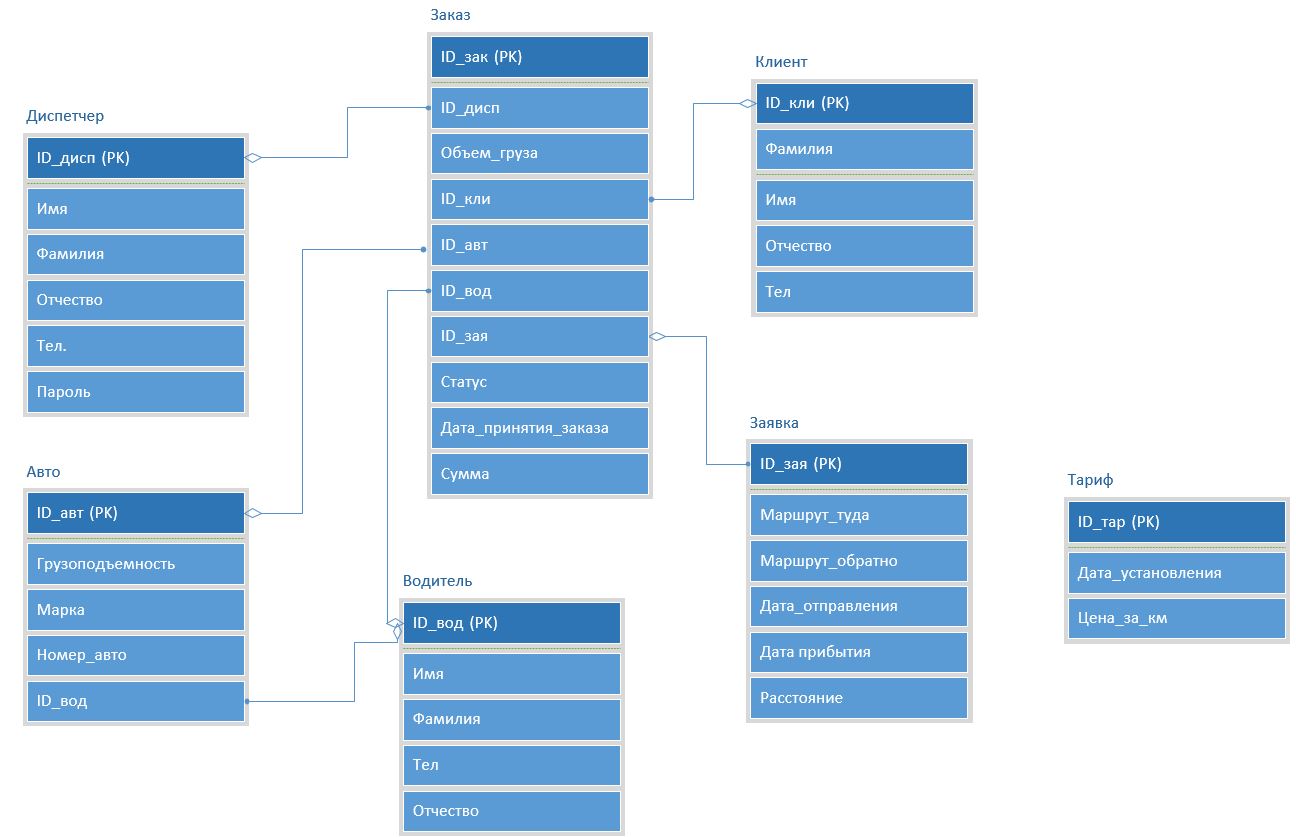


Рисунок 8 ER диаграмма

**Контрольные вопросы**

1. Для каких целей строится диаграмма сущность-связь в проекте?

ER-диаграммы применяются для моделирования и проектирования реляционных баз данных, причем как в плане логических и бизнес-правил (логические модели данных), так и в плане внедрения конкретных технологий (физические модели данных).

1. Расскажите об основных элементах диаграммы ERD.

Сущность – это может быть человек, место, событие или объект, имеющий отношение к данной системе. Сущности представлены на диаграммах ВП прямоугольником и именуются с помощью существительных единственного числа.

Атрибут – это свойство, черта или характеристика сущности, связи или другого атрибута. Сущность может иметь столько атрибутов, сколько необходимо.

Отношения – это описание взаимодействия сущности.

**Вывод**

Получили практический опыт в построении ER диаграммы (сущность-связь).

**Практическая работа №6**

**Тема**

Работа с неструктурированными данными: обработка и импорт в базу данных

**Цель работы**

Закрепить навык работы с данными и СУБД.

**Теоретическая часть**

Система управления базами данных (СУБД) – это комплекс программно-языковых средств, позволяющих создать базы данных и управлять данными. Иными словами, СУБД — это набор программ, позволяющий организовывать, контролировать и администрировать базы данных.

База данных — это упорядоченный набор структурированной информации или данных, которые обычно хранятся в электронном виде в компьютерной системе.

Основные функции СУБД:

* управление данными во внешней памяти (на дисках);
* управление данными в оперативной памяти с использованием дискового кэша;
* журнализация изменений (сохранение истории), резервное копирование и восстановление базы данных после сбоев;
* поддержка языков БД (язык определения данных, язык манипулирования данными).

Неструктурированные данные — данные, которые не соответствуют заранее определённой модели данных, и, как правило, представлены в форме текста с датами, цифрами, фактами, расположенными в нём в произвольной форме.

**Практическая часть**

В СУБД (Microsoft SQL Management Server) была создана база данных (ДЭ) в которую после будут внесены данные (Рис. 9).

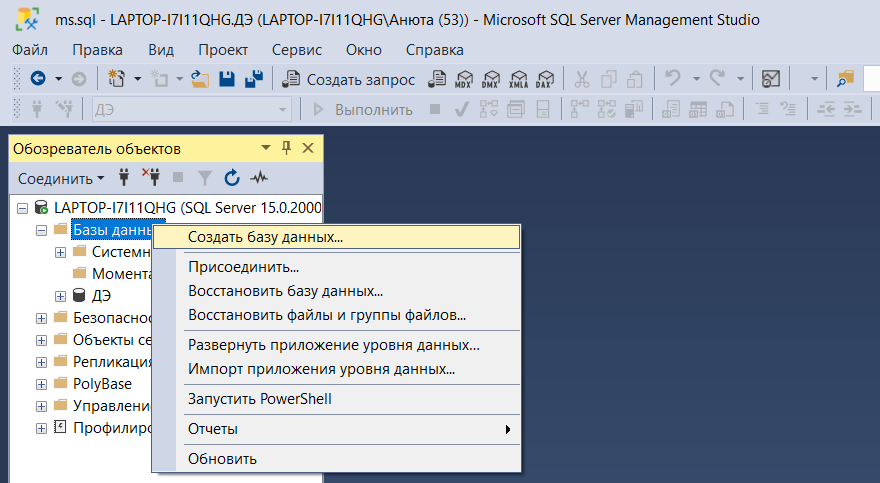


Рисунок 9 Создание базы данных

Перед началом работы нужно восстановить таблицы из скрипта. Для этого необходимо перетащить файл со скриптом в «Обозреватель объектов» СУБД (Рис.10).

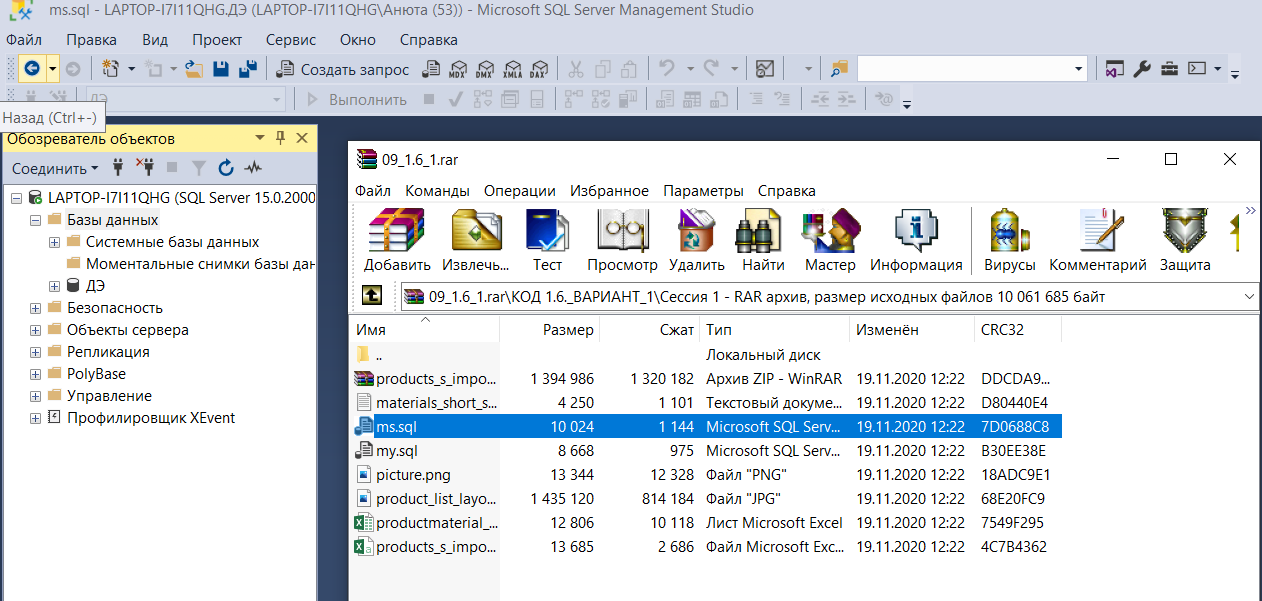


Рисунок 10 Восстановление таблиц

Скрипт открылся в главном окне. Чтобы продолжить работу, нужно сохранить изменения и обновить БД (Рис. 11).

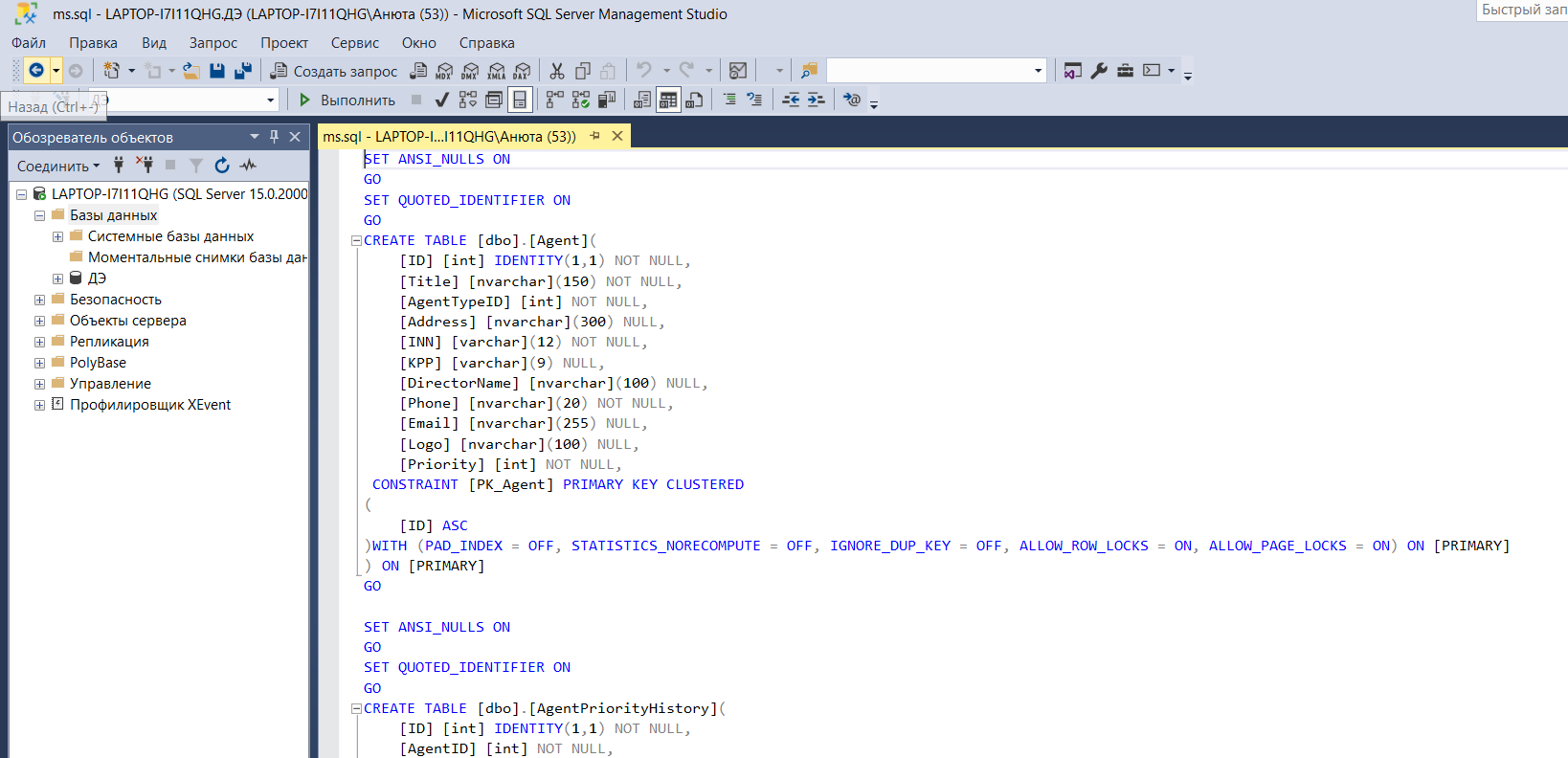


Рисунок 11 Скрипт

После разворачиваем созданную базу, в ней теперь отображаются все нужные таблички (Рис. 12).

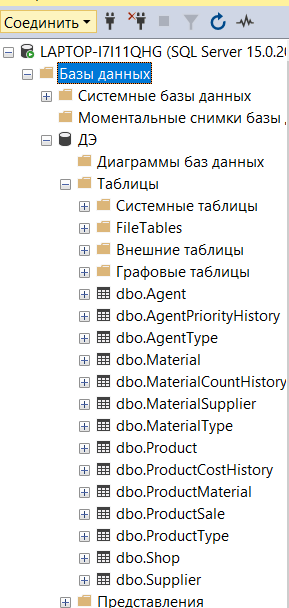


Рисунок 12 Созданные таблицы

Далее нужно неструктурированный текст разделить на столбцы и строки (Рис. 13).

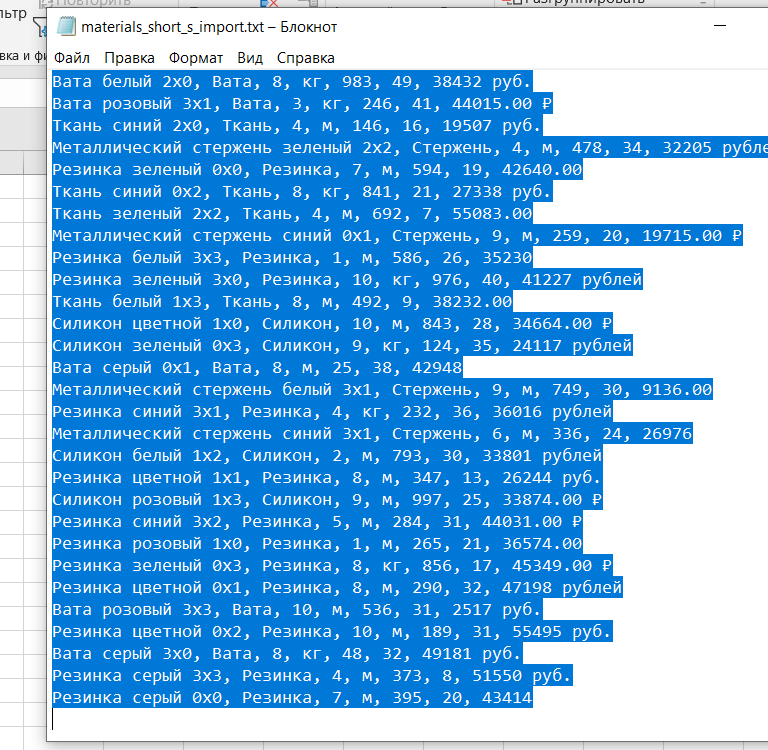


Рисунок 13 Неструктурированные данные

Открываем в Excel пустой лист и в любую ячейку вставляем скопированные данные (Рис. 14).

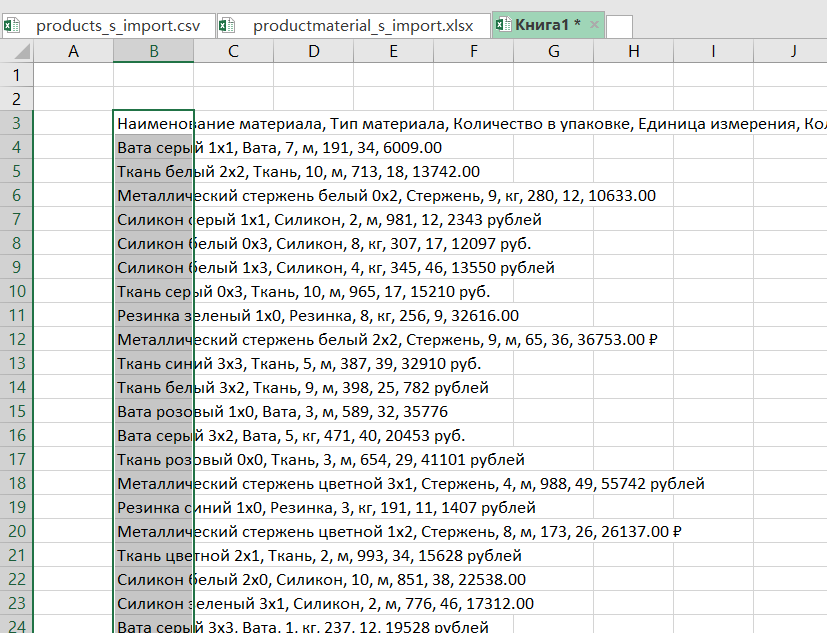


Рисунок 14 Внесенные данные

Во меню сверху выбираем вкладку «Данные» и нажимаем иконку «Текст по столбцам» (Рис. 15).

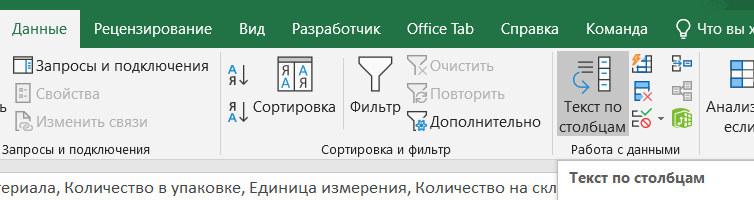


Рисунок 15 Работа с Excel

Далее выбираем нужный формат данных (Рис. 16).

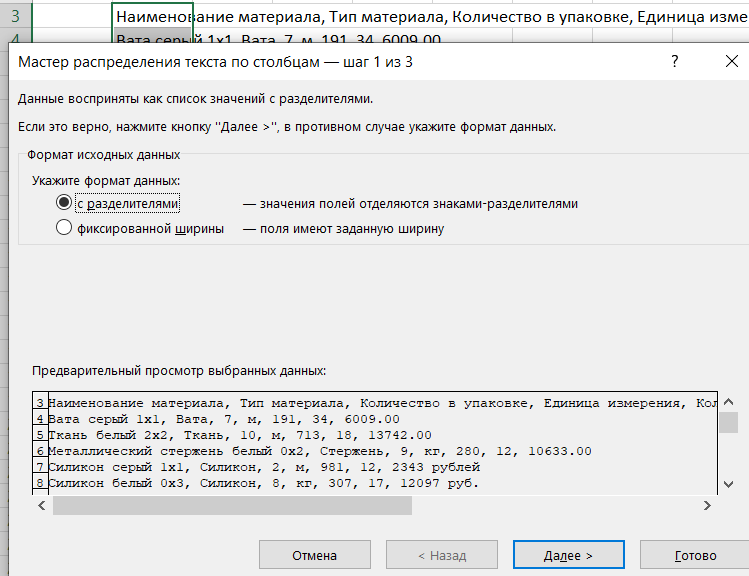


Рисунок 16 Формат данных

И символы разделители по которым будет определяться, где заканчивается и начинается новый столбик (Рис. 17). Третьим шагом нужно нажать «Ок».

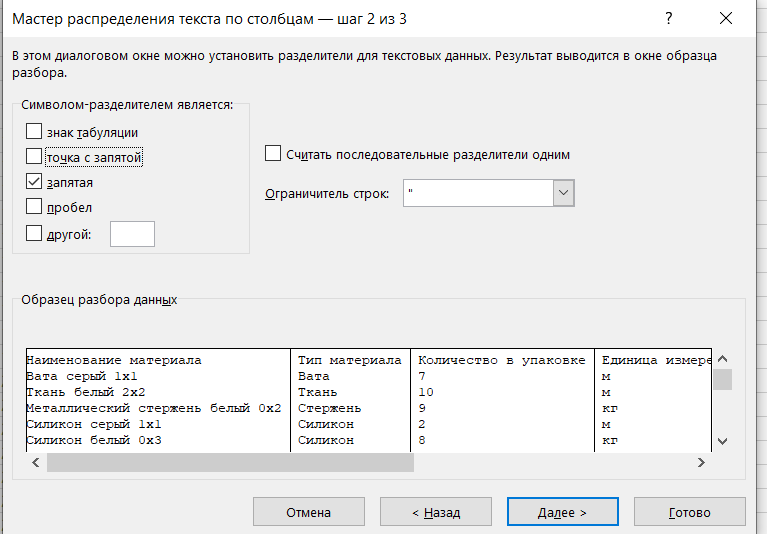


Рисунок 17 Определение разделителей

В конечном результате получаем четко разделенные данные в виде таблицы (Рис.18).

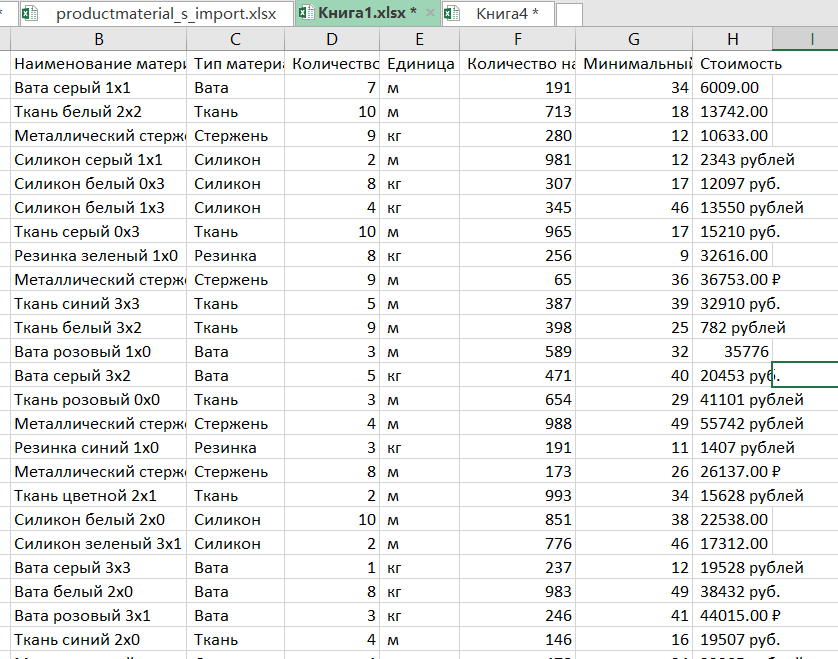


Рисунок 18 Конечная таблица

Заполним одну из таблиц имеющимися данными.

Из обозревателя решений откроем таблицу MaterialType. Открылась табличка пустая с названиями столбцов исходя из которых нужно найти данные (Рис. 19).

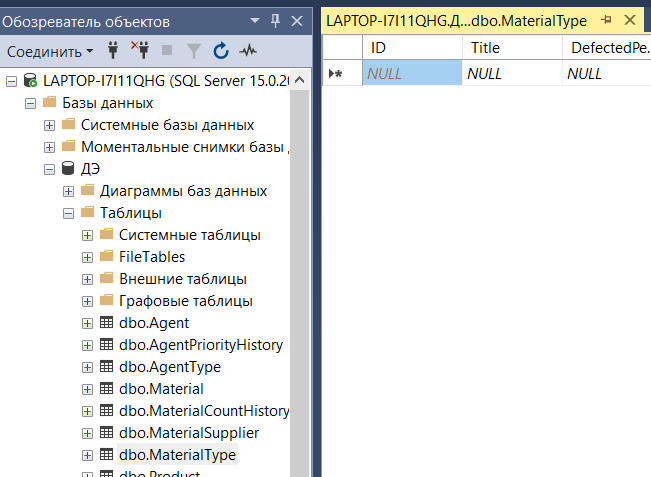


Рисунок 19 Таблица

Возвращаемся к прошлой таблице и копируем на новый лист столбик «Тип материала» (Рис. 20).

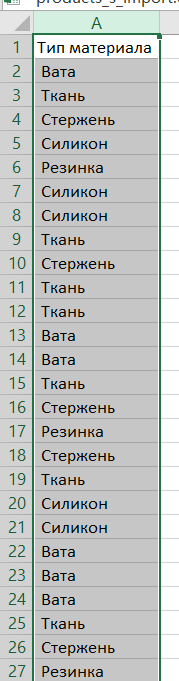


Рисунок 20 Скопированный столбик

Далее выделяем весь столбец и в верхнем меню открываем вкладку «Данные» и нажимаем на иконку «Удалить дубликат» (Рис. 21). После появиться окно с оповещением о количестве дубликатов.

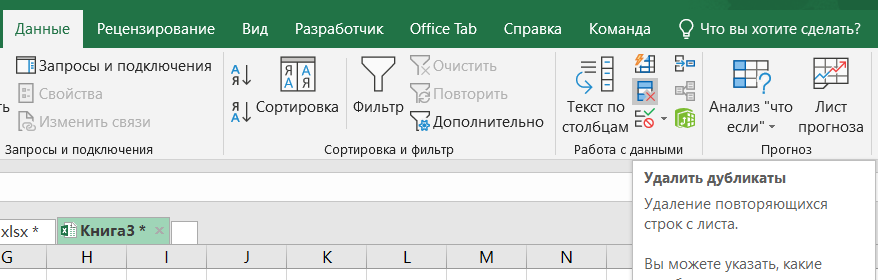


Рисунок 21 Удаление дубликатов

Результат изменений (Рис.22).

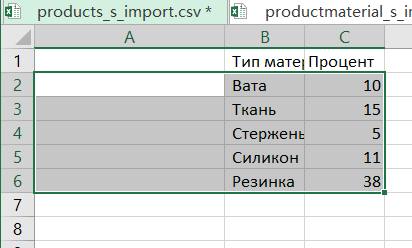


Рисунок 22 Результат

После получившуюся таблицу копируем (заранее добавив пустой столбик под ID поле в БД) и вставляем в таблицу в базе данных (Рис.23).

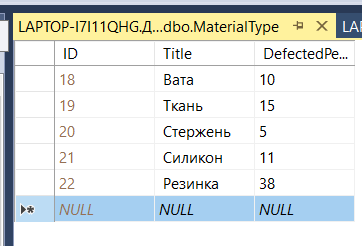


Рисунок 23 Заполненная таблица БД

**Вывод**

В ходе практики закрепили навыки работы с данными и СУБД.