

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**  
( Н И У « Б е л Г У » )

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК  
КАФЕДРА МАТЕМАТИЧЕСКОГО И ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ  
ИНФОРМАЦИОННЫХ СИСТЕМ

**ПРОЕКТИРОВАНИЕ И РЕАЛИЗАЦИЯ CRM ПРИЛОЖЕНИЯ ДЛЯ  
МАГАЗИНА «МИР ДЕРЕВА»**

Выпускная квалификационная работа  
обучающегося по направлению подготовки  
02.03.02. Фундаментальная информатика и информационные технологии  
очной формы обучения, группы 07001301  
Бабенко Александра Адреевича

Научный руководитель  
к.т.н., доцент  
Бурданова Е. В.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛАВА 1. АНАЛИЗ ДЕЯТЕЛЬНОСТИ МАГАЗИНА «МИР ДЕРЕВА».....	5
1.1 Описание деятельности магазина.....	5
1.2 Организационная структура управления предприятием .....	8
1.3 Анализ современных информационных CRM-систем.....	15
1.4 Постановка цели и задач для реализации CRM – приложения «Мир Дерева».....	22
ГЛАВА 2. ПРОЕКТИРОВАНИЕ CRM - СИСТЕМЫ .....	24
2.1 Инфологическое проектирование .....	24
2.2 Даталогическое проектирование .....	28
2.3 Выбор программных средств разработки.....	29
2.4 Проектирование пользовательского интерфейса CRM-системы.....	37
ГЛАВА 3. РАЗРАБОТКА CRM- ПРИЛОЖЕНИЯ «МИР ДЕРЕВА» .....	43
3.1 Разработка базы данных .....	43
3.2 Програмная реализация автоматизированной системы.....	47
3.3 Тестирование контрольного примера .....	53
ЗАКЛЮЧЕНИЕ .....	61
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ .....	62
ПРИЛОЖЕНИЕ .....	66

## ВВЕДЕНИЕ

В современном мире прогрессирует такой вид деятельности как закупка и продажа пиломатериалов. Данный вид деятельности ни когда не будет в убытке, в силу того, что дерево является главным материалом в такой отрасли как строительство, которое, как известно из года в год только прогрессирует и растёт в своём объёме. Одним из таких примеров выступает магазин «Мир Дерева». Главной деятельностью, которого является закупка и продажа пиломатериала у зарубежных поставщиков.

Предметом данной выпускной квалификационной работы является способы проектирования и разработки CRM-приложения для организации «Мир Дерева». Объектом – работа сотрудников магазина «Мир Дерева».

В настоящее время существует ряд CRM-систем для управления бизнес-процессами предприятия, однако их функционал не всегда подходит для определенной специализации предприятия. Именно этим и обуславливается актуальность выбранной темы выпускной квалификационной работы.

Целью данной выпускной квалификационной работы является автоматизация бизнес - процессов учёта закупок и продаж товара при помощи внедрения в организацию CRM – приложения.

Для реализации поставленной цели необходимо решение следующих задач:

- подробное изучение предметной области, а именно деятельности магазина «Мир Дерева»;
- проведение сравнительного анализа современных информационных систем, выявление их недочетов для разработки специализированного приложения «Мир Дерева»;

- проектирование и разработка базы данных для CRM-приложения «Мир Дерева»;
- проектирование пользовательского интерфейса для разрабатываемого программного продукта;
- тестирование контрольного примера разработанного CRM – приложения.

Структура выпускной квалификационной работы состоит из введения, 3 глав, заключения, списка используемой литературы и приложения.

В первой главе проводится анализ предметной области, рассматривается организационная структура управления предприятием, производится анализ современных информационных систем, а также ставятся цели и задачи для реализации программного продукта.

Во второй главе проектируется CRM-приложение, логическая и физическая базы данных, производится выбор программных средств для разработки, разрабатывается пользовательский интерфейс информационной системы.

В третьей главе осуществляется разработка CRM-приложения и производится тестирование разработанного CRM-приложения «Мир Дерева».

Выпускная квалификационная работа включает в себя 61 страницу, 32 рисунка, 1 таблицу, список используемой литературы и приложение.

# ГЛАВА 1. АНАЛИЗ ДЕЯТЕЛЬНОСТИ МАГАЗИНА «МИР ДЕРЕВА»

## 1.1 Описание деятельности магазина

Предприятие «Мир Дерева» зарегистрировано по адресу Белгородская область, Борисовский район, посёлок Борисовка, улица Ленина 31 в 2012 году. На данной территории располагается магазин, складское помещение и навес. Площадь, занимаемая магазином, составляет 97 метров квадратных. Из них торговая площадь магазина -12 метров квадратных, а складские помещения – 48 метров квадратных. Навес расположен рядом с магазином, его площадь – 157 квадратных метров.

Организационно правовая форма «Мир Дерева» – индивидуальный предприниматель. Форма собственности – частная, организаторами являются физические лица. В магазине работает 13 человек.

Основным направлением деятельности магазина «Мир Дерева» является закупка дерева (доска обрезная и полуобрезная) у зарубежных поставщиков и дальнейшей его реализации с определённой наценкой.

В настоящий момент магазин имеет двух зарубежных поставщиков и одного российского. В целях расширения бизнеса ведётся поиск новых поставщиков и оптовых точек сбыта.

Основными породами закупаемого дерева являются: сосна, ель, липа, дуб и лиственница [14].

В 2013 году магазин закупил распилочный станок Ц2Д-5А, в результате чего на территории магазина осуществляется распил дерева. Для данного процесса подбирают сухой пиломатериал, влажность которого должна составлять 10 ( $\pm 2$ ) %, что соответствует всем нормам ГОСТа. Этот процесс необходим для того чтобы после распила не произошла деформация,

получившегося товара. Распил производят под навесом на ленточном станке, в результате чего получается пиломатериал (рейка, брус) [9].

В дальнейшем часть пиломатериала выбраковывается, а остальная его часть скочуется в пачки по четыре штуки и поступает в продажу, либо в складское помещение. А весь выбракованный пиломатериал уценяется и также поступает в продажу. При не соответствии товара ГОСТу он идёт на переработку [2].

В магазине «Мир Дерева» имеется в продаже следующие пиломатериалы [6]:

- доска обрезная;
- доска полуобрезная;
- доска строганая;
- блокхаус;
- имитация бруса;
- евровагонка;
- вагонка липа;
- доска пола;
- щит мебельный;
- площадка;
- подоконник;
- брус;
- рейка;
- раскладка;
- наличник;
- плинтус;
- уголок;
- штапик.

В 2014 году в магазин был приобретен грузовой автомобиль для совершения доставок товара клиенту, а так же набран штат новых

сотрудников. Благодаря покупке нового автомобиля, прибыль предприятия увеличилась вдвое.

Все расходы по обслуживанию машины оплачивает магазин, а мелкие ремонтные работы производит водитель, так как эти обязанности прописаны в его трудовом договоре. При любых серьёзных поломках водитель отгоняет машину в автосервис [16].

Цена доставки зависит от места нахождения клиента. Если же заказ очень большой, то предусмотрена бесплатная доставка. Возможно два варианта доставки: самовывоз и доставка от организации.

При осуществлении доставки от организации нужно учитывать то, что машина за один рейс может привезти не более пяти кубов дерева.

Реализация продукции в магазине осуществляется как в розницу, так в опт. Предприятие «Мир Дерева» сотрудничает:

- с государственными организациями - заключается ряд документов, таких как: договор, накладная (с соответствующими реквизитами организации), приложение к договору. Государственные организации производят оплату только переводом денежных средств на расчётный счёт магазина;

- с частными клиентами – оплата производится наличными при поступлении заказа к клиенту, либо при оформлении заказа.

Для оптовых покупателей предусмотрена система скидок, зависящая от общей суммы заказа. Оформление заказа возможно как на месте, так и под заказ, если данного вида пиломатериала нет в наличие.

При оформлении заказа клиент указывает, необходима ли выгрузка товара или нет, так как предприятие предоставляет грузчиков платно. Сумма выгрузки обговаривается при осуществлении заказа.

Поставки от зарубежных поставщиков осуществляют с помощью наёмного грузового транспорта, который может привезти 40 ( $\pm 2$ ) кубов дерева, это зависит от степени влажности. Данным способом поставляют такие виды пиломатериалов, как: доска обрезная и доска полуобрезная. Заказ

осуществляет по телефону, либо другими коммуникационными средствами связи (Skype или Viber), в целях экономии денежных средств. При заказе обсуждается: ассортимент заказа, количество каждой позиции, сумма доставки и итоговая стоимость всего пиломатериала. Поставки осуществляются примерно 5-7 раз в месяц в летний сезон торговли, а в зимний - 1-2 раза.

Когда осуществляется поставка дерева, директор магазина нанимает двух временных грузчиков для разгрузки грузового автомобиля. Во время выгрузки проверяют качество привезённого дерева, если качество дерева не удовлетворяет ГОСТу, то об этом сообщают поставщику, и обговаривают новые условия стоимости пиломатериала, либо осуществляется возврат. Доставку возвращенного товара оплачивает поставщик. После разгрузки и проверки качества, производится оплата денежным переводом на указанный расчетный счёт поставщика в договоре [19].

Так же поставки пиломатериалов производятся на грузовом автомобиле организации. При данном виде поставки, директор обговаривает все условия покупки, оценивает качество и ассортимент товара, а так же производит оплату наличными расчетом. Этим способом доставляются такие материалы как: евровагонка, доска пола, щит мебельный, подоконник, рейка половая, наличник и плинтус. Данный вид осуществляется один раз в месяц.

Проведя анализ деятельности магазина «Мир Дерева», стало понятно, что основное направление магазина заключается в закупке, распиле, продаже и доставке пиломатериала клиенту, а так же извлечение из всех перечисленных действий прибыли.

## **1.2 Организационная структура управления предприятием**

Предприятие «Мир Дерева» использует линейно-функциональную структуру управления.



На рис. 1.1 показана организационная структура управления предприятием «Мир Дерева».



Рис. 1.1. Организационная структура управления магазина «Мир Дерева»

Магазин «Мир Дерева» имеет пять отделов: отдел продаж, отдел логистики, бухгалтерия и хозяйственный отдел [17].

Владелец магазина по совместительству занимает должность директора. Директор является первым лицом предприятия и имеет все полномочия: лично контролирует работу каждого отдела и в конце каждого месяца просит отчёт о проделанной работе.

Отдел продаж занимается оформлением заказа, изучением договорных обязательств и следит за правильностью их выполнения. Этот отдел следит за планами поставок пиломатериалов клиентам по следующим критериям:

- объёму;
- качеству;
- условию ассортимента заказа;
- срокам;
- документации;
- итоговой суммой.

Данный отдел оповещает о состоянии складских запасов, интересующего пиломатериала, клиенту. После заказа данный отдел оповещает клиента, что к нему выехала машина с пиломатериалом [21].

Отдел закупок находит и налаживает связи с поставщиками дерева. Заключает соответствующий договор и оформляет заказ закупки по критериям:

- ассортимент;
- количеству;
- качеству;
- цене товара;
- дате привоза;
- цене за привоз.

После привоза пиломатериала этот отдел проверяет все критерии, если они соответствуют всем договорённостям, производится оплата. В случае несоответствия всем критериям товар, осуществляется возврат товара обратно поставщикам [22].

Отдел логистики выполняет следующие операции, связанные с выполнением следующих действий товародвижения на предприятии:

- заказ;
- приобретения;
- доставка;
- хранения;
- сортировка;
- отпуск товаров.

Бухгалтерия отвечает за учёт и отчётность предприятия в налоговую службу. Проводит ревизии и инвентаризацию на предприятии. Проверяет соблюдения всех финансовых дисциплин, а в случае недостачи денежных средств в кассе бухгалтер выясняет по бумагам у кого возникла недостача и просит работника возместить не достающую сумму. А так же бухгалтер занимается составление финансовых планов. И в конце каждого месяца

отчитывается у директора за свою проделанную работу и предоставляет план работы на следующий месяц.

Хозяйственный отдел включает в себя две функции: техническую и хозяйственную.

Техническая функция заключается в поддержании работы и обслуживании всех электробытовых приборов на предприятии. Осуществляется закупка всех нужных средств для обслуживания всех приборов. Хозяйственная функция отдела отвечает за хранение и выдачу всего хозяйственного инвентаря работникам, а в случае поломки приобретение нового. Данный отдел ведёт свой журнал всех приборов, которые находятся на складе и в магазине, и отсчитывается бухгалтеру обо всех потраченных средствах [15].

Организационная структура магазина «Мир Дерева» построена на специализированном разделении труда, то есть работа между людьми распределена не просто так, а закреплена за специалистами, которые могут выполнить свою работу лучше других работников. Такую структуру управления сделали для того чтобы все специалисты работали как одно целое.

При данной структуре управления сохраняется преимущество линейной структуры и функциональной структуры в виде специализации управления. В данной структуре есть несколько существенных недостатков:

- данная структура управления не позволяет внедрять всякого рода инноваций;
- всякие изменения требуют значительного пополнения персонала и больших затрат предприятия;
- в структуре централизованное управление, что вызывает перегрузку директора предприятия;
- появляются ограничения для подготовки ценных сотрудников руководителей отделов [28].

На предприятии действует несколько стандартов и правил:

- разделение труда по специализации;
- иерархия уровней управления;
- найм на работу осуществляется с некоторыми квалификационными требованиями в соответствии с должностью, на которую претендует потенциальный работник;
- условия увольнения расписаны в заключаемом трудовом договоре о найме на работу;
- существует система штрафных санкций, разработанная директором магазина;
- зарплата и аванс выплачивается каждый месяц в указанную дату в договоре.

Основные характеристики персонала предприятия являются: численность и структура.

Состав работников в магазине «Мир Дерева» не большая, составляющая 13 человек: директор, начальник отдела продаж, менеджер по оформлению документов, менеджер по закупкам два менеджера по продажам, специалист по логистике, бухгалтер, два грузчика, водитель и два сторожа.

Структура персонала магазина – объединение некоторых групп работников по ряду признаков и категорий [11].

Состав работников может быть представлен в зависимости от возраста и образования.

Возрастной состав персонала магазина – это соотношение работников по возрасту. Возрастная структура характеризуется частью соответствующих возрастов к общему количеству персонала.

Возрастной состав магазина «Мир Дерева», можно представить на диаграмме, рис. 1.2.



Рис. 1.2. Возрастной состав магазина «Мир Дерева»

Состав персонала по уровню образования можно разделить:

- высшее;
- среднее специальное;
- среднее общее образование.

Для структуры образования работников в магазине была составлена диаграмма, представленная на рис. 1.3.



Рис. 1.3. Состав персонала магазина «Мир Дерева » по уровню образования

Таким образом, анализ работников магазина «Мир Дерева», показала, что основная часть работников имеет среднее специальное образование, а возраст составляет в пределах 36-50 лет. Это свидетельствует о стабильной работе кадров [34].

Для улучшения работоспособности персонала в магазине используют мотивацию в виде денежных премий.

В магазине «Мир Дерева» используется окладно-премиальная оплата труда, которая заключается в том, что в определенное число каждого месяца работнику выплачивается сумма, указанная в его трудовом договоре, со всеми заработанными надбавками. Надбавки и премии к зарплате утверждает директор магазина, исходя из проделанной работы работником.

Выплата заработной платы выполняется два раза каждый месяц в Российской валюте «рубли». После двух рабочих недель работник получает аванс в виде 30% от всей зарплаты, а потом в конце рабочего месяца остальную часть денег. Когда день выплаты зарплаты совпадает с выходным или праздничным днём её выплачиваю в последний день перед выходным или праздником.

Так же на предприятии существует три вида надбавок:

- сверхурочная работа;
- выходные дни;
- праздничные дни.

За каждый вид надбавок предусмотрены разные денежные выплаты. Например, за выходные и праздничные дни оплата идёт как за два рабочих дня. А оплата сверхурочной работы варьируется от 30% до 50% от зарплаты за один рабочий день.

В данной организации предусмотрена выплата премий. Премия выплачивается в размере 25% от всей заработной платы работника. Назначением премий занимается директор предприятия. Премию получает лучший сотрудник по итогам каждого рабочего месяца. Данный пример получения премии увеличивает работоспособность в магазине, так как каждый сотрудник стремится стать лучшим в каждом месяце.

Также в магазине предусмотрены поздравительные выплаты в честь дня рождения и нового года.

Любые не регулированные выплаты работникам проводятся в соответствии с Трудовым Кодексом Российской Федерации [30].

В организации так же используют систему штрафных санкций сотрудников. Это помогает контролировать работу сотрудников. А так же помогает корректировать: поведение на рабочем месте, внешний вид, общение с клиентами и выполнение всех работ в нужный срок. Все штрафные санкции описываются в трудовом договоре. Если работник в течение одного месяца получает более трёх штрафных санкции, его увольняют с работы в конце месяца.

Директор магазина «Мир Дерева» должен предоставить комфортные условия работы на местах: кондиционер, кулер с холодной и горячей водой, чайник и микроволновая печь.

На основе анализа организационной структуры магазина, можно отметить, что в магазине "Мир дерева" работает сплочённый, уважающий друг друга коллектив. Наборный персонал предприятия состоит из опытных и высокообразованных специалистов, имеющих заработную плату, состоящую из системы надбавок и премий, а так же умного, умеющего найти подход к каждому сотруднику директора высшего уровня [8].

### **1.3 Анализ современных информационных CRM-систем**

Информационная система является комплексом коммуникационных средств реализующее: сбор, хранение, передачу и переработку данных об объекте. Целью внедрения информационной системы является повышение торговой деятельности предприятия за счёт:

- обработки и хранения повседневной информации;
- автоматизации документооборота предприятия;
- использования в деятельности организации современных способов контроля деятельности сотрудников.

Возможно, выделить несколько требований к информационным системам:

- полный функциональный набор;
- система должна работать на различных системных платформах;
- обеспечивать безопасность данных;
- экономическую эффективность – приумножать прибыль предприятия.

Информационные системы классифицируют по их функциям: операционные информационные системы (системы обработки транзакций, автоматизированные системы управления технологическим процессом, офисные автоматизированные системы) и управленческие информационные системы (информационные системы отчетности, системы поддержки принятия решений, информационные системы топ-менеджмента, программные средства коллективной работы).

В настоящее время существует несколько видов информационных систем:

- ERP – система управления ресурсами предприятия;
- EAM – систему управления активами и фондами;
- CRM – систему управления взаимоотношениями с клиентами;
- SCM – систему управления цепочками поставок.

ERP – это программный продукт, основным назначением которого является управление и планирование производством предприятия. Данные системы в основном используются в таких сферах деятельности:

- серийное сборочное (электроника, машиностроение);
- мелкосерийное и опытное (авиация, тяжелое машиностроение);
- дискретное (металлургия, упаковка);
- непрерывное (нефтедобыча, газопереработка).

EAM – это программный продукт, основным назначением которого является обеспечение управления жизненным циклом активов предприятия с



применением управления техническим обслуживанием. Данные системы в основном используются в таких сферах деятельности:

- управление производственной деятельностью, техническим обслуживанием и ремонтом;
- управление материалами;
- управление персоналом;
- управление финансами.

SCM – это программный продукт, основным назначением которого является установление взаимовыгодных связей предприятия с поставщиками продукции. В данном программном обеспечении операции всю деятельность начиная с создания товара до его продажи [35].

CRM – это программный продукт, основным назначением которого является управление взаимоотношениями с клиентами в условиях высококонкурентного рынка. При помощи данного программного обеспечения возможно заключение длительных договоров на поставку товаров клиенту или предприятию. Для эффективной ее работы необходима ее интеграция с внутренними информационными системами фирмы, а ее бизнес-процессы должны быть устроены так, чтобы обратная связь с клиентом помогала выпускать конкурентоспособные продукты.

CRM-приложение помогает решить ряд задач:

- удержание клиентов;
- привлечение клиентов;
- обеспечение прибыльности по клиентам [27].

На данный момент существуют следующие типы CRM-системы:

- оперативная CRM – обеспечение оперативного доступа к данным по определенному клиенту в ходе взаимодействия с ним в рамках стандартных бизнес-процессов продажи;
- аналитическая CRM – обеспечение синхронизации структур данных и поиск закономерностей для выработки наиболее лучшей стратегии;

- коллаборационная CRM – предоставление возможности клиенту непосредственного участия в процессе производства товара.

При внедрении CRM системы в организацию:

- сокращается повтор действий продаж за счет автоматизации рутинных операций;
- снижение издержек продаж, маркетинга и послепродажной поддержки;
- повышение деятельности рекламной компании организации.

В ходе использования CRM системы вся информация, хранимая в базе данных, становится структурированной и систематизированной, что оптимизирует работу предприятия в целом. Стоит отметить тот факт, что деятельность любого без исключения предприятия зависит от удовлетворенности клиента, в результате чего необходимым является разработка и внедрение CRM-системы [26].

В качестве разработки программного продукта была взята за основу CRM-система.

CRM-системы представляют собой управление следующими операциями, совершаемыми предприятием:

- совершение закупок;
- совершение продаж;
- управление клиентской базы;
- автоматизация прайс-листа;
- автоматизация учета товара на складе;
- автоматизация документооборота предприятия;
- хранение информации о сотрудниках предприятия;
- контроль доступа к хранимой информации в базе данных предприятия;
- хранение истории взаимоотношений с поставщиками и клиентами [25].

Функционал CRM-систем показана на рис. 1.4.



Рис. 1.4. Функционал CRM-систем

В настоящее время существует ряд программных продуктов, обеспечивающих автоматизацию рутинной работы предприятия:

- «Битрикс 24» - корпоративная облачная система, в первую очередь, созданная для эффективного выполнения деятельности организации. Внедрение данного программного обеспечения позволяет ставить задачи для решения, получать задания от руководителя, общаться с другими сотрудниками организации, оперативно отвечать на заявки клиентов. Средства облачной системы помогают контролировать выполнение поставленных бизнес-процессов в каждом отделе компании. В «Битрикс 24» удобно работать с проектами различного вида, в результате чего оптимизируется связь между базой данных, календарем, и задачами. Однако существует ряд недостатков данной информационной системы, к таким можно отнести сложный интерфейс, который требует дополнительного обучения, и отсутствие того [4].

- «SAP CRM» - информационная система, созданная для выполнения комплексных задач, автоматизации процессов поддержки принятия решений, а так же для оптимизации стратегической деятельности организации, и удержания конкурентоспособности предприятия на

российском рынке. Особенностью данной системы является гибкость интерфейса [39].

- «StorVerkCRM» - это программный продукт, разработанный непосредственно для управления базой клиентов организации. Преимуществами данной системы являются: безопасность доступа к базе данных, ограничение прав доступа, многосторонний анализ деятельности, встроенный почтовый файлообменник [38].

- «Asoft CRM» - программный продукт, созданный для управления такими модулями как: продажи, закупки, управление маркетинговой деятельностью, документооборот. Своим интерфейсом «Asoft CRM» напоминает календарь, который открывается сразу же после входа в систему с запланированными делами и задачами. Информационная система обладает интуитивным и простым интерфейсом, в котором может разобраться даже не опытный пользователь [40].

- «CRM-система «Мир дерева» - это специализированное программное обеспечение, созданное для оптимизации деятельности продаж предприятия «Мир дерева». Особенностью данной системы является заточенность под специализированный функционал организации. В данном программном продукте усовершенствована система документооборота, хранение журнала поставщиков, гибкость клиентской базы, а также ведется статистика и анализ ходового товара предприятия.

Сравнение выше описанных CRM-систем проводилось на основе следующего функционала:

- гибкость системы – изменение работоспособности системы при значительном расширении организации;
- хранение журнала поставщиков – хранение истории поставок (количество, качество, стоимость каждого товара по отдельности и итоговую сумму) каждого поставщика на протяжении указанного срока;

- статистика и анализ ходового товара предприятия – проведение статистики и анализа продаж товара, а также наиболее ходовых видов товара;
- оповещения о приближении запасов к критичному запасу – оповещение о приближении товара к критическому запасу;
- простота и удобство интерфейса – система должна быть простой в использовании при совершении, каких либо операций в приложении.

Рейтингование систем производилось по следующим баллам:

- 0 баллов – функция не предусмотрена;
- 1 балл – функция слабо реализована;
- 2 балла – функция реализована, но не в полной мере;
- 3 балла – функция хорошо реализована, и выполняет поставленные перед ней функции.

В таблице 1.1 показано сравнение CRM – систем.

Таблица 1.1

### Сравнение CRM- систем

	«Битрикс 24»	«SAP CRM»	«StorVerkCRM»	«Asoft CRM»	«CRM-система «Мир дерева»
Гибкость системы	3	2	1	3	2
Хранение журнала поставщиков	2	3	3	2	3
Статистика и анализ ходового товара предприятия	1	2	1	0	3
Оповещения о приближении запасов к критичному запасу	1	1	2	2	3
Простота и удобство интерфейса	3	3	2	3	3
Итого	10	9	9	10	14

Проведя анализ всех CRM-систем по вышеуказанным критериям, было определено, что "CRM-система "Мир дерева" является наиболее подходящей, так как по большинству критериев получила наивысший балл. Единственным недостатком данной системы является ее гибкость, в силу того, что в данной организации не удается ее протестировать [29].

#### **1.4 Постановка цели и задач для реализации CRM – приложения «Мир Дерева»**

В результате анализа деятельности магазина «Мир Дерева», стало понятно, что основное направление магазина заключается в закупке, распиле, продаже и доставке пиломатериала клиенту, а так же извлечение из всех перечисленных действий прибыли. Исходя из предметной области предприятия, была поставлена цель выпускной квалификационной работы.

Целью данной выпускной квалификационной работы является автоматизация бизнес - процессов учёта закупок и продаж товара при помощи внедрения в организацию CRM – приложения.

Для реализации поставленной цели необходимо решение следующих задач:

- подробное изучение предметной области, а именно деятельности магазина «Мир Дерева»;
- проведение сравнительного анализа современных информационных систем, выявление их недочетов для разработки специализированного приложения «Мир Дерева»;
- проектирование и разработка базы данных для CRM-приложения «Мир Дерева»;
- проектирование пользовательского интерфейса для разрабатываемого программного продукта;
- разработка и тестирование контрольного примера разработанного программного продукта «Мир Дерева».

Для решения поставленной цели и задач было выбрано следующее программное обеспечение:

- СУБД Firebird 2.1;
- утилита IBExpert;
- среда программирования Embracadero RAD Studio;
- язык программирования C++.

## ГЛАВА 2. ПРОЕКТИРОВАНИЕ CRM- СИСТЕМЫ

### 2.1 Инфологическое проектирование

Инфологическое проектирование базы данных - это процесс конструирования общей информационной модели предприятия на основе отдельных моделей данных пользователей, которая является независимой от особенностей реально используемой СУБД и других физических условий.

Выделяют несколько этапов инфологического проектирования:

- Преобразование локальной концептуальной модели данных в локальную логическую модель;
- Определение набора отношений исходя из структуры локальной логической модели данных;
- Проверка модели с помощью правил нормализации;
- Проверка модели в отношении транзакций пользователей;
- Создание диаграммы сущность-связь;
- Определение требований поддержки целостности данных;
- Обсуждение разработанных локальных логических моделей данных с конечными пользователями.

Инфологическая модель описывает понятия предметной области, их взаимосвязь, а также ограничения на данные, налагаемые предметной областью [32].

Инфологическая модель данных является начальным прототипом будущей базы данных. Эта модель строится в терминах информационных единиц, но без привязки к конкретной СУБД. Более того, инфологическая модель данных обязательно должна быть выражена средствами именно реляционной модели данных. Основным средством разработки инфологической модели данных в настоящий момент являются различные



варианты ER-диаграмм (Entity-Relationship, диаграммы сущность-связь). Одну и ту же ER-модель можно преобразовать как в реляционную модель данных, так и в модель данных для иерархических и сетевых СУБД, или в постреляционную модель даны [1].

Основными понятиями ER-диаграммы являются сущность, связь и атрибут.

Сущность — это реальный или виртуальный объект, имеющий существенное значение для рассматриваемой предметной области, информация о котором подлежит хранению. Если не вдаваться в подробности, то можно считать, что сущности соответствуют таблицам реляционной модели [33].

Под информационным объектом понимается некоторая сущность фрагмента действительности, например организация, документ, сотрудник, место, событие и т. д.

Связи выступают в модели в качестве средства, с помощью которого представляются отношения между сущностями, имеющими место в предметной области. При анализе связей между сущностями могут встречаться бинарные (между двумя типами сущностями), тернарные (между тремя типами сущностями) и общем случае n-арные связи.

В предметной области выделяются различные типы объектов, представляемые в информационной системе в каждый момент времени конечным набором экземпляров данного типа. Каждый тип объекта включает (идентифицируется) присущий ему набор атрибутов (свойств, характерных признаков, параметров).

Связь - это ассоциация между сущностями, включают по одной сущности из каждого участвующего в связи типа сущности.

Атрибут представляет логически неделимый элемент структуры информации, характеризующийся множеством атомарных значений (атрибут «Имя» объекта типа «Лицо», атрибут «Текст» объекта типа «Документ») [23].

Один или некоторая группа атрибутов объекта данного типа могут исполнять роль ключевого атрибута, по которому различаются конкретные экземпляры объектов (объект «Лицо» - ключ совокупность атрибутов «Фамилия», «Имя», «Отчество» или один атрибут «Номер договора»).

Различные типы объектов и различные экземпляры одного типа объекта могут быть охвачены определенными отношениями, которые в рамках ER-модели выражаются связями.

Тип связи это осмысленная ассоциация между сущностями разных типов. Тип связи является набором ассоциаций между двумя типами сущностей. Каждому типу сущности присваивается имя, которое должно описывать его функцию [18].

Связи по признаку множественности могут быть трех типов:

- «один-к-одному»
- «один-ко-многим»
- «многие-ко-многим»

На рис. 2.1 показано инфологическое проектирование базы данных CRM-приложения.

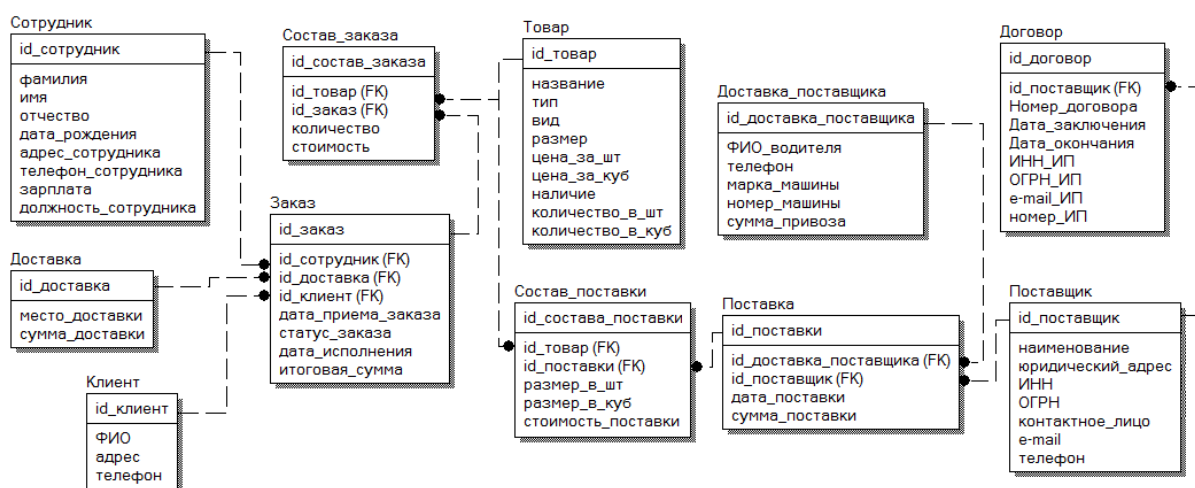


Рис. 2.1. Инфологическое проектирование базы данных CRM-приложения

Сущностями данной базы данных являются: сотрудник, доставка, клиент, заказ, состав заказа, товар, состав поставки, поставка, доставка поставки, поставщик и договор.

В сущность сотрудник входят следующие атрибуты: id\_сотрудник, фамилия, имя, отчество, должность, дата\_рождения, адрес\_сотрудника, телефон\_сотрудника, зарплата.

Сущность доставка включает в себя атрибуты: id\_доставка, место\_доставки, сумма\_доставки.

Сущность клиент включает в себя атрибуты: id\_клиент, ФИО, адрес, телефон.

Сущность состав\_заказа включает в себя атрибуты: id\_состав\_заказа, id\_заказ, id\_товар, количество, стоимость.

Сущность заказ включает в себя атрибуты: id\_заказ, id\_доставка, id\_клиент, дата\_приёма\_заказа, статус\_заказа, дата\_исполнения, итоговая сумма.

Сущность товар включает в себя атрибуты: id\_товар, название, тип, вид, размер, цена\_за\_шт, цена\_за\_куб, наличие, количество\_в\_шт, количество\_в\_куб.

Сущность состав\_поставки включает в себя атрибуты: id\_состав-поставки, id\_товар, id\_поставки, размер\_в\_шт, размер\_в\_куб, стоимость\_поставки.

Сущность поставка включает в себя атрибуты: id\_поставки, id\_доставка-поставки, id\_поставщик, дата\_поставки, сумма\_поставки.

Сущность поставщик включает в себя атрибуты: id\_поставщик, наименование, юридический\_адрес, ИНН, ОГРН, контактное\_лицо, e\_mail, телефон.

Сущность доставка-поставщика включает в себя атрибуты: id\_доставка\_поставщика, ФИО\_водителя, телефон, марка\_машины, номер\_машины, сумма\_привоза.

Сущность договор включает в себя атрибуты: id\_договор, id\_поставщик, номер\_договора, дата\_заключения, дата\_окончания, ИНН\_ИП, ОГРН\_ИП, e\_mail, номер\_ИП.

## 2.2 Даталогическое проектирование

Даталогическое проектирование базы данных - процесс подготовки описания реализации базы данных на вторичных запоминающих устройствах; на этом этапе рассматриваются основные отношения, организация файлов и индексов, предназначенных для обеспечения эффективного доступа к данным, а также все связанные с этим ограничения целостности и средства защиты.

Даталогическое проектирование является третьим и последним этапом создания проекта базы данных, при выполнении которого проектировщик принимает решения о способах реализации разрабатываемой базы данных. Приступая к даталогическому проектированию базы данных, прежде всего необходимо выбрать конкретную целевую СУБД. Поэтому даталогическое проектирование неразрывно связано с конкретной СУБД. Между инфологическим и даталогическим проектированием существует постоянная обратная связь, так как решения, принимаемые на этапе даталогического проектирования с целью повышения производительности системы, способны повлиять на структуру инфологической модели данных.

Как правило, основной целью даталогического проектирования базы данных является описание способа физической реализации логического проекта базы данных [12].

Выделяют несколько этапов даталогического проектирования баз данных:

- Перенос глобальной инфологической модели данных в среду целевой СУБД (Создание базовой функциональной схемы реляционной БД);
- Проектирование таблиц данных в среде целевой СУБД;

- Реализация бизнес-правил предприятия;
- Проектирование физического представления БД;
- Анализ транзакций (выбор важнейших из них);
- Выбор файловой системы;
- Определение вторичных индексов;
- Определение требований к памяти.
- Разработка механизмов защиты.
- Разработка пользовательских видов.
- Определение прав доступа.

На рис. 2.2 показано даталогическое проектирование базы данных CRM-приложения [24].

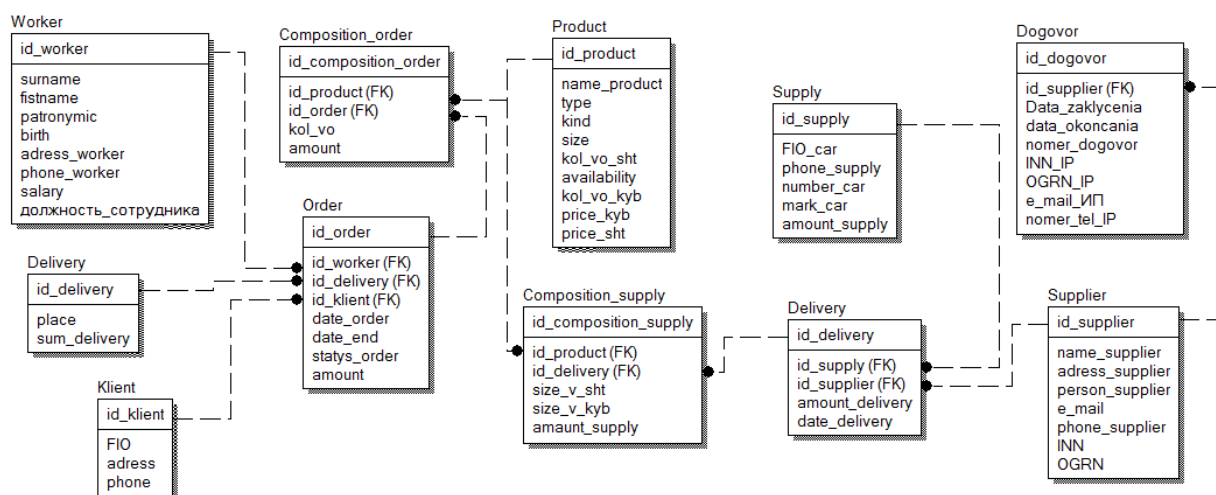


Рис. 2.2. Даталогическое проектирование базы данных CRM-приложения

### 2.3 Выбор программных средств разработки

Для реализации CRM-приложения были выбраны такие программные средства разработки и проектирования как: ERWin, Embercadero Rad Studio XE8, IBExpert, Firebird.

ERwin является одним из средств проектирования базы данных. Он относится к категории I - CASE. ERwin обеспечивает интеграцию моделей

верхнего уровня с моделями нижнего уровня. Модели верхнего уровня создаются на начальных стадиях проектирования информационных систем. Модели нижнего уровня разрабатываются на этапе создания программного кода и тестирования.

Основное предназначение ERwin это проектирование данных, поэтому с точки зрения проектирования баз данных его применение ограничено. Он позволяет создавать модели данных высокого уровня, обеспечивает автоматическое преобразование этих моделей в модели низкого уровня, создаёт графическую схему базы данных и описание данных на уровне программного кода.

В качестве стандартов проектирования в ERwin data modeler применяется два стандарта: IDEF1X и IE (Information Engineering), основанные на диаграммах «сущность-связь».

Схемы баз данных, которые проектируются с помощью ERwin data modeler, делятся на два вида: инфологическая и даталогическая модель.

CASE средство СА ERwin r9 обладает следующими возможностями, важными для проектирования базы данных:

- Графическое представление комплексных структур данных. За счет графических средств ERwin можно в автоматическом режиме создавать модели, которые обеспечивают наглядное представление всей структуры данных;
- Применение стандартных элементов. В ERwin предусмотрен набор типовых моделей (референтные модели), которые позволяют повысить эффективность работы и исключить ошибки дублирования информации;
- Сравнение моделей данных и баз данных. За счет этого инструмента есть возможность провести автоматическое сравнение и синхронизацию элементов бизнес процессов (из моделей данных) с элементами базы данных;
- Интеграция с другими средствами моделирования. ERwin обладает широкими возможностями по экспорту/импорту моделей. За счет

встроенных средств есть возможность обмениваться моделями с другими средствами моделирования, в том числе, поддерживающих UML нотации.

Преимущества, предоставляемые ERwin, можно рассматривать по отношению к CASE средствам, направленным на моделирование данных. С этой точки зрения у пакета СА ERwin r9 существуют следующие преимущества:

- Возможность взаимодействия пользователей. ERwin предоставляет различные возможности по обмену информацией между ролями внутри организации. За счет web портала бизнес аналитики и технические специалисты могут получать доступ к моделям данных в понятных для них представлениях;
- Стандартные представления элементов. Для обеспечения единства представления элементов моделей данных в ERwin используются стандартизованные представления имен объектов, стандартизованные типы данных и стандартные образцы моделей (референтные модели);
- Применение открытой архитектуры. ERwin обеспечивает широкие возможности по интеграции с другими инструментами моделирования процессов и разработки информационных систем (более 120 различных инструментов);
- Визуальное представление больших массивов данных. За счет мощной графической системы и системы навигации можно обеспечить графическое представление моделей данных различной структуры и степени вложенности;
- Совместная работа пользователей с репозиторием. Для эффективной совместной работы над моделями ERwin обеспечивает автоматический контроль версий, управление доступом, управление конфликтами и изменением моделей.

Embarcadero RAD Studio XE8 Architect - это комплексное решение для разработки программного обеспечения, ускоряющее проектирование, программирование и создание взаимодействующих приложений для

Windows, Mac, iOS, Android и IoT, которые интегрируются с корпоративными СУБД. RAD Studio Architect поддерживает все возможности редакции Enterprise, а также содержит мощную среду моделирования и проектирования баз данных - ER/Studio Developer Edition. Поддерживая новые 64-разрядные и универсальные iOS приложения, RAD Studio XE8 заметно упрощает разработку: можно добавлять IoT функции, такие как информирование, о приближении с помощью «маячков» (beacons), повышать производительность приложений и эффективность разработчиков, а также лучше понимать поведение пользователей. Менеджер пакетов GetIt упрощает доступ к библиотекам исходного кода, компонентам и другим элементам IDE.

На данный момент существует множество преимуществ Embarcadero RAD Studio XE8 Architect:

- Поддержка 64-разрядных приложений iOS и универсальных приложений. Чтобы разработчики могли создавать новые приложения для Apple App Store, версия XE8 RAD Studio предлагает новые 64-разрядные компиляторы Object Pascal и C++ для iOS, включая компоновщики и другие инструменты для создания 64-разрядных iOS приложений. RAD Studio XE8 поддерживает универсальные приложения Apple, предлагая 32- и 64-разрядные двоичные коды в одном пакете, а также широкий спектр собственных элементов управления iOS для поддержки уникальных особенностей этой платформы.
- Функция предпросмотра на разных устройствах и форм-факторах. Как создавать взаимодействующие приложения ещё быстрее? В этом поможет функция предпросмотра на разных устройствах в RAD Studio XE8, позволяющая наглядно отобразить пользовательский интерфейс на всех основных форм-факторах. Так можно легко определить, где нужна дополнительная оптимизация. Просматривайте предварительный вид приложения динамически, для любого числа форм-факторов и ориентаций, включая десктопы, планшеты, смартфоны и смарт-часы.



- Информирование о приближении с помощью beacons. Благодаря новой поддержке beacons на уровне компонентов, реализованной в RAD Studio XE8, разработчики могут с лёгкостью добавлять информирование о приближении и местонахождении в готовые приложения для Windows и мобильных платформ.

- AppAnalytics. Сервис AppAnalytics помогает разработчикам отслеживать и анализировать модели поведения их пользователей, собирая анонимную статистику использования приложений, созданных с помощью RAD Studio XE8. AppAnalytics — сервис Embarcadero с оплатой за использование, поддерживающий приложения на VCL и FireMonkey — это означает, что разработчик может использовать этот сервис в приложениях на Windows, OS X, Android и iOS.

- Менеджер пакетов GetIt. Это средство обеспечивает доступ к библиотекам и компонентам на VCL и FireMonkey, а также аналогичным элементам для различных устройств. С помощью менеджера пакетов GetIt можно с лёгкостью выбирать, загружать и обновлять библиотеки исходного кода, компоненты и другие элементы с серверов Embarcadero GetIt, устанавливая их одним кликом. Сюда входят все проекты TurboPack.

- Обновлённые Enterprise Mobility Services (EMS). EMS -готовое серверное решение служащее основой для взаимодействующих распределённых приложений с поддержкой сервера push-уведомлений для iOS и Android. Решение EMS, основанное на открытых стандартах и технологиях, предлагает простой в развертывании сервер промежуточного ПО, отвечающий за аутентификацию пользователей, управление специальными API REST и аналитику, включает встроенную и серверную защищённые базы данных, а также обеспечивает доступ к основным корпоративным базам данных: Oracle, SQL Server, Informix, DB2, PostgreSQL, MySQL и многим другим [3].

Среди других улучшений в RAD Studio XE8: технологии для значительного ускорения процесса разработки в виде встраиваемого модуля

Castalia с рефакторингом Object Pascal, статистикой проектов, анализом кода и поддержкой удобных комбинаций клавиш для редактора кода. Новое средство переноса параметров конфигурации служит для переноса конфигурации из старых версий продукта в новую, а также между различными инсталляциями в рамках одной версии.

По словам Эла Хильвы (Al Hilwa), руководителя исследований в области разработки ПО (IDC), «Разработка приложений для различных платформ — это один из немногих эффективных способов адаптироваться к росту количества платформ. Мультиплатформенная среда разработки, подобная RAD Studio, позволяет создавать собственные приложения для различных платформ, расширяя уже готовые приложения Windows VCL на Delphi и C++ и подключаясь к гаджетам и устройствам, промежуточному ПО, облачным платформам и корпоративным сервисам» [37].

СУБД FireBird является одной из самых популярных в мире бесплатных, кроссплатформенных систем управления базами данных с открытым исходным кодом. Она была разработана на основе исходного кода СУБД Interbase и развивается сегодня независимым международным сообществом. По надёжности, производительности и функциональным возможностям эта система мало в чём уступает признанным лидерам своего класса - Oracle и Microsoft SQL Server.

Firebird полностью поддерживает стандарты ANSI в синтаксисе языка SQL и может работать под управлением многих операционных систем - Windows, Linux, MacOS, Solaris и различных Unix-платформах. Среди достоинств этой системы использование очень развитого языка для хранимых процедур, триггеров и генераторов [5].

Хранимые процедуры и триггеры могут генерировать события, на которые может подписаться клиент. После успешного завершения транзакции (COMMIT) он будет извещён о произошедших событиях и их количестве.

Используя язык PSQL (процедурный SQL) Firebird, возможно создавать сложные хранимые процедуры для обработки данных полностью на стороне сервера. Для генерации отчётов особенно удобны хранимые процедуры с возможностью выборки, возвращающие данные в виде набора записей. Такие процедуры можно использовать в запросах точно так же как и обычные таблицы.

Для каждой таблицы возможно назначение нескольких триггеров, срабатывающих до или после вставки, обновления или удаления записей. Для триггеров используется язык PSQL, позволяя вносить начальные значения, проверять целостность данных, вызывать исключения, и т. д. В Firebird 1.5 появились «универсальные» триггеры, позволяющие в одном триггере обрабатывать вставки, обновления и удаления записей таблицы.

Идея генераторов (последовательностей) делает возможной простую реализацию автоинкрементных полей, и не только их. Генераторы являются 64-битными хранимыми в базе данных счётчиками, работающими независимо от транзакций. Они могут быть использованы для различных целей, таких как генерация первичных ключей, управление длительными запросами в соседних транзакциях, и т. д.

Firebird полностью бесплатна, она не требует ни регистрации, ни оплаты за поддержку. Исходный код этой системы открыт и любой желающий может разрабатывать на его базе собственные некоммерческие проекты, при условии соблюдения требований лицензии IDPL, по которой распространяется Firebird.

В качестве преимуществ Firebird можно отметить многоверсионную архитектуру, обеспечивающую параллельную обработку оперативных и аналитических запросов (это возможно потому, что читающие пользователи не блокируют пишущих), компактность (дистрибутив 5Mb), высокую эффективность и мощную языковую поддержку для хранимых процедур и триггеров.

Firebird является сервером баз данных. Один сервер Firebird может обрабатывать несколько сотен независимых баз данных, каждую с множеством пользовательских соединений. Он является полностью свободным от лицензионных отчислений даже для коммерческого использования.

Среди недостатков: отсутствие кэша результатов запросов, полнотекстовых индексов [7].

IBExpert — GUI-оболочка, предназначенная для разработки и администрирования баз данных InterBase и Firebird, а также для выбора и изменения данных, хранящихся в базах.

Существует множество достоинств IBExpert:

- Поддержка InterBase версий 4.x, 5.x, 6.x, 7.x, 2007 и 2009; Firebird 1.x, 2.x, 3.x; Yaffil 1.x;
- работа одновременно с несколькими базами данных;
- отдельные редакторы для всех объектов БД с синтаксической подсветкой;
- мощный SQL-редактор с историей запросов и возможностью их фонового выполнения;
- автозавершение кода SQL (название таблиц, полей, и т. п.);
- отладчик хранимых процедур и триггеров;
- поиск в метаданных;
- полное и частичное извлечение данных и метаданных;
- анализатор зависимостей объектов баз данных;
- отчёты по метаданным;
- менеджеры пользователей и пользовательских привилегий;
- экспорт данных в различные форматы.

IBExpert обладает множеством облегчающих работу компонентов: визуальный редактор для всех объектов базы данных, редактор SQL и исполнитель скриптов, отладчик для хранимых процедур и триггеров, построитель области, инструмент для импорта данных из различных

источников, собственный скриптовый язык, а также дизайнер баз данных и т.д.

## **2.4 Проектирование пользовательского интерфейса CRM-системы**

Большую роль в проектировании и реализации CRM-приложение играет удобный и функциональный пользовательский интерфейс.

Пользовательский интерфейс – это взаимодействие информационной модели предметной области, средств программирования и ряд взаимодействий сотрудника с базой данных предприятия и компонентами, которые позволяют создание информационной системы в процессе работы с программной средой разработки.

Под информационной системой подразумевается хорошее понимание основного направления деятельности предприятия, создаваемое с помощью визуальных объектов, представляющих ряд и взаимодействие набора некоторых компонентов.

Программный продукт и способы проектирования пользовательского интерфейса определяются составом программ, которые имеются в пользовании у сотрудника предприятия, и от поставленной задачи пред ним.

Эффективная работа сотрудника определяется не только функциональным возможностям находящегося в его пользовании программного обеспечения, но и приемлемостью для сотрудника этих возможностей. Качество пользовательского интерфейса позволяет использовать возможности потенциально имеющихся ресурсов, а так же является самостоятельным описанием программного обеспечения, сопоставимых по значениям с такими критериями: надёжность и эффективное пользование вычисляемых ресурсов [10].

Одно из основных достоинств пользовательского интерфейса состоит в том, что сотрудник всегда ощущает, что он руководит процессами программного обеспечения, а не программа руководит им.

При создании удобного и функционального пользовательского интерфейса, он должен соответствовать небольшому ряду свойств:

- естественность;
- согласованность;
- дружелюбность;
- принцип «обратной связи» с пользователем;
- простота;
- гибкость;
- эстетическая привлекательность.

Естественный интерфейс — это интерфейс, который не позволяет сотруднику менять стандартные для его деятельности возможности решения задач. Данное определение соответствует тому, что выдаваемые приложением результаты и всплывающие окна не должны обосновываться вспомогательной информацией [20].

Согласованность обеспечивает сотрудникам переносить имеющую информацию в решения новых задач, осваивать функционал быстрее, и поэтому больше уделять внимания поставленной задаче, а не проводить время с ознакомлением различного рода компонентов и их выполняемых операций. Таким образом, согласованность делает пользовательский интерфейс более узнаваемым и предсказуемым, так же предоставляет информацию о поведении интерактивных элементов.

В пользовательском интерфейсе для реализации свойства согласованности необходимо учитывать его различные критерии:

- Согласованность в соответствии с товаром. Это означает, что каждая команда системы должна исполнять только одну операцию, при любых изменениях в программном продукте.
- Согласованность в пределах среды разработки. Поддержание согласованности с интерфейсом, предоставляемым операционной системой (например, ОС Windows), пользовательское приложение может «опираться»

на те знания и навыки пользователя, которые он получил ранее при работе с другими приложениями.

- **Согласованность в использовании метафор.** Если поведение некоторого программного объекта выходит за рамки того, что обычно подразумевается под соответствующей ему метафорой, у пользователя могут возникнуть трудности при работе с таким объектом.

Существуют следующие принципы проектирования интерфейса:

- **дружественность интерфейса (иначе принцип прощения пользователя)** – данный принцип предполагает разрешение совокупности наборов действий и предупреждений пользователей о ситуациях, которые могут тем или иным образом навредить системе или сведениям в ней: предполагается возможность отмены или редактирования уже выполненных действий.

Эффективный интерфейс программного продукта обязан позволять избегать ситуации, которые возможно закончатся ошибками системы. Кроме этого, он обязан уметь приспособиться к возможным ошибкам пользователя и облегчать ему процесс устранения результатов этих ошибок [31].

- **Принцип «обратной связи»** - данный принцип предполагает обеспечение обратной связи для действий пользователя. Любое действие пользователя обязано получать зрительное, а бывает что и звуковое доказательство того, что система приняла введенную команду; при всем этом вид реакции, насколько это возможно, должен учесть природу исполненного действия.

- **Простота интерфейса** – данный принцип предполагает обеспечение простого пользовательского интерфейса. При всем этом подразумевается не упрощенство, а обеспечение легкости в его изучении и в применении. Помимо этого, он обязан давать доступ ко всему перечню многофункциональных возможностей, предусмотренным этим продуктом. Однако, реализация доступа к большому объему многофункциональных возможностей и обеспечение простоты работы противоречат друг другу.

Разработка эффективного интерфейса необходима для сбалансировки данных целей.

Одни из вероятных путей поддержки простоты пользовательского интерфейса служит предоставление на экране сведений, минимально нужных для исполнения пользователем следующего шага выполнения задания. А именно, следует игнорировать многословные командные названия и сообщения. Необдуманные названия затрудняют пользователю извлечение значимых данных.

- Гибкость интерфейса программного продукта – данный принцип предполагает способность учитывать подготовленность и производительность труда пользователя. Данное свойство системы подразумевает вероятность изменения конфигурации текстуры разговора. Концепция эластичного интерфейса в текущее время считается одной из главных сфер исследования взаимодействия человека и персонального компьютера. Главной проблемой является выбор и применение признаков для определения потребности внесения конфигураций и их сущностей.

- Эстетическая привлекательность – данный принцип предполагает способность зрительного представления применяемых объектов, гарантирующих передачу необходимых сведений о поведении и действиях различных объектов.

Стоит отметить, что качество интерфейса программного продукта трудно расценить количественными характеристиками, но и объективную оценку можно получить на базе характеристик, а именно:

- временной промежуток, нужный конкретному человеку для достижения значения познаний и навыков по работе с приложением;
- сохранение приобретенных навыков по прошествии некоторого временного промежутка;
- скорость решения задач при помощи программного продукта, в котором учитывается не быстрота работы программы, а время, нужное для достижения поставленной цели;



- личная удовлетворенность человека при работе с программным продуктом.

Учитывая все выше причисленные критерии и условия был спроектирован удобный и функциональный пользовательский интерфейс для CRM – приложения «Мир Дерева».

На рис. 2.3 показан пользовательский интерфейс CRM – приложения «Мир Дерева».

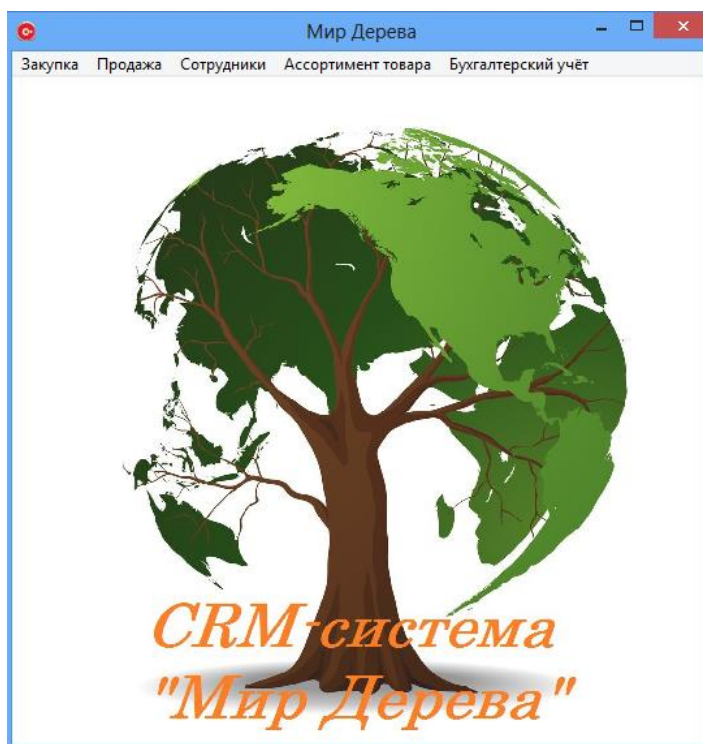


Рис. 2.3. Пользовательский интерфейс CRM-приложения «Мир Дерева»

Весь функционал CRM-приложения «Мир Дерева» находится в панели меню. В меню находятся такие вкладки как:

- закупки;
- продажи;
- сотрудники;
- ассортимент товара;
- бухгалтерский учёт.

В выпадающем списке закупки находятся: поставщики, поставки, договоры, доставка поставки и отчёты.

В выпадающем списке продажи находятся: оформление заказа, просмотр заказа, клиенты, доставка и отчёты.

В вкладке сотрудник можно будет просмотреть весь персонал организации, а так же добавлять, редактировать и удалять данные о рабочих.

В выпадающем списке ассортимент товара находится прайс-лист и товары на складе.

В вкладке бухгалтерский учёт можно будет сформировать отчёты фирмы о прибыли, затратах и чистой прибыли.

## ГЛАВА 3. РАЗРАБОТКА CRM- ПРИЛОЖЕНИЯ «МИР ДЕРЕВА»

### 3.1 Разработка базы данных

В ходе разработки базы данных было создано 11 сущностей и 10 связей. В состав каждой сущности входят соответствующие атрибуты.

На рис. 3.1 показаны все реализованные сущности по спроектированной базе данных автоматизированной системы.

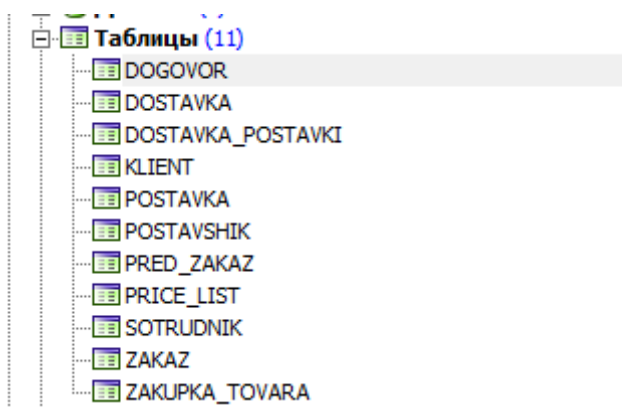


Рис. 3.1. Реализованные сущности базы данных

Для создания атрибутов были разработаны домены. Домен – это описание множества допустимых значений поля таблицы. Описание домена определяет тип данных, его длину и другие ограничения.

На рис. 3.2. показаны все разработанные домены системы.

Домены	Описание	Зависимости	Скрипт	Comparison	To-do		
Имя домена	Тип	Длина	Точ...	Не пусто	Подтип	Кодировка	Коллате
D_ADRES	VARCHAR	80		<input type="checkbox"/>		WIN1251	WIN1251
D_DATA	DATE			<input type="checkbox"/>			
D_FIO	VARCHAR	40		<input type="checkbox"/>		WIN1251	WIN1251
D_INDEX	INTEGER			<input checked="" type="checkbox"/>			
D_KOL_VO	SMALLINT			<input type="checkbox"/>			
D_PHONE	VARCHAR	20		<input type="checkbox"/>		WIN1251	WIN1251
D_SUMMA	FLOAT			<input type="checkbox"/>			
D_TIP	VARCHAR	10		<input type="checkbox"/>		WIN1251	WIN1251

Рис. 3.2. Разработанные домены

На рис. 3.2. видно что было разработано 8 доменов: D\_ADRES, D\_DATA, D\_FIO, D\_INDEX, D\_PHONE, D\_SUMMA и D\_TIP. Каждому из этих доменов был выбран соответствующий тип данных: VARCHAR(80), DATE, VARCHAR(40), INTEGER, SMALLINT, VARCHAR(20), FLOAT, VARCHAR(10).

В ходе разработки каждой сущности были указаны первичные и вторичные ключи для связей. Под первичным ключом понимается атрибут или группа атрибутов, которые единственным образом идентифицируют каждую строку в таблице. Вторичные ключи устанавливаются по полям, которые часто используются при поиске или сортировке данных. В отличие от первичных ключей поля для вторичных ключей могут содержать не уникальные значения.

На рис. 3.3. показаны все разработанные первичные и внешние ключи.

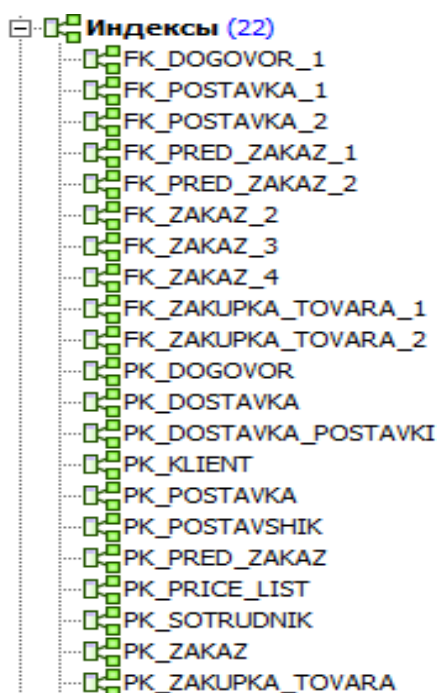


Рис. 3.3. Разработанные первичные и вторичные ключи

К каждой сущности был создан соответствующий генератор. Под генератором понимается специальный объект базы данных, который генерирует уникальные последовательные числа. Данные числа могут быть использованы в качестве идентификаторов.

На рис. 3.4. показаны все разработанные генераторы автоматизированной системы.

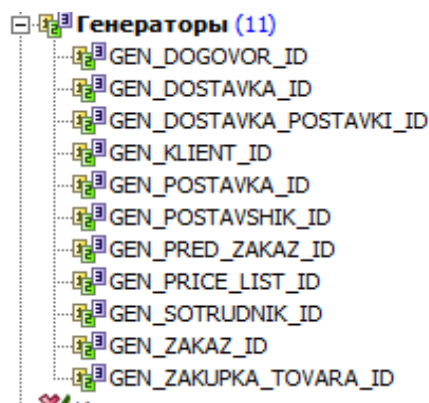


Рис. 3.4. Разработанные генераторы к каждой сущности

К каждой сущности были созданы процедуры: добавления, удаления и редактирования данных. Данные процедуры были сделаны для того, что бы пользователь мог управлять данными всех сущностей базы данных.

В автоматизированной системе была создана процедура добавления данных. Для этого нужно ввести все необходимые данные процедуры INSERT\_SOTRUDNIK. В случае не заполнения всех данных вызывается триггер о том, что нужно заполнить все данные [13].

В листинге 3.1 показан программный код создания процедуры INSERT\_SOTRUDNIK добавления данных.

Листинг 3.1. Программный код процедуры добавления данных.

```
begin
insert into sotrudnik(familia,name,otchestvo,dolgnost,phone_sotrudnika,adres,zarplata)
values(:familia,:name,:otchestvo,:dolgnost,:phone_sotrudnika,:adres,:zarplata);
suspend;
end
```

Помимо добавления данных была создана процедура удаления. Для удаления сотрудника из базы данных необходимо ввести следующую информацию: фамилия, имя, отчество сотрудника.

В листинге 3.2. показан программный код реализации процедуры удаления.

### Листинг 3.2. Создания процедуры удаления данных.

```
begin
delete from sotrudnik
where sotrudnik.familia=:FAMILIA AND sotrudnik.name=:name AND sotrudnik.
otchestvo=:otchestvo;
suspend;
end
```

В автоматизированной системе возможно редактирование занесенной информации в базу данных. Для этого необходимо отредактировать уже существующие данные. В листинге 3.3. показан программный код процедуры редактирования данных.

### Листинг 3.3. Программный код процедуры редактирования данных.

```
begin
update sotrudnik
set familia=:familia,
name=:name,
otchestvo=:otchestvo,
dolgnost=:dolgnost,
phone_sotrudnika=:phone_sotrudnika,
adres=:adres,
zarplata=:zarplata
where (id_sotrudnik = :id_sotrudnik);
end
```

В разработке базы данных было создано несколько представлений. Под представлениями понимаются мощные средства языка SQL, предназначенные для реализации механизмов подсетей пользователей базы данных. Представления позволяют скрыть от пользователей схему базы данных. В листинге 3.4. показан программный код представления состава заказа.

### Листинг 3.4. Программный код представления состава заказа.

```
CREATE VIEW SOSTAV_ZAKAZA_DS(
ID,NAIMENOVANIE_TOVARA,
TIP,VID,
RAZMER,
PRICE_SHTUK,
PRICE)
AS
select PRZ.id_pred_zakaza,PRL.name_tovara,PRL.tip,PRL.vid,PRL.razmer,
```

Продолжение. Листинг 3.4. Программный код представления состава заказа.

```
PRL.summa_shtuk,PRZ.price_tovara  
from pred_zakaz PRZ, price_list PRL  
where PRZ.id_price_list=PRL.id_price_list;
```

После реализации базы данных можно переходить непосредственно к программной реализации автоматизированной системы.

### 3.2 Програмная реализация автоматизированной системы

В начале программной реализации автоматизированной системы был создан контейнер, хранящий в себе невизуальные компоненты, DataModule, на котором были размещены следующие компоненты:

- TIBDatabase – предназначен для подключения к базе данных, так же обеспечивает двустороннюю связь;
- IBTransaction – предназначен для совершения транзакций, т.е. для совершения следующих операций: добавления, редактирования и удаления информации из базы данных;
- TIBTable – предназначен для получения данных из сущностей базы данных;
- TIBQuery – предназначен для отображения созданных представлений в базе данных;
- IBStoredProc – предназначен для совершения следующих операций: добавления, редактирования и удаления информации из базы данных;
- TDataSource - предназначен для отображения данных из каждой таблицы базы данных.

На рис. 3.6. показана информация о подключении к базе данных CRM – приложения «Мир Дерева» .

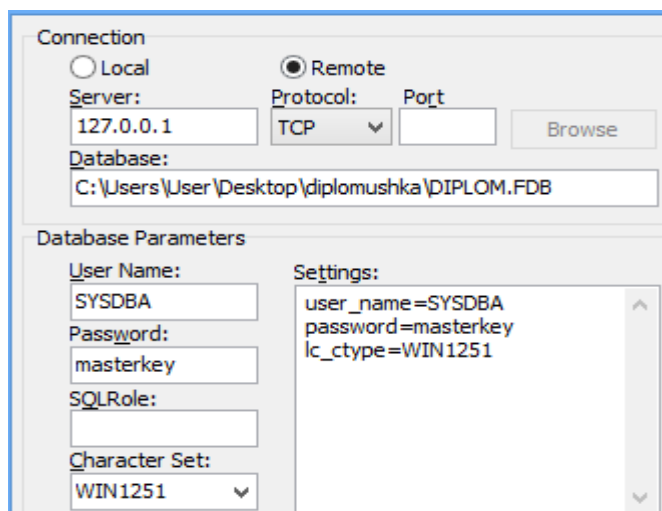


Рис. 3.6. Информация подключения базы данных

Во время разработки автоматизированной системы был использован компонент, хранящий в себе невизуальные компоненты.

На рис. 3.7. показаны все компоненты размещённые на DataModule.

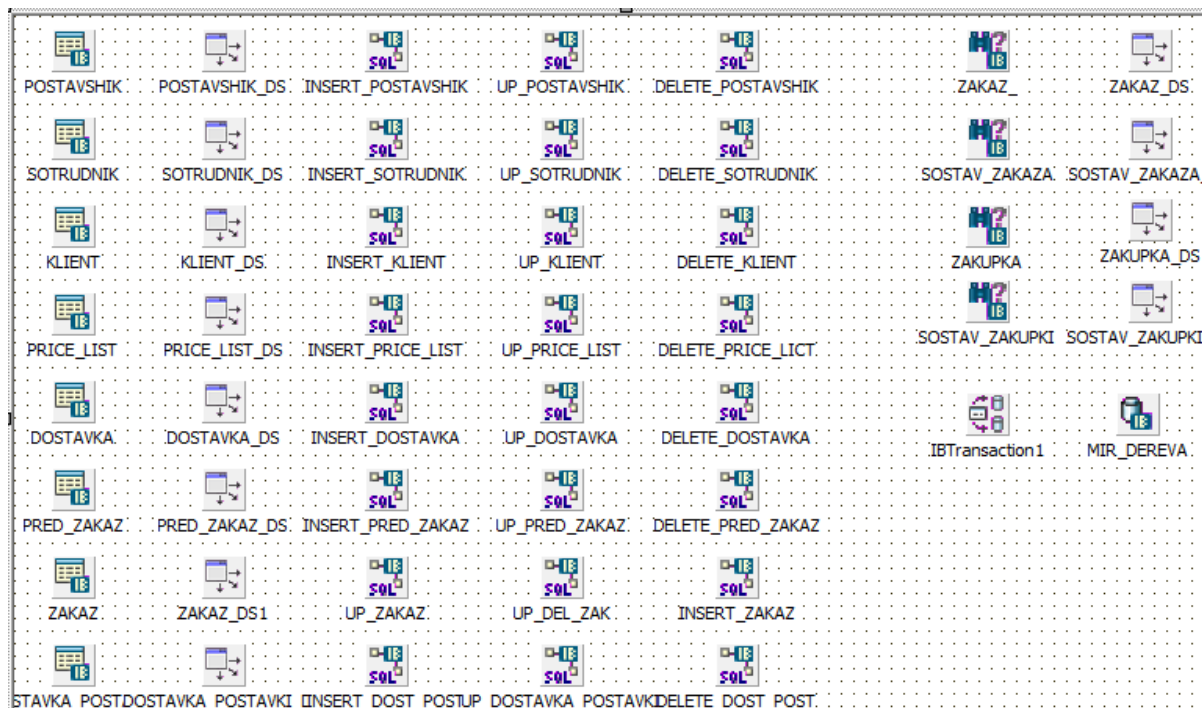


Рис. 3.7. Все компоненты размещённые на DataModule

В ходе создании программного продукта, была создана форма для авторизации сотрудника в системе, который вводит логин и пароль.



На рис. 3.8. показана форма для авторизации в CRM – системе.

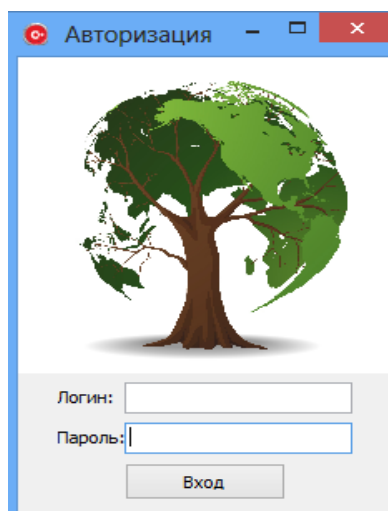


Рис. 3.8. Форма для авторизации в CRM – системе

После авторизации в системе пользователь входит в приложение и в результате чего отображается интерфейс программного продукта.

На рис. 3.9. представлен пользовательский интерфейс CRM-приложения.



Рис. 3.9. Пользовательский интерфейс CRM-приложения «Мир Деревя»

В пользовательском интерфейсе реализовано пять выпадающих списков:

- закупка – в данной вкладке реализованы следующие поля: поставщик, поставка, доставка поставки, договоры и отчёт.

- продажа – в данной вкладке реализованы следующие поля: оформить заказ, просмотр заказов, клиент, доставка и отчёты;
- сотрудники – данная вкладка содержит информацию о составе персонала на предприятии;
- ассортимент товар – в данной вкладке реализованы следующие поля: прайс–лист, который хранит в себе такие поля как: просмотр и печать, и товар на складе;
- бухгалтерский учёт – включает в себя разного вида отчётность: прибыль, расходы и чистая прибыль.

К каждой из этих вкладок реализованы следующие операции: фильтрации, добавления, редактирования, удаления и создание отчётов. В листинге 3.6. показана программная реализация операции удаления.

#### Листинг 3.6. Реализации операции удаления

```
DataModule5->DELETE_SOTRUDNIK->ParamByName("FAMILIA")-
>AsString=Form13->DBText2->Caption;
DataModule5->DELETE_SOTRUDNIK->ParamByName("NAME")-
>AsString=Form13->DBText3->Caption;
DataModule5->DELETE_SOTRUDNIK->ParamByName("OTCHESTVO")-
>AsString=Form13->DBText4->Caption;
DataModule5->DELETE_SOTRUDNIK->Prepare();
DataModule5->DELETE_SOTRUDNIK->ExecProc();
DataModule5->SOTRUDNIK->Active=false;
DataModule5->SOTRUDNIK->Active=true;
```

Представленный программный код выполняет удаление данных из таблицы базы данных по столбцам: FAMILIA, NAME, OTCHESTVO. После того как в компоненты: DBText2, DBText3, DBText4 внесут данные, после нажатия на кнопку «удалить» произойдёт операция удаления. В листинге 3.7. показана программная реализация операции фильтрации.

#### Листинг 3.7. Реализации операции фильтрация

```
DataModule5->SOTRUDNIK->Active=false;
DataModule5->SOTRUDNIK->Filtered=false;
DataModule5->SOTRUDNIK->Filter="FAMILIA LIKE '"+Form6->Edit1->Text+"%";
DataModule5->SOTRUDNIK->Active=true;
DataModule5->SOTRUDNIK->Filtered=true;
```

Представленный программный код выполняет фильтрацию данных в таблице «Сотрудники» по столбу FAMILIA, после того как в поле Edit1 введут данные для фильтрации. В листинге 3.8. показана программная реализация операции добавления.

#### Листинг 3.8. Реализации операции добавления

```
void __fastcall TForm11::Button1Click(TObject *Sender){
    DataModule5->INSERT_SOTRUDNIK->ParamByName("FAMILIA")-
>AsString=Form11->Edit1->Text;
    DataModule5->INSERT_SOTRUDNIK->ParamByName("NAME")-
>AsString=Form11->Edit2->Text;
    DataModule5->INSERT_SOTRUDNIK->ParamByName("OTCHESTVO")-
>AsString=Form11->Edit3->Text;
    DataModule5->INSERT_SOTRUDNIK->ParamByName("ADRES")-
>AsString=Form11->Edit4->Text;
    DataModule5->INSERT_SOTRUDNIK->ParamByName("PHONE_SOTRUDNIKA")-
>AsString=Form11->Edit5->Text;
    DataModule5->INSERT_SOTRUDNIK->ParamByName("DOLGNOST")-
>AsString=Form11->Edit6->Text;
    DataModule5->INSERT_SOTRUDNIK->ParamByName("ZARPLATA")-
>AsString=Form11->Edit7->Text;
    DataModule5->INSERT_SOTRUDNIK->Prepare();DataModule5-
>INSERT_SOTRUDNIK->ExecProc();DataModule5->SOTRUDNIK-
>Active=false;DataModule5->SOTRUDNIK->Active=true;}
```

Представленный программный код. выполняет добавление данных в таблицу SOTRUDNIK. После того как в поля: Edit1, Edit2, Edit3, Edit4, Edit5, Edit6 и Edit7 внесут данные, после нажатия на кнопку «добавить» произойдёт операция добавления. В листинге 3.9. показана программная реализация операции редактирования.

#### Листинг 3.9. Реализации операции редактирования

```
DataModule5->UP_SOTRUDNIK->ParamByName("ID_SOTRUDNIK")-
>AsString=Form12->DBText1->Caption;
    DataModule5->UP_SOTRUDNIK->ParamByName("FAMILIA")->AsString=Form12-
>DBEdit1->Text;
    DataModule5->UP_SOTRUDNIK->ParamByName("NAME")->AsString=Form12-
>DBEdit2->Text;
    DataModule5->UP_SOTRUDNIK->ParamByName("OTCHESTVO")-
>AsString=Form12->DBEdit3->Text;
    DataModule5->UP_SOTRUDNIK->ParamByName("PHONE_SOTRUDNIKA")-
>AsString=Form12->DBEdit4->Text;
    DataModule5->UP_SOTRUDNIK->ParamByName("ADRES")->AsString=Form12-
>DBEdit5->Text;
```

### Продолжение. Листинг 3.9. Реализации операции редактирования

```
DataModule5->UP_SOTRUDNIK->ParamByName("DOLGNOST")-
>AsString=Form12->DBEdit6->Text;
DataModule5->UP_SOTRUDNIK->ParamByName("ZARPLATA")-
>AsString=Form12->DBEdit7->Text;
DataModule5->UP_SOTRUDNIK->Prepare();
DataModule5->UP_SOTRUDNIK->ExecProc();
DataModule5->SOTRUDNIK->Active=false;
DataModule5->SOTRUDNIK->Active=true;
```

Представленный программный код операции выполняет редактирование данных в таблице SOTRUDNIK. После того как в поля: DBText1, DBEdit1, DBEdit2, DBEdit3, DBEdit4, DBEdit5, DBEdit6, DBEdit7 внесут изменения данных, после нажатия на кнопку «редактировать» произойдёт операция редактирования. В листинге 3.10 показана программная реализация операции создания отчёта.

### Листинг 3.10. Реализации операции создания отчёта

```
TfrxReport *Report;
Report = Form17->frxReport1;
Report->Preview = Form17->frxPreview1;
Report->ShowReport();
Form17->WindowState=wsMaximized; //Размер по всему экрану
Form17->Show();
```

Представленный программный код в листинге 3.10 выполняет создание отчёта по данным таблицы заказ, после нажатия на кнопку «сформировать отчёт» формируется отчёт по выбранным данным таблицы.

Все описанные операции: добавление, редактирование, фильтрация и удаления, так же по аналогии реализованны ко всем таблицам базы данных, чтобы все сотрудники могли полностью управлять данными предприятия, которые им положены по должности.

### 3.3 Тестирование контрольного примера

При запуске информационной системы «Мир Дерева» необходимо пройти авторизацию. Для сотрудника магазина «Мир Дерева» сначала нужно ввести логин и пароль для работы в системе.

Авторизация сотрудника в системе изображена на рис. 3.10.

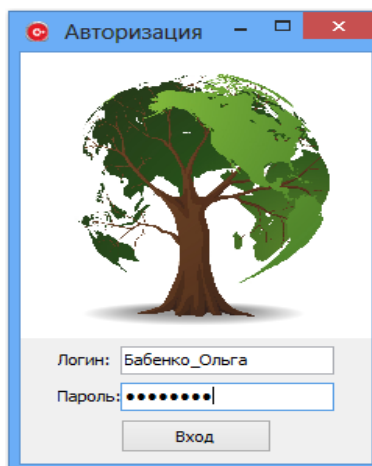


Рис. 3.10. Авторизация сотрудника в системе

После авторизации в системе сотрудник переходит в пользовательский интерфейс. В пользовательском интерфейсе находится весь функционал данной системы.

На рис. 3.11. показан пользовательский интерфейс сотрудника.



Рис. 3.11. Пользовательский интерфейс сотрудника

Для полного тестирования операций: добавления, редактирования, удаления и фильтрации, необходимо выбрать вкладку сотрудники. На рис. 3.12. показана вкладка сотрудник.

№	Фамилия	Имя	Отчество	Должность	Контактный телефон	Адрес места жительства	Зарботная плата
1	Гребеник	Павел	Сергеевич	Сторож	89516347691	ул.Некрасова 78	7000
2	Сытник	Артём	Генадьевич	Грузчик	89206321475	ул.Димитрова 178	12000
3	Бабенко	Ольга	Павловна	Бухгалтер	89205775007	ул.Республиканская 148 а	25000
4	Мамонтов	Игорь	Андреевич	Водитель	89145879632	ул.Суворова 56	18000
5	Панова	Елена	Андреевна	Менеджер по п	89047891236	город Белгород улица Попова дом 46 квартира 35	14000
6	Новиков	Иван	Вячеславович	Грузчик	89192478963	ул. Ленина 123	12000

Below the table, there is a search bar and a control panel with the following elements:

- Поиск:
- Добавить
- 1
- Гребеник
- Павел
- Сергеевич
- Редактировать
- Удалить

Рис. 3.12. Вкладка сотрудник

При нажатии на кнопку «добавить» выводится форма для заполнения информации о сотруднике. На рис. 3.13 представлена форма для ввода информации о сотруднике.

Фамилия     Имя     Отчество

Адрес     Телефон

Должность     Зарботная плата

Рис. 3.13. Форма для ввода информации о сотруднике

После нажатия на кнопку добавить выводится форма об успешном добавлении записи. На рис. 3.14 представлено окно подтверждения совершения операции.

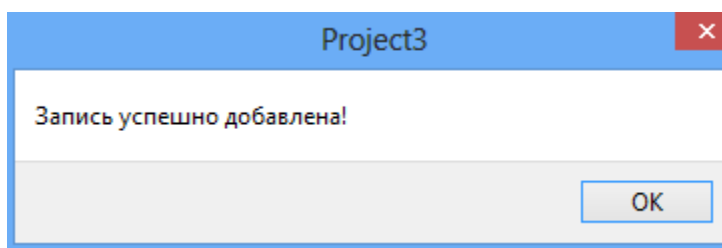


Рис. 3.14. Окно подтверждения совершения операции

После подтверждения о добавлении информации в базу, просмотрим информацию о сотрудниках.

На рис. 3.15 представлены все сотрудники, которые внесены в базу.

№	Фамилия	Имя	Отчество	Должность	Контактный телефон	Адрес места жительства	Зарботная плата
1	Гребеник	Павел	Сергеевич	Сторож	89516347691	ул.Некрасова 78	7000
2	Сытник	Артём	Генадьевич	Грузчик	89206321475	ул.Димитрова 178	12000
3	Бабенко	Ольга	Павловна	Бухгалтер	89205775007	ул.Республиканская 148 а	25000
4	Мамонтов	Игорь	Андреевич	Водитель	89145879632	ул.Суворова 56	18000
5	Панова	Елена	Андреевна	Менеджер по п	89047891236	город Белгород улица Попова дом 46 квартира 35	14000
6	Новиков	Иван	Вячеславович	Грузчик	89192478963	ул. Ленина 123	12000
7	Бабенко	Андрей	Викторович	Директор	89202054676	ул. Республиканская 148 а	30000

Рис. 3.15. Сотрудники магазина

На рис. 3.15. показано, что запись успешно добавилась в базу данных.

При вводе информации в поле поиск расположенное в левом нижнем углу произойдёт фильтрация данных. На рис. 3.16 представлена операция фильтрации.

№	Фамилия	Имя	Отчество	Должность	Контактный телефон	Адрес места жительства	Зарботная плата
3	Бабенко	Ольга	Павловна	Бухгалтер	89205775007	ул.Республиканская 148 а	25000
7	Бабенко	Андрей	Викторович	Директор	89202054676	ул. Республиканская 148 а	30000

Поиск:   3

Рис. 3.16. Операция фильтрации

Для выполнения операции удаления сначала выбираем нужного сотрудника и нажимаем на кнопку «удалить», которая находится в правом нижнем углу, выводится форма подтверждения удаления. На рис. 3.17 представлена форма подтверждения удаления информации о сотруднике.

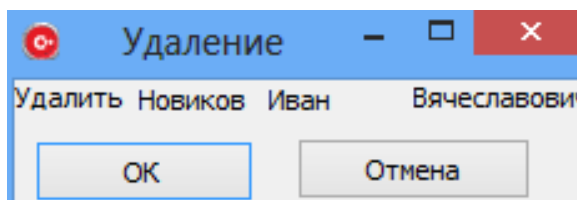


Рис. 3.17. Форма подтверждения удаления информации о сотруднике

Если пользователь подтверждает удаление выводится окно о успешном удалении и данные о сотруднике будут удалены, если нет то вернётся на окно «сотрудники». На рис. 3.18 представлена форма после удаления сотрудника «Новиков Иван Вячеславович».

Сотрудники							
№	Фамилия	Имя	Отчество	Должность	Контактный телефон	Адрес места жительства	Зарботная плата
1	Гребеник	Павел	Сергеевич	Сторож	89516347691	ул.Некрасова 78	7000
2	Сытник	Артём	Генадьевич	Грузчик	89206321475	ул.Димитрова 178	12000
3	Бабенко	Ольга	Павловна	Бухгалтер	89205775007	ул.Республиканская 148 а	25000
4	Мамонтов	Игорь	Андреевич	Водитель	89145879632	ул.Суворова 56	18000
5	Панова	Елена	Андреевна	Менеджер по п	89047891236	город Белгород улица Попова дом 46 квартира 35	14000
7	Бабенко	Андрей	Викторович	Директор	89202054676	ул. Республиканская 148 а	30000

Рис. 3.18. Форма после удаления сотрудника

Для выполнения операции редактирования сначала выбираем нужного сотрудника и нажимаем на кнопку «редактировать», которая находится в правом нижнем углу левее кнопки «удалить», выводится форма редактирования данных о сотруднике.



На рис. 3.19 представлена форма для редактирования информации о сотруднике.

Рис. 3.19. Форма для редактирования информации о сотруднике

Вносим изменения в данные о сотруднике и нажимаем на кнопку «редактировать», выводится окно о уведомлении об успешном редактировании данных о сотруднике и производится возврат на форму сотрудника.

На рис. 3.20 представлена отредактированная информация о сотруднике.

№	Фамилия	Имя	Отчество	Должность	Контактный телефон	Адрес места жительства	Зарботная плата
1	Гребеник	Павел	Сергеевич	Сторож	89516347691	ул. Некрасова 78	7000
2	Сытник	Артём	Генадьевич	Грузчик	89206321489	ул. Ушакова 47	13000
3	Бабенко	Ольга	Павловна	Бухгалтер	89205775007	ул. Республиканская 148 а	25000
4	Мамонтов	Игорь	Андреевич	Водитель	89145879632	ул. Суворова 56	18000
5	Панова	Елена	Андреевна	Менеджер по п	89047891236	город Белгород улица Попова дом 46 квартира 35	14000
7	Бабенко	Андрей	Викторович	Директор	89202054676	ул. Республиканская 148 а	30000

Рис. 3.20. Отредактированная информация о сотруднике

Выполним оформление заказа, который находится в вкладке продажи. При переходе на оформление заказа выводится форма для заполнения информации о клиенте. На рис. 3.21 представлена форма для ввода информации о клиенте.

Оформление заказа (Шаг 1)

№ Выберите фамилию Имя Отчество  
3 Бабенко Ольга Павловна

Информация о клиенте

ФИО клиента Контактный телефон  
1 Алфимова Ольга Петровна 8-999-225-69-96

Адрес  
ул. ушакова 123

Доставка  
Адрес доставки ...  
1 Доставка не нужна

Цена доставки 151

Дата приема 05.06.2017 Статус заказ На рассмотрении

Дата исполнения 05.06.2017

Далее Отмена

Рис. 3.21. Форма для ввода информации о клиенте

Сотрудник при оформлении заказа сначала выбирает свою фамилию из выпадающего списка, потом заполняет информации о сотруднике, если клиент уже оформлял заказы в магазине, то его информацию можно выбрать из списка нажав на кнопку «выбрать», иначе нужно добавить клиента. При добавлении клиента нужно нажать на кнопку «добавить» и ввести требующуюся информацию и подтвердить добавление.

На рис. 3.22 представлена форма для выбора или добавления клиента.

Клиенты

№	ФИО	Адрес	Номер телефона
1	Алфимова Ольга Петровна	ул. ушакова 123	8-999-225-69-96
2	ПЕТРОВА АНАСТАСИЯ АЛЕКСАНДРОВНА	ул.РЕСПУБЛИКАНСКАЯ 148 А	8-904-082-19-41
3	Ковтун Надежда Ивановна	ул. Красноармейская 23	8-920-845-56-56
4	Кривошеё Иван Петрович	ул. Заводская 46	8-919-965-87-45
5	Аносов Артём Николаевич	с.Берёзовка ул. Гостева 35	8-906-587-32-14

Поиск:

1 Алфимова Ольга Петровна

Добавить Редактирование Удаление

Выбрать

Рис. 3.22. Форма для выбора или добавления клиента

Таким же образом выбираем или добавляем информацию о доставке. Выбираем дату оформления заказа и дату исполнения и переходим к шагу 2 оформления заказа. Шаг 2 оформление заказа заключается в выборе нужного товара клиенту.

На рис. 3.23 представлена форма шаг 2 оформление состава заказа.

Оформление заказа (Шаг 2)

Просмотр заказа    Выбрать заказ: 1    ПЕТРОВА АНАСТАСИЯ АЛЕКСАНДРОВНА    ...    Поиск по размеру:

Информация о товаре

№	Наименование товара	Тип	Вид	Размер	Кол-во в куб/пачке	Цена за штуку	Цена за куб
1	Доска	Обрезная	Сосна	50*150*6	22	346	7600

Количество пиломатериала в штуках:     Добавить в заказ

№	NAME_TOVARA	Вид	Тип	Размер	Кол-во в куб/пачке	Наличие	Кол-во	Цена за штуку	Цена за куб
1	Доска	Сосна	Обрезная	50*150*6	22	Вналичии	100	346	7600
2	Доска	Сосна	Обрезная	25*150*6	44	Вналичии	121	178	7800
3	Доска	Сосна	Обрезная	100*100*6	16	Вналичии	89	475	7600
4	Доска	Сосна	Обрезная	150*100*6	11	Вналичии	135	691	7600
5	Доска	Сосна	Обрезная	40*150*6	27	Вналичии	56	282	7600
6	Доска	Сосна	Обрезная	40*100*6	41	Вналичии	87	186	7600
7	Доска	Сосна	Обрезная	25*100*6	66	Вналичии	156	116	7600
8	Доска	Сосна	Обрезная	50*100*6	33	Вналичии	57	231	7600
9	Доска	Сосна	Обрезная	150*150*6	7	Вналичии	40	1086	7600
10	Доска	Сосна	Полуобрезн	25*100*6	66	Вналичии	124	99	6500
11	Доска	Сосна	Полуобрезн	25*150*6	44	Вналичии	186	148	6500

Рис. 3.23 Форма шаг 2 оформление состава заказа

На форме оформление заказа шаг 2 сотрудник магазина выбирает нужный товар клиенту и формирует заказ. При формировании заказа вылетает форма подтверждения, того что товар добавлен в состав заказа. На рис. 3.24 представлена форма подтверждения, того что товар добавлен в состав заказа.

Project3

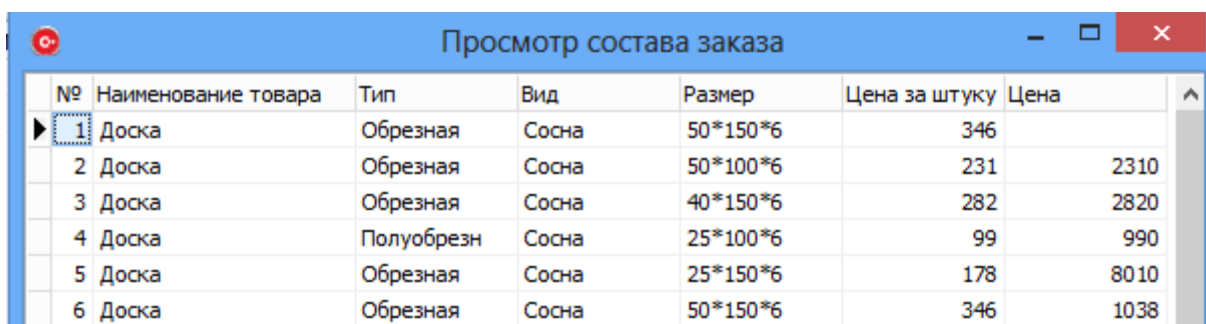
Товар добавлен в заказ

OK

Рис. 3.24. Форма подтверждения состава заказа

Так же для облегчения поиска товара реализована фильтрация в верхнем правом углу. После завершения оформления заказа можно

просмотреть состав заказа нажав на кнопку «просмотр заказа», которая находится в левом верхнем углу. На рис. 3.25 представлена форма просмотра состава заказа клиента.



№	Наименование товара	Тип	Вид	Размер	Цена за штуку	Цена
1	Доска	Обрезная	Сосна	50*150*6	346	
2	Доска	Обрезная	Сосна	50*100*6	231	2310
3	Доска	Обрезная	Сосна	40*150*6	282	2820
4	Доска	Полуобрезн	Сосна	25*100*6	99	990
5	Доска	Обрезная	Сосна	25*150*6	178	8010
6	Доска	Обрезная	Сосна	50*150*6	346	1038

Рис. 3.25. Форма для выбора или добавления клиента

После того как состав заказа набран окно оформление заказа закрывается и формируется накладная в виде отчёта и подаётся на печать. Клиент оплачивает заказ и ожидает доставки.

## ЗАКЛЮЧЕНИЕ

В настоящее время для автоматизации бизнес-процессов предприятия используются системы взаимодействия с клиентами, которые иначе называются CRM-системами. Данные системы имеют широкое применение в силу автоматизации рутинной работы сотрудников любых предприятий.

В ходе написания выпускной квалифицированной работы было создано CRM приложение для автоматизации деятельности отдела закупок и продаж предприятия «Мир Дерева».

Исходя из цели выпускной квалификационной работы был решен ряд задач, а именно:

- подробно изучена предметная область – деятельность предприятия «Мир Дерева»;
- проведен сравнительный анализ современных информационных систем, выявлены их недочеты для разработки специализированного приложения «Мир Дерева»;
- спроектирована и разработана база данных для CRM-приложения «Мир Дерева»;
- спроектирован функциональный пользовательский интерфейс программного продукта.
- проведено тестирование контрольного примера CRM – приложения.

Внедрение разработанного программного продукта «Мир Дерева» позволит автоматизировать бизнес-процессы закупок и продаж товара, в результате чего уменьшится рутинная работа сотрудников.

Разработанное CRM приложение внедрено в практическую деятельность магазина «Мир Дерева», о чем имеется акт о внедрении.

## СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Агальцов, В.П. Базы данных. В 2-х т. Т. 2. Распределенные и удаленные базы данных: Учебник / В.П. Агальцов. - Москва: ИД ФОРУМ, НИЦ ИНФРА-М, 2013. - 272 с.
2. Барабанщиков, Ю.Г. Строительные материалы и изделия / Ю.Г. Барабанщиков. - Москва: Академия (Academia), 2013. - 608 с.
3. Бейли, Л. Изучаем SQL / Л. Бейли. - Москва: Питер 2012. - 573 с.
4. Битрикс24 | Создание отдела продаж [Электронный ресурс]. – Режим доступа: [http://video.1c-bitrix.ru/b24\\_books/salesbook.pdf](http://video.1c-bitrix.ru/b24_books/salesbook.pdf) (дата обращения 19.04.2017г).
5. Бодров, О.А. Предметно-ориентированные экономические информационные системы / О.А. Бодров. – Москва: ГЛТ, 2013. - 244 с.
6. Бойтемиров, Ф.А. Конструкции из дерева и пластмасс / Ф.А. Бойтемиров. - Москва: Издательский центр «Академия», 2013 - 288 с.
7. Бондарь, А. InterBase и Firebird. Практическое руководство для умных пользователей и начинающих разработчиков / А. Бондарь. – СПб.: БХВ-Петербург, 2012. – 585 с.
8. Бухалков, М.И. Организация производства и управление предприятием: Учебник / М.И. Бухалков. - Москва: ИНФРА-М, 2013. - 506 с.
9. Бузинов, М. М. Искусство резьбы по дереву: учебное пособие / М. М. Бузинов, М. Г.Потапов. - Москва: Антиква, 2014. - 207 с.
10. Васильков, А.В. Информационные системы и их безопасность: Учебное пособие / А.В. Васильков, А.А. Васильков, И.А. Васильков. - М.: Форум, 2013. - 528 с.
11. Гаврилов, Л.П. Управление предприятием: финансовые и инвестиционные решения: Учебное пособие / Л.П. Гаврилов. - Москва: Финансы и статистика, 2012. - 184 с.

12. Голицына, О.Л. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - Москва: Форум, 2012. - 400 с.
13. Грофф Дж. Р. SQL. Полное руководство/ Дж. Р. Грофф, П.Н. Вайнберг, Э. Дж. Оппель - Москва: Вильямс, 2012. - 573 с.
14. Гусарчук, Д.М. 300 ответов любителю художественных работ по дереву / Д.М. Гусарчук. - Москва: Лесная промышленность, 2012. - 208 с.
15. Гукова, А.В. Управление предприятием: финансовые и инвестиционные решения: Курс лекций для бакалавров: Учебное пособие / А.В. Гукова, И.Д. Аникина, Р.С. Беков. - Москва: ФиС, ИНФРА, 2012. - 184 с.
16. Гукова, А.В. Управление предприятием: финансовые и инвестиционные решения: Учебное пособие / А.В. Гукова. - Москва: Финансы и статистика, 2012. - 184 с.
17. Зайцев, Н.Л. Экономика, организация и управление предприятием: Учебное пособие / Н.Л. Зайцев. - Москва: ИНФРА-М, 2012. - 455 с.
18. Кара-Ушанов, В.Ю. SQL - язык реляционных баз данных / В.Ю. Кара-Ушанов. - Москва: Питер, 2016. - 156 с.
19. Кириленко, В.С. Деревянные дома / В.С. Кириленко. - М.: АСТ, 2013. - 104 с.
20. Кудинов, А. CRM: Практика эффективного / А. Кудинов. – Москва: ИЦ РИОР, 2012. – 568 с.
21. Кудрявцев, Е.М. Организация планирование и управление предприятием: Учебник / Е.М. Кудрявцев. - Москва: АСВ, 2011. - 464 с.
22. Кудрявцев, Е.М. Организация, планирование и управление предприятием / Е.М. Кудрявцев. - Москва: АСВ, 2011. - 416 с.
23. Кузин, А.В. Базы данных: Учебное пособие для студ. высш. учеб. заведений / А.В. Кузин, С.В. Левонисова. - Москва: ИЦ Академия, 2012. - 320 с.
24. Кузин, С.В. SQL - язык реляционных баз данных / А. В. Кузин, С. В. Левонисова. - Москва: «Академия», 2016. - 359 с.

25. Никитин, А.В. Управление предприятием (фирмой) с использованием информационных систем: Учебное пособие / А.В. Никитин, И.А. Рачковская, И.В. Савченко. - Москва: ИНФРА-М, 2009. - 188 с.
26. Никитин, А.В. Управление предприятием (фирмой) с использованием информационных систем: Учебное пособие / А.В. Никитин, И.А. Рачковская, И.В. Савченко. - Москва: ИНФРА-М, 2010. - 188 с.
27. Одинцов, Б.Е. Информационные системы управления эффективностью бизнеса: Учебник и практикум для бакалавриата и магистратуры / Б.Е. Одинцов. - Люберцы: Юрайт, 2016. - 206 с.
28. Родионова, В.Н. Организация производства и управление предприятием: Учебное пособие / В.Н. Родионова, О.Г. Туровец. - Москва: ИЦ РИОР, 2005. - 128 с.
29. Рязанцев, А. Как внедрить CRM - систему за 50 дней / А. Рязанцев. - Москва: Питер, 2017. - 188 с.
30. Рязанцева, Н. 1С: Предприятие 8.0. Управление производственным предприятием. Секреты работы / Н. Рязанцева. - СПб.: ВHV, 2013. - 704 с.
31. Райхельд, Ф. Искренняя лояльность. Ключ к завоеванию клиентов на всю жизнь / Ф. Райхельд, Р. Марки. - СПб.: ВHV, 2012. - 605 с.
32. Советов, Б.Я. Базы данных: теория и практика: Учебник для бакалавров / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовской. - М.: Юрайт, 2013. - 463 с.
33. Фуфаев, Э.В. Базы данных: Учебное пособие для студентов учреждений среднего профессионального образования / Э.В. Фуфаев, Д.Э. Фуфаев. - Москва: ИЦ Академия, 2012. - 320 с.
34. Ширяев, В.И. Управление предприятием: Моделирование, анализ, управление / В.И. Ширяев, И.А. Баев, Е.В. Ширяев. - Москва: КД Либроком, 2015. - 272 с.
35. Шульц, О. Практикум по SAP. Руководство для новичков и конечных пользователей / О. Шульц. - Москва: Питер, 2015. - 456 с.



36. Embracadero RAD Studio [Электронный ресурс]. – Режим доступа: <https://www.embarcadero.com/ru/products/rad-studio> (дата обращения 05.04.2017г).

37. StorVerkCRM | Обзор продукта [Электронный ресурс]. – Режим доступа: <http://crm.storverk.ru/product.html> (дата обращения 16.04.2017г).

38. SapCRM | Обзор продукта [Электронный ресурс]. – Режим доступа: <http://sapcrm.ru/product.html> (дата обращения 15.05.2017г).

39. Asoft CRM | Обзор продукта [Электронный ресурс]. – Режим доступа: <http://rissoft.ru/catalog/detail.php?ID=65> (дата обращения 15.04.2017г).

## ПРИЛОЖЕНИЕ

### Главное окно приложения "Unit1.h"

```

#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
#include "Unit3.h"
#include "Unit4.h"
#include "Unit5.h"
#include "Unit6.h"
#include "Unit15.h"
#include "Unit17.h"
#include "Unit18.h"
#include "Unit20.h"
#include "Unit21.h"
#include "Unit22.h"
#include "Unit23.h"
#include "Unit24.h"
#include "Unit28.h"
#include "Unit30.h"

//-----
#pragma package(smart_init)
#pragma link "frxClass"
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner){ }
void __fastcall TForm1::N11Click(TObject *Sender){
    Form6->Visible=true;}
void __fastcall TForm1::N3Click(TObject *Sender){
    Form3->Visible=true;}
void __fastcall TForm1::N5Click(TObject *Sender){
    Form15->Visible=true;}
void __fastcall TForm1::pdf1Click(TObject *Sender){
    TfrxReport *Report;
    Report = Form17->frxReport1;
    Report->Preview = Form17->frxPreview1;
    Report->ShowReport();
    Form17->WindowState=wsMaximized; //Размер по всему экрану
    Form17->Show();}
void __fastcall TForm1::N4Click(TObject *Sender){
    Form18->Visible=true;}
void __fastcall TForm1::N29Click(TObject *Sender){
    Form20->Visible=true;}
void __fastcall TForm1::N6Click(TObject *Sender){
    Form21->Visible=true;}
void __fastcall TForm1::N30Click(TObject *Sender){

```

```

Form22->Visible=true;}
void __fastcall TForm1::N18Click(TObject *Sender){
Form4->Visible=true;}
void __fastcall TForm1::N7Click(TObject *Sender){
Form23->Visible=true;}
void __fastcall TForm1::N32Click(TObject *Sender){
Form24->Visible=true;}
void __fastcall TForm1::N16Click(TObject *Sender){
Form28->Visible=true;}
void __fastcall TForm1::pdf3Click(TObject *Sender){
Form30->Visible=true;}

```

### **Вкладка поставщики "Unit3.h"**

```

#include <vcl.h>
#pragma hdrstop
#include "Unit1.h"
#include "Unit3.h"
#include "Unit5.h"
#include "Unit7.h"
#include "Unit8.h"
#include "Unit10.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm3 *Form3;
__fastcall TForm3::TForm3(TComponent* Owner): TForm(Owner){}
void __fastcall TForm3::ДобавитьClick(TObject *Sender){
Form7->Visible=true;}
void __fastcall TForm3::УдалитьClick(TObject *Sender){
Form8->Visible=true;}
void __fastcall TForm3::РедактироватьClick(TObject *Sender){
Form10->Visible=true;}

```

### **Вкладка клиенты "Unit4.h"**

```

#include <vcl.h>
#pragma hdrstop
#include "Unit4.h"
#include "Unit5.h"
#include "Unit14.h"
#include "Unit28.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm4 *Form4;
__fastcall TForm4::TForm4(TComponent* Owner): TForm(Owner){}
void __fastcall TForm4::Button3Click(TObject *Sender){
Form14->Visible=true;}
void __fastcall TForm4::Edit1Change(TObject *Sender){
DataModule5->KLIENT->Active=false;
DataModule5->KLIENT->Filtered=false;
DataModule5->KLIENT->Filter="FIO LIKE "+Form4->Edit1->Text+"% ";
DataModule5->KLIENT->Active=true;
DataModule5->KLIENT->Filtered=true;}

```

```
void __fastcall TForm4::Button4Click(TObject *Sender){
Form28->Visible=true;}
void __fastcall TForm4::Button5Click(TObject *Sender){
Close();}
```

### **Вкладка сотрудники "Unit6.h"**

```
#include <vcl.h>
#pragma hdrstop
#include "Unit6.h"
#include "Unit5.h"
#include "Unit11.h"
#include "Unit12.h"
#include "Unit13.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm6 *Form6;
__fastcall TForm6::TForm6(TComponent* Owner)
: TForm(Owner){}
void __fastcall TForm6::Button1Click(TObject *Sender){
Form11->Visible=true;}
void __fastcall TForm6::Button2Click(TObject *Sender){
Form12->Visible=true;}
void __fastcall TForm6::Button3Click(TObject *Sender){
Form13->Visible=true;}
void __fastcall TForm6::Edit1Change(TObject *Sender){
DataModule5->SOTRUDNIK->Active=false;
DataModule5->SOTRUDNIK->Filtered=false;
DataModule5->SOTRUDNIK->Filter="FAMILIA LIKE '"+Form6->Edit1->Text+"% ";
DataModule5->SOTRUDNIK->Active=true;
DataModule5->SOTRUDNIK->Filtered=true;}
```

### **Вкладка доставки "Unit22.h"**

```
#include <vcl.h>
#pragma hdrstop
#include "Unit22.h"
#include "Unit5.h"
#include "Unit25.h"
#include "Unit26.h"
#include "Unit28.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm22 *Form22;
__fastcall TForm22::TForm22(TComponent* Owner)
: TForm(Owner){}
void __fastcall TForm22::Button2Click(TObject *Sender){
Form25->Visible=true;}
void __fastcall TForm22::Button3Click(TObject *Sender){
Form26->Visible=true;}
void __fastcall TForm22::Button1Click(TObject *Sender){
Form28->Visible=true;}
```

```

Close();}
void __fastcall TForm22::Edit1Change(TObject *Sender){
DataModule5->DOSTAVKA->Active=false;
DataModule5->DOSTAVKA->Filtered=false;
DataModule5->DOSTAVKA->Filter="ADRES_DOSTAVKI LIKE '"+Form22->Edit1-
>Text+"%";
DataModule5->DOSTAVKA->Active=true;
DataModule5->DOSTAVKA->Filtered=true;}

```

## Вкладка оформление заказа шаг 1 "Unit28.h"

```

#include <vcl.h>
#pragma hdrstop
#include "Unit28.h"
#include "Unit27.h"
#include "Unit5.h"
#include "Unit4.h"
#include "Unit22.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm28 *Form28;
__fastcall TForm28::TForm28(TComponent* Owner)
: TForm(Owner)
{DBLookupComboBox1->KeyValue=DBLookupComboBox1->ListSource->DataSet-
>FieldName(DBLookupComboBox1->KeyField)->Value;}
void __fastcall TForm28::Button1Click(TObject *Sender){
try{
DataModule5->INSERT_ZAKAZ->ParamByName("ID_SOTRUDNIK")-
>AsString=Form28->DBText1->Caption;
DataModule5->INSERT_ZAKAZ->ParamByName("ID_DOSTAVKA")-
>AsString=Form28->DBText2->Caption;
DataModule5->INSERT_ZAKAZ->ParamByName("ID_KLIENT")->AsString=Form28-
>DBText3->Caption;
DataModule5->INSERT_ZAKAZ->ParamByName("DATA_ZAKAZA")-
>AsString=Form28->DateTimePicker1->Date;
DataModule5->INSERT_ZAKAZ->ParamByName("DATA_ISPOLNENIE")-
>AsString=Form28->DateTimePicker3->Date;
DataModule5->INSERT_ZAKAZ->ParamByName("STATUS_ZAKAZA")-
>AsString=Form28->Edit6->Text;
DataModule5->INSERT_ZAKAZ->ParamByName("SUMMA_ZAKAZA")-
>AsString="0";
DataModule5->INSERT_ZAKAZ->Prepare();
DataModule5->INSERT_ZAKAZ->ExecProc();
DataModule5->ZAKAZ->Active=false;
DataModule5->ZAKAZ->Active=true;}
catch (EIBInterBaseError *x){
ShowMessage(x->Message);}
Form27->Visible=true;
Close();}
void __fastcall TForm28::Button2Click(TObject *Sender){
Close();}
void __fastcall TForm28::Button3Click(TObject *Sender){
Form4->Visible=true;}

```

```
void __fastcall TForm28::Button4Click(TObject *Sender){
Form22->Visible=true;}
```

## **Вкладка оформление заказа шаг 2 "Unit27.h"**

```
#include <vcl.h>
#pragma hdrstop
#include "Unit27.h"
#include "Unit5.h"
#include "Unit16.h"
#include "Unit15.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm27 *Form27;
__fastcall TForm27::TForm27(TComponent* Owner)
: TForm(Owner){ }
void __fastcall TForm27::Button1Click(TObject *Sender){
Form16->Visible=true; }
void __fastcall TForm27::Edit2Change(TObject *Sender){
DataModule5->PRICE_LIST->Active=false;
DataModule5->PRICE_LIST->Filtered=false;
DataModule5->PRICE_LIST->Filter="RAZMER LIKE '"+Form27->Edit2->Text+"%";
DataModule5->PRICE_LIST->Active=true;
DataModule5->PRICE_LIST->Filtered=true;}
void __fastcall TForm27::Button2Click(TObject *Sender){
float a,b,c;
a=StrToFloat(Form27->Edit1->Text);
b=StrToFloat(Form27->DBText7->Caption);
c=a*b;
try{
DataModule5->INSERT_PRED_ZAKAZ->ParamByName("ID_ZAKAZ")-
>AsString=Form27->DBText6->Caption;
DataModule5->INSERT_PRED_ZAKAZ->ParamByName("ID_PRICE_LIST")-
>AsString=Form27->DBText1->Caption;
DataModule5->INSERT_PRED_ZAKAZ->ParamByName("KOL_VO_TOVARA")-
>AsString=Form27->Edit1->Text;
DataModule5->INSERT_PRED_ZAKAZ->ParamByName("PRICE_TOVARA")-
>AsString=c;
DataModule5->INSERT_PRED_ZAKAZ->Prepare();
DataModule5->INSERT_PRED_ZAKAZ->ExecProc();
DataModule5->ZAKAZ->Active=false;
DataModule5->ZAKAZ->Active=true;
ShowMessage("Товар добавлен в заказ");}
catch (EIBInterBaseError *x){
ShowMessage(x->Message);} }
void __fastcall TForm27::Button3Click(TObject *Sender){
Form15->Visible=true;}
```

## **Редактирование данных вкладки доставка "Unit25.h"**

```
#include <vcl.h>
#pragma hdrstop
#include "Unit25.h"
```

```

#include "Unit5.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm25 *Form25;
__fastcall TForm25::TForm25(TComponent* Owner)
: TForm(Owner){}
void __fastcall TForm25::Button1Click(TObject *Sender){
try{
DataModule5->UP_DOSTAVKA->ParamByName("ID_DOSTAVKA")-
>AsString=Form25->DBText1->Caption;
DataModule5->UP_DOSTAVKA->ParamByName("ADRES_DOSTAVKI")-
>AsString=Form25->DBEdit1->Text;
DataModule5->UP_DOSTAVKA->ParamByName("SUMMA_DOSTAVKI")-
>AsString=Form25->DBEdit2->Text;
DataModule5->UP_DOSTAVKA->Prepare();
DataModule5->UP_DOSTAVKA->ExecProc();
DataModule5->DOSTAVKA->Active=false;
DataModule5->DOSTAVKA->Active=true;
ShowMessage("Запись успешно отредактирована!");}
catch (EIBInterBaseError *x){
ShowMessage(x->Message);}
Close();}
void __fastcall TForm25::Button2Click(TObject *Sender){
Form25->DBEdit1->Text="";
Form25->DBEdit2->Text="";}

```

### **Удаление данных из вкладки доставка "Unit26.h"**

```

#include <vcl.h>
#pragma hdrstop
#include "Unit26.h"
#include "Unit22.h"
#include "Unit5.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm26 *Form26;
__fastcall TForm26::TForm26(TComponent* Owner)
: TForm(Owner){}
void __fastcall TForm26::Button1Click(TObject *Sender){
try {
Form26->DBText1->Caption=Form22->DBText1->Caption;
DataModule5->UP_DEL_ZAK->ParamByName("ID_DOSTAVKA1")-
>AsString=Form26->DBText1->Caption;
Form26->DBText1->Caption="1";
DataModule5->UP_DEL_ZAK->ParamByName("ID_DOSTAVKA")-
>AsString=Form26->DBText1->Caption;
DataModule5->UP_DEL_ZAK->Prepare();
DataModule5->UP_DEL_ZAK->ExecProc();
DataModule5->ZAKAZ->Active=false;
DataModule5->ZAKAZ->Active=true;
DataModule5->DELETE_DOSTAVKA->ParamByName("ADRES_DOSTAVKI")-
>AsString=Form26->DBText2->Caption;

```

```

DataModule5->DELETE_DOSTAVKA->ParamByName("SUMMA_DOSTAVKI")-
>AsString=Form26->DBText3->Caption;
DataModule5->DELETE_DOSTAVKA->Prepare();
DataModule5->DELETE_DOSTAVKA->ExecProc();
DataModule5->DOSTAVKA->Active=false;
DataModule5->DOSTAVKA->Active=true;
ShowMessage("Запись успешно удалена!");
Close();}
catch (EIBInterBaseError *x){
ShowMessage(x->Message);} }
void __fastcall TForm26::Button2Click(TObject *Sender){
Close();}

```

### Добавление данных в вкладку сотрудники "Unit11.h"

```

#include <vcl.h>
#pragma hdrstop
#include "Unit11.h"
#include "Unit5.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm11 *Form11;
__fastcall TForm11::TForm11(TComponent* Owner)
: TForm(Owner){ }
void __fastcall TForm11::Button1Click(TObject *Sender){
try{
DataModule5->INSERT_SOTRUDNIK->ParamByName("FAMILIA")-
>AsString=Form11->Edit1->Text;
DataModule5->INSERT_SOTRUDNIK->ParamByName("NAME")-
>AsString=Form11->Edit2->Text;
DataModule5->INSERT_SOTRUDNIK->ParamByName("OTCHESTVO")-
>AsString=Form11->Edit3->Text;
DataModule5->INSERT_SOTRUDNIK->ParamByName("ADRES")-
>AsString=Form11->Edit4->Text;
DataModule5->INSERT_SOTRUDNIK->ParamByName("PHONE_SOTRUDNIKA")-
>AsString=Form11->Edit5->Text;
DataModule5->INSERT_SOTRUDNIK->ParamByName("DOLGNOST")-
>AsString=Form11->Edit6->Text;
DataModule5->INSERT_SOTRUDNIK->ParamByName("ZARPLATA")-
>AsString=Form11->Edit7->Text;
DataModule5->INSERT_SOTRUDNIK->Prepare();
DataModule5->INSERT_SOTRUDNIK->ExecProc();
DataModule5->SOTRUDNIK->Active=false;
DataModule5->SOTRUDNIK->Active=true;
ShowMessage("Запись успешно добавлена!");
Form11->Edit1->Text="";
Form11->Edit2->Text="";
Form11->Edit3->Text="";
Form11->Edit4->Text="";
Form11->Edit5->Text="";
Form11->Edit6->Text="";
Form11->Edit7->Text="";}

```



```

catch (EIBInterBaseError *x){
ShowMessage(x->Message);} }
void __fastcall TForm11::Button2Click(TObject *Sender){
Form11->Edit1->Text="";
Form11->Edit2->Text="";
Form11->Edit3->Text="";
Form11->Edit4->Text="";
Form11->Edit5->Text="";
Form11->Edit6->Text="";
Form11->Edit7->Text="";}

```

## Редактирование данных вкладки сотрудники "Unit12.h"

```

#include <vcl.h>
#pragma hdrstop
#include "Unit12.h"
#include "Unit6.h"
#include "Unit5.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm12 *Form12;
__fastcall TForm12::TForm12(TComponent* Owner)
: TForm(Owner){ }
void __fastcall TForm12::Button1Click(TObject *Sender){
try{
DataModule5->UP_SOTRUDNIK->ParamByName("ID_SOTRUDNIK")-
>AsString=Form12->DBText1->Caption;
DataModule5->UP_SOTRUDNIK->ParamByName("FAMILIA")->AsString=Form12-
>DBEdit1->Text;
DataModule5->UP_SOTRUDNIK->ParamByName("NAME")->AsString=Form12-
>DBEdit2->Text;
DataModule5->UP_SOTRUDNIK->ParamByName("OTCHESTVO")-
>AsString=Form12->DBEdit3->Text;
DataModule5->UP_SOTRUDNIK->ParamByName("PHONE_SOTRUDNIKA")-
>AsString=Form12->DBEdit4->Text;
DataModule5->UP_SOTRUDNIK->ParamByName("ADRES")->AsString=Form12-
>DBEdit5->Text;
DataModule5->UP_SOTRUDNIK->ParamByName("DOLGNOST")-
>AsString=Form12->DBEdit6->Text;
DataModule5->UP_SOTRUDNIK->ParamByName("ZARPLATA")-
>AsString=Form12->DBEdit7->Text;
DataModule5->UP_SOTRUDNIK->Prepare();
DataModule5->UP_SOTRUDNIK->ExecProc();
DataModule5->SOTRUDNIK->Active=false;
DataModule5->SOTRUDNIK->Active=true;
ShowMessage("Запись успешно отредактирована!");}
catch (EIBInterBaseError *x){
ShowMessage(x->Message);}
Close();}
void __fastcall TForm12::Button2Click(TObject *Sender){
Form12->DBEdit1->Text="";
Form12->DBEdit2->Text="";
Form12->DBEdit3->Text="";

```

```
Form12->DBEdit4->Text="";
Form12->DBEdit5->Text="";
Form12->DBEdit6->Text="";
Form12->DBEdit7->Text="";}
```

### Удаление данных из вкладки сотрудники "Unit13.h"

```
#include <vcl.h>
#pragma hdrstop
#include "Unit13.h"
#include "Unit5.h"
#include "Unit6.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm13 *Form13;
__fastcall TForm13::TForm13(TComponent* Owner)
: TForm(Owner){}
void __fastcall TForm13::Button1Click(TObject *Sender){
try {
Form13->DBText2->Caption=Form6->DBEdit1->Text;
Form13->DBText3->Caption=Form6->DBEdit2->Text;
Form13->DBText4->Caption=Form6->DBEdit3->Text;
DataModule5->DELETE_SOTRUDNIK->ParamByName("FAMILIA")-
>AsString=Form13->DBText2->Caption;
DataModule5->DELETE_SOTRUDNIK->ParamByName("NAME")-
>AsString=Form13->DBText3->Caption;
DataModule5->DELETE_SOTRUDNIK->ParamByName("OTCHESTVO")-
>AsString=Form13->DBText4->Caption;
DataModule5->DELETE_SOTRUDNIK->Prepare();
DataModule5->DELETE_SOTRUDNIK->ExecProc();
DataModule5->SOTRUDNIK->Active=false;
DataModule5->SOTRUDNIK->Active=true;
ShowMessage("Запись успешно удалена!");
Close();}
catch (EIBInterBaseError *x){
ShowMessage(x->Message);} }
void __fastcall TForm13::Button2Click(TObject *Sender){
Close();}
```

### Добавление данных в вкладку поставщики "Unit13.h"

```
#include <vcl.h>
#pragma hdrstop
#include "Unit7.h"
#include "Unit5.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm7 *Form7;
__fastcall TForm7::TForm7(TComponent* Owner)
: TForm(Owner){
Form7->Memo1->Text="";}
void __fastcall TForm7::Button1Click(TObject *Sender){
try{
```

```

DataModule5->INSERT_POSTAVSHIK-
>ParamByName("NAZVANIE_POSTAVSHIKA")->AsString=Form7->Edit1->Text;
DataModule5->INSERT_POSTAVSHIK->ParamByName("INN")->AsString=Form7-
>Edit2->Text;
DataModule5->INSERT_POSTAVSHIK->ParamByName("KPP")->AsString=Form7-
>Edit3->Text;
DataModule5->INSERT_POSTAVSHIK->ParamByName("OGRN")->AsString=Form7-
>Edit4->Text;
DataModule5->INSERT_POSTAVSHIK->ParamByName("ADRES")-
>AsString=Form7->Memo1->Text;
DataModule5->INSERT_POSTAVSHIK->ParamByName("KONTAKTNOE_LICO")-
>AsString=Form7->Edit5->Text;
DataModule5->INSERT_POSTAVSHIK->ParamByName("E_MAIL")-
>AsString=Form7->Edit6->Text;
DataModule5->INSERT_POSTAVSHIK-
>ParamByName("PHONE_POSTAVSHIKA")->AsString=Form7->Edit7->Text;
DataModule5->INSERT_POSTAVSHIK->Prepare();
DataModule5->INSERT_POSTAVSHIK->ExecProc();
DataModule5->POSTAVSHIK->Active=false;
DataModule5->POSTAVSHIK->Active=true;
ShowMessage("Запись успешно добавлена!");
Form7->Edit1->Text="";
Form7->Edit2->Text="";
Form7->Edit3->Text="";
Form7->Edit4->Text="";
Form7->Memo1->Text="";
Form7->Edit5->Text="";
Form7->Edit6->Text="";
Form7->Edit7->Text="";}
catch (EIBInterBaseError *x){
ShowMessage(x->Message);} }
void __fastcall TForm7::Button2Click(TObject *Sender){
Form7->Edit1->Text="";
Form7->Edit2->Text="";
Form7->Edit3->Text="";
Form7->Edit4->Text="";
Form7->Memo1->Text="";
Form7->Edit5->Text="";
Form7->Edit6->Text="";
Form7->Edit7->Text="";}

```

### **Редактирование данных вкладки поставщика "Unit13.h"**

```

#include <vcl.h>
#pragma hdrstop
#include "Unit10.h"
#include "Unit5.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm10 *Form10;
__fastcall TForm10::TForm10(TComponent* Owner)
: TForm(Owner){}
void __fastcall TForm10::Button1Click(TObject *Sender){

```

```

try{
    DataModule5->UP_POSTAVSHIK->ParamByName("ID_POSTAVSHIK")-
>AsString=Form10->DBText1->Caption;
    DataModule5->UP_POSTAVSHIK->ParamByName("NAZVANIE_POSTAVSHIKA")-
>AsString=Form10->DBEdit1->Text;
    DataModule5->UP_POSTAVSHIK->ParamByName("INN")->AsString=Form10-
>DBEdit2->Text;
    DataModule5->UP_POSTAVSHIK->ParamByName("KPP")->AsString=Form10-
>DBEdit3->Text;
    DataModule5->UP_POSTAVSHIK->ParamByName("OGRN")->AsString=Form10-
>DBEdit4->Text;
    DataModule5->UP_POSTAVSHIK->ParamByName("ADRES")->AsString=Form10-
>DBMemo1->Text;
    DataModule5->UP_POSTAVSHIK->ParamByName("KONTAKTNOE_LICO")-
>AsString=Form10->DBEdit5->Text;
    DataModule5->UP_POSTAVSHIK->ParamByName("E_MAIL")->AsString=Form10-
>DBEdit6->Text;
    DataModule5->UP_POSTAVSHIK->ParamByName("PHONE_POSTAVSHIKA")-
>AsString=Form10->DBEdit7->Text;
    DataModule5->UP_POSTAVSHIK->Prepare();
    DataModule5->UP_POSTAVSHIK->ExecProc();
    DataModule5->POSTAVSHIK->Active=false;
    DataModule5->POSTAVSHIK->Active=true;
    ShowMessage("Запись успешно отредактирована!");}
catch (EIBInterBaseError *x){
    ShowMessage(x->Message);}
Close();}
void __fastcall TForm10::Button2Click(TObject *Sender){
    Form10->DBEdit1->Text="";
    Form10->DBEdit2->Text="";
    Form10->DBEdit3->Text="";
    Form10->DBEdit4->Text="";
    Form10->DBEdit5->Text="";
    Form10->DBEdit6->Text="";
    Form10->DBEdit7->Text="";}

```

### Удаление данных из вкладки поставщика "Unit13.h"

```

#include <vcl.h>
#pragma hdrstop
#include "Unit8.h"
#include "Unit5.h"
#include "Unit3.h"
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm8 *Form8;
__fastcall TForm8::TForm8(TComponent* Owner)
: TForm(Owner){}
void __fastcall TForm8::Button1Click(TObject *Sender){
try {
    Form8->DBText1->Caption=Form3->DBEdit1->Text;
    Form8->DBText2->Caption=Form3->DBText1->Caption;

```

```
DataModule5->DELETE_POSTAVSHIK->ParamByName("ID")->AsString=Form8-  
>DBText2->Caption;  
DataModule5->DELETE_POSTAVSHIK->Prepare();  
DataModule5->DELETE_POSTAVSHIK->ExecProc();  
DataModule5->POSTAVSHIK->Active=false;  
DataModule5->POSTAVSHIK->Active=true;  
ShowMessage("Запись успешно удалена!");  
Close();}  
catch (EIBInterBaseError *x){  
ShowMessage(x->Message);}  
void __fastcall TForm8::Button2Click(TObject *Sender){  
Close();}
```

Выпускная квалификационная работа выполнена мной самостоятельно.  
Все использованные в работе материалы и концепции из опубликованной научной литературы и других источников имеют ссылки на них.

«\_\_\_» \_\_\_\_\_ Г.

---

*(подпись)*

---

*(Ф.И.О.)*