

Министерство науки и высшего образования РФ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ

Заведующий кафедрой

\_\_\_\_\_ О.В. Непомнящий

« \_\_\_\_ » \_\_\_\_\_ 2024 г.

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

Комплексная автоматизация торговой и складской деятельности  
предприятия

090401 Информатика и вычислительная техника

09.04.01.11 «Вычислительные системы и сети»

Руководитель	_____	_____	доцент каф. ВТ, канд. техн. наук	Н.Ю. Сиротинина
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Выпускник	_____	_____		В.Н. Шереметов
	<i>подпись</i>	<i>дата</i>		
Рецензент	_____	_____	доцент каф. СИИ, канд. техн. наук	А.А. Мазуров
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	
Нормоконтролер	_____	_____	доцент каф. ВТ, канд. техн. наук	Н.Ю. Сиротинина
	<i>подпись</i>	<i>дата</i>	<i>должность, ученая степень</i>	

Красноярск 2024

## СОДЕРЖАНИЕ

РЕФЕРАТ .....	4
ВВЕДЕНИЕ.....	3
1 Анализ проблемы и постановка задачи.....	5
1.1 Характеристика предприятия и процесса торгового и складского учета.....	5
1.2 Существующая технология автоматизации учета.....	6
1.3 Выявление недостатков существующей системы учета.....	7
1.4 Обоснование разработки системы автоматизации торгового и складского учета. ....	8
1.5 Анализ существующих вариантов решения задачи. ....	9
1.6 Выбор и обоснование своего варианта решения задачи.....	12
1.7 Расширенная постановка задачи. ....	13
1.8 Выводы по главе 1.....	19
2 Проектная часть.....	20
2.1 Обоснование выбора платформы разработки. ....	20
2.2 Проектирование функциональных подсистем.....	25
2.3 Архитектура разрабатываемой системы.....	31
2.4 Разработка концептуальной (инфологической) модели данных. ..	34
2.5 Выбор системы управления базами данных. ....	37
2.6 Разработка физической модели базы данных. ....	39
2.7 Выводы по главе 2.....	45
3 Разработка автоматизированной системы. ....	46
3.1 Общая концепция разрабатываемой системы.....	46
3.2 Описание объектов системы. ....	51
3.3 Разработка отчетов и запросов. ....	68
3.4 Разработка интерфейса системы. ....	75
3.5 Руководство по установке, настройке и сопровождению программы. ....	83

3.6	Руководство пользователя.....	86
3.7	Выводы по главе 3.....	92
ЗАКЛЮЧЕНИЕ .....		94
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....		96
ПРИЛОЖЕНИЕ А. ТЕКСТ ПРОГРАММНЫХ МОДУЛЕЙ .....		99

## РЕФЕРАТ

Пояснительная записка содержит 99 листа, 70 рисунков, 33 использованных источников, 1 приложение.

АВТОМАТИЗАЦИЯ, ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ, БАЗЫ ДАННЫХ, ТОРГОВЫЙ УЧЕТ, СКЛАДСКОЙ УЧЕТ ПРОГРАММИРОВАНИЕ, 1С:ПРЕДПРИЯТИЕ, ТЕХНОЛОГИЧЕСКАЯ ПЛАТФОРМА, ПРИКЛАДНОЕ РЕШЕНИЕ, КОНФИГУРАЦИЯ.

Объектом разработки является автоматизированная информационная система торгового и складского учета ООО «Продторг».

Цель работы – разработка прикладного решения для платформы «1С:Предприятие 8.3», реализующего задачу ведения торгового и складского учета.

В процессе исследования была изучена структура предприятия и процесс ведения учета, определены основные потоки данных и формы отчетов, обоснована необходимость автоматизации процесса.

В процессе проектирования на базе технологической платформы «1С:Предприятие 8.3» разработана структура объектов конфигурации, реализованы программные модули этих объектов, разработаны формы представления исходных, промежуточных и конечных данных.

Пояснительная записка содержит:

- анализ предметной области;
- описание процесса проектирование системы;
- описание программной реализации системы.

В результате реализации проекта получено работающее прикладное решение для платформы «1С:Предприятие 8.3».

Степень внедрения: система на стадии внедрения.

Пояснительная записка выполнена в текстовом редакторе MicrosoftWord 2010, слайд-презентация выполнена в MS PowerPoint 2010.

## ВВЕДЕНИЕ

Тема настоящего исследования заключается в проработке АИС, посредством которой будет осуществлена автоматизация учета деятельности общества с ограниченной ответственностью «Продторг». Известно о том, что это компания, которая осуществляет оптовую торговлю продуктами питания.

Как показывает практика многочисленных предприятий, на данный момент вести торговый учет практически невозможно, если не будет организована работа с прочими видами учета, которые тесным образом связаны с торговым учетом. Речь идет о складском учете, о расчете с клиентами и так далее. Все эти виды учета должны быть автоматизированы. Частная задача решается в рамках настоящего исследования [1].

Актуальность его заключается в том, чтобы понизить расходы, которые обычно направляются на формирование отчетов, а также документов. Наконец, стоит важная задача – предоставить возможности, с которыми сотрудники предприятия смогут постоянно контролировать процессы торгового учета [2].

Если говорить о таком направлении деятельности, как учет наличной и безналичной оплаты, то с уверенностью можно утверждать о том, что это выступает финальным этапом, который наступает в рамках торгового учета. Чтобы покупатель мог вовремя, в полном объеме оплатить товар, предстоит оформить процедуру реализации товара. Для этого, соответственно, выделяется документ, посредством которого для покупателя возникает новое обязательство. Суть его состоит в оплате за товар. Чтобы оформить реализацию продукции, товар непосредственно должен быть на витрине или же на складе. Далее, товар на витрину или на склад можно поместить в том случае, если заказать его у поставщика. Уже потом будут непосредственно изготавливаться напитки, с учетом оформления поступления момента прихода на склад, например.

Итак, все это говорит о том, что прорабатываемая АИС должна прорабатываться в качестве комплексной системы, с помощью которой

осуществляется автоматизация работы субъекта хозяйствования в сфере складского, а также торгового учета. Прежде всего, создаваемая система должна иметь возможности для введения нормативно-справочных данных; иметь возможность фиксации многочисленных фактов ведения хозяйственной активности предприятия, что связаны с движением товаров. Наконец, должна быть возможность внутри создаваемой АИС – получать отчетность в том или ином разрезе [3]. Центральная часть внутри системы – это подсистема учета, как наличной, так и безналичной оплаты от покупателей за купленную продукцию.

Цель настоящего исследования заключается в том, чтобы обосновать потребность проработки, собственно, создания системы автоматизации для складского, а также для торгового учета ООО «Продторг».

В ходе достижения поставленной цели решались следующие задачи:

- анализ проблемы и постановка задачи;
- проектирование системы;
- разработка автоматизированной системы.

Объект исследования – ООО «Продторг».

В качестве предмета настоящего исследования выступает процесс складского, а также торгового учета, в том числе процесс безналичной и наличной оплаты за купленные товары. После процесса исследования этот этап формализуется, а также реализуется в форме АИС.

В перспективе проработанная АИС будет внедрена в компании, а значит, ее будут каждый день эксплуатировать в работе сотрудники.

# 1 Анализ проблемы и постановка задачи.

## 1.1 Характеристика предприятия и процесса торгового и складского учета.

Среди многочисленных задач, которые предстоит решить в рамках настоящего исследования, проводится анализ работы ООО «Продторг». Заведение имеет ассортимент продукции – продовольственные товары.

Как и любое другое предприятие, ООО «Продторг» работает ради извлечения прибыли за счет перепродажи партий продуктов.

На рисунке 1.1 будет представлена организационная структура управления предприятия.



Рисунок 1.1 — Организационная структура управления ООО «ПРОДТОРГ».

Известно о том, что на предприятии специалисты, которые занимаются учетом наличия продукции, обрабатывают первичную информацию (расходные и приходные накладные, платежные документы, договора на поставку продукции, прочее). В отделе, который обрабатывает материалы, на основании первичных данных эксперты формируют отчеты для начальства своего отдела,

а также для генерального директора. Точно такие схемы, направленные на обработку документов, используются и в прочих департаментах субъекта хозяйствования.

## **1.2 Существующая технология автоматизации учета.**

Практика показывает, что в компании сегодня в целях ведения оперативного торгового учета применяется автоматизированная система. Она начала работу несколько лет назад. Исследование показало, что такая система преимущественно позволяет проводить расчеты с клиентами. Однако с течением времени система устарела, как морально, так и технически. Впрочем, также было установлено, что система автоматизирована не полностью. Говоря об учете по складу товаров внутри системы, то этот процесс не представлен вообще никак. Это говорит о том, что отчеты готовятся вручную с применением средств «Microsoft Excel» и «Microsoft Word». В конечном итоге, такие работы вынуждают тратить большое количество времени, а сам персонал вынужден концентрировать внимание на соответствующей работе. Далее стоит детально рассмотреть основные этапы, которые предшествуют ведению управленческого торгового учета на предприятии.

Кроме прочего, это средство применяется для достижения следующих целей:

- планировать договора;
- контроль над исполнением договоров.

Из первичных документов (ТТН, ТН, счет-фактура), формируются сведения о приходе которые вносятся в файл «Товарный отчет». Листы «Товарного отчета» распечатываются и сшиваются в книгу. Потом каждый лист получает подкрепление в виде первичных, специальных документов.

Чтобы внести оплату, для поставщиков предлагается безналичный расчет. В крайнем случае, всегда доступен проверенный вариант – расчеты с применением наличных средств. Если же оплата по договору осуществляется

наличными, то в действие вступает правило: сумма оплаты по 1 договору не должна быть выше 100000 рублей. В таком случае компания получает от контрагента кассовый чек, который в последствии прикрепляет к счету контрагента.

Далее, если рассматривать операции продаж, то они выполняются на основании безналичного расчета с использованием договоров. В других случаях применяется кассовый аппарат, после чего выдается, соответственно, кассовый чек. В конце операционного дня кассир регистрирует кассовые операции внутри Кассовой книги. Передача выручки осуществляется главным кассиром в банк с применением квитанции.

Контрагенты к своим заказам прилагают заказную смету; для нее действует произвольная форма. Точно такая смета формируется компанией на тот случай, если заказываются товары к поставщиков-контрагентов. Когда осуществляется расчет наличными с поставщиками, деньги заимствуются с кассы.

### **1.3 Выявление недостатков существующей системы учета.**

Было установлено, что имеющаяся на предприятии система учета имеет недостаток. Он связан с тем, что не имеется единого структурированного подхода к применению ПО в рамках локальной сети субъекта хозяйствования. В приоритет всегда ставится уровень, с которым пользователь владеет «MS Word» и «MS Excel». Как показывает практика, этот уровень у тех или иных пользователей не всегда одинаков. Что касается менеджеров, которые на высоком уровне владеют указанными программными продуктами, на самом деле, подобные пакеты становятся помощниками в работе. Но для тех, кто не имеет столь высоких познаний, применение информационных технологий становится бременем, а не помощником [4].

Итак, появляются трудности с оперативным выполнением договорных условий по отгрузке продукции, по ее поставке покупателям. Некоторые

проблемы также возникают при контроле над исполнением собственных платежных обязательств по причине большого количества данных, что встречаются в рамках одного сезона продаж. Соответственно, графики составляются вручную. Эти факторы, вне всяких сомнений, являются негативными; понижают производительность труда, увеличивают время на работу, приводят к ошибкам в работе. Как итог, страдают финансовые параметры работы предприятия, прибыль.

#### **1.4 Обоснование разработки системы автоматизации торгового и складского учета.**

Не возникает сомнений в том, что интеграция АИС по складскому и торговому учету – это операция, которая позволит значительно снизить накладные, а также управленческие расходы. Кроме прочего, значительно вырастет время на прием заявок от клиентов; улучшится логистика поставок, а простой сотрудников предприятия – практически будут нивелированы.

Не менее важным фактом является то, что с АИС работать намного удобнее, чем с простыми носителями информации, формируемыми на бумаге. Далее, многопользовательский режим работы – это еще одно важное преимущество, которое дает АИС [3].

Подробный разбор всех аспектов бизнес-процессов, сопряженных с ведением учёта коммерческих операций торговой компании, позволяет на основании глубокого анализа прийти к заключению об относительно низкой эффективности и высокой трудоёмкости решения данного класса задач с использованием традиционных подходов, базирующихся на работе с бумажной документацией.

Существенным преимуществом, которое обеспечивает применение автоматизированных информационных систем, является повышенный уровень защищённости сохраняемых в них данных. В отличие от ситуации с материальными носителями, когда утрата отдельного документа неизбежно

вела к утрате соответствующей информации, при электронном хранении всегда существует возможность восстановления файла по техническим или иным причинам.

Это принципиально меняет подходы к обеспечению надёжности и доступности критически важных для бизнеса данных в пользу автоматизированных систем в сравнении с традиционными методами ведения учёта.

Соответственно, далее будет целесообразно назвать некоторые преимущества, которые даёт АИС на базе ВТ в целях решения задач при построении АИС складского, а также торгового учёта [5]:

- увеличение темпов внесения данных;
- удобство отбора, а также поиска информации;
- увеличение скорости при формировании максимально точной отчетности;
- обеспечение безопасности информации;
- возможность многопользовательской работы.

### **1.5 Анализ существующих вариантов решения задачи.**

Для достижения цели совершенствования системы управления предприятием, осуществляющим коммерческую деятельность на рынке, существует множество эффективных информационных технологий, позволяющих оптимизировать все ключевые бизнес-процессы. Одним из ярких примеров функционального решения является автоматизированная система управления 1С: Управление торговлей 8, разработанная ведущей отечественной компанией 1С в сфере прикладных программных продуктов [6].

Данное масштабное прикладное программное обеспечение обеспечивает непрерывную автоматизацию широкого спектра процессов, характеризующих деятельность торговой организации, включая: ведение учёта поступлений и реализации товаров, материально-вещественных ценностей; формирование и

контроль за выполнением коммерческих предложений контрагентам; ведение переговорного процесса с партнерами; оптимизацию закупок и складского хозяйства; подготовку комплексной отчетности для анализа бизнес-процессов и принятия управленческих решений; электронный документооборот с контрагентами.

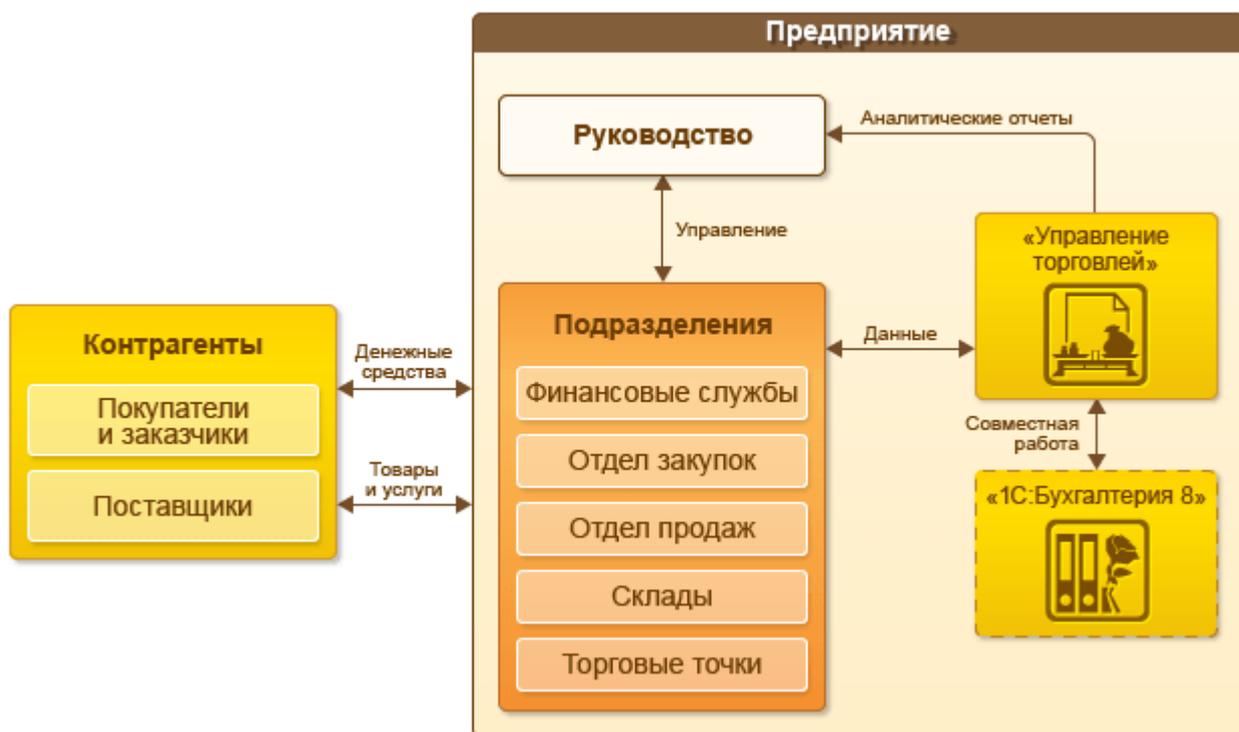


Рисунок 1.2 — Структура системы "1С:Управление торговлей 8".

В рамках данного комплексного многопланового исследования был осуществлён всесторонний анализ целого ряда программных продуктов, которые потенциально могли бы обеспечить необходимую автоматизацию ключевого для предприятия бизнес-процесса – приёмки и дальнейшей обработки заказов от клиентов.

Вначале подробно рассматривалось одно из наиболее известных и распространённых программных решений, относящееся к так называемому "громоздкому" типу. Экспертная оценка детально выявила чрезмерную излишнюю сложность его функциональных возможностей, неудобный и явно не оптимальный пользовательский интерфейс, затрудняющий быструю и

надёжную работу с заказчиками. Кроме того, платная схема лицензирования данного программного продукта была признана нецелесообразной в рамках ресурсных ограничений текущего проекта [6].

Наряду с этим подробно рассматривалась и специализированная программа "Клад-Перл", ориентированная на автоматизацию управления товарными потоками. Несмотря на то, что она позволяла осуществлять полноценный контроль над движением товаров на всех стадиях, её универсальность и невозможность гибкой адаптации под конкретную узкую задачу проекта сделали её фактически неприменимой. Кроме того, присутствовали такие ограничения, как платная лицензия и необходимость обучения сотрудников [7].

Аналогичным образом был проведен анализ широко известного программного решения "1С: Торговля и склад", однако сверхфункциональность данного программного обеспечения, отсутствие специализированных инструментов для быстрого и точного ввода данных при общении с клиентами в процессе приема заказов, а также прекращение его официальной поддержки и развития разработчиком делают его непригодным для применения в рамках настоящей исследовательской работы [8].

После всестороннего изучения существующих готовых программных решений было принято обоснованное и взвешенное решение о разработке собственного специализированного программного продукта с учетом следующих принципиальных моментов. Во-первых, максимальное использование имеющихся в распоряжении компании программно-аппаратных ресурсов с целью повышения экономической эффективности и рационального расходования средств. Во-вторых, адаптация проектных решений к существующему технопарку и информационной инфраструктуре компании для обеспечения совместимости. В-третьих, обеспечение полной совместимости с уровнем пользовательских навыков и квалификацией персонала компании.

Для минимизации затрат и экономии бюджета было принято решение создать специализированное программное обеспечение в виде конфигурации на

базе широко распространенной и популярной системы "1С: Предприятие 8.3", сочетающей преимущества проверенной и надежной платформы с возможностью реализации узкоспециализированного функционала, необходимого для решения конкретных задач автоматизации бизнес-процессов компании по приему и обработке заказов клиентов.

Таким образом, выбранный подход позволит создать экономически эффективное, оптимизированное и полностью адаптированное программное решение, в полной мере соответствующее специфическим требованиям предприятия и уровню квалификации пользователей, максимально задействовав при этом существующую информационно-технологическую инфраструктуру компании [9].

### **1.6 Выбор и обоснование своего варианта решения задачи.**

В процессе выполнения проекта, практически сразу была определена направленность на создание своего ПП. Прежде всего, это обуславливается тем, что в своей разработке в любое время можно реализовать характерные, исключительно важные для своего предприятия возможности. Вовсе не исключается тот факт, что новые требования могут появиться в процессе работы системы. И такие требования можно будет учесть только лишь в том случае, если будет собственная разработка, ведь исходных кодов у продаваемых на рынке ПП нет, а значит, дополнения туда внести невозможно [10].

Итак, создавать систему нужно на базе определенной распространенной, универсальной системы. Сам факт распространенности уже говорит о том, что функционал программного продукта в будущем смогут поддерживать любые эксперты. Универсальность – это качество, которое дает возможность дополнять систему функциями, которые на начальных этапах в ПП заложены не будут. В качестве примера подобной системы можно выбрать 1С: Предприятие 8.3 [6].

Новая система автоматизации будет проектироваться в качестве продукта комплексной автоматизации, в рамках которого будут воедино собраны все векторы работы субъекта хозяйствования [11].

### **1.7 Расширенная постановка задачи.**

Разрабатываемая автоматизированная информационная система (АИС) должна обеспечивать комплексную и всестороннюю автоматизацию ключевых бизнес-процессов торгового предприятия на всех уровнях. С целью обеспечения оптимальной функциональности, гибкости и удобства использования архитектура АИС предусматривает ее модульное построение с разделением на следующие основные подсистемы: подсистему ведения и управления справочной информацией, подсистему торгового и складского учета операций, а также подсистему взаиморасчетов и работы с контрагентами [12].

Подсистема "Справочники" является ядром и основой информационной базы данных и предназначена для хранения, обработки и управления базовыми нормативно-справочными сведениями, необходимыми для полноценного и корректного функционирования остальных модулей АИС в комплексе. Она разделена на три отдельные составляющие модуля в соответствии с предметными областями:

1. "Контрагенты" - предназначена для ведения реестра контрагентов предприятия (покупателей и поставщиков) с возможностью регистрации полных реквизитных данных, контактной информации, истории взаимодействия и учета взаиморасчетов.

2. "Номенклатура" - обеспечивает хранение, классификацию и оперативный учет полного ассортимента товарных позиций и номенклатуры. На рисунке 1.3 наглядно представлены ключевые функции данной подсистемы, включающие управление товарным каталогом, учет текущих товарных остатков на складах, назначение и регулирование торговых наценок и т.д.

3. "Организация" - содержит полные реквизитные данные и учетную информацию юридического лица-владельца внедряемой АИС на предприятии.

Полноценное внедрение и корректная первичная настройка базовых компонентов подсистемы "Справочники" являются критически важными для последующего корректного функционирования всех остальных компонентов АИС и бесперебойного автоматизированного ведения ключевых бизнес-процессов на торговом предприятии. Именно от качества и полноты реализации данного основополагающего модуля напрямую зависит общая эффективность использования информационной системы в целом.



Рисунок 1.3 — Ведение номенклатуры.

На рисунке 1.4 представлены функции подсистемы «Контрагенты».



Рисунок 1.4 — Ведение контрагентов.

На рисунке 1.5 показан состав функций подсистемы «Организация».



Рисунок 1.5 — Ведение данных организации.

Функции подсистемы «Продажи» ее приведены на рисунке 1.6.



Рисунок 1.4 — Функции подсистемы «Продажи».

Функции подсистемы «Склад» представлены на рисунке 1.7.

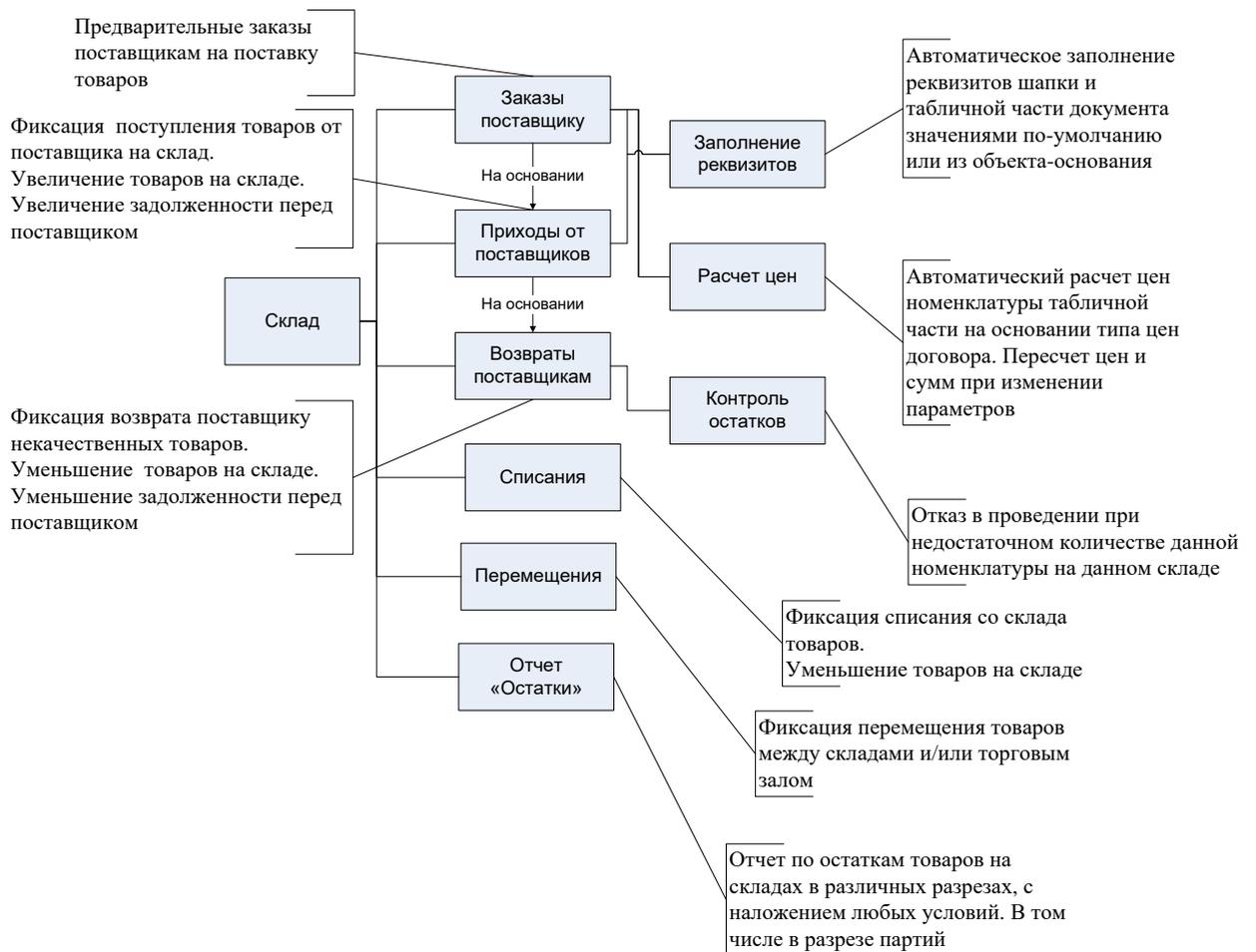


Рисунок 1.5 — Функции подсистемы «Склад».

Функции подсистемы «Взаиморасчеты» отражены на рисунке 1.8.



Рисунок 1.6 — Функции подсистемы «Взаиморасчеты».

## 1.8 Выводы по главе 1.

Рассмотрена текущая система работы предприятия и выявлены недостатки. На основе этой информации был выбран и обоснован способ разработки АИС. Проанализированы существующие технологии автоматизации учета деятельности предприятий, выбран более оптимальный вариант в виде "1С: Предприятие 8.3". На основе вышперечисленного была составлена расширенная постановка задачи, в которой подробно продемонстрирована структура разрабатываемой АИС.

## **2 Проектная часть.**

### **2.1 Обоснование выбора платформы разработки.**

Как было обоснованно и аргументированно показано ранее, в качестве базовой платформы для разработки специализированного приложения по автоматизации процессов приема и обработки заказов была выбрана популярная и широко распространенная система "1С: Предприятие 8.3". Данное программное решение де-факто стало отраслевым стандартом и наиболее популярным инструментом для создания коммерческих приложений, ориентированных на российский рынок и специфику ведения бизнеса в России. Ключевыми преимуществами и сильными сторонами платформы "1С: Предприятие" являются наличие мощных и гибких инструментов для построения эффективных и масштабируемых систем складского и торгового учета операций, а также встроенная поддержка широкого спектра различных учетных задач и бизнес-процессов.

Высокая популярность и распространенность "1С: Предприятие 8.3" обусловлена наличием обширной линейки типовых отраслевых решений и конфигураций, охватывающих практически все сферы коммерческой деятельности и отрасли экономики. Благодаря этому коммерческие предприятия различных форм собственности, масштабов и направлений бизнеса имеют возможность выбрать оптимальное типовое решение, максимально соответствующее специфике их деятельности, и на его основе развернуть полнофункциональную информационную систему автоматизации управления.

Выбор "1С: Предприятие 8.3" в качестве базовой платформы для разработки специализированного приложения обусловлен рядом ключевых факторов и преимуществ. Во-первых, это обеспечивает максимальную совместимость создаваемого продукта с существующими корпоративными стандартами и практиками автоматизации бизнес-процессов в России. Во-

вторых, разработчики получают в распоряжение проверенный и зарекомендовавший себя инструментарий, позволяющий в кратчайшие сроки создать эффективное специализированное решение высокого качества. В-третьих, за счет возможности использования и интеграции готовых типовых конфигураций существенно сокращаются затраты на реализацию большей части стандартного функционала, уже присутствующего в базовых отраслевых решениях.

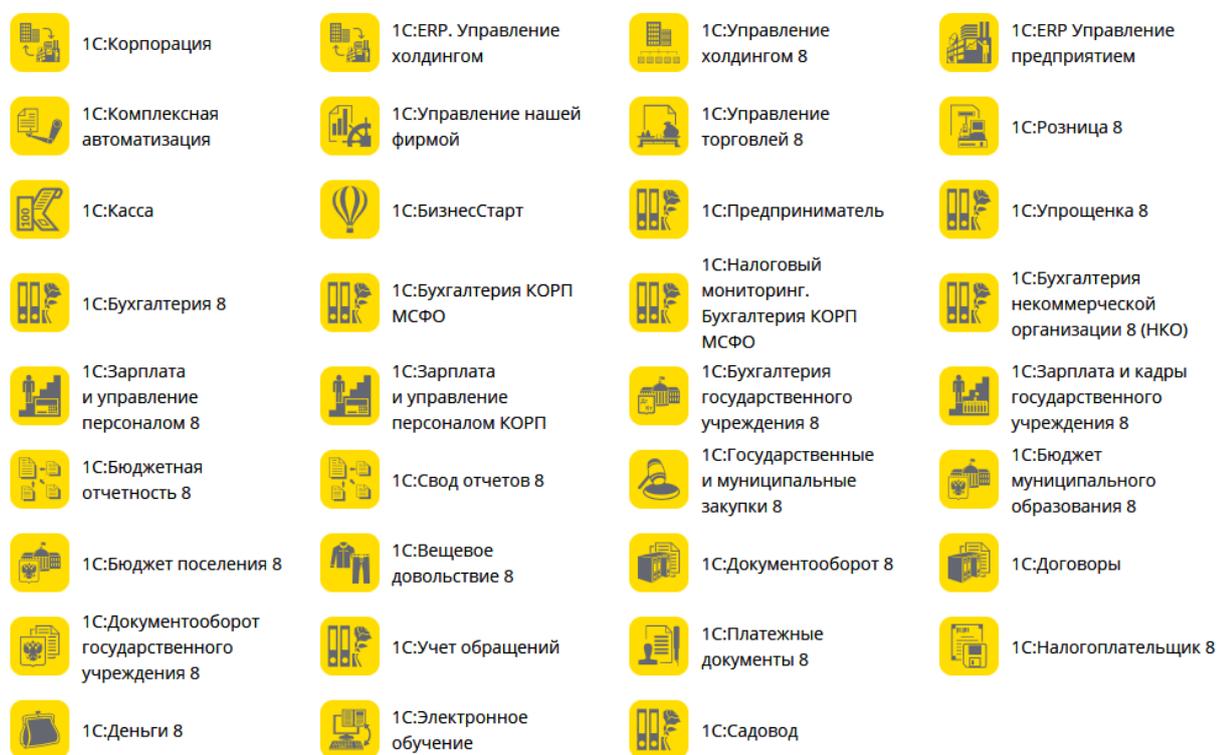


Рисунок 2.1 — Список типовых решений фирмы «1С».

Помимо широкого спектра типовых решений, охватывающих потребности и специфику деятельности большинства коммерческих предприятий различных отраслей, технологическая платформа "1С: Предприятие 8.3" предлагает целый ряд специализированных отраслевых решений и конфигураций, ориентированных на комплексную автоматизацию характерных бизнес-процессов в различных узкоспециализированных сферах деятельности. В качестве наглядного примера можно привести решение "1С:

Бухгалтерия птицефабрики", название которого четко и недвусмысленно указывает на его уникальную предметную область применения и отраслевую специализацию. Следует особо отметить, что линейка специализированных отраслевых решений "1С" является чрезвычайно обширной и охватывает практически все существующие сектора экономики и сферы коммерческой деятельности.

Одним из ключевых факторов высокой популярности и широкого распространения платформы "1С: Предприятие 8.3" у разработчиков и конечных пользователей является наличие мощной и разветвленной системы профессиональной пользовательской технической поддержки. На текущий момент по всей стране функционирует и оказывает услуги свыше 7500 официальных партнерских центров и офисов, обеспечивающих полный комплекс сервисных услуг для заказчиков и пользователей решений "1С" по внедрению, сопровождению и техническому обслуживанию.

Для разработчиков прикладных решений и специализированных конфигураций на базе универсальной технологической платформы "1С: Предприятие 8.3" одним из главных преимуществ и сильных сторон является жесткое разделение самой платформы и создаваемых на ее основе прикладных конфигураций. Такой системный подход обеспечивает открытость прикладных решений для модификации и доработки, а также позволяет рассматривать платформу как универсальную среду для создания принципиально новых приложений практически с нуля, выходящую далеко за рамки классического исключительно бухгалтерского программного обеспечения.

Функциональные возможности "1С: Предприятие 8.3" действительно колоссальны по своему объему и разнообразию (частично они представлены на рисунке 2.2). Богатый и гибкий инструментарий данной технологической платформы позволяет классифицировать ее как полноценную Framework-систему для разработки приложений корпоративного класса с широчайшим спектром функций. По совокупности функциональных возможностей и технологических преимуществ данное программное решение не только не

уступает мировым аналогам от зарубежных вендоров, но и во многих ключевых аспектах превосходит западные предложения на рынке.

Таким образом, обширность линейки типовых и специализированных отраслевых решений, развитая система профессиональной пользовательской поддержки, а также универсальность самой технологической платформы и богатство ее инструментария обуславливают высокую популярность, востребованность и распространенность "1С: Предприятие 8.3" как в качестве готового прикладного решения для автоматизации, так и в роли среды разработки специализированных приложений под требования конкретных заказчиков.

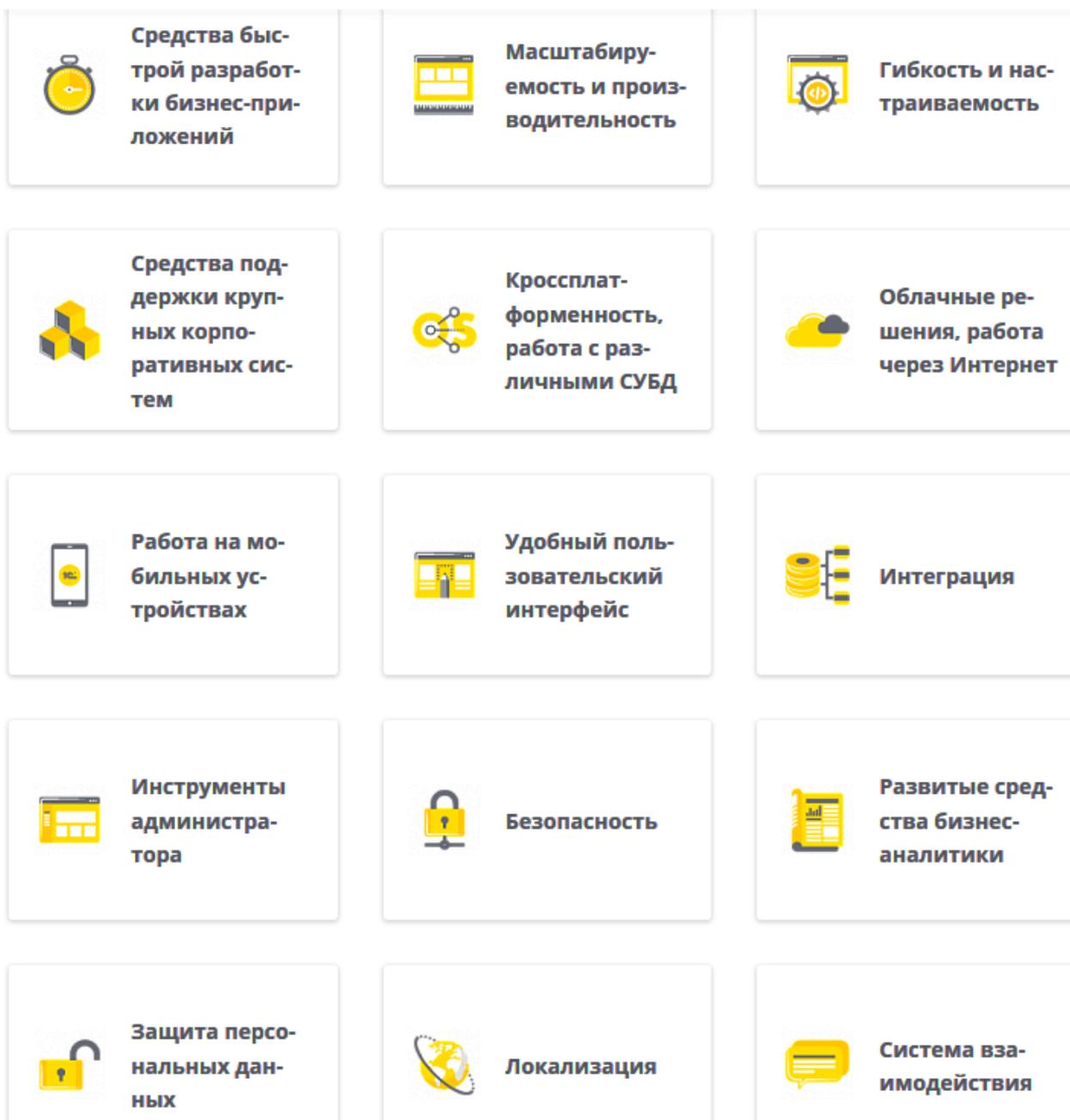


Рисунок 2.2 — Некоторые из возможностей платформы «1С:Предприятие 8.3».

Необходимо отметить, что программная платформа 1С: Предприятие 8.3, будучи предназначенной для автоматизированного моделирования сложных бизнес-процессов, реализованных на отечественных промышленных и коммерческих предприятиях, не может рассматриваться в качестве универсального инструмента для разработки приложений любого функционала. Данное масштабное программное решение имеет четко обозначенную

предметную сферу применения, охватывающую автоматизацию широкого спектра экономических и организационных процессов, характерных для современного бизнеса.

Однако именно ограниченная узкоспециализированная направленность платформы 1С: Предприятие 8.3 позволяет максимально точно учитывать специфику бизнес-моделей российских компаний при разработке методологий, алгоритмов и технологий программирования. Это обеспечивает гибкость, высокую эффективность и скорость реализации прикладных решений, их последующую модификацию под динамично меняющиеся рыночные условия, максимальную отдачу от использования данной платформы в целях автоматизации всех ключевых бизнес-процессов предприятия.

## **2.2 Проектирование функциональных подсистем.**

Для всестороннего представления архитектуры информационной системы, разрабатываемой на базе программной платформы 1С:Предприятие 8.3, целесообразно использовать модель подсистем, позволяющую выделить основополагающие функциональные блоки по принципу их логического назначения и взаимосвязи. Данный подход обеспечивает структурирование сложного программного обеспечения по направлениям деятельности, реализуемым каждым из блоков.

Так, на рисунке 2.3 представлен набор ключевых подсистем, задающих основную функциональную архитектуру: подсистема управления справочниками и метаданными, отвечающая за целостность и доступность справочной информации; подсистема учета основных и вспомогательных бизнес-процессов; подсистема взаимодействия с конечными и административными пользователями через веб-интерфейсы и интеллектуальные формы; подсистема аналитики и формирования отчетных данных на основе массивов сведений из учётных регистров.

Такая модель даёт полное представление логической структуры системы, взаимосвязей между её элементами, позволяет подробно описать предметную область и функциональные особенности.

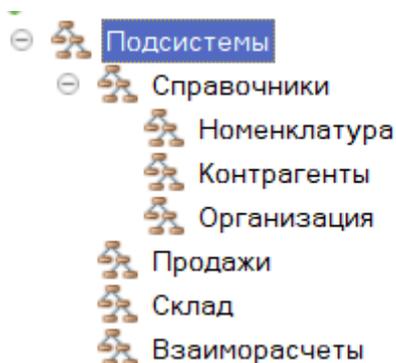


Рисунок 2.3 — Состав подсистем конфигурации.

Подсистема «Номенклатура», являющаяся подсистемой подсистемы «Справочники», объединяет в себе объекты, относящиеся к ведению номенклатуры продукции. Ее состав представлен на рисунке 2.2.

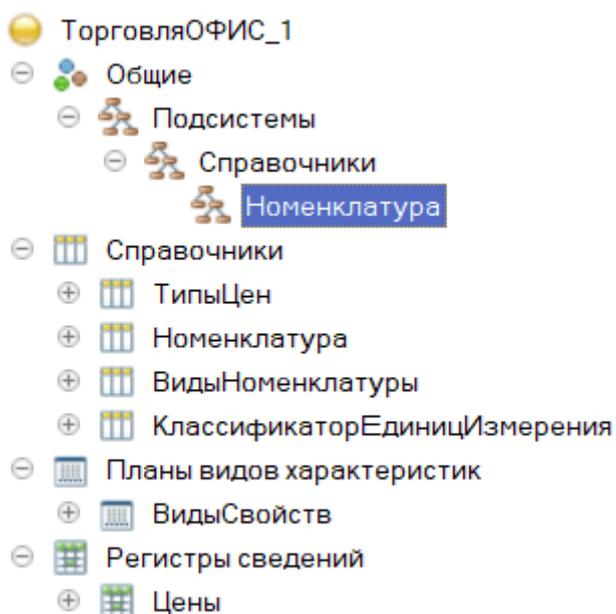


Рисунок 2.4 — Состав подсистемы «Номенклатура».

В подсистему «Контрагенты», которая также является подсистемой подсистемы «Справочники» входят объекты, представленные на рисунке 2.3.

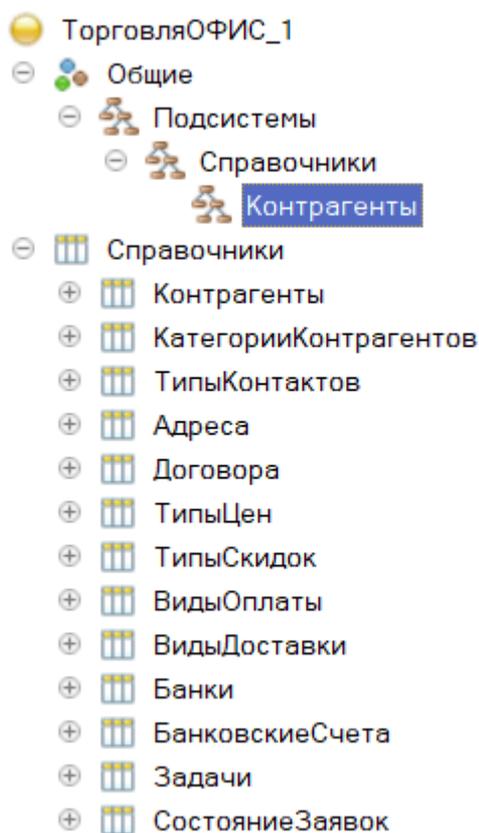


Рисунок 2.5 — Состав подсистемы «Контрагенты».

Третья подсистема «Организация» (Рисунок 2.6) подсистемы «Справочники» объединяет в себе объекты, относящиеся к ведению данных собственной организации.

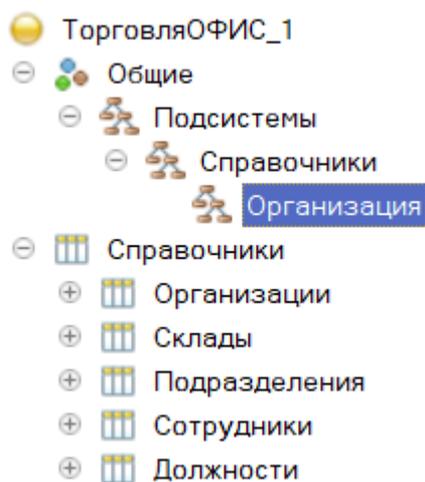


Рисунок 2.6 — Состав подсистемы «Организация».

Непосредственно ведение документов по хозяйственной деятельности разбито на три подсистемы: «Продажи», «Склад» и «Взаиморасчеты» функционально и интерфейсно объединяющие в себе объекты конфигурации, относящиеся к тому или иному блоку. Состав подсистемы «Продажи» представлена на рисунке 2.5. Состав подсистемы «Склад» представлена на рисунке 2.6. Состав подсистемы «Взаиморасчеты» представлена на рисунке 2.7.

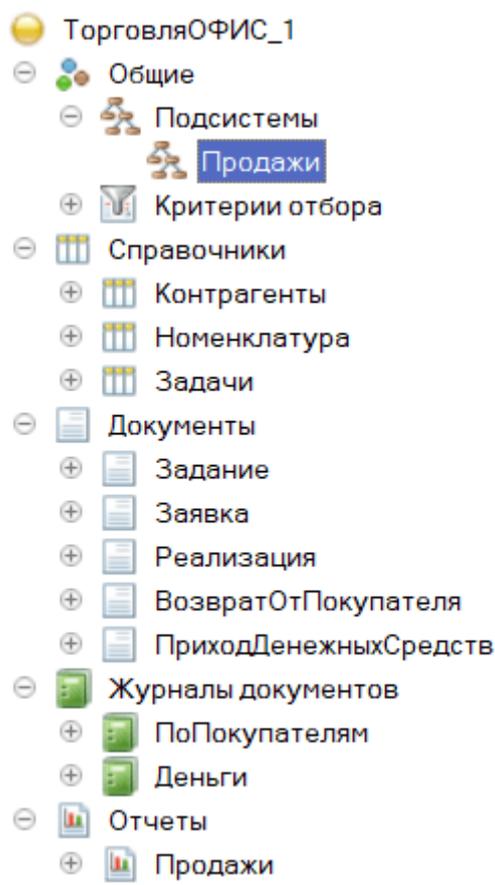


Рисунок 2.7 — Состав подсистемы «Продажи».

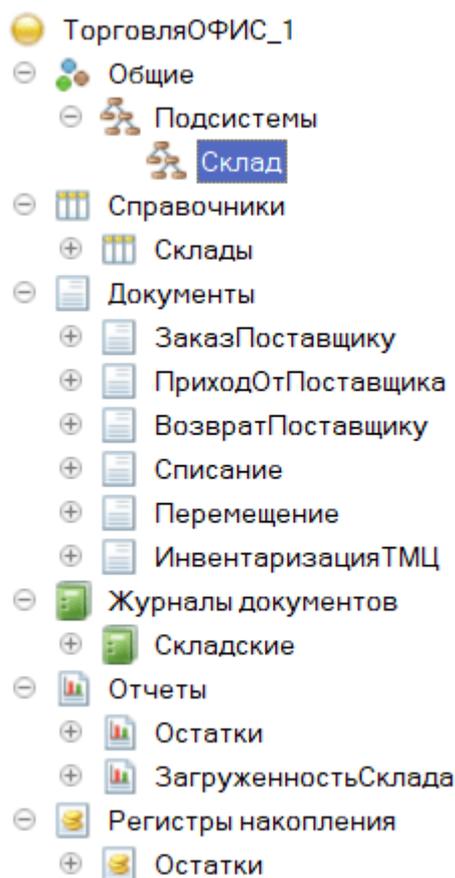


Рисунок 2.8 — Состав подсистемы «Склад».

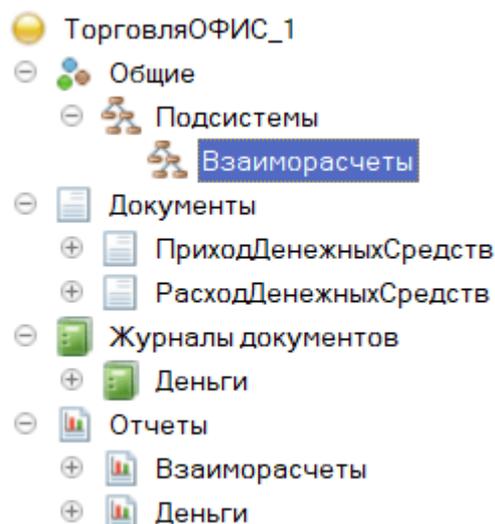


Рисунок 2.9 — Состав подсистемы «Взаиморасчеты»

Важной особенностью платформы «1С:Предприятие 8.3» является то, что задав вложенность и состав подсистем, программист тем самым определяет

состав главного меню разрабатываемого приложения. Точнее этот состав создается автоматически на основе командного интерфейса подсистем, задаваемого программистом на основе состава подсистем. Например, на рисунке 2.8 показан командный интерфейс подсистемы «Продажи». В результате платформа «1С:Предприятие 8.3» автоматически на основе командного интерфейса создаст разделы главного меню и структуру каждого раздела. В качестве примера на рисунке 2.9 показан состав раздела «Продажи» как его видит пользователь.

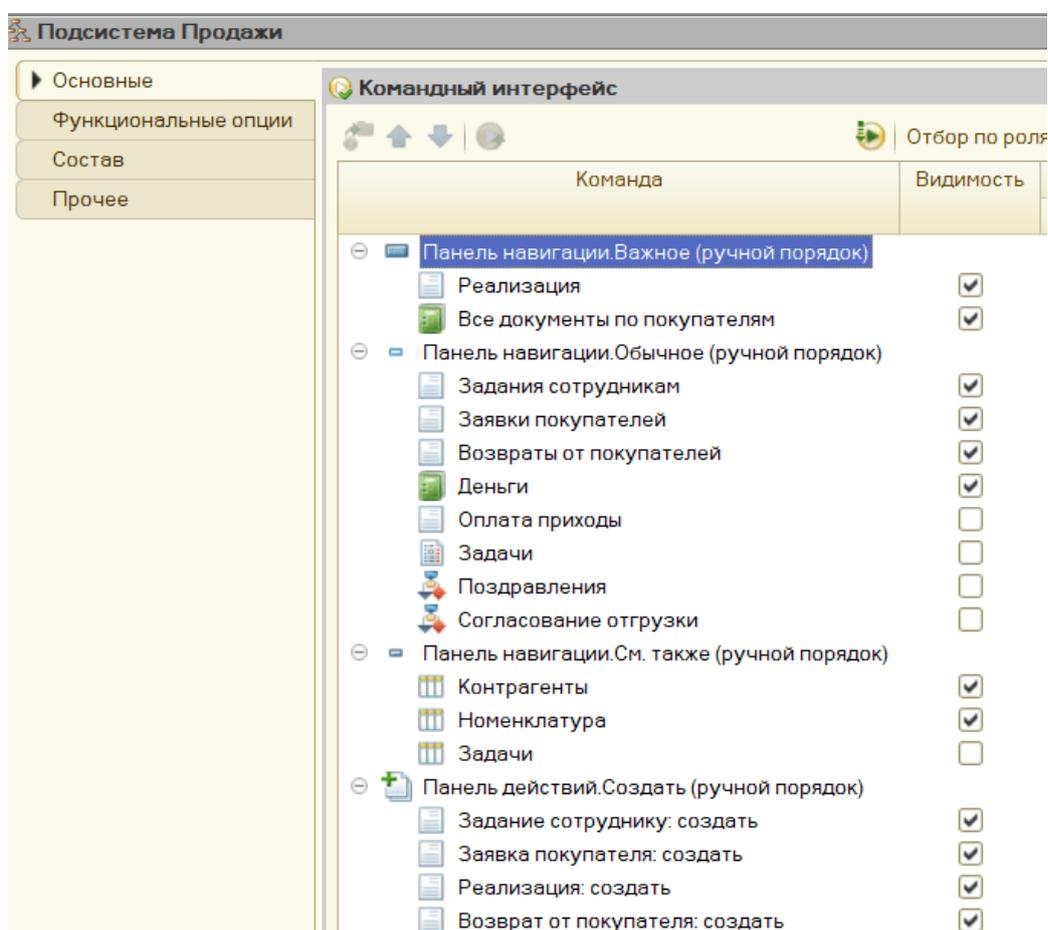


Рисунок 2.10 — Командный интерфейс подсистемы «Продажи».

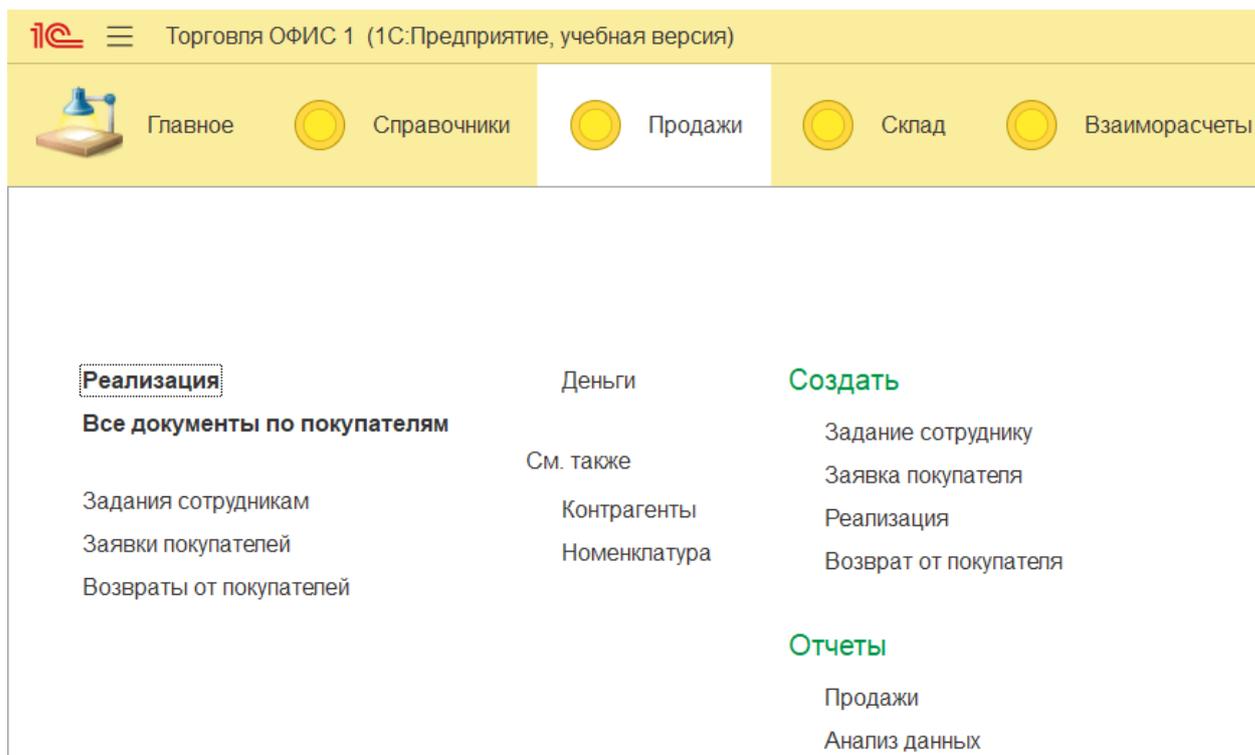


Рисунок 2.11 — Состав раздела «Продажи».

Таким образом, создав иерархическую структуру подсистем разрабатываемой конфигурации и задав командный интерфейс каждой подсистемы мы автоматически создали удобный структурированный интерфейс доступа пользователя ко всем функциям разрабатываемой системы.

### 2.3 Архитектура разрабатываемой системы.

Как было подробно разъяснено и акцентировано ранее, в рамках данной выпускной квалификационной работы рассматривается процесс разработки специализированной корпоративной конфигурации на базе популярной и широко используемой технологической платформы "1С: Предприятие 8.3". Соответственно, архитектура создаваемой автоматизированной информационной системы (АИС) должна строго соответствовать всем требованиям, стандартам и спецификациям, предъявляемым к решениям, построенным на основе данной универсальной платформы [6].

Архитектура АИС, разработанной с использованием технологической платформы "1С: Предприятие 8.3", обладает рядом специфических уникальных черт и характеристик:

- полностью отсутствует необходимость в выделении и использовании дополнительного специализированного сервера приложений, так как все соответствующие функции реализуются и выполняются непосредственно мощной серверной частью самой платформы "1С".

- система имеет классическую распространенную файл-серверную архитектуру с централизованным единым хранилищем всех данных в единой базе данных.

- в качестве полнофункциональной системы управления базами данных (СУБД) используется встроенная в технологическую платформу "1С" собственная проприетарная СУБД.

База данных и основная информационная база создаваемой автоматизированной информационной системы должны быть размещены на выделенном специализированном файл-сервере в пределах локальной корпоративной сети предприятия-заказчика. Это обеспечит возможность быстрого и удобного доступа всех авторизованных пользователей к актуальным оперативным данным непосредственно с их рабочих клиентских станций. В случае достаточной производительности и вычислительной мощности уже имеющегося на предприятии сервера, он может быть использован для хранения базы данных и информационной базы без необходимости установки дополнительного стороннего программного обеспечения или СУБД, кроме самой технологической платформы "1С: Предприятие 8.3" с уже интегрированной в нее полнофункциональной СУБД [13].

На клиентских рабочих станциях конечных пользователей также не требуется выполнять дополнительную установку и настройку каких-либо сторонних программных продуктов и компонентов, кроме клиентской части самой платформы "1С: Предприятие 8.3". При этом операционная система,

установленная на клиентских машинах, может быть любой из семейства ОС Windows.

Рисунок 2.10 наглядно отображает в виде упрощённой схемы модель взаимосвязанности ключевых элементов программного и информационного обеспечения, которые были интегрированы в структуру разрабатываемой автоматизированной корпоративной информационной системы на базе технологической программной платформы "1С: Предприятие 8.3".

Данная схематическая модель в доступной форме демонстрирует иерархию взаимодействия различных уровней программного обеспечения - от уровня операционной системы и серверных компонентов до уровня приложений и интегрированных в систему информационных баз данных. Таким образом, иллюстрация позволяет проиллюстрировать логику взаимосвязанности программных и информационных ресурсов, входящих в состав рассматриваемой архитектуры автоматизированной системы управления предприятием в целом.

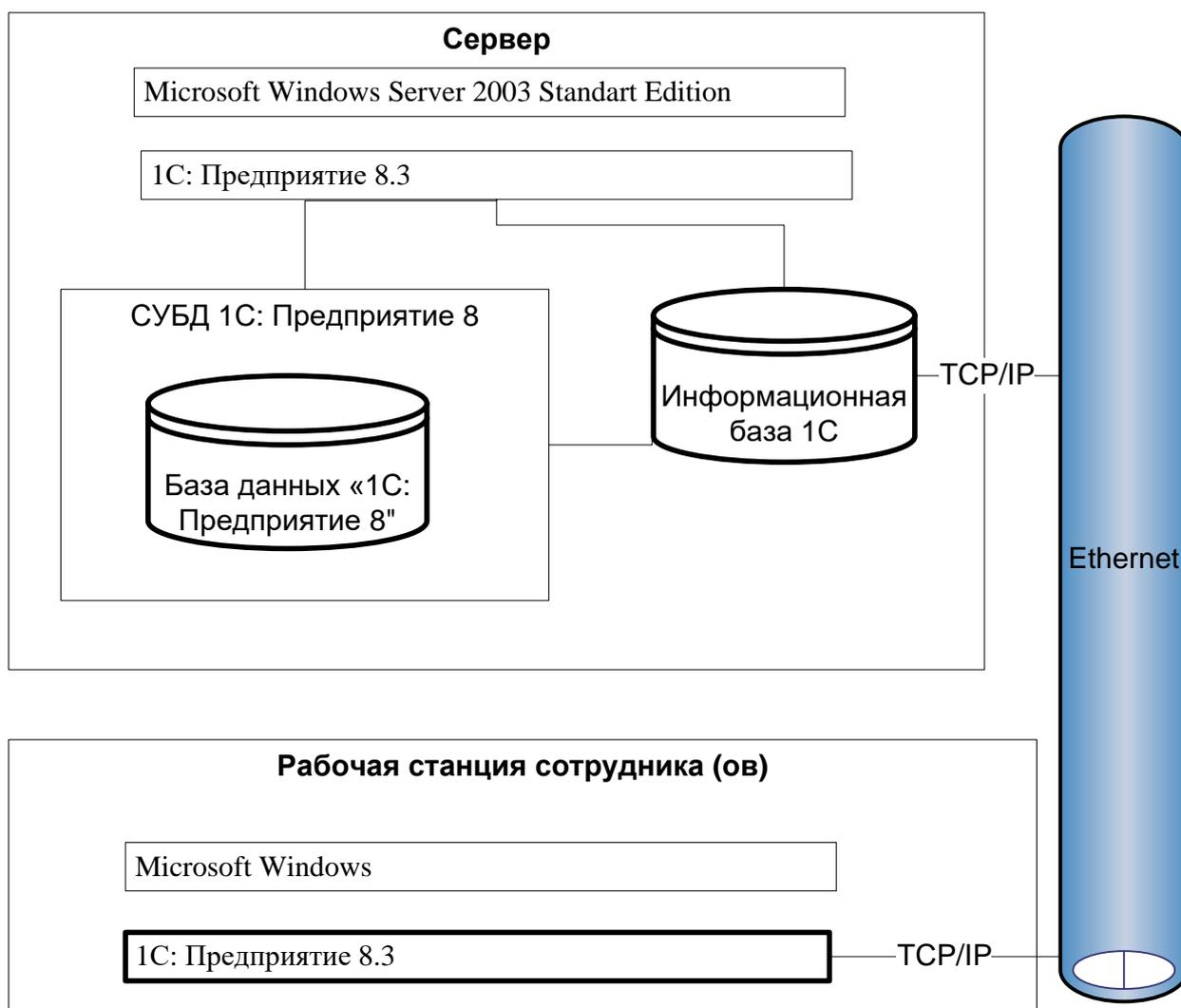


Рисунок 2.12 — Архитектура разрабатываемой системы.

## 2.4 Разработка концептуальной (инфологической) модели данных.

Завершив комплекс проектировочных работ, разработчики успешно создали информационную модель перспективной автоматизированной системы, отразившую сущности, объекты и отношения между ними, характерные для её функциональной среды.

В частности, модель представляла собой детальную схематическую спецификацию структуры данных и логических блоков программного обеспечения с указанием функционального назначения каждого элемента. Как следует из рисунка 2.11, в ней были чётко указаны основные типы сведений, хранимых в информационной базе, состав и взаимосвязь справочных данных,

алгоритмы обработки потоков информации отдельными модулями для эффективного управления разнообразными предметными процессами.

Полученная модель в полной мере охватывала структуру и компоновку элементов информационной среды, обеспечивая концептуально-функциональное представление каждой ее части и целостности в целом. Это позволяло использовать ее как основу для последующей реализации проекта на уровне информационных технологий.

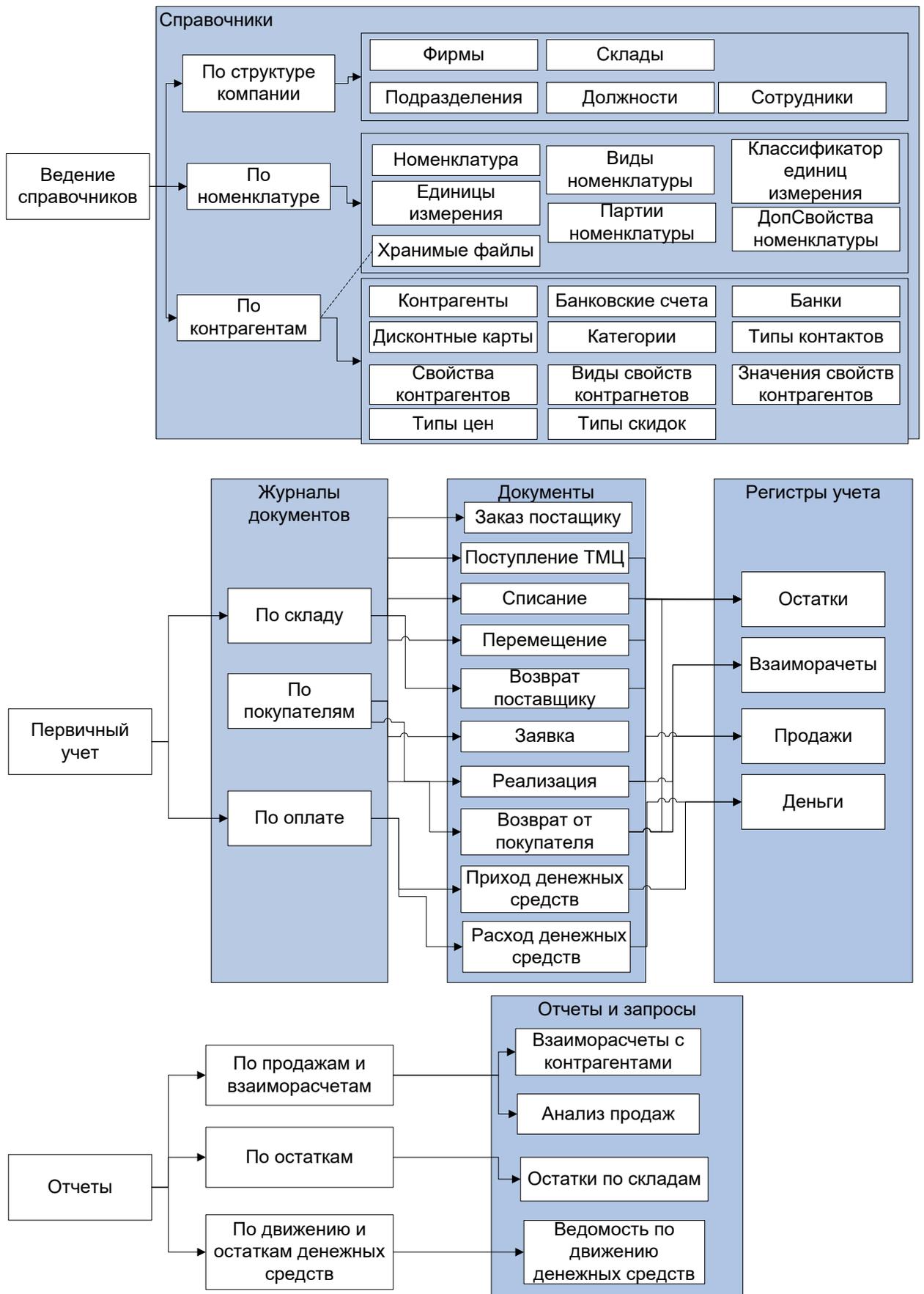


Рисунок 2.13 — Информационная модель.

## 2.5 Выбор системы управления базами данных.

Известно о том, что 1С: Предприятие 8.3 работает с некоторыми другими СУБД – рисунок 2.12.

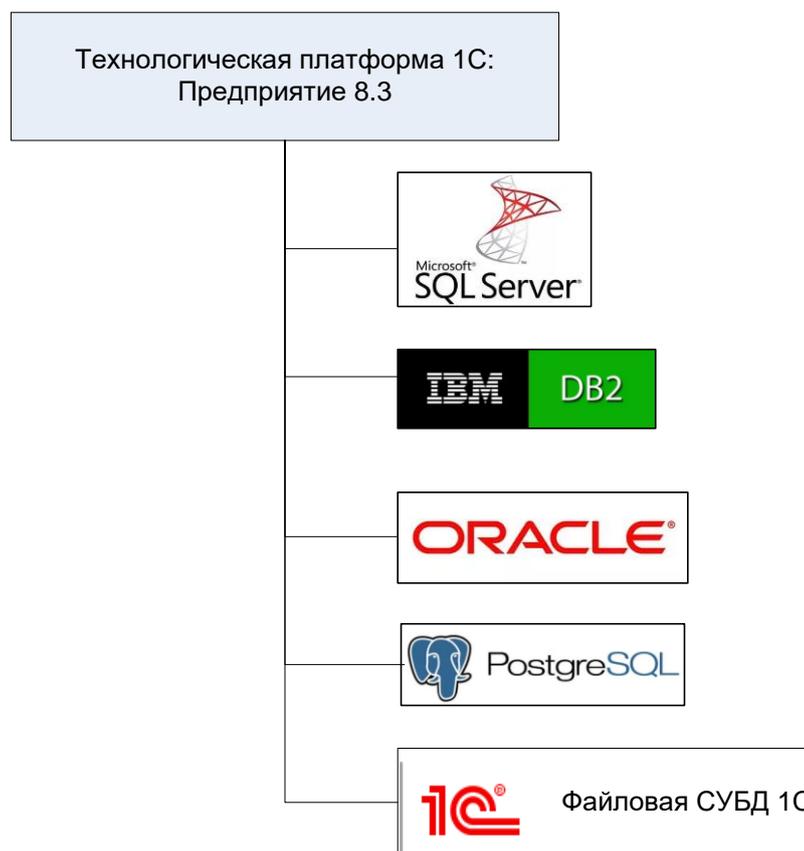


Рисунок 2.14 — СУБД, поддерживаемые платформой «1С:Предприятие 8.3».

Отметим, что разработанное прикладное программное решение не имеет жесткой зависимости от конкретной системы управления базами данных и может быть в полной мере адаптировано для работы с любой другой реляционной базой данных, поддерживаемой используемой программной платформой.

Более того, предусмотрена возможность переноса целостной информационной базы между различными системами управления базами данных для обеспечения гибкости и независимости от выбранной СУБД. Как

видно на схематическом изображении рисунка 2.13, созданное программное обеспечение взаимодействует напрямую с объектами предметной области, получая все необходимые сведения из главного хранилища и записывая результаты обработки операций назад в данные СУБД.

При этом конкретная реализация доступа к данным осуществляется на уровне выбранной базы данных, которая обеспечивает соблюдение всех требований стандарта SQL в отношении хранения, поиска, редактирования информации в базе в соответствии со схемой данных.

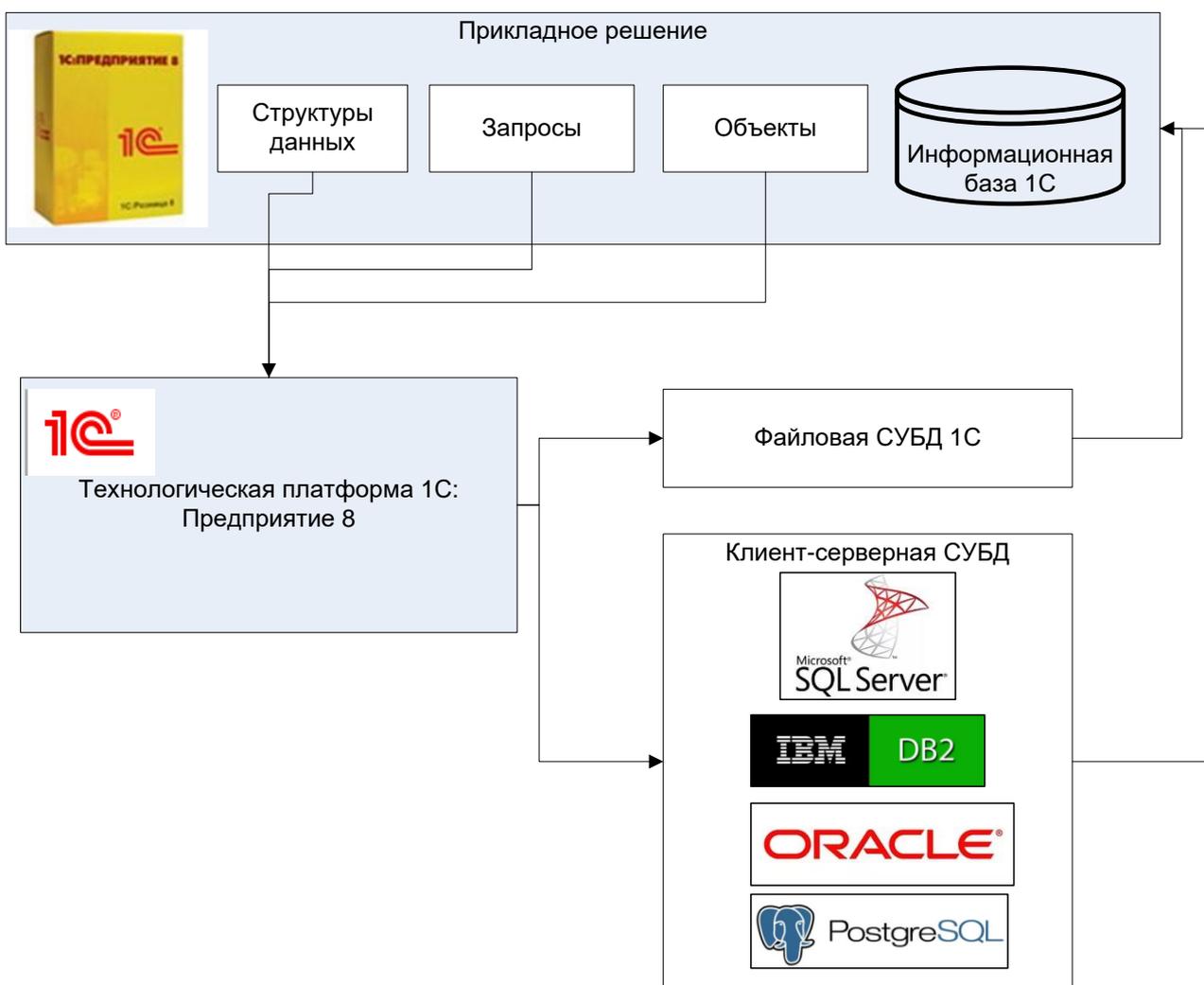


Рисунок 2.15 — Модель работы с базой данных платформы «1С:Предприятие 8».

С уверенностью можно утверждать о том, что при создании нового прикладного решения или в случае доработки имеющегося – не нужно привязываться к характерным чертам конкретной СУБД. Разработка осуществляется на основе платформенных механизмов.

## **2.6 Разработка физической модели базы данных.**

Далее нужно описать структуру прикладных объектов внутри конфигурации создаваемой АИС – в форме ER-диаграммы [14]. На рисунке 2.14 представлена структура справочников подгруппы «Номенклатура».

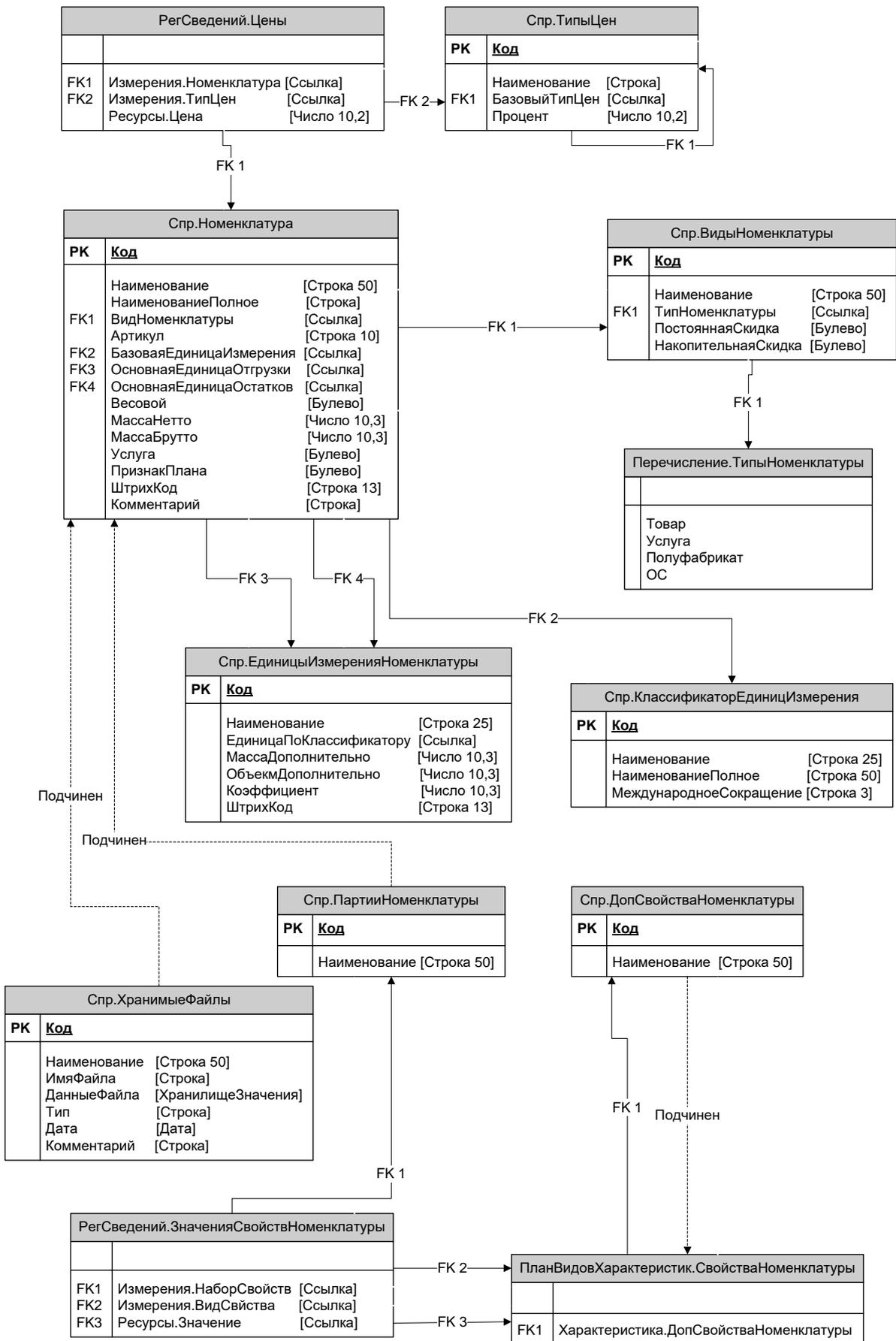


Рисунок 2.16 — Структура справочников подгруппы «Номенклатура».

Структура справочников подгруппы «Контрагенты» приведена на Рисунок 2.17.

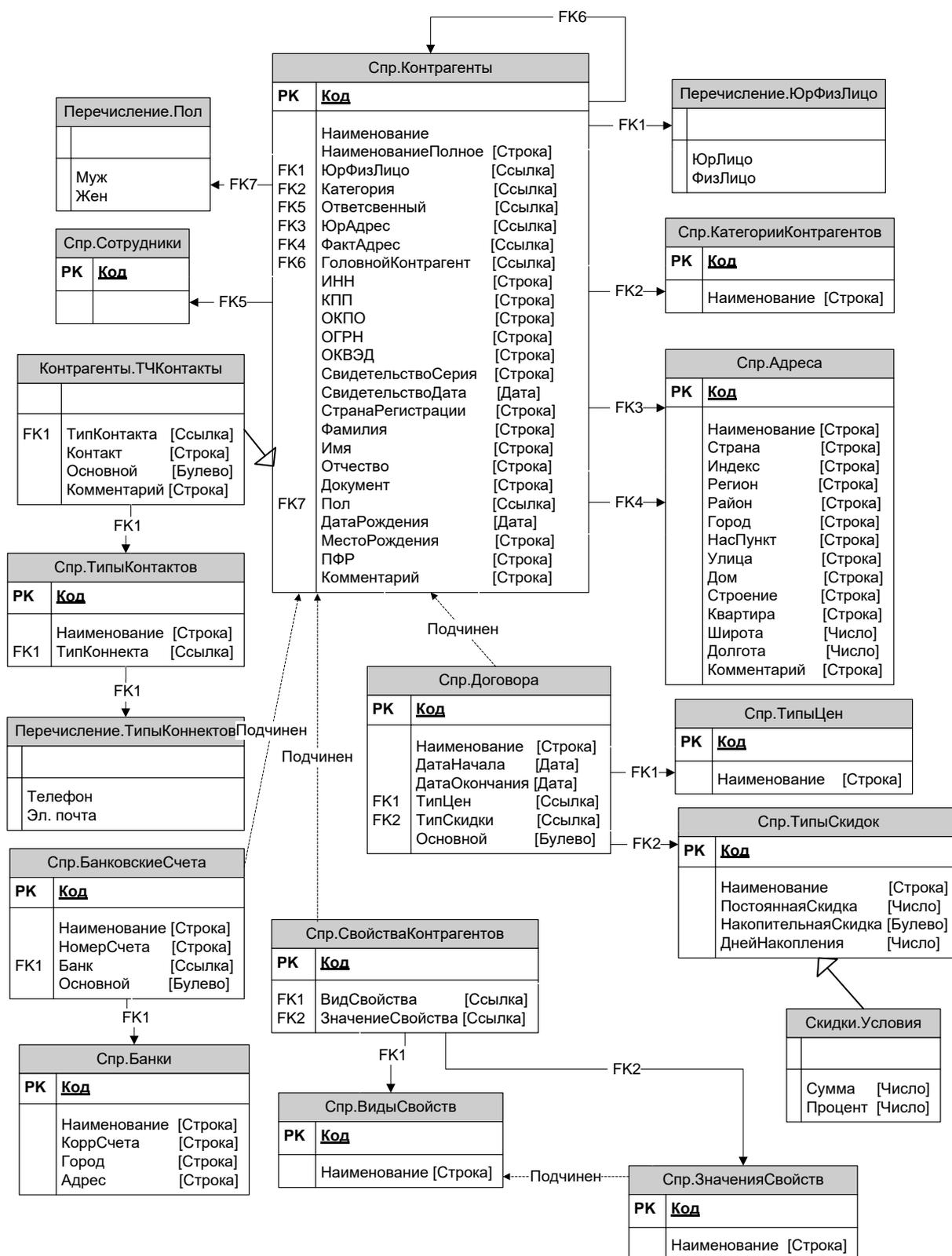


Рисунок 2.17 — Структура справочников подгруппы «Контрагенты».

Структура справочников, описывающих собственную организацию, приведена на Рисунок 2.18.

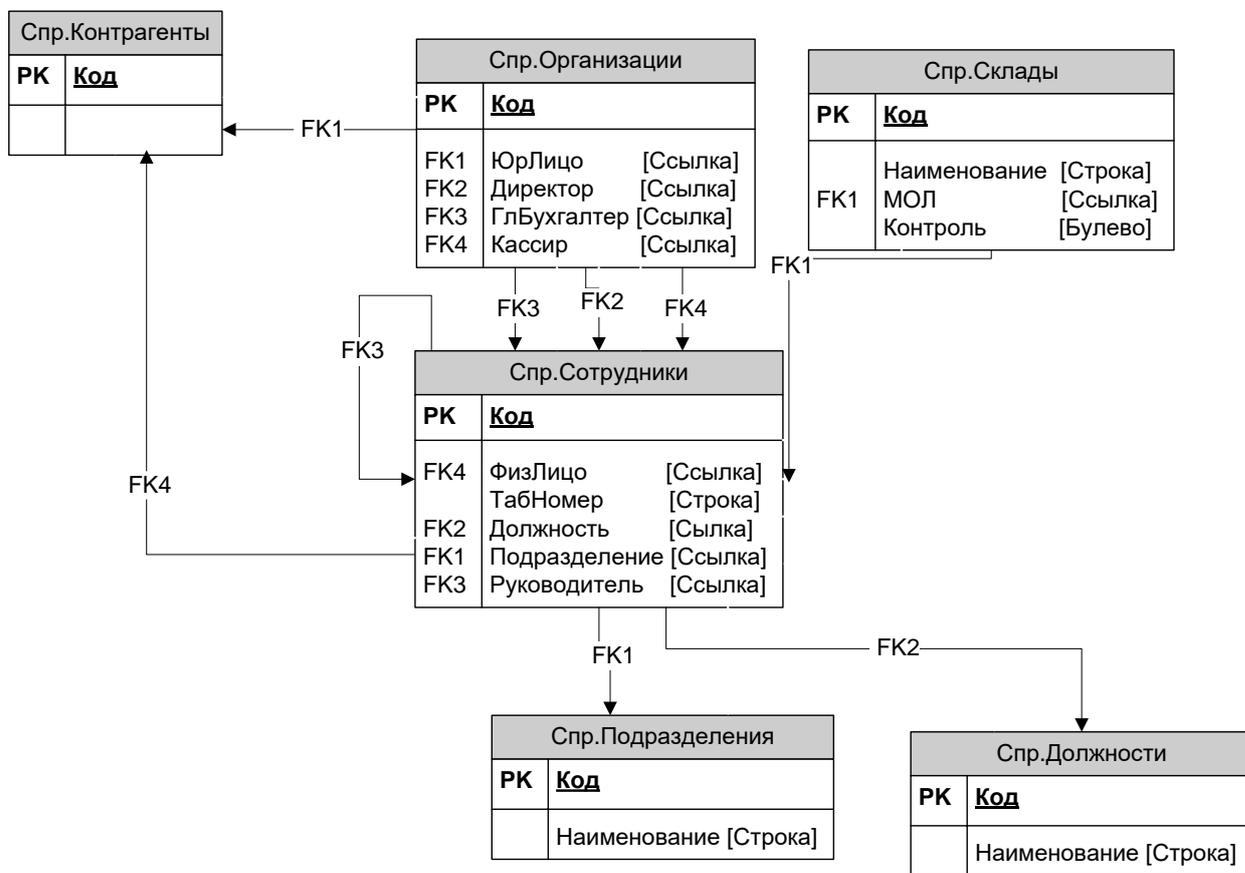


Рисунок 2.18 — Структура справочников подгруппы «Организация».

Структура документов торгового учета приведена на Рисунок 2.19.

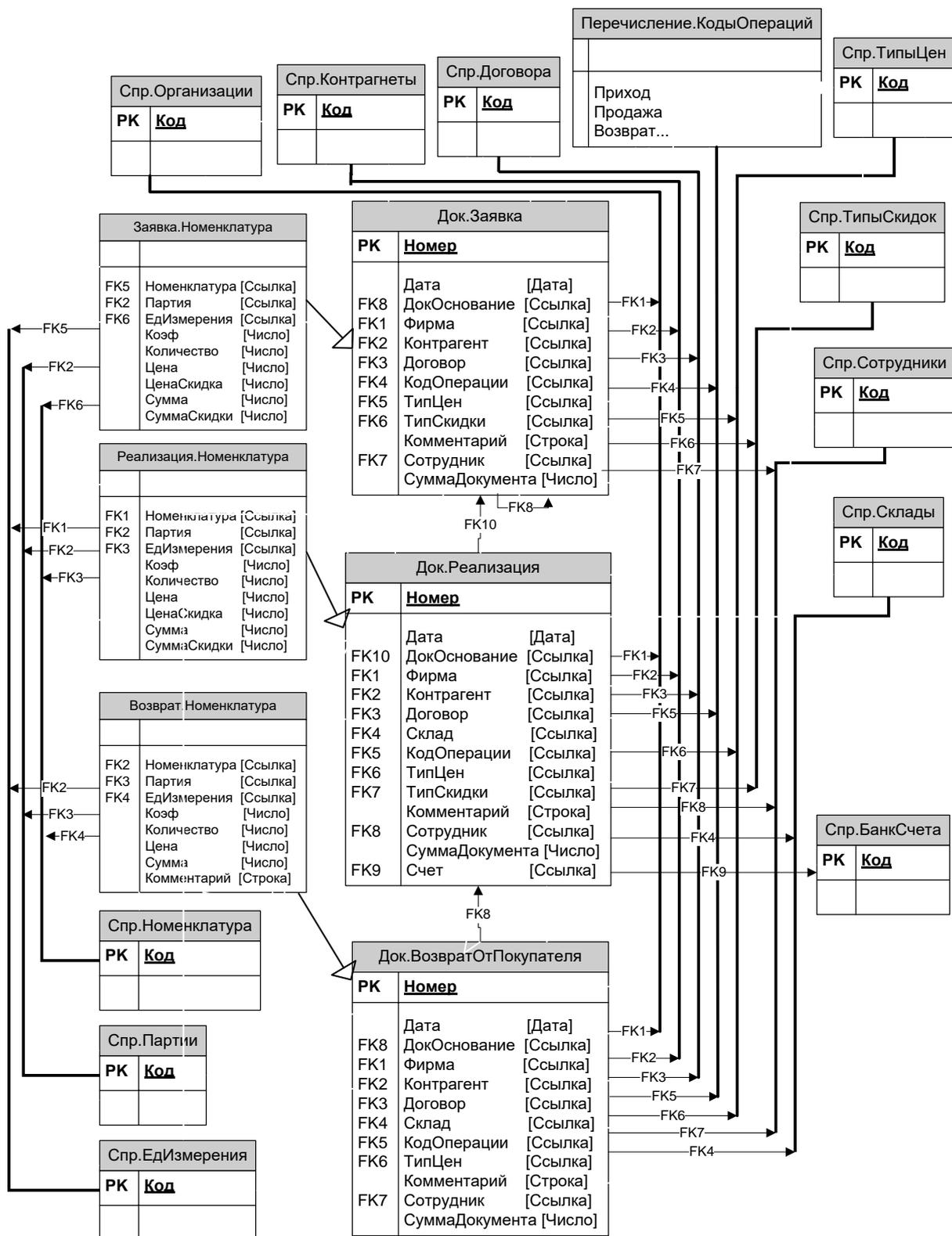


Рисунок 2.19 — Структура документов торгового учета.

Структура регистров накопления приведена на Рисунок 2.20.

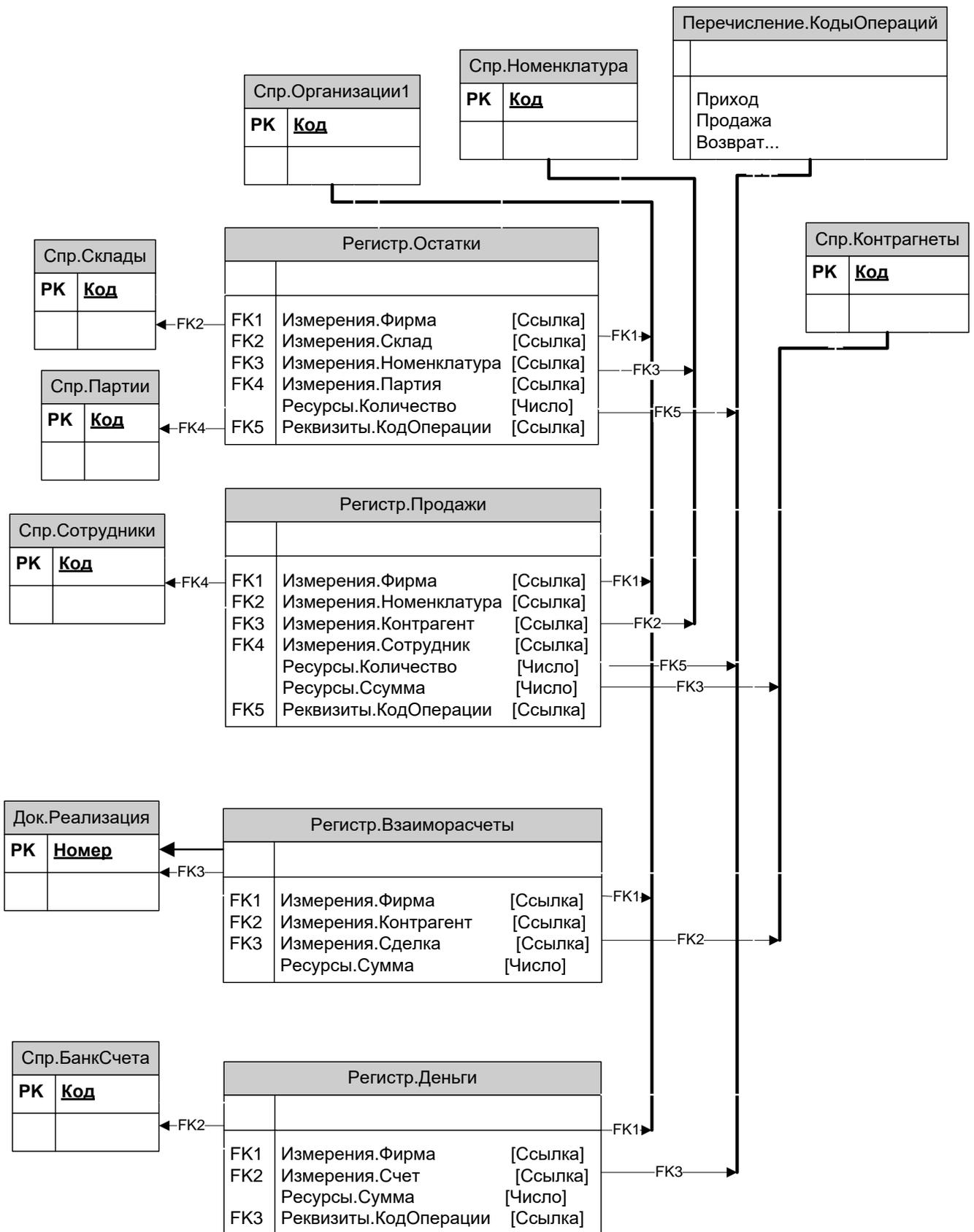


Рисунок 2.20 — Структура регистров накопления.

## **2.7 Выводы по главе 2.**

Обоснован выбор платформы для разработки "1С: Предприятие 8.3" за счет наличия наличие мощных и гибких инструментов и обеспечения максимальной совместимости. Спроектированы и описаны функциональные подсистемы разрабатываемой АИС. Описана архитектура разрабатываемой системы. Проведена разработка модели данных. Касаясь выбора СУБД - нет жесткой привязки к конкретной системе, есть вариант адаптирования к любой другой СУБД. С учетом этого описана и разработана физическая модель базы данных, с подробным описанием структур.

### **3 Разработка автоматизированной системы.**

#### **3.1 Общая концепция разрабатываемой системы.**

Новый релиз программной платформы 1С: Предприятие 8.3 внедряет возможность создания так называемых управляемых приложений, что кардинальным образом затрагивает особенности взаимодействия с пользователем через графический интерфейс. В частности, стоит отметить инновационную модель командного интерфейса, переработанную архитектуру системы рабочих экранов и оконных форм, а также новаторскую для платформы модель распределения функциональных обязанностей между вычислительными компонентами серверной и клиентской частей приложения [5].

Рисунок 3.1 демонстрирует конкретный реальный пример функционирования такого управляемого приложения в одном из возможных программно-конфигурационных режимов. Это позволяет в наглядном виде продемонстрировать существенные особенности взаимодействия разрабатываемой автоматизированной системы с её конечным пользователем на практике. Таким образом вводятся фундаментальные новшества в подходах к организации работы с данными и бизнес-процессами в рамках данной технологической платформы:

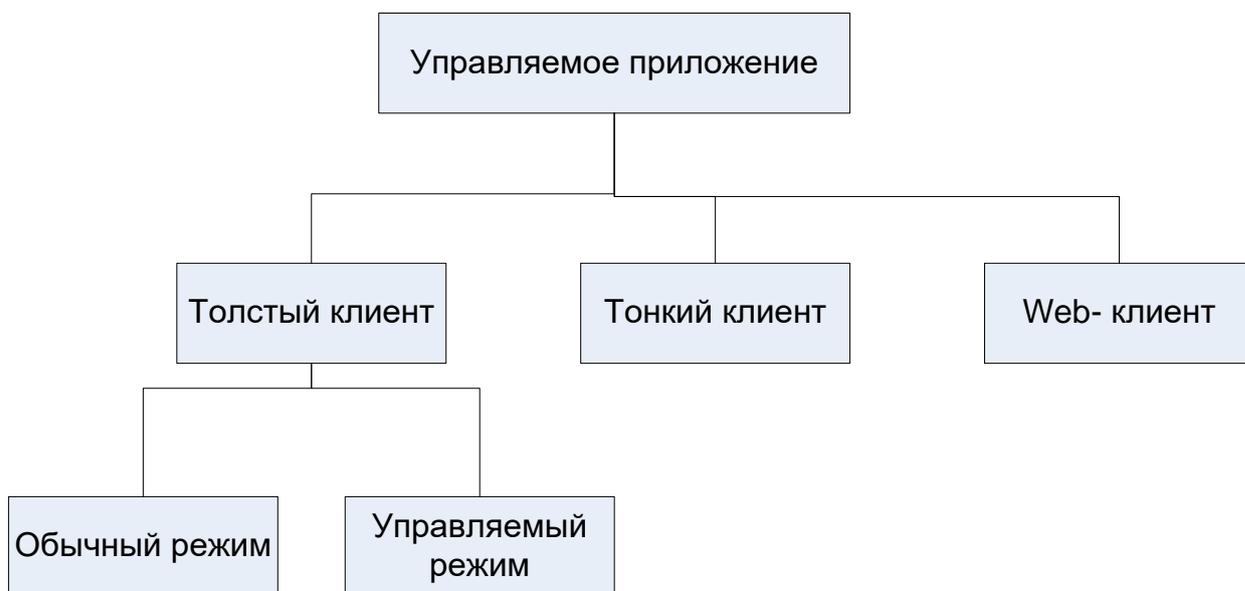


Рисунок 3.1 — Поддерживаемые режимы клиентов платформы «1С:Предприятие 8.3».

Следует подчеркнуть и акцентировать внимание на том, что процесс разработки современного управляемого приложения существенно и принципиально отличается от процесса создания обычного классического клиент-серверного приложения на базе технологической платформы "1С: Предприятие". В первую очередь, это касается и затрагивает изменения в подходе к проектированию и разработке пользовательского интерфейса, в частности командного интерфейса, который теперь характеризуется декларативным построением вместо традиционного императивного программирования. Кроме того, существенным образом изменяется и парадигма работы с формами пользовательского интерфейса: теперь интерфейс только декларируется и описывается разработчиком, а его визуализация и отображение осуществляется самой платформой "1С" в момент вызова и открытия соответствующей формы [11].

Однако подобные значительные изменения и преобразования затрагивают лишь поверхностный видимый уровень, связанный с представлением и визуализацией пользовательского интерфейса. На более глубоком фундаментальном уровне происходят принципиальные архитектурные

преобразования и трансформации. Режим работы "Тонкий клиент", реализуемый исключительно в рамках управляемого приложения, представляет собой классическую клиент-серверную модель с трехзвенной распределенной архитектурой. Веб-клиент и тонкий клиент обеспечивают возможность удаленного запуска и развертывания приложения в облачной среде, а также доступ к нему через веб-браузер. Большинство типовых конфигураций от компании "1С" уже переведены на работу в современной парадигме управляемых приложений [2].

При создании и разработке управляемого приложения программист должен четко осознавать и понимать, где именно на каком уровне архитектуры будет исполняться код конкретной процедуры, функции или методики: на клиентском уровне или на серверном уровне, а также какие именно данные из информационной базы будут доступны в процессе исполнения и обработки этого кода [15].

Как правило, клиентское приложение максимально облегчается и оптимизируется, и не может напрямую самостоятельно выполнять какие-либо запросы или обращения к базе данных, что позволяет существенно снизить нагрузку на сетевые ресурсы и каналы связи. В крайнем случае, клиентское приложение может быть реализовано в упрощенном виде как обычный веб-браузер, отображающий только пользовательский интерфейс, в то время как вся сложная ресурсоемкая обработка и манипуляция с данными осуществляется централизованно на уровне сервера [16].

Однако подобная архитектурная оптимизация и распределение нагрузки сопряжены с определенными сложностями и проблемами для разработчиков. Многие привычные для традиционного структурного или объектно-ориентированного программирования операции и подходы в рамках управляемого приложения выполняются совершенно иначе, что создает ряд трудностей для программистов на начальном этапе освоения новой парадигмы программирования [2].

Тем не менее, если абстрагироваться от специфических особенностей и нюансов разработки управляемых приложений, то в целом общий концептуальный подход к проектированию и программированию остается прежним: технологическая платформа "1С" по-прежнему представляет собой комплекс взаимосвязанных прикладных объектов, каждый из которых максимально приближен и соответствует определенной конкретной предметной области [17].

На рисунке 3.2 показана общая структура конфигурации.

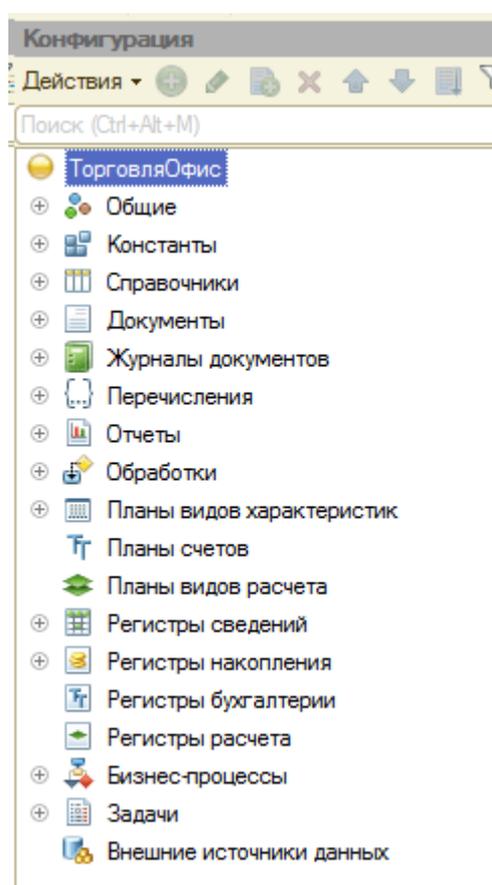


Рисунок 3.2 — Общая структура конфигурации.

На следующем ключевом этапе процесса разработки осуществляется детальное проектирование и программирование, так называемой, "базовой" части конфигурации, включающей в себя целый комплекс различных прикладных объектов технологической платформы "1С: Предприятие 8.3".

Автоматизированная информационная система, которая создается и разрабатывается в рамках данной выпускной квалификационной работы, по своей сути, представляет собой специализированное прикладное решение для технологической платформы "1С", состоящее из совокупности многочисленных узкоспециализированных платформенных объектов самой конфигурации. На наглядном рисунке 3.3 представлены и визуализированы различные основные типы прикладных объектов, которые были задействованы и использованы для реализации поставленной проектной задачи по автоматизации. В совокупности эти многочисленные объекты образуют единую целостную конфигурацию итогового прикладного решения.

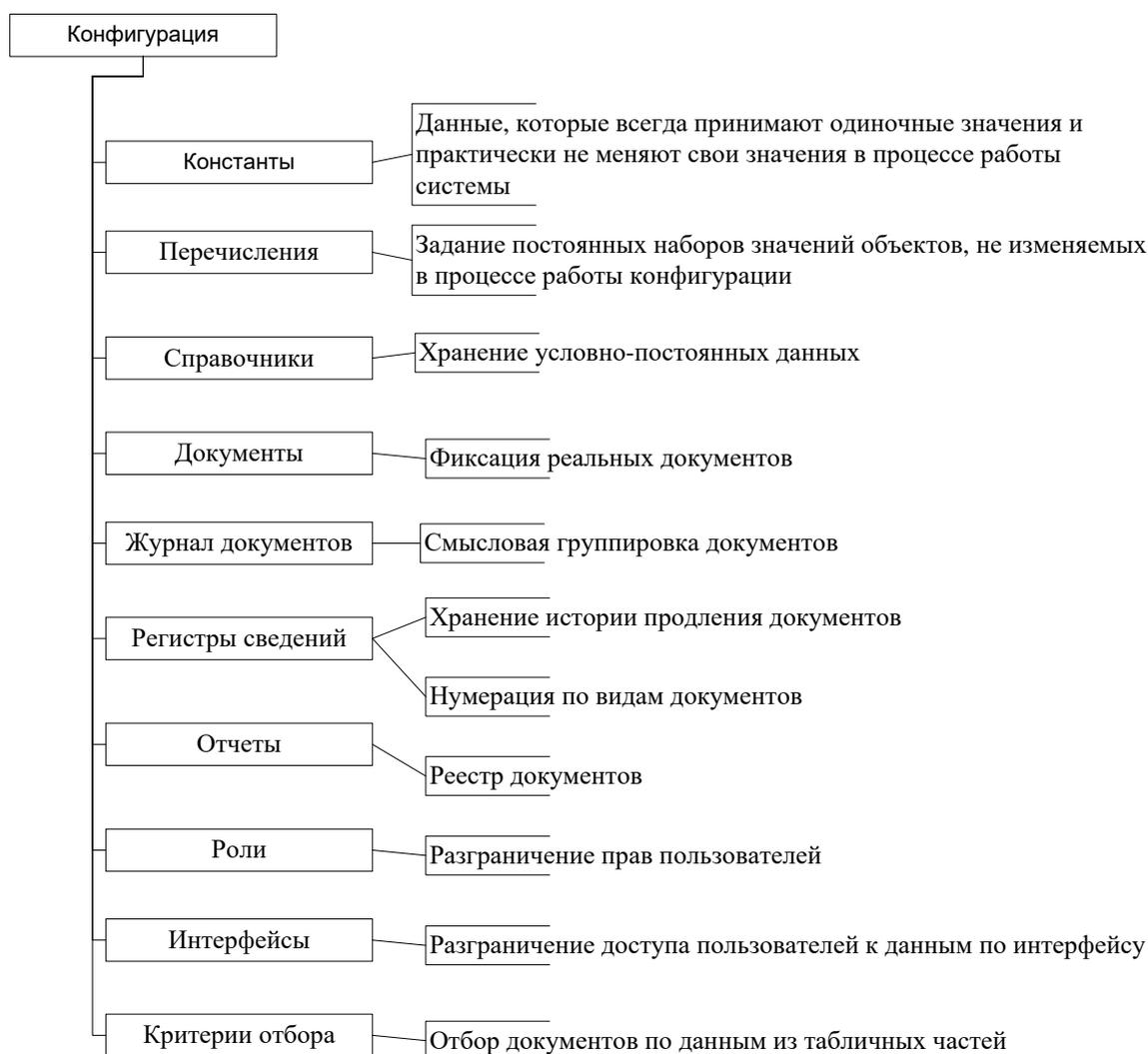


Рисунок 3.3 — Структурная схема проекта.

### **3.2 Описание объектов системы.**

В информационной модели автоматизированной системы управления торговыми операциями, одной из ключевых задач которой является эффективное ведение учёта поступления, реализации и остатков широкого круга товарных позиций, логично выделить подсистему, отвечающую за сбор и хранение сводной справочной информации о каждом товарном наименовании.

В частности, фундаментальное значение имеет структурированный номенклатурный справочник - сложный информационный модуль, реализованный в виде отдельной подсистемы программного комплекса и предназначенный для централизованного хранения, поиска и редактирования всесторонней совокупности сведений, характеризующих любой товар с точки зрения его технических, экономических и организационных параметров, а также данных, необходимых для дальнейшего ведения учёта всех операций с этим товаром.

В его состав заносятся сведения об ассортиментной принадлежности товара, первичных технико-экономических характеристиках, единицах измерения, ценовой политике и многом другом.

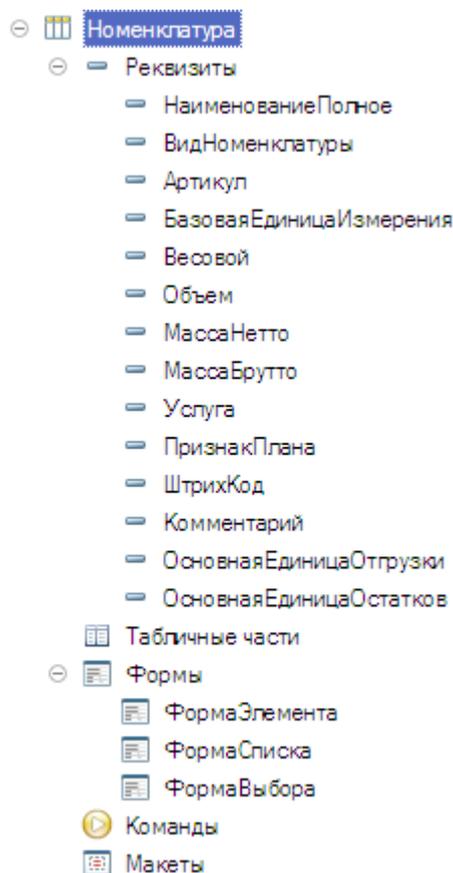


Рисунок 3.4 — Справочник «Номенклатура».

Одной из наиболее фундаментальных особенностей проектируемой масштабируемой информационной системы предпринимательской деятельности является её способность на базе структурированных данных обеспечивать тщательный учёт поступления, реализации и остатков материальных активов с учётом их группировки в гомогенные складские партии.

Понятие «партия» в рамках данной концепции означает уникальный объект, описываемый совокупностью идентифицирующих признаков, включая наименование и набор характеристик, каждое значение которых носит строго индивидуальный характер, определяя своеобразие данной партии.

Детальные взаимосвязи ключевых составляющих информационной модели системы, позволяющие осуществлять учётные процессы на уровне отдельных партий товаров и формировать комплексную отчётность о наличных



Проведя детальный анализ фундаментальных справочных модулей "Единицы измерения номенклатуры" и "Классификатор единиц измерений", ответственных за формализацию знаний об отношениях между различными измерительными единицами, можно сделать вывод о возможности ведения учёта наличия и движения зарегистрированных наименований товаров и материалов с указанием конкретной единицы измерения для каждого наименования.

При этом необходимо назначить одну основную единицу для каждого товара, относительно которой будут приводиться все остальные альтернативные единицы.

Детальная логическая структура упомянутых справочников, подробно продемонстрированная на схематическом изображении рисунка 3.6, отражает наличие в них всей ключевой информации об эквивалентности единиц измерения и их свойствах, необходимой для функционирования сложного механизма учёта движения товаров с учётом разнообразия единиц их количественной характеристики.

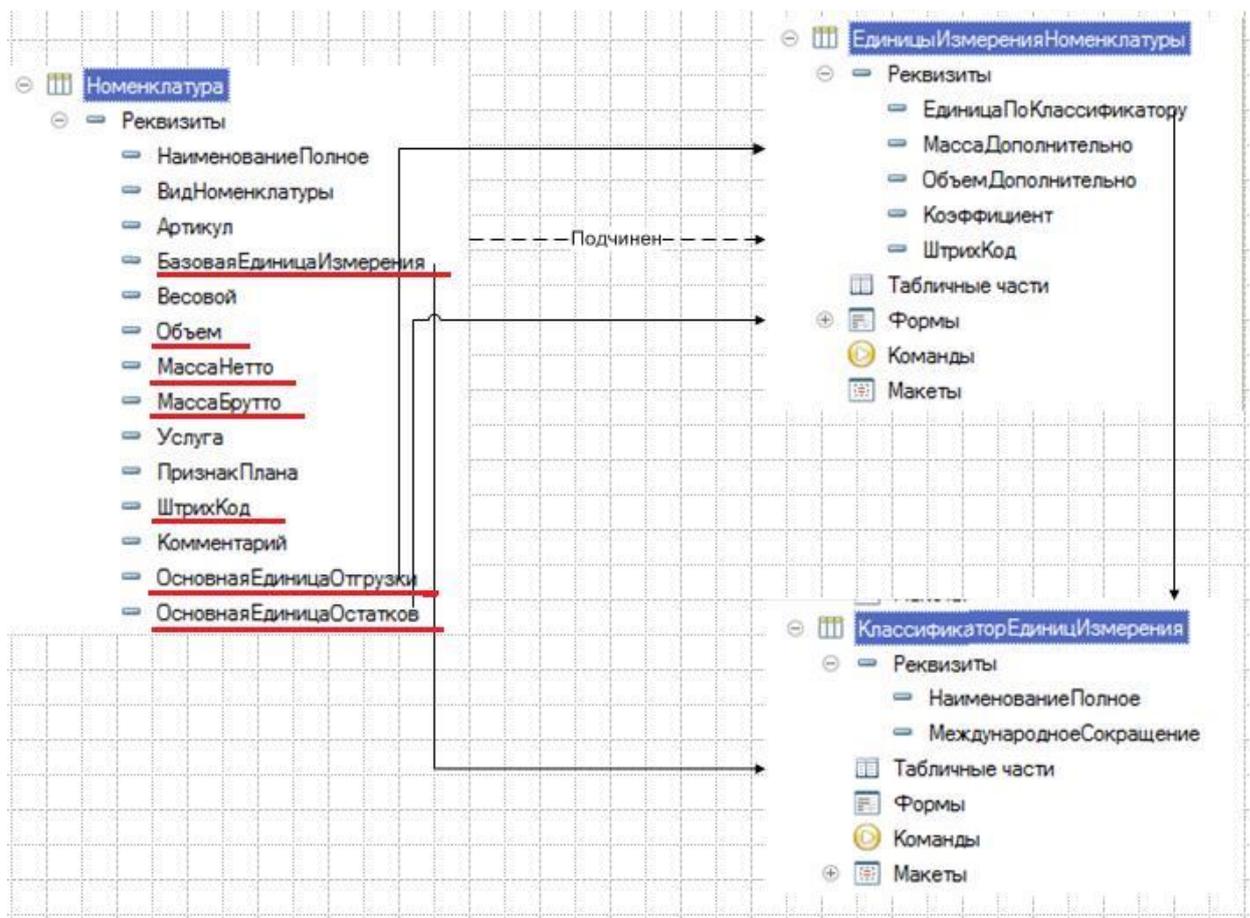


Рисунок 3.6 — Прикладные объекты, отвечающие за ведение единиц измерения номенклатуры.

В ходе разработки логической структуры масштабируемой информационной системы управления коммерческой деятельностью предприятия одним из важнейших аспектов является организация процесса учета и динамического корректирования показателей стоимости реализуемой продукции и оказываемых услуг.

Для решения данной задачи, как правило, используется справочно-аналитический модуль «Цены», представляющий собой периодически обновляемую базу данных, содержащую информацию о значениях коммерческой стоимости каждого наименования в разрезе применимых ценообразующих коэффициентов. При этом расчёт цены производится относительно основной единицы измерения наименования.

Взаимодействие модуля «Цены» с другими объектами информационной модели, обеспечивающими реализацию описанного механизма, детально представлено на структурной схеме рисунка 3.7.

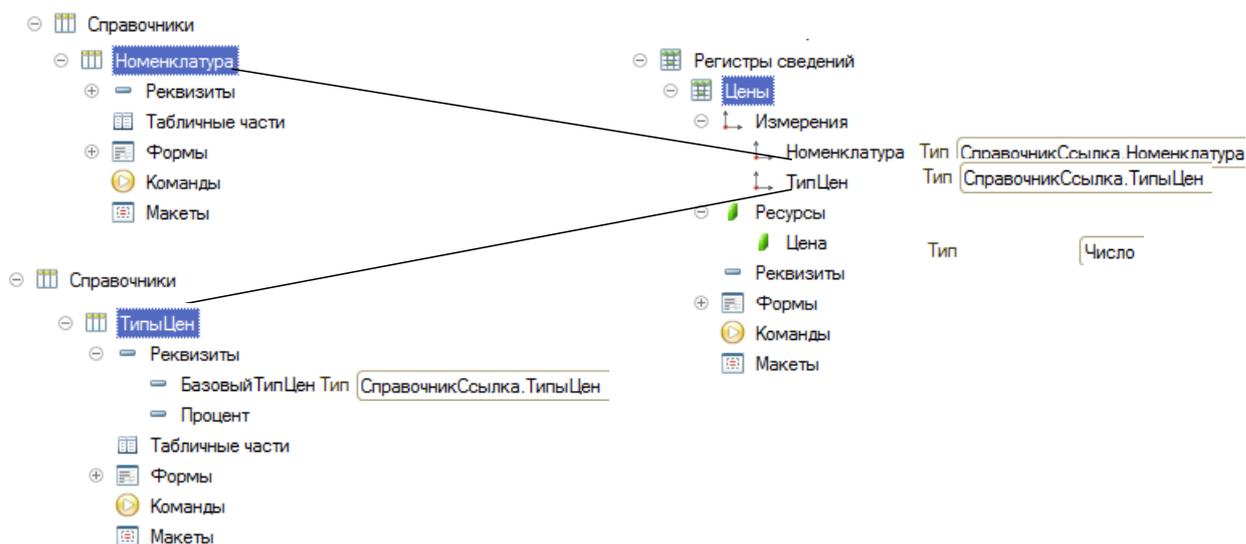


Рисунок 3.7 — Реализация механизма учета цен.

В рамках проектируемой масштабируемой информационной системы управления коммерческими операциями предприятия предусмотрено функциональное решение, позволяющее обеспечить учет и управление данными, содержащимися во внешних информационных источниках.

Для этого спроектирован справочный модуль "Внешние данные", имплементированный в виде структурированной реляционной БД и содержащий атрибут данных с типом "Хранилище файлов".

По данному атрибуту размещаются сами файлы, которые при загрузке с внешнего носителя помещаются в ИБС, где могут быть открыты ассоциированными приложениями в соответствии с форматом.

Взаимодействие с такими файлами в процессе загрузки, изменения и просмотра осуществляется через интерфейс самого справочника с использованием встроенных команд.

Детальная логическая схема взаимосвязи объектов данной системы показана на рисунке 3.8.

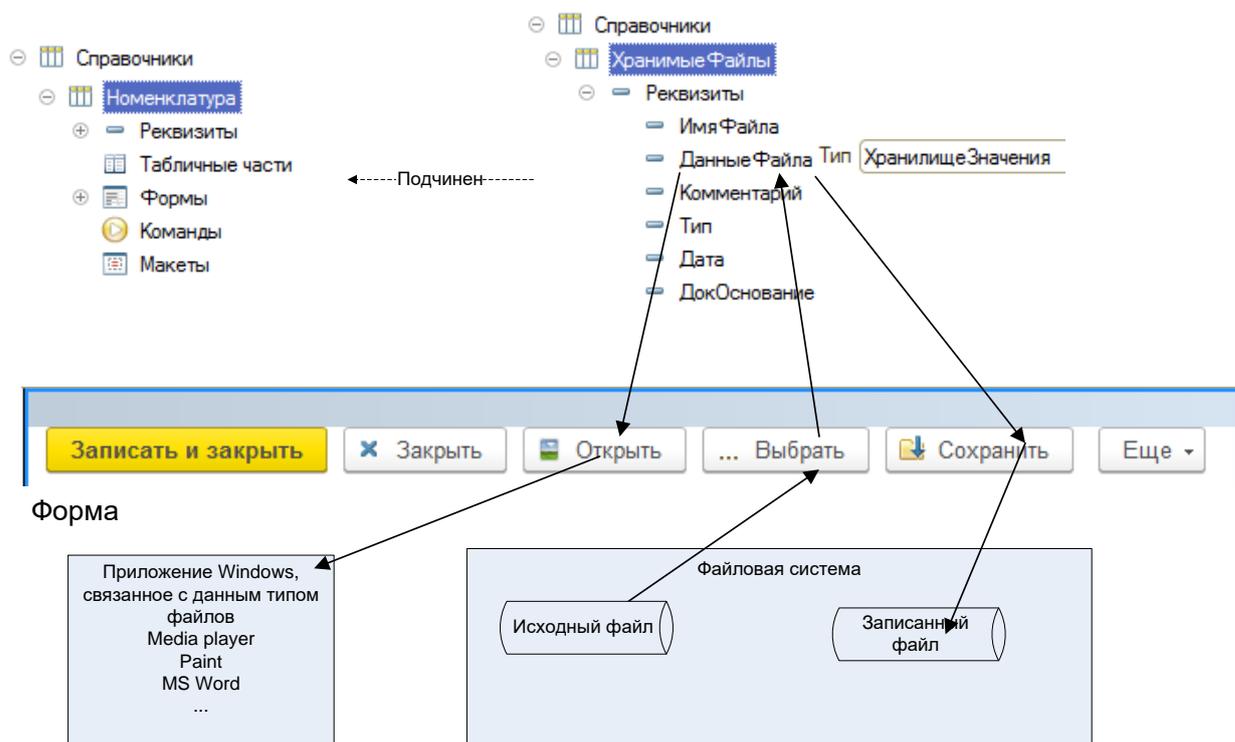


Рисунок 3.8 — Реализация механизма связанных файлов.

В ООО "Продторг" функция постоянного контекстно-зависимого анализа обширной базы данных организаций и частных лиц, осуществляющих взаимодействие с компанией в рамках коммерческой деятельности, является одним из центральных процессов масштабируемой информационной экосистемы.

Для реализации указанной функции задействовано специализированное хранилище информации "Контрагенты", включающее широкий спектр структурированных данных о юридических и физических лицах, индивидуальных предпринимателях, а также сотрудниках самой организации.

Следует отметить, что объём атрибутивных характеристик контрагентов разработан с целью максимально полного описания контекста взаимоотношений.

Вместе с тем, масштаб данных требует специальных интерфейсов для удобного ознакомления, в связи с чем разработан компонент "Профиль контрагента" с функцией страничного представления информации.

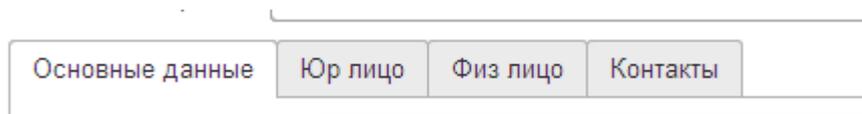


Рисунок 3.9 — Закладки формы элемента справочника «Контрагенты».

В процессе проектирования высокопроизводительной масштабируемой информационной системы управления коммерческой деятельностью предусмотрена гибкая концепция адаптивного хранения объёмных атрибутивных данных о внешних контрагентах, параметры которых могут изменяться с течением времени под воздействием множества внутренних и внешних факторов.

Такая концепция обусловлена трудностями прогнозирования полной структуры данных на этапе проектирования с учётом возможного появления новых характеристик объектов учёта в процессе эксплуатации системы.

Для реализации масштабируемого подхода используются специализированные справочные БД переменной структуры с возможностью динамического расширения набора полей без модификации исходного кода.

Структурный интерфейс взаимосвязи ключевых компонентов модели, отражённый на диаграмме рисунка 3.10, позволяет эффективно решать задачи хранения, обновления и анализа дополнительных свойств контрагентов.

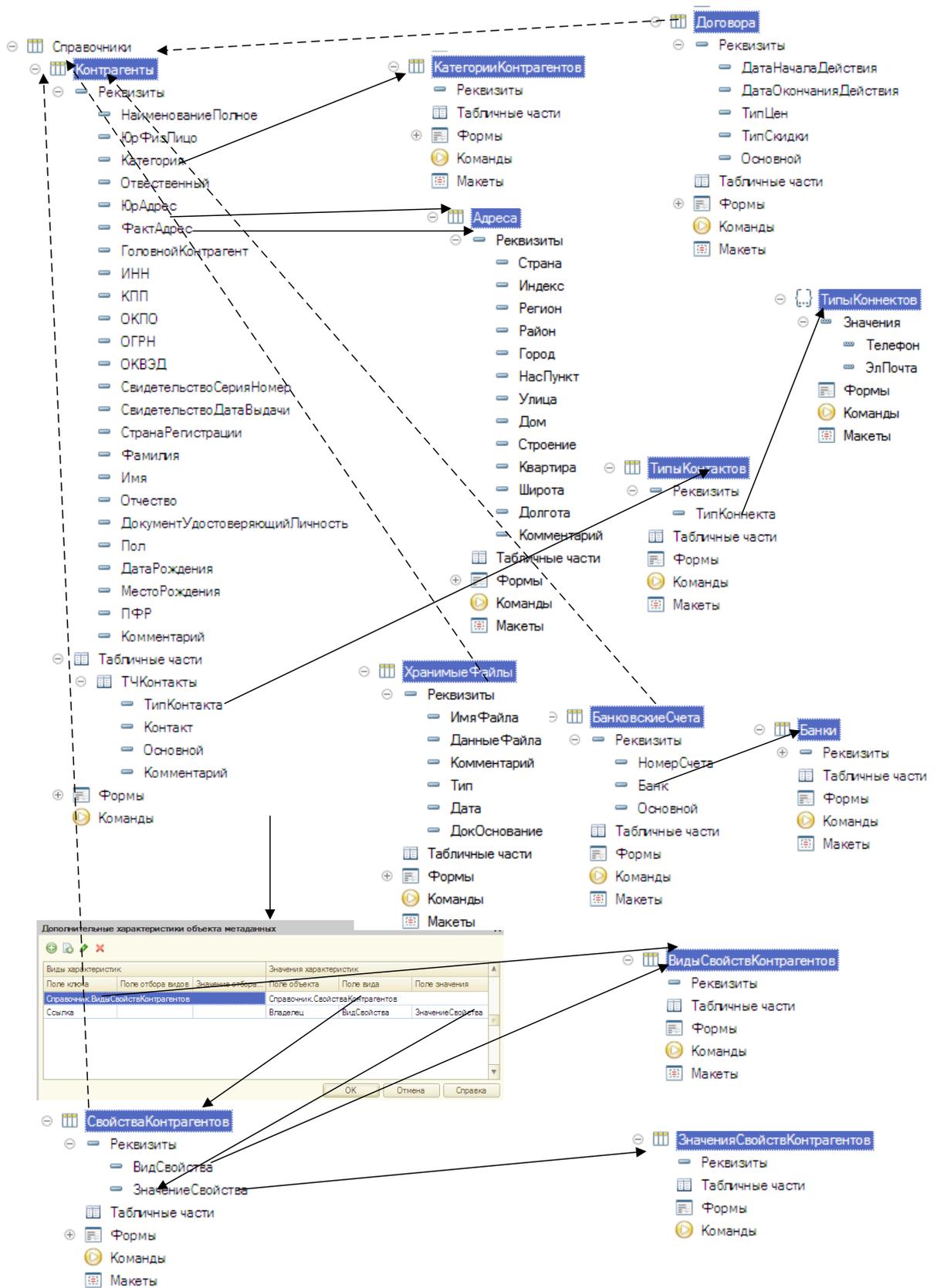


Рисунок 3.10 — Структура данных, позволяющая хранить свойства контрагентов.

В проектируемой интегрированной информационной системе предполагается разработка масштабируемого модуля, обеспечивающего систематизированное сопоставление базовых структурированных данных о контрагентах и описываемой ими номенклатуре с дополнительными сведениями, расположенными во внешних информационных источниках различного типа.

Для реализации указанной задачи предусматривается создание справочной единицы "Внешние ссылочные данные", выполняющей функции посредника между внутренней информационной базой и внешними носителями посредством хранения метаданных, связывающих первичные сущности "Номенклатура" и "Контрагенты" со вспомогательной информацией.

Алгоритм работы данного модуля будет аналогичным механизму интеграции с внешними файлами номенклатурных характеристик, обеспечивая возможность консолидации данных из разнородных источников посредством единой точки сопоставления — справочника "Внешние ссылочные данные".

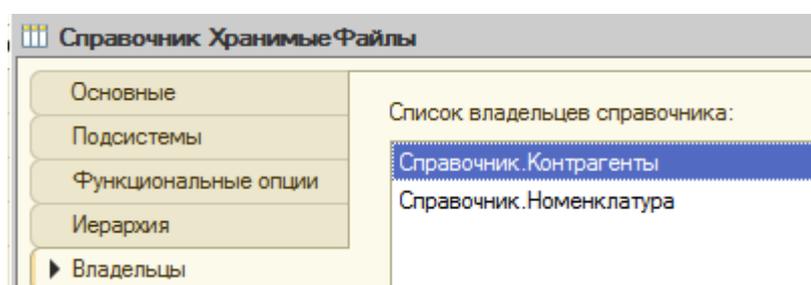


Рисунок 3.11 — Подчиненность справочника «ХранимыеФайлы».

В рамках функционального блока "Реализации" предусмотрен комплексный подсистемный модуль, обеспечивающий расчёт интегральных стоимостных показателей по единицам реализуемой товарно-материальной продукции и продукции услуг с учётом широкого спектра ценообразующих факторов, включающих оба основных подхода — фиксированную и накопительную методики начисления торговых сбавок, налогов и сборов в зависимости от объёмов, динамики и истории сотрудничества.

Структурная схема взаимодействия ключевых компонентов информационной модели, реализующих указанный расчётный функционал в рамках делового документа "Реализация товаров/услуг", подробно представлена на архитектурном рисунке 3.12.

В частности, проектируются модули работы со справочными базами ценообразующей информации, атрибутами контрагентов и оснований начисления, а также компонент, рассчитывающий итоговую стоимость по формулам статического и накопительного типа.

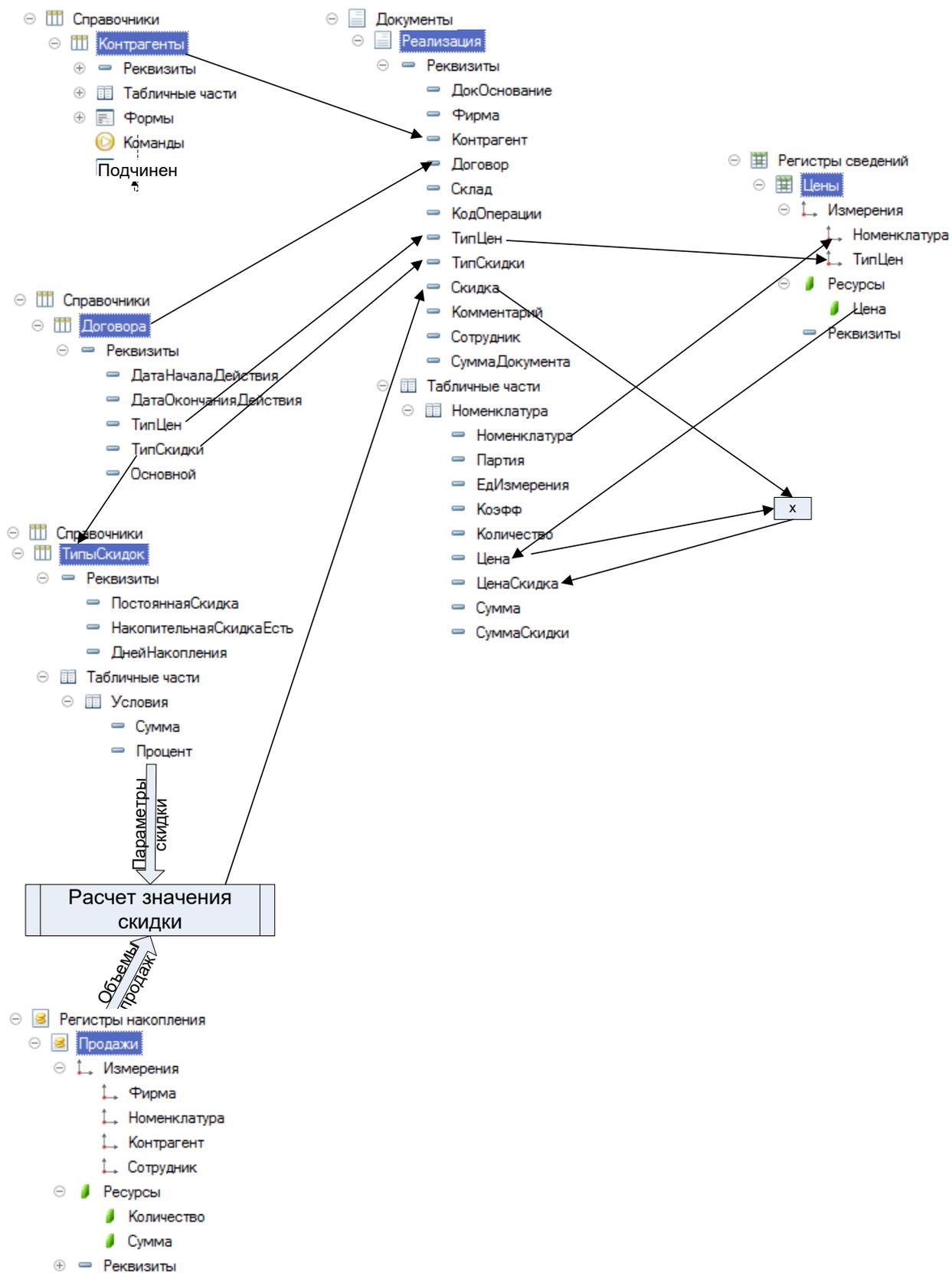


Рисунок 3.12 — Механизм расчета цен.

В новой интегрированной информационной системе управления коммерческой деятельностью предусматривается формализованное

функционально-ориентированное структурирование ключевых операционных процессов посредством специализированной документно-ориентированной модели взаимодействия.

К группе модулей "Продажи" относятся документы: "Реализация товаров и услуг", "Заказ покупателя", "Заказ поставщикам", а также иные. К модулю "Расчеты с контрагентами" - "Денежные поступления", "Денежные расходы" и т.д.

В процессе заполнения документов задействуется встроенный механизм генерации взаимосвязанных объектов на основе друг друга, позволяющий автоматизировать процесс заполнения многих полей и формирования цепочек взаимосвязанных документов.

Структурная схема рисунка 3.13 детально отражает архитектуру взаимодействия ключевых компонентов предлагаемой модели, реализующих указанный функционал.

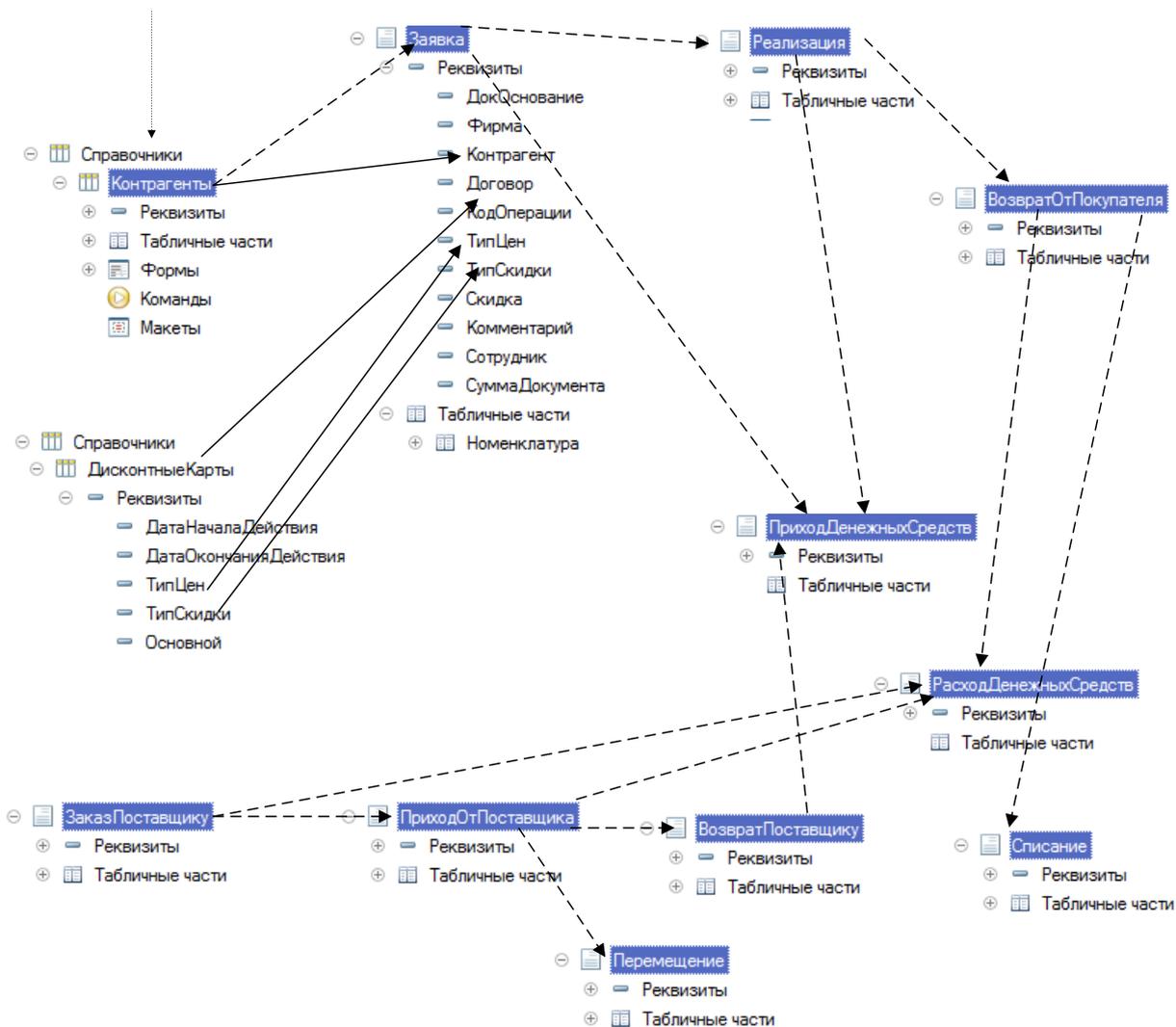


Рисунок 3.13 — Механизм ввода документов на основании.

В процессе разработки предлагаемой интегрированной информационной системы предусматривается использование встроенных механизмов платформы 1С для организации эффективного хранения и накопления данных об операциях, фиксируемых посредством различных документов.

В частности, задействуется объект "Регистры накопления" - специализированная база данных, обеспечивающая учетное отслеживание движения и наличия документов в системе. Указанные регистры позволяют формализовать процессы записи первичной документации и отчетности, предоставляя возможности для последующего анализа и отчётности.

Структура и функциональное назначение основных регистров, реализуемых в рамках разрабатываемой конфигурации в соответствии с общей

методологией платформы 1С, подробно отражены на архитектурной схеме рисунка 3.14.

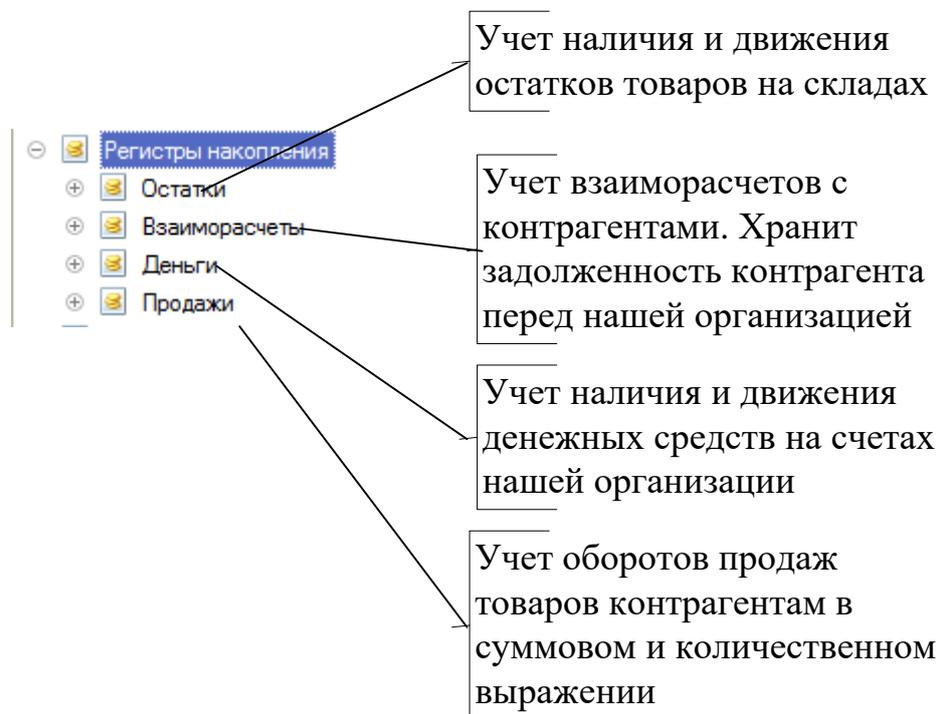


Рисунок 3.14 — Состав регистров конфигурации.

В дальнейшем будет представлена интегральная модель управления документооборотом, формализующая алгоритмы перемещения структурированной информации между специализированными регистрами накопления данных в соответствии с потоками документации.

Регистр "Операции реализации" носит сугубо документальный характер и предназначен для учета фактов заключения сделок купли-продажи товаров и услуг в разрезе типологизированных операций без направленной связи с изменением остатков активов.

Данный регистр обеспечивает накопление информации об оборотах документов, сформированных на основе унифицированной номенклатуры признаков учета, без учета их влияния на финансовые показатели системы.

Структурная схема взаимодействия регистров позволит систематизировать потоки учетной информации для целей анализа и отчетности.

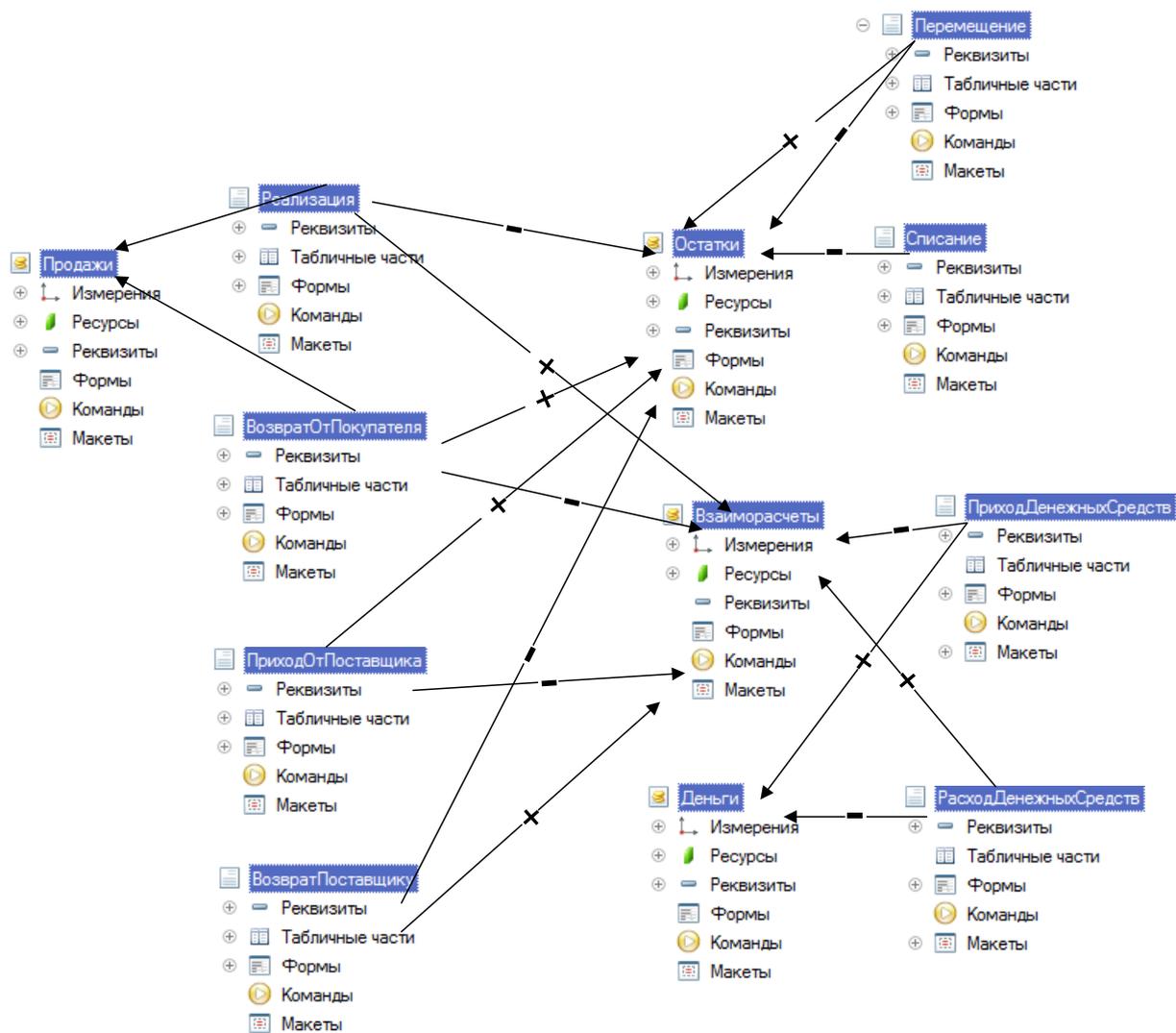


Рисунок 3.15 — Движения документов по регистрам.

На последующем структурном рисунке 3.16 будет детально отражён алгоритм заполнения различных типов сведений - измерений, реквизитов, ресурсов - в ходе проведения разнотипных документов, в частности документа «Реализация товаров/услуг».

Данная модель процесса заполнения документальной информации будет представлена в формализованном виде на уровне взаимосвязи ключевых компонентов предлагаемой информационной системы.

В частности, будут показаны:

- потоки данных между справочниками и документами;
- логика актуализации измерений, реквизитов и остатков ресурсов;
- последовательность выполнения операций при заполнении реквизитов документа.

Данный подход позволит формализовать процесс заполнения первичной учётной информации и обеспечит прозрачность взаимодействия различных элементов предлагаемой системы.

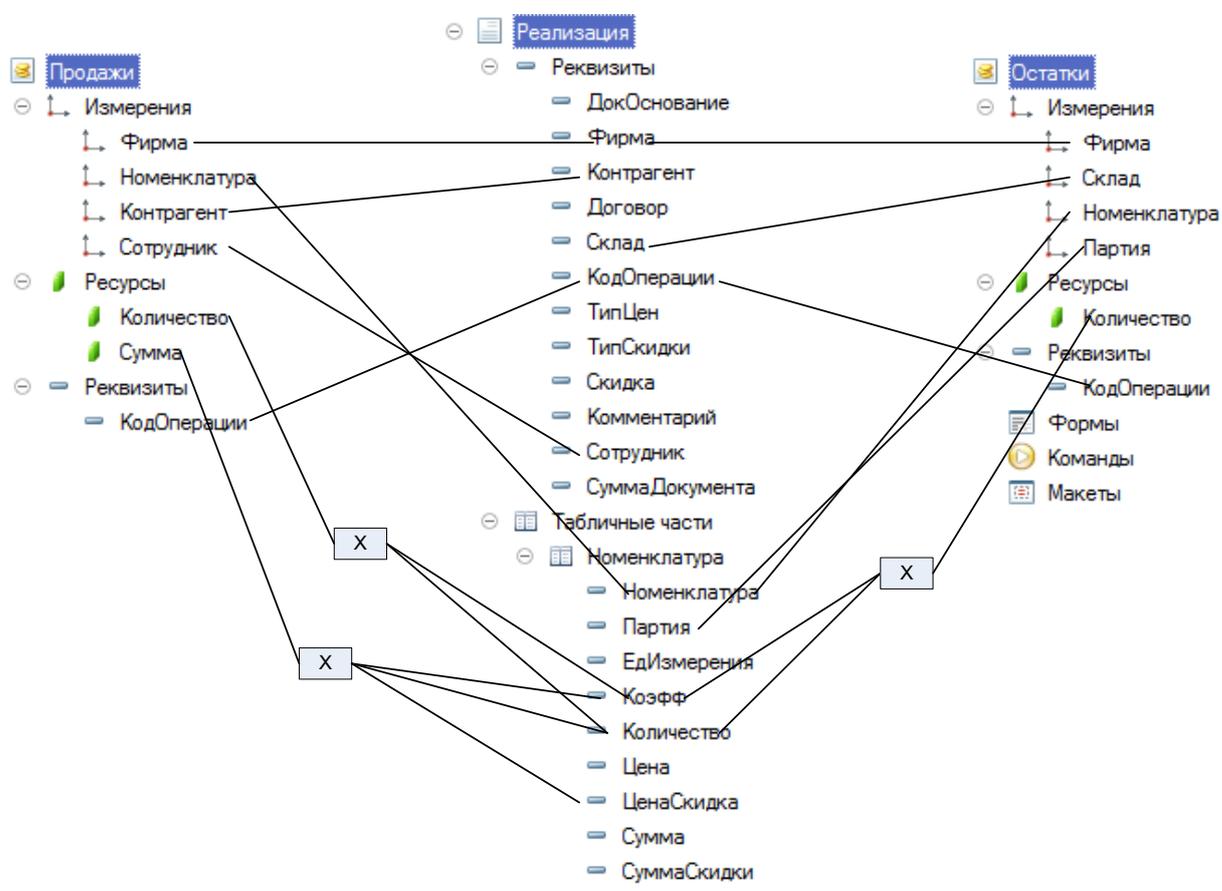


Рисунок 3.16 — Заполнение структуры регистров из документа при проведении.

Ну и наконец скажем о том, что именно регистры являются источниками данных для построения отчетов.

### 3.3 Разработка отчетов и запросов.

Исходной информацией, на базе которой получают отчеты – есть журнал электронных документов. Функциональная и технологическая схема в рамках задачи под названием «Получение отчетов» представлена на рисунке 3.17.

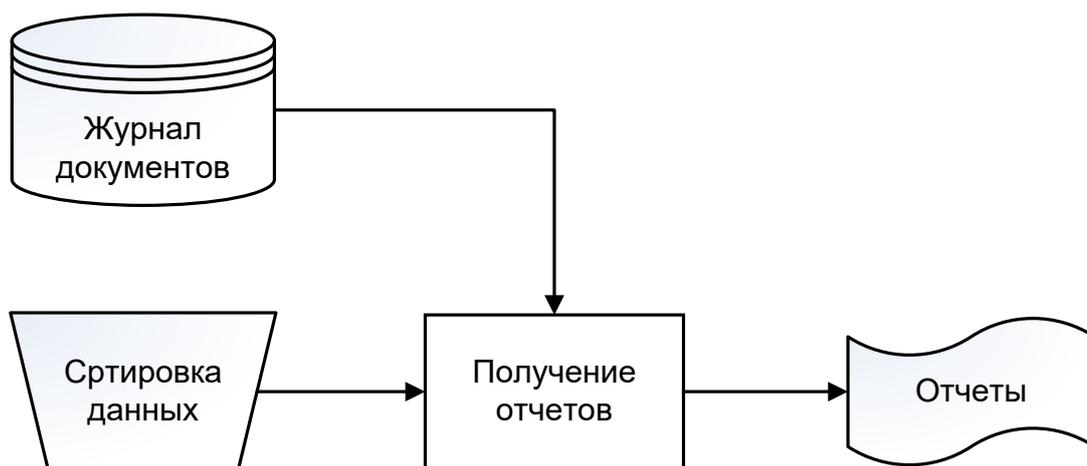


Рисунок 3.17 — Функционально-технологическая схема задачи «Получение отчетов».

В новой АИС отчеты созданы посредством системы компоновки данных. Работа системы компоновки показана на рисунке 3.18.

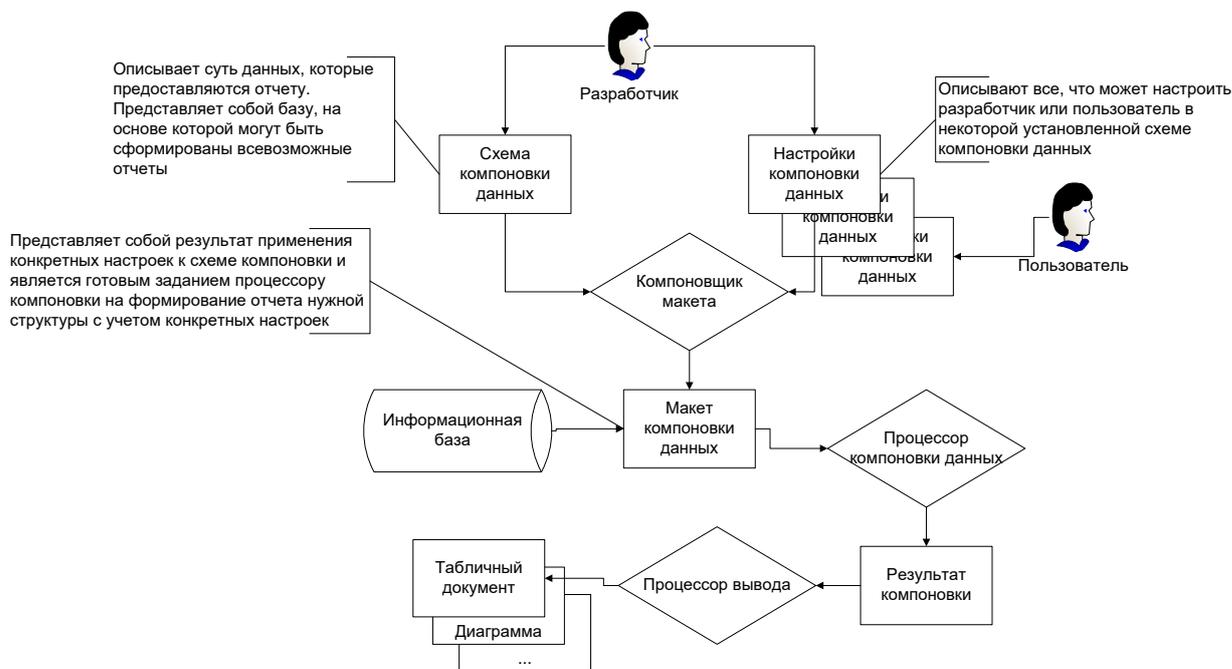


Рисунок 3.18 — Схема работы системы компоновки данных.

Для моделирования совокупного феномена применяются принципы системного анализа. С их помощью в новой обобщенной схеме описываются характеристики данных комплекса. Настраечные параметры предоставляют возможность установить четко фиксируемые рамки конкретной модели отчета, упорядоченную структуру вводимых показателей и логику их последовательной классификации, а также многие другие детерминанты.

В режиме функционирования системы как целостного образования, согласно установленной методики "Предприятие", предусмотрен вариант отчета "Анализ показателей партии", как показано схематически на рисунке с номером 3.19.

← → ☆ Остатки (По партиям)

Сформировать Настройки... Изменить вариант... Выбрать вариант... Сохранить вариант...

Дата начала:  Начало этого месяца

Дата окончания:  ...

Склад:

Параметры: Дата начала: 01.04.2021

Номенклатура	Партия	На начало	Приход	Расход	На конец
Конструктор Lego # 3	Регистратор		60,000		60,000
Конструктор Lego # 3			60,000		60,000
	Приход от поставщика 000000002 от 01.04.2021 0:00:00		60,000		60,000
Конструктор Lego Fabrica большой			50,000	50,000	
Конструктор Lego Fabrica большой			50,000	50,000	
	Приход от поставщика 000000002 от 01.04.2021 0:00:00		50,000		50,000
	Реализация 000000001 от 06.04.2021 0:00:00	50,000		50,000	
Мягкая игрушка "Медведь" большой		300,000		2,000	298,000
Мягкая игрушка "Медведь" большой	Цвет: белый	300,000		2,000	298,000
		300,000			300,000
	Реализация 000000002 от 13.04.2021 13:52:23	300,000		2,000	298,000
Мягкая игрушка "Слон" большой		220,000		222,000	-2,000
Мягкая игрушка "Слон" большой	Цвет - синий	220,000		222,000	-2,000
		220,000			220,000
	Возврат поставщику 000000001 от 06.04.2021 0:00:00	220,000		20,000	200,000
	Инвентаризация ЦО-000001 от 12.04.2021 0:00:00	200,000		200,000	
	Реализация 000000002 от 13.04.2021 13:52:23			2,000	-2,000
<b>Итого</b>		<b>520,000</b>	<b>110,000</b>	<b>274,000</b>	<b>356,000</b>

Рисунок 3.19 — Отчет «Остатки» в варианте «Отчет по партиям».

База всякого отчета, который формируется на основе системы компоновки данных – схема, которая в обязательном порядке должна задаваться под отчет – рисунок 3.20.

Отчет Остатки

Имя:

Синоним:

Комментарий:

Основная схема компоновки данных:

Расширенное представление:

Пояснение:

Действия <Назад Далее> Закрывать Справка

Рисунок 3.20 — Задание схемы компоновки данных.

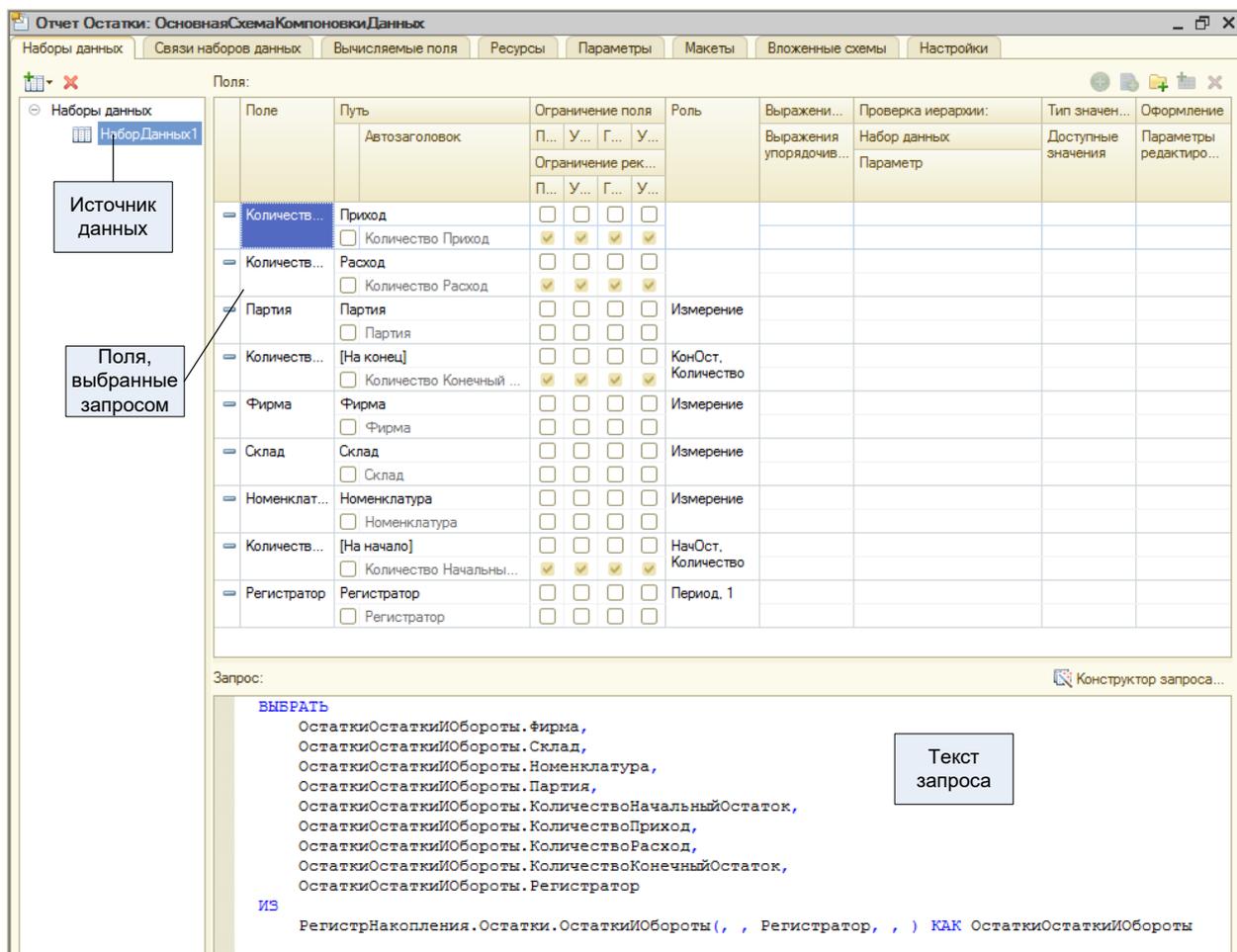


Рисунок 3.21 — Конструктор схемы компоновки данных.

Для целей создания единой интегральной системы сбора и анализа ключевых показателей, отражающих исследуемые процессы во всей их полноте и взаимосвязи, задействуется разветвленный конструкторский аппарат. Данная методологическая база обеспечивает комплексное и всестороннее представление о основополагающих детерминантах, систематизация которых необходима для формирования интегральной картины предстоящего исследования в виде многопараметрических данных. Как детально отображено на рисунке 3.21, применяемый подход комплексного конструирования позволяет в полном объеме отразить логическую цепочку последовательных преобразований для создания будущей обобщающей модели как интегрального образования. Настроечные параметры предоставляют возможность установить четкие рамки конкретной конфигурации отчета, структуру вводимых и

классифицируемых величин, другие детерминанты. Так, на рисунке 3.22 демонстрируется пример настройки «базисного» варианта отчета по остаткам ресурсов, а на рисунке 3.23 – настройка конфигурации «Отчета по партиям» как одной из разновидностей.

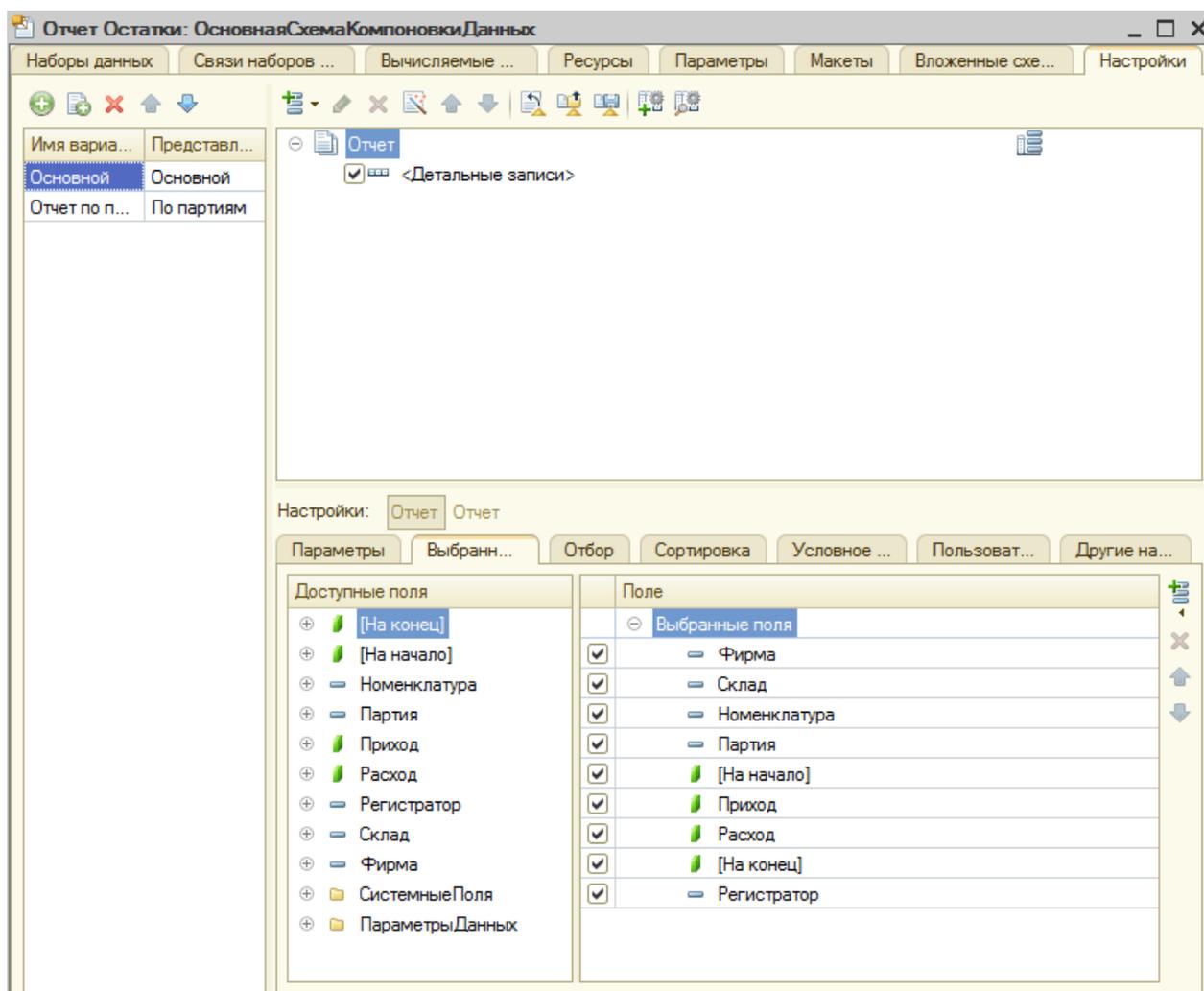


Рисунок 3.22 — Настройка варианта «Основной» отчета «Остатки».

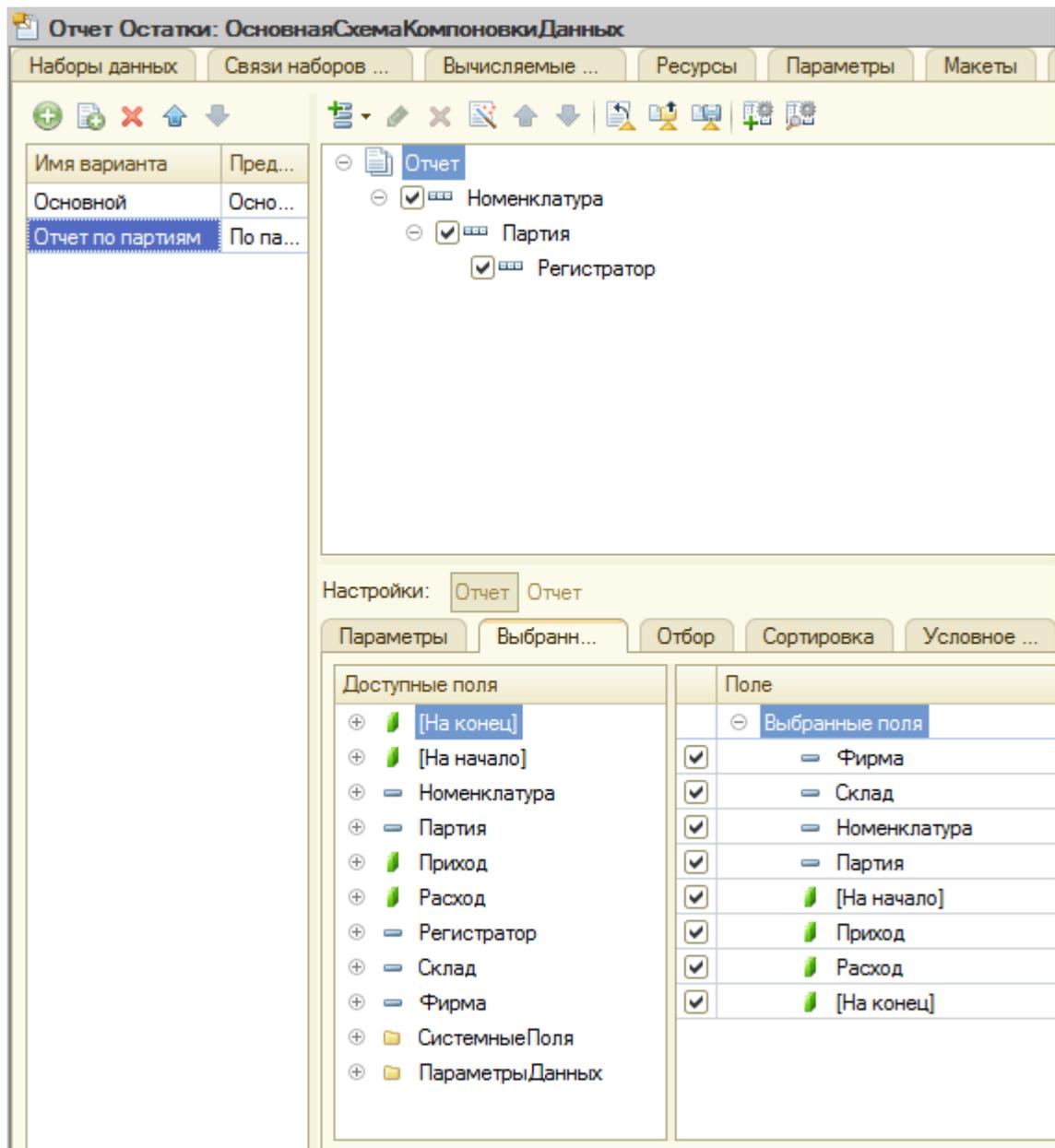


Рисунок 3.23 — Настройка варианта «Отчет по партиям» отчета «Остатки».

По итогам проведенных действий, в работе системы в режиме «Предприятие» можно получить отчет «Отчет по партиям». Внешне он показан на рисунке 3.19.

← → ☆ Остатки (По партиям)

**Сформировать** [Настройки...] [Изменить вариант...] [Выбрать вариант...] [Сохранить вариант...]

Дата начала:  Начало этого месяца Склад:

Дата окончания:  ..

Параметры: Дата начала: 01.04.2021

Номенклатура	Партия Регистратор	На начало	Приход	Расход	На конец
Конструктор Lego # 3			60,000		60,000
Конструктор Lego # 3			60,000		60,000
	Приход от поставщика 000000002 от 01.04.2021 0:00:00		60,000		60,000
Конструктор Lego Fabrics большой			50,000	50,000	
Конструктор Lego Fabrics большой			50,000	50,000	
	Приход от поставщика 000000002 от 01.04.2021 0:00:00		50,000		50,000
	Реализация 000000001 от 06.04.2021 0:00:00	50,000		50,000	
Мягкая игрушка "Медведь" большой		300,000		2,000	298,000
Мягкая игрушка "Медведь" большой	Цвет: белый	300,000		2,000	298,000
		300,000			300,000
	Реализация 000000002 от 13.04.2021 13:52:23	300,000		2,000	298,000
Мягкая игрушка "Слон" большой		220,000		222,000	-2,000
Мягкая игрушка "Слон" большой	Цвет - синий	220,000		222,000	-2,000
		220,000			220,000
	Возврат поставщику 000000001 от 06.04.2021 0:00:00	220,000		20,000	200,000
	Инвентаризация ЦО-000001 от 12.04.2021 0:00:00	200,000		200,000	
	Реализация 000000002 от 13.04.2021 13:52:23			2,000	-2,000
<b>Итого</b>		<b>520,000</b>	<b>110,000</b>	<b>274,000</b>	<b>356,000</b>

Рисунок 3.24 — Отчет «Остатки» в варианте «Отчет по партиям».

Посредством системы компоновки информации выстраиваются прочие отчеты конфигурации – рисунок 3.25.



Рисунок 3.25 — Состав отчетов системы.

### 3.4 Разработка интерфейса системы.

Если основополагающие принципы проектирования архитектуры взаимодействия различных элементов программного обеспечения строятся на парадигме функционирования в условиях режима "Управляемого программного комплекса", то концептуальные положения формирования юзер-интерфейса как основного инструмента взаимодействия с пользователем подвергаются значительной трансформации. Его внутренняя структура и логика построения существенно отличаются от характерных для простых самоуправляемых приложений подходов. Новая концепция ФПВВ направлена на обеспечение эффективного, комфортного и максимально соответствующего специфике решаемых задач взаимодействия пользователя со всем программно-аппаратным комплексом в рамках интегрированной среды в соответствии с прикладными задачами с целью обеспечения максимальной результативности [18].

Далее более детально рассмотрим основополагающие структурные элементы данной концепции во взаимосвязи: подсистему панельных инструментов и центральное рабочее пространство взаимодействия; командно-ориентированную систему управления; унифицированные форматы ввода-вывода информации и данных.

В центральном рабочем модуле реализуется возможность перехода к любому функциональному подразделу программного комплекса. На рисунке 3.26 на научно обоснованном уровне отражен один из возможных вариантов оптимального размещения необходимого комплекта панельных инструментов.

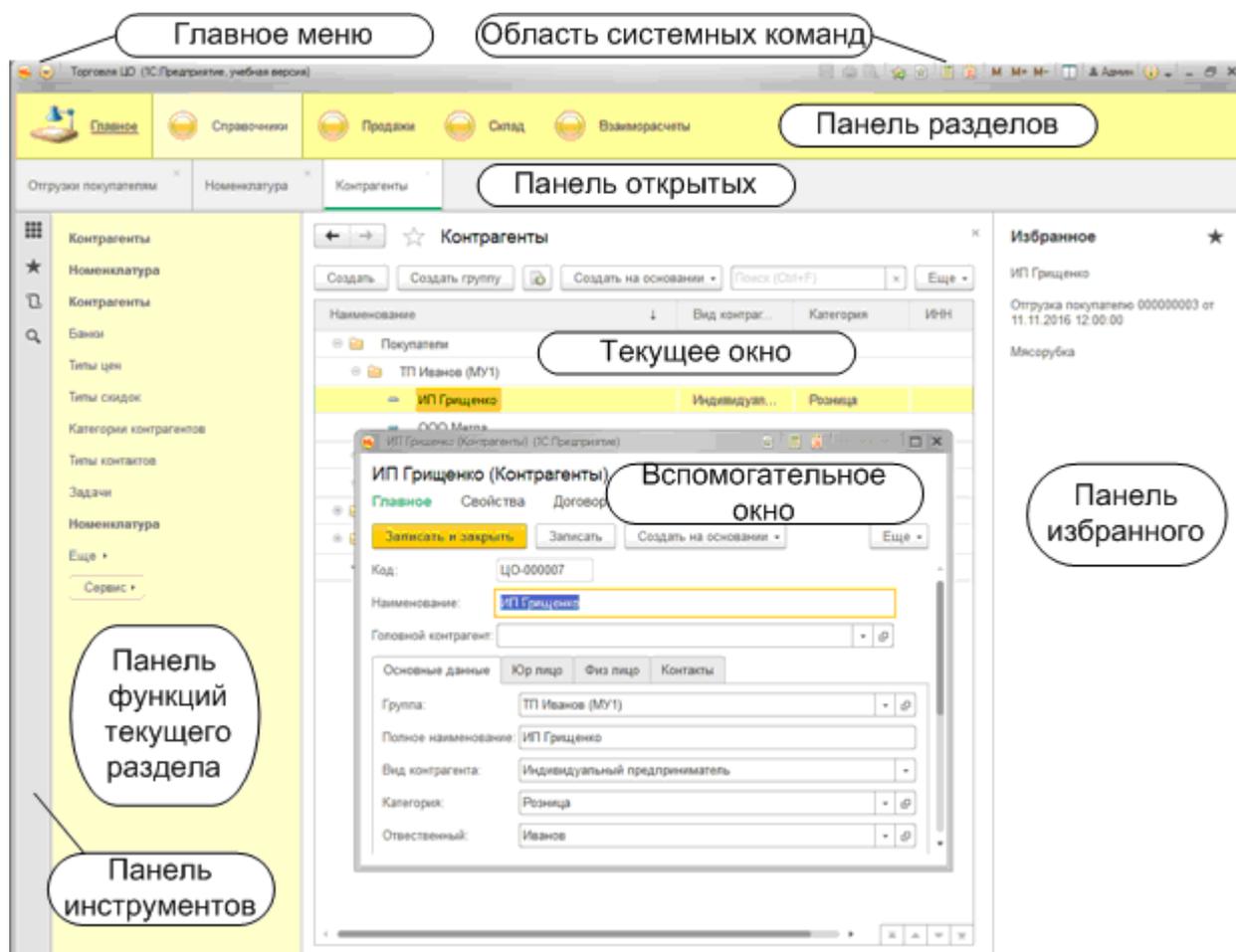


Рисунок 3.26 — Панели основного окна.

В центральном универсальном рабочем модуле основополагающая система управления всеми функциональными блоками и подсистемами, а также навигационные разделы для перемещения между ними, формируются путем задействования иерархически структурированной модели взаимосвязанных подсистем, заложенных на этапе конфигурирования в конструкторскую среду создаваемого автоматизированного информационного комплекса. Разработчику на основании этой модели необходимо получить полную информацию о логике функционирования всего программного обеспечения в составе единой системы. Затем, используя данные информационную модель и возможности редактора командного интерфейса, осуществляется конструирование панели разделов центрального универсального окна, как показано графически на рисунке 3.27 [9]. Полученная таким образом привязка элементов интерфейса к подсистемам позволяет автоматически сгенерировать оптимальную командную структуру

навигационных разделов для максимально удобного взаимодействия, как это демонстрируется на рисунке 3.28.

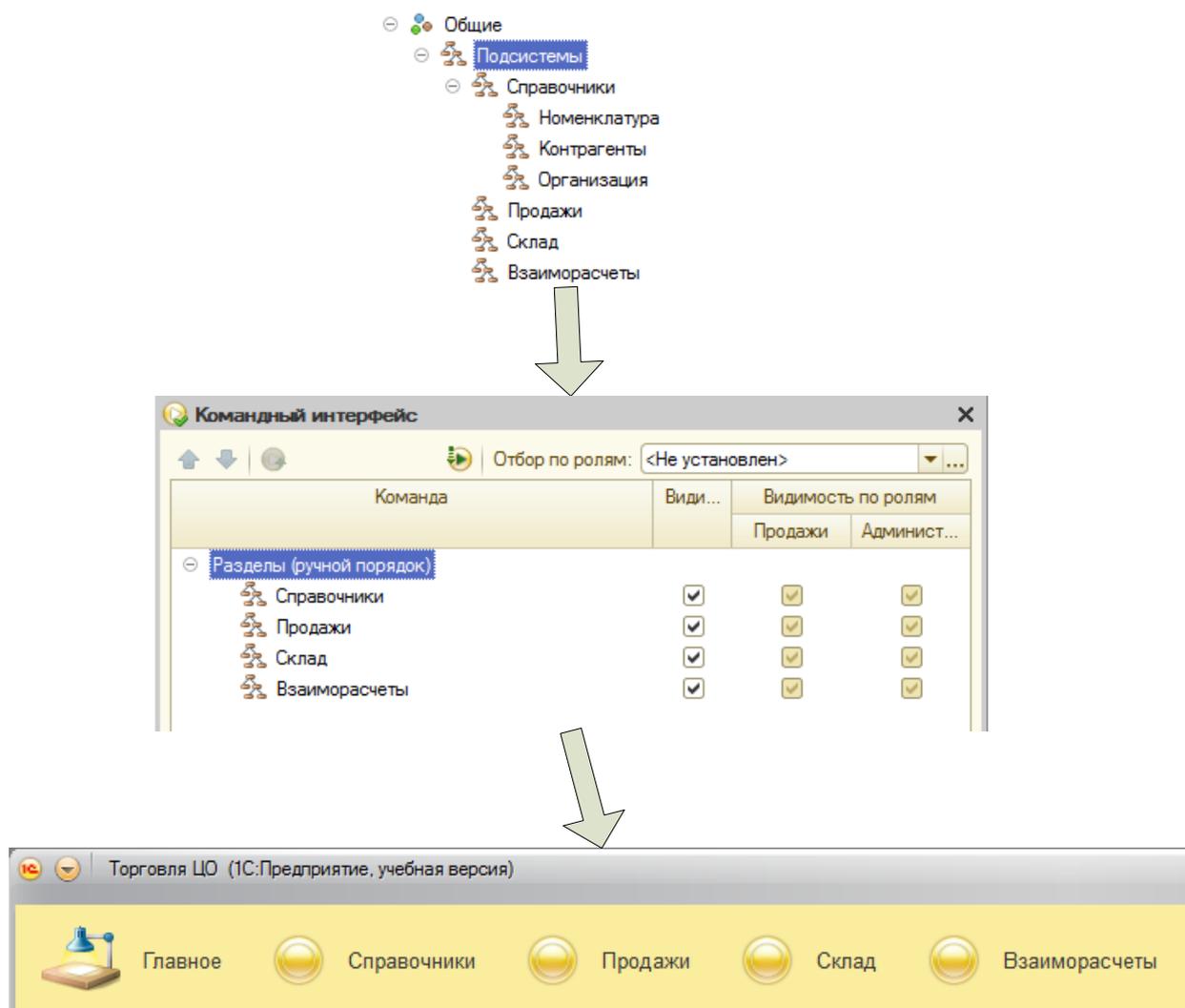


Рисунок 3.27 — Создание разделов на основе подсистем.

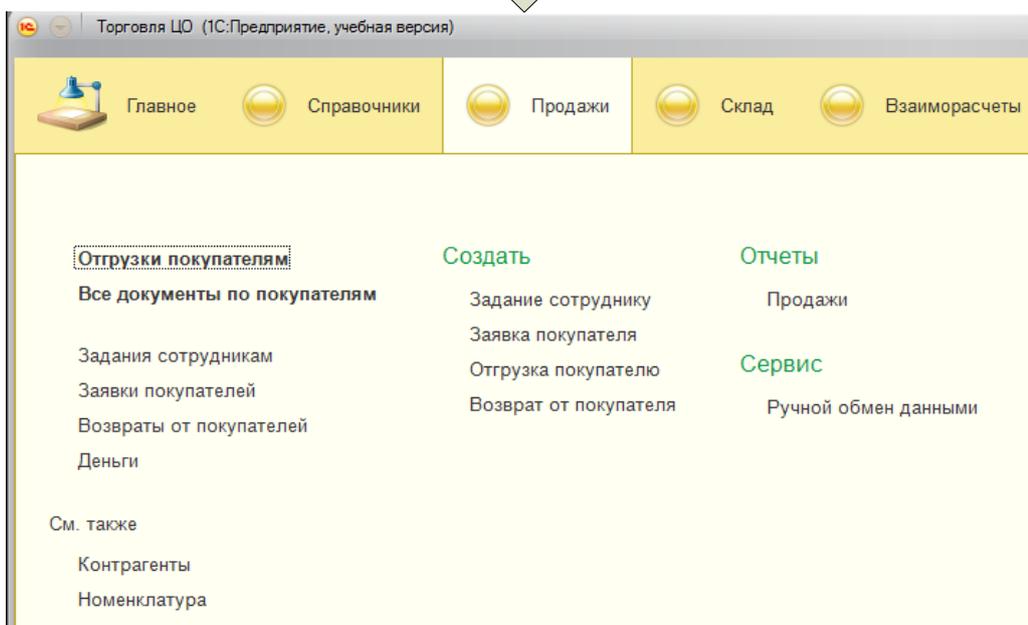
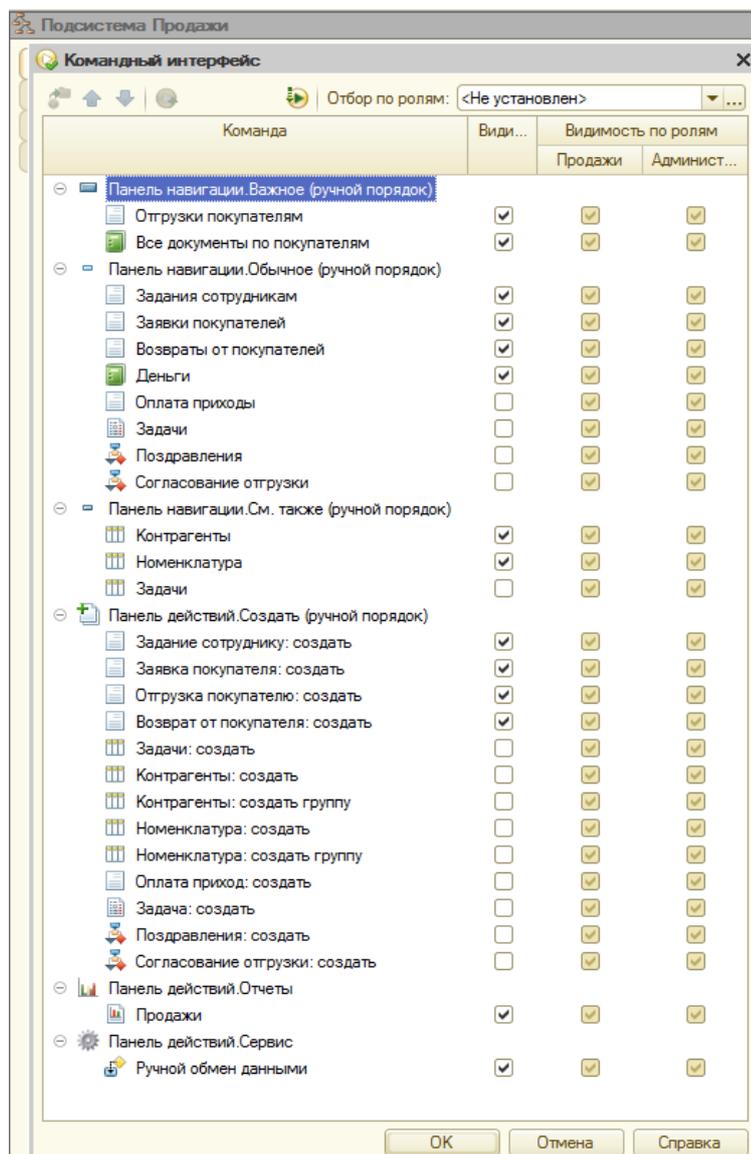


Рисунок 3.28 — Создание командного интерфейса раздела (подсистемы).

Обратим внимание, что командный интерфейс для всевозможных форматов ввода-вывода информации также может быть задан декларативным способом. Платформа автоматически распознает на основе внутренней логики структуру прикладного объекта. Далее, используя редактор командного интерфейса форм, разработчик определяет, какие элементы управления попадут в состав указанного интерфейса для данного объекта [13]. На рисунке 3.29 посредством конкретного примера отражена возможность создания оптимального прикладного интерфейса для справочной формы «Контрагенты» как одного из центральных информационных объектов в системе. Путем такого декларативного подхода достигается максимальная гибкость и адаптивность интерфейса к структуре любого объекта.

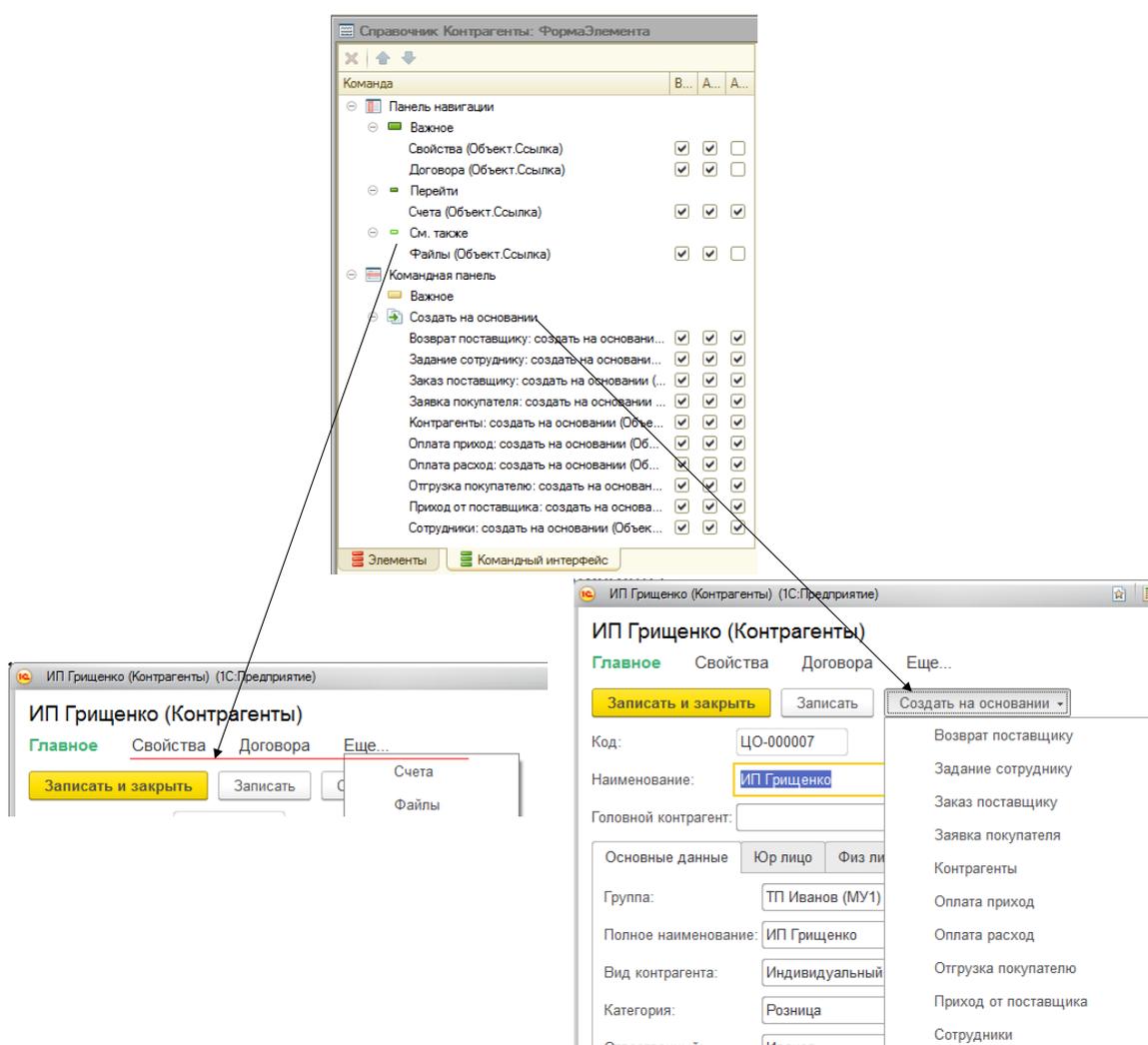


Рисунок 3.29 — Создание командного интерфейса формы элемента справочника «Контрагенты».

Подобно простым самодостаточным приложениям, интерактивные форматы представления информации служат основополагающим инструментом визуализации, редактирования и обработки рабочих данных как ключевого элемента автоматизированных систем [6].

На приведённом ниже рисунке 3.30 продемонстрирован пример размещения в интерфейсе формата необходимых элементов управления, обеспечивающих прямые ссылки на дополнительные источники сведений, характеризующих центральный объект данной формы. Посредством реализации такого подхода достигается максимально высокий уровень наглядности и удобства работы с полным объёмом взаимосвязанных данных о текущем объекте, что является одним из фундаментальных принципов проектирования эффективных современных интерфейсов пользователя. Это, в свою очередь, позволяет обеспечить оптимальный уровень производительности труда пользователей автоматизированной системы [19].

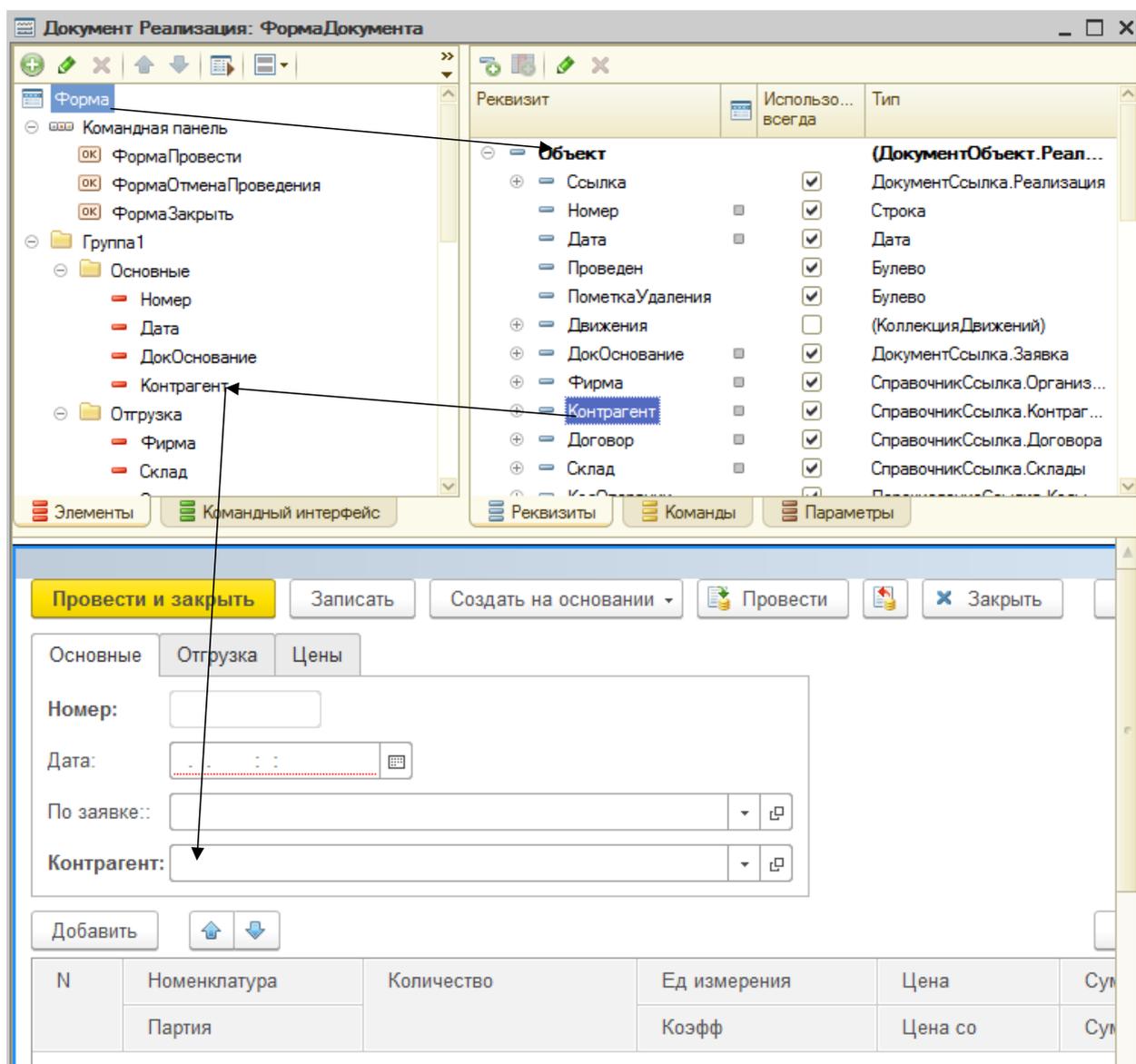


Рисунок 3.30 — Размещение реквизитов формы.

Интерактивный формат имеет определённую совокупность функциональных возможностей, которые реализуются посредством различных типов команд управления. К их числу можно отнести: стандартный набор команд конкретного формата, глобальные команды, распространяющие своё действие на всю конфигурацию в целом, а также пользовательские команды, создающиеся непосредственно разработчиком в соответствии с конкретными требованиями. Кроме того, предусмотрена возможность генерации параметризованных команд для вызова других форматов на основе заданных параметров и шаблонов [9].

На приведённом ниже рисунке 3.31 иллюстрируется процесс, в рамках которого в интерфейс командной панели добавляется дополнительная пользовательская команда наряду со стандартными элементами для обеспечения максимальной функциональности. Такой подход обеспечивает гибкость и эволюционную способность к расширению функциональных возможностей форматов посредством возможности создания индивидуальных команд непосредственно разработчиком под конкретные задачи взаимодействия с информацией.

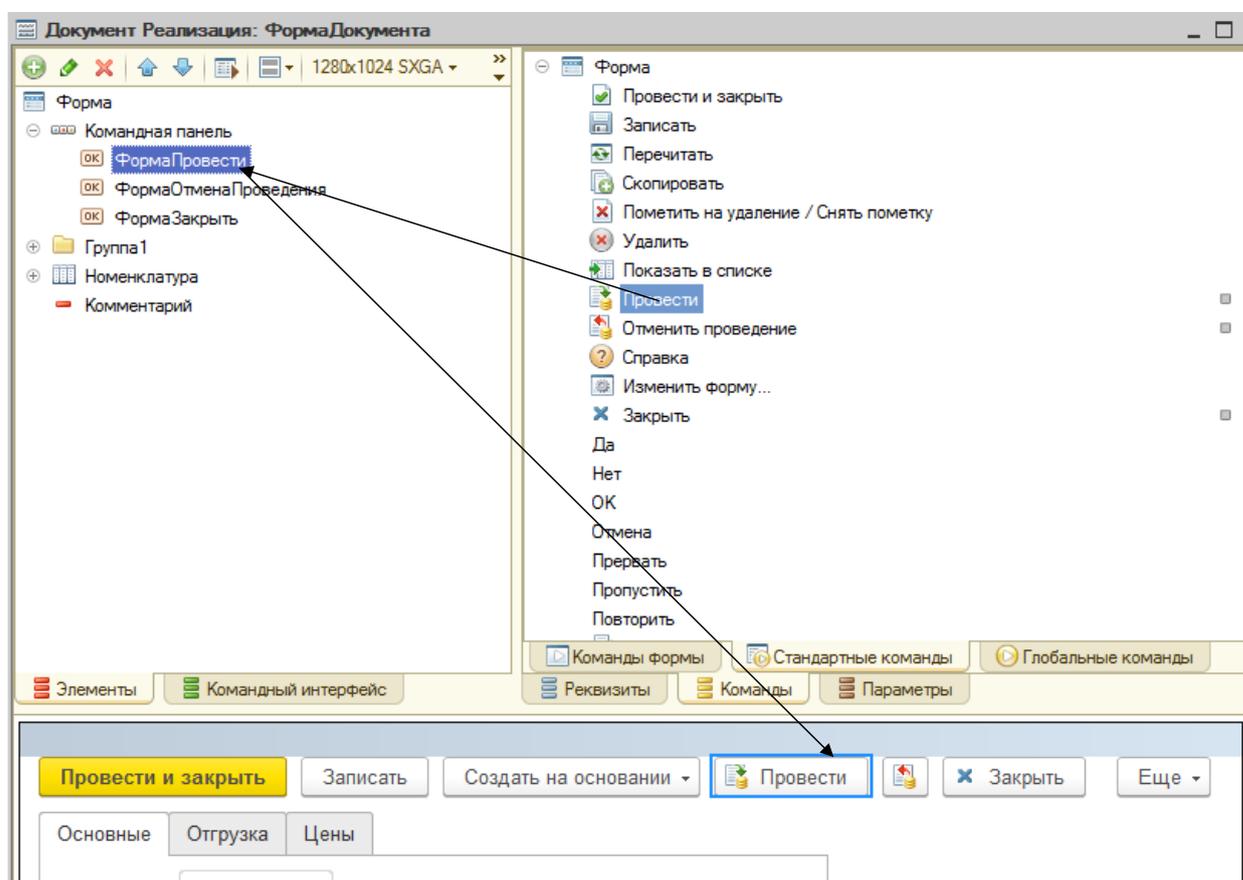


Рисунок 3.31 — Размещение команд формы.

В результате мы получили форму документа «Реализация», представленную на Рисунок 3.32.

← → ☆ Реализация 000000001 от 06.04.2021 0:00:00

Основное Взаиморасчеты Остатки Продажи

Провести и закрыть Записать Создать на основании Провести X Закрыть

Основное Отгрузка Цены

Номер: 000000001 Вид реализации: Продажа

Дата: 06.04.2021 0:00:00

Основание: Заявка покупателя 000000001 от 05.04.2021 0:00:00

Контрагент: ООО "Детский центр" Объект:

Добавить

N	Номенклатура Партия	Количество	Ед измерения		Цена	Сумма со скидк...
			Козэфф	Сумма скидки		
1	Конструктор Lego Fabrica большой	5,000	упак		870,00	43 065,00
					10,00	861,30
1	Итого мест:	5,000			Сумма итого:	43 065,00
					Скидка:	435,00

Рисунок 3.32 — Результирующая форма документа «Реализация».

### 3.5 Руководство по установке, настройке и сопровождению программы.

Подробно охарактеризуем последовательную процедуру развертывания ранее созданной информационной системы с учетом всех результатов, полученных на завершающей стадии ее тестирования и отладки.

Полученный в результате выгрузки объем данных представлен в виде единого архивного файла формата 1Cv8.dt, включающего в себя весь массив заполненной информации. Для организации хранения данной базы необходимо заранее подготовить уникальную папку-контейнер на выделенном накопителе объемных данных и скопировать в нее упомянутый архивный файл.

Далее иницируется процесс интеграции базы данных в общую систему 1С:Предприятие посредством вызова контекстного меню программы путем нажатия кнопки "Добавить" в главном окне, после чего следует активировать пункт "Импорт из файла" для загрузки ранее подготовленной базы из архива и включения ее в общий список доступных информационных ресурсов платформы (Рисунок 3.33).

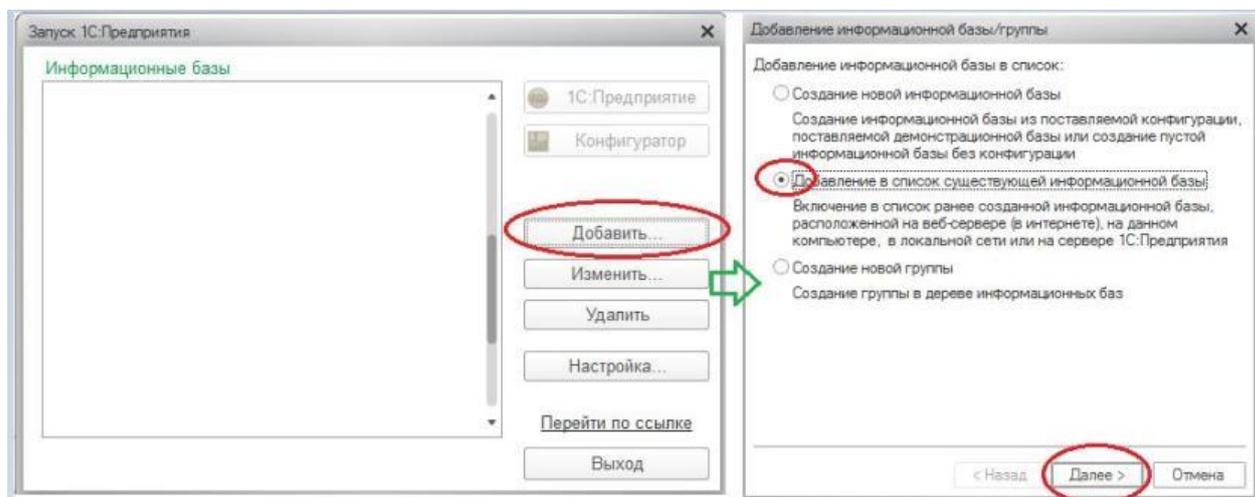


Рисунок 3.33 — Добавление новой информационной базы. Шаг 1.

В последующем открывается специальное окно настройки всех параметров нового проекта, в котором необходимо указать его уникальное наименование и выбрать соответствующую ранее сформированную папку для хранения всех файлов проекта. После перехода к следующему этапу формирования конфигурации отображается дополнительное окно параметров, где как правило можно сохранить базовые настройки по умолчанию и подтвердить готовность к продолжению работы командой "Готово", как дано иллюстрации на снимке экрана 3.34.

В результате, в общем перечне доступных информационных баз появляется элемент, соответствующий новому проекту. Необходимо выполнить его выбор и запустить среду разработки 1С:Предприятие 8.3 в режиме конфигурирования. Далее система автоматически определит, что данный объект не содержит каких-либо начальных данных и предложит без проволочек создать полностью новую конфигурацию, как иллюстрируется на последующем рисунке 3.35.

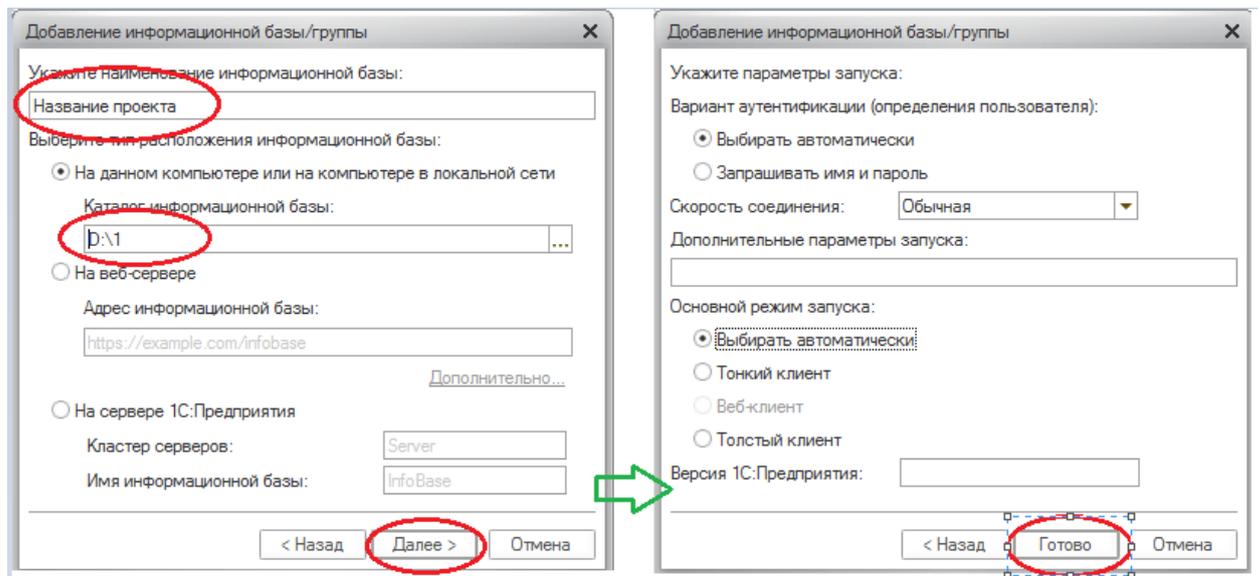


Рисунок 3.34 — Добавление новой информационной базы. Шаг 2.

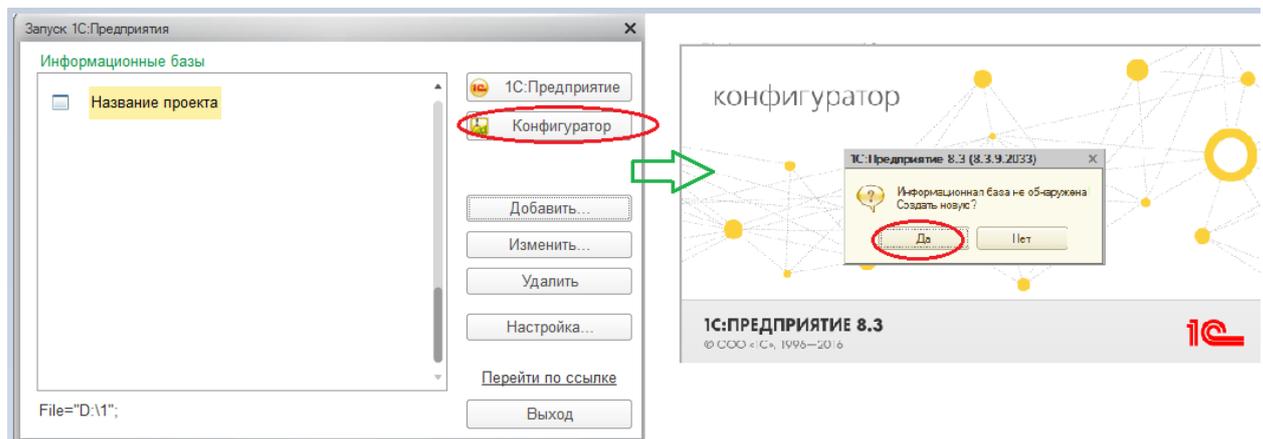


Рисунок 3.35 — Первый запуск «Конфигуратора».

Далее следует перейти в главное меню рабочей среды configurator 1C и выбрать раздел "Администрирование", а затем подпункт "Загрузить информационную базу", как иллюстрируется на приведенном снимке экрана 3.36. Необходимо указать точный путь к ранее подготовленному архивному файлу с расширением 1Cv8.dt, хранящемуся во временной папке проекта. После чего нужно будет подтвердить все последующие запросы системы касательно завершения операции импорта.

В конечном итоге происходит загрузка выбранной базы данных в активную рабочую среду, однако без сохранения изменений в конфигурации. Чтобы завершить процедуру успешно, необходимо выбрать пункт меню

"Отладка - Начать отладку" на панели инструментов, тем самым иницилируя фиксацию состояния базы и переключение среды 1С:Предприятие 8.3 в специальный режим отладки для проведения диагностики и коррекции возможных ошибок.

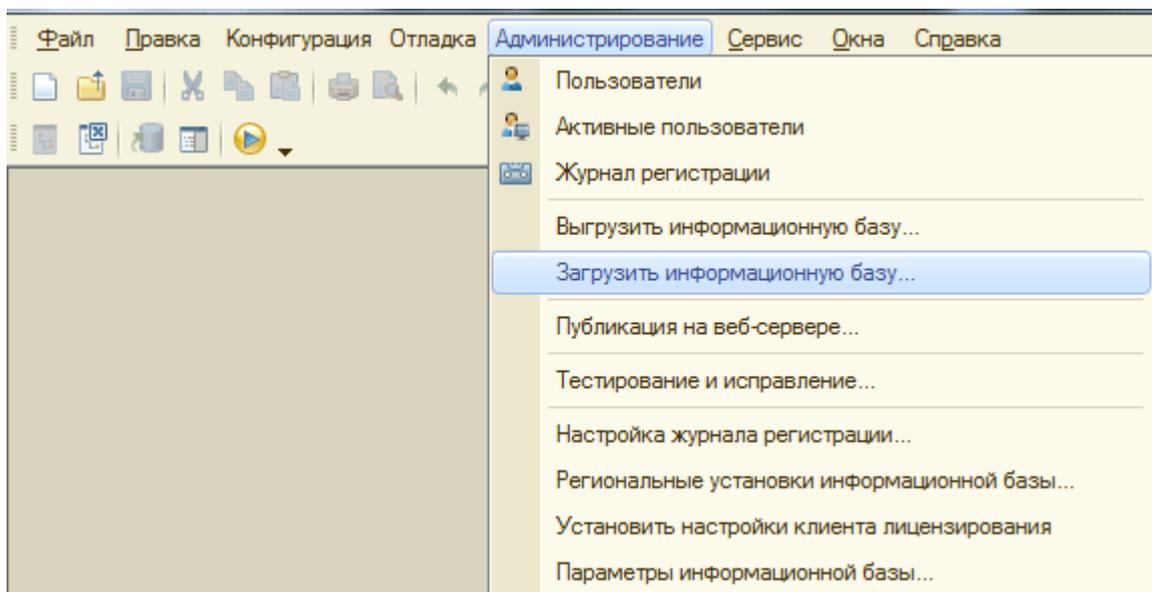


Рисунок 3.36 — Загрузка информационной базы из архива.

В перспективе включение прикладного решения можно проводить в режиме «Предприятие». В списке выбирается нужная база.

### 3.6 Руководство пользователя.

Работа с информационной базой данных начинается с тщательной процедуры заполнения вспомогательных справочных регистров, роли которых заключаются в хранении справочных значений различных параметров и атрибутов. Указанные значения впоследствии могут быть использованы для комплектации разнообразных документов как набор входных требуемых реквизитов. Одним из наиболее значимых и обширных по объему информации являются справочники "Контрагенты" и "Номенклатура" - фундаментальный справочник, отражающий весь спектр товарно-материальных ценностей.

Справочник "Номенклатура" активизируется через вызов меню "Справочники", за чем следует выбор одноимённой команды. Данный справочник наделён иерархической структурой для группировки элементов по логическим признакам, обеспечивая навигацию по вложенным категориям посредством древовидного представления, как показано на приведённой иллюстрации 3.37. Это обеспечивает высокую эффективность поиска требуемой информации.

← → ★ Номенклатура

Создать Создать группу [иконка] [иконка] [иконка] [иконка] ✕ Закрыть [Остатки](#) [Наборы свойств](#) [Ед. измерения](#) [Цены](#) [Мин](#)

Наим-ие

▼ Консервы мясные

— Тушенка говяжья 500 гр (Королево)

★ Тушенка говяжья 500 гр (Королево) (Номенклатура)

Основное [Остатки](#) [Наборы свойств](#) [Ед. измерения](#) [Минимальный остаток](#)

**Записать и закрыть** Записать

Код: 000000005

Наим-ие: Тушенка говяжья 500 гр (Королево)

Полное: Тушенка говяжья 500 гр (Королево)

Группа: Консервы мясные [иконка]

Вид: Услуги логистики [иконка]

Ед. изм. баз: шт [иконка]

Ед. остатков: [иконка]

Ед. отгрузки: [иконка]

Объем: 0,000 [иконка] Масса нетто: 0,000 [иконка] Масса брутто: [иконка]

Штрих код: [иконка]

Комментарий: [иконка]

Рисунок 3.37 — Список номенклатуры.

Рассматривая особенности функционирования справочника "Контрагенты", следует отметить полное аналогичное соответствие процессов работы с ним справочнику "Номенклатура". Доступ к данному вспомогательному реестру организован посредством вызова команды соответствующего имени из динамического меню раздела "Справочники". Как

и в случае со справочником "Номенклатура", группировка элементов в данном ресурсе основана на иерархической структуре.

В целях обеспечения просмотра и возможной правки сведений конкретного контрагента, необходимо использовать двойной щелчок, вызывающий открытие отдельно оформленного окна с настройками и реквизитами выбранного элемента, как иллюстрирует изображение 3.38. Количество заполняемых реквизитов может значительно варьироваться в зависимости от конкретных потребностей, при этом особое значение имеет раздел "Контакты", содержащий как юридические, так и физические адресные данные выбранного контрагента.

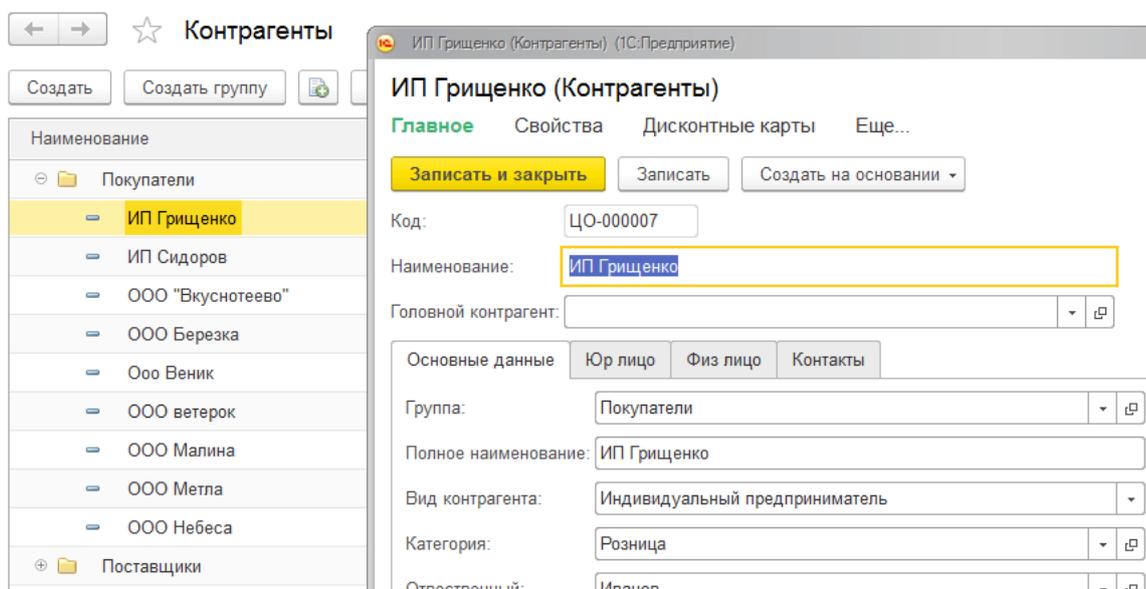


Рисунок 3.38 — Справочник «Контрагенты».

Вся агрегированная совокупность функциональных документов, предусмотренных возможностями информационной системы, логически структурирована на три ключевые группы в зависимости от предметно-процессуального назначения их использования. Первая группа включает в себя проектно-коммерческие документы; вторая - документы, относящиеся к управлению складскими операциями и снабжению; третья - бухгалтерские документы финансового учета, организующие обработку как безналичных, так и наличных расчётов.

При этом значительная часть типов документов связана между собой процессуальными зависимостями, в силу чего в определённых ситуациях они могут одновременно относиться к нескольким группам в зависимости от контекста их заполнения/обработки. Последовательность цепочки формирования и влияния документов друг на друга графически представлена на рисунке 3.39.

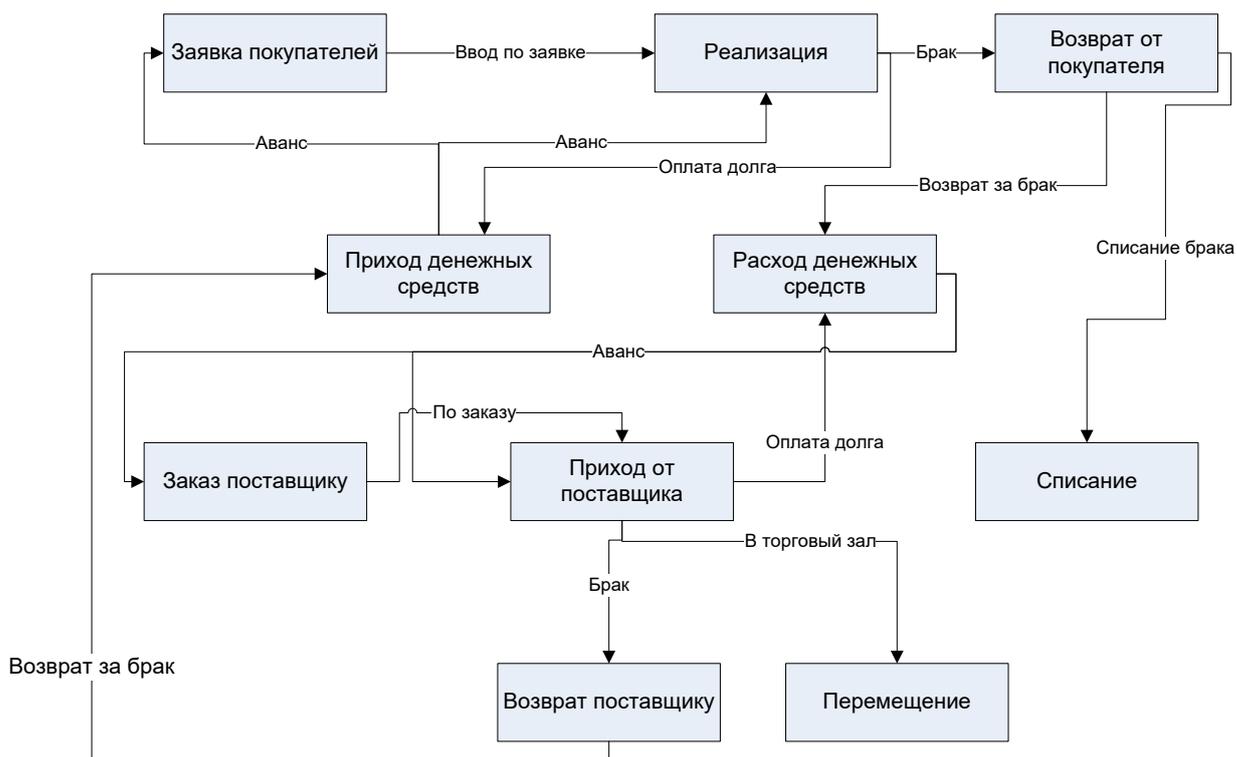


Рисунок 3.39 — Последовательность ввода документов.

Вариативность функциональных типов документов, объединенных в рамках определенных групп посредством логических процессуальных взаимосвязей, предполагает формирование сводных регистров-журналов для удобства работы пользователей. В частности, в рассматриваемой информационной системе реализован механизм формирования комплексного журнала "Все документы по покупателям", интегрирующего весь спектр информации об операциях с контрагентами, а также журнала "Деньги", отображающего денежные поступления и расходы, как в безналичной, так и в наличной формах.

Структура данных указанных журналов документов представлена на снимках экрана 3.40 и 3.41. Данные регистры обеспечивают всевозможный функциональный просмотр процессов работы с различными типами документов в рамках логических групп посредством унифицированного графического интерфейса, что является ключевым условием для оперативного контроля хода бизнес-процессов и принятия управленческих решений на его основе.

Дата	Номер	Тип документа	Фирма	Контрагент	Склад
15.04.2024 0:00:00	000000002	Реализация	ООО "ПродТорг"	Магазин "У Дома" ООО "Велес" (Индустриальная, 15)	Склад №1
16.04.2024 0:00:00	000000001	Реализация	ООО "ПродТорг"	Коровин Владимир Сергеевич	Склад №1
17.04.2024 0:00:00	000000003	Реализация	ООО "ПродТорг"	Магазин "У Дома" ООО "Велес" (Индустриальная, 15)	Склад №1

Рисунок 3.40 — Журнал документов по покупателям.

Дата	Номер	Тип документа	Фирма	Счет	Контрагент
15.04.2024 0:00:00	000000002	Реализация	ООО "ПродТорг"		Магазин "У Дома" ООО "Велес" ...
16.04.2024 0:00:00	000000001	Реализация	ООО "ПродТорг"		Коровин Владимир Сергеевич
17.04.2024 0:00:00	000000003	Реализация	ООО "ПродТорг"		Магазин "У Дома" ООО "Велес" ...
02.05.2024 16:28:02	000000001	Оплата приход	ООО "ПродТорг"	Основной счет	Магазин "У Дома" ООО "Велес" ...

Рисунок 3.41 — Журнал «Деньги».

Для целей демонстрации основных особенностей процесса работы с функциональными документами рассмотрим подробно пример документа "Отгрузка покупателю". Доступ к нему реализован посредством вызова соответствующей команды из динамического меню раздела "Продажи".

При заполнении табличной части предоставляется выбор номенклатурных позиций, после чего система производит автоматическую генерацию данных об единицах измерения каждой позиции и расчёт итоговых сумм с учётом применимых тарифов и скидок. Внешний вид раскрытого документа демонстрируется на иллюстрации 3.42.

Как показано на примере, структура ввода и логика обработки данных рассмотренного документа типична и для всех остальных функциональных документов аналогичного назначения. Порядок их вызова, а также структура заполняемой информации строится по унифицированным принципам, обеспечивающим согласованность между документами при последовательном их заполнении в рамках бизнес-процессов.

N	Номенклатура	Партия	Количество	Ед изме...	Ко...	Цена	Цена со скидкой	Сумма со скид...
1	Сыр "Сметанковый" 75% (Балаево)		3,000	кг	1,00	720,00	705,60	2 116,80
2	Тушенка говяжья 500 гр (Короплево)		5,000	шт	1,00	380,00	372,40	1 862,00

Рисунок 3.42 — Документ «Отгрузка покупателю».

Одной из ключевых особенностей работы с отчётными формами является возможность конфигурирования неограниченного числа параметров для каждой отчётной формы, позволяющая на основе единой структуры получать данные, фильтруемые различными критериями выборки информации.

Система обеспечивает функциональность сохранения конфигурации параметров запроса, благодаря чему в рамках одной логической единицы - отчётной формы - можно формировать множество её функциональных вариантов для решения самых разнообразных аналитических и управленческих задач.

Процесс параметризации отчёта детально демонстрируется на рисунке 3.43. Результат генерации отчёта для заданного конкретного набора настроек отображается на иллюстрации 3.44. Данная функциональная возможность обеспечивает максимальную гибкость и универсальность инструментария отчётности.

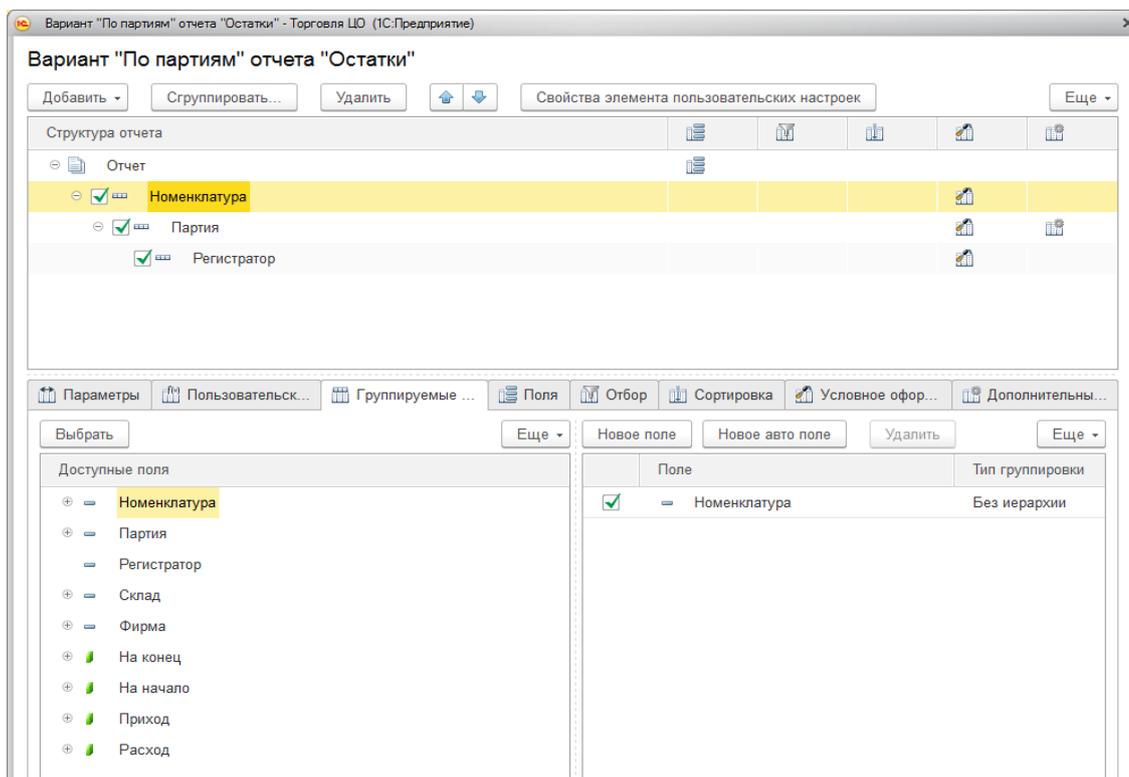


Рисунок 3.43 — Настройка варианта «по партиям» отчета «Остатки».

← → ☆ Остатки (По партиям)

Сформировать Изменить вариант... Выбрать вариант... Сохранить вариант... Настройки... ✕ Закрыть

Дата начала:  Начало этого месяца  Склад:

Дата окончания:  ..

Номенклатура	Партия	На начало	Приход	Расход	На конец
	Регистратор				
Колбаса "Финский сервилат" 500 гр (Папа может)			200,000	72,000	128,000
Колбаса "Финский сервилат" 500 гр (Папа может)	До 31.05.24		200,000	72,000	128,000
	Приход от поставщика 000000001 от 01.04.2024 0:00:00		200,000		200,000
	Инвентаризация МУ1-00001 от 04.04.2024 0:00:00	200,000		2,000	198,000
	Реализация 000000001 от 16.04.2024 0:00:00	198,000		20,000	178,000
	Реализация 000000003 от 17.04.2024 0:00:00	178,000		50,000	128,000
Сыр "Сметанковый" 75% (Балаево)			10,000	3,000	7,000
Сыр "Сметанковый" 75% (Балаево)			10,000	3,000	7,000
	Приход от поставщика 000000002 от 02.04.2024 0:00:00		10,000		10,000
	Реализация 000000002 от 15.04.2024 0:00:00	10,000		3,000	7,000
Тушенка говяжья 500 гр (Короллево)			100,000		100,000
Тушенка говяжья 500 гр (Короллево)			100,000		100,000
	Приход от поставщика 000000002 от 02.04.2024 0:00:00		100,000		100,000
<b>Итого</b>			<b>310,000</b>	<b>75,000</b>	<b>235,000</b>

Рисунок 3.44 — Результат варианта «По партиям» отчета «Остатки».

### 3.7 Выводы по главе 3.

На основе проектной части предыдущей главы была проведена разработка итогового проекта. Приведены общая структура проекта, реализованы механизмы торгового и складского учета, документации и номенклатуры. Реализована система отчетов и запросов. Разработан интерфейс

системы. Описано подробное руководство пользователя вместе с руководством по установке настройке и сопровождению.

## ЗАКЛЮЧЕНИЕ

В рамках настоящей выпускной квалификационной работы был проведен детальный всесторонний комплексный анализ существующей деятельности предприятия ООО «Продторг», тщательно изучена его организационная структура, специфика всех бизнес-процессов и особенности процесса внедрения новой системы автоматизации. Очень глубоко и скрупулезно была исследована главная цель и задача по полной автоматизации торгового и складского учета в данной компании. Успешное решение этой масштабной задачи позволило провести реинжиниринг и оптимизацию ключевых бизнес-процессов на предприятии.

Также был выполнен детализированный сравнительный анализ существующих методов и автоматизированных информационных систем, предназначенных для торгового учета, по результатам которого был сделан обоснованный вывод о необходимости разработки новой специализированной уникальной АИС. Было сформулировано техническое задание на создание такой системы, изучены все доступные средства и технологии разработки, и произведен взвешенный выбор оптимального решения для наиболее эффективной реализации поставленных задач. Также были проанализированы современные системы управления базами данных, и произведен хорошо аргументированный выбор наиболее подходящей и лучшей СУБД для данного конкретного проекта автоматизации.

На следующем этапе было выполнено детальное проектирование базы данных и информационной модели будущей автоматизированной системы. Была тщательно разработана схема эффективного кодирования всех классификаторов базы данных. Построено дерево основных функций и составлен подробный сценарий типового диалога для АИС торгового учета. По совокупным результатам проектирования в итоге была получена полноценная информационная модель с четким определением всех входных и выходных данных системы.

Были спроектированы и реализованы все основные ключевые объекты итогового прикладного решения: различные документы, справочники, регистры сведений и накопления, а также необходимые запросы и отчеты. Созданы полноценные экранные формы интерфейса, программные модули обработки данных и запросы к базе данных для соответствующих объектов разработанной системы.

В результате была разработана комплексная автоматизированная информационная система, полностью охватывающая все необходимые аспекты и стороны ведения торговой деятельности предприятия. Результаты тестирования подтвердили работоспособность и корректность созданной системы. Внедрение разработанной АИС на производственных мощностях предприятия ООО «Продторг» позволит успешно нормализовать и оптимизировать все процессы учета торговых операций на предприятии.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Калянов Г.Н. CASE-технологии. Консалтинг при автоматизации предприятий.-М.: СИНТЕГ, 2021.-276с.;
2. Новости и технологии торговли [Электронный ресурс]. – Режим доступа: <http://www.torgrus.com>;
3. ГОСТ Р ИСО/МЭК 12207-99, Руководящий документ, Госстандарт России, Москва, 2004;
4. Максимов Н.В., Попов И.И., Голицына О.Л. Информационные системы. Учебное пособие. Форум, 2019 г. - 496 с.;
5. Котлер Ф. Маркетинг менеджмент / Пер. с англ. под ред. Л.А. Волковой, Ю.Н. Каптунеревского. – СПб.: Питер, 2019. – 756с.р;
6. "1С:Предприятие 8. Система программ [Электронный ресурс]. - <http://v8.1c.ru/>;
7. Проектирование экономических информационных систем: Учебник/Смирнова Г.Н., Сорокин А.А., Тельнов Ю.Ф.; Под ред. Тельнова Ю.Ф. – М.:Финансы и статистика, 2021. - 512 с.;
8. Профессиональная разработка в системе «1С:Предприятие 8» (+DVD-ROM). Издание 2. Ажеронок В.А., Габец А.П., Гончаров Д.И., Козырев Д.В., Кухлевский Д.С., Островерх А.В., Радченко М.Г., Хрусталева Е.Ю. Под редакцией М. Г. Радченко. 870 с. 1С-Пабблишинг, 2020 г.;
9. Разработка управляемого интерфейса (+CD). Серия "1С:Профессиональная разработка". Ажеронок В.А., Островерх А. В., Радченко М. Г., Хрусталева Е. Ю. 740 с. 1С-Пабблишинг, 2021 г.;
10. Диго С.М. Проектирование и использование баз данных Учебник. М.: Финансы и статистика. 2018 г;
11. Матищев А.Н. Эффективность рекламы. – М.: Издательство «Финпресс», 2018. – 416с.;
12. Основы построения баз данных под ред. А.Д. Хомоненко Санкт-Петербург, 2021;

13. 1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы. Радченко М.Г., Хрусталева Е.Ю. 620 с. 1С-Паблишинг, 2021 г.;
14. Методология функционального моделирования IDEF0, Руководящий документ, Госстандарт России.;
15. Программирование для всех [Электронный ресурс]. – Режим доступа: <http://www.realcoding.net/>;
16. ГОСТ Р ИСО/МЭК 12207-99 «Информационная технология. Процессы жизненного цикла программных средств» (принят и введен в действие постановлением Госстандарта РФ от 23.12.1999 №675-ст).;
17. 1С:Предприятие 8.3. Практическое пособие разработчика. Примеры и типовые приемы (+диск). М.Г. Радченко, Е.Ю. Хрусталева. 968 с. 1С-Паблишинг, 2019 г.;
18. Архитектура и работа с данными "1С:Предприятия 8.3". Серия «1С:Профессиональная разработка». Радченко М. Г., Хрусталева Е. Ю. 650 с. 1С-Паблишинг, 2020 г.;
19. Решение специальных прикладных задач в "1С:Предприятии 8.3". Серия "1С:Профессиональная разработка". Гончаров Д. И., Хрусталева Е. Ю. 935 с. 1С-Паблишинг, 2021 г.;
20. Реализация прикладных задач в системе "1С:Предприятие 8.3" (+CD). Серия "1С:Профессиональная разработка". Габец А.П., Козырев Д.В., Кухлевский Д.С., Хрусталева Е.Ю. 830 с. 1С-Паблишинг, 2020 г.;
21. 1С:Предприятие 8.2. Коротко о главном. Новые возможности версии 8.3. Радченко М.Г. 690 с. 1С-Паблишинг, 2020 г.;
22. 1С:Предприятие 8.2. Коротко о главном. Новые возможности версии 8.3. Радченко М.Г. 690 с. 1С-Паблишинг, 2020 г.;
23. Вендров А.М. CASE технологии Современные методы и средства проектирования информационных систем М.: Финансы и статистика, 2019. - 176 с.: ил.;

24. ГОСТ Р ИСО/МЭК 15271-2002 «Информационная технология. Руководство по применению ГОСТ Р ИСО/МЭК 12207-99 (Процессы жизненного цикла программных средств)» (принят и введен в действие постановлением Госстандарта РФ от 05.06 2002 №227-ст).;
25. Вендров А.М. Проектирование программного обеспечения экономических информационных систем. – М.: Финансы и статистика, 2020.- 187с.;
26. Репин В.В., Елиферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов. – М.: РИА «Стандарты и качество», 2018. – 408 с., илл. – (Серия «Практический менеджмент»);
27. Петров В.Н. Информационные системы. – СПб.: Питер, 2003. – 688 с.;
28. Информационные системы в экономике: Учебник для студ. высш. учеб. заведений / В.Б. Уткин, К.В. Балдин. — М.: Издательский центр «Академия», 2019. — 288 с.;
29. Арутюнов В.В. Теория экономических информационных систем: Московская финансово-промышленная академия. М., 2021. – 192 с.;
30. Балдин К.В., Уткин В.Б. Информационные системы в экономике. М.:Издательско-торговая корпорация «Дашков и К», 2021. – 395 с.;
31. Гагарина Л.Г., Киселев Д.В., Федотова Е.Л. Разработка и эксплуатация автоматизированных информационных систем / под ред. проф. Л.Г. Гагариной. – Москва: ИД «Форум»: Инфра-М, 2019 – 384 с.;
32. Ефимов Е.Н., Патрушина С.М., Панферова Л.Ф., Хашиева Л.И. Информационные системы в экономике. – Москва: ИКЦ «МарТ», 2019. – 352 с.;
33. Липаев В.В. Программная инженерия. Методологические основы. М.: ТЕИС, 2020. – 608 с.

## ПРИЛОЖЕНИЕ А.

### ТЕКСТ ПРОГРАММНЫХ МОДУЛЕЙ

Общий модуль «Общий Срв»

Функция СоздатьОписаниеТипов(НазваниеТипа, Длина = "", Точность = "")

Экспорт

    мас = Новый Массив;

    мас.Добавить(Тип(НазваниеТипа));

    Если Длина <> "" Тогда

        Если Точность = "" Тогда

            КвалифСтроки = Новый КвалификаторыСтроки(Длина, ДопустимаяДлина.Переменная);

        Иначе

            КвалифЧисла = новый КвалификаторыЧисла(Длина, Точность, ДопустимыйЗнак.Любой);

        КонецЕсли;

    КонецЕсли;

    Возврат Новый ОписаниеТипов(мас, КвалифСтроки, КвалифЧисла);

КонецФункции

Функция ПолучитьОсновнойТипЦен() Экспорт

    Возврат Константы.ОсновнойТипЦен.Получить();

КонецФункции

Функция ПолучитьОсновнойТипСкидки() Экспорт

    Возврат Константы.ОсновнойТипСкидок.Получить();

КонецФункции

// для одной строки документов при расчете в ней цены и суммы  
Процедура РассчитатьСуммуСтроки(Дата, Номенклатура, Количество, Коэфф, ТипЦен, Скидка=0, Цена, ЦенаСоСкидкой=0, Сумма, СуммаСкидки=0) Экспорт

    Цена = ПолучитьЦену(Дата, Номенклатура, ТипЦен);

    ЦенаСоСкидкой = Цена - ((Цена\*Скидка)/100);

    Сумма = ЦенаСоСкидкой \* Количество\*Коэфф;

    СуммаСкидки = Цена\*Количество\*Коэфф -

    ЦенаСоСкидкой\*Количество\*Коэфф;

КонецПроцедуры

// ПО КОНТРАГЕНТАМ

// получить ответственного сотрудника для контрагента

Функция ПолучитьОтветственного(КонтрагентСсылка) Экспорт

    Если Не КонтрагентСсылка.Пустая() Тогда

        Возврат КонтрагентСсылка.Ответственный;

    КонецЕсли;

    Возврат Неопределено;

КонецФункции

```

// получить основной договор контрагента
Функция ПолучитьОсновнойДоговор(ДляКонтрагента, НаДату, ТипЦен=0,
ТипСкидки=0, Скидка=0) Экспорт

    глПустаяДата = Дата(01,01,0001,00,00,00);
    Договор = Неопределено;

    Если НаДату = глПустаяДата Тогда// если дата еще не известна - то
берем на текущую дату
        НаДату = ТекущаяДата();
    КонецЕсли;
    // запросом выбираем все договора этого контрагента с признаком
Основной и действующие по дате
    Запрос = Новый Запрос;
    Запрос.Текст =
"ВЫБРАТЬ
|     Договора.Ссылка                КАК Договор,
|     Договора.Владелец              КАК Владелец,
|     Договора.Основной              КАК Основной,
|     Договора.ДатаНачалаДеяствия    КАК ДатаНачалаДеяствия,
|
|     Договора.ДатаОкончанияДеяствия КАК ДатаОкончанияДеяствия
|
| ИЗ
|     Справочник.Договора КАК Договора
|     ГДЕ Владелец = &ДляКонтрагента И
|     Основной = ИСТИНА И
|     ДатаНачалаДеяствия <= &НаДату
// |     ДатаОкончанияДеяствия >= &НаДату
|";

    Запрос.УстановитьПараметр("ДляКонтрагента", ДляКонтрагента);
    Запрос.УстановитьПараметр("НаДату", НаДату);
    Результат = Запрос.Выполнить();

    Выборка = Результат.Выбрать();
    Пока Выборка.Следующий() Цикл // если в выборке - несколько - то
выбираем первый попавшийся
        Договор = Выборка.Договор;
        Прервать;
    КонецЦикла;

    Если Договор = Неопределено Тогда
        ТипЦен = Константы.ОсновнойТипЦен.Получить(); // по
умолчанию
        ТипСкидки = Константы.ОсновнойТипСкидок.Получить(); //
по умолчанию
    Иначе
        ТипЦен = Договор.ТипЦен; // из договора
        ТипСкидки = Договор.ТипСкидки; // из договора

```

```
КонецЕсли;  
Скидка = СкидкаПоТипуСкидок(НаДату, ТипСкидки, ДляКонтрагента); //  
значение скидки
```

```
Возврат Договор;
```

```
КонецФункции
```

```
// получить основной банковский счет контрагента/организации  
Функция ПолучитьОсновнойСчет(ДляКонтрагента) Экспорт
```

```
    // запросом выбираем все договора этого контрагента с признаком  
Основной и действующие по дате
```

```
    Запрос = Новый Запрос;
```

```
    Запрос.Текст =
```

```
    "ВЫБРАТЬ
```

```
    | Счета.Ссылка                КАК Счет,  
    | Счета.Владелец             КАК Владелец,  
    | Счета.Основной             КАК Основной
```

```
    | ИЗ
```

```
    | Справочник.БанковскиеСчета КАК Счета  
    |   ГДЕ Владелец = &ДляКонтрагента И  
    |   Основной = ИСТИНА
```

```
    |";
```

```
    Запрос.УстановитьПараметр("ДляКонтрагента", ДляКонтрагента);
```

```
    Результат = Запрос.Выполнить();
```

```
    Выборка = Результат.Выбрать();
```

```
    Пока Выборка.Следующий() Цикл // если в выборке - несколько - то  
выбираем первый попавшийся
```

```
        Возврат Выборка.Счет;
```

```
    КонецЦикла;
```

```
    Возврат Неопределено;
```

```
КонецФункции
```

```
// получить тип цен из договора
```

```
Функция ПолучитьТипЦенИзДоговора(ДоговорСсылка) Экспорт
```

```
    Если Не ДоговорСсылка.Пустая() Тогда
```

```
        Возврат ДоговорСсылка.ТипЦен;
```

```
    КонецЕсли;
```

```
    Возврат Неопределено;
```

```
КонецФункции
```

```
// получить тип скидки из договора
```

```
Функция ПолучитьТипСкидкиИзДоговора(ДоговорСсылка) Экспорт
```

```
Если Не ДоговорСсылка.Пустая() Тогда
    Возврат ДоговорСсылка.ТипСкидки;
КонецЕсли;
Возврат Неопределено;
```

КонецФункции

```
// рассчитать значение скидки (в %) по типу скидки (постоянная) и объему
продаж контрагента (накопительная)
Функция СкидкаПоТипуСкидок(НаДату, ТипСкидки, Контрагент) Экспорт
```

```
    Если ТипСкидки.Пустая() Тогда
        Возврат 0;
    КонецЕсли;
    Скидка = ТипСкидки.ПостояннаяСкидка; // как минимум есть постоянная
    Если Не ТипСкидки.НакопительнаяСкидкаЕсть Тогда // если
накопительной нет - то конец
        Возврат Скидка;
    КонецЕсли;
    Если ТипСкидки.Условия.Количество() = 0 Тогда // если условия
накопительной не заданы - то конец
        Возврат Скидка;
    КонецЕсли;
    СуммаПродаж = ПолучитьСуммуПродаж(НаДату, ТипСкидки.ДнейНакопления,
Контрагент); // получить сумму продаж контрагента за период
    ТЗ = ТипСкидки.Условия.Выгрузить();
    ТЗ.Сортировать("Сумма");
    СкидкаНакопительная = 0;
    Для Каждого Строка Из ТЗ Цикл // найти , в какую строку условий
попадает сумма продаж
        Если СуммаПродаж >= Строка.Сумма Тогда
            СкидкаНакопительная = Строка.Процент;
        КонецЕсли;
    КонецЦикла;
    Возврат Скидка + СкидкаНакопительная;
```

КонецФункции

```
// рассчитать сумму продаж по контрагенту за период
Функция ПолучитьСуммуПродаж(ДатаОкончания, ДнейСчитать, ПоКонтрагенту)
```

```
    Запрос = Новый Запрос;
    Запрос.Текст =
"ВЫБРАТЬ
| ПродажиОбороты.Контрагент КАК Контрагент,
| ПродажиОбороты.СуммаОборот
| ИЗ
| РегистрНакопления.Продажи.Обороты(&Начало, &Окончание) КАК
ПродажиОбороты
| ГДЕ Контрагент = &ПоКонтрагенту
```

```

| СГРУППИРОВАТЬ ПО
| ПродажиОбороты.Контрагент,
| ПродажиОбороты.СуммаОборот
|";

Запрос.УстановитьПараметр("ПоКонтрагенту", ПоКонтрагенту);
Запрос.УстановитьПараметр("Начало",
НачалоДня(ДобавитьМесяц(ДатаОкончания, -(ДнейСчитать/30))) );
Запрос.УстановитьПараметр("Окончание", КонецДня(ДатаОкончания));

Результат = Запрос.Выполнить();
Выборка = Результат.Выбрать();
Пока Выборка.Следующий() Цикл
    Возврат Выборка.СуммаОборот;
КонецЦикла;

Возврат 0;

КонецФункции

// ПО НОМЕНКЛАТУРЕ
Функция ПолучитьОсновнуюЕдиницуОтгрузкиНоменклатуры(НоменклатураСсылка)
Экспорт

    Если Не НоменклатураСсылка.Пустая() Тогда
        Возврат НоменклатураСсылка.ОсновнаяЕдиницаОтгрузки;
    КонецЕсли;
    Возврат Неопределено;

КонецФункции

Функция ПолучитьКоэффициент(ЕдИзмеренияСсылка) Экспорт

    Если Не ЕдИзмеренияСсылка.Пустая() Тогда
        Возврат ЕдИзмеренияСсылка.Коэффициент;
    КонецЕсли;
    Возврат 0;

КонецФункции

// получить цену номенклатуры
Функция ПолучитьЦену(НаДату, ВыбНоменклатура, ВыбТипЦен) Экспорт

    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        | Цены.Номенклатура КАК Номенклатура,
        | Цены.ТипЦен КАК ТипЦен,
        | Цены.Цена КАК Цена
        |ИЗ
        | РегистрСведений.Цены.СрезПоследних(&ВыбДата) КАК Цены

```

```

|ГДЕ
|    ТипЦен          = &ВыбТипЦен И
|    Номенклатура = &ВыбНоменклатура
|";
Запрос.УстановитьПараметр("ВыбНоменклатура", ВыбНоменклатура);
Запрос.УстановитьПараметр("ВыбТипЦен", ВыбТипЦен);
Запрос.УстановитьПараметр("ВыбДата", НаДату);

    Результат = Запрос.Выполнить();
Выборка = Результат.Выбрать();
Пока Выборка.Следующий() Цикл
    Возврат Выборка.Цена;
КонецЦикла;

    Возврат 0;

```

КонецФункции

```

// заполнить остатки номенклатуры
// в ТЗНоменклатура передается список запрашиваемой номенклатуры
// в ней же заполняется столбец "Количество"
Процедура    ПолучитьОстаткиНоменклатуры(НаДату,    ВыбФирма,    ВыбСклад,
ТЗНоменклатура) Экспорт

```

```

    СЗНоменклатура = Новый СписокЗначений;
    Для Каждого ТекНоменклатура Из ТЗНоменклатура Цикл
        СЗНоменклатура.Добавить(ТекНоменклатура.Номенклатура);
    КонецЦикла;

```

```

    Запрос = Новый Запрос;
    Запрос.Текст =
        "ВЫБРАТЬ
        |    Остатки.Номенклатура      КАК Номенклатура,
        |    Остатки.Фирма              КАК Фирма,
        |    Остатки.Склад              КАК Склад,
        |    Остатки.КоличествоОстаток    КАК Количество
        |ИЗ
        |    РегистрНакопления.Остатки.Остатки(&НаДату) КАК Остатки
        |ГДЕ
        |    Номенклатура В (&СЗНоменклатура) И
        |    Фирма = &ВыбФирма И
        |    Склад = &ВыбСклад
        |";

```

```

    Запрос.УстановитьПараметр("СЗНоменклатура", СЗНоменклатура);
    Запрос.УстановитьПараметр("ВыбФирма", ВыбФирма);
    Запрос.УстановитьПараметр("ВыбСклад", ВыбСклад);
    Запрос.УстановитьПараметр("НаДату", НаДату);

```

```

    Результат = Запрос.Выполнить();

```

```

Выборка = Результат.Выбрать();
Пока Выборка.Следующий() Цикл
    СтрокаТЗ = ТЗНоменклатура.Найти(Выборка.Номенклатура,
"Номенклатура");
    Если СтрокаТЗ <> Неопределено Тогда
        СтрокаТЗ.Количество = Выборка.Количество;
    КонецЕсли;
КонецЦикла;

```

КонецПроцедуры

Справочник «Контрагенты»

```

Процедура ПриУстановкеНовогоКода(СтандартнаяОбработка, Префикс)
    Префикс = Обмен.ПолучитьПрефиксНомера();
КонецПроцедуры

```

```

Процедура ОбработкаЗаполнения(ДанныеЗаполнения, СтандартнаяОбработка)
    // ввод на основании адреса доставки контрагента-основания
    Если ТипЗнч(ДанныеЗаполнения) = Тип("СправочникСсылка.Контрагенты")
    Тогда
        // это создается адрес доставки для головного
контрагента-основания
        ЮрФизЛицо = Перечисления.ЮрФизЛицо.АдресДоставки;
        ГоловнойКонтрагент = ДанныеЗаполнения.Ссылка;
        Родитель =
Константы.ПапкаАдресовДоставки.Получить();
        // некоторые реквизиты берем из головного контрагента
        Наименование = ДанныеЗаполнения.Наименование;
        НаименованиеПолное = ДанныеЗаполнения.НаименованиеПолное;
        Категория = ДанныеЗаполнения.Категория;
        Отвественный = ДанныеЗаполнения.Отвественный;
        ЮрАдрес = ДанныеЗаполнения.ЮрАдрес;
        ФактАдрес = ДанныеЗаполнения.ФактАдрес;
        ИНН = ДанныеЗаполнения.ИНН;
        КПП = ДанныеЗаполнения.КПП;
        ОКПО = ДанныеЗаполнения.ОКПО;

        // ?? надо
        ОГРН = ДанныеЗаполнения.ОГРН;
        ОКВЭД =
ДанныеЗаполнения.ОКВЭД;
        СвидетельствоСерияНомер =
ДанныеЗаполнения.СвидетельствоСерияНомер;
        СвидетельствоДатаВыдачи =
ДанныеЗаполнения.СвидетельствоДатаВыдачи;
        СтранаРегистрации =
ДанныеЗаполнения.СтранаРегистрации;
        Фамилия = ДанныеЗаполнения.Фамилия;

```

```

Имя = ДанныеЗаполнения.Имя;
Отчество = ДанныеЗаполнения.Отчество;
ДокументУдостоверяющийЛичность =
ДанныеЗаполнения.ДокументУдостоверяющийЛичность;
ДатаРождения =
ДанныеЗаполнения.ДатаРождения;
МестоРождения =
ДанныеЗаполнения.МестоРождения;
ПФР = ДанныеЗаполнения.ПФР;
Комментарий =
ДанныеЗаполнения.Комментарий;

```

```

// контакты берем из головного контрагента
Для Каждого ТекСтрокаТЧКонтакты Из ДанныеЗаполнения.ТЧКонтакты
Цикл
    НоваяСтрока = ТЧКонтакты.Добавить();
    НоваяСтрока.Комментарий =
ТекСтрокаТЧКонтакты.Комментарий;
    НоваяСтрока.Контакт = ТекСтрокаТЧКонтакты.Контакт;
    НоваяСтрока.Основной = ТекСтрокаТЧКонтакты.Основной;
    НоваяСтрока.ТипКонтакта =
ТекСтрокаТЧКонтакты.ТипКонтакта;
    КонецЦикла;
КонецЕсли;
КонецПроцедуры

```

Документ «Заявка»

Функция ПриПроведении()

```

Если Фирма.Пустая() Тогда
    Сообщить("Зполните организацию");
    Возврат Ложь;
КонецЕсли;
Если Контрагент.Пустая() Тогда
    Сообщить("Зполните контрагента");
    Возврат Ложь;
КонецЕсли;

Если Номенклатура.Количество() = 0 Тогда
    Сообщить("Зполните список товаров");
    Возврат Ложь;
КонецЕсли;

Возврат Истина;

```

КонецФункции

Процедура ОбработкаПроведения(Отказ, Режим)

```
Если Не ПриПроведении() Тогда // проверить возможность проведения
    Отказ = Истина;
    Возврат;
КонецЕсли;
```

КонецПроцедуры

Процедура ОбработкаЗаполнения(Основание, Значение)

```
// Если вводится новый из интерфейса
Если Основание = Неопределено Тогда
    Фирма = Константы.ОсновнаяФирма.Получить();
    Контрагент = Константы.ОсновнойКонтрагент.Получить();
    Сотрудник = Контрагент.Отвественный;
// ответственный сотрудник
    Договор = ОбщийСрв.ПолучитьОсновнойДоговор(Контрагент,
Дата, ТипЦен, ТипСкидки, Скидка); // основной договор, цены, скидки
КонецЕсли;

// Если вводится на основании документа Задание
Если ТипЗнч(Основание) = Тип("ДокументСсылка.Задание") Тогда
    Фирма = Основание.Фирма;
    Контрагент = Основание.Контрагент; //
основание - это контрагент
    Сотрудник = Контрагент.Отвественный; // из контрагента
    Договор = ОбщийСрв.ПолучитьОсновнойДоговор(Контрагент,
Дата, ТипЦен, ТипСкидки, Скидка); // основной договор, цены, скидки
КонецЕсли;

// Если вводится на основании Контрагента
Если ТипЗнч(Основание) = Тип("СправочникСсылка.Контрагенты") Тогда
    Фирма = Константы.ОсновнаяФирма.Получить(); // по
умолчанию
    Склад = Константы.ОсновнойСклад.Получить(); // по
умолчанию
    Контрагент = Основание; //
основание - это контрагент
    Сотрудник = Контрагент.Отвественный; // из контрагента
    Договор = ОбщийСрв.ПолучитьОсновнойДоговор(Контрагент,
Дата, ТипЦен, ТипСкидки, Скидка); // основной договор, цены, скидки
КонецЕсли;

Если ТипЗнч(Основание) = Тип("Структура") Тогда //если вводится
новый из списка с установленным отбором
    // то данные заполняются автоматически из отбора "Заполнять из
данных заполнения"
```

КонецЕсли

КонецПроцедуры

Процедура ПередЗаписью()

СуммаДокумента = Номенклатура.Итог("Сумма");

КонецПроцедуры

// изменился контрагент - изменить договор, тип цен, тип скидки, скидку и пересчитать цены

&НаКлиенте

Процедура КонтрагентПриИзменении(Элемент)

Объект.Сотрудник

=

ОбщийСрв.ПолучитьОтветственного(Объект.Контрагент);

Объект.Договор

=

ОбщийСрв.ПолучитьОсновнойДоговор(Объект.Контрагент, Объект.Дата);

ДоговорПриИзменении(Элемент);

КонецПроцедуры

// изменился договор - изменить тип цен, тип скидки, скидку и пересчитать цену

&НаКлиенте

Процедура ДоговорПриИзменении(Элемент)

Если Не Объект.Договор.Пустая() Тогда

Объект.ТипЦен =

ОбщийСрв.ПолучитьТипЦенИзДоговора(Объект.Договор);

Объект.ТипСкидки

=

ОбщийСрв.ПолучитьТипСкидкиИзДоговора(Объект.Договор);

Иначе

Объект.ТипЦен = ОбщийСрв.ПолучитьОсновнойТипЦен();

Объект.ТипСкидки = ОбщийСрв.ПолучитьОсновнойТипСкидки();

КонецЕсли;

Объект.Скидка = ОбщийСрв.СкидкаПоТипуСкидок(Объект.Дата,

Объект.ТипСкидки, Объект.Контрагент);

ПересчитатьЦены();

КонецПроцедуры

// изменился тип цен - пересчитать цены (скидка не изменилась)

&НаКлиенте

Процедура ТипЦенПриИзменении(Элемент)

ПересчитатьЦены();

КонецПроцедуры

// изменился тип скидки - пересчитать скидку и пересчитать цены

&НаКлиенте

Процедура ТипСкидкиПриИзменении(Элемент)

```
Объект.Скидка = ОбщийСрв.СкидкаПоТипуСкидок(Объект.Дата,  
Объект.ТипСкидки, Объект.Контрагент);  
ПересчитатьЦены();  
КонецПроцедуры
```

```
// изменилась скидка вручную - персчитать цены  
&НаКлиенте  
Процедура СкидкаПриИзменении(Элемент)
```

```
ПересчитатьЦены();  
КонецПроцедуры
```

```
Процедура ПересчитатьЦены()
```

```
Для Каждого ТекНоменклатура Из Объект.Номенклатура Цикл  
ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,  
ТекНоменклатура.Номенклатура,
```

```
ТекНоменклатура.Количество,
```

```
ТекНоменклатура.Коэфф,
```

```
Объект.ТипЦен,
```

```
Объект.Скидка,
```

```
ТекНоменклатура.Цена,
```

```
ТекНоменклатура.ЦенаСкидка,
```

```
ТекНоменклатура.Сумма,
```

```
ТекНоменклатура.СуммаСкидки);
```

```
КонецЦикла;
```

```
КонецПроцедуры
```

```
&НаКлиенте
```

```
Процедура НоменклатураНоменклатураПриИзменении(Элемент)
```

```
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество = 1;
```

```
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф = 1;  
ОсновнаяЕдиницаОтгрузки =
```

```
ОбщийСрв.ПолучитьОсновнуюЕдиницуОтгрузкиНоменклатуры(ЭтотОбъект.Элементы.  
Номенклатура.ТекущиеДанные.Номенклатура);
```

```
Если Не ОсновнаяЕдиницаОтгрузки.Пустая() Тогда
```

```
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЕдИзмерения =  
ОсновнаяЕдиницаОтгрузки;
```

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф =  
ОбщийСрв.ПолучитьКоэффициент(ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанн  
ые.ЕдИзмерения);  
КонецЕсли;

ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,  
Объект.ТипЦен,  
Объект.Скидка,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЦенаСкидка,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.СуммаСкидки);

КонецПроцедуры

&НаКлиенте

Процедура НоменклатураКоличествоПриИзменении(Элемент)

ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,  
Объект.ТипЦен,  
Объект.Скидка,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЦенаСкидка,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.СуммаСкидки);

КонецПроцедуры

&НаКлиенте

Процедура НоменклатураЕдИзмеренияПриИзменении(Элемент)

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф  
ОбщийСрв.ПолучитьКоэффициент(ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанн  
ые.ЕдИзмерения);

ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,

Объект.ТипЦен,

Объект.Скидка,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЦенаСкидка,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.СуммаСкидки);

КонецПроцедуры

Документ «Реализация»

Функция ПриПроведении()

Если Фирма.Пустая() Тогда  
Сообщить("Зполните организацию");  
Возврат Ложь;

КонецЕсли;

Если Контрагент.Пустая() Тогда  
Сообщить("Зполните контрагента");  
Возврат Ложь;

КонецЕсли;

Если Склад.Пустая() Тогда  
Сообщить("Зполните склад");  
Возврат Ложь;

КонецЕсли;

Если Номенклатура.Количество() = 0 Тогда  
Сообщить("Зполните список товаров");  
Возврат Ложь;

КонецЕсли;

Возврат Истина;

КонецФункции

Функция ПроверитьОстатки()

// если в таб части несколько строк с одной и той же номенклатурой/партией, то "простая" проверка не  
// будет правильно проверять остатки. Необходимо свернуть одинаковую номенклатуру/партии

ТЗОстатки = Новый ТаблицаЗначений;  
ТипНоменклатура =  
ОбщийСрв.СоздатьОписаниеТипов("СправочникСсылка.Номенклатура");  
ТЗОстатки.Колонки.Добавить("Номенклатура", ТипНоменклатура);  
ТипКоличество = ОбщийСрв.СоздатьОписаниеТипов("Число", 15, 2);  
ТЗОстатки.Колонки.Добавить("Количество", ТипКоличество);

Для Каждого ТекСтрокаНоменклатура Из Номенклатура Цикл  
НоваяСтрока = ТЗОстатки.Добавить();  
НоваяСтрока.Номенклатура = ТекСтрокаНоменклатура.Номенклатура;  
НоваяСтрока.Количество = 0;  
КонецЦикла;  
ТЗОстатки.Свернуть("Номенклатура", "Количество");  
ОбщийСрв.ПолучитьОстаткиНоменклатуры(Дата, Фирма, Склад,  
ТЗОстатки); // остстки запросом

ТЗЗапрашиваемаяНоменклатура = Новый ТаблицаЗначений;  
ТЗЗапрашиваемаяНоменклатура.Колонки.Добавить("Номенклатура",  
ТипНоменклатура);  
ТЗЗапрашиваемаяНоменклатура.Колонки.Добавить("Количество",  
ТипКоличество);  
Для Каждого ТекСтрокаНоменклатура Из Номенклатура Цикл  
НоваяСтрока = ТЗЗапрашиваемаяНоменклатура.Добавить();  
НоваяСтрока.Номенклатура = ТекСтрокаНоменклатура.Номенклатура;  
НоваяСтрока.Количество =  
ТекСтрокаНоменклатура.Количество\*ТекСтрокаНоменклатура.Коэфф;  
КонецЦикла;  
ТЗЗапрашиваемаяНоменклатура.Свернуть("Номенклатура", "Количество");

Для Каждого ТекЗапрашиваемыйОстаток Из ТЗЗапрашиваемаяНоменклатура  
Цикл  
Остаток =  
ТЗОстатки.Найти(ТекЗапрашиваемыйОстаток.Номенклатура, "Номенклатура");  
Если Остаток = Неопределено Тогда  
Сообщить("По номенклатуре " +  
ТекЗапрашиваемыйОстаток.Номенклатура + " запрашиваемое количество меньше  
имеющегося остатка");  
Возврат Ложь;  
КонецЕсли;

```

        Если Остаток.Количество < ТекЗапрашиваемыйОстаток.Количество
Тогда
        Сообщить("По номенклатуре " +
ТекЗапрашиваемыйОстаток.Номенклатура + " запрашиваемое количество меньше
имеющегося остатка");
        Возврат Ложь;
        КонецЕсли;
КонецЦикла;

        Возврат Истина
КонецФункции

```

Процедура ОбработкаПроведения(Отказ, Режим)

```

        Если Не ПриПроведении() Тогда // проверить возможность проведения
            Отказ = Истина;
            Возврат;
        КонецЕсли;

        Если Склад.Контроль Тогда
            Если Не ПроверитьОстатки() Тогда // контроль остатков
                Отказ = Истина;
                Возврат;
            КонецЕсли;
        КонецЕсли;

        // регистр остатков Остатки
        Для Каждого ТекСтрокаНоменклатура Из Номенклатура Цикл
            Движение = Движения.Остатки.Добавить();
            Движение.ВидДвижения = ВидДвиженияНакопления.Расход;
            Движение.Период = Дата;
            Движение.Фирма = Фирма;
            Движение.Склад = Склад;
            Движение.КодОперации = КодОперации;
            Движение.Номенклатура = ТекСтрокаНоменклатура.Номенклатура;
            Движение.Партия = ТекСтрокаНоменклатура.Партия;

            Движение.Количество =
ТекСтрокаНоменклатура.Количество*ТекСтрокаНоменклатура.Коэфф;
        КонецЦикла;

        // регистр остатков Взаиморасчеты
        Если Счет.Пустая() Тогда // если в документе указан счет-касса
            Движение = Движения.Взаиморасчеты.Добавить();
            Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
            Движение.Период = Дата;
            Движение.Фирма = Фирма;
            Движение.Контрагент = Контрагент;
            Движение.Сделка = ЭтотОбъект.Ссылка;
            Движение.Сумма = Номенклатура.Итог("Сумма");
        КонецЕсли;

```

```

// регистр остатков Деньги
Если Не Счет.Пустая() Тогда // если в документе указан счет-касса
    Движение = Движения.Деньги.Добавить();
    Движение.ВидДвижения = ВидДвиженияНакопления.Приход;
    Движение.Период = Дата;
    Движение.Фирма = Фирма;
    Движение.Счет = Счет;
    Движение.КодОперации = КодОперации;
    Движение.Сумма = Номенклатура.Итог("Сумма");
КонецЕсли;

```

```

// регистр оборотов Продажи
Для Каждого ТекСтрокаНоменклатура Из Номенклатура Цикл
    Движение = Движения.Продажи.Добавить();
    Движение.Период = Дата;
    Движение.Фирма = Фирма;
    Движение.Контрагент = Контрагент;
    Движение.Сотрудник = Сотрудник;
    Движение.КодОперации = КодОперации;
    Движение.Номенклатура =
ТекСтрокаНоменклатура.Номенклатура;
    Движение.Количество =
ТекСтрокаНоменклатура.Количество*ТекСтрокаНоменклатура.Коефф;
    Движение.Сумма =
ТекСтрокаНоменклатура.Сумма;
КонецЦикла;

```

КонецПроцедуры

Процедура ОбработкаЗаполнения(Основание, Значение)

```

// Если вводится новый из интерфейса
Если Основание = Неопределено Тогда
    Фирма = Константы.ОсновнаяФирма.Получить();
    Склад = Константы.ОсновнойСклад.Получить();
    Контрагент = Константы.ОсновнойКонтрагент.Получить();
    Сотрудник = Контрагент.Отвественный;
// ответственный сотрудник
    Договор = ОбщийСрв.ПолучитьОсновнойДоговор(Контрагент,
Дата, ТипЦен, ТипСкидки, Скидка); // основной договор, цены, скидки
    Счет = Константы.ОсновнойСчет.Получить();
КонецЕсли;

// Если вводится на основании Заявки
Если ТипЗнч(Основание) = Тип("ДокументСсылка.Заявка") Тогда
    // Заполнение шапки
    ДокОснование = Основание; // документ-
основание
    Фирма = Основание.Фирма; // из основания

```

```

Склад = Константы.ОсновнойСклад.Получить();

Контрагент = Основание.Контрагент; // из
основания
Сотрудник = Контрагент.Отвественный; // из контрагента
Договор =
ОбщийСрв.ПолучитьОсновнойДоговор(Контрагент, Дата, ТипЦен, ТипСкидки,
Скидка); // основной договор, цены, скидки
Счет = Константы.ОсновнойСчет.Получить();
// табличная часть
Для Каждого ТекСтрокаНоменклатура Из Основание.Номенклатура
Цикл // по всем записям табличной части
НоваяСтрока = Номенклатура.Добавить();
НоваяСтрока.Номенклатура =
ТекСтрокаНоменклатура.Номенклатура;
НоваяСтрока.Количество =
ТекСтрокаНоменклатура.Количество;
НоваяСтрока.ЕдИзмерения =
ТекСтрокаНоменклатура.ЕдИзмерения;
НоваяСтрока.Коэфф =
ТекСтрокаНоменклатура.Коэфф;

ОбщийСрв.РассчитатьСуммуСтроки(ТекущаяДата(),
ТекСтрокаНоменклатура.Номенклатура,
ТекСтрокаНоменклатура.Количество,
ТекСтрокаНоменклатура.Коэфф,
ТипЦен,
Скидка,
НоваяСтрока.Цена,
НоваяСтрока.ЦенаСкидка,
НоваяСтрока.Сумма,
НоваяСтрока.СуммаСкидки);
КонецЦикла;
КонецЕсли;

// Если вводится на основании Контрагента
Если ТипЗнч(Основание) = Тип("СправочникСсылка.Контрагенты") Тогда
Фирма = Константы.ОсновнаяФирма.Получить(); // по
умолчанию
Склад = Константы.ОсновнойСклад.Получить(); // по
умолчанию
Контрагент = Основание; //
основание - это контрагент

```

```
Сотрудник = Контрагент.Отвественный; // из контрагента
Договор = ОбщийСрв.ПолучитьОсновнойДоговор(Контрагент,
Дата, ТипЦен, ТипСкидки, Скидка); // основной договор, цены, скидки
Счет = Константы.ОсновнойСчет.Получить();
КонецЕсли;
```

```
//Если вводится новый из списка с установленным отбором
Если ТипЗнч(Основание) = Тип("Структура") Тогда
// то данные заполняются автоматически из отбора "Заполнять из
данных заполнения"
КонецЕсли
```

КонецПроцедуры

```
Процедура ПередЗаписью()
СуммаДокумента = Номенклатура.Итог("Сумма");
КонецПроцедуры
```

```
// изменился контрагент - изменить договор, тип цен, тип скидки, скидку и
пересчитать цены
&НаКлиенте
Процедура КонтрагентПриИзменении(Элемент)
```

```
Объект.Сотрудник =
ОбщийСрв.ПолучитьОтвественного(Объект.Контрагент);
Объект.Договор =
ОбщийСрв.ПолучитьОсновнойДоговор(Объект.Контрагент, Объект.Дата);
//Если Не Объект.Договор = Неопределено Тогда
ДоговорПриИзменении(Элемент);
//КонецЕсли;
КонецПроцедуры
```

```
// изменился договор - изменить тип цен, тип скидки, скидку и пересчитать
цену
&НаКлиенте
Процедура ДоговорПриИзменении(Элемент)
```

```
Если Не Объект.Договор.Пустая() Тогда
Объект.ТипЦен =
ОбщийСрв.ПолучитьТипЦенИзДоговора(Объект.Договор);
Объект.ТипСкидки =
ОбщийСрв.ПолучитьТипСкидкиИзДоговора(Объект.Договор);
Иначе
Объект.ТипЦен = ОбщийСрв.ПолучитьОсновнойТипЦен();
Объект.ТипСкидки = ОбщийСрв.ПолучитьОсновнойТипСкидки();
КонецЕсли;
Объект.Скидка =
ОбщийСрв.СкидкаПоТипуСкидок(Объект.Дата,
Объект.ТипСкидки, Объект.Контрагент);
ПересчитатьЦены();
```

КонецПроцедуры

```
// изменился тип цен - пересчитать цены (скидка не изменилась)
&НаКлиенте
Процедура ТипЦенПриИзменении(Элемент)
```

```
    ПересчитатьЦены();
КонецПроцедуры
```

```
// изменился тип скидки - пересчитать скидку и пересчитать цены
&НаКлиенте
Процедура ТипСкидкиПриИзменении(Элемент)
```

```
    Объект.Скидка      =      ОбщийСрв.СкидкаПоТипуСкидок(Объект.Дата,
Объект.ТипСкидки, Объект.Контрагент);
    ПересчитатьЦены();
КонецПроцедуры
```

```
// изменилась скидка вручную - пересчитать цены
&НаКлиенте
Процедура СкидкаПриИзменении(Элемент)
```

```
    ПересчитатьЦены();
КонецПроцедуры
```

```
Процедура ПересчитатьЦены()
```

```
    Для Каждого ТекНоменклатура Из Объект.Номенклатура Цикл
        ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,
ТекНоменклатура.Номенклатура,
```

```
        ТекНоменклатура.Количество,
```

```
        ТекНоменклатура.Коэфф,
```

```
        Объект.ТипЦен,
```

```
        Объект.Скидка,
```

```
        ТекНоменклатура.Цена,
```

```
        ТекНоменклатура.ЦенаСкидка,
```

```
        ТекНоменклатура.Сумма,
```

```
        ТекНоменклатура.СуммаСкидки);
```

```
    КонецЦикла;
```

```
КонецПроцедуры
```

```
&НаКлиенте
```

```
Процедура НоменклатураНоменклатураПриИзменении(Элемент)
```

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество = 1;  
 ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф = 1;  
 ОсновнаяЕдиницаОтгрузки =  
 ОбщийСрв.ПолучитьОсновнуюЕдиницуОтгрузкиНоменклатуры(ЭтотОбъект.Элементы.  
 Номенклатура.ТекущиеДанные.Номенклатура);  
 Если Не ОсновнаяЕдиницаОтгрузки.Пустая() Тогда  
 ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЕдИзмерения =  
 ОсновнаяЕдиницаОтгрузки;  
 ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф =  
 ОбщийСрв.ПолучитьКоэффициент(ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанн  
 ые.ЕдИзмерения);  
 КонецЕсли;

ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,  
 ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,  
 ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,  
 ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,  
 Объект.ТипЦен,  
 Объект.Скидка,  
 ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,  
 ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЦенаСкидка,  
 ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма,  
 ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.СуммаСкидки);

КонецПроцедуры

&НаКлиенте  
 Процедура НоменклатураКоличествоПриИзменении(Элемент)

ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,  
 ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,  
 ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,  
 ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,  
 Объект.ТипЦен,  
 Объект.Скидка,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЦенаСкидка,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.СуммаСкидки);

КонецПроцедуры

&НаКлиенте

Процедура НоменклатураЕдИзмеренияПриИзменении(Элемент)

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф =  
ОбщийСрв.ПолучитьКоэффициент(ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанн  
ые.ЕдИзмерения);

ОбщийСрв.РассчитатьСуммуСтроки(Объект.Дата,  
ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Номенклатура,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Количество,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Коэфф,

Объект.ТипЦен,

Объект.Скидка,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Цена,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.ЦенаСкидка,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.Сумма,

ЭтотОбъект.Элементы.Номенклатура.ТекущиеДанные.СуммаСкидки);

КонецПроцедуры

Министерство науки и высшего образования РФ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»  
Институт космических и информационных технологий

Кафедра вычислительной техники

УТВЕРЖДАЮ  
Заведующий кафедрой  
\_\_\_\_\_ О.В. Непомнящий  
« \_\_\_\_ » \_\_\_\_\_ 2024 г.

**МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ**

Комплексная автоматизация торговой и складской деятельности  
предприятия

090401 Информатика и вычислительная техника

09.04.01.11 «Вычислительные системы и сети»

Руководитель	<u>Н. Сиротина</u> подпись	<u>19.06.2024</u> дата	доцент каф. ВТ, канд. техн. наук должность, ученая степень	Н.Ю. Сиротина
Выпускник	<u>Шереметов</u> подпись	<u>19.06.2024</u> дата		В.Н. Шереметов
Рецензент	<u>А.А. Мазуров</u> подпись	<u>19.06.2024</u> дата	доцент каф. СИИ, канд. техн. наук должность, ученая степень	А.А. Мазуров
Нормоконтролер	<u>Н. Сиротина</u> подпись	<u>19.06.2024</u> дата		Н.Ю. Сиротина

Красноярск 2024