**Министерство образования Московской области**

**Государственное бюджетное профессиональное образовательное учреждение Московской области «Люберецкий техникум имени Героя Советского Союза, летчика-космонавта Ю.А. Гагарина»**

ОТЧЁТ

по учебной практике

//ПМ 02. «Осуществление интеграции программных модулей»

Теплухин Макар Алексеевич

Специальность: 09.02.07 **«Информационные системы и программирование»**

Группа: № 185 , курс 4

Руководители работы Жирнова Юлия Витальевна

г. Люберцы

2021 год

**Содержание**

[Введение 3](#_Toc89100432)

[Изучение предметной области 4](#_Toc89100433)

[Постановка задачи 7](#_Toc89100434)

[Практическая часть 8](#_Toc89100435)

[Выводы 9](#_Toc89100436)

[Литературные источники 10](#_Toc89100437)

# Введение

Профессиональный модуль «Осуществление интеграции программных модулей» являются важными профессиональным модулями для специальности 09.02.07 «Информационные системы и программирование», обуславливающим вид для профессиональной деятельности учащегося.

Учебная практика представляет собой важнейшую составную часть учебного процесса по подготовке специалистов, способствует повышению общего уровня профессиональной подготовки, закреплению и углублению полученных теоретических знаний по дисциплинам.

Целью учебной практики является приобретение навыков программирования на языке C# (WPF). А так же работа с базой данных и её импорт.

Для достижения данной цели требуется решить следующие задачи:

* Проанализировать предметную область.
* Привести к нормальному виду данные для импорта.
* Импортировать базу данных.
* Создать приложение WPF, связанное с БД.

Осваиваемые профессиональные компетенции (ПК):

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

# Изучение предметной области

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.

Развитие клиент-серверных технологий во второй половине 80-х было обусловлено развитием двух ключевых направлений, активно разрабатываемых с конца 70-х годов: персональных компьютеров с одной стороны, и компьютерных сетей — с другой. Долгое время СУБД были доступны лишь для мэйнфреймов, и лишь благодаря росту производительности процессоров для домашних компьютеров и мини-ЭВМ разработчики СУБД (как, например, Oracle) начали создавать соответствующие версии своих продуктов. Одной из первых СУРБД для ПК стала Oracle v3, выпущенная в 1983 году. На тот момент немногочисленные владельцы ПК использовали их в основном для разработки приложений и тестирования[1]. Одним из ключевых этапов в развитии СУБД стал 1986 год. К этому времени появилось ещё несколько компаний-разработчиков СУБД, одной из самых заметных из них стала компания Sybase, основанная двумя годами ранее. К 1986 году Sybase начала комплектовать интеллектуальные рабочие станции (как правило, разработки Sun Microsystems или Apollo Computer) с серверами базы данных (разработанных, например, Oracle). При этом сама клиент-серверная технология сделала возможным отделение модулей обработки информации (т. н. back end) от модулей интерфейса (т. н. front end). Учтя постоянный рост проникновения компьютерных сетей, поставщики решений перешли к задачам распределения остальных задач (например, форматирование отчётов, проверка данных и т. д.) среди рабочих станций сети, оставив серверу выполнять лишь задачи, требующие централизованного решения (хранение и защита данных, оптимизация потока выполнения запросов и т. д.)[1].

C# (произносится си шарп) — объектно-ориентированный язык программирования. Разработан в 1998—2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота[7] как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270.

C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML

Переняв многое от своих предшественников — языков C++, Delphi, Модула, Smalltalk и, в особенности, Java — С#, опираясь на практику их использования, исключает некоторые модели, зарекомендовавшие себя как проблематичные при разработке программных систем, например, C# в отличие от C++ не поддерживает множественное наследование классов (между тем допускается множественная реализация интерфейсов).

C# разрабатывался как язык программирования прикладного уровня для CLR и, как таковой, зависит, прежде всего, от возможностей самой CLR. Это касается, прежде всего, системы типов C#, которая отражает BCL. Присутствие или отсутствие тех или иных выразительных особенностей языка диктуется тем, может ли конкретная языковая особенность быть транслирована в соответствующие конструкции CLR. Так, с развитием CLR от версии 1.1 к 2.0 значительно обогатился и сам C#; подобного взаимодействия следует ожидать и в дальнейшем (однако, эта закономерность была нарушена с выходом C# 3.0, представляющего собой расширения языка, не опирающиеся на расширения платформы .NET). CLR предоставляет C#, как и всем другим .NET-ориентированным языкам, многие возможности, которых лишены «классические» языки программирования. Например, сборка мусора не реализована в самом C#, а производится CLR для программ, написанных на C# точно так же, как это делается для программ на VB.NET, J# и др. Из-за технических ограничений на отображение (стандартные шрифты, браузеры и т. д.), а также из-за того, что знак диеза ♯ не представлен на стандартной клавиатуре компьютера, при записи имени языка программирования используют знак решётки (#). Это соглашение отражено в Спецификации языка C# ECMA-334. Тем не менее, на практике «Майкрософт» использует знак диеза.

# Постановка задачи

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.

С# - Объектно-ориентированный язык программирования. Разработан в 1998-2001 годах группой инженеров компании Microsoft под руководством Андерса Хейлсберга и Скотта Вильтаумота как язык разработки приложений для платформы Microsoft .NET Framework. Впоследствии был стандартизирован как ECMA-334 и ISO/IEC 23270. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (в том числе операторов явного и неявного приведения типа), делегаты, атрибуты, события, переменные, свойства, обобщённые типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ, исключения, комментарии в формате XML.).

В данном отчете необходимо описать 3 выполненных задания по языку разметки HTML. Всего по данной теме было выполнено 30 заданий. Также, необходимо описать 5 заданий выполненных по языку программирования C#

При выполнении данных заданий, студент должен:

* Закрепить знания
* Быть готовым объяснить выполненные задания преподавателю
* Научиться чему-то новому
* Проанализировать предметную область

# [Практическая](#_Задания_HTML) часть

Задание 1 (Рисунок 1, 2, 3, 4) Имеющиеся файлы Excel приводим к нормальному виду для импорта в БД.



Рисунок 1. Нормальный вид данных для импорта в БД

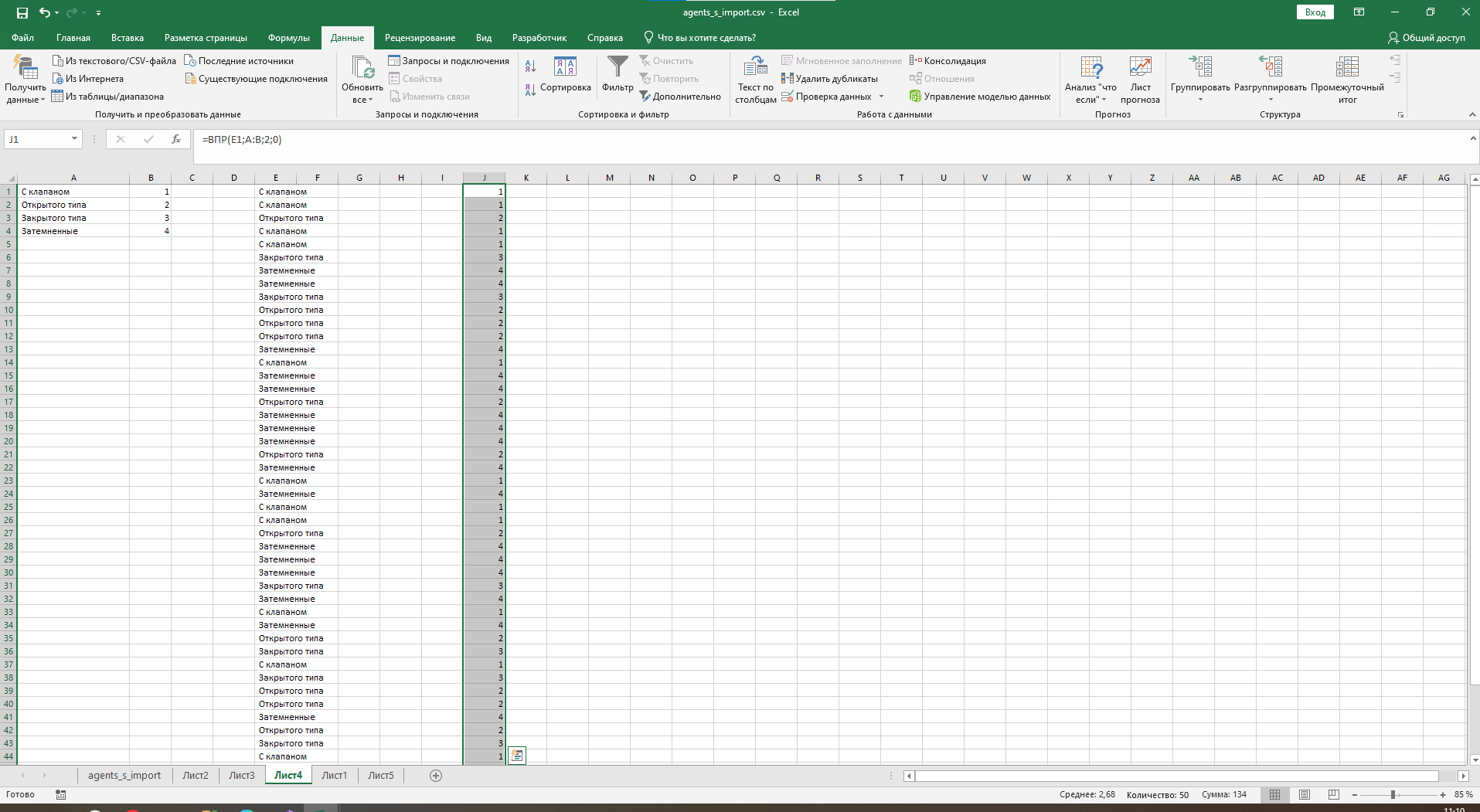


Рисунок 2. Использование функции ВПР

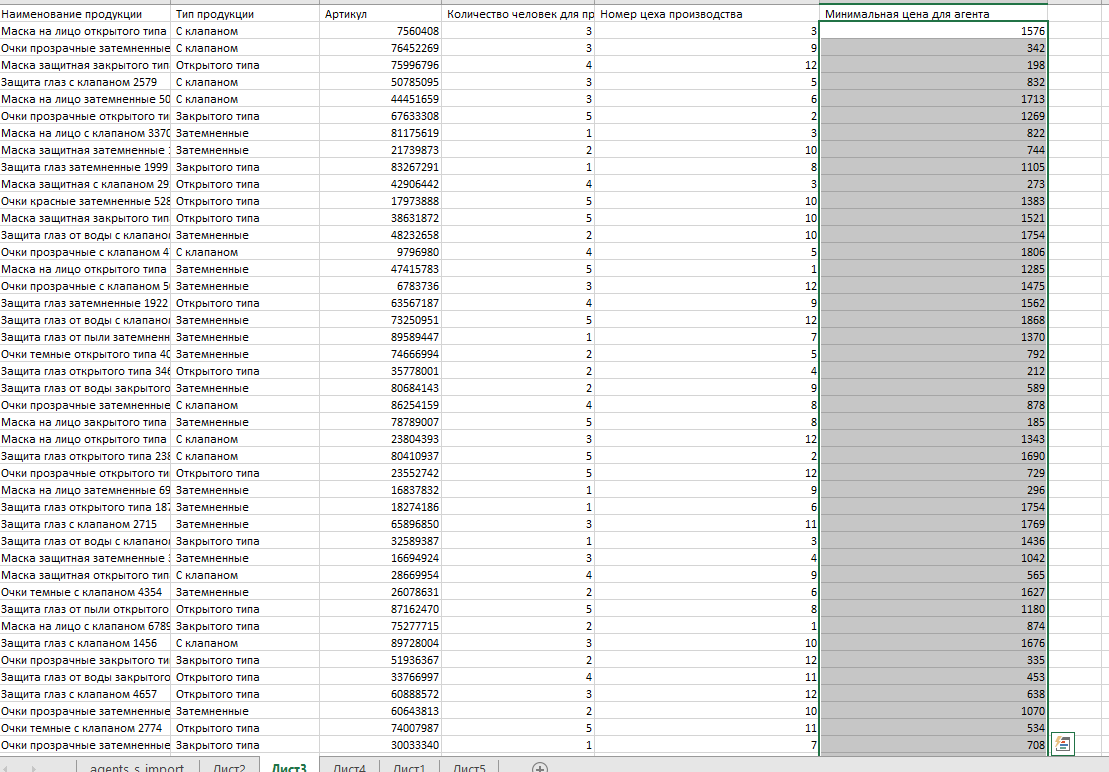


Рисунок 3. Продукция

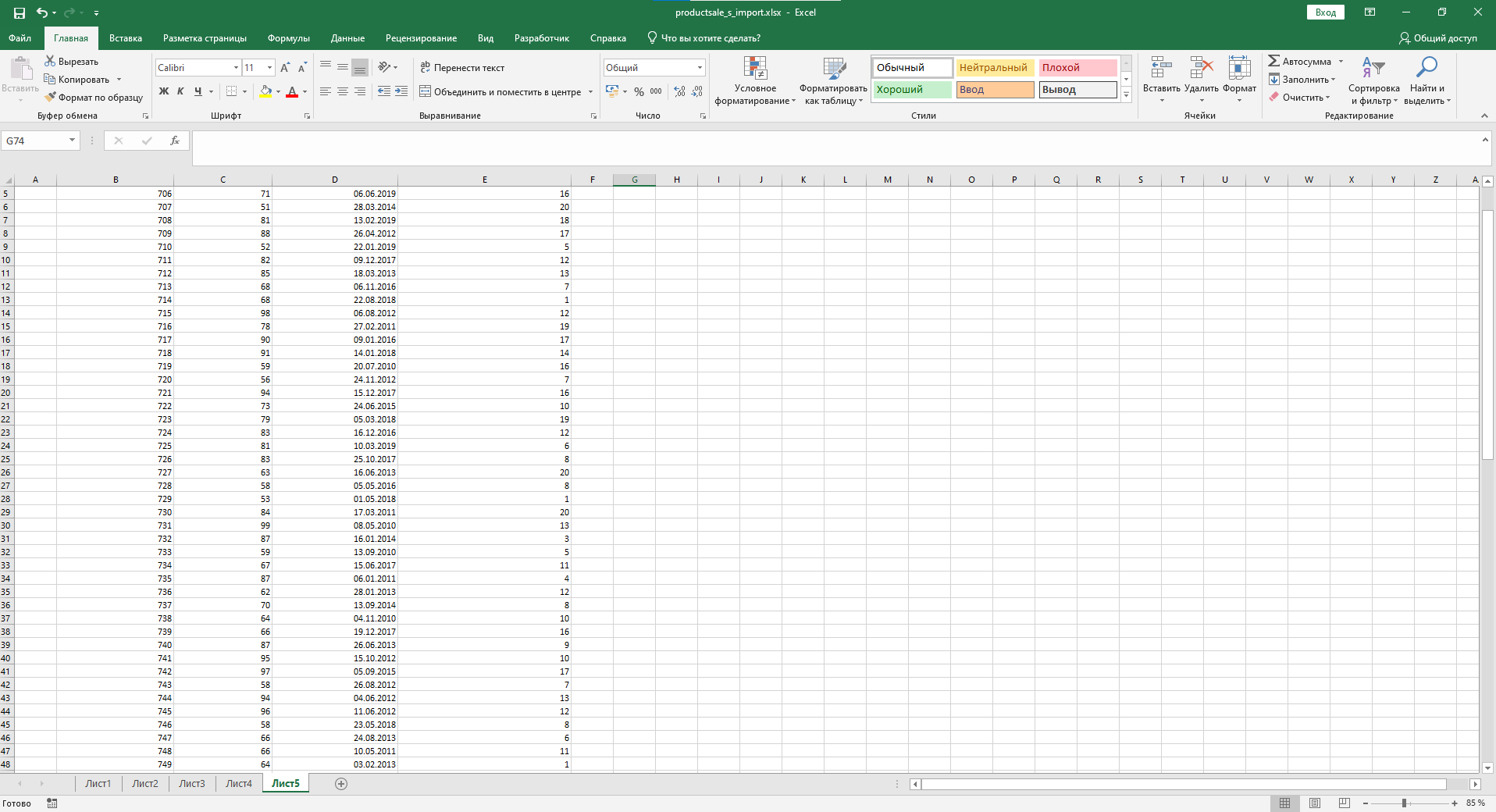


Рисунок 4. Нормальный вид данных для импорта в БД

Задание 2 (Рисунок ) Работа с Базой Данных

Начало формы

Конец формы

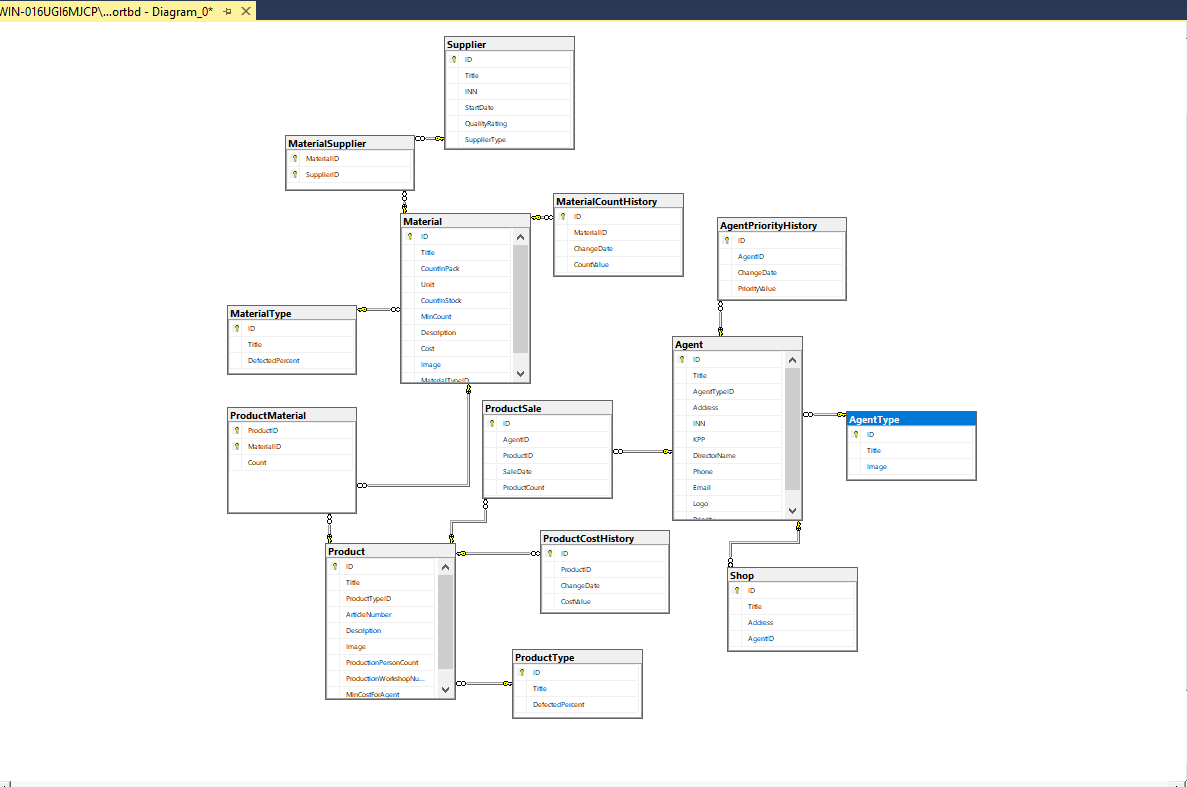


Рисунок. Диаграмма отношений

**Выводы**

Были выполнены следующие задачи:

* Проанализирована предметная область.
* Приведены данные к нормальному виду.
* Импорт данных в БД.
* Освоен язык C# и созданы несколько простых программ.

Осваиваемые профессиональные компетенции (ПК):

ПК 2.1. Разрабатывать требования к программным модулям на основе анализа проектной и технической документации на предмет взаимодействия компонент.

ПК 2.2. Выполнять интеграцию модулей в программное обеспечение.

ПК 2.3. Выполнять отладку программного модуля с использованием специализированных программных средств.

ПК 2.4. Осуществлять разработку тестовых наборов и тестовых сценариев для программного обеспечения.

ПК 2.5. Производить инспектирование компонент программного обеспечения на предмет соответствия стандартам кодирования

# Литературные источники

1. [Орлов С.А.](http://library.sgu.ru/cgi-bin/irbis64r_15/cgiirbis_64.exe?LNG&amp;Z21ID&amp;I21DBN=NIKA&amp;P21DBN=NIKA&amp;S21STN=1&amp;S21REF=1&amp;S21FMT=fullwebr&amp;C21COM=S&amp;S21CNR=20&amp;S21P01=0&amp;S21P02=1&amp;S21P03=A%3D&amp;S21STR=%D0%9E%D1%80%D0%BB%D0%BE%D0%B2%2C%20%D0%A1%D0%B5%D1%80%D0%B3%D0%B5%D0%B9%20%D0%90%D0%BB%D0%B5%D0%BA%D1%81%D0%B0%D0%BD%D0%B4%D1%80%D0%BE%D0%B2%D0%B8%D1%87) Программная инженерия. Технологии разработки программного обеспечения: учебник / С. А. Орлов. – 5-е изд., обновл. и доп. – Москва; Санкт- Петербург: Питер, 2016. – 640 с.
2. Скит Джон. C# для профессионалов. Тонкости программирования / Скит Джон – Москва; Вильямс, 2019. – 608.
3. Павловская Т.А. С/С++. Программирование на языке высокого уровня. СПб.: Питер, 2017. – 461 с.
4. Семакин И. Г. Основы алгоритмизации и программирования: учебник для студ. учреждений сред. проф. образования / И. Г. Семакин, А. П. Шестаков. — М.: Издательский центр «Академия», 2017. — 304 с
5. Хабибулин И.Ш. Программирование на языке высокого уровня. C/C++. – СПб.: БХВ– Петербург, 2017. – 512 с.