

Министерство науки и высшего образования РФ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Дагестанский государственный технический университет»

Факультет Информационных систем, финансов и аудита
Направление 09.03.03-«Прикладная информатика»
Профиль: «Прикладная информатика в экономике»
Кафедра Информационных технологий и прикладной информатики в
экономике

Допустить к защите:
заведующий кафедрой ИТиПИВЭ,
д.э.н., профессор Абдулгалимов А.М.

(подпись)

« ____ » _____ 2020г.

Пояснительная записка

к выпускной квалификационной работе (бакалаврской)

на тему:

Разработка OLAP системы для торговой сети магазинов DNS

Дипломник _____ В.М. Валиев
Руководитель _____ М.М. Мурадов
Нормоконтролер _____ Н.А. Гаджиева

Министерство науки и высшего образования Российской Федерации
ФГБОУ ВО
«Дагестанский государственный технический университет»

Факультет: Информационных систем, финансов и аудита
Направление: 09.03.03 – «Прикладная информатика»
Профиль: «Прикладная информатика в экономике»
Кафедра: Информационных технологий и прикладной информатики в экономике

УТВЕРЖДАЮ
 Заведующий кафедрой ИТиПИВЭ,
 д.э.н., профессор А.М. Абдулгалимов

_____ подпись
 « 02 » 03 2020 г.

ЗАДАНИЕ
на дипломный проект

Студенту(ке) 4 курса И631 группы _____ Валиеву Вали Магомедовичу

_____ (Ф.И.О)

1. Тема дипломного проекта Разработка OLAP системы для торговой сети магазинов DNS

2. Тема утверждена приказом ректора по университету от 29.02.2020г. № 339-С

3. Исходные данные (технические; экономические; организационные и другие требования) для выполнения дипломного (ой) проекта (работы).

3.1. Данные нормативно-правовых документов функционирования сети магазинов DNS, данные по продажам товаров, нормативно-справочные данные.

3.2. Ввод информации должен осуществляться как с клавиатуры, так и с помощью других устройств.

3.3. Используемая вычислительная техника по своим характеристикам должна иметь параметры, позволяющие эффективно работать операционной системе Windows 7/10 и выше, пакетам программ используемым в дипломном проекте, а также самой разработанной программе.

3.4. Программные средства, используемые в дипломном проекте, должны функционировать в операционной среде не ниже Windows 7/10.

3.5. Технико-экономическое обоснование дипломного проекта, а так же сам дипломный проект по содержанию и оформлению должны соответствовать требованиям методических указаний к выполнению ДП, изданных на кафедре ИТиПИВЭ

4. Содержание пояснительной записки (перечень вопросов подлежащих разработке)

4.1. Введение: общие сведения о дипломном проекте; цели и задачи проекта; название объекта управления, вычислительная техника, на которую ориентирован дипломный проект; новизна и актуальность разработки; перечень задач, решаемых в ДП.

4.2. Аналитическая часть: технико-экономическая характеристика объекта управления; экономическая сущность комплекса решаемых задач; обоснование необходимости и цели использования вычислительной техники; общая характеристика организации машинной обработки информации; формализация расчетов; обоснование проектных решений по технологии сбора и обработки информации.

4.3. Проектная часть: инфологическая или информационная модель предметной области и ее описание; характеристика входной оперативной информации (входных документов);

описание файлов и записей в базах данных; характеристика промежуточной и результирующей информации; машинная реализация комплекса задач; схема взаимосвязи программных модулей; блок-схемы алгоритмов основных расчетных модулей.

4.4. Оценка экономической эффективности проекта: выбор и обоснование методики расчета экономической эффективности проекта; расчет показателей экономической эффективности проекта

5. Перечень рекомендуемой литературы:

5.1. Абдулгалимов А.М., Мурадов М.М., Адеева М.Г. Методические указания к выполнению дипломных проектов студентами специальности 080801 – «Прикладная информатика в экономике». – Махачкала : ДГТУ, 2013.

5.2. Архангельский А.Я. «Программирование в С++ Builder 6» - Москва: «Издательство БИНОМ», 2008г.- 1152с.

5.3. Уткин В.Б., Балдин К.В. «Информационные системы в экономике». Учебник для студентов ВУЗа - 2-ое изд.- Москва: Издательский центр «Академия», 2006г.

5.4. Диго С.М. Проектирование и использование баз данных. - М.: Финансы и статистика, 2011.

5.5. <http://DNS-dagestan.ru> – сайт ООО "DNS".

6. Перечень разрабатываемого графического (иллюстративного) материала:

Наименование графического материала	Количество листов	Формат
Постановка задач проекта	1	A1
Структурная схема объекта управления	1	A1
Инфологическая модель предметной области	1	A1
Схема взаимосвязи программных модулей	1	A1
Выходные формы документов	1	A1

7. Консультанты по разделам дипломного (ой) проекта (работы)

Раздел дипломного (ой) проекта (работы)	Ф.И.О. консультанта
Аналитическая часть	Адеева М.Г.
Проектная часть	Мурадов М.М.
Экономическая часть	Тагиев Р.Х.

8. Календарный план-график выполнения по проектированию

Содержание работы	Объем работы в %	Контрольные сроки
1. Введение	5	02.03. – 07.03.20г.
2. Аналитическая часть	35	
Технико-экономическая характеристика ОУ	10	10.03. – 14.03.20г.
Эконом. сущность комплекса задач.	10	16.03. – 21.03.20г.
Обоснование и выбор ВТ, организация машинной обработки	5	23.03. – 28.03.20г.
Формализация расчетов и обоснование проектных решений по информационному и программному обеспечению	10	30.03. – 04.04.20г.
3. Проектная часть	50	
Инфологическая, даталогическая и физическая модели предметной области	25	06.04. – 22.04.20г.
Машинная реализация комплекса задач.	25	23.04. – 16.05.20г.
4. Обоснование экономической эффективности проекта.	10	18.05. – 30.05.20г.

Дата выдачи задания

«02» марта 2020 г.

Дата сдачи дипломного (ой) проекта (работы) на кафедру

« 01 » июня 2020 г.

Руководитель дипломного (ой) проекта (работы) _____

подпись

Ф.И.О.

Студент _____

подпись

Ф.И.О.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	6
1. АНАЛИТИЧЕСКАЯ ЧАСТЬ.....	9
1.1. Техничко-экономическая характеристика объекта управления.....	9
1.2. Экономическая сущность комплекса задач	11
1.3. Обоснование необходимости и цели использования вычислительной техники для решения данного комплекса задач	17
1.4. Общая характеристика организации машинной обработки	19
1.5. Формализация расчетов	21
1.6. Обоснование проектных решений по информационному обеспечению комплекса задач	22
1.7. Обоснование проектных решений по программному обеспечению комплекса задач	26
1.8. Обоснование проектных решений по технологии сбора, передачи, обработки и выдачи информации	33
2. ПРОЕКТНАЯ ЧАСТЬ.....	35
2.1. Информационное обеспечение комплекса задач.....	35
2.1.1. Инфологическая модель и её описание	35
2.1.2. Характеристика входной информации	38
2.1.3. Характеристика результатной информации	49
2.2. Машинная реализация комплекса задач	52
2.2.1. Схема взаимосвязи программных модулей	52
2.2.2. Организация технологического процесса сбора, передачи, обработки и выдачи информации	58
2.2.2.1. Схема технологического процесса сбора, передачи, обработки и выдачи информации и ее описание	58
2.2.2.2. Инструкционные карты основных операций технологического процесса.....	59
3. ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ.....	67

3.1. Выбор и обоснование методики расчета экономической эффективности проекта	67
3.1.1. Разработка плана выполнения работ.....	70
3.1.2. Трудоемкость разработки программного обеспечения	75
3.2. Расчет показателей экономической эффективности проекта	77
3.2.1. Смета затрат на разработку программы	77
3.2.2. Расчет среднегодовых затрат на функционирование системы	79
3.2.3. Расчет показателей экономической эффективности	84
ЗАКЛЮЧЕНИЕ	85
СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ	86
ПРИЛОЖЕНИЯ	88

ВВЕДЕНИЕ

На сегодняшний день проблема хранения и обработки больших массивов информации является одной из важных во всем мире. Для больших компаний, в частности крупных ритейлеров, эта проблема является особенно актуальной, так как через них за один день могут проходить тысячи и миллионы транзакций. Тысячи различных решений были придуманы для решения данной проблемы. Результатом является то, что практически во всех компаниях используют системы автоматизации повседневной деятельности – OLTP-системы. Но помимо этого не менее важной является обработка и анализ всей информации. Почти всегда данную проблему практически невозможно решить средствами существующих OLTP-систем, так как в их основе лежат другие принципы, и они решают другие задачи. В результате чего информация недоступна управляющему персоналу компаний, которые отвечают за принятие решений. Это приводит к тому, что возникает потребность в системах, которые могли бы использовать информацию из OLTP-систем, не нарушая ход их работы, и при этом главной целью которых была бы поддержка принятия аналитических решений. Такими системами являются OLAP-системы или хранилища данных.

Процесс принятия решений является неотъемлемой частью управления любым бизнесом[6]. Ежедневно в компаниях принимаются различные решения по дальнейшему развитию и текущим операциям. Некоторые из этих решений могут целиком изменить положение компании на рынке.

Процесс принятия решений важен для любой компании, будь то огромная сеть магазинов по всему миру, или небольшой магазин. Каждый день владельцы и менеджеры решают, какие продукты им закупать, какие скидки и кому предложить, какие услуги предоставлять. Зачастую именно от того, какое решение будет принято, зависит будущее бизнеса. Правильное решение может быть выбрано исключительно исходя из актуальной информации о текущем положении дел в компании. И если для небольших компа-

ний данная информация может быть получена и без внедрения дополнительных систем, то для большого и среднего бизнеса обработка и анализ всего объема информации вручную практически невозможна.

Хранилища данных, будучи спроектированы правильным образом, предоставляют все необходимые механизмы для доступа к информации, важной для принятия решений в компании в любой момент времени. А так же, делают доступ к информации максимально удобным, а саму информацию максимально достоверной. Гибкость хранилищ данных позволяет обеспечить все будущие потребности в компании, за счет внесения исключительно небольших изменений в архитектуру хранилища.

Целью данного дипломного проекта является проектирование OLAP-системы для торговой сети магазинов DNS.

Для достижения поставленной цели, необходимо решить следующие задачи:

- изучить предметную область, т. е. процесс управления в торговой сети магазинов (понятие, сущность, миссию, организационную структуру и систему управления);
- провести анализ существующей организации бизнес-процессов;
- осуществить постановку задачи автоматизации бизнес-процессов;
- провести работы по созданию проекта ИС и с помощью разработанной ИС осуществить анализ эффективности OLAP-системы;
- разработать программное обеспечение для управления учета торговых операций и проведения многомерного анализа;
- выполнить расчет экономической эффективности разработанной автоматизированной информационной системы.

Дипломный проект состоит из введения, 3-х глав, заключения, списка литературы, которая использовалась при написании дипломного проекта и приложений.

Во введении выявлена актуальность рассматриваемой темы, указаны цель и задачи, определены предмет и объект исследования.

В первой главе рассмотрены теоретические аспекты процессов протекающих в объекте управления, обоснован выбор технических и программных средств автоматизации, информационного обеспечения и методов сбора и обработки информации.

Во второй главе представлены результаты разработки информационного и программного обеспечения OLAP-системы.

В третьей главе представлены результаты экономического обоснование разработки OLAP-системы.

В заключении сделаны выводы по проделанной работе.

При проектировании использовались СУБД Microsoft Access 2017, объектно-ориентированный язык программирования Delphi XE, CASE-средство ERWin, которое позволяет создавать графические модели бизнес-процессов.

Для расчета экономической эффективности автоматизированной системы выбрана методика расчета экономической эффективности, выполнен расчет показателей экономической эффективности. Расчет показал экономическую эффективность разработки.

1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

1.1. Техничко-экономическая характеристика объекта управления

Компания DNS смогла за короткие сроки стать продавцом техники и электроники на территории всей России. Нестандартный подход к продвижению товаров и отношениям с покупателями смогли обеспечить высокую популярность ритейлеру. За время своего развития DNS смогла построить диверсифицированный бизнес всероссийского масштаба.

Краткая информация:

- Название компании: *DNS*.
- Правовая форма деятельности: *Группа компаний*.
- Вид деятельности: *розничная продажа цифровой, бытовой техники, производство, строительство*.
- Выручка за 2019 год: *151,9 млрд руб.*
- Бенефициары: *Дмитрий Алексеев, Константин Богданенко, Юрий Карцов и др.*
- Численность персонала: *более 15 тыс. чел.*
- Сайт компании: www.dns-shop.ru.

Группа компаний DNS обладает крупной сетью розничных магазинов по реализации цифровой и бытовой техники. Помимо этого, ведется производство, сборка ноутбуков и других технических товаров. Большую долю в объеме продаж занимает интернет-торговля. История организации насчитывает 10 лет, и за это время DNS стала одним из лидеров российского рынка, предлагая качественные товары, сервис и привлекательные цены. Штаб-квартира расположена во Владивостоке.

Компания DNS (Digital Network System) была основана командой из 10 человек, друзей-программистов, во Владивостоке в 1998 году. Основатели будущего лидера российского рынка электроники и бытовой техники сняли офисное помещение, где и открыли магазин компьютеров, в одном из помещений которого также осуществляли сборку техники.

Основными источниками выручки были:

- аксессуары для компьютеров – 30%;
- ноутбуки, нетбуки – 18%;
- мобильные телефоны, смартфоны – 16%;
- планшеты – 13%;
- телевизионная техника – 11% и др.

Помимо небольших магазинов, DNS стала открывать большие супермаркеты и гипермаркеты площадью свыше 1000 кв. м – к 2013 году их насчитывалось 25 шт., включая гипермаркеты «Фрау-техника». Сборочные производства были запущены в Новосибирске и подмосковном регионе.

По размеру выручки DNS является одной из ведущих бизнес-организаций России и занимает 46-е место в списке крупнейших частных компаний РФ по версии журнала Forbes в 2017 году.

Таблица 1.

Изменение выручки компании DNS за 2010-2017 гг.

Год	Размер выручки
2019	233,4 млрд
2018	151,9 млрд
2017	135 млрд
2016	115,9 млрд
2015	110,6 млрд
2014	86,4 млрд
2013	30,2 млрд
2012	23 млрд

Следует отметить, что успех DNS возможен благодаря продуманной маркетинговой и ценовой политике. Многие товары потребители могут приобрести по ценам ниже, чем у конкурентов.

В 2017 г. компания DNS пришла в Республику Дагестан. Так в г. Махачкале в открыты 4 ТЦ. Это следующие супермаркеты:

- [ТЦ Ардар](#) ул. Малыгина, дом 37, корпус 3;
- [ТЦ Верас](#) ул. Гамидова, д. 18, корп. Ж;

- ТЦ Маэстро ул. Имама Шамиля, дом 4, корпус 1;

- ТЦ Метро ул. Толстого, д. 6.

Сеть магазинов «DNS» обладает полной хозяйственной самостоятельностью в вопросах определения формы управления, принятия хозяйственных решений, сбыта, установления цен, оплаты труда, распределения чистой прибыли[14].

Структурная схема организации «DNS Дагестан» изображена на рис. 1.1.

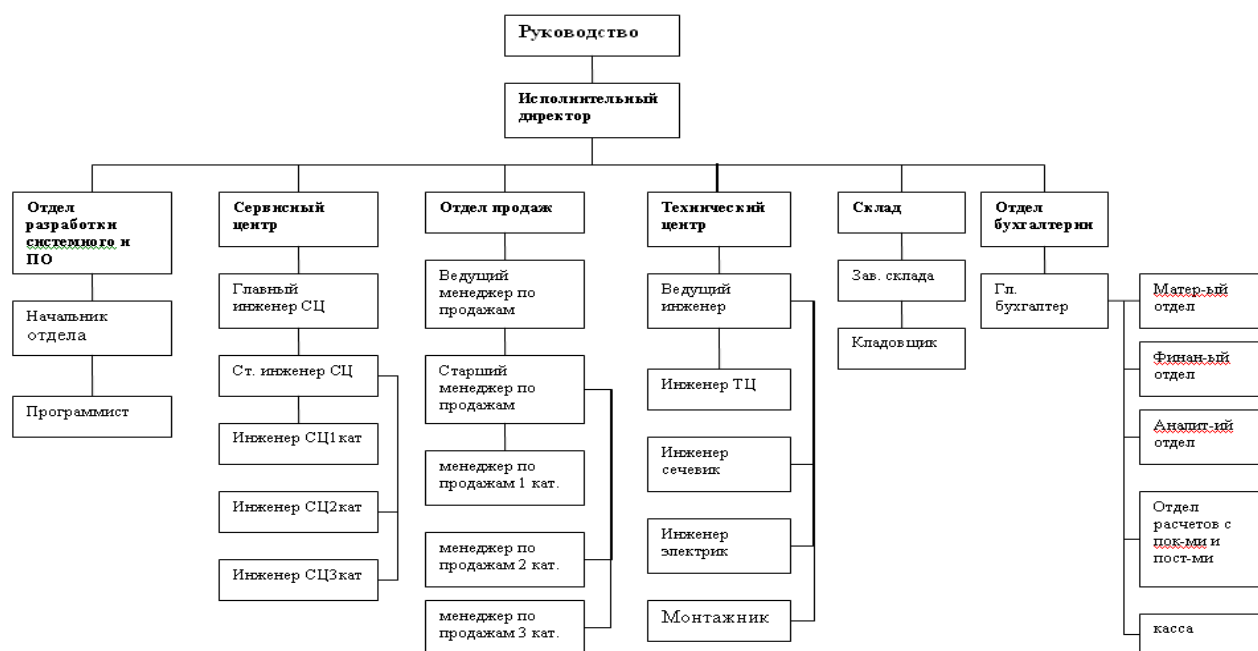


Рис. 1.1. Организационная структура «ДНС – Дагестан»

1.2. Экономическая сущность комплекса задач

Информационные системы масштаба предприятия, как правило, содержат приложения, предназначенные для комплексного многомерного анализа данных, их динамики, тенденций и т.п. Такой анализ в конечном итоге призван содействовать принятию решений. Нередко эти системы так и называются — системы поддержки принятия решений.

Принять любое управленческое решение, невозможно не обладая необходимой для этого информацией, обычно количественной. Для этого необходимо создание хранилищ данных. Хранилище данных – это процесс сбора,

отсеивания и предварительной обработки данных с целью представления результирующей информации пользователям для статистического анализа и аналитических отчетов. Ральф Кинболл описывал хранилища данных как «место, где люди могут получить доступ к своим данным». Он же сформулировал основные требования к хранилищам данных:

- поддержка высокой скорости данных из хранилища;
- поддержка внутренней непротиворечивости данных;
- возможность получения и сравнения данных;
- наличие удобных утилит просмотра данных хранилища;
- полнота и достоверность хранимых данных;
- поддержка качественного процесса пополнения данных.

Хранилище данных (англ. Data Warehouse) - предметно-ориентированная информационная база данных, специально разработанная и предназначенная для подготовки отчетов и бизнес-анализа с целью поддержки принятия решений в организации. Строится на базе систем управления базами данных и систем поддержки принятия решений. Данные, поступающие в хранилище данных, как правило, доступны только для чтения.

Использование технологии хранилищ данных предполагает наличие в системе следующих компонентов:

- оперативных источников данных;
- средств переноса и трансформации данных;
- метаданных – включают каталог хранилища и правила преобразования данных при загрузке их из оперативных баз данных;
- реляционного хранилища;
- OLAP-хранилища;
- средств доступа и анализа данных.

Назначение перечисленных компонентов таково. Оперативные данные собираются из различных источников. Поступившие оперативные данные очищаются, интегрируются и складываются в реляционные хранилище. Они уже доступны для анализа при помощи средств построения отчетов. Затем

данные (полностью или частично) подготавливаются с использованием средств переноса и трансформации данных для OLAP-анализа, который реализуется применением средств доступа и анализа данных. При этом они могут быть загружены в специальную базу данных OLAP или оставаться в реляционном хранилище.

Важнейшим элементом хранилища являются метаданные, т.е. данные о структуре, размещении, трансформации данных, которые используются любыми процессами хранилища. Метаданные могут быть востребованы для различных целей, например: извлечения и загрузки данных; обслуживании хранилища и запросов. Метаданные для различных процессов могут иметь различную структуру, т.е. для одного и того же элемента данных может существовать несколько вариантов метаданных.

Итак, хранилища данных являются структурированными. Они содержат базовые данные, которые образуют единый источник для обработки данных во всех системах поддержки принятия решений. Элементарные данные, присутствующие в хранилище, могут быть представлены в различной форме. Хранилища данных исключительно велики, поскольку в них содержатся интегрированные и детализированные данные.

Эти характеристики являются общими для всех хранилищ данных. Но, несмотря на то, что хранилища обладают общими свойствами, разные типы хранилищ имеют свои индивидуальные особенности.

Для работы с хранилищем данных используются СУБД, к которым предъявляются специальные требования:

- высокая производительность загрузки данных;
- возможность обработки данных на уровне загрузки;
- наличие средств управления качеством данных;
- высокая производительность запросов;
- широкая масштабируемость по размеру и количеству пользователей;
- возможность организации сети хранилищ данных;
- наличие средств администрации хранилищ данных;

- поддержка интегрированного многомерного анализа;
- расширенный набор функциональных средств запросов.

OLAP(On-Line Analytical Processing)– это технология комплексного многомерного анализа данных, это ключевой компонент организации хранилищ данных. В 1993 г. эта технология была описана Эдгером Коддом. Для упрощения анализа была предложена и разработаны концепция хранилища данных. Предполагается, что такое хранилище содержит сведения, поступающие от разных источников, а так же интегрированные данные, получаемые в результате анализа первичных данных. Естественно, для поддержки предложенной концепции потребовались специальные средства управления процессом хранения и обработки информации, к которым относятся инструментальные средства OLAP-технологии.

Выделяют три категории данных в хранилище:

- детальные – данные, соответствующие элементарным фиксируемым событиям на предприятии (например, оказание услуги);
- агрегированные – данные, полученные путем суммирования детальных данных по какому-либо измерению (например, по дате оказания услуги, либо по ее типу). Для соблюдения требования быстрого доступа к информации часть агрегированных данных, к которым пользователи обращаются наиболее часто, необходимо хранить в хранилище. При этом невозможно избежать избыточности данных. Для ее минимизации те данные, к которым пользователи обращаются достаточно редко, должны вычисляться в ходе выполнения запросов;
- метаданные – «данные о данных»; информация о данных, содержащихся в хранилище. Метаданные описывают, что хранится в хранилище, кто использует эти данные, где они хранятся, как они используются, когда и с какой целью производятся операции над ними.

Хранилища данных состоят из двух типов таблиц: таблицы фактов (facts) и таблицы измерений (dimension). Таблицы фактов содержат информацию о событиях, значимых для компании и анализа ее деятельности (про-

дажи, оказание услуг). Измерения – это атрибуты событий, по которым может производиться фильтрация или агрегация фактов. Измерения могут иметь иерархию и связываются с таблицами фактов с помощью внешних ключей.

Связи между фактами и измерениями обуславливают две возможные схемы представления данных:

1. Звезда – таблица фактов денормализована. С ней связан набор таблиц измерений, характеризующих события, заносимые в таблицу фактов. Данная схема понятна пользователю и может быть удобно представлена на диаграмме (рис. 1.2).

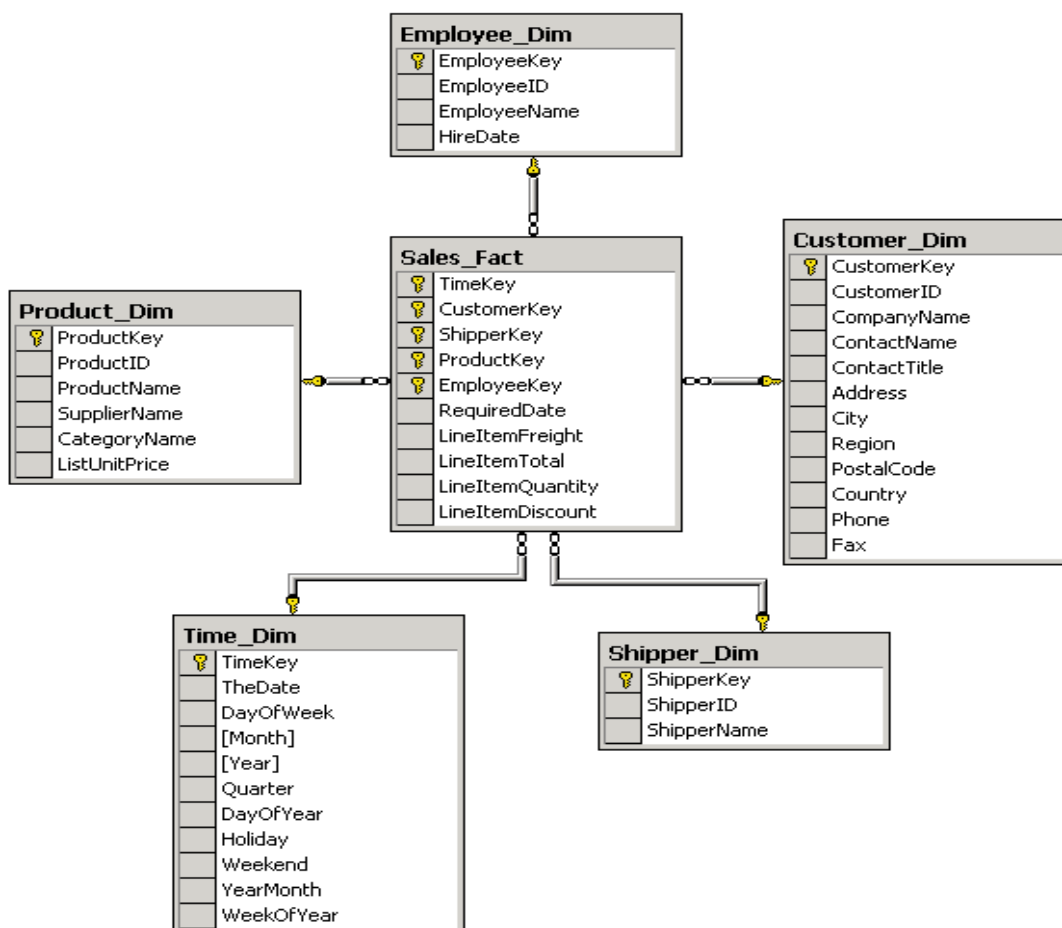


Рис.1.2. – Схема «звезда»

2. Снежинка – развитие схемы «звезда», в которой отдельные таблицы создаются для разных уровней иерархии таблиц измерений. Это позволяет избежать избыточности данных в таблицах измерений, что ведет к

ускорению процесса анализа данных. Однако при этом усложняется процедура добавления данных в хранилище, так как необходимо работать с большим числом таблиц. Кроме того, такая схема менее прозрачна для пользователя (рис.1.3).

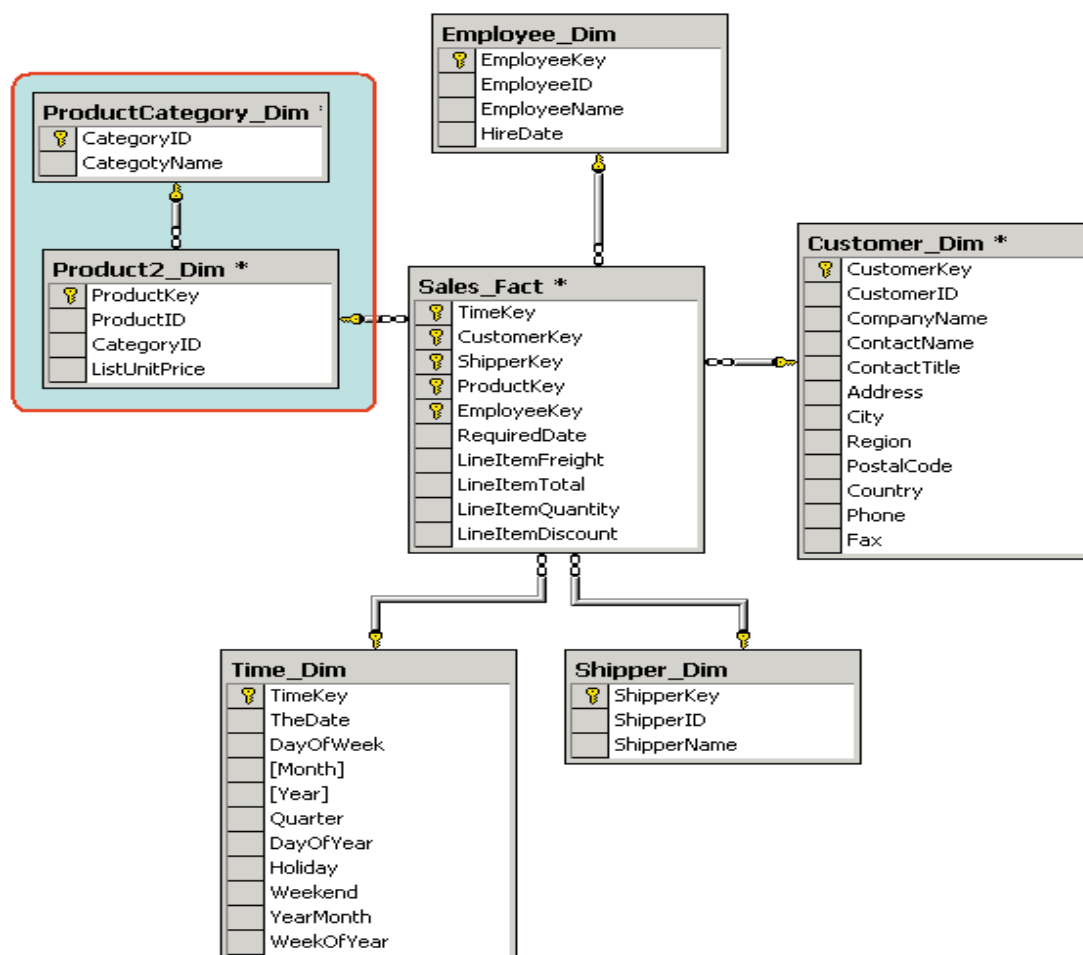


Рис. 1.3. – Схема «снежинка»

Таким образом, хранилище данных – это база данных с особой структурой, предназначенной для упрощения последующей обработки данных.

Суть технологии OLAP состоит в загрузке данных из различных источников (или из хранилища данных, в котором они уже загружены) и представлении их в виде структуры, которая называется OLAP-куб. Куб – это данные из таблицы фактов, агрегированные по всем уровням всех измерений (рис.1.4).

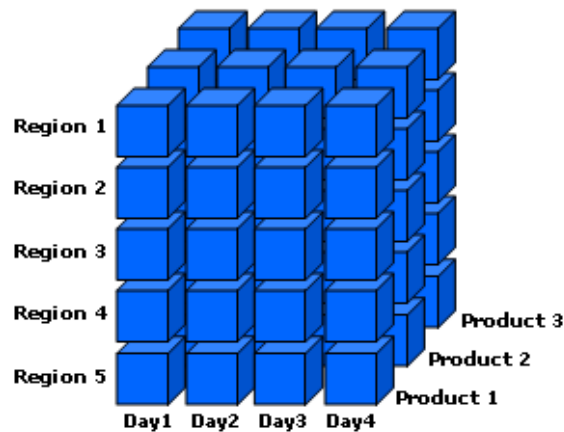


Рис. 1.4 – OLAP-куб

Представление данных в виде кубов позволяют совершать над ними ряд операций:

- срез (Slice) – выбор всех данных таблиц фактов, соответствующих определенному значению какого-либо измерения. В сущности, данная операция представляет собой получение n -мерной проекции $n+1$ -мерного куба;
- вращение (Rotate) – изменение измерений, представленных в отчете, например, замена столбцов таблицы строками или замена одного измерения другим, не содержащимся в отчете;
- консолидация и детализация (Drill Up и Drill Down) – переход от детального представления данных к агрегированному и наоборот.

Несмотря на то, что системы OLAP имеют собственные средства создания отчетов, их возможности сильно ограничены. Для эффективной визуализации необходимо использовать сторонние инструменты, например, аналитические панели индикаторов.

1.3. Обоснование необходимости и цели использования вычислительной техники

При принятии управленческого решения необходимо произвести анализ большого массива информации, качество которого зависит во многом от точности расчетов, объема, используемых алгоритмов и т.д. Понятно, что без

использования вычислительной техники это произвести практически невозможно, а если эти решения принимались на основе расчетов, то периодичность таких расчетов была довольно низкой, так как полный расчет занимал большое количество времени. Использование ВТ существенно облегчило вычислительный процесс.

Рассмотрим несколько вариантов ПЭВМ.

1.КомпьютерНИКСС5100а

ПЭВМС5100а (C5335LNa): A8 7650K / 8 Гб / 1 Тб / RADEONR7 / DVDRW / Win10 HomeAMDPROA-SeriesA8 APUforDesktops|8 ГбDDR3 1866 МГцRAM |1 Тб |Windows 10 Домашняя|23.6" ЖКмониторASUSVS247NRBK (LCD, Wide, 1920x1080, D-Sub, DVI)|SamsungSL-M2020 (A4, лазерный, 20 стр / мин, 8Мб, USB2.0)

2.КомпьютерHP 290 G1 Microtower

ПЭВМHP 290 G1 Microtower< 2RT88ES#ACB>i3 7100 / 4 / 1Тб / DVD-RW / DOS /IntelCorei3 7100/4 ГБРАМ /1 ТбHDD /ВстроенныйDVD-RW /IntelHDGraphics 630 /DOS/23.6" ЖКмониторASUSVS247NRBK (LCD, Wide, 1920x1080, D-Sub, DVI)/ПринтерлазерныйHPLaserJetProM104aRU (G3Q36A) A4

3.Компьютер HP 290 G1 Microtower + V214a Monitor

ПЭВМ HP 290 G1 Microtower + V214a Monitor < 2TP49ES#ACB > i3 7100 / 4 / 1Тб / DVD-RW / DOS / 20.7"/Intel Core i3 7100 /4 ГБ RAM /1 ТбHDD /Встроенный DVD-RW /Intel HD Graphics 630 /DOS/23.6" ЖКмонитор ASUS VS247NR BK (LCD, Wide, 1920x1080, D-Sub, DVI/HP LaserJet Ultra M106w (A4, 22стр/мин, 128Мб, USB2.0)

Изучив характеристики ВТ остановим свой выбор на следующей комплектации:

- ПроцессорIntel Core i3.
- Материнская плата 2TP49ES#ACB+ Видео 4МБ +Сетевая карта..
- Оперативная память 4.0ГБ.
- Жесткий диск (винчестер) 500ТБайт.

- 23.6" ЖК монитор ASUSVS247NRBK (LCD, Wide, 1920x1080).
- Операционная система WINDOWS10.
- Принтер – Принтер лазерный HPLaserJetProM104.

1.4.Общая характеристика организации машинной обработки

В условиях всеобщей компьютеризации появилась возможность минимизировать затраты на выполнение комплекса поставленных задач. В первую очередь отпадает необходимость в выполнении рутинных ручных подсчётов с использованием калькуляторов, при которых сохранялась большая вероятность ошибок. Неизменны только лишь функции, связанные со сбором первичной входной информации.

Входной документ – это описатель некоторых файлов, условий, требований, количественных и качественных параметров [3]. Эти документы используется в качестве входных компонентов для других операций. В данном дипломном проекте по разработке OLAP системы входными документами являются данные торговых операций, данные о сотрудниках, о товарах, поставщиках и покупателях.

Каждый входной документ имеет свою форму документа для ввода данных, модуль документа, определяющий действия производимые документом в информационной базе, модуль формы документа, который определяет порядок ввода данных в входной документ и процедуру обработки данных.

Цель создания информационной системы - обеспечение организации достоверной, своевременной и достаточной экономической информацией для принятия решений по составлению аналитического заключения. Для того чтобы снабжать аналитический уровень необходимой информацией, создаются информационные хранилища, которые предназначены для объединения, как различных текущих операционных данных, так и данных других типов.

Практическая реализация проекта построения информационного хранилища влечет за собой проблемы:

1. Необходимость серьезной программно-аппаратной платформы;
2. Зависимость от взаимопонимания на всех уровнях предприятия;
3. Отсутствие методики технологии создания базы метаданных и преодоление психологического барьера.

Для ввода, хранения и обработки данных при организации внутримашинной обработки данных необходима база данных. В процессе выполнения внутримашинной обработки данных необходимо в явном виде описывать все информационные образования:

реквизиты,
показатели,
базы данных,
назначение,
связи и способы представления.

Программа — это некоторое проектное решение (алгоритм) по реализации заданных функций и процедур по обработке данных, записанное в виде функциональных спецификаций, схем алгоритма, алгоритма на машинных кодах. В процессе создания внутримашинной обработки данных программы как объекты разработки могут иметь различные состояния, и следовательно различные формы документального отображения этих состояний.

В основу конструирования АРМ положены следующие основные принципы:

1. Максимальная ориентация на конечного пользователя, достигаемая созданием инструментальных средств адаптации АРМ к уровню подготовки пользователя, возможностей его обучения и самообучения.
2. Формализация профессиональных знаний, то есть возможность предоставления с помощью АРМ самостоятельно автоматизировать новые функции и решать новые задачи в процессе накопления опыта работы с си-

стемой.

3. Проблемная ориентация АРМ на решение определенного класса задач, объединенных общей технологией обработки информации, единством режимов работы и эксплуатации.

4. Модульность построения, обеспечивающая сопряжение АРМ с другими элементами системы обработки информации, а также модификацию и наращивание возможностей АРМ без прерывания его функционирования.

5. Эргономичность, то есть создание для пользователя комфортных условий труда и дружественного интерфейса общения с системой

1.5. Формализация расчетов

В разрабатываемой автоматизированной системе производятся разные аналитические операции по группировке и подсчету сумм по группам, выбранных записей. Расчеты выполняются непосредственно специальными элементами OLAP-систем. Хотя данные расчеты не требуют программирования и соответственно формализации, представим некоторые положения.

Во-первых, представим скрипт запроса формирующего набор данных для последующего анализа.

```
Select o.SaleDate, p.ProdName, c.CustName, e.Fio,  
count(o.SaleDate), Sum (SumSale), Sum(Quantaty)  
From Orders AS o  
INNER JOIN Product AS p ON (o.ProductNo = p.ProductNo)  
INNER JOIN Customer AS c ON (o.CustNo = c.CustNo)  
INNER JOIN Employee AS e ON (o.EmpNo = e.EmpNo)  
GROUP BY o.SaleDate, p.ProdName, c.CustName, e.Fio
```

Данный запрос возвращает совокупные характеристики с группировкой по дате продажи, названию товара, поставщиках и сотрудниках. Поля

группировки определяют измерения гиперкуба. Результирующие данные определяются по следующим формулам:

$$K_j = \sum_{i=1}^m n_{i,j} \quad (1.1)$$

где n_i – количество товаров в i - торговой операции (продажи), j - наименования; K_j = количество проданных товаров j - наименования.

$$SK_j = \sum_{i=1}^m S n_{i,j} \quad (1.2)$$

где $S n_i$ – стоимость товаров в i - торговой операции (продажи) , j - наименования; SK_j = сумма стоимостей проданных товаров j - наименования.

1.6. Обоснование проектных решений по информационному обеспечению комплекс задач

Информационное обеспечение информационных систем (ИС) - это совокупность единой системы классификации и кодирования технико-экономической информации, унифицированных систем документации и массивов информации, используемых в информационных системах. Сущность информационного обеспечения ИС состоит в информационном отображении условий, состояния и результатов производственного процесса и обмене информацией между органом и объектом управления для регулирования его деятельности.

Основные идеи современной информационной технологии базируются на концепции баз данных (БД). Согласно данной концепции основой информационной технологии являются данные, организованные в БД, адекватно отражающие реалии действительности в той или иной предметной области и обеспечивающие пользователя актуальной информацией в соответствующей предметной области. Обычно различают три класса СУБД, обеспечивающих работу иерархических, сетевых и реляционных моделей. Однако различия между этими классами постепенно стираются, причем, видимо, будут появляться другие классы, что вызывается, прежде всего интенсивными работами в области баз знаний (БЗ) и объектно-

ориентированной инфотехнологией. Поэтому традиционной классификацией пользуются все реже, но мы пока будем придерживаться именно ее, как наиболее устоявшуюся. Каждая из указанных моделей обладает характеристиками, делающими ее наиболее удобной для конкретных приложений.

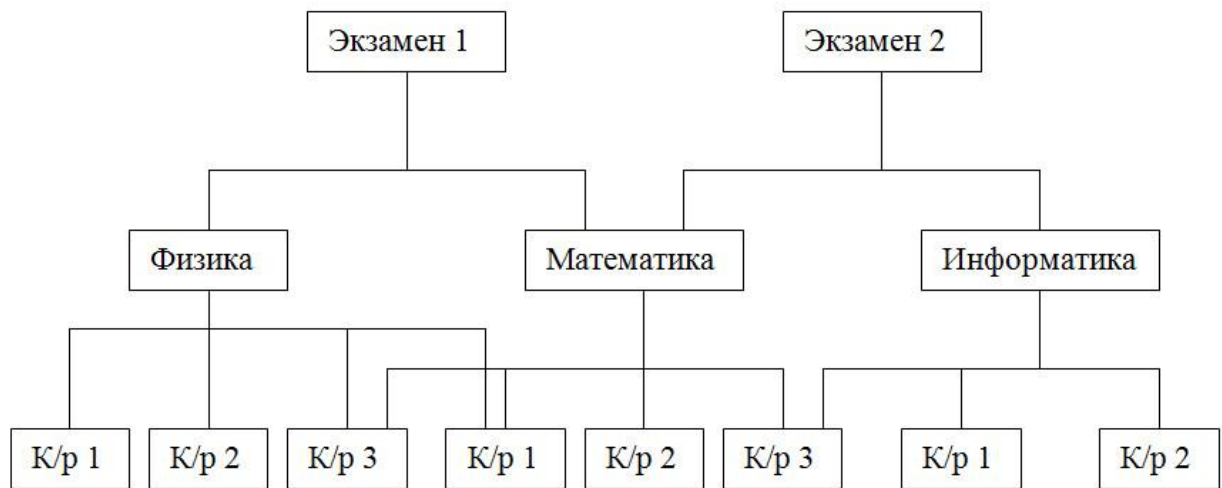
«Иерархические структуры» подробнее описывает положительные и отрицательные черты иерархической модели. Окружающий мир переполнен иерархическими данными. Любая группа объектов, в которой один объект может быть «родителем» для произвольного числа других объектов, организована в виде иерархического дерева. При работе с иерархиями используется «семейная» терминология (родители, внуки, предки, потомки), поскольку семья является самым распространённым примером объектов (в данном случае – людей), объединённых иерархическими отношениями. В то же время место объекта в иерархическом дереве - не более чем условное обозначение связи с другими объектами. Иерархическая структура всего лишь помогает сохранить и найти данные.

Как в иерархических структурах, в сетевых при описании данных обычно указываются характеристики записей каждого типа, способствующие более эффективному размещению данных во внешней памяти и более быстрому доступу к ним. К таким характеристикам относятся: размеры полей записи (минимальные, средние, максимальные), состав ключа, допустимый набор символов, интервалы значений и т.д. Иерархические и сетевые базы данных часто называют базами данных с навигацией. Это название отражает технологию доступа к данным, используемую при написании обрабатывающих программ на языке манипулирования данными. При этом, очевидно, что доступ к данным по путям, не предусмотренным при создании базы данных, может потребовать неразумно большого времени. Повышая эффективность доступа к данным, и сокращая таким образом, время ответа на запрос, принцип навигации вместе с этим повышает и степень зависимости программ и данных. Обрабатывающие программы оказываются

жестко привязанными к текущему состоянию структуры базы данных и должны быть переписаны при ее изменениях. Операции модификации и удаления данных требует переустановки указателей, а манипулирование данными остается записеориентированным. Кроме того, принцип навигации не позволяет существенно повышать уровень языка манипулирования данными, чтобы сделать его доступным пользователю-непрограммисту, или даже программисту-непрофессионалу. Для поиска записи-цели в иерархической или сетевой структуре программист должен вначале определить путь доступа, а затем просмотреть все записи, лежащие на этом пути, - шаг за шагом. Насколько запутанной являются схемы представления иерархических и сетевых баз данных, настолько и трудоемким является проектирование конкретных прикладных систем на их основе. Как показывает, опыт длительные сроки разработки прикладных систем нередко приводят к тому, что они постоянно находятся в стадии разработки и доработки. Указанные и некоторые другие проблемы, с которыми столкнулись разработчики и пользователи иерархических и сетевых систем послужили стимулом к созданию реляционной структуры данных и реляционных СУБД.



а)



б)

Рис. 1.5. Структура иерархической (а) и сетевой (б) СУБД

Реляционная технология значительно упрощает процесс проектирования баз данных. Разделение логического и физического уровней данных упрощает процесс отображения "уровня реального мира", в структуре данных, которую в дальнейшем нужно поддерживать. Поскольку реляционная структура сама по себе концептуально проста, она позволяет реализовывать небольшие и/или простые (и поэтому легкие для создания) базы данных, такие как персональные, сама возможность реализации которых никогда даже бы не рассматривалась в старых более сложных системах. Реляционная структура данных особенно удобна для использования в базах данных распределенной архитектуры - она позволяет получать доступ к любым информационным элементам, хранящимся в узлах сети ЭВМ. Необходимо обратить особое внимание на высокоуровневый аспект реляционного подхода, который состоит в множественной обработке записей. Благодаря этому значительно возрастает потенциал реляционного подхода, который не может быть достигнут при обработке по одной записи, и прежде всего это касается оптимизации.

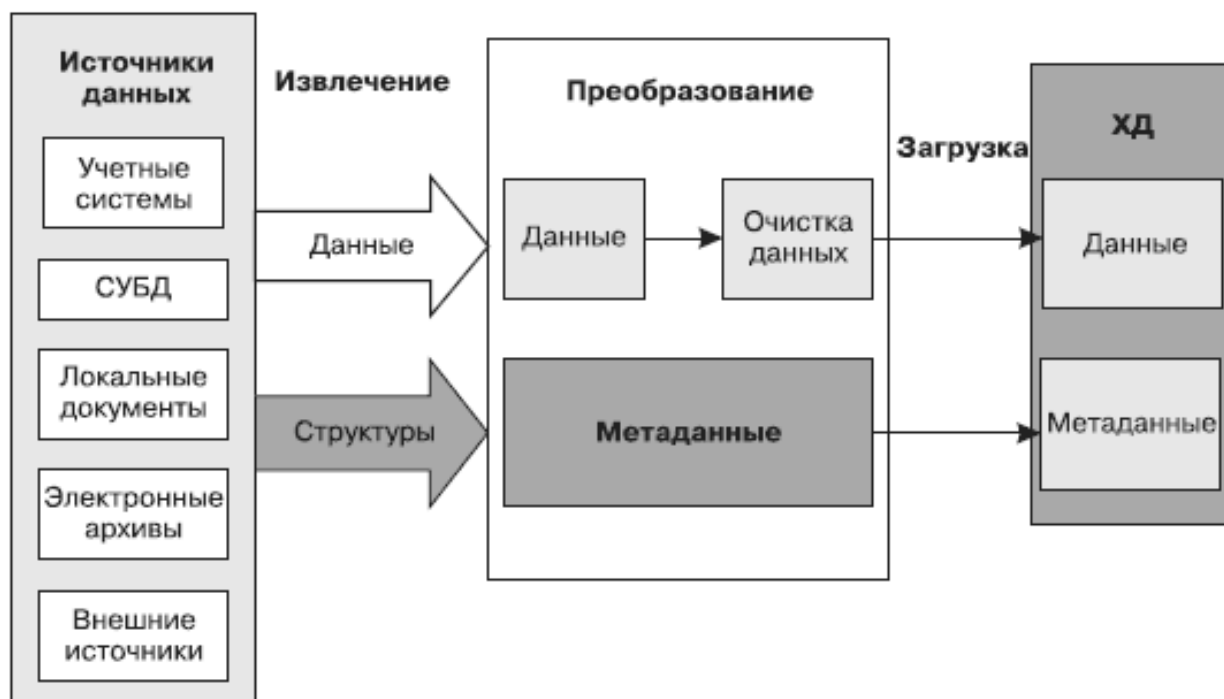


Рис.1.6. Структура реляционного хранилища данных.

Структура реляционного хранилища данных реляционного типа являются наиболее распространенным на всех классах ЭВМ, а на персональных компьютерах занимают доминирующее положение. Данная модель позволяет определять:

- операции по запоминанию и поиску данных;
- ограничения, связанные с обеспечением целостности данных.

В настоящее время существует целый ряд реляционных хранилищ данных, в той или иной мере отвечающих данному определению. Таким образом, при проектировании хранилища данных для предприятия используем реляционную структуру.

1.7. Обоснование проектных решений по программному обеспечению комплекса задач

Разработка хранилищ данных может выполняться с помощью специальных средств разработки, а также с помощью средств СУБД и систем программирования. Выбор во многом зависит от размеров хранилища данных, распределении данных БД и предпочтения разработчика. В общем средства

разработки хранилищ данных условно можно разделить на клиентские и серверные OLAP-средства.

Клиентские OLAP-средства представляют собой приложения, осуществляющие вычисление агрегатных данных (сумм, средних величин, максимальных или минимальных значений) и их отображение, при этом сами агрегатные данные содержатся в кэше внутри адресного пространства такого OLAP-средства.

Преимущества применения серверных OLAP-средств по сравнению с клиентскими OLAP-средствами сходны с преимуществами применения серверных СУБД по сравнению с настольными: в случае применения серверных средств вычисление и хранение агрегатных данных происходят на сервере, а клиентское приложение получает лишь результаты запросов к ним, что позволяет в общем случае снизить сетевой трафик, время выполнения запросов и требования к ресурсам, потребляемым клиентским приложением. Отметим, что средства анализа и обработки данных масштаба предприятия, как правило, базируются именно на серверных OLAP-средствах, например, таких как Oracle Express Server, Microsoft SQL Server 2000 Analysis Services, Hyperion Essbase, продуктах компаний Crystal Decisions, BusinessObjects, Cognos, SAS Institute. Поскольку все ведущие производители серверных СУБД производят (либо лицензировали у других компаний) те или иные серверные OLAP-средства, выбор их достаточно широк и почти во всех случаях можно приобрести OLAP-сервер того же производителя, что и у самого сервера баз данных.

Отметим, что многие клиентские OLAP-средства (в частности, Microsoft Excel 2000, Seagate Analysis и др.) позволяют обращаться к серверным OLAP-хранилищам, выступая в этом случае в роли клиентских приложений, выполняющих подобные запросы. Помимо этого имеется немало продуктов, представляющих собой клиентские приложения к OLAP-средствам различных производителей.

В качестве примера серверного OLAP-средства мы рассмотрим аналитические службы Microsoft (Microsoft Analysis Services), входящие в состав Microsoft SQL Server 2008 Enterprise Edition.

Пользователей аналитических служб можно условно разделить на две группы: администраторов, создающих или модифицирующих OLAP-кубы, и обычных пользователей-аналитиков, читающих данные из OLAP-кубов с целью создания аналитических отчетов. Эти группы используют разные технологии доступа к OLAP-данным. Ниже мы выясним, какие технологии предназначены для этих двух категорий пользователей.

SQL DSO

Decision Support Objects (DSO) — это набор библиотек, содержащих COM-объекты, позволяющие создавать и модифицировать многомерные базы данных и содержащиеся в них объекты (кубы, коллективные измерения и т.д.). Отметим, что Analysis Manager — приложение, использующее SQL DSO, — входит в состав аналитических служб.

Эти библиотеки можно использовать для разработки собственных приложений, в которых осуществляется создание или модификация многомерных баз данных, в том числе и для реализации действий, не предусмотренных в клиентских утилитах, входящих в состав аналитических служб (рис. 6).

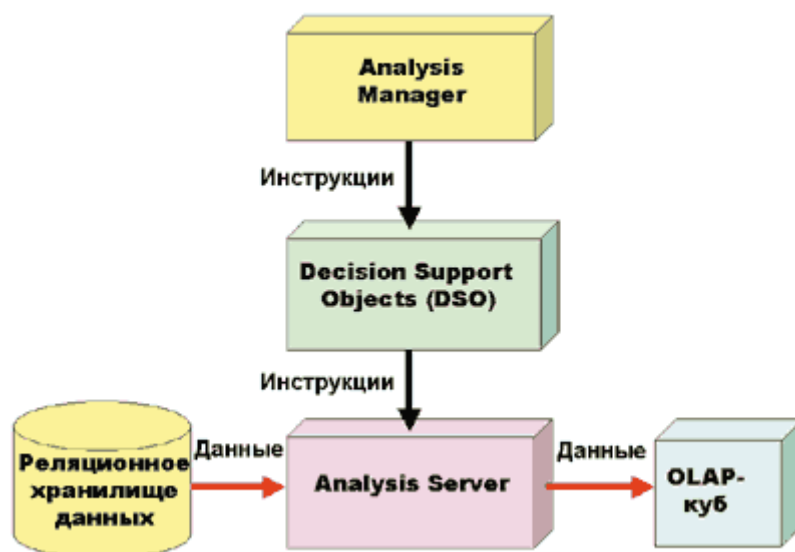


Рис. 1.7. Приложение, использующее SQL DSO

PivotTable Service можно использовать в любой 32-разрядной версии Windows для просмотра серверных OLAP-кубов, а также для создания, модификации и чтения локальных OLAP-кубов, созданных в клиентском приложении, реализуя таким образом клиентскую OLAP-функциональность. Эти библиотеки реализуют кэширование в клиентском приложении данных, полученных как с OLAP-сервера, так и из реляционных источников данных. Помимо этого они позволяют осуществлять кэширование данных и на OLAP-сервере, повышая тем самым производительность работы с ним в случае обращения к одним и тем же данным нескольких пользователей.

Analysis Manager представляет собой утилиту, входящую в состав аналитических служб и предназначенную главным образом для администраторов баз данных OLAP рис. 1.7. В составе Analysis Manager имеется простейшее средство просмотра многомерных данных, представляющее собой элемент управления ActiveX, использующий для доступа к данным OLE DB for OLAP.

Category Name	Product Name	Sales		
		2004	2005	2006
All Category Name	All Category Name Total	\$1,767,795.00	\$1,717,467.94	\$1,717,995.79
Beverages	Beverages Total	\$217,365.74	\$177,519.00	\$189,824.50
	Condiments Total	\$106,867.11	\$177,966.50	\$52,364.50
	Teas Total	\$1,099.00	\$179.00	\$1,724.50
	Chai Antiques Caramel Sauce	\$9,967.90	\$9,990.00	\$9,204.80
	Chai Antiques Lemon Miel	\$9,967.90	\$9,990.00	\$9,794.60
	Teas Honey	\$1,099.00	\$1,000.00	\$1,479.00
	Beverages Non-Alcoholic %	\$1,099.00	\$179.00	\$1,724.50
	San Mateo	\$1,099.00	\$1,000.00	\$1,479.00
Condiments	Londoner Fine Old Pepper	\$1,000.00	\$1,479.00	\$1,274.10
	Londoner Hot Sauce (Dry)	\$1,000.00	\$1,000.00	\$1,000.00
	Northwoods Curry Sauce (S)	\$1,000.00	\$1,000.00	\$1,000.00
	Original Penicillin Sauce	\$1,000.00	\$1,000.00	\$1,000.00
	Tempo Sauce	\$1,000.00	\$1,000.00	\$1,000.00
	Tempo Sauce	\$1,000.00	\$1,000.00	\$1,000.00
Condiments	Condiments Total	\$1,000.00	\$1,000.00	\$1,000.00
Dairy Products	Dairy Products Total	\$1,000.00	\$1,000.00	\$1,000.00
Wine/Beer	Wine/Beer Total	\$1,000.00	\$1,000.00	\$1,000.00
Hot/Spicy	Hot/Spicy Total	\$1,000.00	\$1,000.00	\$1,000.00
Produce	Produce Total	\$1,000.00	\$1,000.00	\$1,000.00

Рис. 1.8. Analysis Manager

Приложения Microsoft Office. Из других клиентских приложений, не входящих в состав аналитических служб, но часто используемых для просмотра OLAP-кубов, следует назвать приложения Microsoft Office, в частности Microsoft Excel. С помощью Excel можно обращаться к серверным OLAP-кубам, получая их двух- и трехмерные сечения на листах рабочих книг Excel в виде сводных таблиц, а также создавать локальные OLAP-кубы в виде файлов на основе реляционных данных, доступных с помощью OLE DB (рис. 1.9).

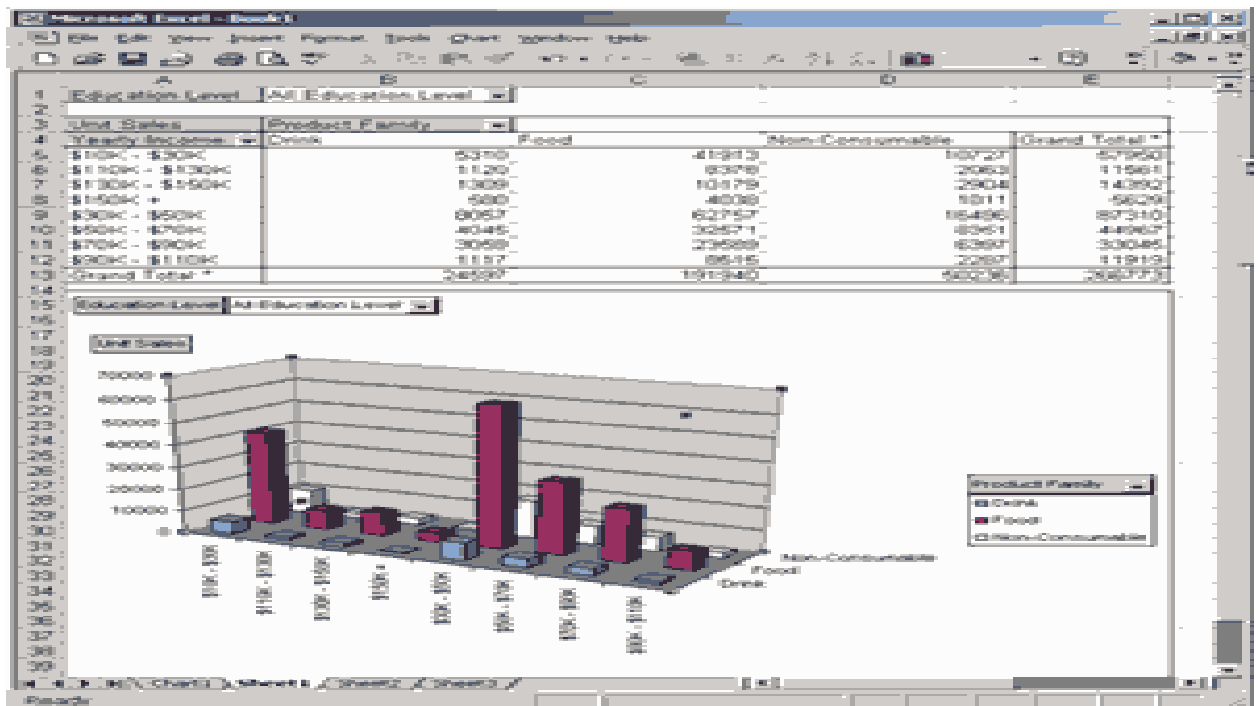


Рис. 1.9. Применение Excel в качестве OLAP-клиента

Кроме того, в состав Microsoft Office Web Components входит элемент управления ActiveX PivotTable List, позволяющий реализовать сходную функциональность как в обычном Windows-приложении, так и на HTML-странице, предназначенной для применения внутри корпоративной сети (рис. 9).

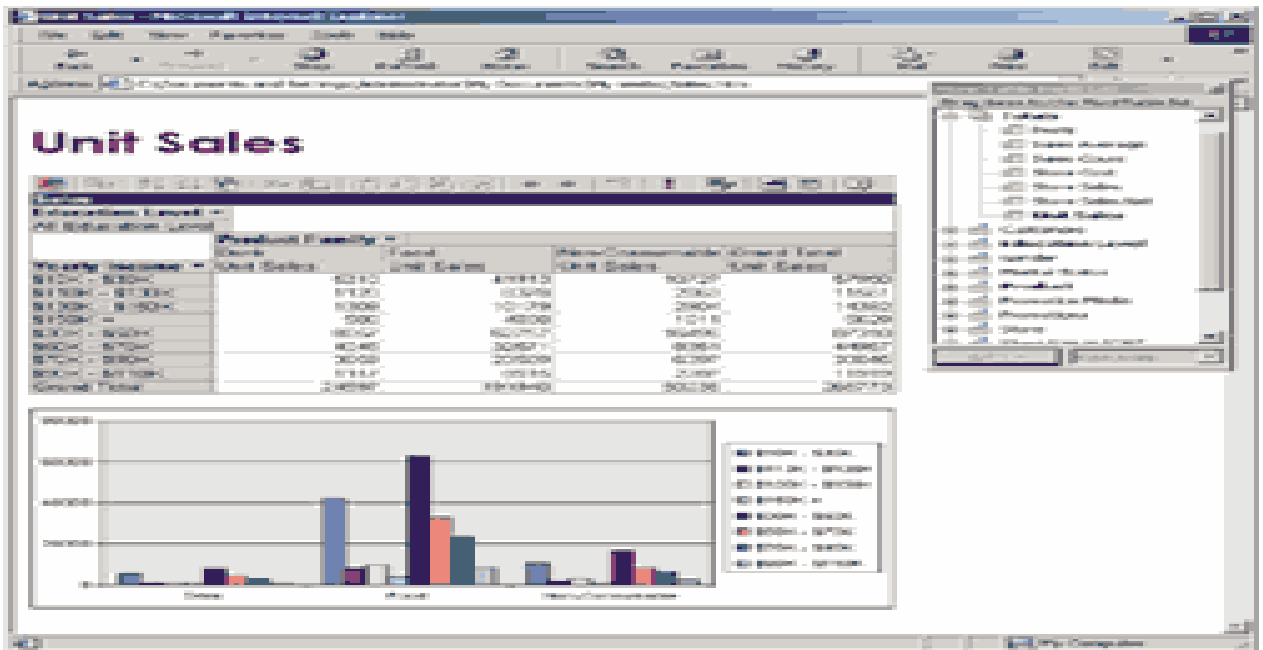


Рис. 1.10. Применение элемента управления PivotTable List в качестве OLAP-клиента

Еще одним инструментом разработки хранилища данных является система визуального объектно-ориентированного программирования Delphi XE. В Delphi XE имеется специальный набор OLAP компонентов, расположенных на вкладке Decision Cube (куб решений). Некоторые из этих компонентов похожи на компоненты на вкладках Data Access и Data Controls. Рассмотрим назначение OLAP компонентов.

TDecisionQuery – невизуальный компонент для организации SQL запросов к БД, на основании результатов которых строится гиперкуб. Аналог компонента TADOQuery. Вполне допустимо использовать и обычный компонент TADOQuery, однако текст запросов придется писать вручную, в то время как TDecisionQuery обладает специальным редактором для построения запросов.

TDecisionCube – невизуальный компонент для построения гиперкуба.

TDecisionSource – невизуальный компонент источника данных из гиперкуба. Аналог компонента TDataSource.

TDecisionPivot – визуальный компонент для выбора и настройки требуемого сечения гиперкуба.

TDecisionGrid – визуальный компонент для отображения сечения гиперкуба. Аналог компонента *TDBGrid*. *TDecisionGraph* – визуальный компонент для отображения диаграммы на основании данных из сечения гиперкуба. Аналог компонента *TDBChart*.

Microsoft Office Access или просто **Microsoft Access** — реляционная система управления базами данных (СУБД) корпорации Microsoft. Входит в состав пакета Microsoft Office. Имеет широкий спектр функций, включая связанные запросы, связь с внешними таблицами и базами данных. Благодаря встроенному языку VBA, в самом Access можно писать приложения, работающие с базами данных.

Так как Microsoft Access является современным приложением Windows, можно использовать в работе все возможности DDE (динамический обмен данными) и OLE (связь и внедрение объектов). DDE позволяет осуществлять обмен данными между Access и любым другим поддерживающим DDE приложением Windows. В Microsoft Access можно при помощи макросов или Access Basic осуществлять динамический обмен данными с другими приложениями.

OLE является более изощренным средством Windows, которое позволяет установить связь с объектами другого приложения или внедрить какие-либо объекты в базу данных Access. Такими объектами могут быть картинки, диаграммы, электронные таблицы или документы из других поддерживающих OLE приложений Windows.

В Microsoft Access для обработки данных базовых таблиц используется мощный язык SQL (структурированный язык запросов). Используя SQL можно выделить из одной или нескольких таблиц необходимую для решения конкретной задачи информацию. Access значительно упрощает задачу обработки данных. Совсем не обязательно знать язык SQL. При любой обработке данных из нескольких таблиц Access использует однажды заданные связи между таблицами.

1.8.Обоснование проектных решений по технологии сбора, передачи, обработки и выдачи информации

Для того, чтобы автоматизированная система функционировала эффективно, важно обеспечение взаимодействия персонала системы со средствами переработки информации. Взаимодействие осуществляется с помощью периферийного оборудования. Функция взаимодействия складывается из двух компонент:

- сбор (ввод) и регистрация первичной информации;
- выдача результатов переработки информации.

Эти компоненты совмещаются часто в одном техническом средстве.

Сущность сбора и регистрации, как этапа обработки данных заключается в определении и фиксации на машинных носителях количественных и качественных значений показателей, отражающих состояние объекта управления. Информация регистрируется либо одновременно, либо после операции сбора.

От полноты, достоверности и своевременности получаемой первичной информации зависит не только решение конкретной экономической задачи, но и эффективность управления в целом. Поэтому важнейшей задачей организации сбора и регистрации данных является наличие системы контроля для обеспечения полноты, правильности, комплектности и непротиворечивости данных[10].

Сбор и регистрация информации с помощью ЭВМ может быть:

- автоматизированной;
- автоматической.

Первичной информацией для данной автоматизированной системы являются данные о продажах товаров в торговой сети DNS, данные о сотрудниках, товарах, поставщиках и покупателях.

Автоматизированный способ формирования исходной информации предполагает, что операции сбора и регистрации будут выполняться с помо-

щью технических средств, позволяющих заносить данные на машинные носители.

Автоматический способ позволяет формировать исходные данные без участия человека.

В моей работе сбор информации осуществляется вводом данных о продажах в сети магазинов DNS с помощью технических средств, как с клавиатуры, так и с машинных носителей (флэш-карты, CD-диски).

Правильность хода расчета на ЭВМ контролируется или применением головной программы, вызывающей в необходимые моменты нужные массивы информации и подпрограммы, или оператором ЭВМ, который регистрирует выполнение отдельных этапов расчета по сообщениям машины.

Обработанные данные отображаются пользователю программой в виде экранных форм представленных в виде окон, которые при необходимости можно просмотреть на экране или распечатать на бумажный носитель.

Результатная информация выдается в виде отчетов, таблиц, заключения. Данные отчеты выдаются на экран дисплея пользователю. Также эти отчеты можно сохранить на внешние носители информации или распечатать на бумаге.

2. ПРОЕКТНАЯ ЧАСТЬ

2.1. Информационное обеспечение комплекса задач

2.1.1. Инфологическая или информационная модель (модель данных) и её описание

Основными направлениями принятия решений являются объемы продаж и выбор ассортимента торговли. Прежде чем вводить новые ассортиментные линии необходимо знать состояние дел на данный момент, это же немаловажно при разработке анализе работы компании.

При проектировании автоматизированной системы анализа продаж в крупной торговой компании необходимо разработать два вида информационного обеспечения:

- OLTP – online Transact Process;
- OLAP – Online Analysis Process.

Первый вид информационного обеспечения OLTP – онлайн-обработка транзакций, которая представляет собой реляционную базу данных, состоящую из совокупности взаимосвязанных таблиц баз данных. Как и любая база данных при ее проектировании должны быть выполнены определенные операции, которые можно сгруппировать в следующие этапы – построения моделей:

- этап инфологического проектирования;
- этап даталогического проектирования;
- этап физического проектирования [10].

Инфологическая модель представляет собой описание предметной области с использованием специальных языковых средств, формул, графов и т.д. В инфологической модели отображаются классы объектов и связи между ними.

Классом объектов называют совокупность объектов, обладающих одинаковым набором свойств.

Каждому объекту в классе объектов присваивается свое уникальное имя (идентификатор). Каждому классу объектов в модели присваивается уникальное имя. Связи между объектами и характеризующими его свойствами изображаются в виде линий, соединяющих обозначение объекта и его свойств.

Даталогическая модель базы данных является моделью логического уровня и представляет собой отображение логических связей между элементами данных безотносительно к их содержанию и среде хранения. Эта модель строится в терминах информационных единиц, допускаемых конкретной СУБД, в среде которой проектируем базу данных.

Физическая модель базы данных определяет используемые запоминающие устройства, способы хранения и организации данных. Инфологическая модель учета фактов продажи товаров в торговой сети DNS выглядит следующим образом (рис.2.1).

Информационное обеспечение OLAP- системы представляет собой избыточную базу данных, построенную на основе OLTP-системы, обеспечивающую приемлемую скорость обработки больших массивов данных. Для построения информационного обеспечения OLAP-системы создаются таблицы фактов, в данной работе это факт продажи товаров (FactSaling), и таблицы измерения:

- измерение времени;
- измерение сотрудники;
- измерение продуктов - товаров;
- измерение продавцов;
- измерение сотрудник;
- измерение поставщик;
- измерение документ продажи.

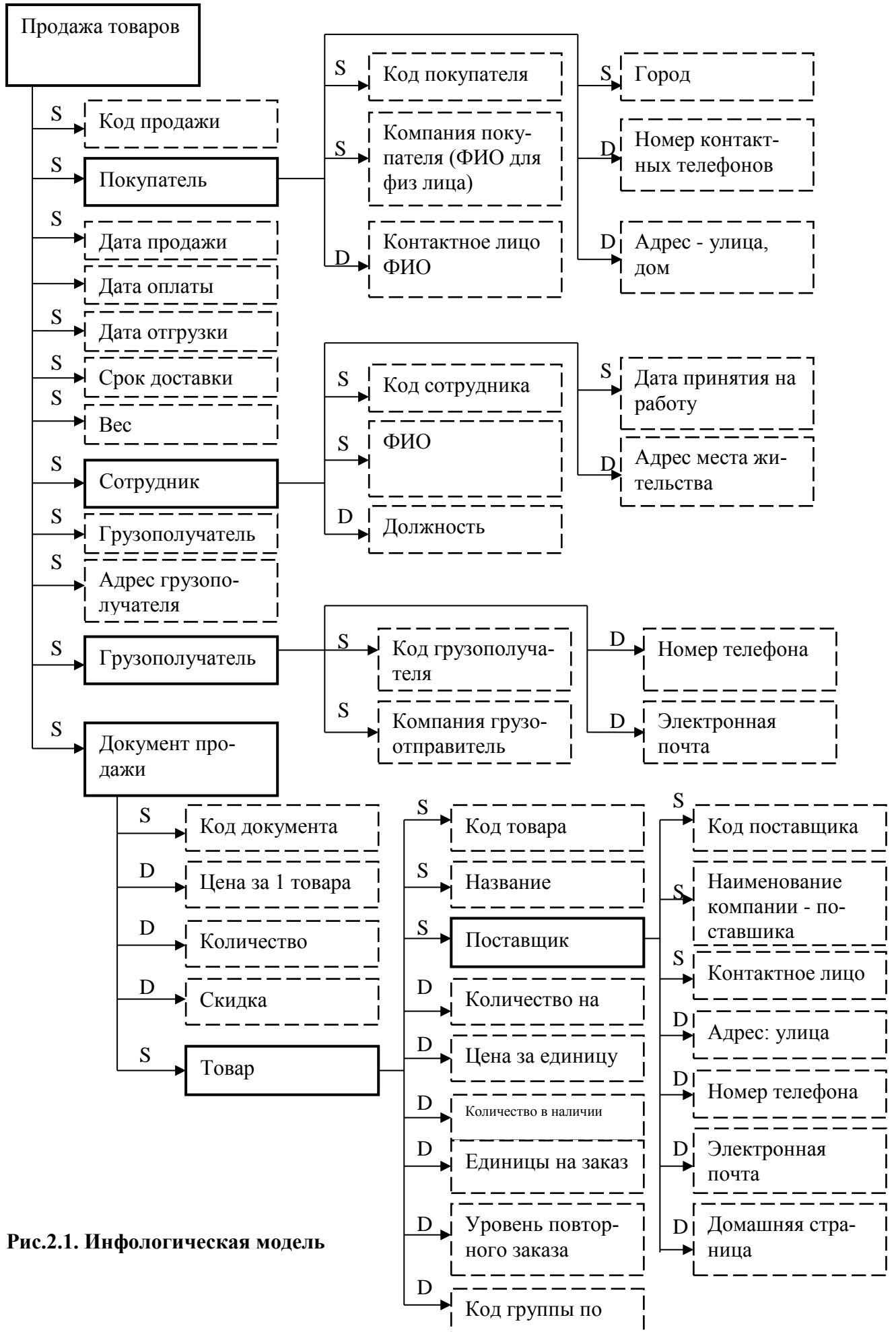


Рис.2.1. Инфологическая модель

2.1.2. Характеристика входной информации

2.1.2.1. Описание входной оперативной информации (входных документов и макетов размещения данных)

В соответствии с разработанной инфологической моделью создадим даталогическую модель, которая опишет данные в логических единицах используемой в дальнейшем СУБД. При переходе от инфологической модели к даталогической используем алгоритмы перехода. Согласно таким алгоритмам, предложенным С.М. Диго, каждому простому объекту соответствует отдельное отношение, в случае если у объекта нет множественных свойств. Так как это условие выполняется, то каждому объекту соотнесем отдельную таблицу.

Таблица 2.1

Таблица «Продажа»

№	Название	Тип данных	Размер	Ограничения	Назначение
1	OrderID	Счетчик		PrimaryKey	Код продажи
2	CustomerID	Числовой		Not NULL	Код покупателя
3	EmployeeID	Числовой		Not NULL	Код сотрудника
4	OrderDate	Дата/время			Дата продажи
5	RequiredDate	Дата/время			Дата оплаты
6	ShippedDate	Дата/время			Дата отгрузки
7	ShipVia	Числовой			Доставить не позднее чем количество дней
8	Freight	Числовой			Вес
9	ShipName	Текстовый	40		Грузополучатель
10	ShipAddress	Текстовый	100		Адрес грузополучателя
11	ShipCity	Текстовый	100		Город грузополучателя
12	ShipRegion	Текстовый	100		Регион грузополучателя
13	ShipperID	Числовой		Not NULL	Код грузополучателя

Таблица «Покупателя»

№	Название	Тип данных	Размер	Ограничения	Назначение
1	CustomersID	Счетчик		PrimaryKey	Код покупателя
2	CompanyName	Текстовый	100	Not NULL	Компания покупателя (ФИО для физ лица)
3	ContactName	Текстовый	100	Not NULL	Контактное лицо ФИО
4	ContactTitle	Текстовый	100		Должность контактного лица
5	Address	Текстовый	100		Адрес - улица, дом
6	City	Текстовый	100		Город
7	Region	Текстовый	100		Регион
8	PostalCode	Текстовый	6		Почтовый индекс
9	Country	Текстовый	100		Страна
10	Phone	Текстовый	15		Номер контактных телефонов
11	Email	Текстовый	50		Почтовый индекс

Таблица 2.3

Таблица «Поставщик»

№	Название	Тип данных	Размер	Ограничения	Назначение
1	SupplierID	Счетчик		PrimaryKey	Код поставщика
2	CompanyName	Текстовый	100	Not NULL	Наименование компании - поставщика (ФИО - физ. лица)
3	ContactName	Текстовый	100	Not NULL	Контактное лицо
4	ContactTitle	Текстовый	100		Должность контактного лица
5	Address	Текстовый	100		Адрес: улица, дом, квартира
6	City	Текстовый	100		Город
7	Region	Текстовый	100		Регион
8	PostalCode	Текстовый	100		Почтовый индекс
9	Country	Текстовый	100		Страна
10	Phone	Текстовый	100		Номер телефона
11	Email	Текстовый	100		Электронная почта
12	HomePage	Текстовый	100		Домашняя страница

Таблица 2.5.

Таблица «Поставщик»

№	Название	Тип данных	Размер	Ограничения	Назначение
1	ProductID	Счетчик		PrimaryKey	Код продукта - товара
2	ProductName	Текстовый	100	Not NULL	Название
3	SupplierID	Числовой		Not NULL	Код поставщика
4	CategoryID	Числовой		Not NULL	Код категории товара
5	QuantityPerUnit	Числовой		Not NULL	Количество на единицу
6	UnitPrice	Денежный		Not NULL	Цена за единицу
7	UnitsInStock	Числовой			Количество в наличии
8	UnitsOnOrder	Числовой			Единицы на заказ
9	ReorderLevel	Числовой			Уровень повторного заказа
10	Discontinued	Числовой			Скидки
11	KodPrice	Числовой			Код группы по ценам

Таблица 2.6.

Таблица «Документ о продаже»

№	Название	Тип данных	Размер	Ограничения	Назначение
1	OrderID	Счетчик		PrimaryKey	Код продажи товара
2	ProductID	Числовой		Not NULL	Код продукта
3	UnitPrice	Денежный		Not NULL	Цена продажи единицы продукта
4	Quantity	Числовой		Not NULL	Количество
5	Discount	Числовой			Скидка

Таблица 2.7.

Таблица «Грузоотправитель»

№	Название	Тип данных	Размер	Ограничения	Назначение
1	ShipperID	Счетчик		PrimaryKey	Код грузоотправителя
2	CompanyName	Текстовый	100	Not NULL	Компания грузоотправитель
3	Phone	Текстовый	15	Not NULL	Номер телефона
4	Email	Текстовый	60	Not NULL	Электронная почта

Таблица 2.8.

Таблица «Сотрудник»

№	Название	Тип данных	Размер	Ограничения	Назначение
1	EmployeeID	Счетчик		PrimaryKey	Код сотрудника
2	LastName	Текстовый	100	Not NULL	Фамилия сотрудника
3	FirstName	Текстовый	100	Not NULL	Имя сотрудника
4	Title	Текстовый	100	Not NULL	Должность
5	TitleOfCourtesy	Текстовый	8	Not NULL	Код должности
6	BirthDate	Дата/время		Not NULL	Дата рождения
7	HireDate	Дата/время			Дата принятия на работу
8	Address	Текстовый	100		Адрес места жительства
9	City	Текстовый	100		Город
10	Region	Текстовый	100		Регион

Таблица 2.9.

Таблица «Категория»

№	Название	Тип данных	Размер	Ограничения	Назначение
1	CategoryID	Счетчик		PrimaryKey	Код категории
2	CategoryName	Текстовый	100	Not NULL	Название категории товара-продукта
3	Description	Поле MEMO			Описание
4	Picture	Поле объекта OLE			Фото - или рисунок

Ниже представлена схема разработанной по инфологической модели OLTP системы, необходимой для полного функционирования учета торговых операций гипермаркета DNS.

2.1.2.2. Описание файлов базы данных

Реализуем разработанные структуры в MS Access (рис.2.2-2.9). Система состоит из 8 основных таблиц. Таблица Orders сохраняет данные продажи товара (код продажи, дата продажи, дата отгрузки, код товара, код поставщика, код менеджера, осуществившего продажу, код платежного документа, количество, стоимость продажи, скидка).

Имя поля	Тип данных	Описание
OrderID	Счетчик	Код продажи
CustomerID	Числовой	Код покупателя
EmployeeID	Числовой	Код сотрудника
OrderDate	Дата/время	Дата продажи
RequiredDate	Дата/время	Дата оплаты
ShippedDate	Дата/время	Дата отгрузки
ShipVia	Числовой	Доставить не позднее чем количество дней
Freight	Числовой	Вес
ShipName	Текстовый	Грузополучатель
ShipAddress	Текстовый	Адрес грузополучателя
ShipCity	Текстовый	Город грузополучателя
ShipRegion	Текстовый	Регион грузополучателя
ShipperID	Числовой	Код грузополучателя

Рис.2.2. Таблица Orders.

С этой таблицей связана таблица Customer, в которую записываются данные об покупателе. Если покупатель осуществил несколько покупок в гипермаркете, то при повторных покупках его данные уже будут в системе. Помимо прочего наличие данных о покупателях позволит определить насколько правильно работает система обслуживания клиентов (покупателей).

Имя поля	Тип данных	Описание
CustomersID	Счетчик	Код покупателя
CompanyName	Текстовый	Компания покупателя (ФИО для физ лица)
ContactName	Текстовый	Контактное лицо ФИО
ContactTitle	Текстовый	Должность контактного лица
Address	Текстовый	Адрес - улица, дом
City	Текстовый	Город
Region	Текстовый	Регион
PostalCode	Текстовый	Почтовый индекс
Country	Текстовый	Страна
Phone	Текстовый	Номер контактных телефонов
Email	Текстовый	Почтовый индекс

Рис.2.3. Таблица Customers.

Таблица Product является каталогом всех товаров, которыми торгует гипермаркет DNS, в ней хранится название товара, цена, код поставщика, а так же описание и характеристики товара. С данной таблицей связана таблица Supplier.

Имя поля	Тип данных	Описание
ProductID	Счетчик	Код продукта - товара
ProductName	Текстовый	Название
SupplierID	Числовой	Код поставщика
CategoryID	Числовой	Код категории товара
QuantityPerUnit	Числовой	Количество на единицу
UnitPrice	Денежный	Цена за единицу
UnitsInStock	Числовой	Количество в наличии
UnitsOnOrder	Числовой	Единицы на заказ
ReorderLevel	Числовой	Уровень повторного заказа
Discontinued	Числовой	Снятые
KodPrice	Числовой	Код группы по ценам

Рис.2.4. Таблица Products.

Она необходима для того, чтобы можно было группировать все товары по поставщикам, например, определять с каким поставщиком лучше работать. Таблица содержит основные данные о поставщике товаров: код, наименование, контактное лицо, адрес и т.д.

Имя поля	Тип данных	Описание
SupplierID	Счетчик	Код поставщика
CompanyName	Текстовый	Наименование компании - поставщика (ФИО - физ. лица)
ContactName	Текстовый	Контактное лицо
ContactTitle	Текстовый	Должность контактного лиуа
Address	Текстовый	Адрес: улица, дом, квартира
City	Текстовый	Город
Region	Текстовый	Регион
PostalCode	Текстовый	Почтовый индекс
Country	Текстовый	Страна
Phone	Текстовый	Номер телефона
Email	Текстовый	Электронная почта
HomePage	Текстовый	Домашняя страница

Рис.2.5. Таблица Suppliers

Таблица Document является таблицей, хранящей данные платежного документа: код, наименование, дата.

Имя поля	Тип данных	Описание
OrderID	Числовой	Код продажи товара
ProductID	Числовой	Код продукта
UnitPrice	Денежный	Цена продажи единицы продукта
Quantity	Числовой	Количество
Discount	Числовой	Скидка

Рис.2.6. Таблица OrderDetails

Таблица Employees служит для хранения данных сотрудников, работающих в торговых залах. Данная информация позволяет определить, как хорошо работает тот или иной сотрудник торгового предприятия DNS.

Имя поля	Тип данных	Описание
EmployeeID	Счетчик	Код сотрудника
LastName	Текстовый	Фамилия сотрудника
FirstName	Текстовый	Имя сотрудника
Title	Текстовый	Должность
TitleOfCourtesy	Текстовый	Код должности
BirthDate	Дата/время	Дата рождения
HireDate	Дата/время	Дата принятия на работу
Address	Текстовый	Адрес места жительства
City	Текстовый	Город
Region	Текстовый	Регион

Рис.2.7. Таблица Employees.

Таблица Shipper служит для хранения данных грузоотправителей, информация о которых заносится в таблицу продаж.

Имя поля	Тип данных	Описание
ShipperID	Счетчик	Код грузоотправителя
CompanyName	Текстовый	Компания грузоотправитель
Phone	Текстовый	Номер телефона
Email	Текстовый	Электронная почта

Рис.2.8. Таблица Shipper

Таблица Categories служит для хранения данных о категориях товаров в торговых залах.

Имя поля	Тип данных	Описание
CategoryID	Счетчик	Код категории
CategoryName	Текстовый	Название категории товара- продукта
Description	Поле МЕМО	Описание
Picture	Поле объекта OLE	Фото - или рисунок

Рис.2.9. Таблица Categories

Связи между таблицами базы данных

Система управления базами данных (СУБД) обычно поддерживает 4 основных типа отношений между таблицами:

- один-к-одному (одной записи в первой таблице соответствует одна запись во второй);
- один-ко-многим (одной записи в первой таблице соответствует много записей во второй);
- много-к-одному (многим записям в первой таблице соответствует одна запись во второй);
- много-ко-многим (одной записи в первой таблице соответствует много записей во второй и одной записи во второй таблице соответствует много записей в первой) [19].

Между таблицами созданы следующие связи:

-таблицей «Customers» и таблицей «Orders» по полю CustomerID (рис.2.10).

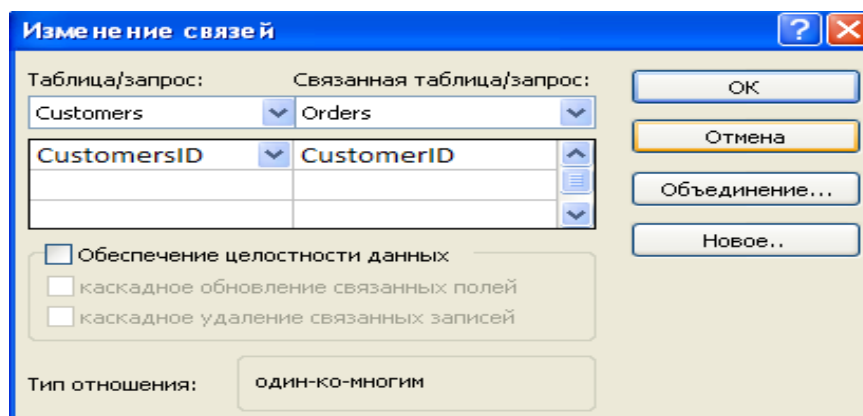


Рис.2.10. Окно связи таблиц Customers и Orders

-таблицей «Shipper» и таблицей «Orders» по полю KodKl (рис.2.11).

Изменение связей

Таблица/запрос: Shipper Связанная таблица/запрос: Orders

ShipperID	ShipperID

Обеспечение целостности данных
 каскадное обновление связанных полей
 каскадное удаление связанных записей

Тип отношения: один-ко-многим

OK, Отмена, Объединение..., Новое..

Рис.2.11. Окно связи таблиц Shipper и Orders

-таблицей «OrderDetails» и таблицей «Orders» по полю OrderID (рис.2.12).

Изменение связей

Таблица/запрос: Orders Связанная таблица/запрос: OrderDetails

OrderID	OrderID

Обеспечение целостности данных
 каскадное обновление связанных полей
 каскадное удаление связанных записей

Тип отношения: один-ко-многим

OK, Отмена, Объединение..., Новое..

Рис.2.12. Окно связи таблиц OrderDetails и Orders

-таблицей «Employees» и таблицей «Orders» по полю EmployeeID (рис.2.13).

Изменение связей

Таблица/запрос: Employees Связанная таблица/запрос: Orders

EmployeeID	EmployeeID

Обеспечение целостности данных
 каскадное обновление связанных полей
 каскадное удаление связанных записей

Тип отношения: один-ко-многим

OK, Отмена, Объединение..., Новое..

Рис.2.13. Окно связи таблиц Employees и Orders

-таблицей «Products» и таблицей «OrderDetails» по полю ProductID (рис.2.14).

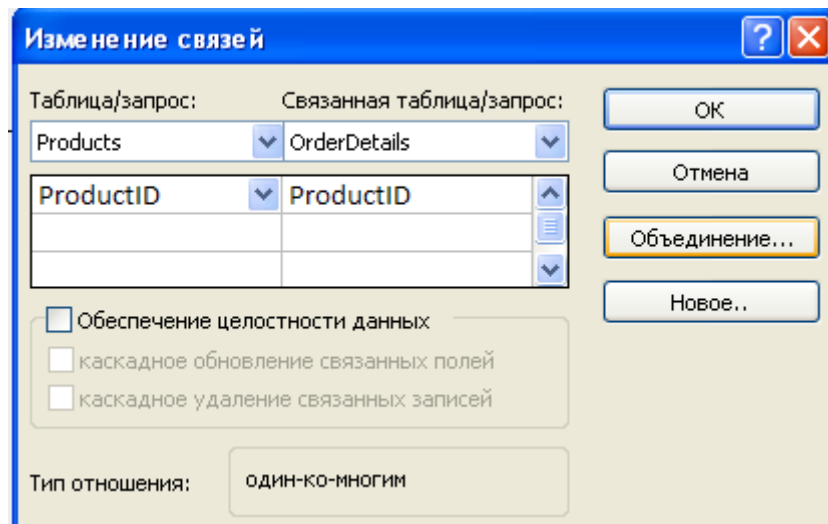


Рис.2.14. Окно связи таблиц Products и OrderDetails

-таблицей «Products» и таблицей «Suppliers» по полю SupplierID (рис.2.15).

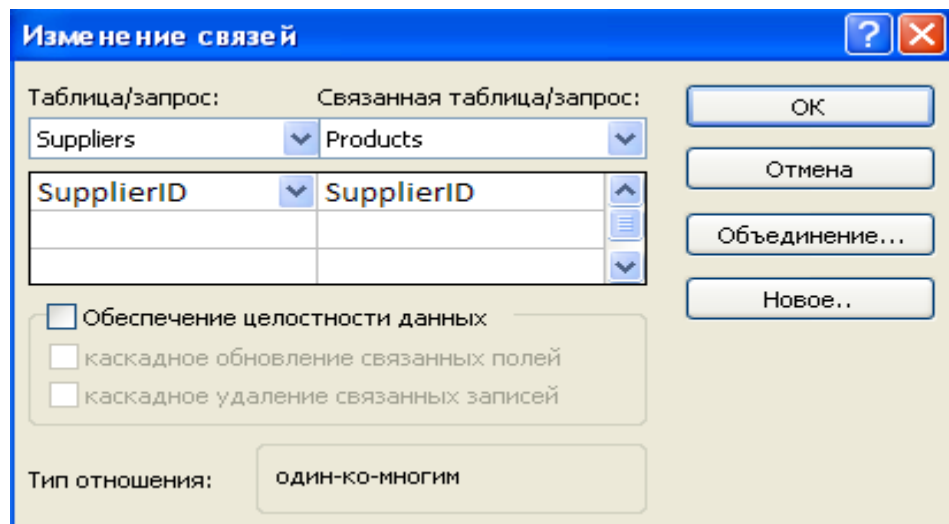


Рис.2.15. Окно связи таблиц Suppliers и Products

-таблицей «Categories» и таблицей «Products» по полю CategoryID (рис.2.16)

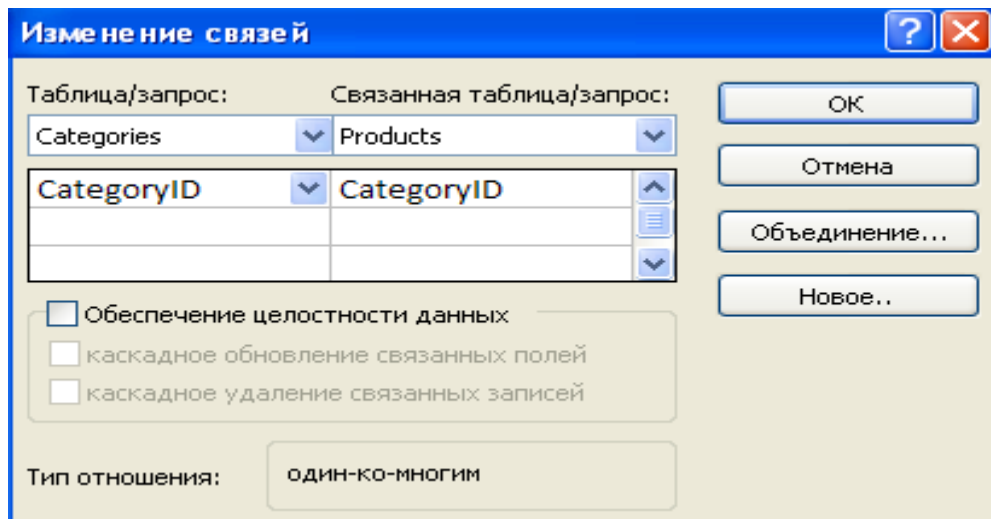


Рис.2.16. Окно связи таблиц Categories и Products

После установления связей между таблицами получим итоговую OLTP – систему (рис.2.17).

Подводя итоги, можно сказать, что данная схема позволяет производить все основные операции, необходимые для полноценного учета и анализа торговой деятельности гипермаркета DNS.

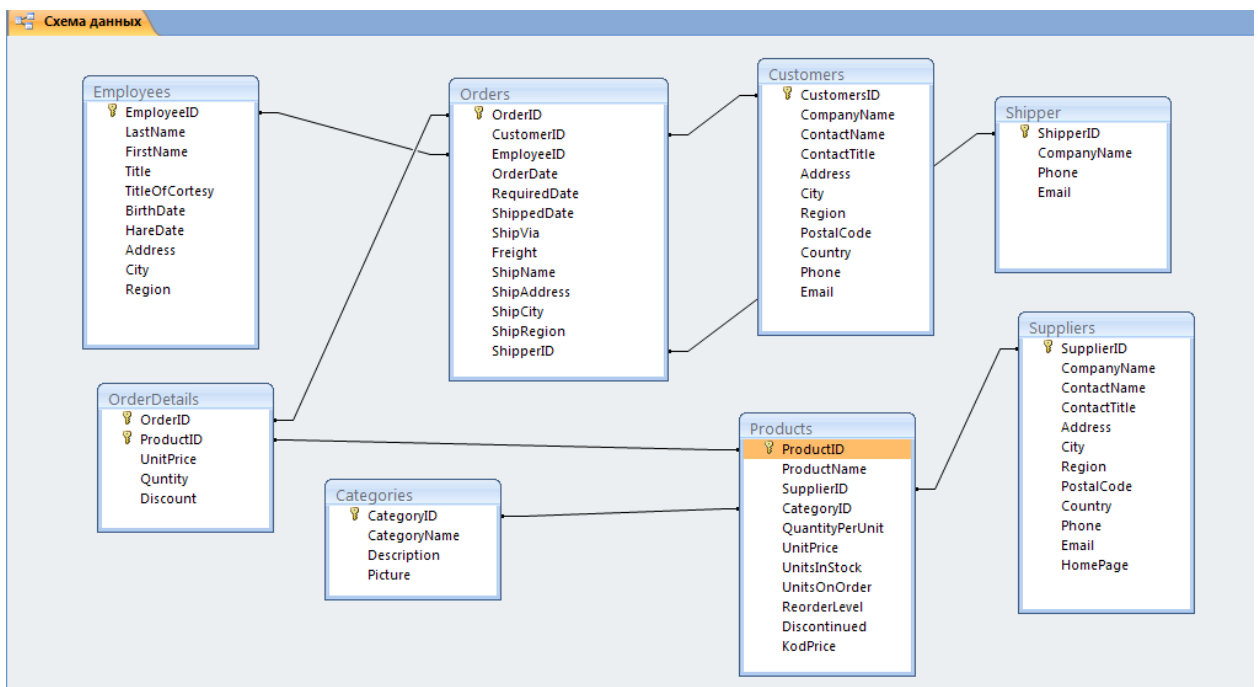


Рис.2.17. OLTP - система

2.1.3. Характеристика результатной информации

2.1.3.1. Макеты выходных документов

В качестве результатной информации разрабатываемый пакет программ должен представлять полную информацию об обслуживании клиента в торговой сети, выполнять рутинную работу помощи заполнения справочников, выдавать аналитические отчеты и диаграммы. Результатные данные выводятся как на экран для просмотра, так и на принтер для дальнейшего применения. Результатные данные, выводимые на экран компьютера, представляют собой экранные формы, а результатные данные, выводимые на принтер – макеты документов. Рассмотрим эти экранные формы и макеты документов.

К таким документам в данной системе относятся следующие документы:

- отчет «Анализ продаж в денежном выражении», позволяющий провести анализ продаж в гиперкубе по измерениям времени, товар, сотрудник;

Товар	Покупатель	Дата продажи				
		январь, 2020	февраль, 2020	март, 2020	апрель, 2020	май, 2020
LED телевизор Haier LE24K6	Вагабов	12 334,00р.				
	ООО Квантум			56 000,00р.		
	Sum	12 334,00р.		56 000,00р.		
LED телевизор Haier LE32K6	ООО КТР				45 660,00р.	
	ООО ЦАТ			88 800,00р.		
	Сидоров А.П.	34 588,00р.				
	Sum	34 588,00р.		88 800,00р.	45 660,00р.	
LED телевизор Haier LE40K6	ООО КТР	23 445,00р.				
	Sum	23 445,00р.				
LED телевизор Haier LE43K6	ООО Квантум					56 000,00р.
	Sum					56 000,00р.
LED телевизор Haier LE43K6	Петрова А.П.	45 500,00р.				
	Sum	45 500,00р.				
LED телевизор LG 32LK510B	Абдуллаев А.П.					82 999,00р.
	ООО Квантум		23 000,00р.			
	Sum		23 000,00р.			82 999,00р.
LED телевизор Samsung UE3	ООО Стар		54 000,00р.			
	Sum		54 000,00р.			
Ultra HD (4K) LED телевизор	ООО Техноснаб			45 000,00р.		

Рис.2.18. отчет «Анализ продаж в денежном выражении».

– отчет «Анализ продаж в количественном выражении», позволяющий провести анализ продаж в гиперкубе по измерениям времени, товар, сотрудник;

Кол-во продаж	Товар	Покупатель	Дата продажи				
			январь, 2020	февр, 2020	мар, 2020	апр, 2020	май, 2020
LED телевизор Haier LE24K6	Вагабов		1				
	ООО Квантум				1		
	Sum		1		1		
LED телевизор Haier LE32K6	ООО КТР					1	
	ООО ЦАТ				1		
	Сидоров А.П.		1				
	Sum		1		1	1	
LED телевизор Haier LE40K6	ООО КТР		1				
	Sum		1				
LED телевизор Haier LE43K6	ООО Квантум						1
	Sum						1
LED телевизор Haier LE43K6	Петрова А.П.		1				
	Sum		1				
LED телевизор LG 32LK510B	Абдуллаев А.П.						2
	ООО Квантум			1			
	Sum			1			2
LED телевизор Samsung UE3	ООО Стар			1			
	Sum			1			
Ultra HD (4K) LED телевизор	ООО Техноснаб				1		

Рис.2.19. отчет «Анализ продаж в количественном выражении».

– отчет «Анализ продаж по количеству проданных товаров», позволяющий провести анализ продаж в гиперкубе по измерениям времени, товар, сотрудник;

Кол. прод. товаров	Товар	Покупатель	Дата продажи				
			январь, 2020	февр, 2020	мар, 2020	апр, 2020	май, 2020
LED телевизор Haier LE24K6	Вагабов		1				
	ООО Квантум				1		
	Sum		1		1		
LED телевизор Haier LE32K6	ООО КТР					1	
	ООО ЦАТ				3		
	Сидоров А.П.		2				
	Sum		2		3	1	
LED телевизор Haier LE40K6	ООО КТР		1				
	Sum		1				
LED телевизор Haier LE43K6	ООО Квантум						1
	Sum						1
LED телевизор Haier LE43K6	Петрова А.П.		2				
	Sum		2				
LED телевизор LG 32LK510B	Абдуллаев А.П.						2
	ООО Квантум			2			
	Sum			2			2
LED телевизор Samsung UE3	ООО Стар			1			
	Sum			1			
Ultra HD (4K) LED телевизор	ООО Техноснаб				1		

Рис.2.20. отчет «Анализ продаж по проданным товарам».

– Диаграммы проданных товаров в разрезе куба – дата продажи, количество проданных товаров, а также по измерениям товар, покупатель или менеджер (рис.2.21 – 2.23).

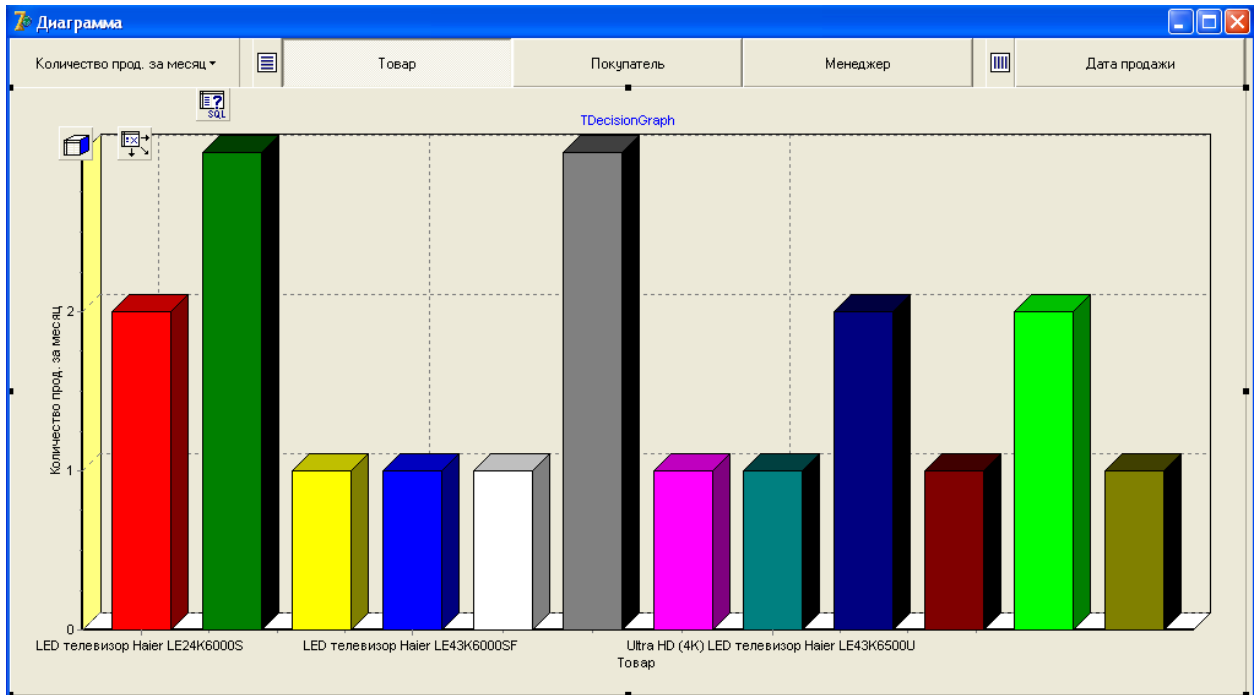


Рис.2.21. Диаграммы проданных товаров по количеству.

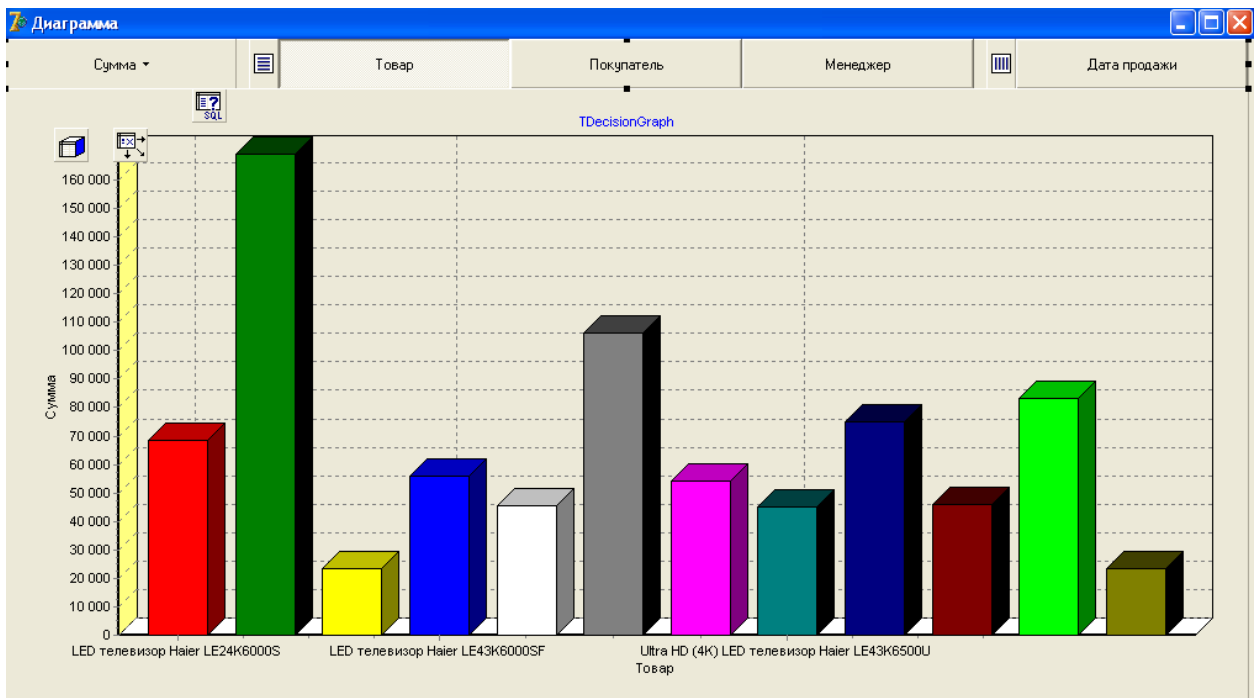


Рис.2.22. Диаграммы проданных товаров по суммам.

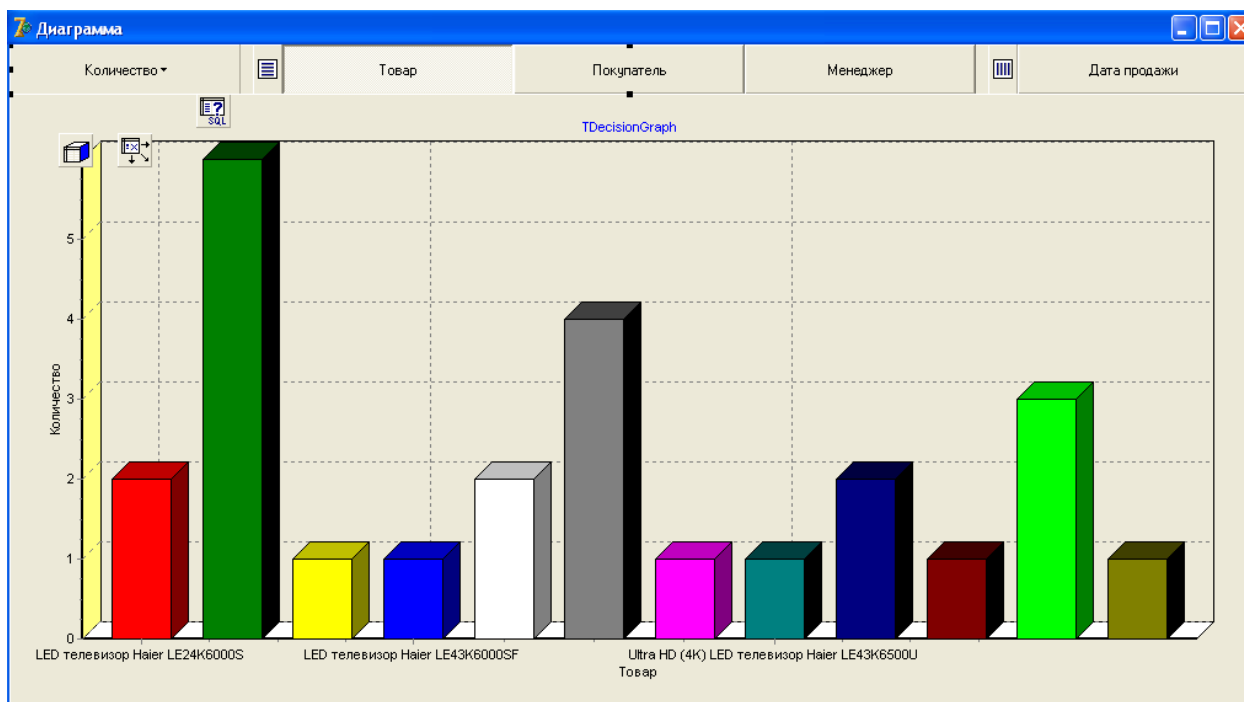


Рис.2.23. Диаграммы проданных товаров по количеству.

2.2. Машинная реализация

2.2.1. Схема взаимосвязи программных модулей и информационных файлов

Проектируемая OLAP – системы для торгового предприятия «ДНС», должна обеспечивать:

1. Просмотр, редактирование данных введённой информации.
2. Ввод информации о товарах, сотрудниках, поставщиках, документов о продажах, продажах в таблицы баз данных.
3. Формирование гиперкубов для проведения анализа данных.
4. Вывод выходной информации и распечатку отчетных форм.
5. Корректный выход из программы с закрытием всех объектов программного приложения

В данной дипломном проекте в качестве языка программирования выбран **Delphi XE**, позволяющий создавать профессиональные программные приложения. Программное приложение создается как проект, состоящий из некоторого числа приложений, каждое из которых состоит из процедур обрабатывающих некоторые события, которые происходят с компонентами приложения [17]. При работе с базами данных обязательными компонентами яв-

ляются компоненты с библиотеки компонентов Доступ к данным (**DataAccess**) – компонента таблиц, запросов и источников данных.

Центральным модулем программы является модуль-форма **Form1** с кодом, хранящимся в программе `unit1.pas`. Данный модуль предназначен для открытия таблиц баз данных (компоненты **TTable**, **TDataSource**), просмотра данных за прошедшие периоды (компоненты **TNavigator**, **TComboBox**, **TLabel**, **TEdit**), вызов других модулей (компоненты **TButton**, **TBitbtn**). Здесь же разместим световое меню расположенное в верхней строке формы (компонента **TMainMenu**). В нем разместим следующие пункты «Данные», «OLAP-кубы», «Справка», и пункт «Выход» (рис.2.24).

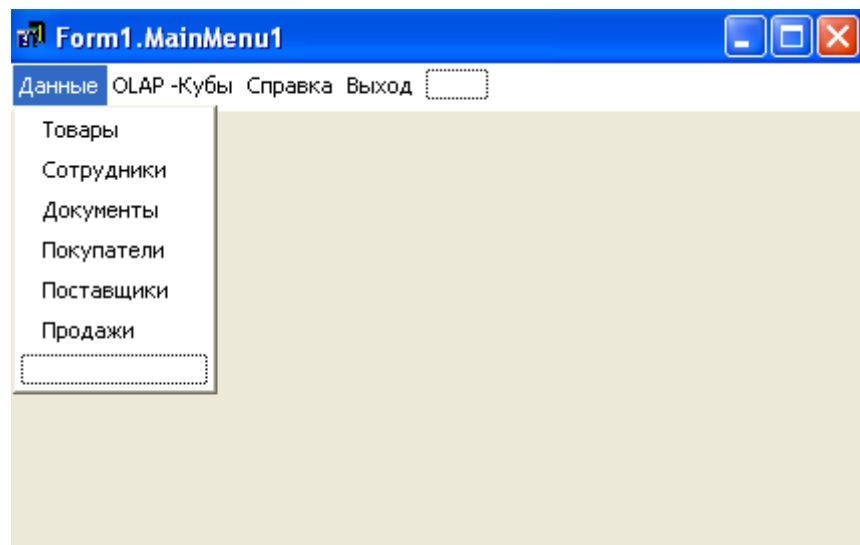


Рис.2.24. Проектирование главного меню информационной системы Main-Menu1

Данный модуль открывает стандартное окно в среде Windows. Становится доступным кнопки управления, список всех пользователей в табличной форме и атрибуты пользователей. Все остальные модули программы, блоки, подпрограммы вызываются при помощи управляющих кнопок, «Горячих клавиш» и ключей.

Программное приложение имеет следующие модули и информационные файлы:

- таблица базы данных «Категории товаров» Categories;

- таблица базы данных «Покупатели» Customers;
- таблица базы данных «Сотрудники» Employees;
- таблица базы данных «Детализация продажи» OrderDetails;
- таблица базы данных «Продажа» Orders;
- таблица базы данных «Продукты» Products;
- таблица базы данных «Грузоотправитель» Shipper;
- таблица базы данных «Поставщик» Suppliers.

Для проведения многомерного анализа данных продаж крупного торгового предприятия необходимо на основе разработанной OLTP – системы перейти к OLAP – системе.

Процесс проектирования любого хранилища (OLAP-системы), как уже было сказано, делится на следующие составляющие:

- Выбор бизнеса процесса
- Выбор таблицы фактов
- Выбор таблицы измерений
- Выбор количественных показателей

В основе хранилища данных лежит процесс торговли товарами. Элементарным событием этого процесса является непосредственно факт покупки –продажи товара (Fact_Saling). Можно было так же взять факт оплаты покупателем товара в качестве элементарного события, но с точки зрения процесса принятия решений факт покупки более интересен.

Измерениями являются таблицы с записями о покупателе, о дате покупки, о продавце (сотруднике торгового предприятия – менеджера зала) и о товаре. При этом данные поставщика товара будут записаны непосредственно в информацию о товаре, так как создание дополнительного измерения в данном случае избыточно.

Показателями будут количество продаж, количество товаров купленных покупателем, стоимость купленных товаров.

В результате проектирования получилась следующая схема хранилища данных (рис.2.25).

Данная схема позволяет строить все необходимые отчеты для стратегического принятия решений в сфере анализа торговой деятельности предприятия DNS

Вкупе с развитыми средствами для написания и отладки кода - специализированным текстовым редактором, оптимизирующим компилятором и отладчиком, Delphi предоставляет разработчикам средства создания OLAP - Cube. Эти средства в виде визуальных компонент сгруппированы на странице DecisionCube палитры библиотеки визуальных компонент.

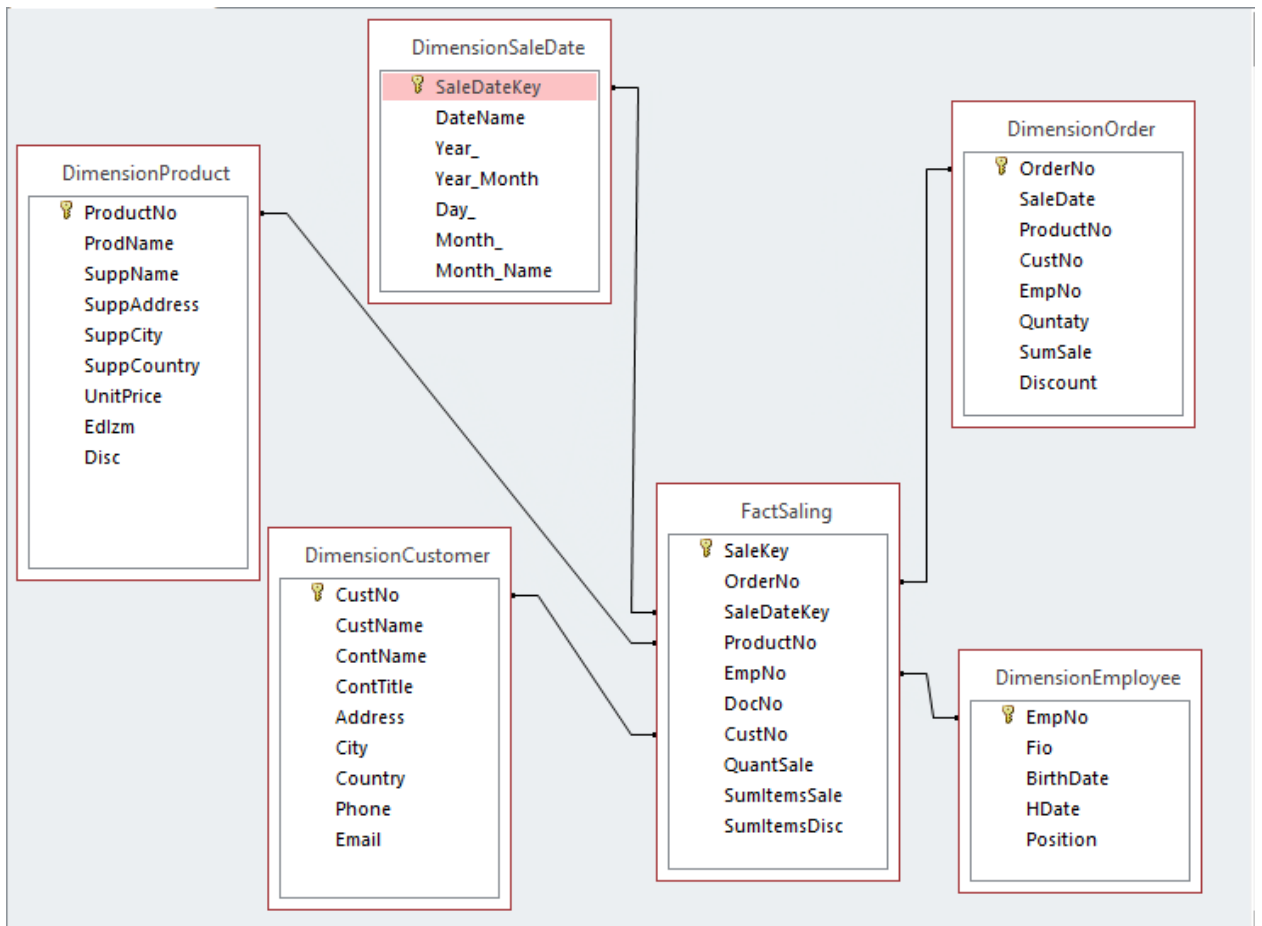


Рис. 2.25. Схема хранилища данных.

К таким компонентам относятся компоненты следующие:

- DecisionCube – компонент, реализующий многомерный куб;
- DecisionQuery - компонент, определяющий набор данных;

- DecisionSource – источник данных приспособленный к работе с многомерными кубами;
- DecisionPivot - компонент, дающий возможность управлять измерениями куба ;
- DecisionGrid - компонент, определяющий данные, соответствующие выбору в многомерном кубе;
- DecisionGraph - компонент, отображающий графики соответствующие выбору сделанному пользователем в многомерном кубе.

Используя указанные компоненты создадим программное приложение.

В автоматизированной системе разработаны и используются следующие модули:

- dns.dpr – программный модуль описания используемых в программном приложении объектов, форм, функций, переменных ;
- unit1.pas - программный модуль формирования главного окна системы (Form1);
- unit2.pas - программный модуль формирования справочника товаров (Form2);
- unit3.pas - программный модуль формирования справочника сотрудников (Form3);
- unit4.pas - программный модуль формирования справочника покупателей (Form4);
- unit5.pas - программный модуль формирования справочника документов (Form5);
- unit6.pas - программный модуль ввода и просмотра данных продаж товаров (Form6);
- unit7.pas - программный модуль справочника поставщиков (Form7);
- unit10.pas - программный модуль формирования OLAP –куба «Анализ продаж по измерениям» с выдачей информации в виде таблицы (Form10);
- unit11.pas - программный модуль формирования OLAP –куба «Диа-

граммы продаж по измерениям» с выдачей информации в виде диаграммы (Form11).

Таким образом, схема взаимосвязи программных модулей имеет вид представленный на рис. 2.26.

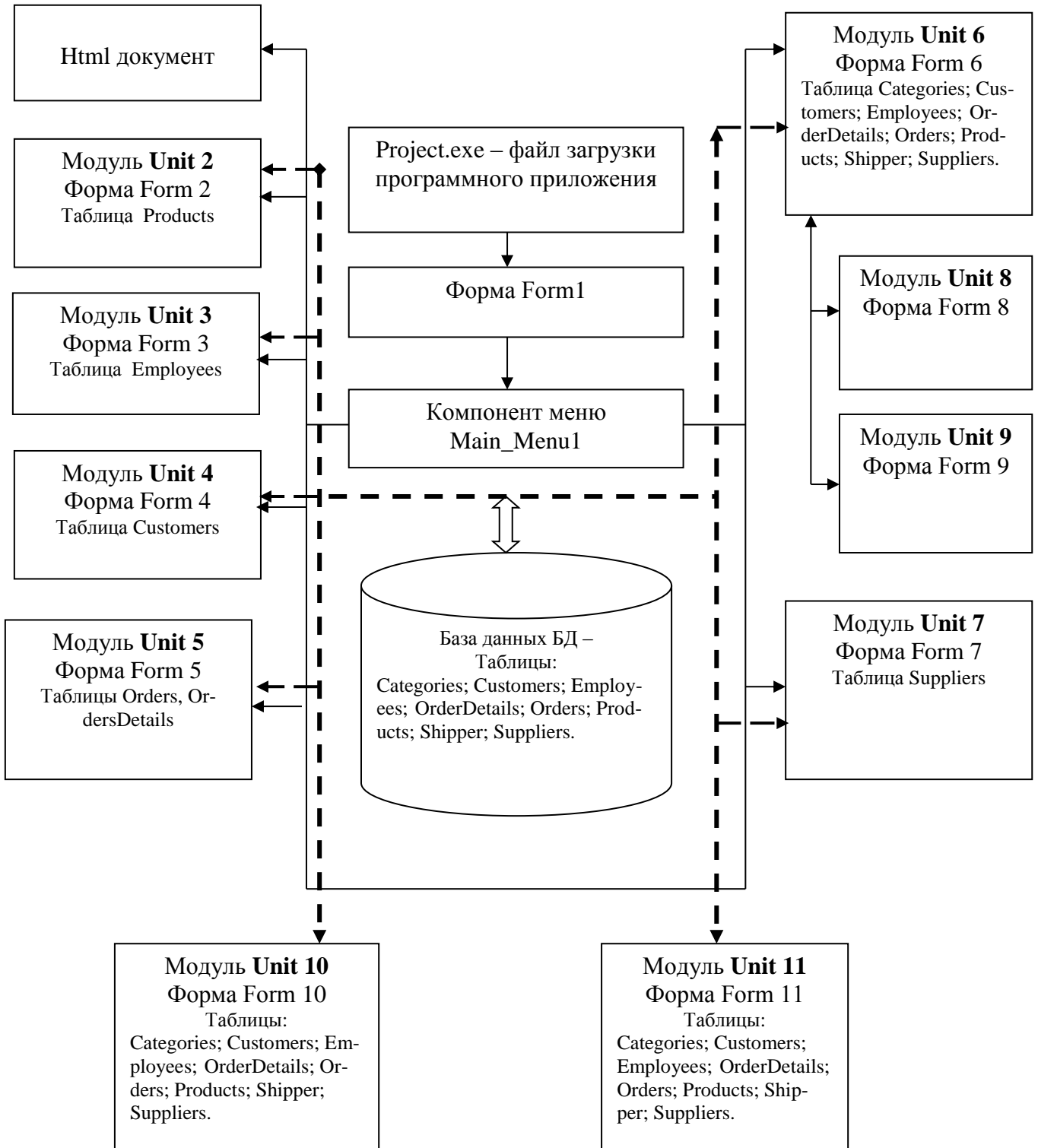


Рис.2.26. Схема взаимосвязи программных модулей и информационных файлов

2.2.2. Организация технологического процесса сбора, передачи, обработки и выдачи информации

2.2.2.1. Схема технологического процесса сбора, передачи, обработки и выдачи информации и её описание

Схема технологического процесса сбора, передачи, обработки и выдачи информации представлена на рис.2.27.

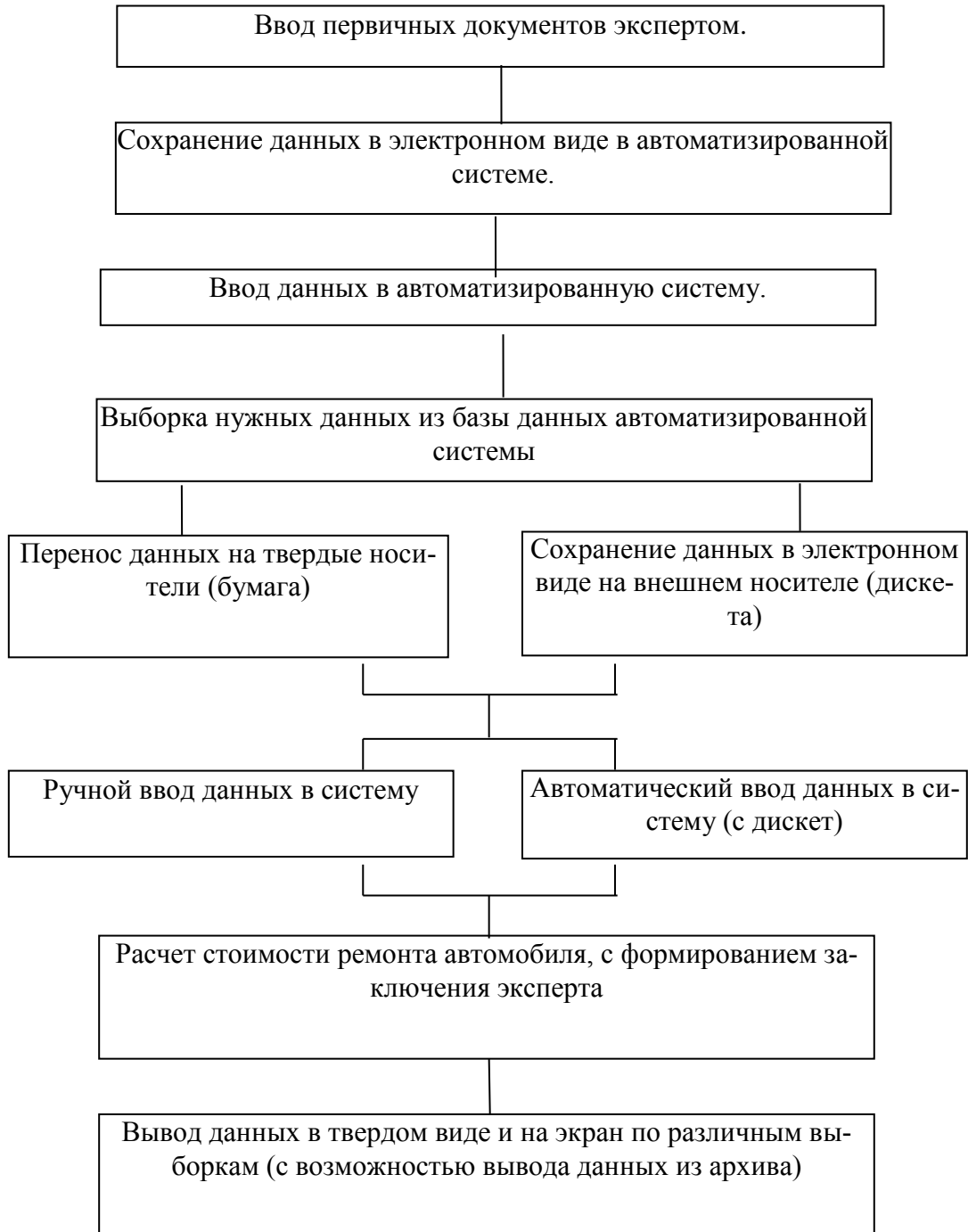


Рис.2.27. Сбор, передача, обработка и выдача информации

2.2.2.2. Инструкционные карты основных операций технологического процесса

При запуске аналитической системы, включающей в себя хранилище данных по продажам торгового предприятия и осуществляющейся файлом DNS.exe, на экране открывается окно системы (рис. 2.28).

Дата	Товар	Покупатель	ФИО менеджера	Число прод	Всего	Кол. пр. тов.
12.01.2020	LED телевизор Haier LE24K6000S	Вагабов	Касимов А.П.	1	12 334,00р.	1
15.01.2020	LED телевизор Haier LE40K6000SI	ООО КТП	Увайсов П.Р.	1	23 445,00р.	1
25.01.2020	LED телевизор Haier LE43K6500TI	Петрова А.П.	Абдуллаева Д.К.	1	45 500,00р.	2
30.01.2020	LED телевизор Haier LE32K6000S	Сидоров А.П.	Талибов А.П.	1	34 588,00р.	2
03.02.2020	LED телевизор Samsung UE32N450	ООО Стар	Магомедов П.Р.	1	54 000,00р.	1
06.02.2020	Ultra HD (4K) LED телевизор LG 65	Абдуллаев А.П.	Рагимов А.П.	1	60 000,00р.	2
12.02.2020	Ultra HD (4K) LED телевизор LG 43	Кульманова А.П.	Магомедов П.Р.	1	40 000,00р.	1
26.02.2020	LED телевизор LG 32LK510BPLD	ООО Квантум	Васильев А.П.	1	23 000,00р.	2
02.03.2020	Ultra HD (4K) LED телевизор Haier	ООО Техноснаб	Увайсов П.Р.	1	45 000,00р.	1
05.03.2020	Ultra HD (4K) LED телевизор Samst	Вагабов	Карибов А.П.	1	23 400,00р.	1
08.03.2020	LED телевизор Haier LE32K6000S	ООО ЦАТ	Магомедов П.Р.	1	88 800,00р.	3
25.03.2020	LED телевизор Haier LE24K6000S	ООО Квантум	Увайсов П.Р.	1	56 000,00р.	1
12.04.2020	Ultra HD (4K) LED телевизор LG 55	Иванов А.П.	Рагимов А.П.	1	45 999,00р.	1
17.04.2020	LED телевизор Haier LE32K6000S	ООО КТП	Рагимов А.П.	1	45 660,00р.	1

Рис.2.28. Главное окно системы.

Управление программного приложения осуществляется посредством системного меню, а также панели быстрых кнопок. Системное меню состоит из четырех пунктов. Рассмотрим основные операции выполняемые при помощи этого меню. Первый пункт «Данные» открывает 6 команд:

- Товары.
- Сотрудники.
- Документы.
- Покупатели.
- Поставщики.
- Продажи.

Выбор каждого из этих пунктов открывает диалоговое окно справочники с возможностью просмотра, редактирования и при необходимости ввода новых данных (рис.2.29 – 2.34).

№	Название	КП	Ед. изм	Цена
6	LED телевизор Haier LE24K6000S	3 шт.		13 999,00р.
5	LED телевизор Haier LE40K6000SF	3 шт.		18 999,00р.
4	Ultra HD (4K) LED телевизор Haier L	2 шт.		39 999,00р.
3	LED телевизор Haier LE43K6500TF	2 шт.		24 999,00р.
2	LED телевизор Haier LE32K6000S	2 шт.		12 999,00р.
14	Ultra HD (4K) LED телевизор Samsung	6 шт.		56 999,00р.
13	LED телевизор Samsung UE32N4500A	6 шт.		16 999,00р.
12	LED телевизор Samsung UE43J5272A	6 шт.		26 999,00р.
11	Ultra HD (4K) LED телевизор LG 65U	1 шт.		59 999,00р.
10	Ultra HD (4K) LED телевизор LG 55U	1 шт.		44 999,00р.
9	Ultra HD (4K) LED телевизор LG 43U	5 шт.		31 999,00р.
8	LED телевизор LG 32LK510BPLD	4 шт.		15 999,00р.
7	Ultra HD (4K) LED телевизор Haier L	3 шт.		27 999,00р.

Экран
 Диагональ: 40" (101.6 см)
 Технология: LED
 Стандарт HD: Full HD
 Разрешение экрана: 1920x1080 Пикс
 Яркость: 250 кд/м²
 Контрастность: 5000:1
Функции
 Воспроизведение видео через USB: Есть
Интерфейсы
 HDMI: 2
 Кол-во разъемов USB: 1
 Цвет: Черный

Рис.2.29. Форма «Данные – Товары».

№	ФИО
8	Васильев А.П.
7	Талибов А.П.
6	Рагимов А.П.
5	Касимов А.П.
4	Увайсов П.Р.
3	Абдуллаева Д.К.
2	Карибов А.П.
1	Магомедов П.Р.

Данные сотрудника

Код сотрудника:

ФИО:

Дата рождения:

Дата поступления на работу:

Должность:

Рис.2.30. Форма «Данные – Сотрудники».

Данные: Покупатели

Покупатель
ООО ЦАТ
ООО Техноснаб
Вагабов
ООО КТР
Иванов А.П.
Петрова А.П.
Сидоров А.П.
ООО Стар
Абдуллаев А.П.
Кульманова А.П.
ООО Квантум

Код покупателя: 10

Покупатель: Иванов А.П.

в лице: _____

должность: _____

Адрес: Гаджиева, д.56, кв.56

Город: Махачкала

Страна: РФ

Телефон: _____

E-mail: _____

Рис.2.31. Форма «Данные – Покупатели».

Данные: Продажи

№	Дата
20	15.05.2020
19	12.05.2020
18	08.05.2020
17	04.05.2020
16	02.05.2020
15	12.04.2020
14	17.04.2020
13	25.03.2020
12	15.03.2020
11	08.03.2020
10	05.03.2020
9	02.03.2020
8	26.02.2020
7	12.02.2020

Код продажи: 20

Дата продажи: 15.05.2020

Дата отгрузки: 15.05.2019

Товар: 11

Покупатель: 8

Продавец: 3

№ документа: 20

Количество: 1

Стоимость, сумма: 23 344,00р.

Скидка: 4 500,00р.

Рис.2.32. Форма «Данные – Продажи»

Пункт «OLAP – Кубы» позволяет формировать различные отчеты, которые можно разделить на аналитические данные по различным срезам куба и графическое представление многомерного анализа.

С помощью разработанного приложения были реализованы следующие отчеты, подотчеты и диаграммы:

The screenshot shows a software application window titled "Анализ продаж" (Sales Analysis). The window contains a pivot table with the following data:

Сумма продаж *	Товар	Покупатель	ФИО менеджера	Дата продажи
				январь, 2020
				февраль, 2020
				апрель, 2020
				май, 2020
				Sum
	Товар			
	LED телевизор Haier LE24K6			12 334,00р.
	LED телевизор Haier LE32K6			34 588,00р.
	LED телевизор Haier LE40K6			23 445,00р.
	LED телевизор Haier LE43K6			
	LED телевизор Haier LE43K6			56 000,00р.
	LED телевизор Haier LE43K6			45 500,00р.
	LED телевизор LG 32LK510B			23 000,00р.
	LED телевизор Samsung UE3			54 000,00р.
	Ultra HD (4K) LED телевизор			45 000,00р.
	Ultra HD (4K) LED телевизор			40 000,00р.
	Ultra HD (4K) LED телевизор			45 999,00р.
	Ultra HD (4K) LED телевизор			34 999,00р.
	Ultra HD (4K) LED телевизор			60 000,00р.
	Ultra HD (4K) LED телевизор			23 400,00р.
	Ultra HD (4K) LED телевизор			23 400,00р.
	Sum			115 867,00р.

Рис.2.33. Отчет по проданным товарам в разрезе суммы.

Данный отчет позволяет просмотреть результаты анализа проданных товаров с разбивкой по дате продажи. Данный отчет далее будет использоваться как подотчет для других сущностей. При нажатии на «Покупатель» конкретного клиента происходит спуск на более низкий уровень детализации, где например можно посмотреть на все товары, купленные клиентом:

Анализ продаж							
Сумма продаж	Товар	Покупатель	ФИО менеджера		Дата продажи		
Товар	Покупатель	Дата продажи					
		январь, 2020	февраль, 2020	март, 2020	апрель, 2020	май, 2020	
LED телевизор Haier LE24K6	Вагабов	12 334,00р.					
	ООО Квантум			56 000,00р.			
	Sum	12 334,00р.		56 000,00р.			
LED телевизор Haier LE32K6	ООО КТР				45 660,00р.		
	ООО ЦАТ			88 800,00р.			
	Сидоров А.П.	34 588,00р.					
	Sum	34 588,00р.		88 800,00р.	45 660,00р.		
LED телевизор Haier LE40K6	ООО КТР	23 445,00р.					
	Sum	23 445,00р.					
LED телевизор Haier LE43K6	ООО Квантум					56 000,00р.	
	Sum					56 000,00р.	
LED телевизор Haier LE43K6	Петрова А.П.	45 500,00р.					
	Sum	45 500,00р.					
LED телевизор LG 32LK510B	Абдуллаев А.П.					82 999,00р.	
	ООО Квантум		23 000,00р.				
	Sum		23 000,00р.			82 999,00р.	
LED телевизор Samsung UE3	ООО Стар		54 000,00р.				
	Sum		54 000,00р.				
Ultra HD (4K) LED телевизор	ООО Техноснаб			45 000,00р.			

Рис.2.34. Отчет по проданным товарам, с разбивкой по покупателям

При нажатии на услугу в свою очередь происходит переход на отчет по конкретной услуге, где можно посмотреть на все факты покупки данной услуги:

Анализ продаж							
Сумма продаж	Товар	Покупатель	ФИО менеджера		Дата продажи		
Товар	Покупатель	ФИО менеджера		Дата продажи			
				январь, 2020	февраль, 2020	март, 2020	апрель, 2020
LED телевизор Haier LE24K6	Вагабов	Касимов А.П.		12 334,00р.			
		Sum		12 334,00р.			
	ООО Квантум	Увайсов П.Р.				56 000,00р.	
		Sum				56 000,00р.	
LED телевизор Haier LE32K6	ООО КТР	Рагимов А.П.					45 660,00р.
		Sum					45 660,00р.
	ООО ЦАТ	Магомедов П.Р.			88 800,00р.		
LED телевизор Haier LE40K6	Сидоров А.П.	Талибов А.П.	34 588,00р.				
		Sum	34 588,00р.				
	ООО КТР	Увайсов П.Р.	23 445,00р.				
		Sum	23 445,00р.				
LED телевизор Haier LE43K6	ООО Квантум	Талибов А.П.					
		Sum					
	Петрова А.П.	Абдуллаева Д.К.	45 500,00р.				

Рис.2.35. Отчет по проданным товарам, с разбивкой по покупателям и менеджерам

Так же можно посмотреть на отчет по всем товарам, когда-либо проданным предприятием DNS:

Товар	Сумма продаж
LED телевизор Haier LE24K6	68 334,00р.
LED телевизор Haier LE32K6	169 048,00р.
LED телевизор Haier LE40K6	23 445,00р.
LED телевизор Haier LE43K6	56 000,00р.
LED телевизор Haier LE43K6	45 500,00р.
LED телевизор LG 32LK510B	105 999,00р.
LED телевизор Samsung UE3	54 000,00р.
Ultra HD (4K) LED телевизор	45 000,00р.
Ultra HD (4K) LED телевизор	74 999,00р.
Ultra HD (4K) LED телевизор	45 999,00р.
Ultra HD (4K) LED телевизор	83 344,00р.
Ultra HD (4K) LED телевизор	23 400,00р.
Sum	795 068,00р.

Рис.2.36. Все купленные товары

И непосредственно отчет (Рис.2.37) по всем сотрудникам осуществившим продажи, с указанием сумм по каждому месяцу. Данный отчет позволяет определить качество работы сотрудников компании. Отчет может быть сформирован как по суммам, так и по количеству проданных товаров (Рис.2.38)

ФИО менеджера	январь, 2019	фев, 2019	мар, 2019	апр, 2019	май, 2019	Sum
Абдуллаева Д.К.	45 500,00р.				23 344,00р.	68 844,00р.
Васильев А.П.		23 000,00р.			48 999,00р.	71 999,00р.
Карибов А.П.			23 400,00р.			23 400,00р.
Касимов А.П.	12 334,00р.					12 334,00р.
Магомедов П.Р.		94 000,00р.	88 800,00р.			182 800,00р.
Рагимов А.П.		60 000,00р.		91 659,00р.		151 659,00р.
Талибов А.П.	34 588,00р.				90 999,00р.	125 587,00р.
Увайсов П.Р.	23 445,00р.		101 000,00р.		34 000,00р.	158 445,00р.
Sum	115 867,00р.	177 000,00р.	213 200,00р.	91 659,00р.	197 342,00р.	795 068,00р.

Рис.2.37. Отчет по качеству работы сотрудников в денежном эквиваленте.

Так же были построены диаграммы, которые отображают, например суммарную популярность товаров, как за весь период, так и за отдельный месяц (Рис.2.39).

Анализ продаж						
Кол-во продаж	Товар	Покупатель	ФИО менеджера	Дата продажи		
				янв, 2019	фев, 2019	мар, 2019
						апр, 2019
						май, 2019
	Товар	ФИО менеджера				
	LED телевизор Haier LE24K6000S	Касимов А.П.		1		
		Увайсов П.Р.				1
		Sum		1		1
	LED телевизор Haier LE32K6000S	Магомедов П.Р.				1
		Рагимов А.П.				1
		Талибов А.П.		1		
		Sum		1		1
	LED телевизор Haier LE40K6000S	Увайсов П.Р.		1		
		Sum		1		
	LED телевизор Haier LE43K6000S	Талибов А.П.				1
		Sum				1
	LED телевизор Haier LE43K6000S	Абдуллаева Д.К.		1		
		Sum		1		
	LED телевизор LG 32LK510BPLD	Васильев А.П.			1	
		Увайсов П.Р.				1
		Sum			1	2
	LED телевизор Samsung UE32N4500AU	Магомедов П.Р.			1	
		Sum			1	
	Ultra HD (4K) LED телевизор Haier LE43K6500U	Увайсов П.Р.				1

Рис.2.38. Отчет по качеству работы сотрудников в количественном эквиваленте.

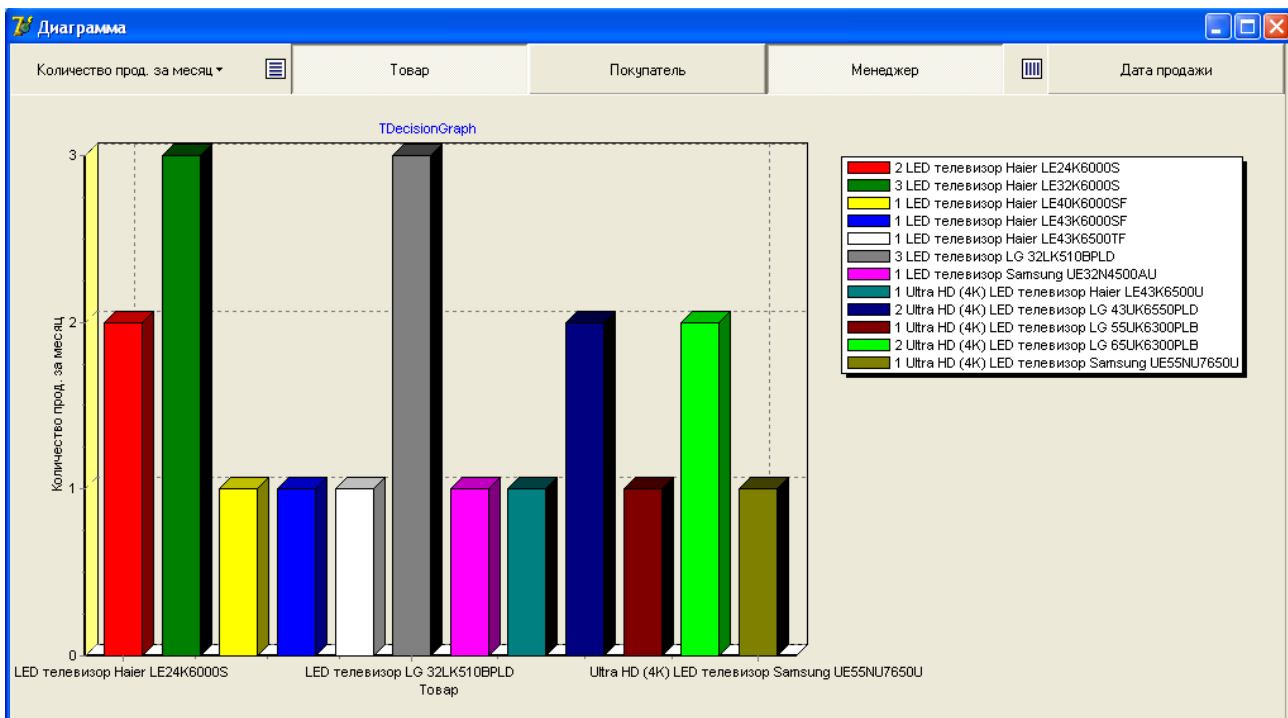


Рис.2.39. Распределение популярности по товару

Фильтрация и выборка по дате и любому другому полю возможна для любого созданного отчета.

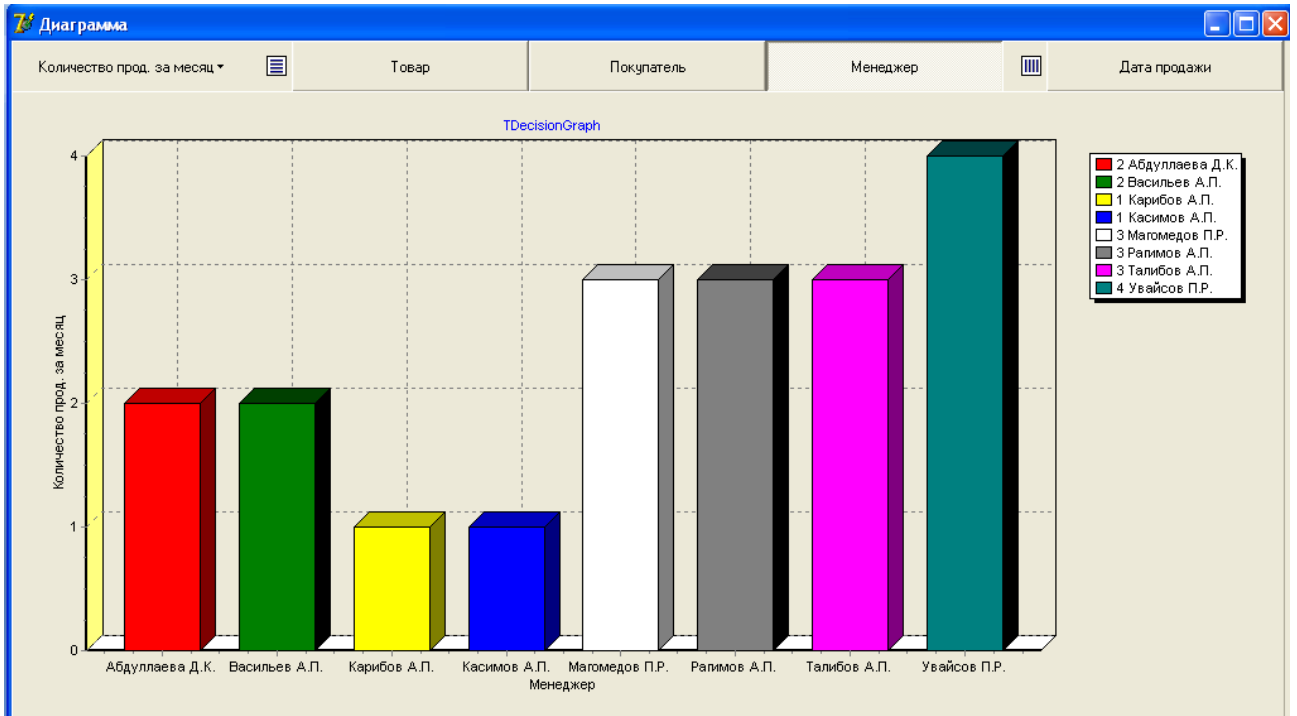


Рис.2.40. Распределение продаж по продавцам

3.ОБОСНОВАНИЕ ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА

3.1. Выбор и обоснование методики расчета экономической эффективности проекта

В соответствии с ГОСТ 24.702 - 85 целесообразные варианты построения АИС выбираются путем балансирования показателей приращения эффекта \mathcal{E} , получаемого за счет создания или совершенствования АИС, и затрат Q . Математически эту задачу формулируют в виде [1]:

$$\text{MAX } \mathcal{E} \text{ при } Q = \text{const.}$$

или в виде обратной задачи:

$$\text{MIN } Q \text{ при } \mathcal{E} = \text{const.}$$

При оценке эффективности СОЭИ используют обобщающие и частные показатели.

К основным обобщающим показателям экономической эффективности относятся:

1. Расчетный коэффициент эффективности капитальных вложений:

$$E_p = \Delta\Pi / K$$

где $\Delta\Pi$ - годовая экономия (годовой прирост прибыли), руб.;

K - единовременные затраты, руб.

2. Годовой экономический эффект:

$$\mathcal{E} = \Delta\Pi - K * E_n,$$

где E_n - нормативный коэффициент эффективности капитальных вложений (в соответствии с [1] $E_n = 0,15$).

Произведение $K * E_n$ в данном случае следует рассматривать как нормативную прибыль, которая должна быть получена от внедрения системы.

3. Срок окупаемости :

$$T = K/\Delta\Pi = 1/E_p.$$

Расчет перечисленных обобщающих показателей предполагает предварительное вычисление частных показателей, характеризующих создаваемую или модернизируемую АИС, таких как

- годовая экономия (годовой прирост прибыли);
- единовременные затраты на разработку и внедрение системы;
- среднегодовая трудоемкость функционирования;
- длительность обработки информации;
- надежность технических средств;
- увеличение затрат вследствие ненадежности КТС (комплекса технических средств), руб.
- достоверность и др.

Годовая экономия $\Delta\Pi$ рассчитывается следующим образом:

$$\Delta\Pi = Z_{\text{с}} - Z_{\text{п}},$$

где $Z_{\text{с}}$, $Z_{\text{п}}$ - приведенные к одному году затраты на обработку информации соответственно при существующем и предполагаемом вариантах организации АИС.

Среднегодовые затраты на обработку информации ($Z_{\text{с}}$, $Z_{\text{п}}$), приведенные выше в формуле, должны определяться с учетом всех стадий жизненного цикла АИС:

$$Z_{\text{п}} = (P+C)/T_{\text{экс}} + \Phi + A/T_{\text{мод}} + Z,$$

где P - стоимость приобретения и освоения используемых средств автоматизации проектирования, руб.;

C - единовременные затраты на создание и внедрение системы, не учитываемые в себестоимости машино-часа, руб.;

$T_{\text{экс}}$ - предполагаемый срок эксплуатации АИС, лет;

Φ - среднегодовые затраты на функционирование АИС (текущие затраты), руб.;

A - единовременные затраты на модернизацию (адаптацию системы к изменившимся условиям применения), руб.;

$T_{\text{мод}}$ - среднее время между смежными периодами модернизации, лет;
 Z - среднегодовая сумма потерь вследствие ненадежности АИС, руб.

Показатель P равен нулю, если при создании АИС привлекаются только штатные средства программного обеспечения ЭВМ (операционные системы и их утилиты, трансляторы с алгоритмических языков и т.д.). В остальных случаях значение показателя P определяется на основании соответствующих преискурантов.

Единовременные затраты на создание АИС (C) в общем виде равны сумме затрат на проектирование (R) и удельных затрат на приобретение, монтаж, наладку используемых средств ($K_{\text{вт}}$), однако, вследствие того, что $K_{\text{вт}}$ учитывается при расчете себестоимости машино-часа, во избежание двойного счета значение C в большинстве случаев следует принимать равным R .

Единовременные затраты на проектирование (R) определяются следующим образом:

$$R = S_{\text{тз}} * T_{\text{тз}} + S_{\text{тп}} * T_{\text{тп}} + S_{\text{рп}} * T_{\text{рп}} + S_{\text{вн}} * T_{\text{вн}},$$

где $T_{\text{тз}}$, $T_{\text{тп}}$, $T_{\text{рп}}$, $T_{\text{вн}}$ - трудоемкость соответствующей стадии создания системы - ТЗ, ТП, РП, внедрение;

$S_{\text{тз}}$, $S_{\text{тп}}$, $S_{\text{рп}}$, $S_{\text{вн}}$ - себестоимости чел.-дня проектировщика на соответствующих стадиях создания системы.

Срок предполагаемой эксплуатации определяется в соответствии с периодами морального старения соответствующей техники (8 лет).

Среднегодовые затраты на функционирование Φ определяется на основе построения оценки технологического процесса обработки информации (включая вне машинные и внутри машинные процессы):

$$\Phi = \sum_{i \in I} T_i^{\Phi} * C_i^{\Phi} * N_i + \sum_{i \in Y} M_i * C_i^M * N_i$$

где T_i^Φ - трудоемкость i - й операции технологических процессов до машинной (включая ручную обработку данных) и после машинной обработки данных, чел.-часы (нормо-часы);

C_i^Φ - себестоимость чел.-часа (нормо-часа) при выполнении i - й операции, руб.;

N_i - количество реализаций i - й операции в течение года;

M_i - машинное время необходимое для выполнения i - й операции;

C_i^M - себестоимость машино-часа при выполнении i - й операции;

I - множество операций, входящих в технологические процессы до машинной, внутри машинной и после машинной обработки информации;

Y - подмножество операций обработки информации на ЭВМ.

Трудоемкость выполнения i - й операции технологических процессов до машинной и после машинной обработки данных определяется следующим образом:

$$T_i^\Phi = Q_i / N_i$$

где Q_i - объем информации, обрабатываемой на i - й операции

N_i - среднечасовая норма выработки при выполнении i - й операции.

Трудоемкость выполнения i - й технологической операции на ЭВМ вычисляется по формуле

$$M_i = Q_i / N_i^M$$

где N_i^M - производительность машины при выполнении i - й технологической операции.

Затраты на единовременную адаптацию (модернизацию) оцениваются так же, как и затраты на проектирование, с той лишь разницей, что дополнительно учитывается коэффициент уменьшения трудоемкости, равной 0,5.

3.1.1. Разработка плана выполнения работ

Для организации работ по созданию автоматизированной системы необходимо составить оптимальный план выполнения работ. Для этой цели

составляется сетевой график и оптимизируется для получения заданной длительности процесса (длина критического пути) и приемлемого коэффициента использования ресурса (более 0.8). Все эти операции можно проделать вручную, что достаточно трудоемко и занимает много времени, вероятность ошибки при этом также велика. Выходом из положения является автоматизированная оптимизация сетевого графика с помощью ЭВМ. При этом, от пользователя требуется лишь составление первоначального плана работ.

При разработке данного дипломного проекта были проведены работы:

1. Разработка технического задания на дипломное проектирование;
2. Постановка задачи;
3. Утверждение технического задания;
4. Анализ технического задания;
5. Технико-экономическая характеристика объекта управления;
6. Описание экономической сущности решаемого комплекса задач;
7. Контроль руководителя;
8. Анализ существующих методов решения поставленной задачи;
9. Обоснование выбора ВТ;
10. Обоснование выбора состава информационного и программного обеспечения комплекса задач;
11. Изучение особенностей технологического процесса сбора, передачи, обработки и выдачи информации;
12. Построение информационной модели решаемого комплекса задач;
13. Контроль руководителя;
14. Разработка структур таблиц БД;
15. Разработка программного приложения для функционирования OLAP-системы;
16. Разработка макетов представления входных документов;
17. Разработка форм выходных документов;
18. Опытная эксплуатация;
19. Отладка разработанного программного комплекса;

20. Устранение недочетов;
21. Разработка инструкций для пользователей программы;
22. Контроль руководителя;
23. Расчет показателей экономической эффективности проекта;
24. Контроль руководителя;
25. Оформление пояснительной записки к дипломному проекту;

Таблица 3.1

Список работ

№	ч-д	Наименование работ	Тож	Z
1	0-1	Разработка технического задания на дипломное проектирование	1	2
2	1-2	Постановка задачи	1	2
3	2-3	Утверждение технического задания	2	1
4	3-4	Анализ технического задания	1	1
5	4-5	Технико-экономическая характеристика объекта управления	3	1
6	3-5	Описание экономической сущности решаемого комплекса задач	4	1
7	5-6	Контроль руководителя	0,5	1
8	5-7	Анализ существующих методов решения поставленной задачи	1	1
9	7-8	Обоснование выбора ВТ	1	1
10	8-9	Обоснование выбора состава информационного и программного обеспечения комплекса задач	2	2
11	9-10	Изучение особенностей технологического процесса сбора, передачи, обработки и выдачи информации	14	1
12	10-11	Построение информационной модели решаемого комплекса задач	2	1
13	11-12	Контроль руководителя	0,5	1
14	11-13	Разработка структур таблиц БД	8	1
15	13-14	Разработка программного приложения для OLAP-системы торговой сети DNS	17	1
16	14-15	Разработка макетов представления входных документов	4	1
17	15-16	Разработка форм выходных документов	3	1
18	16-17	Опытная эксплуатация	1	1
19	17-18	Отладка разработанного программного комплекса	1	1
20	18-19	Устранение недочетов	2	1
21	19-20	Разработка инструкций для пользователей программы	1	1
22	20-21	Контроль руководителя	1	1
23	20-22	Расчет показателей экономической эффективности проекта	9	1
24	22-23	Контроль руководителя	1	1
25	22-24	Оформление пояснительной записки к дипломному проекту	2	1

Фиктивные работы (7-8), (13-14), (22-23), (24-25) в таблице не обозначены.

По табл. 3.1 построим сетевой график и определим критический путь ($T_{кр}$)

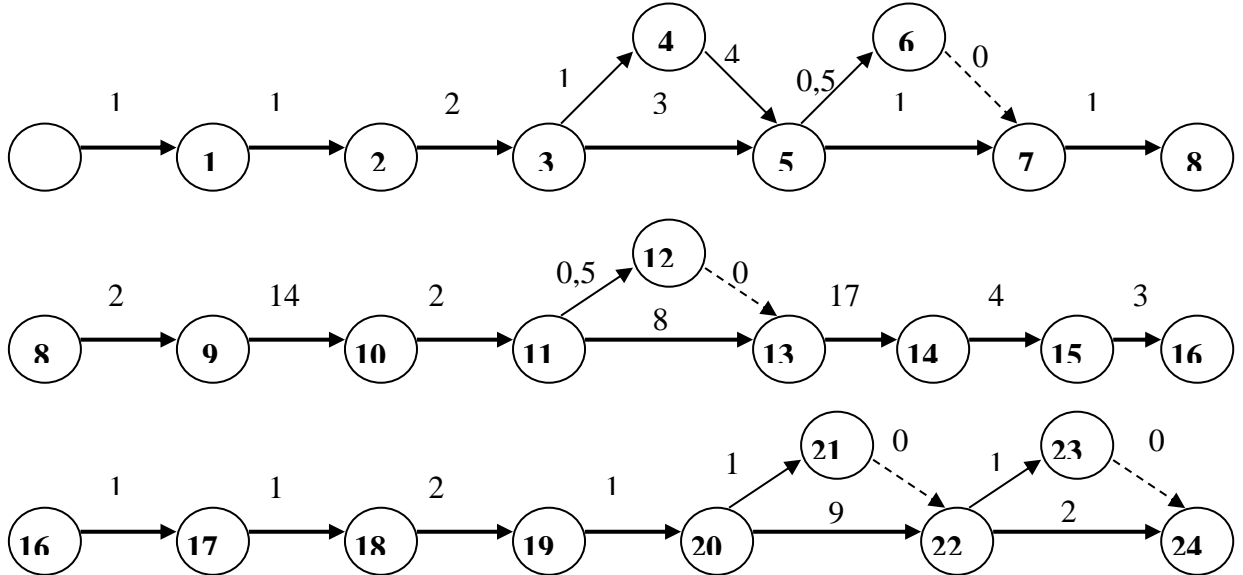


Рис.3.1. Сетевой график

После построения сетевого графика определяются основные временные параметры сетевого графика: ранний и поздний сроки наступления событий $T_i^{(p)}$, $T_i^{(n)}$; ранние и поздние сроки начала и окончания работ $t_{ij}^{(pH)}$, $t_{ij}^{(пн)}$, $t_{ij}^{(p.o)}$, $t_{ij}^{(п.о)}$; резервы времени работ и событий $r_{ij}^{(п)}$, $r_{ij}^{(св)}$, R_i . Результаты расчетов заносятся в табл. 3.2.

Таблица 3.2

Расчет параметров сетевого графика

№	Код		Параметры работ и событий в днях												
	i		t_{ij}	$T_i^{(p)}$	$T_i^{(n)}$	$T_j^{(p)}$	$T_j^{(n)}$	$t_{ij}^{(pH)}$	$t_{ij}^{(пн)}$	$t_{ij}^{(p.o)}$	$t_{ij}^{(п.о)}$	$r_{ij}^{(п)}$	$r_{ij}^{(св)}$	R_i	
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
1	0	1	1	0	0	1	1	1	0	1	1	0	0	0	
2	1	2	1	1	1	2	2	2	1	2	2	0	1	0	
3	2	3	2	2	2	4	4	4	2	4	4	0	2	0	
4	3	4	1	4	4	5	5	3	4	5	5	0	4	0	
5	4	5	3	5	5	9	9	8	6	9	9	0	6	0	

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
6	3	5	4	4	4	7	7	8	3	7	7	0	3	0
7	5	6	0,5	9	9	9,5	9,5	9,5	9	9,5	9,5	0	9	0
8	5	7	1	9	9	10	10	10	9	10	10	0	9	0
9	7	8	1	10	10	11	11	11	10	11	11	0	10	0
10	8	9	2	11	11	13	13	13	11	13	13	0	11	0
11	9	10	14	13	13	27	27	27	13	27	27	0	13	0
12	10	11	2	27	27	29	29	29	27	29	29	0	27	0
13	11	12	0,5	29	29	29,5	29,5	29,5	29	29,5	30	0	29	0
14	11	13	8	29	29	37	37	37	29	37	37	0	29	0
15	13	14	17	37	37	54	54	54	37	54	54	0	37	0
16	14	15	4	54	54	58	58	41	54	58	58	0	54	0
17	15	16	3	58	58	61	61	61	58	61	61	0	58	0
18	16	17	1	61	61	62	62	62	61	62	62	0	61	0
19	17	18	1	62	62	63	63	63	62	63	63	0	62	0
20	18	19	2	63	63	65	65	65	63	65	65	0	63	0
21	19	20	1	65	65	66	66	66	65	66	66	0	65	0
22	20	21	1	66	66	67	67	67	66	67	67	0	66	0
23	20	22	9	66	66	75	75	75	66	75	75	0	66	0
24	22	23	1	75	75	76	76	76	75	76	76	0	75	0
25	22	24	2	75	75	77	77	77	75	77	77	0	75	0

Разность между продолжительностью критического пути и продолжительностью любого полного пути является резервом времени этого пути R:

$$R(L_s) = T_{кр} - t(L_s)$$

Расчёт продолжительности и резервов времени путей сетевого графика производится в табл. 3.3.

Таблица 3.3

Расчет продолжительности путей сетевого графика

Обозначение пути	Последовательность событий пути	t(Ls)	R(Ls)
1	2	3	4
L1	0-1-2-3-5-7-8-9-10-11-13-14-15-16-17-18-19-20-22-24	75	2
L2	0-1-2-3-4-5-7-8-9-10-11-13-14-15-16-17-18-19-20-22-24	77	0

1	2	3	4
L3	0-1-2-3-5-6-7-8-9-10-11-13-14-15-16-17-18-19-20-22-24	74,5	2,5
L4	0-1-2-3-5-7-8-9-10-11-12-13-14-15-16-17-18-19-20-22-24	67,5	9,5
L5	0-1-2-3-5-7-8-9-10-11-13-14-15-16-17-18-19-20-21-22-24	69	8
L6	0-1-2-3-5-7-8-9-10-11-13-14-15-16-17-18-19-20-22-23-24	76	1
L7	0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15-16-17-18-19-20-21-22-23-24	58	19
L8	0-1-2-3-4-5-6-7-8-9-10-11-13-14-15-16-17-18-19-20-22-24	76,5	0,5

Критический путь данного сетевого графика имеет параметры: $T_{кр} = 77$ дней.

3.1.2. Трудоемкость разработки программного обеспечения

Если известны по опыту работы или заданы по нормативам затраты труда на подготовку описания задачи t_o , исследование алгоритма решения задачи t_n , разработку блок-схемы алгоритма t_a , программирование по готовой блок-схеме t_n , отладку программы на ЭВМ $t_{отл}$, подготовку документации по задаче t_d , то трудоемкость разработки программного обеспечения решения задачи можно рассчитать по формуле (3.1)

$$t = t_o + t_n + t_a + t_n + t_{отл} + t_d. \quad (3.1)$$

Составляющие затрат труда, в свою очередь, можно определить через условное число операторов в разрабатываемом программном обеспечении. В их число входят операторы, которые необходимо написать программисту в процессе работы над задачей с учетом возможных уточнений в постановке задачи и совершенствования алгоритма. Условное число операторов Q в программе задачи может быть определено по формуле

$$Q = qc(1 + p) = 750 * 1,5 * (1 + 0,05) = 1181,25$$

где q - предполагаемое число операторов; c - коэффициент сложности программы; p - коэффициент коррекции программы в ходе ее разработки. Помимо названных выше используются коэффициенты квалификации разработчика алгоритмов и программ - k и увеличения затрат труда вследствие недостаточного или некачественного описания задачи - B .

При оценке затрат труда на разработку задачи предполагается, что подготовка описания задачи осуществляется одними исполнителями, а все остальные работы - другими. Затраты труда на подготовку описания задачи точно определить невозможно, так как это связано с творческим характером работы.

Затраты труда на изучение описания задачи $t_{и}$ с учетом уточнения описания и квалификации программиста могут быть определены по формуле, чел-ч:

$$t_{и} = QB / (75 \div 85)k = 1181,25 * 1,3 / 83 * 1,1 = 20,35$$

Затраты труда на разработку алгоритма решения задачи t_a рассчитываются по формуле, чел-ч

$$t_a = Q / (20 \div 25)k = 1181,25 / 23 * 1,1 = 56,49$$

Затраты труда на составление программы по готовой блок-схеме $t_{п}$ определяются по формуле, чел-ч:

$$t_{п} = Q / (20 \div 25)k = 1181,25 / 23 * 1,1 = 56,49$$

Затраты труда на отладку программы на ЭВМ $t_{отл}$ рассчитывается по следующим формулам, чел-ч:

при автономной отладке одной задачи

$$t_{отл} = Q / (45)k = 1181,25 / 45 * 1,1 = 28,88$$

при комплексной отладке задачи

$$t_{отл}^k = 1,5t_{отл} = 1,5 * 28,88 = 43,32$$

Затраты труда на подготовку документации по задаче t_d определяются по формуле, чел-ч:

$$t_d = t_{др} + t_{до},$$

где $t_{др}$ - затраты труда на подготовку материалов в рукописи, равные $Q/(15 \div 20)k = 1181,25/18 * 1,1 = 72,19$;

$t_{до}$ - затраты труда на редактирование, печать и оформление документации, равные $0,75t_{др} = 0,75 * 96,25 = 54,14$

$$t_d = 72,19 + 54,14 = 126,33$$

Полная средняя трудоемкость разработки программы:

$$t_{р.п} = 0,83Q/k = 0,83 * 1181,25/1,1 = 891,31$$

Трудоемкость разработки программного обеспечения решения данной задачи равна:

$$t = t_o + t_{и} + t_a + t_{п} + t_{отл} + t_d = 8 + 20,35 + 56,49 + 56,49 + 28,88 + 126,33 = 296,54.$$

$$296,54/8 = 37 \text{ чел.-дней.}$$

3.2. Расчет показателей экономической эффективности проекта

3.2.1. Смета затрат на разработку программы

Затраты на разработку программы состоят из:

- прямой производственной заработной платы;
- дополнительной заработной платы;
- начислений на заработную плату;
- услуги сторонних организаций ;
- накладные расходы;
- отчисления в пенсионный фонд;
- отчислений на социальное страхование;
- отчисление на медицину.

Необходимо определить полную себестоимость, прибыль, налог на полную себестоимость, налог на прибыль, отпускную цену разработки.

Рассмотрим отдельно каждую статью сметы затрат:

1. Фзарп – прямая производственная заработная плата работников, занятых разработкой программы определяется исходя из продолжительности

- сти разработки, количества исполнителей и их месячного должностного оклада;
2. Фдоп – дополнительная заработная плата принимается 20% от основной производственной заработной платы;
 3. Н – Начисления на заработную плату составляют 13% от общей заработной платы;
 4. Смаш – услуги сторонних организаций;
 5. Фес – отчисления в единый социальный налог 30% (пенсионный фонд – 22% от общей заработной платы, социальное страхование – 2,9 % от общей заработной платы, на медицину составляют 5,1% от общей заработной платы) [19];
 6. Нр – накладные расходы определяются исходя из установленного их процента в общих затратах на разработку программы ($Np = (Фзарп + Фдоп + Н + Смаш) * b / (1 - b)$), ($b=0,2$).

Определим себестоимость разработанной программы.

Имеем: количество разработчиков (М) - 3 чел.; период времени разработки (Траз) - 77 дней; оклад разработчика (Сок) – 2696 руб, оклад руководителя и консультанта по 25000 руб; использованные средства проектирования компьютер стандарта РС – IntelCorei3; период использования ЭВМ (Тэвм) – 37 дней; стоимость машинного часа (Смаш) – 25 руб.

Необходимо определить единовременные затраты на разработку программы (R) (полную себестоимость программы), прибыль, налог на полную себестоимость, налог на прибыль, отпускную цену разработки.

В соответствии с этим вычислим соответствующие им показатели затрат на разработку.

Вычислим себестоимость одного человеко-дня на стадии ($T_1=77-37=40$ чел.-дн.), когда не пользовались средствами проектирования (S_1):

(На первом этапе Смаш равны нулю).

$Фзарп=2696+(25000/(26*8))*12+(25000/(26*8))*4=5700,81$ руб.

$Фдоп=0,2*Фзарп=0,2*5700,81=1140,16$ руб.

$$H = (\Phi_{\text{зарп}} + \Phi_{\text{доп}}) * 13\% = (5700,81 + 1140,16) * 0,13 = 889,33 \text{ руб.}$$

$$\Phi_{\text{пенс}} = (\Phi_{\text{зарп}} + \Phi_{\text{доп}}) * 22\% = (5700,81 + 1140,16) * 0,22 = 1505,01 \text{ руб.}$$

$$\Phi_{\text{соц.страх.}} = (\Phi_{\text{зарп}} + \Phi_{\text{доп}}) * 2,9\% = (5700,81 + 1140,16) * 0,029 = 198,39 \text{ руб.}$$

$$\Phi_{\text{мед}} = (\Phi_{\text{зарп}} + \Phi_{\text{доп}}) * 5,1\% = (5700,81 + 1140,16) * 0,051 = 348,89 \text{ руб.}$$

$$H_p = (\Phi_{\text{зарп}} + \Phi_{\text{доп}} + H + C_{\text{маш}}) * (b / (1 - b)) = 1938,82 \text{ руб.}$$

$$S_1 = (5700,81 + 1140,16 + 889,33 + 1505,01 + 198,39 + 348,89 + 1938,82) / 26 = 558,16 \text{ руб.}$$

Вычислим себестоимость одного человеко-дня на стадии ($T_2 = 37$ чел.-дн.), когда разработчики пользовались средствами проектирования (арендовали ЭВМ):

$$C_{\text{маш}} = 25 \text{ руб.} * 8 = 200 \text{ руб.}$$

$$S_2 = S_1 + (C_{\text{маш}} + C_{\text{маш}} * b / (1 - b)) = 558,16 + (200 + 200 * 0,2 / 0,8) = 637,81 \text{ руб.}$$

По формуле получим:

$$R = \sum_{i=1}^2 S_i * T_i = S_1 * T_1 + S_2 * T_2 = 45925,46 \text{ руб.}$$

Определим отпускную цену разработанной программы с учетом нормативной прибыли, налога на прибыль, налога на добавленную стоимость, которые определяются, соответственно, как 15% от себестоимости (R), 20% от прибыли и 20% от добавленной стоимости[18]:

Стоимость программы (Ц) равна:

$$Ц = R + 0,15 * R / (1 - 0,20) = 45925,46 + 0,15 * 45925,46 / 0,8 = 54536,49 \text{ руб.}$$

$$C_{\text{отп}} = Ц + 0,20 * Ц = 54536,49 + 0,18 * 54536,49 = 65443,79 \text{ руб.}$$

При этом нормативная чистая прибыль равна:

$$0,15 * R = 0,15 * 45925,46 = 6888,82 \text{ руб.}$$

Итак, отпускная цена разработки:

$$C_{\text{отп}} = 65443,79 \text{ руб.}$$

3.2.2. Расчет среднегодовых затрат на функционирование АИС

Расчет среднегодовых затрат на функционирование рассматриваемой OLAP -системы торговой сети магазинов DNS проводится для существую-

щей системы и разрабатываемой. Существующая система представляет собой выполнение расчетов ручным способом. Наименования работ и трудовые и стоимостные затраты на эти работы при ручной (существующей) и машинной (предлагаемой) обработке информации приведены в табл. 3.4 и 3.5. В этих же таблицах приведены результаты расчетов.

Таблица 3.4

Существующая система обработки:

№ п/п	Наименование операции технологического процесса решения комплекса задач	Оборудование	Ед.изм	Объем работы	Норма выработки в час	Трудоемкость (гр.5/гр.6)
1	2	3	4	5	6	7
1	Приём, регистрация, входных документов.	---	док /стр.	200	20,57	9,72
2	Сортировка	---	док. /стр.	200	40	5,00
3	Анализ продаж в торговой сети (сравнительный метод, доходный, затратный)	РС IBM-совмест.	док /стр.	10	0,25	40,00
4	Заполнение документов	РС IBM-совмест.	док-т	5	1,53	3,27
5	Контроль, регистрация выдача значений	---	док-т	5	42,35	0,12
Одноразовое решение					58,11	
Итого за год					697,31	

№ п/п	Средне- часовая з/п оператора (руб)	Часовая амортиз. (руб)	Часовая стоимость наклад. Расх. (руб)	Стоимость работы оборуд. (гр.8 + гр.9 + гр.10)	Стоимостные затраты (гр.7 * гр.11) (руб)
1	8	9	10	11	12
1	112,18	0	62,50	166,67	1620,48
2	112,18	0	62,50	166,67	833,33
3	112,18	0	62,50	166,67	6666,67
4	112,18	0	62,50	166,67	544,66
5	112,18	0	62,50	166,67	19,68
Одноразовое решение					9684,82
Итого за год					116217,87

Пояснения к табл.:

$$1) N_{\text{выр1}} = 3600 / (((V_{\text{сд}} + V_{\text{пд}}) * K_{\text{с}}) + V_{\text{рд}}) = 3600 / (((2 + 3) * 30) + 25) = 20,57$$

где $V_{\text{сд}}$ – время сверки даты,

$V_{\text{пд}}$ – время поиска нужных данных,

$K_{\text{с}}$ – число строк в документе;

$V_{\text{рд}}$ – время регистрации документа.

$$2) N_{\text{выр2}} \text{ – принимается исходя из имеющегося опыта (40).}$$

$$3) N_{\text{выр3}} = 3600 / ((V_{\text{зз}} * K_{\text{зс}} + V_{\text{рас}}) * K_{\text{с}}) = 3600 / ((0,9 * 87) + 135) * 30 = 0,25$$

где $V_{\text{зз}}$ – время записи 1 знака,

$K_{\text{зс}}$ – количество знаков в строке,

$K_{\text{с}}$ – количество строк,

$V_{\text{рас}}$ – время расчета,

$$4) N_{\text{выр4}} = 3600 / (V_{\text{зз}} * K_{\text{зс}} * K_{\text{с}}) = 3600 / (0,9 * 87 * 30) = 1,53$$

где $V_{\text{зз}}$ – время записи 1 знака,

$K_{зс}$ – количество знаков в строке;

K_c – количество строк,

5) $N_{выр5}$ - принимается исходя из имеющегося опыта

$$N_{выр5} = 3600 / 85 = 42,35$$

Средняя почасовая зарплата операторов вычисляется следующим образом: «зарплата за месяц» / («количество дней» * «длительность рабочей смены») = $17500 / (26 * 6) = 112,18$ руб.

Накладные расходы = 60% от почасовой заработной платы:

$$= 112,18 * 0,6 = 62,50 \text{ руб.}$$

Среднегодовые затраты при существующей системе обработке информации равны сумме затрат по всем операциям: 116217,87 руб.

Таблица 3.5

Машинная обработка:

№ п/п	Наименование операции технологического процесса решения комплекса задач	Оборудование	Ед. изм	Объем работы	Норма выработки в час	Трудоемкость (гр.5/гр.6)
1	2	3	4	5	6	7
1	Приём, регистрация входных документов.	---	док /стр.	210	20,57	10,21
2	Ввод данных в ЭВМ	IntelCore i3	знак	6000	7200	0,83
3	Анализ продаж товаров	Intel Core i3	знак	6000	6000	1,00
4	Печать выходных форм.	Intel Core i3 HP LaserJet Pro M104a RU	док.	5	111,92	0,04
5	Контроль, регистрации выдача значений.	---	док.	5	42,35	0,12
Одноразовое решение					12,21	
Итого за год					146,46	

Продолжение табл. 3.5.

№ п/п	Средне- часовая з/п оператора (руб)	Часовая амортиз. (руб)	Часовая стоимость наклад. расх. (руб)	Стоимость работы оборуд. (гр.8 + гр.9 + гр.10)	Стоимостные затраты (гр.7 * гр.11) (руб)
1	8	9	10	11	12
1	112,18	0	62,50	166,67	1701,51
2	112,18	2	62,50	168,67	140,56
3	112,18	2	62,50	168,67	168,67
4	112,18	2	62,50	168,67	7,54
5	112,18	2	62,50	168,67	19,91
Одноразовое решение					2038,18
Итого за год					24458,13

Пояснения к табл.:

$$1) N_{\text{выр1}} = 3600 / (((V_{\text{сд}} + V_{\text{пд}}) * K_{\text{с}}) + V_{\text{рд}}) = 3600 / (((2 + 3) * 30) + 25) = 20,57$$

где $V_{\text{сд}}$ – время сверки даты,

$V_{\text{пд}}$ – время поиска нужных данных,

$K_{\text{с}}$ – число строк в документе;

$V_{\text{рд}}$ – время регистрации документа.

$$2) N_{\text{выр2}} = 3600 / V_{\text{зз}} = 3600 / 0,5 = 7200$$

где $V_{\text{зз}}$ – время записи одного знака.

$$3) N_{\text{выр3}} = 3600 / V_{\text{обр.д}} = 3600 / 0,6 = 6000$$

где $V_{\text{обр.д}}$ – время обработки данных.

$$4) N_{\text{выр4}} = 3600 / (V_{\text{фд}} + K_{\text{сз}} / C_{\text{п}}) = 3600 / (0,5 + 7600 / 240) = 111,92$$

где $V_{\text{фд}}$ – время формирования документа,

$K_{\text{сз}}$ – среднее количество знаков в документе,

$C_{\text{п}}$ – скорость принтера;

5) $N_{\text{выр5}}$ принимается исходя из имеющегося опыта

$$N_{\text{выр5}} = 3600 / 85 = 42,35$$

Среднегодовые затраты при машинной обработке информации равны сумме затрат по всем операциям: $\Phi = 24458,13$ руб.

$$З_{\text{п}} = (P+C)/T_{\text{экс}} + \Phi = 54536,49/8 + 24458,13 = 31275,2 \text{ руб.}$$

3.2.3. Расчет показателей экономической эффективности

1. Расчетный коэффициент эффективности капитальных вложений:

$$E_p = \Delta\Pi / K$$

где $\Delta\Pi$ - годовая экономия (годовой прирост прибыли), руб.;

K - единовременные затраты, руб.

$$\Delta\Pi = Z_6 - Z_n = 116217,87 - 31275,2 = 84942,67 \text{ руб.}$$

$$E_p = 84942,67 / 54536,49 = 1,56$$

2. Годовой экономический эффект:

$$\mathcal{E} = \Delta\Pi - K * E_n,$$

где E_n - нормативный коэффициент эффективности капитальных вложений

$$(E_n = 0,15).$$

$$\mathcal{E} = 84942,67 - 54536,49 * 0,15 = 76762,2 \text{ руб.}$$

3. Срок окупаемости:

$$T = K / \Delta\Pi = 1 / E_p.$$

$$T = 54536,49 / 84942,67 = 0,64 \text{ года.}$$

ЗАКЛЮЧЕНИЕ

Хранилища данных, будучи спроектированы правильным образом, предоставляют все необходимые механизмы для доступа к информации, важной для принятия решений в компании в любой момент времени. А так же, делают доступ к информации максимально удобным, а саму информацию максимально достоверной. Гибкость хранилищ данных позволяет обеспечить все будущие потребности в компании, за счет внесения исключительно небольших изменений в архитектуру хранилища.

В данном дипломном проекте разработана OLAP-система для торговой сети магазинов DNS. Для достижения поставленной цели были решены следующие задачи:

- изучена предметная область, т. е. процесс управления в торговой сети магазинов;
- проведен анализ существующей организации бизнес-процессов;
- осуществлена постановка задачи автоматизации бизнес-процессов;
- создана ИС и с помощью разработанной ИС осуществлен анализ эффективности OLAP-системы;
- разработано программное обеспечение для управления учета торговых операций и проведения многомерного анализа;
- выполнен расчет экономической эффективности разработанной автоматизированной информационной системы.

При проектировании использовались СУБД Microsoft Access 2017, объектно-ориентированный язык программирования Delphi XE, CASE-средство ERWin, которое позволяет создавать графические модели бизнес-процессов.

Для расчета экономической эффективности автоматизированной системы выбрана методика расчета экономической эффективности, выполнен расчет показателей экономической эффективности. Расчет показал экономическую эффективность разработки

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Абдулгалимов А.М., Мурадов М.М., Адеева М.Г. Методические указания к выполнению дипломных проектов студентами специальности 080801 – «Прикладная информатика в экономике». - Махачкала: ДГТУ, 2013
2. Автоматизированные информационные технологии в экономике. /Под общей ред. И.Т. Трубилина. – М.: Финансы и статистика, 2003.
3. Автоматизированные информационные технологии в экономике. /Под ред. Г.А. Титоренко. – М.: Компьютер, ЮНИТИ, 2004.
4. Архангельский А.Я. Программирование в С++ Builder - М: ЗАО «Издательство БИНОМ» 2010.
5. Архангельский А.Я. Программирование в С++ Builder 6 – М: ЗАО «Издательство БИНОМ», 2003
6. Бабаев Ю.А., Петров А. М., Мельникова Л. А. Бухгалтерский учет. Учебник для бакалавров. - Проспект, 2015.
7. Басовский Л. Е. Экономический анализ. М., 2003
8. Брусакова И.А., Чертовской В.Д. Информационные системы и технологии в экономике - М.: Финансы и статистика, 2007
9. Грабауров В. Информационные технологии – М.: Издательство «Современный мир», 2006.
10. Диго С.М. Проектирование баз данных. М.: Финансы и статистика, 2002.
11. Захарьин В.Р. Все о бухгалтерских проводках. - Эксмо, 2012.
12. Информационные системы в экономике. Под ред. В.В. Дик. – М.: Финансы и статистика, 1996.
13. ИцикБен-Ган Microsoft SQL Server 2012. Высокопроизводительный код T-SQL. Оконные функции. –СПб.:БХВ-Петербург, 2013
14. Карминский А.М., Черников Б.В. Информационные системы в экономике, в 2 частях, ч.1 – М.: Финансы и статистика, 2006

15. Карпова Т. Базы данных: модели, разработка, реализация. – СПб.: Питер, 2011.
16. Кириллов В.В., Основы проектирования реляционных баз данных. СУБД – СПб: БХВ-Петербург, 2008.
17. Климова Л. М. Delphi 7. Основы программирования. Решение типовых задач. Самоучитель / Л.М. Климова. - М.: КУДИЦ-Образ, 2017
18. Марков А.С. Базы данных. Введение в теорию и методологию. М.: Финансы и статистика, 2002.
19. Методические указания к выполнению лабораторного практикума по дисциплине «Базы данных». Абдулгалимов А.М., Мурадов М.М., Махачкала: ДГТУ, 2012.
20. Налоговый кодекс Российской Федерации. М.: «Издательство ЭЛИТ», 2007.
21. Палий В.Ф. Современный бухгалтерский учет. – М.: Изд-во «Бухгалтерский учет», 2012.
22. Репин В., Бизнес-процессы. Моделирование, внедрение, управление – М.: Манн, Иванов и Фербер, 2013
23. Фигурнов В. Э. IBM PC для пользователя. - М.: ИНФРА-М, 1997.
24. Шутенко Ю. Visual Foxpro для профессионалов – СПб:БХВ – Петербург, 2009.

Интернет-источники

1. <http://DNS-dagestan.ru> – сайт ООО "DNS".
2. <http://www.informationweek.com/differences-of-opinion/17800088>
3. <http://www.kimballgroup.com/2003/09/17/the-bottom-up-misnomer>

```

#ifndef Unit1H
#define Unit1H
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <Menus.hpp>
//-----
class TForm1 : public TForm
{
__published:    // IDE-managed Components
    TMainMenu *MainMenu1;
    TMenuItem *N1;
    TMenuItem *N2;
    TMenuItem *N3;
    TMenuItem *N4;
    TMenuItem *N5;
    TMenuItem *N6;
    TMenuItem *N7;
    TMenuItem *N8;
    TMenuItem *N9;
    TMenuItem *N10;
    TMenuItem *N11;
    TMenuItem *N12;
    TMenuItem *N13;
    TMenuItem *N14;
    TMenuItem *N15;
    TMenuItem *N16;
    TLabel *Label1;
    void __fastcall N3Click(TObject *Sender);
    void __fastcall N4Click(TObject *Sender);
    void __fastcall N5Click(TObject *Sender);
    void __fastcall N9Click(TObject *Sender);
    void __fastcall N10Click(TObject *Sender);
    void __fastcall N11Click(TObject *Sender);
    void __fastcall N13Click(TObject *Sender);
    void __fastcall N14Click(TObject *Sender);
    void __fastcall N15Click(TObject *Sender);
    void __fastcall N16Click(TObject *Sender);
    void __fastcall N1Click(TObject *Sender);
    void __fastcall N8Click(TObject *Sender);
    void __fastcall N7Click(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm1(TComponent* Owner);
};
//-----
extern PACKAGE TForm1 *Form1;
//-----
#endif
//-----
#include <vcl.h>
#include <ShellAPI.h>
#pragma hdrstop
#include "Unit1.h"
#include "Unit2.h"
#include "Unit3.h"

```



```

#include "Unit4.h"
#include "Unit5.h"
#include "Unit6.h"
#include "Unit7.h"
#include "Unit8.h"
#include "Unit9.h"
#include "Unit10.h"
#include "Unit11.h"
#include "Unit12.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm1 *Form1;
//-----
__fastcall TForm1::TForm1(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm1::N3Click(TObject *Sender)
{
if (!Form2->Visible) Form2->Show();
}
//-----
void __fastcall TForm1::N4Click(TObject *Sender)
{
if (!Form3->Visible) Form3->Show();
}
//-----
void __fastcall TForm1::N5Click(TObject *Sender)
{
if (!Form4->Visible) Form4->Show();
}
//-----
void __fastcall TForm1::N9Click(TObject *Sender)
{
if (!Form5->Visible) Form5->Show();
}
//-----
void __fastcall TForm1::N10Click(TObject *Sender)
{
if (!Form6->Visible) Form6->Show();
}
//-----
void __fastcall TForm1::N11Click(TObject *Sender)
{
Form7->QuickRep1->Preview();
}
//-----
void __fastcall TForm1::N13Click(TObject *Sender)
{
Form8->QuickRep1->Preview();
}
//-----
void __fastcall TForm1::N14Click(TObject *Sender)
{
Form9->QuickRep1->Preview();
}
//-----
void __fastcall TForm1::N15Click(TObject *Sender)
{
Form10->QuickRep1->Preview();
}

```

```

}
//-----
void __fastcall TForm1::N16Click(TObject *Sender)
{
Form1->QuickRep1->Preview();
}
//-----
void __fastcall TForm1::N1Click(TObject *Sender)
{
ShellExecute(Handle,NULL,"d:\\VUZ\\spravka.htm",NULL,NULL,SW_RESTORE);
}
//-----
void __fastcall TForm1::N8Click(TObject *Sender)
{
Form1->Close();
}
//-----
void __fastcall TForm1::N7Click(TObject *Sender)
{
if (!Form12->Visible) Form12->Show();
}
//-----
#include <vcl.h>
#pragma hdrstop
#include "Unit2.h"
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm2 *Form2;
//-----
__fastcall TForm2::TForm2(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm2::Button3Click(TObject *Sender)
{
Table1->Delete();
}
//-----
#ifndef Unit2H
#define Unit2H
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <DB.hpp>
#include <DBCtrls.hpp>
#include <DBGrids.hpp>
#include <DBTables.hpp>
#include <ExtCtrls.hpp>
#include <Grids.hpp>
//-----
class TForm2 : public TForm
{
__published: // IDE-managed Components
    TLabel *Label1;
    TLabel *Label2;
    TLabel *Label3;
    TLabel *Label4;
    TLabel *Label5;

```

```

TLabel *Label6;
TEdit *Edit1;
TEdit *Edit2;
TEdit *Edit3;
TEdit *Edit4;
TEdit *Edit5;
TEdit *Edit6;
TDataSource *DataSource1;
TDBGrid *DBGrid1;
TDBNavigator *DBNavigator1;
TButton *Button1;
TButton *Button2;
TTable *Table1;
TLabel *Label7;
TSmallintField *Table1Year;
TSmallintField *Table1Gos_vuz;
TSmallintField *Table1Kom_vuz;
TIntegerField *Table1Ch_stud;
TIntegerField *Table1Kol_abit;
TSmallintField *Table1Vipusk;
TButton *Button3;
void __fastcall FormActivate(TObject *Sender);
void __fastcall Table1AfterScroll(TDataSet *DataSet);
void __fastcall Button1Click(TObject *Sender);
void __fastcall Button2Click(TObject *Sender);
void __fastcall Button3Click(TObject *Sender);
private:
public:
__fastcall TForm2(TComponent* Owner);
};
//-----
extern PACKAGE TForm2 *Form2;
//-----
#endif
//-----
#include <vcl.h>
#pragma hdrstop
#include "Unit3.h"
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm3 *Form3;
//-----
__fastcall TForm3::TForm3(TComponent* Owner)
: TForm(Owner)
{
}
//-----
void __fastcall TForm3::FormActivate(TObject *Sender)
{
Edit1->Text = Table1->FieldByName("Year")->AsString;
Edit2->Text = Table1->FieldByName("Gos_sh")->AsString;
Edit3->Text = Table1->FieldByName("Kom_sh")->AsString;
Edit4->Text = Table1->FieldByName("Sh_gosSH")->AsString;
Edit5->Text = Table1->FieldByName("Sh_komSH")->AsString;
}
//-----
void __fastcall TForm3::Table1AfterScroll(TDataSet *DataSet)
{
Edit1->Text = Table1->FieldByName("Year")->AsString;
Edit2->Text = Table1->FieldByName("Gos_sh")->AsString;
Edit3->Text = Table1->FieldByName("Kom_sh")->AsString;
}

```

```

Edit4->Text = Table1->FieldByName("Sh_gosSH")->AsString;
Edit5->Text = Table1->FieldByName("Sh_komSH")->AsString;
}
//-----

void __fastcall TForm3::Button2Click(TObject *Sender)
{
Table1->Edit();
Table1->FieldByName("Year")->AsInteger = StrToInt(Edit1->Text);
Table1->FieldByName("Gos_sh")->AsInteger = StrToInt(Edit2->Text);
Table1->FieldByName("Kom_sh")->AsInteger = StrToInt(Edit3->Text);
Table1->FieldByName("Sh_gosSH")->AsInteger = StrToInt(Edit4->Text);
Table1->FieldByName("Sh_komSH")->AsInteger = StrToInt(Edit5->Text);
Table1->Post();
}
//-----

void __fastcall TForm3::Button1Click(TObject *Sender)
{
Table1->Append();
}
//-----

void __fastcall TForm3::Button3Click(TObject *Sender)
{
Table1->Delete();
}
//-----

#ifndef Unit3H
#define Unit3H
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>
#include <Forms.hpp>
#include <DB.hpp>
#include <DBGrids.hpp>
#include <DBTables.hpp>
#include <Grids.hpp>
#include <DBCtrls.hpp>
#include <ExtCtrls.hpp>
//-----
class TForm3 : public TForm
{
__published: // IDE-managed Components
    TLabel *Label1;
    TLabel *Label2;
    TLabel *Label3;
    TLabel *Label4;
    TLabel *Label5;
    TButton *Button1;
    TButton *Button2;
    TEdit *Edit1;
    TEdit *Edit2;
    TEdit *Edit3;
    TEdit *Edit4;
    TEdit *Edit5;
    TDataSource *DataSource1;
    TDBGrid *DBGrid1;
    TTable *Table1;
    TLabel *Label6;
    TSmallintField *Table1Year;
    TSmallintField *Table1Gos_sh;

```

```

TSmallintField *Table1Kom_sh;
TIntegerField *Table1Sh_gosSH;
TSmallintField *Table1Sh_komSH;
TButton *Button3;
TDBNavigator *DBNavigator1;
void __fastcall FormActivate(TObject *Sender);
void __fastcall Table1AfterScroll(TDataSet *DataSet);
void __fastcall Button2Click(TObject *Sender);
void __fastcall Button1Click(TObject *Sender);
void __fastcall Button3Click(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm3(TComponent* Owner);
};
//-----
extern PACKAGE TForm3 *Form3;
//-----
#endif
//-----
#include <vcl.h>
#pragma hdrstop
#include "Unit4.h"
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm4 *Form4;
//-----
__fastcall TForm4::TForm4(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm4::FormActivate(TObject *Sender)
{
    Edit1->Text = Table1->FieldByName("Year")->AsString;
    Edit2->Text = Table1->FieldByName("Sdd")->AsString;
    Edit3->Text = Table1->FieldByName("Z_p")->AsString;
}
//-----
void __fastcall TForm4::Table1AfterScroll(TDataSet *DataSet)
{
    Edit1->Text = Table1->FieldByName("Year")->AsString;
    Edit2->Text = Table1->FieldByName("Sdd")->AsString;
    Edit3->Text = Table1->FieldByName("Z_p")->AsString;
}
//-----
/-----
void __fastcall TForm4::Button2Click(TObject *Sender)
{
    Table1->Edit();
    Table1->FieldByName("Year")->AsInteger = StrToInt(Edit1->Text);
    Table1->FieldByName("Sdd")->AsInteger = StrToInt(Edit2->Text);
    Table1->FieldByName("Z_p")->AsInteger = StrToInt(Edit3->Text);
    Table1->Post();
}
//-----
#ifndef Unit4H
#define Unit4H
//-----
#include <Classes.hpp>
#include <Controls.hpp>
#include <StdCtrls.hpp>

```

```

#include <Forms.hpp>
#include <DB.hpp>
#include <DBGrids.hpp>
#include <DBTables.hpp>
#include <Grids.hpp>
#include <DBCtrls.hpp>
#include <ExtCtrls.hpp>
//-----
class TForm4 : public TForm
{
__published: // IDE-managed Components
    TTable *Table1;
    TLabel *Label1;
    TLabel *Label2;
    TLabel *Label3;
    TEdit *Edit1;
    TEdit *Edit2;
    TEdit *Edit3;
    TDBGrid *DBGrid1;
    TDataSource *DataSource1;
    TButton *Button1;
    TButton *Button2;
    TLabel *Label4;
    TSmallintField *Table1Year;
    TSmallintField *Table1Sdd;
    TSmallintField *Table1Z_p;
    TButton *Button3;
    TDBNavigator *DBNavigator1;
    void __fastcall FormActivate(TObject *Sender);
    void __fastcall Table1AfterScroll(TDataSet *DataSet);
    void __fastcall Button1Click(TObject *Sender);
    void __fastcall Button3Click(TObject *Sender);
    void __fastcall Button2Click(TObject *Sender);
private: // User declarations
public: // User declarations
    __fastcall TForm4(TComponent* Owner);
};
//-----
extern PACKAGE TForm4 *Form4;
//-----
#endif
//-----
#include <vcl.h>
#include <math.h>
#pragma hdrstop

#include "Unit5.h"
#include "Unit1.h"
//-----
#pragma package(smart_init)
#pragma resource "*.dfm"
TForm5 *Form5;
//-----
__fastcall TForm5::TForm5(TComponent* Owner)
    : TForm(Owner)
{
}
//-----
void __fastcall TForm5::Button1Click(TObject *Sender)
{
    int i,j,k;
    float x[12][4],y[12],xt[4][12],c[4][8],e[4],a[4];
    for (i=0;i<12;i++)

```

```

{
    Table1->First();
    Table1->MoveBy(i);
    x[i][0]=1;
    x[i][1]=Table1->FieldByName("K_abit")->AsFloat;
    x[i][2]=Table1->FieldByName("K_sholn")->AsFloat;
    x[i][3]=Table1->FieldByName("Sr_doh")->AsFloat;
    y[i]=Table1->FieldByName("Vipusk")->AsFloat;
}
for (i=0;i<12;i++)
{
    for(j=0;j<4;j++)
    {
        xt[j][i]=x[i][j];
    }
}
for (i=0;i<4;i++)
{
    for(j=0;j<4;j++)
    {
        c[i][j]=0;
        for (k=0;k<12;k++)
        {
            c[i][j]=c[i][j]+xt[i][k]*x[k][j];
        }
    }
}
for (i=0;i<4;i++)
{
    e[i]=0;
    for (j=0;j<12;j++)
    {
        e[i]=e[i]+xt[i][j]*y[j];
    }
}
for (i=0;i<4;i++)
{
    for (j=4;j<8;j++)
    {
        if ((i+4)==j)
            c[i][j]=1;
        else
            c[i][j]=0;
    }
}
float s,s1;
for (i=0;i<4;i++)
{
    s=c[i][i];
    for (j=0;j<8;j++)
    {
        c[i][j]=c[i][j]/s;
    }
    for (k=0;k<4;k++)
    {
        if (k!=i)
        {
            s1=c[k][i];
            for (j=0;j<8;j++)
            {
                c[k][j]=c[k][j]-s1*c[i][j];
            }
        }
    }
}

```

```

    }
}
for (i=0;i<4;i++)
{
    a[i]=0;
    for (j=4;j<8;j++)
    {
        a[i]=a[i]+c[i][j]*e[j-4];
    }
}
Edit1->Text=ceil(a[0]*1000)/1000;
Edit2->Text=ceil(a[1]*1000)/1000;
Edit3->Text=ceil(a[2]*1000)/1000;
Edit4->Text=ceil(a[3]*1000)/1000;
Table2->Edit();
Table2->FieldByName("K1")->AsFloat =ceil(a[0]*1000)/1000;
Table2->FieldByName("K2")->AsFloat =ceil(a[1]*1000)/1000;
Table2->FieldByName("K3")->AsFloat =ceil(a[2]*1000)/1000;
Table2->FieldByName("K4")->AsFloat =ceil(a[3]*1000)/1000;
Table2->Post();
Table2->Refresh();
}
//-----

void __fastcall TForm6::FormCreate(TObject *Sender)
{
    float k1,k2,k3,k4;
    Table2->Active=true;

    k1=Table2->FieldByName("K1")->AsFloat;
    k2=Table2->FieldByName("K2")->AsFloat;
    k3=Table2->FieldByName("K3")->AsFloat;
    k4=Table2->FieldByName("K4")->AsFloat;
}
//-----

void __fastcall TForm6::Button2Click(TObject *Sender)
{
    int i;
    float y[12],yr[12];
    i=0;
    Table1->First();
    while(!Table1->Eof)
    {
        y[i]=Table1->FieldByName("Vipusk")->AsFloat;
        yr[i]=Table1->FieldByName("Raschet")->AsFloat;
        Table1->Next();
        i++;
    }
    float Y1=0,s=0, s1=0,s2=0;
    for(i=0;i<12;i++)
    {
        s+=y[i];
        {
            s1=s1+pow((y[i]-Y1),2);
            s2=s2+pow((y[i]-yr[i]),2);
        }
    }
    float F,R;
    R=1-(s2/s1);
    F=(R/4)/((1-R)/8);
    Edit1->Text=ceil(F*1000)/1000;
}
//-----

void __fastcall TForm6::Button3Click(TObject *Sender)

```



```

{
int i;
float x1[12],x2[12],x3[12];
i=0;
Table1->First();
while(!Table1->Eof)
{
x1[i]=Table1->FieldByName("K_abit")->AsFloat;
x2[i]=Table1->FieldByName("K_sholn")->AsFloat;
x3[i]=Table1->FieldByName("Sr_doh")->AsFloat;
Table1->Next();
i++;
}
float sum=0,sum1=0,sum2=0,x11[12],x22[12],x33[12],x1r,x2r,x3r,yras;
for(i=0;i<11;i++)
{
x11[i]=x1[i+1]-x1[i];
sum=sum+x11[i];
Table3->First();
Table3->MoveBy(i);
Table3->Edit();
Table3->FieldByName("X11")->AsFloat = x11[i];
}
x1r=x1[11]+sum/12;
Edit3->Text=ceil(x1r*1000)/1000;
for(i=0;i<11;i++)
{
x22[i]=x2[i+1]-x2[i];
sum1=sum1+x22[i];
Table3->First();
Table3->MoveBy(i);
Table3->Edit();
Table3->FieldByName("X22")->AsFloat = x22[i];
}
x2r=x2[11]+sum1/12;
Edit6->Text=ceil(x2r*1000)/1000;
for(i=0;i<11;i++)
{
x33[i]=x3[i+1]-x3[i];
sum2=sum2+x33[i];
Table3->First();
Table3->MoveBy(i);
Table3->Edit();
Table3->FieldByName("X33")->AsFloat = x33[i];
}
x3r=x3[11]+sum2/12;
Edit7->Text=ceil(x3r*1000)/1000;
float k1,k2,k3,k4;
Table2->Active=true;
k1=Table2->FieldByName("K1")->AsFloat;
k2=Table2->FieldByName("K2")->AsFloat;
k3=Table2->FieldByName("K3")->AsFloat;
k4=Table2->FieldByName("K4")->AsFloat;
Edit8->Text=ceil((k1+k2*x1r+k3*x2r+k4*x3r)*1000)/1000;
Table3->Refresh();
}

```


Результаты проверки уникальности

Антиплагиат - возможность бе... | СМИ назвали нулевого зараже... | Почта Mail.ru | Результаты проверки - Антиплагиат | Вывод отчета на печать - Антиплагиат

users.antiplagiat.ru/report/print/19?v=1&cc=0&short=true

АНТИПЛАГИАТ
ТВОРИТЕ СОБСТВЕННЫМ УМОМ

Отчет о проверке на заимствования №1


 Автор: muradovm70@mail.ru / ID: 6411014
Проверяющий: muradovm70@mail.ru / ID: 6411014
Отчет предоставлен сервисом «Антиплагиат»: <http://users.antiplagiat.ru>

ИНФОРМАЦИЯ О ДОКУМЕНТЕ

№ документа: 19
Начало загрузки: 18.06.2020 15:28:37
Длительность загрузки: 00:00:04
Имя исходного файла:
ДипломДНС_Валиев.pdf
Название документа: ДипломДНС_Валиев
Размер текста: 1 кБ
Символов в тексте: 119432
Слов в тексте: 13394
Число предложений: 795

ИНФОРМАЦИЯ ОБ ОТЧЕТЕ

Последний готовый отчет (ред.)
Начало проверки: 18.06.2020 15:28:42
Длительность проверки: 00:00:04
Комментарии: не указано
Модули поиска: Модуль поиска Интернет



ЗАИМСТВОВАНИЯ
33,25%

САМОЦИТИРОВАНИЯ
0%

ЦИТИРОВАНИЯ
0%

ОРИГИНАЛЬНОСТЬ
66,75%

Заимствования — доля всех найденных текстовых пересечений, за исключением тех, которые система отнесла к цитированиям, по отношению к общему объему документа.
Самоцитирования — доля фрагментов текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника, автором или соавтором которого является автор проверяемого документа, по отношению к общему объему документа.
Цитирования — доля текстовых пересечений, которые не являются авторскими, но система посчитала их использование корректным, по отношению к общему объему документа. Сюда относятся оформленные по ГОСТу цитаты; общеупотребительные выражения; фрагменты текста, найденные в источниках из коллекций нормативно-правовой документации.
Текстовое пересечение — фрагмент текста проверяемого документа, совпадающий или почти совпадающий с фрагментом текста источника.
Источник — документ, проиндексированный в системе и содержащийся в модуле поиска, по которому проводится проверка.
Оригинальность — доля фрагментов текста проверяемого документа, не обнаруженных ни в одном источнике, по которому шла проверка, по отношению к общему объему документа.
Заимствования, самоцитирования, цитирования и оригинальность являются отдельными показателями и в сумме дают 100%, что соответствует всему тексту проверяемого документа.
Обращаем Ваше внимание, что система находит текстовые пересечения проверяемого документа с проиндексированными в системе текстовыми источниками. При этом система является вспомогательным инструментом, определение корректности и правомерности заимствований или цитирований, а также авторства текстовых фрагментов проверяемого документа остается в компетенции проверяющего.

15:46 18.06.2020

Дипломник

Валиев В.М.

Руководитель

Мурадов М.М.

Заведующий кафедрой
ИТиПИВЭ

Абдулгалимов А.М.