

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 ПРИКЛАДНОЙ ИНФОРМАТИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ

**РАЗРАБОТКА АРМ МЕНЕДЖЕРА АВТОТРАНСПОРТНОГО
ПРЕДПРИЯТИЯ**

Выпускная квалификационная работа

обучающегося по направлению подготовки 09.03.03 Прикладная
информатика в экономике
заочной формы обучения, группы 07001251
Долженко Инны Ивановны

Научный руководитель
доцент, к.ф.-м.н
Беленко В. А.

БЕЛГОРОД 2017

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1 Технико-экономическая характеристика предметной области.....	6
1.1 Изучение предметной области	6
1.1.1 Характеристика организации.....	6
1.1.2 Краткая характеристика отдела продаж.....	7
1.2 Постановка задачи исследования.....	9
1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи.....	10
1.4 Постановка задачи	17
1.5 Анализ существующих разработок.....	19
2 Обоснование проектных решений по автоматизированному решению экономико-информационных задач.	24
2.1 Обоснование проектных решений по техническому обеспечению	24
2.2 Обоснование проектных решений по информационному обеспечению... ..	26
2.3 Обоснование проектных решений по программному обеспечению.....	28
2.4 Обоснование проектных решений по технологическому обеспечению....	35
3 Разработка проекта АРМ менеджера транспортного предприятия	37
3.1 Информационное обеспечение задачи (комплекса задач, АРМ).....	37
3.1.1 Используемые классификаторы и системы кодирования.....	37
3.1.2 Характеристика базы данных	37
3.1.2.1 Характеристика инфологической модели БД	37
3.1.2.2 Характеристика даталогической модели БД.....	41
3.1.3 Характеристика результатной информации.....	43
3.1.3.1 Характеристика таблиц с результатной информацией.....	43
3.1.3.2 Характеристика результатных документов.....	45
3.2 Программное обеспечение задачи (комплекса задач, АРМ).....	47
3.3 Технологическое обеспечение задачи (комплекса задач, АРМ)	49
3.4 Описание контрольного примера реализации проекта.....	52

3.5 Целесообразность разработки с экономической точки зрения.....	55
3.6 SWOT-анализ разработки	56
3.7 Калькуляция себестоимости научно-технической продукции	59
ЗАКЛЮЧЕНИЕ	63
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ	65
ПРИЛОЖЕНИЕ. ТЕКСТ ОСНОВНЫХ ПРОГРАММНЫХ МОДУЛЕЙ.....	68

ВВЕДЕНИЕ

Автоматизированная система представляет собой систему, состоящую из персонала и комплекса средств автоматизации его деятельности, реализующую информационную технологию выполнения установленных функций.

Основной целью создания автоматизированной системы является категоризация и стандартизация автоматизируемого процесса, что позволяет обеспечивать стабильность работы системы, прозрачность её контроля и анализа слабых мест и основания для её развития либо свёртывания.

Если деятельность организаций правильно автоматизирована, то она в достаточной мере упрощает принятие решений и уменьшает время, которое требуется для решения проблем руководителям любого уровня.

Предметом данной выпускной квалификационной работы является деятельность общества с ограниченной ответственностью «Транспортно-экспедиционная компания Белгорода». Общество с ограниченной ответственностью «Транспортно-экспедиционная компания Белгорода» оказывает транспортно-экспедиционные услуги по организации грузоперевозок автомобильным транспортом по территории России, странам СНГ и Европы. Данная компания довольно успешно работает на рынке грузоперевозок с 2012 года.

Работникам ООО «Транспортно-экспедиционная компания Белгорода» приходится работать с большим объемом информации о клиентах, о заказах клиентов, о маршрутах перевозок и перевозимых грузах. Следует отметить, что используемая в процессе работы ООО «Транспортно-экспедиционная компания Белгорода» информация хорошо поддается структурированию, что, в свою очередь, позволяет эффективно использовать персональный компьютер для обработки и хранения этих данных.

Целью выпускной квалификационной работы является повышение эффективности работы автотранспортного предприятия за счет внедрения

разработанного автоматизированного рабочего места менеджера автотранспортного предприятия ООО «Транспортная-экспедиционная компания Белгорода»

Разрабатываемая информационная система будет реализована в отделе продаж ООО «Транспортно-экспедиционная компания Белгорода». Человек, занимающийся нахождением и привлечением новых клиентов, приемом заказов на перевозку грузов, анализом рынка и формированием общего имиджа компании будет являться пользователем этой системы.

Для автоматизации учета заказов, ведения базы данных по клиентам, водителям и грузам, необходимо решить следующие задачи:

- Провести анализ предметной области;
- Разработать требования АРМ менеджера автотранспортного предприятия;
- Обосновать выбор инструментальных средств;
- Реализовать АРМ менеджера автотранспортного предприятия;
- Обосновать экономическую эффективность внедрения результатов проекта.

Данная выпускная квалификационная работа состоит из введения, трех разделов, заключения, списка использованной литературы и приложения.

1 Технико-экономическая характеристика предметной области

1.1 Изучение предметной области

1.1.1 Характеристика организации

Объектом данной выпускной квалификационной работы является деятельность общества с ограниченной ответственностью «Транспортно-экспедиционная компания Белгорода».

ООО «Транспортно-экспедиционная компания Белгорода» осуществляет перевозку и экспедирование грузов автомобильным транспортом как по территории России, так и по странам СНГ. По территории России грузовые перевозки осуществляются между такими городами как Белгород, Воронеж, Москва, Санкт-Петербург, Екатеринбург, Самара, Смоленск, Уфа, Нижний Новгород, Ростов на Дону, Краснодар, Казань, Иркутск, Мурманск, Курск и др.

ООО «Транспортно-экспедиционная компания Белгорода» предоставляет услуги по доставке различных видов грузов по территории России, начиная от организации разовых поставок небольших партий товаров до реализации крупномасштабных проектов по перевозке больших партий груза, а также негабаритных или тяжеловесных грузов, с использованием различных видов транспорта, на долгосрочной основе или единовременно.[3]

ООО «Транспортно-экспедиционная компания Белгорода» имеет огромную базу автомобильного транспорта, что позволяет решать автотранспортные задачи любой сложности. Сотрудники компании подбирают заказчику нужный автомобиль и предлагают оптимальную технологию перевозки.[11]

В распоряжении компании ООО «Транспортно-экспедиционная компания Белгорода» находятся автомобили зарубежных и отечественных марок (INTER, IVECO, VOLVO, MAN, MERSEDES BENZ, КАМАЗ, ГАЗ):

1) полуприцепы, рефрижераторы и изотермические прицепы и (грузоподъемность 10-22 тонн, полезный объем 30-96м³);

2) тентованные прицепы и полуприцепы (грузоподъемность 10-25 тонн, полезный объем 30-120м³);

3) малотоннажные автомобили (грузоподъемность от 0,8 до 3,5 тонн);

4) бортовые автомобили (грузоподъемность до 30 тонн).

При организации работ по логистике сотрудники организации ООО «Транспортно-экспедиционная компания Белгорода» предлагают следующие услуги:[9]

1) работы по загрузке товаров и выгрузке товаров;

2) выбор наиболее рационального варианта перевозки, доставка груза в согласованные сроки;

3) страхование груза;

4) отслеживание передвижения груза , оповещение заказчика о каждой стадии выполненных работ ;

5) сопровождение груза и при необходимости – предоставление охраны;

6) подготовка всей сопроводительной документации.

Структурные подразделения ООО «Транспортно-экспедиционная компания Белгорода» будут рассмотрены в следующем подразделе выпускной квалификационной работы. [24,36]

1.1.2 Краткая характеристика отдела продаж

Схема организационной структуры ООО «Транспортно-экспедиционная компания Белгорода» представлена на рисунке 1.1.

Функции нахождения и привлечения новых клиентов, организации работы с имеющимися клиентами, приемом заказов на перевозку грузов, оформление договоров на оказание транспортно-экспедиционных услуг, анализ рынка, организацию рекламы и формирование общего имиджа компании выполняет отдел продаж [15].

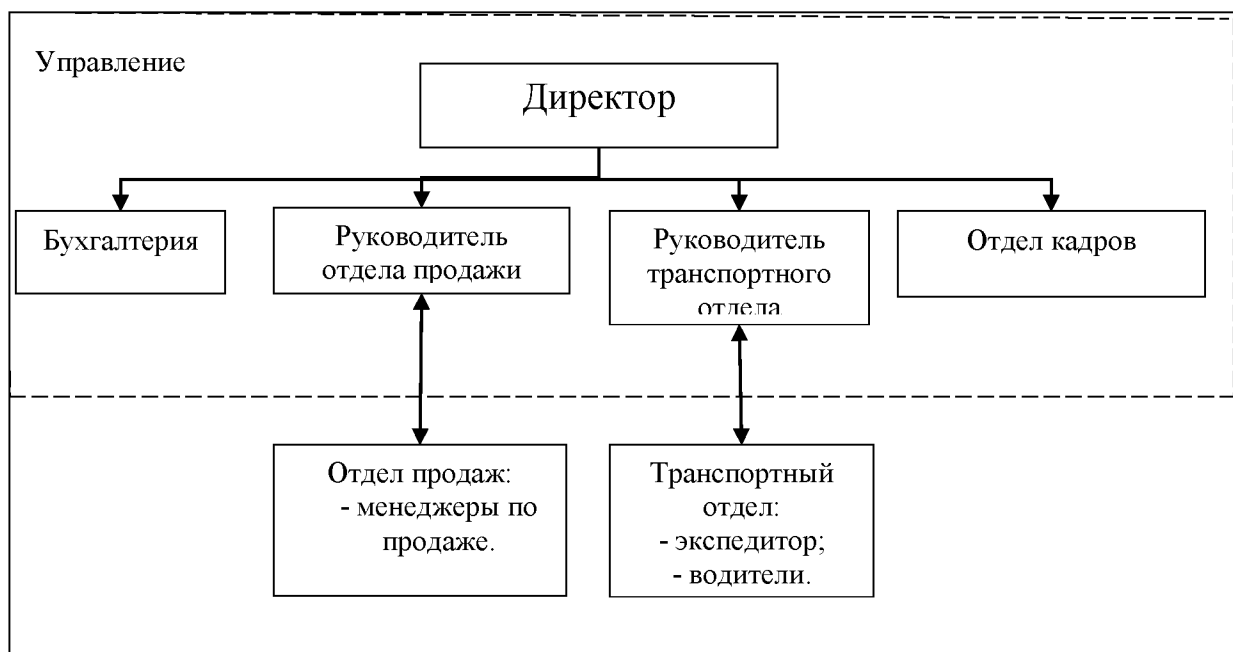


Рисунок 1.1 - Организационная структура ООО «Транспортно-экспедиционная компания Белгорода»

Функции транспортного отдела заключаются в следующем:

- Транспортное обеспечение деятельности предприятия. Разработка годовых, квартальных, месячных и оперативных планов-графиков автотранспортных грузоперевозок на основе планов получения материалов и сырья, а также отгрузки готовой продукции;
- Разработка маршрутов движения;
- Организация централизованных межорганизационных и внутриорганизационных перевозок;
- Определение потребности и производство расчетов на необходимые организации автотранспортные средства, загрузочно-отгрузочное оборудование, оборудование для ремонта и запчастей;
- Утверждение графиков работ, объемов перевозок с учетом конкретных условий, формирование грузопотоков, составов бригад, бригадных заданий;
- Передача материально-технических ресурсов на склады предприятия и передача готовой продукции на склады покупателей;

- Рациональное использование транспортных средств в соответствии с установленными нормами их грузоподъемности и вместимости;
- Проверка сохранности упаковки и наличия вложений в соответствии с сопроводительной документацией, составление в установленных случаях актов на обнаруженную недостачу или порчу грузов, документов;
- Ведение планов-графиков о наличии транспортных средств под загрузкой и отгрузкой по периодам суток, о работе механизмов;
- Справочно-информационная работа о поступающих и отправляемых грузах, сроках выполнения доставок, условия перевозки груза, загрузочно-отгрузочных и коммерческих операций [29].

Разработка будет осуществляться для отдела продаж.

ООО «Транспортно-экспедиционная компания Белгорода» располагается по адресу 308019, РФ, Белгородская обл., г. Белгород, ул. Магистральная 55Г

1.2 Постановка задачи исследования

Задачей данной выпускной квалификационной работы является разработка автоматизированного рабочего места менеджера автотранспортного предприятия. Место решения задачи: отдел продаж ООО «Транспортно-экспедиционная компания Белгорода».

Одна из основных задач менеджера по продажам в автотранспортном предприятии – это четкое соблюдение технологии перевозки. Однако в его обязанности входит решать и другие вопросы, а именно: менеджер является связующим звеном между сотрудниками компании и ее клиентами. Он должен иметь четкое представление о задаче, поставленной перед ним, и уметь донести сущность задачи до работников компании. От того, насколько квалифицированно менеджер грузоперевозок справится со своими обязанностями, зависит, будет ли клиент и далее пользоваться услугами данной транспортной компании [12].

Помимо приема и обработки заказов на грузоперевозки, менеджер в ООО «Транспортно-экспедиционная компания Белгорода» решает следующие задачи:

- Выбирает маршрут доставки;
- Рассчитывает стоимость заказа;
- Подбирает подходящее транспортное средство;
- Подбирает штат сотрудников, подходящих для выполнения заказа;
- Осуществляет мониторинг подготовки, перевозки и доставки груза.

Таким образом, в работе отдела продаж существует много однообразной работы, которую возможно автоматизировать.

Хранение документов в классическом виде, то есть в виде бумаг, безусловно, менее рационально, чем хранение их в электронном виде. При хранении документации на диске значительно упрощается поиск необходимого заказа, появляется возможность сохранять данные за неограниченный промежуток времени и не путаться в них, облегчается изменение любых документов, составление большого количества квитанций. Поэтому задача выпускной квалификационной работы является важной и неотъемлемой частью эффективного управления автотранспортным предприятием [31].

Результатными документами АРМ менеджера автотранспортного предприятия являются:

- Отчет по клиентам;
- Отчеты по заказам.

1.3 Обоснование необходимости и цели использования вычислительной техники для решения задачи

В данной выпускной квалификационной работе решается задача автоматизации деятельности специалиста по приему заказов в автотранспортном предприятии [13].

В технологическом процессе данной задачи можно выделить три этапа. Первый этап заключается в сборе, регистрации, передачи данных о клиентах, грузах и маршрутах для дальнейшей обработки.

Второй этап заключается в проектировании. Для детального представления имеющейся технологии деятельности предприятия предлагается рассмотреть ее с использованием функциональных диаграмм с применением методологии IDEF0.

IDEF0 — Function Modeling — методология функционального моделирования и графическая нотация, которая представляет возможность формализованного представления при описании бизнес-процессов. Преимущество IDEF0 обусловлено тем, что в ней делается акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность (WorkFlow).

Стандарт IDEF0 рассматривает предприятие как набор функций, причем имеется некое правило, согласно которому наиболее значимая функция находится в верхнем левом углу. Так же существует правило стороны, согласно которому стрелка входа всегда приходит в левую кромку активности, — стрелка управления — в верхнюю кромку, — стрелка механизма — нижняя кромка, — стрелка выхода — правая кромка [10].

Описание представляется как «чёрный ящик» с входами, выходами, управлением и механизмом. Такое представление позволяет постепенно осуществить детализацию до необходимого уровня. Кроме того имеются словари описания активностей и стрелок для правильного представления и понимания бизнес-процессов. При формировании таких словарей имеется возможность описания смысла, который вкладывается в определенную активность либо стрелку.

При организации бизнес-проектов, которые в своей основе предполагают моделирование различных организационных или административных процессов данная модель может применяться довольно успешно.

Регламентированные нормы, а так же правила и должностные инструкции работников определяют всю деятельность предприятия. На рисунке 1.2 представлена структура фирмы, персонал которой осуществляет всю деятельность предприятия.

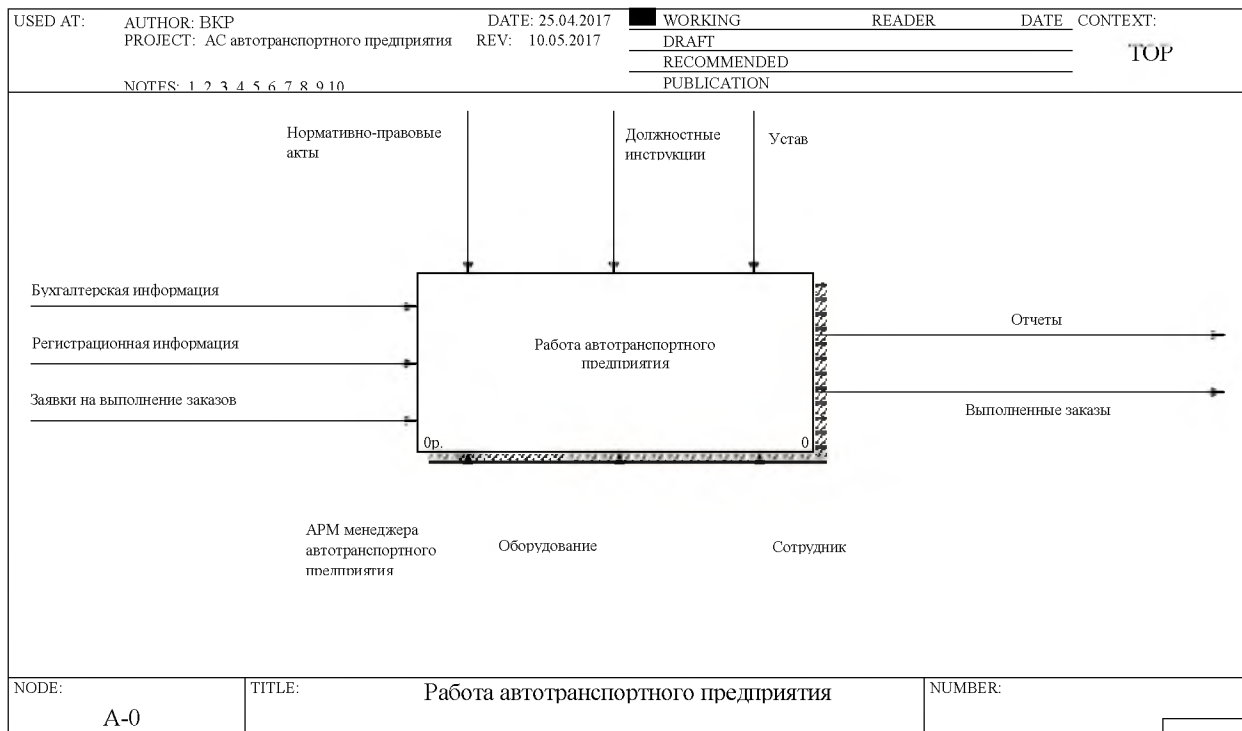


Рисунок 1.2 – Работа автотранспортного предприятия

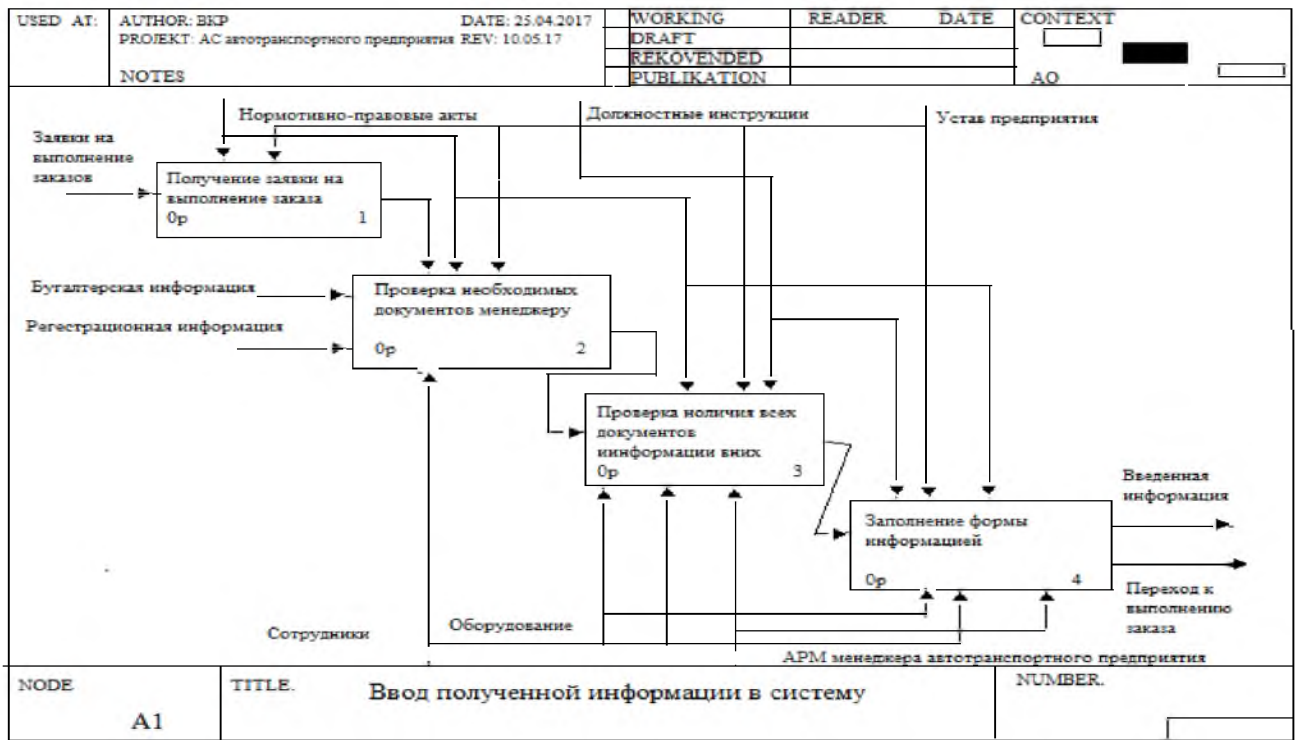


Рисунок 1.4 – Ввод полученной информации

На рисунке 1.5 показан процесс проверки наличия всех необходимых документов для оформления заявки [14,36].

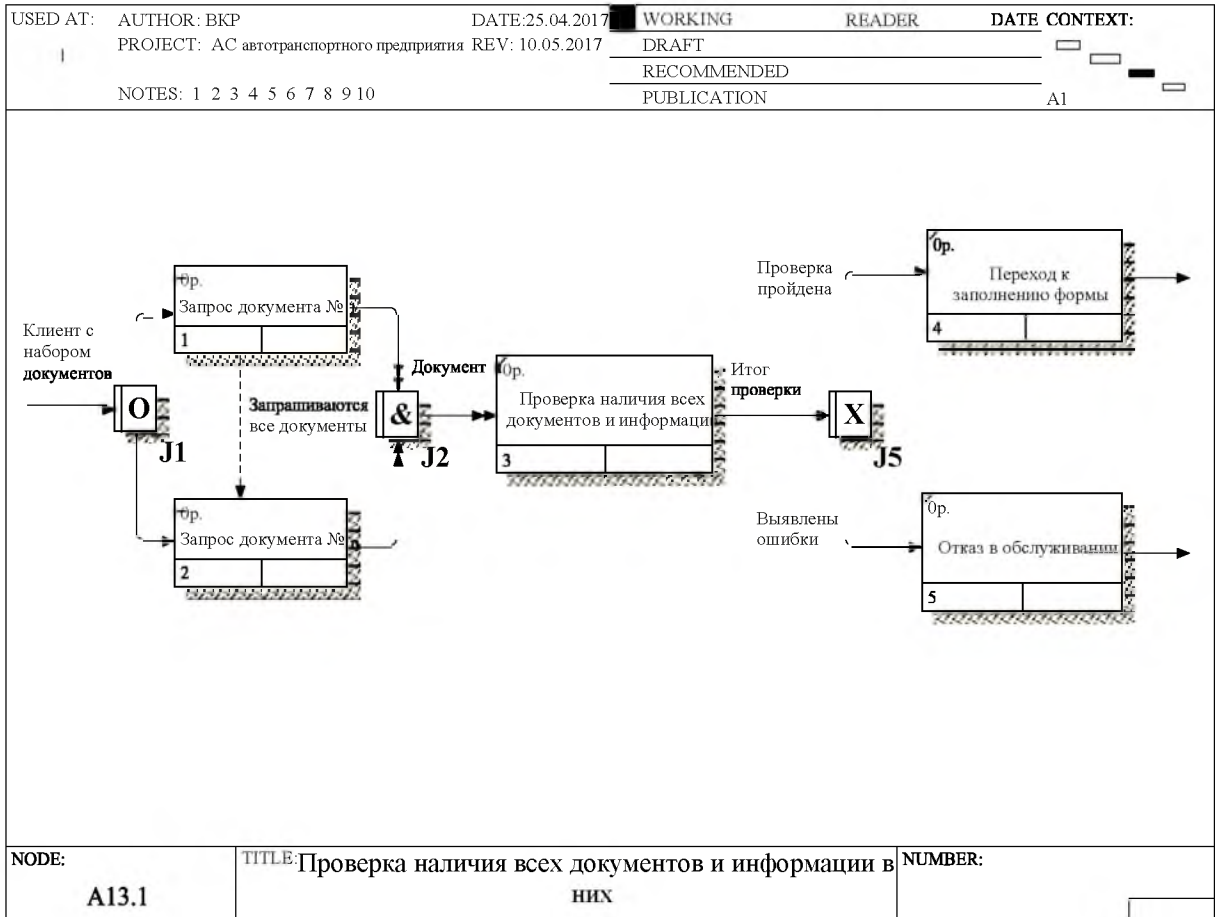


Рисунок 1.5 –Проверка наличия всех документов

На рисунке 1.6 демонстрируется процесс формирования отчетов.

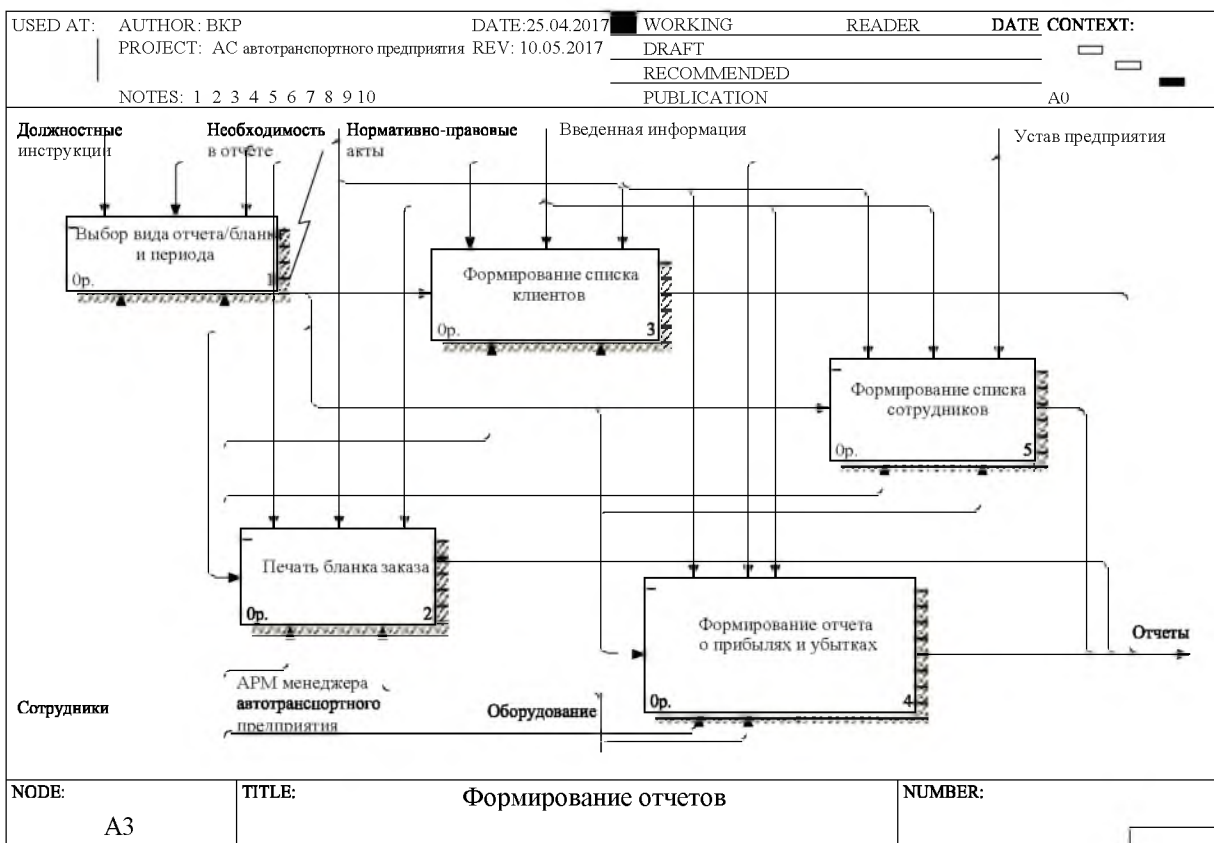


Рисунок 1.6 – Формирование отчетов

Целью третьего этапа является осуществление накопления, сортировки, корректировки, обработки данных получение результатов [31].

При этом требуется учитывать следующие требования:

- наличие возможностей обрабатывать данные на ЭВМ;
- при обработке различных данных гарантировать минимальные трудовые и денежные затраты;
- обеспечение подлинности обрабатываемой информации.

Отметим несколько факторов, за счет которых могут быть выполнены эти требования:

- уменьшение объема данных, предназначенных для обработки;
- повышение производительности организации за счет более высококвалифицированных работников и улучшения условий трудовой деятельности;

- уменьшение количества операций, особенно производимых в ручную, за счет автоматизации производства;
- разработка и использование системы ввода информации, предполагающей жесткий контроль.

В представленной работе описывается решение комплекса задач с использованием вычислительной техники, что обуславливается определенными факторами. При решении сформулированных задач используется вычислительная техника, так как объем и качество входной и выходной информации не позволяет их решить качественно, корректно и быстро без ее использования [10].

Другие факторы, определяющие методы решения поставленных задач с использованием вычислительной техники – это необходимость эффективного анализа заказов, как принимаемых, так и выдаваемых, возможность ведения актуальной базы данных по клиентам и сотрудникам [8] .

Создание автоматизированного рабочего места менеджера автотранспортного предприятия необходимо для того, чтобы:

- увеличить скорость обработки данных,
- избежать дублирования информации,
- уменьшить число операций.

Представляется целесообразным разработать информационную систему, которая будет реализована в качестве удаленной базы данных на основе клиент-серверной технологии доступа. Серверная часть будет разработана в СУБД FireBird 1.5.0. В качестве инструмента работы с удаленными базами данных выбрана консоль IVExpert. Клиентская часть будет создана в среде программирования C++Builder [5].

1.4 Постановка задачи

При ознакомлении с деятельностью организации ООО «Транспортно-

экспедиционная компания Белгорода», а также проведении анализа бизнес-процессов была поставлена задача автоматизации деятельности менеджера автотранспортного предприятия для повышения эффективности отдела продаж, скорости обработки информации о клиентах и заказах, контроля сроков оплаты.

Отметим назначение АРМ и основные функции, которые она должна выполнять:

- Оперативное получение информации о клиентах с использованием функции поиска;
- Добавление и редактирование заказов на поставку;
- Добавление и редактирование информации о клиентах;
- Добавление и редактирование информации о водителях;
- Учет информации о контактах с клиентами;
- Ведение справочников маршрутов и видов перевозок;
- Выдача и формирование отчетов.

Следует отметить, что работа менеджера довольно рутинна, в данном представлении: работа замедляется, возникает большое количество ошибок, оформление документов может быть некорректно. Ввиду всего этого начальнику отдела продаж необходимо проверять и перепроверять созданные документы, созданные менеджером, и суммы оплаты, указанные в документах. Вся эта работа является очень трудоемкой и требует больших затрат времени, что ведет к сужению возможности оперативного получения информации [10].

Таким образом, с учетом приведенных выше недостатков, возникает необходимость автоматизации работы менеджера автотранспортного предприятия. Автоматизация работы менеджера позволит надежно хранить и обрабатывать информацию, при значительном снижении трудоемкости, и повысить достоверность и оперативность получения результатной информации и итоговых отчетных документов.

Также автоматизация отдела продаж необходима для того, чтобы:

- избежать повтор информации,

- увеличить скорость обработки данных,
- уменьшить количество операций.

1.5 Анализ существующих разработок

Прежде чем разрабатывать предлагаемую информационную систему, представляется целесообразным провести анализ существующих на данный момент разработок. Среди систем автоматизации транспортных компаний, представленных на российском рынке, можно выделить следующие три разработки:

1) Программный продукт «TransTrade». Данная программа для транспортных компаний позволяет быстро и удобно решать задачи автотранспортной логистики. Данный программный продукт является многопользовательским и представляет собой транспортную программу, которая имеет гибкую настройку полномочий и прав. Этот программный продукт был разработан для получения решений транспортных задач. Автоматизация заключается в выполнении ряда задач и включает в себя: учет всей возможной информации о перевозках, выбор транспортных организаций для исполнения заказа среди привлеченных, подбор частных водителей, контроль осуществления перевозок в режиме реального времени, подготовка необходимых документов (их формирование и печать), возможность просмотра отчетной документации. Стоимость перевозок может быть рассчитана различными способами: по определенным тарифам, учитывающим вес и объем продукции, расстояние, на которое необходимо доставить груз, почасовая оплата; по фиксированному тарифу за маршрут; по ценам, являющимися договорными; так же оплата может носить частный характер при определенных условиях. Следует отметить, что задание тарифных ставок для клиентов и исполнителей осуществляется отдельно. Формирование и печать платежных документов осуществляется мгновенно и основанием для этого служит

зарегистрированный заказ. Платежные документы на печать могут выводиться в различном режиме: по единственному заказу, по какой-либо совокупности заказов, имеется возможность вывести на печать документы за какой-либо период осуществления транспортных перевозок [19].

Кроме этого, данная транспортная программа имеет возможность оперативного формирования различных документов, а именно: печатная форма заявки на фирменном бланке компании, путевой лист, товарно-транспортная накладная, приказы о командировках, детализации оказания услуг для клиента и по исполнителю. Множество наглядно-полезных сводных отчетов по всевозможным условиям и критериям. [11]

2) Программа «Бизнес Про». Данный программный продукт в настоящее время представляет собой программный комплекс, предназначенный для автоматизации предприятий, которые работают в сфере логистики и управлении перевозками. Данный программный продукт осуществляет автоматизацию всех бизнес-процессов транспортного предприятия, которое предоставляет услуги по перевозке различных грузов.

Программа «Бизнес Про» может быть использована в деятельности не только предприятий, которые занимаются транспортной логистикой и представляют собой экспедиторские компании, но и различных торговых подразделений и фирм, занимающихся производством. Такие предприятия могут использовать данный программный продукт в качестве инструмента, позволяющего отслеживать перемещения собственных грузов.

Возможности программного продукта:

- Предоставление информации о состоянии и местонахождении перевозимых грузов на каждом этапе перевозки;
- Возможность оформления предварительных заказов, возможность оформления заказов через Internet, представление многовариантных проектов перевозки грузов и возможность сохранения истории возможных вариантов перевозки, предложенных заказчиком;

- Учет грузов, которые поступают. В программном продукте имеется возможность предоставления не только обобщенной информации о грузе, но и возможность использования разбивки по товарным позициям;
- Возможность регистрации перемещения грузов: привоз, отправка в рейс, включая погрузку, разгрузка, оформление на хранение, доставка заказчику и т.д. При этом имеется возможность ведения статусов грузов;
- Возможность учета операций с грузом, таких как обработка груза, формирование партий, консолидация;
- Возможность осуществления калькуляции доходов и расходов, причем как плановой, так и фактической. Возможность автоматического расчета стоимости грузоперевозки с использованием различных алгоритмов;
- Возможность создания и ведения базы данных различных транспортных средств, включая автомобили, вагоны, суда и т.д. Учет в базе данных большого набора характеристик, используемых транспортных средств, таких как грузоподъемность, имя владельца, вмещаемый объем, место приписки и т.д.
- Возможность составлять мультимодальные перевозки;
- Возможность предоставления услуги доставки «дверь-дверь», предполагающая доставку в места приемки и/или выдачи груза имеющимся транспортом или арендуемым;
- Возможность создания комплекта документов, необходимых для сопровождения грузов;
- Возможность регистрации истории отношений и взаимных расчетов с контрагентами;
- Ведение учета финансовых операций, создание финансовой отчетности, дополнительный учет по задаваемым параметрам, например таких, как российский, международный и т.д., возможность использовать нескольких валют.

3) Программный продукт «proLOG». Данный программный продукт представляет собой интегрированный модульный программный комплекс для транспортного предприятия. Данный комплекс включает все виды деятельности предприятия начиная от продажи услуг по перевозкам, заканчивая возможностью создания управленческих отчетов. Программный комплекс «proLOG» для автоматизации логистики использует бизнес правила, которые позволяют осуществлять контроль ключевых показателей продажной, производственной и финансовой деятельности предприятия [30]. Важной частью является функция работы с клиентами, осуществляющая допуск клиентов. Данная функция позволяет контролировать превышение кредитного лимита клиента, запрещать осуществление сделок с клиентами, у которых имеется превышение кредитного лимита или имеется просроченная дебиторская задолженность. В рассматриваемом программном комплексе имеется функция, позволяющая контролировать стоимость груза. Все приведенные выше возможности позволяют сделать управление транспортной компанией более прогнозируемым и надежным [32].

а) Маркетинг и CRM. Модуль Маркетинг и CRM предназначен для ведения подробной базы контрагентов, клиентов, ведение истории работы с контрагентами, регистрация договоров, возможность проведение маркетинговых исследований, возможность осуществления индивидуальных маркетинговых задач, экспорт маркетинговых данных для последующего анализа в специализированных программах [33].

б) Производство. Модуль Производство дает возможность работникам предприятия фиксировать движение информации в процессе предоставляемых экспедиционных услуг в секторе авто- и смешанных грузоперевозок, включая сопровождение различных сборных грузов, таможенное оформление, страхование, терминальную обработку и очистку. Данный модуль предоставляет возможность осуществлять контроль сроков погрузки, разгрузки ТС, отслеживать всевозможное движение ТС; в значительной мере облегчает

процесс подготовки документов, необходимых для сопровождения грузов, оригиналы которых хранятся в системе.

в) Финансы и контроль. Модуль Финансы и контроль позволяет обеспечить сбор и хранение информации о счетах и актах, которые выставляются заказчикам и подлежат оплате, о текущей дебиторской и кредиторской задолженности и позволяет контролировать движения финансовых ресурсов начиная от заявки и заканчивая оплатой клиента.

г) Сбыт и Снабжение. Модуль «Сбыт и Снабжение» предоставляет возможность реализации полного цикла обслуживания процесса продажи логистических и транспортных услуг и позволяет проанализировать информационную базу клиентов и поставщиков с возможностью автоматического формирования документов. [13]

Исходя из функциональных возможностей и требований выше перечисленных систем, а так же из потребностей ООО «Транспортно-экспедиционная компания Белгорода», представляется целесообразным разработка программного продукта, выполняющего только те операции, которые осуществляет менеджер автотранспортного предприятия ООО «Транспортно-экспедиционная компания Белгорода» [34].

Разрабатываемая система обеспечит:

1. Ведение электронной базы постоянных клиентов;
2. Ведение истории заказов клиентов;
3. Создание и редактирование заказа;
4. Ведение справочников маршрутов и видов перевозок;
5. Получение различного вида отчетов.

2 Обоснование проектных решений по автоматизированному решению экономико-информационных задач.

2.1 Обоснование проектных решений по техническому обеспечению

Техническое обеспечение представляет собой комплекс технических средств, которые предназначены для работы информационной системы, а также соответствующую документацию на эти средства и технологические процессы.

В комплекс технических средств входят: [35]

- устройства сбора, накопления, обработки, передачи и вывода информации;
- компьютеры различных моделей;
- линий связи и устройства передачи данных;
- эксплуатационные материалы и др;
- оргтехника различного типа и устройства, позволяющие осуществлять автоматический съем информации.

Основными данными для выбора технических средств являются:

- характеристика технологического процесса обработки информации;
- технические характеристики оборудования, которое используется в качестве составной части КТС службы управления персоналом;
- характеристика задач, предназначенных для решения службой управления персоналом организации.

При подборе оборудования необходим учет задач, которые будут решаться с использованием данного оборудования. Такими характеристиками являются:

- носители информации, как входной так и выходной, которые могут быть представлены в виде документов, типизированных бланков, различных машинных носителей информации и т. д.;
- вычислительная нагрузка;

- объем входной и выходной информации по используемым носителям;
- вид и способ представления выходных результатов решения задач пользователям;
- временные рамки, отведенные для выполнения работ по решению задач управления персоналом.

При подборе оборудования необходимо учитывать состав и назначение используемого оборудования и его основные характеристики:

- надежность работы технических средств;
- производительность при выполнении технологических операций;
- совместимость работы оборудования различных типов, в том числе персональных компьютеров;
- состав и количество персонала, который обслуживает технические средства;
- цена оборудования;
- площадь, требуемая для размещения оборудования.

Состав и основные характеристики ПЭВМ, предназначенной для решения рассматриваемого комплекса экономических задач, приведены в таблице 2.1.

Таблица 2.1- Основные характеристики ЭВМ

1	2
Описание характеристики	Характеристики
Конфигурация системного блока	
Материнская плата	
Процессор	Pentium 4
Оперативная память	1 Gb
Емкость жесткого диска	200 Gb
Монитор	
Размер	19"
Безопасность	ТСО 03

Продолжение таблицы 2.1.

1	2
Описание характеристики	Характеристики
Мультимедиа	Нет
Принтер	
Тип	Лазерный
Формат	A4

2.2 Обоснование проектных решений по информационному обеспечению

Информационное обеспечение (ИО) представляет собой совокупность единой системы классификации и кодирования информации, унифицированных систем документации, схем информационных потоков, циркулирующих в организации, методология построения баз данных. Данная подсистема используется для своевременного представления информации, принятия управленческих решений [1].

Информационное обеспечение автоматизированных информационных систем состоит из внемашинного ИО (информация, воспринимаемая человеком без технических средств) – это классификаторы технико-экономической информации и документы; и внутримашинного ИО (совокупность всех данных, которые записаны на машинных носителях, сгруппированы по определенным признакам) – это макеты / экранные формы для ввода первичных данных в ЭВМ или ввода результатной информации [2].

Состав и содержание выходных документов должны быть договорены с заказчиком. Обязательным выходным документом системы является квитанция-договор.

При организации БД транспортного предприятия представляется целесообразным использовать типы баз данных, представленные ниже.

Основная база – это база данных, записи в которой хранят определенную информацию, при чем данные записи не имеют между собой никакой связи.

Основные базы данных не имеют связи между собой посредством своих полей. Они могут иметь связи между собой только посредством разделения дополнительной базы, которая имеет название база связи. Записи из основных БД могут подвергаться изменениям, а именно могут быть удалены, модифицированы, запакованы или упорядочены. К основным базам данных в рассматриваемой системе относятся базы, заполняемые информацией о заказах. [4]

Справочник – это БД, в которой находятся общие данные, разделяемые другими компонентами. Такие базы могут использоваться для хранения информации, при изменении которой в одном месте, необходимо немедленное отображение изменений во всех других, где данная информация используется. Достигается это путем ссылок на справочник. Такая гибкость при добавлении изменений не представляет возможности осуществления действий с изменением порядка записей в справочнике, а также удаление записей из справочника недопустимо. К справочникам в рассматриваемой системе можно отнести следующие базы данных: справочник клиентов, справочник маршрутов, справочник водителей и справочник видов перевозок [6].

При организации связей между базами данных необходимо следовать определенным правилам обращения с записями:

- Если возникает необходимость удаления записи из какой-либо базы или подчиненной базы, то все записи, которые имеют отношение к удаляемой в подчиненных базах нижнего уровня так же должны быть удалены.
- Если возникает необходимость удаления записи из справочника, то должны быть удалены и все ссылки на удаляемую запись во всех базах, которые имеют связи со справочником.
- Если возникает необходимость внесения дополнений структуры базы, то подчиненную базу следует применять там, где количество хранимой информации точно не известно.

Справочник является эффективным для быстрого ввода информации,

которая представляет собой логически связанную группу[22].

Использование справочника является целесообразным в том случае, когда есть необходимость группировки данных по какому-либо признаку. Также следует отметить, что справочник представляет собой средство для классификации объектов.

2.3 Обоснование проектных решений по программному обеспечению

Программное обеспечение (ПО) – совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ. ПО делится на два комплекса:

- общее (операционные системы, операционные оболочки, компиляторы, интерпретаторы, программные среды для разработки прикладных программ, СУБД, сетевые программы и т.д.);

- специальное (совокупность прикладных программ, разработанных для конкретных задач в рамках функциональных подсистем, и контрольные примеры).

К общесистемному программному обеспечению относятся программы, рассчитанные на широкий круг пользователей и предназначенные для организации вычислительного процесса и решений часто встречающихся задач обработки информации. Общесистемное программное обеспечение позволяет расширить функциональные возможности ЭВМ, автоматизировать планирование очередности вычислительных работ, осуществлять контроль и управление процессом обработки данных, а также автоматизировать работу программистов. К данному классу ПО следует отнести операционные системы, интегрированные средства управления задачами. В рамках создания данной АИС планируется использование операционной системы Windows 7 [16].

Windows 7 — операционная система семейства Windows NT корпорации Microsoft. По состоянию на 2017 год Windows 7 является самой популярной операционной системой в мире. Операционная система поступила в продажу 22

октября 2009 года меньше, чем через три года после выпуска предыдущей операционной системы Windows Vista [26].

Автоматизированное рабочее место в системе управления представляет собой проблемно – ориентированный комплекс технических, программных, лингвистических (языковых) и других средств, установленный непосредственно на рабочем месте пользователя и предназначенный для автоматизации операций взаимодействия пользователя с ЭВМ в процессе проектирования и реализации задач [7].

На сегодняшний день основой проектирования и разработки АРМ становится реляционная база данных.

В реляционных базах данных данные собраны в таблицы. Запросы к реляционным базам данных возвращает таблицу, которая повторно может участвовать в следующем запросе. Данные в одних таблицах, как правило, являются связанными с данными других таблиц. В реляционной базе данных каждая таблица имеет первичный ключ, который представляет собой поле или комбинацию полей, которые единственным образом идентифицируют каждую строку в таблице. Реляционная модель получила наибольшее распространение в СУБД для персональных компьютеров благодаря своей простоте и естественности представления. [16,18]

При возникновении первых компьютеров возникла и задача долгосрочного хранения и обработки информации. При решении этой задачи в конце 60-х г. были разработаны и внедрены специальные программные продукты, которые получили название «системы управления базами данных» (СУБД). В последующие годы СУБД претерпели значительные изменения. В последующие годы, а именно в конце 80-х годов активно стала развиваться система управления реляционными базами данных (СУРБД). В конечном итоге СУРБД стали определенным стандартом и возникла необходимость унифицировать работу с ними, в связи с чем был разработан специальный язык управления именно реляционными базами данных запросов (SQL). [5]

Вычисления клиент-сервер дают преимущества модели сетевых

вычислений с доступом к совместно используемым данным и высокие характеристики производительности, присущие модели вычислений с хост-машиной.

При проектировании информационной системы по принципу «клиент-сервер» предполагается, что реализованные в ее рамках прикладные программы имеют распределенный характер. Данный факт предполагает, что какая-то часть функций прикладной программы реализуется в программе-клиенте, а какая-то часть в программе-сервере. Для взаимодействия программы-клиента и программы-сервера определяется некоторый протокол. [17]

При описании принципа технологии «клиент-сервер» следует отметить, что функций стандартного интерактивного приложения подразделяются на четыре группы. Все четыре группы имеют разную природу. Первая группа представляет собой функции ввода и отображения информации (данных). Во вторую группу входят прикладные функции, которые определяются непосредственно предметной областью. [4]

В третью группу включены фундаментальные функции хранения и управления информационными ресурсами, к которым можно отнести файловые системы, базы данных и др. Функции, являющиеся служебными и представляющие собой связки между функциями трех первых групп принадлежат к четвертой группе.

Таким образом в приложениях принято выделять несколько логических компонент:

- компонент представления, реализующий функции первой группы;
- компонент доступа к информационным ресурсам, поддерживающий функции третьей группы;
- прикладной компонент, поддерживающий функции второй группы;
- протокол взаимодействия.

Различия в реализациях технологии «клиент-сервер» определяются четырьмя факторами. Во-первых, тем, в какие виды программного обеспечения

интегрирован каждый из этих компонентов. Во-вторых, тем, какие механизмы программного обеспечения используются для реализации функций всех четырех групп. В-третьих - как логические компоненты распределяются между компьютерами в сети. В-четвертых, какие механизмы используются для связи компонентов между собой.

Выделяются четыре подхода, реализованные в следующих моделях:

- модель файлового сервера (File Server - FS);
- модель доступа к удаленным данным (Remote Data Access - RDA);
- модель сервера базы данных (DataBase Server - DBS);
- модель сервера приложений (Application Server - AS). [5]

Разрабатываемая в выпускной квалификационной работе база данных будет иметь модель доступа к удаленным данным RDA. Предполагающая, что коды компонента представления и прикладного компонента совмещены и выполняются на компьютере-клиенте. Последний поддерживает как функции ввода и отображения данных, так и чисто прикладные функции. Доступ к информационным ресурсам обеспечивается либо операторами специального языка (языка SQL, если речь идет о базах данных) или вызовами функций специальной библиотеки (если имеется соответствующий интерфейс прикладного программирования - API).

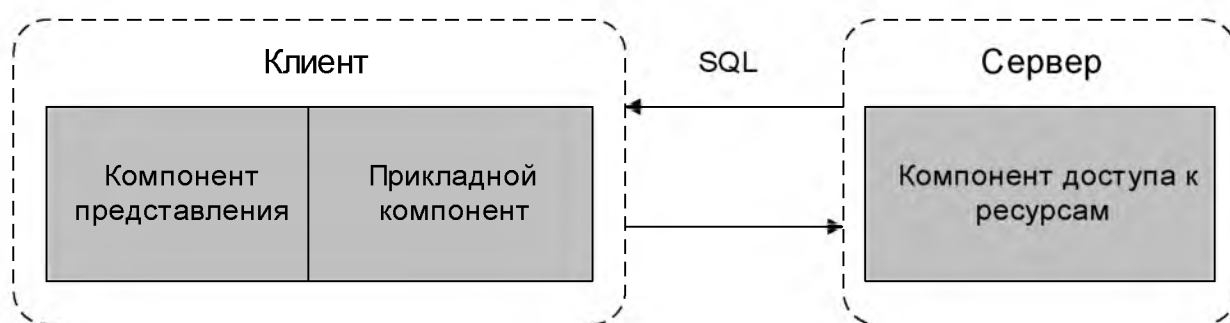


Рисунок 2.1 - Модель доступа к удаленным данным

Клиент направляет запросы к информационным ресурсам (например, к базам данных) по сети удаленному компьютеру. На нем функционирует ядро СУБД, которое обрабатывает запросы, выполняя предписанные в них действия,

и возвращает клиенту результат, оформленный как блок данных. При этом инициатором манипуляций с данными выступают программы, выполняющиеся на компьютерах-клиентах, в то время как ядру СУБД отводится пассивная роль - обслуживание запросов и обработка данных.

RDA-модель избавляет от недостатков, присущих как системам с централизованной архитектурой, так и системам с файловым сервером.

Прежде всего, перенос компонента представления и прикладного компонента на компьютеры-клиенты существенно разгружает сервер БД, минимизируя общее число процессов ОС. Также уменьшается загрузка сети, так как по ней передаются от клиента к серверу не запросы на ввод-вывод (как в системах с файловым сервером), а запросы на языке SQL, а их объем существенно меньше. Основное достоинство RDA-модели заключается в унификации интерфейса «клиент-сервер» в виде языка SQL.

В настоящее время существует большие возможности для выбора необходимого сервера базы данных (SQL-сервера). SQL-серверами являются промышленные СУБД, такие как: InterBase, Oracle, SyBase, Microsoft SQL Server, FireBird и др. [4]

Каждый из вышеперечисленных SQL-серверов имеет свои преимущества и особенности, связанные, например, со структурой базы данных и реализацией языка SQL, которые необходимо учитывать при разработке приложения. [27]

Для реализации в выпускной квалификационной работе поставленной задачи в качестве SQL-сервера был выбран FireBird 1.5.0. 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 используется в различных промышленных системах с 2001 г. Это коммерчески независимый проект C и C++ программистов, технических

советников и разработчиков мультиплатформенных систем управления базами данных, основанный на исходном коде, выпущенном корпорацией Borland 25 июля 2000 года в виде свободной версии Interbase 6.0. [2]

Firebird предлагает удобный путь миграции от коммерческих СУБД с закрытым исходным кодом к продуктам Open Source, в то же время, предлагая разработчику намного больше свободы в создании приложений. Firebird позволяет сделать этот переход легко и быстро. СУБД Firebird имеет практически все черты, характерные лучшим серверам баз данных без нанесения сколько-нибудь значительного вреда производительности, что подтверждают многочисленные промышленные тесты. [8]

В качестве инструмента работы с удаленными базами данными выбрана консоль IBExpert.

IBExpert — GUI-оболочка (графический интерфейс пользователя), предназначенная для разработки и администрирования баз данных InterBase и Firebird, т.е. реляционная система управления базами данных. Она позволяет не только полностью управлять структурами баз данных, но также и создавать хранимые процедуры, триггеры, а также отлаживать их в пошаговом режиме, как это принято в «обычных» языках программирования (C++, Delphi, Java, PHP и т.д.). Визуальные особенности редактирования IBExpert позволяют даже новичку в данной области быстро создавать БД и добавлять объекты в нее без особых трудов и временных затрат. IBExpert - наилучший всесторонний GUI инструмент для Firebird и InterBase на рынке сегодня. Выше были перечислены только некоторые из его особенностей. [10]

В данной работе для создания клиентского приложения была использована среда визуального программирования C++Builder 6.0. Следует отметить, что в среде программирования C++Builder разработка приложений является предельно простой. При разработки программного приложения большая часть времени уходит на разработку интерфейса, а в среде C++Builder данный процесс упрощен. Увеличение скорости создания интерфейса осуществляется за счет простого размещения необходимых компонент на

поверхности Windows-окна, а дальнейшую настройку компонентов можно осуществить используя специальный инструмент (Object Inspector), что, необходимо отметить, значительно ускоряет создание интерфейса разрабатываемого приложения. Инструмент Object Inspector позволяет связывать события компонентов с кодом его обработки используя готовую команду из списка простым нажатием мыши и т.д. Так же, создатель приложения может использовать мощные средства отладки, включая пошаговые выполнения команд процессора, контекстную справочную систему, которая является удобной в использовании и средства, позволяющие осуществлять коллективную работу над различными проектами. В C++ Builder имеется многозвенная технология доступа к данным, которая обеспечивает возможность создания масштабируемых приложений, слабо или относительно слабо зависящих от сервера БД. Данная возможность достигается путем использования среднего звена в методах обработки информации. [9]

Следует отметить, что, таким образом, C++ Builder подходит для разработки широкого круга приложений. Данная среда может быть использована для абсолютно простых приложений, которые создаются максимально быстро и не рассчитаны на выполнение сложных функций, а так же для больших, серьезных, возможно корпоративных проектов, предполагающих использование большим количеством пользователей, которые могут исчисляться в десятках и даже сотнях. При этом имеется возможность использования последних достижений в мире компьютерной техники с наименьшими временными и ресурсными затратами. [28]

Выбор необходимого программного обеспечения для проектирования и разработки автоматизированного рабочего места менеджера в транспортном агентстве был сделан, опираясь на вышеперечисленные характеристики и свойства различных программных средств.

2.4 Обоснование проектных решений по технологическому обеспечению

Технологический процесс обработки информации – это упорядоченная последовательность действий по обработке данных, информации, знаний до получения необходимого пользователю результата.

В отделе продаж ООО «Транспортно-экспедиционная компания Белгорода», при существующей технологии решения задачи. вычислительная техника не использовалась, т.е. все документы менеджер выписывал вручную, а информация о заказах заносилась в рабочие журналы. Это являлось фактором снижения эффективности и качества работы сотрудников данного экономического объекта.

Технологический процесс сбора информации в создаваемой информационной системе, будет представлять собой ввод первичной информации, контроль за допустимостью значений, обеспечение ввода данных путем выбора их из справочников с заранее внесенными в них данными. После того, как пользователь заполнил все необходимые поля, информационная система сохраняет полученную информацию в базе данных.

Для обработки данных в автоматизированной информационной системе используется пакетный режим обработки. Характерные черты данного режима обработки данных:

- Ввод потока данных осуществляется с локальных устройств ввода.
- Выполнение режима включает три фазы: подготовку, выполнение и завершение процесса. При этом первая фаза требует определения последовательности действий и ввода исходных данных. Вторая фаза предполагает логическое преобразование исходных файлов, создание и упорядочения рабочих файлов, обработку информации и формирование выходных данных, осуществляя контроль результатов решения.
- Печать результата. Эти особенности необходимо рассмотреть в связи со спецификой функциональной задачи. Применение пакетного режима

позволяет уменьшить вмешательство оператора в процесс решения задачи (оператор необходим только на этапе предварительного ввода данных), исключает возможность вмешательства пользователя и, следовательно, изменения последовательности выполняемых действий. Однако за счет этого появляется более полная загрузка оборудования, которое начинает работать по жесткому графику.

3 Разработка проекта АРМ менеджера транспортного предприятия

3.1 Информационное обеспечение задачи (комплекса задач, АРМ)

3.1.1 Используемые классификаторы и системы кодирования

Классификаторы представляют собой систематический свод, перечень каких-либо объектов, которые позволяют находить каждому из объектов свое место, и имеют определенное (обычно числовое) обозначение. Классификатор является стандартным кодовым языком документов, финансовых отчетов и автоматизированных систем. Назначение классификатора:

- систематизация наименований кодируемых объектов;
- возможность обобщения информации по заданной совокупности признаков;
- однозначная интерпретация одних и тех же объектов в различных задачах;
- возможность поиска и обмена информацией между различными внутрифирменными подразделениями и внешними информационными системами;
- возможность сопоставления одних и тех же показателей, содержащихся в формах статистической отчетности;
- экономия памяти компьютера при размещении кодируемой информации.

Любой классификатор включает код объекта и полное наименование объекта. Код объекта – это компактная и удобная форма представления полного наименования объекта. Различные классификаторы имеют разную структуру кода. Локальные классификаторы, как правило, имеют простейшую порядковую систему кодирования. В специальной литературе их иногда называют справочниками или каталогами. Существуют следующие системы

кодирования: фасетная, порядковая, серийно-порядковая, иерархическая, штриховая. Описание классификаторов представлены в таблице 3.1.

Таблица 3.1.- Описание классификаторов

Наименование кодируемого множества объектов	Значность кода	Система кодирования	Система классификации	Вид классификатора
Код Водителя	3	Порядковая	Отсутствует	Локальный
Код Клиента	3	Порядковая	Отсутствует	Локальный
Номер Заказа	3	Порядковая	Отсутствует	Локальный
Номер Маршрута	3	Порядковая	Отсутствует	Локальный
Номер Перевозки	3	Порядковая	Отсутствует	Локальный

3.1.2 Характеристика базы данных

3.1.2.1 Характеристика инфологической модели БД

База данных предназначена для отображения реального мира, то есть его какой-либо части. В зависимости от цели, с которой создается информационная система вообще и база данных в частности, зависит полнота ее описания.[27]

В базе данных отображается предметная область и для адекватного ее отображения создателю базы данных необходимо иметь представление обо всех нюансах предметной области и умение адекватно их отобразить. Следовательно, при создании базы данных какой-либо предметной области необходима информация и представление о ее функционировании и проектировщик базы данных на первом этапе должен выяснить данные вопросы. Предметная область должна иметь полное предварительно описание. Для описания предметной области можно использовать естественный язык, однако такое описание будет иметь определенные недостатки, такие как громоздкость описания и, во многих случаях, неоднозначность трактовки.

Таким образом, при описании предметной области рекомендовано использовать формальные языковые средства.

Инфологическая модель (ИЛМ) – это описание предметной области, которое использует специальные языковые средства для ее представления. При этом языковые средства являются независимыми от тех программных средств, которые будут использоваться далее.

При построении инфологической модели необходимо учитывать, что она должна быть пригодна для использования не только для СУБД, но и для работы с другими программными средствами при реализации информационной системы.

Инфологическая модель используется для представления информации о предметной области. Инфологическое проектирование представляет собой этап создания ИЛМ.

Требования, которые предъявляются к инфологической модели (ИЛМ):

- непротиворечивость (трактовка модели не может быть неоднозначной);
- адекватное отображение;
- легкость использования всеми пользователями;
- легкость увеличения (возможность расширения, то есть гарантировать ввод новых данных при неизменности ранее определенных);
- гибкий язык (язык должен быть применим как при ручном, так и при автоматизированном проектировании).

Из самого названия ИЛМ следует основное к ней требование, которое заключается в адекватности отображения предметной области. Таким образом, язык, который используется для представления ИЛМ обязан обладать выразительными возможностями для представления различных явлений, которые имеются в предметной области. ИЛМ не должна быть противоречивой. Она должна быть отражением потребностей и взглядов всех, кто пользуется системой и представлять собой единое интегрированное описание предметной

области. Не допускается категорически неоднозначная трактовка используемой модели.

Цель инфологического моделирования заключается в создании точного и полного отображение реального мира, которое в дальнейшем будет использоваться в качестве источника информации для построения БД.

Для всех категорий пользователей инфологической модели необходимо легкое ее восприятие. Инфологическая модель представляет собой ядро системы проектирования. При проектировании автоматизированной системы обработки информации инфологическая модель должна содержать всю достаточную и необходимую информацию. Информация из ИЛМ корреспондирует со словарной системой и другими компонентами банка данных.[23]

На рисунках 3.1 и 3.2 приведены логическая и физическая модели данных, которые использованы для решения задач, поставленных в данной выпускной квалификационной работе.

Данные модели показывают основные сущности, ключевые поля и атрибуты, входящие в каждую сущность. Также показаны информационные связи и потоки информации, позволяющие решить поставленные задачи.

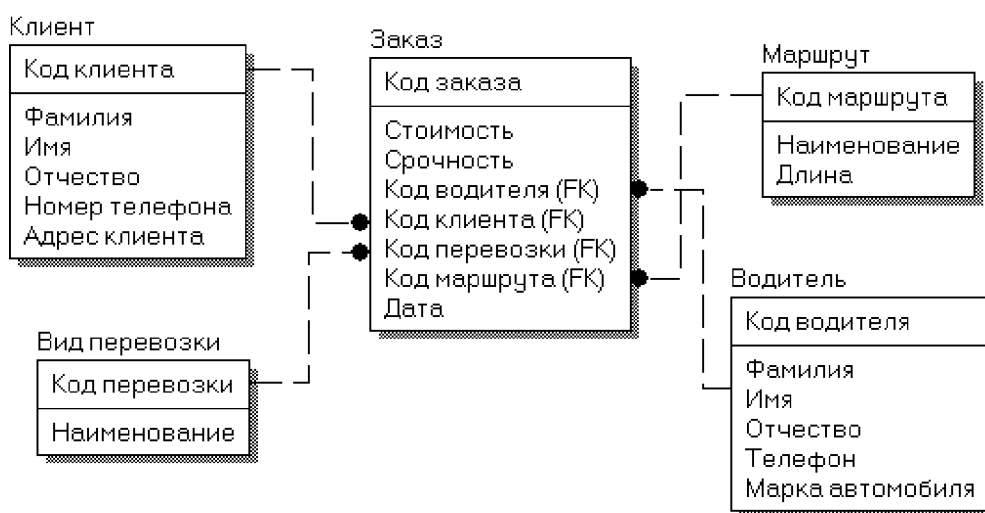


Рисунок 3.1 - Логическая модель данных

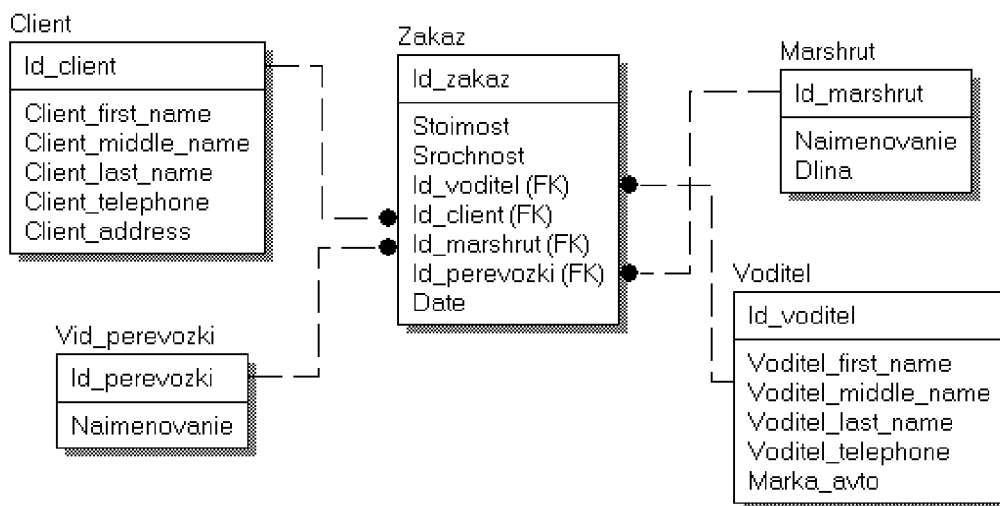


Рисунок 3.2 - Физическая модель данных

3.1.2.2 Характеристика даталогической модели БД

Даталогическое проектирование представляет собой организацию данных, выделенных на предыдущем этапе проектирования в форму, принятую в выбранной СУБД. Любая СУБД работает с допустимыми для нее логическими единицами данных, а также допускает использование определенных правил композиции логических структур более высокого уровня из составляющих информационных единиц более низкого уровня. Также, следует отметить, что многие СУБД накладывают количественные и иные ограничения на структуру базы данных. Поэтому прежде чем приступить к построению даталогической модели, необходимо детально изучить особенности СУБД, определить факторы, влияющие на выбор проектного решения, ознакомиться с существующими методиками проектирования, а также провести анализ имеющихся средств автоматизации проектирования, возможности и целесообразности их использования. [21]

Хотя даталогическое проектирование является проектированием логической структуры базы данных, на него оказывают влияние возможности физической организации данных, предоставляемые конкретной СУБД. Поэтому знание особенностей физической организации данных является полезным при

проектировании логической структуры.

Логическая структура базы данных, а также сама заполненная данными база данных являются отображением реальной предметной области. Поэтому на выбор проектных решений самое непосредственное влияние оказывает специфика отображаемой предметной области, отраженная в инфологической модели.

Конечным результатом даталогического проектирования является описание логической структуры базы данных на ЯОД (язык описания данных). Однако если проектирование выполняется «вручную», то для большей наглядности сначала необходимо построить схематическое графическое изображение структуры базы данных. При этом должно быть обеспечено однозначное соответствие между конструкциями языка описания данных и графическими обозначениями информационных единиц и связей между ними. Графическое представление используется и при автоматизированном проектировании структуры базы данных как интерфейсное средство общения с проектировщиком, и при документировании проекта.

Спроектировать логическую структуру базы данных означает определить все информационные единицы и связи между ними, задать их имена; если для информационных единиц возможно использование разных типов, то необходимо определить их тип. Следует также задать некоторые количественные характеристики, например длину поля.

Под даталогической понимается модель, отражающая логические взаимосвязи между элементами данных безотносительно их содержания и физической организации. При этом даталогическая модель разрабатывается с учетом конкретной реализации СУБД, также с учетом специфики конкретной предметной области на основе ее инфологической модели.

Таблица 3.2 -Даталогическая модель базы данных

Сущность	Идентификатор таблицы	Атрибут	Идентификатор поля	Тип поля
Водитель	VODITEL	Код водителя	ID	smallint
		Фамилия	FIRST_NAME	varchar(20)
		Имя	MIDDLE_NAME	varchar(20)
		Отчество	LAST_NAME	varchar(20)
		Телефон	TELEPHONE	varchar(20)
		Марка автомобиля	MARKA_AVTO	varchar(60)
Клиент	CLIENT	Код клиента	ID	smallint
		Фамилия	FIRST_NAME	varchar(20)
		Имя	MIDDLE_NAME	varchar(20)
		Отчество	LAST_NAME	varchar(20)
		Адрес	ADRES	varchar(60)
		Телефон	TELEPHONE	varchar(20)
Заказ	ZAKAZ	Код заказа	ID	smallint
		Стоимость	COST	numeric(15,2)
		Срочность	SROCHNOST	varchar(3)
		Код клиента	ID_CLIENT	smallint
		Код маршрута	ID_MARSHRUT	smallint
		Код перевозки	ID_PEREVOZKI	smallint
		Код водителя	ID_VODITEL	smallint
		Дата	DATE	numeric
Маршрут	MARSHRUT	Код маршрута	ID	smallint
		Наименование	NAIMENOVANIE	varchar(60)
		Длина	DLINA	varchar(20)
Вид перевозки	VID_PEREVOZKI	Код перевозки	ID	smallint
		Наименование	NAIMENOVANIE	varchar(60)

3.1.3 Характеристика результатной информации

3.1.3.1 Характеристика таблиц с результатной информацией

В данном разделе представлены таблицы с результатной информацией, полученные в результате запросов к другим таблицам БД с условно-постоянной информацией. В разработанной ранее БД был создан просмотр «ZAKAZ_INFO» использующий данные из следующих таблиц базы:

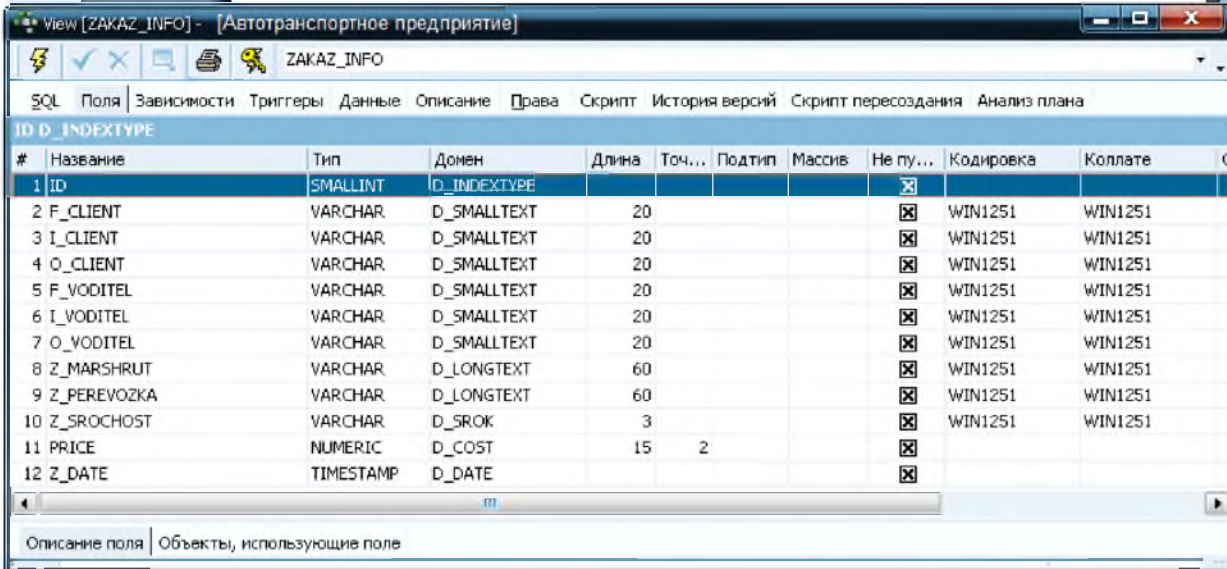
- таблица Zakaz поля ID, SROCHNOST, COST и DATE;
- таблица CLIENT поля FIRST_NAME, MIDDLE_NAME и LAST_NAME;

– таблица VODITEL поля FIRST_NAME, MIDDLE_NAME и LAST_NAME;

– таблица Marshrut поле NAIMENOVANIE;

– таблица VID_PEREVOZKI поле NAIMENOVANIE.

Данные, полученные с помощью просмотра, не сохраняются отдельно в БД, а



#	Название	Тип	Домен	Длина	Точ...	Подтип	Массив	Не пу...	Кодировка	Коллате
1	ID	SMALLINT	ID_INDEXTYPE					<input checked="" type="checkbox"/>		
2	F_CLIENT	VARCHAR	D_SMALLTEXT	20				<input checked="" type="checkbox"/>	WIN1251	WIN1251
3	I_CLIENT	VARCHAR	D_SMALLTEXT	20				<input checked="" type="checkbox"/>	WIN1251	WIN1251
4	O_CLIENT	VARCHAR	D_SMALLTEXT	20				<input checked="" type="checkbox"/>	WIN1251	WIN1251
5	F_VODITEL	VARCHAR	D_SMALLTEXT	20				<input checked="" type="checkbox"/>	WIN1251	WIN1251
6	I_VODITEL	VARCHAR	D_SMALLTEXT	20				<input checked="" type="checkbox"/>	WIN1251	WIN1251
7	O_VODITEL	VARCHAR	D_SMALLTEXT	20				<input checked="" type="checkbox"/>	WIN1251	WIN1251
8	Z_MARSHRUT	VARCHAR	D_LONGTEXT	60				<input checked="" type="checkbox"/>	WIN1251	WIN1251
9	Z_PEREVOZKA	VARCHAR	D_LONGTEXT	60				<input checked="" type="checkbox"/>	WIN1251	WIN1251
10	Z_SROCHOST	VARCHAR	D_SROK	3				<input checked="" type="checkbox"/>	WIN1251	WIN1251
11	PRICE	NUMERIC	D_COST	15	2			<input checked="" type="checkbox"/>		
12	Z_DATE	TIMESTAMP	D_DATE					<input checked="" type="checkbox"/>		

Рисунок 3.3 -Просмотр «ZAKAZ_INFO»

На основании данного просмотра формируется отчёт по заказам (вкладка Отчеты / Отчеты по заказам). Ниже приведён листинг данного просмотра.

```
CREATE VIEW ZAKAZ_INFO(  
    ID,  
    F_CLIENT,  
    I_CLIENT,  
    O_CLIENT,  
    F_VODITEL,  
    I_VODITEL,  
    O_VODITEL,  
    Z_MARSHRUT,
```

```

        Z_PEREVOZKA,
        Z_SROCHOST,
        PRICE,
        Z_DATE)
AS
select zakaz.id, client.first_name, client.middle_name, client.last_name,
        voditel.first_name,          voditel.middle_name,          voditel.last_name,
marshrut.naimenovanie, vid_perevozki.naimenovanie, zakaz.srochnost, zakaz.cost,
zakaz."DATE"
        from zakaz, client, voditel, vid_perevozki, marshrut
        where client.id = zakaz.id_client and voditel.id=zakaz.id_voditel and
vid_perevozki.id = zakaz.id_perevozki and marshrut.id=zakaz.id_marshrut

```

3.1.3.2 Характеристика результатных документов

В разрабатываемой программе конечных результатных документов не предусмотрено, но предусмотрены отчеты позволяющие просмотреть информацию о клиентах (вкладка Отчеты / Отчеты по клиентам) и информация о заказах (вкладка Отчеты / Отчеты по заказам). Данная информация представляется в виде упорядоченного списка предназначенного для чтения и вывода на печать, так же данные отчёты можно отдельно сохранить с расширением *.QRP, рисунки 3.4 и 3.5.

Клиенты					
Номер	Фамилия	Имя	Отчество	Телефон	Адрес
1	Миронов	Олег	Петрович	8-920-659-95-89	г. Белгород ул. Победы д 7 кв 13
3	Козлова	Анна	Максимовна	56-95-89	г. Белгород ул. Щорса д 34 кв 7
4	Максомов	Кирилл	Олегович	8-915-637-69-95	г. Белгород ул. Некрасова д 13 кв 25
5	Иванов	Иван	Иванович	8-904-498-95-73	г. Белгород пр. Славы д 68 кв 77

Рисунок 3.4 - Отчёт «Клиенты»

Заказы						
Номер	Стоимость	Срочность	Клиент	Маршрут	Водитель	Дата
1	1620	Нет	Миронов	Белгород - Курск	Клинов	04.05.2017
2	1685	Да	Козлова	Белгород - Короча	Зыков	06.05.2017
3	3500	Да	Миронов	Белгород - Короча	Клинов	07.05.2017
4	5700	Нет	Козлова	Белгород - Короча	Кассов	07.05.2017
5	2400	Да	Миронов	Белгород - Короча	Клинов	16.05.2017
6	5500	Нет	Козлова	Белгород - Курск	Зыков	17.05.2017
7	3000	Да	Максомов	Белгород - Москва	Кассов	17.05.2017
8	4000	Да	Максомов	Белгород - Москва	Клинов	19.05.2017

Рисунок 3.5 - Отчёт «Заказы»

Так же в данном приложении была разработана вкладка «Графики», предназначенная для отображения данных о выручке водителей за отчётный период в графическом наглядном виде, рисунок 3.6.



Рисунок 3.6 - Вкладка «Графики»

3.2 Программное обеспечение задачи (комплекса задач, АРМ)

Программное обеспечение современного компьютера представляет собой набор программ, с помощью которых реализуются его многочисленные функции:

- использование всех возможностей аппаратуры компьютера и его периферийных устройств в целях наилучшего и наискорейшего решения возникающих задач;
- специальная организация вычислительного процесса;
- реализация удобного взаимодействия компьютера с пользователем и пользователя с компьютером (удобный интерфейс);
- эффективное решение сложных прикладных задач в различных областях предметной деятельности человека;
- комплексная автоматизация решения сложных задач оценивания данных и принятия решений для различных систем объектов;

- качественное и быстрое решение типовых задач в различных областях предметной деятельности человека;
- эффективная организация многопользовательского режима работы компьютеров и работы в локальных и глобальных компьютерных сетях;
- эффективная разработка новых программных средств;
- решение интеллектуальных задач планирования и прогнозирования деятельности, проведения экспертизы, обучения, тестирования знаний и др. в различных областях предметной деятельности человека.

СУБД предоставляет возможность управлять различными по структуре массивами данных. В практической деятельности часто возникают задачи, заключающиеся в создании специализированных информационных систем для объектов, в которых имеется большое количество элементов различного вида из соответствующего массива данных, связанных между собой различными математическими и логическими соотношениями. При создании таких систем используются СУБД, в которых необходимая обработка данных осуществляется наиболее естественным для пользователей способом – с удобным представлением входных данных, выходных форм, графиков, диаграмм, запросов на поиск необходимых данных и др. Такой СУБД является FireBird 1.5.0. В качестве инструмента работы с удаленными базами данными выбрана консоль IVExpert. [20]

Изначально можно выделить и детализировать два подмножества функций: реализующих служебные функции (например, проверки пароля, ведения календаря, архивации баз данных, тьютора и др.) и реализующих основные функции управления и обработки данных: ввода первичной информации, обработки, ведения справочников, ответов на запросы и др.

К служебным функциям, которые реализованы в приложении можно отнести обеспечение целостности данных, ввод корректной информации и транзакции, передачу данных на печать.

Так же приложение обеспечивает полный цикл обработки атрибутивных и пространственных данных: от ввода исходной информации в базу данных до

консолидации данных, их анализа и представления конечному пользователю. Основными функциональными блоками приложения являются блоки редактирования данных, блок анализа и представления данных. Приложение редактирования данных обеспечивает ввод, обновление и удаление данных пользователем. Для редактирования свойств объектов используются формы ввода. Все вышеперечисленные возможности реализуются при помощи хранимых процедур БД, коих 16, рисунок 3.7.

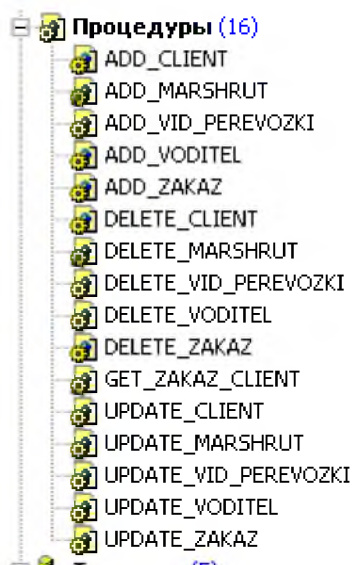


Рисунок 3.7 - Дерево процедур

3.3 Технологическое обеспечение задачи (комплекса задач, АРМ)

Технологическое обеспечение включает описание организации технологии сбора, передачи, обработки и выдачи информации и отражает последовательность операций, начиная от способа сбора первичной информации и заканчивая формированием резульатной информации.

Технологический процесс обработки информации представляет собой упорядоченную последовательность действий по обработке данных, информации, знаний до получения необходимого пользователю результата. Отсюда следует, что понятие информационной технологии подразумевает решение экономических и управленческих задач, связанное с выполнением ряда операций по сбору необходимой для решения этих задач информации,

переработке ее по некоторым алгоритмам и выдачи лицу, принимающее решение в удобной для него форме.

Технологический процесс обработки информации во многом зависит от характера решаемых задач, используемых технических средств, систем контроля, числа пользователей и многих других факторов. Технологический процесс обработки информации включает следующие операции (действия):

1) Сбор данных, информации, знаний – это процесс регистрации, фиксации, записи детальной информации (данных, знаний) о событиях, объектах (реальных и абстрактных), связях, признаках и соответствующих действиях. При этом иногда выделяют в отдельные операции «сбор данных и информации» и «сбор знаний». Сбор данных и информации – процесс идентификации и получения данных от различных источников, группирование полученных данных и представление их в форме, необходимой для ввода в ЭВМ. Сбор знаний – получение информации о предметной области от специалистов – экспертов и представление в форме, необходимой для записи в базу знаний.

2) Обработка данных, информации, знаний. Обработка – понятие широкое и включает в себя несколько взаимосвязанных операций. К обработке можно отнести такие операции как: проведение расчетов, выборку, поиск, объединение, слияние, сортировка, фильтрацию и др. Следует помнить, что обработка – это систематическое выполнение операций над данными, процесс преобразования вычисления, анализа и синтеза любых форм данных, информации и знаний, посредством систематического выполнения операций над ними. При определении такой операции, как обработка выделяют: обработку данных, обработку информации, обработку знаний. Обработка данных представляет собой процесс управления данными (цифры, символы и буквы) и преобразование их в информацию. Обработка информации – переработка информации определенного типа (текстового, звукового, графического), преобразование ее в информацию другого типа.

Однако использование новейших современных технологий обеспечивает

комплексное представление и одновременную обработку информации любого вида (текст, графика, аудио, видео, мультипликация), ее преобразование. Понятие обработки знаний связано с понятием экспертных систем (или систем искусственного интеллекта), позволяющих на основании правил и предоставляемых пользователем фактов распознать ситуацию, поставить диагноз, сформулировать решение, дать рекомендации по выбору действий.

3) Генерация данных, информации, знаний – процесс организации, реорганизации и преобразования данных (информации, знаний) в требуемую пользователем форму, в том числе и путем ее обработки. Например, процесс получения форматированных отчетов (документов).

4) Хранение данных, информации, знаний – процессы накопления, размещения, выработки и копирования данных (информации, знаний) для дальнейшего их использования (обработки и передачи).

5). Передача данных, информации, знаний – процесс распространения данных (информации, знаний) среди пользователей посредством средств и систем коммуникаций и путем перемещения (пересылки) данных от источника (отправителя) к приемнику (получателю).

Разработанная программа для работы с данными, использует хранимые процедуры и представления из базы данных. Клиент-серверная СУБД позволяет обмениваться клиенту и серверу минимально необходимыми объёмами информации. При этом основная вычислительная нагрузка ложится на сервер, основные процессы обработка, генерация, хранение, передача, этим и занимается сервер и установленная на нём СУБД. Клиент может выполнять функции предварительной обработки перед передачей информации серверу, но в основном его функции заключаются в организации доступа пользователя к серверу.

Сбором данных занимается сам пользователь. Так как хранятся статические накопительные данные для текущей работы и подготовки отчётов, их удобно отображать и редактировать в таблицах, поэтому пользовательское приложение имеет ряд вкладок, на которых размещены таблицы с

соответствующей информацией, редактирование данных производится в специальных полях ввода практически на всех закладках приложения.

3.4 Описание контрольного примера реализации проекта

Тестирование программы будет проводиться следующим образом. Для начала необходимо ввести всю необходимую информацию клиента, рисунок 3.8.

Номер	Фамилия	Имя	Отчество	Телефон	Адрес
1	Миронов	Олег	Петрович	8-920-659-95-89	г. Белгород ул. Победы д 7 кв 13
3	Козлова	Анна	Максимовна	56-95-89	г. Белгород ул. Щорса д 34 кв 7
4	Максимов	Кирилл	Олегович	8-915-637-69-95	г. Белгород ул. Некрасова д 13 кв 25

Номер заказа	Стоимость заказа	Маршрут	Вид перевозки
1	1620	Белгород - Курск	Срочная перевозка
3	3500	Белгород - Короча	Срочная перевозка
5	2400	Белгород - Короча	Срочная перевозка

Рисунок 3.8 - Форма ввода данных о клиенте

Как видно из рисунка 3.9 новый клиент добавлен. Вся остальная необходимая информация вносится аналогичным способом. Так же на всех формах в наличии имеются поля редактирования данных, удаления из БД, все данные можно как отфильтровать, так и отсортировать по заданным параметрам.

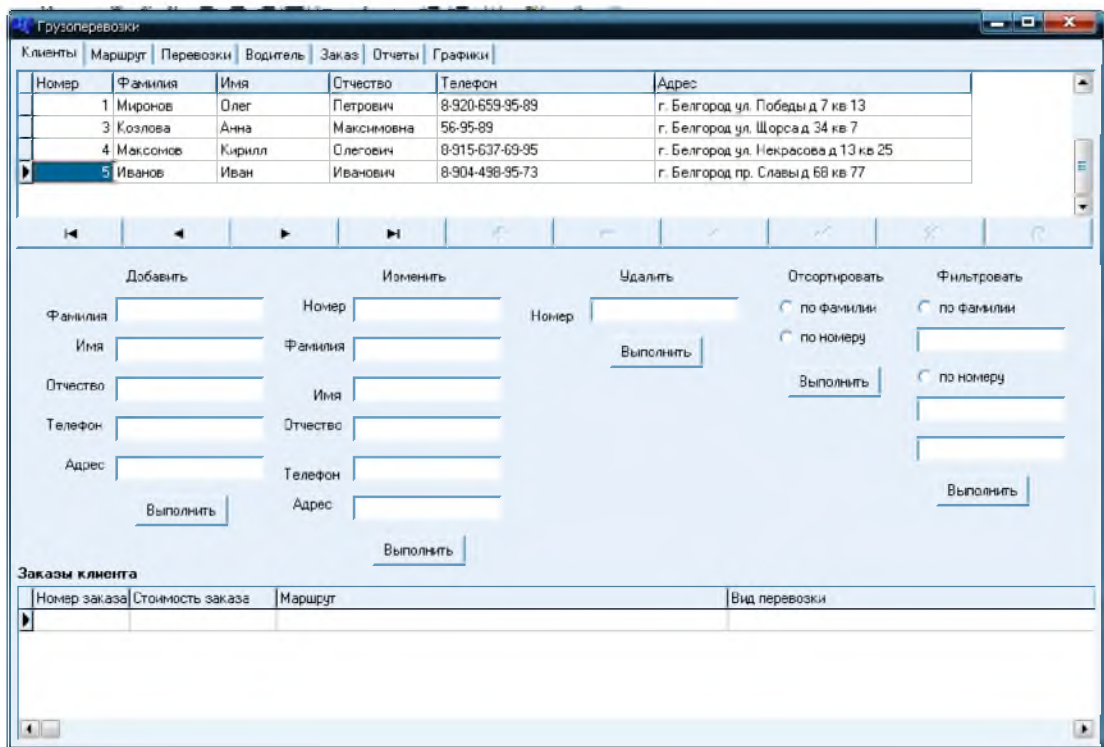


Рисунок 3.9 - Добавлен новый клиент

При выборе клиента снизу отображаются все сделанные им ранее заказы. У нашего клиента заказов пока нет. Добавим на закладке «Заказы» новый заказ от этого клиента, рисунок 3.10.

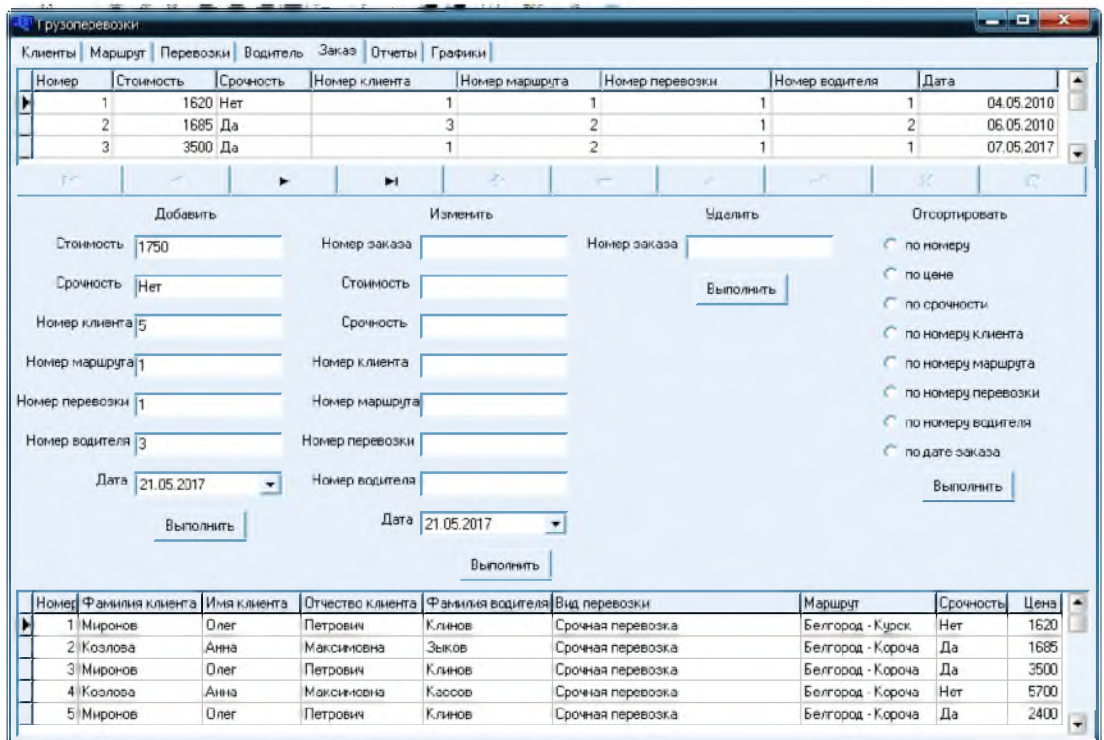


Рисунок 3.10. Форма ввода нового заказа

После того как будет добавлен заказ, его можно будет увидеть в таблице заказов и в отчете по заказам, рисунок 3.11, а так же это повлияет на график выручки водителей, рисунок 3.12.

Заказы						
Номер	Стоимость	Срочность	Клиент	Маршрут	Водитель	Дата
1	1620	Нет	Мионов	Белгород - Курск	Клинов	04.05.2017
2	1685	Да	Козлова	Белгород - Короча	Зыков	06.05.2017
3	3500	Да	Мионов	Белгород - Короча	Клинов	07.05.2017
4	5700	Нет	Козлова	Белгород - Короча	Кассов	07.05.2017
5	2400	Да	Мионов	Белгород - Короча	Клинов	16.05.2017
6	5500	Нет	Козлова	Белгород - Курск	Зыков	17.05.2017
7	3000	Да	Максомов	Белгород - Москва	Кассов	17.05.2017
8	4000	Да	Максомов	Белгород - Москва	Клинов	19.05.2017

Рисунок 3.11 - Отчёт по заказам

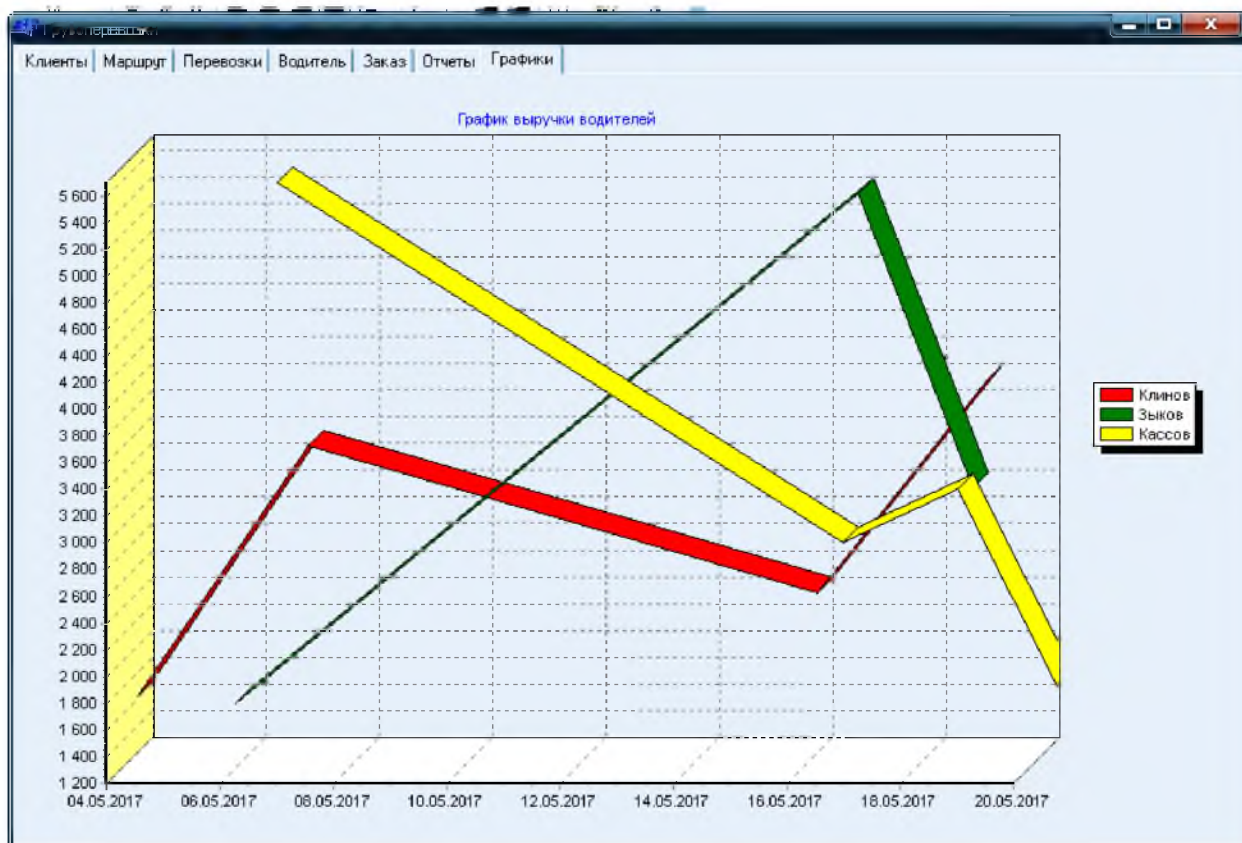


Рисунок 3.12 - График выручки водителей

В результате ввода экспериментальных данных было выявлено, что все модули функционируют правильно, ошибок в работе приложения и базы данных выявлено не было.

Программный код приложения представлен в приложении.

В данной главе смоделирован новый вариант организации информационной системы предметной области «КАК ДОЛЖНО БЫТЬ». С помощью кейс средств смоделирована инфологическая и даталогическая модель базы данных. Согласно которых было построено пользовательское приложение для учёта заказов на грузоперевозки. Также приведён контрольный пример на основе ввода тестовых данных для отображения правильности работы приложения.

3.5 Целесообразность разработки с экономической точки зрения

К основным показателям экономической эффективности относятся: экономический эффект, коэффициент экономической эффективности капитальных вложений, срок окупаемости капитальных вложений.

Экономический эффект – результат внедрения, какого-либо мероприятия, в нашем случае контроля за выполнением и финансированием договоров и контрактов, выраженный в стоимостной форме, в виде экономии от его осуществления. Так, для автотранспортного предприятия основными источниками экономии являются:

- за счет использования программного продукта повышение показателей основной деятельности;
- сокращение количества работников, необходимых для получения, обработки и переработки используемой информации;
- изменение в сторону уменьшения объемов вычислительных работ и повышение технического уровня;

- увеличение коэффициента использования вычислительных ресурсов, средств подготовки и передачи информации;
- уменьшение затрат на материалы, которые являются эксплуатационными;
- изменение в сторону повышения объемов и в сторону уменьшения сроков обработки информации.

3.6 SWOT-анализ разработки

SWOT-анализ – метод анализа в стратегическом планировании, заключающийся в разделении факторов и явлений на четыре категории:

- Сильные стороны,
- Слабые стороны,
- Возможности,
- Угрозы.

Методология построения SWOT – матрицы заключается в том, что сначала весь мир делится на две части – внешнюю среду и внутреннюю, а затем события в каждой из этих частей – на благоприятные и неблагоприятные.

Процесс заполнения матрицы должен быть выполнен максимально тщательно, т.к. на результат может повлиять даже такая вещь, как последовательность заполнения клеток матрицы. На практике число SWOT-факторов по каждой оси матрицы может быть значительным и существуют специальные процедуры их ранжирования и свертывания.

SWOT-анализ предполагает выполнение трех основных шагов.

1) шаг. Упорядочивание и структурирование информации. На данном этапе происходит разделение факторов на внешние и внутренние. Необходимо придерживаться следующих правил при заполнении этих полей:

а) Позиция. Анализ проводится с позиции потребителей, то есть сильные и слабые стороны могут считаться таковыми, если их так воспринимают потенциальные потребители.

б) Объективность. При проведении анализа необходимо соблюдать объективность, исключается использование устаревшей или ненадежной информации.

в) Точность. Необходимо придерживаться точности в формулировках, что бы пункты имели конкретное весомое значение для потенциальных потребителей.

г) Различия. Важно различать элементы внешней и внутренней среды. Внутренняя среда – поддается влиянию компании, внешняя не контролируется.

2) шаг. Градация факторов. На этом этапе необходимо градировать факторы по степени важности.

В SWOT-анализе необходимо не только вскрыть угрозы и возможности, но и попытаться оценить их с точки зрения того, сколь важным для фирмы является учет в стратегии своего поведения каждой из выявленных угроз и возможностей. Для такой оценки применяется метод позиционирования каждой конкретной возможности или угрозы.

3) шаг. Формирование стратегических ответов. Этап на котором заполняются поля СиВ (сила и возможности), СиУ (сила и угрозы), СлВ (слабость и возможности), СлУ (слабость и угрозы) и даются ответы на вопросы обозначенные в этих полях.

Каждое из полей SWOT-анализа несет свою целевую и смысловую нагрузку. Поле СиВ показывает возможности для развития, поле СиУ в основном возможности при выработке построении «обороны», поле СлВ показывает насколько компания в состоянии преодолеть свои слабые стороны за счет открывающихся возможностей, поле СлУ определяет насколько усложняют возможные угрозы слабые стороны предприятия.

Последнее поле (слабость и угрозы) является наиболее сложным при заполнении матрицы. Это поле необходимо для того, что бы методика SWOT-

анализа не приводила к приятной картине легкости реализуемой стратегии и делает анализ более объективным.

В таблице 3.3.- приведена SWOT-матрица, составленная для ООО «Транспортно-экспедиционная компания Белгорода» и рассматривающая возможность внедрения разработок выпускной квалификационной работы.

Таблица 3.3.- SWOT – матрица

Сильные стороны	Возможности		Угрозы		Итого
	1. Расширение круга пользователей	2. Совершенствование разработки	1. Появление новых конкурентов	2. Быстрое моральное устаревание	
1. Низкая стоимость разработки	++	0	++	0	+4
2. Широкая функциональность	++	+	+	++	+6
3. Обеспечение сопровождения	+	+	+	+	+4
Итого	+5	+2	+4	+3	+14
Слабые стороны					
1. Недостаточное финансирование	--	--	-	-	-6
2. Дефицит квалифицированных кадров	--	--	0	--	-6
Итого	-4	-4	-1	-3	-12
Общий итог	+1	-2	+3	0	+2

Проанализировав полученную SWOT-матрицу, можно сделать следующие выводы:

1) Трудно выделить самые важные сильные стороны, они все в достаточной степени важны, но, по-видимому, наиболее важным достоинством разработки является ее широкая функциональность. В дальнейшем необходимо обращать особое внимание на обеспечение и расширение этой стороны разработки;

2) Все выделенные слабые стороны разработки являются в одинаковой степени опасными;

3) Из рассмотренных возможностей более реальной представляется возможность расширения круга пользователей, хотя данная возможность при существующих слабостях весьма проблематична. Совершенствование же разработки при недостатке финансирования и нехватке кадров вообще невозможно;

4) Из двух представленных угроз наиболее опасной является быстрое моральное устаревание программы. Появление конкурентов представляется маловероятным вследствие наличия сильных сторон разработки.

5) Заключение о перспективности разработки. В настоящий момент при реализации разработки имеется две достаточно большие проблемы. Поэтому первоочередным представляется поиск дополнительного финансирования и повышение квалификации персонала организации. Если отмеченные слабости будут преодолены, тогда разработка данной выпускной квалификационной работы может стать более перспективной, так как у нее имеется ряд очень важных достоинств.

3.7 Калькуляция себестоимости научно-технической продукции

Затраты на автоматизированную обработку информации определяются следующим образом:

1) Материалы. Материальные затраты указаны в таблице 3.4.

Таблица 3.4.- Материальные затраты

Наименование материальных затрат	Ед. изм.	Кол-во	Цена без НДС с учетом комиссионных вознаграждений, таможенных пошлин и транспортных затрат	Сумма
CD-R	шт.	10	15	150
Бумага ZOOM, пачка 500 листов	шт.	1	350	350
Картридж для принтера HP	шт.	1	2750	2750
Итого:				3250

2) Спецоборудование для научных (экспериментальных) работ Для создания АРМ менеджера транспортного агентства необходимо, помимо создания программной части, еще и техническое ее оснащение. К техническим средствам относятся: персональный компьютер, монитор, клавиатура, сетевой фильтр, мышь и принтер. Затраты на покупку компьютера и оргтехники представлены в таблице 3.5.

Таблица 3.5.- Затраты на покупку оргтехники

№	Наименование	Стоимость (руб.)
1.	Компьютер класса «Pentium»	17790,00
2.	Монитор 19,5" Asus VS207T	6200,00
3.	Клавиатура проводная Genius KB-110X	790,00
4.	Мышь Genius DX-120	390,00
5.	Сетевой фильтр PILOT PRO	1100,00
6.	Принтер HP LaserJet Pro P1102	7600,00
Итого		33870,00

3) Трудовые затраты Разработка выполнялась инженером-программистом в течение 45 рабочих дней при восьмичасовом рабочем дне. Месячный фонд времени работы инженера-программиста 166,25 часов, среднемесячная заработная плата 15000 руб. В таблице №3.6 представлена оценка трудоемкости разработки.

Таблица 3.6- Оценка трудоемкости разработки

№	Наименование этапа	Трудоемкость этапа, часы
1	Анализ задания и знакомство с темой	30
2	Поиск и анализ возможных решений поставленной задачи	35
3	Выбор основного решения по созданию системы	20
4	Согласование решения с заказчиком	15
5	Составление технического задания и листа требований на доработку	50
6	Разработка программного продукта	120
7	Внедрение программного продукта	40
8	Отладка	40
9	Составление и согласование проекта пакета служебных инструкций	10
Итого		360

Основная заработная плата разработчика составила:

$$C_{\text{ин}} = \frac{15000}{166,25} * 8 * 45 = 32481,20 \text{ руб.}$$

Дополнительная заработная плата составляет 20%

$$Z_{\text{дон}} = 0,2 * Z_{\text{осн}} = 0,2 * 32481,20 = 6496,24 \text{ руб.}$$

Затраты на оплату труда с учетом поясного коэффициента (25%)

$$Z_{\text{ТР}} = 1,25 * (Z_{\text{осн}} + Z_{\text{дон}}) = 1,25 * (32481,20 + 6496,24) = 48721,80 \text{ руб.}$$

Отчисления на социальные нужды.

Единый социальный налог.

а) отчисления в Пенсионный фонд (28% от затрат на оплату труда)

$$0,28 * 48721,80 = 13642,10 \text{ руб.};$$

б) отчисления в Фонд Социального страхования (4% от затрат на оплату труда)

$$0,04 * 48721,80 = 1948,82 \text{ руб.};$$

в) отчисления в Федеральный Фонд обязательного медицинского страхования (0,2% от затрат на оплату труда)

$$0,002 * 48721,80 = 97,44 \text{ руб.};$$

г) отчисления в Территориальный Фонд обязательного медицинского страхования (3,4% от затрат на оплату труда)

$$0,034 * 48721,80 = 1656,54 \text{ руб.}$$

Итого единый социальный налог 17344,90 руб.

Страховой взнос на обязательное социальное страхование от несчастных случаев на производстве и профессиональных заболеваний (0,2% от затрат на оплату труда)

$$0,002 * 48721,80 = 97,44 \text{ руб.}$$

Итого отчисления на социальные нужды 17442,34 руб.

4) Накладные расходы составляют 50% от затрат на оплату труда:

$$0,50 * 48721,80 = 24360,90 \text{ руб.}$$

Таблица 3.7- Статьи затрат

№	Наименование статей затрат	Сумма
1	Материалы	3250
2	Спецоборудование для научных (экспериментальных) работ	33870,00
3	Затраты на оплату труда работников, непосредственно занятых созданием научно-технической продукции	48721,80
4	Отчисления на социальные нужды	17442,34
5	Прочие прямые расходы	0,00
6	Накладные расходы	24360,90
7	Итого:	127645,04

Следовательно, всего на разработку затрачено 360 часов или 45 рабочих дней при восьмичасовом рабочем дне.

ЗАКЛЮЧЕНИЕ

В выпускной квалификационной работе был выполнен обзор и анализ предметной области. В рамках этого была рассмотрена деятельность транспортного агентства ООО «Транспортно-экспедиционная компания Белгорода». Был проведен анализ бизнес-процессов организации с помощью методологии функционального моделирования IDEF0. В результате данного анализа была выявлена необходимость внедрения автоматизированной системы, сформулированы цель и задачи дипломной работы.

В качестве методов и средств решения поставленных задач была использована клиент - серверная технология, которая позволяет существенно снизить нагрузку на рабочие станции системы (клиентские рабочие места) и переложить выполнение всей бизнес - логики на сервер. Вся накапливаемая информация хранится в базе данных под управлением СУБД Firebird. В качестве инструментального средства для разработки клиентского приложения была использована среда программирования C++ Builder, в качестве технологии доступа к данным InterBase Express.

По результатам выпускной квалификационной работы было разработано АРМ менеджера автотранспортного предприятия, которое позволяет:

- Учитывать информацию о контактах с клиентами;
- Оперативно получать информацию о клиентах с использованием функции поиска;
- Добавлять и редактировать информацию о клиентах;
- Формировать различные отчеты;
- Добавлять и редактировать информацию о водителях;
- Вести справочники маршрутов и вида перевозок;
- Добавлять и редактировать данные заказов.

Заключительной задачей, решение которой было отражено в данной работе, было обоснование экономической эффективности внедрения результатов проекта. Таким образом, цель выпускной квалификационной работы была достигнута.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Архангельский, А.Я. Приемы программирования в С++ Builder. Механизмы Windows, сети [Текст] / А.Я. Архангельский. – Москва: Бином 2004. – 656с.
2. Бондарь, А. InterBase и Firebird. Практическое руководство для умных пользователей и начинающих разработчиков. [Текст] / А.Бондарь. - СПб.: БХВ-Петербург, 2007г. – 383с.
3. Горев, А.Н., Эффективная работа с СУБД [Текст] / , С. Горев, А.Н., Ахаян, Р., Макашарипов, С. - СПб.: Питер, 1997.
4. Кренке Давид. Теория и практика построения баз данных.[Текст] / Кренке Давид. - 9-е изд. — СПб.: - Питер, 2005. – 859с.
5. Елманова, Н.И. Работа с базами данных в Borland С++ Builder[Текст]/ Н. Елманова. - Центр информационных Технологий Компьютер-Пресс, 1997, N 6, 7.
6. Свириденко, С.В. Информация и информационные технологии в современном обществе [Текст] /С.В. Свириденко. М.: МНЕПУ, 1993.
7. Хелен, Бори. Firebird: руководство разработчика баз данных [Текст] / Хелен, Бори. - СПб.: «БХВ-Петербург», 2007г.
8. Хомоненко, А.С. Работа с БД в С++Builder [Текст] / А.С.Хоменко. - Санкт-Петербург, «БХВ-Петербург», 2006г.
9. Афонин, А.М. Промышленная логистика: Учебное пособие [Текст] / А.М.Афонин.- Москва: Форум, 2013.-304с.
10. Гайдоенко, А.А. Логистика[Текст] / Гайдоенко. - Москва. КноРус, 2014.-267с.
11. Голубчик, А. М. Транспортно-экспедиторский бизнес: создание, становление, управление[Текст] / А. М. Голубчик. – М.: ТрансЛит, 2011. – 317 с.-
12. Герасимов, Б.И. Основы логистики [Текст]/ Б.И. Герасимов, В.В. Жариков, В.Д. Жариков. / Москва: - ИНФРА-М, 2010. – 304с.

13. Курочкин, Д.В. Логистика: [транспортная, закупочная, производственная, распределительная, складирование, информационная]: курс лекций [Текст]/ Д.В. Курочкин. – Минск: ФУАинформ, 2012. – 268с.
14. Николайчук, В.Е. Логистический менеджмент: учебник [Текст] / В.Е. Николайчук. – Москва: Дашков и К, 2012. – 978с.
15. Голубчик, А.М. Транспортно-экспедиторский бизнес: создание, становление, управление [Текст] / А.М. голубчик. – Москва: ТрансЛит, 2011. – 317с.
16. Голицына О.Л. Базы данных [Текст] / О.Л. Голицына, Н.В. Максимов, И.И. Попов. – Москва: Форум, 2004. – 352с.
17. Карпова И.П. Базы данных: Учебное пособие [Текст] / И.П. Карпова. – СПб.: Питер, 2013. – 240с.
18. Советов, Б.Я. Базы данных: теория и практика: Учебник для бакалавров [Текст] / Б.Я. Советов, В.В. Цехановский, В.Д. Чертовский.- Москва: Юрайт, 2013 – 463с.
19. Березин, В.М. Начальный курс С и С++ [Текст] / Б.И. Березин, С.Б. Березин. – Москва: ДИАЛОГ-МИФИ, 1996. – 137с.
20. Балдин, К.В. Информационные системы в экономике [Текст] / К.В. Балдин – Москва: ИНФРА-М, 2012 – 365с.
21. Гейн, А.Г. Основы информатики и вычислительной техники [Текст] / А.Г. Гейн. – Москва: Просвещение, 2014. – 245с.
22. Бородакий, Ю.В. Информационные технологии. Методы, процессы, системы [Текст] / Ю.В. Бородакий, Ю.Г. Лободинский. – Москва: ГЛТ, 2004. – 456с.
23. Гаврилов, М.В. Информационные технологии в коммерции: Учебное пособие [Текст] /Л.П. Гаврилов. – Москва: НИЦ ИНФРА-М, 2013. -238с.
24. Гвоздева, В.А Информатика, автоматизированные технологии и системы: Учебник [Текст] / В.А. Гвоздева. – Москва: ИД ФОРУМ НИЦ ИНФРА-М, 2013. - 544с.

25. Горев, А.Э. Информационные технологии в транспорте: Учебник для академического бакалавриата [Текст] / А.Э. Горев. – Люберцы: Юрайт, 2016. – 271с.
26. Емельянов, С.В. Информационные технологии и вычислительные системы [Текст] / С.В. Емельянов. – Москва: Ленанд, 2011. 84с.
27. Коноплева, И.А. Информационные технологии [Текст] / И.А. Коноплева, О.А. Хохлова, А.В. Денисов. – Москва: Проспект, 2015. – 328 с.
28. Петрунин, Ю.Ю. Информационные технологии анализа данных: Учебное пособие [Текст] / Ю.Ю. Петрунин. – Москва: КДУ, 2010. – 292. с.
29. Николайчук, В.Е. Транспортно-складская логистика [Текст] / В.Е. Николайчук. – Москва: Дашков и К, 2011. – 365с
30. Левиков, Г.А. Логистика, транспорт и экспедирование: Краткий словарь-справочник [Текст] / Г.А. Левиков. – Москва: ТрансЛит, 2008. – 224с.
31. Миротин, Л.Б. Логистика в автомобильном транспорте / Л.Б. Миротин. – Вологда: Инфра-Инженерия, 2015. -237с.
32. Холзнер, С. Visual C++ Учебный курс [Текст] / С. Холзнер. – СПб:Питер, 2001. – 576с.
33. Гуриков, С.Р. Введение в программирование на языке Visual C++ [Текст]/ С.В. Гуриков – Москва: ИД ФОРУМ НИЦ ИНФРА-М, 2013. - 448с.
34. Адрианова, А.А. Объектно-ориентированное программирование на C++ [Текст] / А.А. Адрианова, Л.Н. Исмагилов, Т.М. Мухтарова. – Казань: Казанский федеральный университет, 2012. – 124с.
35. Фридман, А.Л. Языки программирования C++ [Текст] / А.Л. Фридман. – Москва: Бином, 2006.- 523
36. Голубчик, А. М. Транспортно-экспедиторский бизнес: создание, становление, управление [Текст] / А. М. Голубчик. – М.: ТрансЛит, 2011. – 317 с.-

ПРИЛОЖЕНИЕ

Текст основных программных модулей

Unit1

```
//-----  
-----  
#include <vcl.h>  
#pragma hdrstop  
#include "Unit1.h"  
#include "Unit2.h"  
//-----  
-----  
#pragma package(smart_init)  
#pragma resource "*.dfm"  
TForm1 *Form1;  
//-----  
-----  
__fastcall TForm1::TForm1(TComponent* Owner)  
    : TForm(Owner)  
{  
}  
//-----  
-----  
void __fastcall TForm1::Button1Click(TObject *Sender)  
{  
    DataModule2->IBStoredProc1->ParamByName("F_NAME")->AsString=Edit1->Text;  
    DataModule2->IBStoredProc1->ParamByName("I_NAME")->AsString=Edit2->Text;  
    DataModule2->IBStoredProc1->ParamByName("O_NAME")->AsString=Edit3->Text;  
    DataModule2->IBStoredProc1->ParamByName("CL_TELEPHONE")->AsString=Edit4->Text;  
    DataModule2->IBStoredProc1->ParamByName("CL_ADRES")->AsString=Edit5->Text;  
    DataModule2->IBStoredProc1->Prepare();  
    DataModule2->IBStoredProc1->ExecProc();  
    Edit1->Text="";  
    Edit2->Text="";  
    Edit3->Text="";  
    Edit4->Text="";  
    Edit5->Text="";  
    DataModule2->IBQuery1->Close();  
    DataModule2->IBQuery1->Open();  
}  
//-----  
-----  
void __fastcall TForm1::Button2Click(TObject *Sender)
```

```

{
DataModule2->IBStoredProc2->ParamByName ("ID") -
>AsInteger=StrToInt (Edit6->Text) ;
DataModule2->IBStoredProc2->ParamByName ("F_NAME")->AsString=Edit7-
>Text;
DataModule2->IBStoredProc2->ParamByName ("I_NAME")->AsString=Edit8-
>Text;
DataModule2->IBStoredProc2->ParamByName ("O_NAME")->AsString=Edit9-
>Text;
DataModule2->IBStoredProc2->ParamByName ("CL_TELEPHONE") -
>AsString=Edit10->Text;
DataModule2->IBStoredProc2->ParamByName ("CL_ADRES") -
>AsString=Edit11->Text;
DataModule2->IBStoredProc2->Prepare ();
DataModule2->IBStoredProc2->ExecProc ();
Edit6->Text="";
Edit7->Text="";
Edit8->Text="";
Edit9->Text="";
Edit10->Text="";
Edit11->Text="";
DataModule2->IBQuery1->Close ();
DataModule2->IBQuery1->Open ();
}
//-----
-----

void __fastcall TForm1::Button3Click(TObject *Sender)
{
DataModule2->IBStoredProc3->ParamByName ("NOMER")->AsString=Edit12-
>Text;
DataModule2->IBStoredProc3->Prepare ();
DataModule2->IBStoredProc3->ExecProc ();
Edit12->Text="";
DataModule2->IBQuery1->Close ();
DataModule2->IBQuery1->Open ();
}
//-----
-----

void __fastcall TForm1::Button4Click(TObject *Sender)
{
if (RadioButton1->Checked==true){ //по фамилии
DataModule2->IBQuery1->SQL->Clear ();
DataModule2->IBQuery1->SQL->Add("select * from CLIENT order by
FIRST_NAME");}
if (RadioButton2->Checked==true){//по номеру
DataModule2->IBQuery1->SQL->Clear ();
DataModule2->IBQuery1->SQL->Add("select * from CLIENT order by
ID");}
DataModule2->IBQuery1->Open ();
}
//-----
-----

```

```

void __fastcall TForm1::Button5Click(TObject *Sender)
{
if (RadioButton3->Checked==true){// по фамилии
DataModule2->IBQuery1->SQL->Clear();
DataModule2->IBQuery1->SQL->Add("select * from CLIENT WHERE
FIRST_NAME like '"+Edit13->Text +"%' ");}
if (RadioButton4->Checked==true){// по номеру (диапазон)
DataModule2->IBQuery1->SQL->Clear();
DataModule2->IBQuery1->SQL->Add("select * from CLIENT WHERE
ID<='"+Edit14->Text +" AND ID>='"+Edit15->Text +"");}
DataModule2->IBQuery1->Open();
}
//-----
-----

void __fastcall TForm1::Button6Click(TObject *Sender)
{
DataModule2->IBStoredProc4->ParamByName ("MARSHRUT_NAIM") -
>AsString=Edit16->Text;
DataModule2->IBStoredProc4->ParamByName ("MARSHRUT_DLINA") -
>AsString=Edit17->Text;
DataModule2->IBStoredProc4->Prepare();
DataModule2->IBStoredProc4->ExecProc();
Edit16->Text="";
Edit17->Text="";
DataModule2->IBQuery3->Close();
DataModule2->IBQuery3->Open();
}
//-----
-----

void __fastcall TForm1::Button7Click(TObject *Sender)
{
DataModule2->IBStoredProc6->ParamByName ("ID") -
>AsInteger=StrToInt (Edit18->Text);
DataModule2->IBStoredProc6->ParamByName ("MARSHRUT_NAIM") -
>AsString=Edit19->Text;
DataModule2->IBStoredProc6->ParamByName ("MARSHRUT_DLINA") -
>AsString=Edit20->Text;
DataModule2->IBStoredProc6->Prepare();
DataModule2->IBStoredProc6->ExecProc();
Edit18->Text="";
Edit19->Text="";
Edit20->Text="";
DataModule2->IBQuery3->Close();
DataModule2->IBQuery3->Open();
}
//-----
-----

void __fastcall TForm1::Button8Click(TObject *Sender)
{
DataModule2->IBStoredProc5->ParamByName ("NOMER") ->AsString=Edit21-
>Text;
DataModule2->IBStoredProc5->Prepare();
}

```

```

DataModule2->IBStoredProc5->ExecProc();
Edit12->Text="";
DataModule2->IBQuery3->Close();
DataModule2->IBQuery3->Open();
}
//-----
-----

void __fastcall TForm1::Button9Click(TObject *Sender)
{
DataModule2->IBStoredProc7->ParamByName("NAIM_PEREVOZKI")->
AsString=Edit23->Text;
DataModule2->IBStoredProc7->Prepare();
DataModule2->IBStoredProc7->ExecProc();
Edit23->Text="";
DataModule2->IBQuery4->Close();
DataModule2->IBQuery4->Open();
}
//-----
-----

void __fastcall TForm1::Button10Click(TObject *Sender)
{
DataModule2->IBStoredProc8->ParamByName("ID")->
AsInteger=StrToInt(Edit24->Text);
DataModule2->IBStoredProc8->ParamByName("NAIM_PEREVOZKI")->
AsString=Edit25->Text;
DataModule2->IBStoredProc8->Prepare();
DataModule2->IBStoredProc8->ExecProc();
Edit24->Text="";
Edit25->Text="";
DataModule2->IBQuery4->Close();
DataModule2->IBQuery4->Open();
}
//-----
-----

void __fastcall TForm1::Button11Click(TObject *Sender)
{
DataModule2->IBStoredProc9->ParamByName("NOMER")->AsString=Edit26->
Text;
DataModule2->IBStoredProc9->Prepare();
DataModule2->IBStoredProc9->ExecProc();
Edit26->Text="";
DataModule2->IBQuery4->Close();
DataModule2->IBQuery4->Open();
}
//-----
-----

void __fastcall TForm1::Button12Click(TObject *Sender)
{
DataModule2->IBStoredProc10->ParamByName("F_NAME")->
AsString=Edit27->Text;
DataModule2->IBStoredProc10->ParamByName("I_NAME")->
AsString=Edit28->Text;

```

```

DataModule2->IBStoredProc10->ParamByName("O_NAME") -
>AsString=Edit29->Text;
DataModule2->IBStoredProc10->ParamByName("VODITEL_TEL") -
>AsString=Edit30->Text;
DataModule2->IBStoredProc10->ParamByName("VOD_MARKA") -
>AsString=Edit31->Text;
DataModule2->IBStoredProc10->Prepare();
DataModule2->IBStoredProc10->ExecProc();
Edit27->Text="";
Edit28->Text="";
Edit29->Text="";
Edit30->Text="";
Edit31->Text="";
DataModule2->IBQuery5->Close();
DataModule2->IBQuery5->Open();
}
//-----
-----

void __fastcall TForm1::Button13Click(TObject *Sender)
{
DataModule2->IBStoredProc11->ParamByName("ID") -
>AsInteger=StrToInt(Edit37->Text);
DataModule2->IBStoredProc11->ParamByName("F_NAME") -
>AsString=Edit32->Text;
DataModule2->IBStoredProc11->ParamByName("I_NAME") -
>AsString=Edit33->Text;
DataModule2->IBStoredProc11->ParamByName("O_NAME") -
>AsString=Edit34->Text;
DataModule2->IBStoredProc11->ParamByName("CL_TELEPHONE") -
>AsString=Edit35->Text;
DataModule2->IBStoredProc11->ParamByName("CL_AVTO") -
>AsString=Edit36->Text;
DataModule2->IBStoredProc11->Prepare();
DataModule2->IBStoredProc11->ExecProc();
Edit37->Text="";
Edit32->Text="";
Edit33->Text="";
Edit34->Text="";
Edit35->Text="";
Edit36->Text="";
DataModule2->IBQuery5->Close();
DataModule2->IBQuery5->Open();
}
//-----
-----

void __fastcall TForm1::Button14Click(TObject *Sender)
{
DataModule2->IBStoredProc12->ParamByName("NOMER") -
>AsString=Edit38->Text;
DataModule2->IBStoredProc12->Prepare();
DataModule2->IBStoredProc12->ExecProc();
Edit38->Text="";
}

```



```

DataModule2->IBQuery5->Close();
DataModule2->IBQuery5->Open();
}
//-----
-----
void __fastcall TForm1::Button15Click(TObject *Sender)
{
if (RadioButton6->Checked==true){
DataModule2->IBQuery3->SQL->Clear();
DataModule2->IBQuery3->SQL->Add("select * from MARSHRUT order by
NAIMENOVANIE");}
if (RadioButton5->Checked==true){//по номеру
DataModule2->IBQuery3->SQL->Clear();
DataModule2->IBQuery3->SQL->Add("select * from MARSHRUT order by
ID");}
DataModule2->IBQuery3->Open();
if (RadioButton7->Checked==true){//по номеру
DataModule2->IBQuery3->SQL->Clear();
DataModule2->IBQuery3->SQL->Add("select * from MARSHRUT order by
DLINA");}
DataModule2->IBQuery3->Open();
}
//-----
-----
void __fastcall TForm1::Button16Click(TObject *Sender)
{
if (RadioButton8->Checked==true){//по номеру
DataModule2->IBQuery4->SQL->Clear();
DataModule2->IBQuery4->SQL->Add("select * from VID_PEREVOZKI order
by ID");}
DataModule2->IBQuery4->Open();
if (RadioButton9->Checked==true){//по номеру
DataModule2->IBQuery4->SQL->Clear();
DataModule2->IBQuery4->SQL->Add("select * from VID_PEREVOZKI order
by NAIMENOVANIE");}
DataModule2->IBQuery4->Open();
}
//-----
-----
void __fastcall TForm1::Button17Click(TObject *Sender)
{
if (RadioButton10->Checked==true){//по номеру
DataModule2->IBQuery5->SQL->Clear();
DataModule2->IBQuery5->SQL->Add("select * from VODITEL order by
ID");}
DataModule2->IBQuery5->Open();
if (RadioButton11->Checked==true){//по номеру
DataModule2->IBQuery5->SQL->Clear();
DataModule2->IBQuery5->SQL->Add("select * from VODITEL order by
FIRST_NAME");}
DataModule2->IBQuery5->Open();
}
}

```

```

//-----
-----
void __fastcall TForm1::Button18Click(TObject *Sender)
{
DataModule2->IBStoredProc13->ParamByName("ZAKAZ_COST")-
>AsString=Edit39->Text;
DataModule2->IBStoredProc13->ParamByName("ZAKAZ_SROCHNOST")-
>AsString=Edit40->Text;
DataModule2->IBStoredProc13->ParamByName("ZAKAZ_CLIENT")-
>AsString=Edit41->Text;
DataModule2->IBStoredProc13->ParamByName("ZAKAZ_MARSHRUT")-
>AsString=Edit42->Text;
DataModule2->IBStoredProc13->ParamByName("ZAKAZ_PEREVOZKI")-
>AsString=Edit43->Text;
DataModule2->IBStoredProc13->ParamByName("ZAKAZ_VODITEL")-
>AsString=Edit44->Text;
DataModule2->IBStoredProc13->ParamByName("DATE_ZAK")-
>AsString=DateToStr(DateTimePicker1->Date);
DataModule2->IBStoredProc13->Prepare();
DataModule2->IBStoredProc13->ExecProc();
Edit39->Text="";
Edit40->Text="";
Edit41->Text="";
Edit42->Text="";
Edit43->Text="";
Edit44->Text="";
DataModule2->IBQuery6->Close();
DataModule2->IBQuery6->Open();
DataModule2->IBQuery8->Close();
DataModule2->IBQuery8->Open();
DataModule2->IBQuery9->Close();
DataModule2->IBQuery9->Open();
DataModule2->IBQuery10->Close();
DataModule2->IBQuery10->Open();
}
//-----
-----

void __fastcall TForm1::Button19Click(TObject *Sender)
{
DataModule2->IBStoredProc14->ParamByName("ID")-
>AsInteger=StrToInt(Edit45->Text);
DataModule2->IBStoredProc14->ParamByName("ZAKAZ_COST")-
>AsString=Edit46->Text;
DataModule2->IBStoredProc14->ParamByName("ZAKAZ_SROCHNOST")-
>AsString=Edit47->Text;
DataModule2->IBStoredProc14->ParamByName("ZAKAZ_CLIENT")-
>AsString=Edit48->Text;
DataModule2->IBStoredProc14->ParamByName("ZAKAZ_MARSHRUT")-
>AsString=Edit49->Text;
DataModule2->IBStoredProc14->ParamByName("ZAKAZ_PEREVOZKI")-
>AsString=Edit50->Text;
DataModule2->IBStoredProc14->ParamByName("ZAKAZ_VODITEL")-

```

```

>AsString=Edit51->Text;
DataModule2->IBStoredProc14->ParamByName("DATE_ZAK")-
>AsString=DateToStr(DateTimePicker2->Date);
DataModule2->IBStoredProc14->Prepare();
DataModule2->IBStoredProc14->ExecProc();
Edit45->Text="";
Edit46->Text="";
Edit47->Text="";
Edit48->Text="";
Edit49->Text="";
Edit50->Text="";
Edit51->Text="";
DataModule2->IBQuery6->Close();
DataModule2->IBQuery6->Open();
DataModule2->IBQuery8->Close();
DataModule2->IBQuery8->Open();
DataModule2->IBQuery9->Close();
DataModule2->IBQuery9->Open();
DataModule2->IBQuery10->Close();
DataModule2->IBQuery10->Open();
}
//-----
-----
void __fastcall TForm1::Button20Click(TObject *Sender)
{
DataModule2->IBStoredProc15->ParamByName("NOMER")-
>AsString=Edit52->Text;
DataModule2->IBStoredProc15->Prepare();
DataModule2->IBStoredProc15->ExecProc();
Edit52->Text="";
DataModule2->IBQuery6->Close();
DataModule2->IBQuery6->Open();
DataModule2->IBQuery8->Close();
DataModule2->IBQuery8->Open();
DataModule2->IBQuery9->Close();
DataModule2->IBQuery9->Open();
DataModule2->IBQuery10->Close();
DataModule2->IBQuery10->Open();
}
//-----
-----
void __fastcall TForm1::Button21Click(TObject *Sender)
{
QuickRep1->Preview();
}
//-----
-----
void __fastcall TForm1::Button22Click(TObject *Sender)
{
QuickRep2->Preview();
}
//-----

```

```

-----
void __fastcall TForm1::Button23Click(TObject *Sender)
{
if (RadioButton12->Checked==true){//по номеру
DataModule2->IBQuery6->SQL->Clear();
DataModule2->IBQuery6->SQL->Add("select * from ZAKAZ order by
ID");}
DataModule2->IBQuery6->Open();
if (RadioButton13->Checked==true){
DataModule2->IBQuery6->SQL->Clear();
DataModule2->IBQuery6->SQL->Add("select * from ZAKAZ order by
COST");}
DataModule2->IBQuery6->Open();
if (RadioButton14->Checked==true){
DataModule2->IBQuery6->SQL->Clear();
DataModule2->IBQuery6->SQL->Add("select * from ZAKAZ order by
SROCHNOST");}
DataModule2->IBQuery6->Open();
if (RadioButton15->Checked==true){
DataModule2->IBQuery6->SQL->Clear();
DataModule2->IBQuery6->SQL->Add("select * from ZAKAZ order by
ID_CLIENT");}
DataModule2->IBQuery6->Open();
if (RadioButton16->Checked==true){
DataModule2->IBQuery6->SQL->Clear();
DataModule2->IBQuery6->SQL->Add("select * from ZAKAZ order by
ID_MARSHRUT");}
DataModule2->IBQuery6->Open();
if (RadioButton17->Checked==true){
DataModule2->IBQuery6->SQL->Clear();
DataModule2->IBQuery6->SQL->Add("select * from ZAKAZ order by
ID_PEREVOZKI");}
DataModule2->IBQuery6->Open();
if (RadioButton18->Checked==true){
DataModule2->IBQuery6->SQL->Clear();
DataModule2->IBQuery6->SQL->Add("select * from ZAKAZ order by
ID_VODITEL");}
DataModule2->IBQuery6->Open();
if (RadioButton19->Checked==true){
DataModule2->IBQuery6->SQL->Clear();
DataModule2->IBQuery6->SQL->Add("select * from ZAKAZ order by
DATE");}
DataModule2->IBQuery6->Open();
}
//-----
-----

```

Unit 2

```

//-----
-----

```

```

#include <vcl.h>
#pragma hdrstop
#include "Unit2.h"
//-----
-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TDataModule2 *DataModule2;
//-----
-----
__fastcall TDataModule2::TDataModule2(TComponent* Owner)
    : TDataModule(Owner)
{
}
//-----
-----
void __fastcall TDataModule2::IBQuery1AfterScroll(TDataSet
*DataSet)
{
DataModule2->IBQuery2->Close();
DataModule2->IBQuery2->ParamByName("NOMER_CLIENT")->
AsInteger=DataModule2->IBQuery1->FieldByName("ID")->AsInteger;
DataModule2->IBQuery2->Open();
}
//-----
-----
void __fastcall TDataModule2::IBQuery6AfterScroll(TDataSet
*DataSet)
{
DataModule2->IBQuery2->Close();
DataModule2->IBQuery2->ParamByName("NOMER_CLIENT")->
AsInteger=DataModule2->IBQuery1->FieldByName("ID")->AsInteger;
DataModule2->IBQuery2->Open();
}
//-----
-----

```