

Министерство образования и науки Российской Федерации
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

**ИНСТИТУТ МАТЕМАТИКИ, ФИЗИКИ И ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ**

Кафедра «Прикладная математики и информатика»

09.03.03 ПРИКЛАДНАЯ ИНФОРМАТИКА

БИЗНЕС-ИНФОРМАТИКА

БАКАЛАВРСКАЯ РАБОТА

на тему Проектирование и реализация хранилища данных для анализа
бизнес деятельности компании

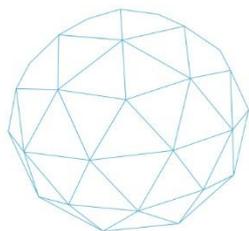
Студент Павлов В.В.

Руководитель Очеповский А.В.

Допустить к защите

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский

« » 20 г.



Тольятти 2017



Росдистант
ВЫСШЕЕ ОБРАЗОВАНИЕ ДИСТАНЦИОННО

Аннотация

Темой данной выпускной квалификационной работы является «Проектирование и реализация хранилища данных для анализа бизнес деятельности компании».

Работа выполнена студентом Тольяттинского Государственного Университета, института математики, физики и информационных технологий, группы ПИбд-1202а, Павловым Владимиром Владимировичем.

Объектом исследования являются процессы анализа бизнес информации хранилищ данных.

Предметом исследования является информационная система на основе хранилища данных для принятия решений в области продаж.

Целью работы является разработка хранилища данных для анализа бизнес-деятельности ООО «Экран».

Задачами работы являются:

- осуществление моделирования бизнес-процессов компании,
- разработка концептуальной и физической модели данных,
- проведение оценки эффективности разработки.

Выпускная квалификационная работа состоит из введения, трех глав, заключения, списка литературы и приложения.

В первой главе проводится анализ компании и разрабатываются модели данных, а также даются необходимые теоретические сведения о хранилищах данных.

Во второй главе происходит разработка хранилища данных.

В третьей главе проводится анализ эффективности разработанного проекта.

В заключении дается общая оценка, а также путь развития и применения работы.

Выпускная квалификационная работа содержит пояснительную записку объемом 54 страницы, включая 22 рисунок, 6 таблиц, список литературы из 16 наименований, приложение.

Содержание

ВВЕДЕНИЕ	4
1 ФОРМИРОВАНИЕ СПИСКА ТРЕБОВАНИЙ К ХРАНИЛИЩУ ДАННЫХ... 7	7
1.1 Описание деятельности ООО «Экран».....	7
1.2 Анализ существующей технологии анализа бизнес-деятельности	10
1.3 Формирование требований к новой информационной технологии анализа бизнес деятельности	14
1.4 Анализ решений для разработки хранилища данных и OLAP	15
1.5 Разработка архитектуры информационной системы для анализа данных	19
1.6 Концептуальное моделирование данных информационной системы для анализа данных.....	20
Вывод по главе	22
2 РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ ДЛЯ СИСТЕМЫ АНАЛИЗА ДАННЫХ.....	23
2.1 Обоснование выбора платформы для создания информационной системы	23
2.2 Разработка физической модели хранилища данных	26
2.2.1 Определение таблиц измерений	28
2.2.2 Определение таблицы фактов.....	28
2.3 Реализация хранилища данных	30
2.4 Тестирование приложения	33
Вывод по главе	35
3. ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА.....	37
3.1 Расчет показателей экономической эффективности проекта.....	37
3.2 Расчет базовых показателей экономической эффективности проекта.....	39
3.3 Показатели эффективности от внедрения проекта.....	40
Вывод по главе	44
ЗАКЛЮЧЕНИЕ	45
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ	46
ПРИЛОЖЕНИЕ А Спецификация требований к программному обеспечению.	48

ПРИЛОЖЕНИЕ Б Скрипт создания базы данных хранилища	51
ПРИЛОЖЕНИЕ В Таблицы хранилища данных	56
ПРИЛОЖЕНИЕ Г Тестирование приложения	58

ВВЕДЕНИЕ

Проблема хранения и обработки информации является очень важной в мире. Она важна и для больших компаний, через которые проходят тысячи транзакций в день, так и для более мелких. Несмотря на то, что для хранения повседневной информации существуют десятки решений, которая реализована в виде OLTP-систем (Online Transaction Processing), данное решение не подходит для обработки и анализа данных, необходимых для принятия решений.

В связи с этим возникла потребность в создании системы, использующих информацию из OLTP-систем, существенно не влияя на их работу. Для которых главной целью была бы поддержка в принятии аналитических решений. Такие системы получили название хранилищ данных.

Хранилище данных – спроектированная специальным образом информационная база данных, предназначенная для подготовки отчётов и бизнес-анализа с целью поддержки принятия решений в организации. Хранилища данных, будучи спроектированы правильным образом, предоставляют все необходимые механизмы для доступа к информации, важной для принятия решений в компании в любой момент времени. Что в свою очередь позволяет сделать доступ к информации более удобным, а саму информацию максимально достоверной.

На данный момент существует актуальная проблема сбора информации о текущем состоянии дел в компании для процесса принятия решений, которая может быть решена с помощью создания хранилища данных.

Практическая новизна представленной работы заключается в разработке подходящего решения к проектированию хранилища данных в области продаж, на основании анализа методов проектирования, а также разработка хранилища для конкретной компании, которая работает в данной области. Разработанное решение может быть использовано для дальнейшего проектирования хранилищ данных.

Таким образом, *объектом* исследования в представленной работе является процессы анализа бизнес-информации хранилищ данных. *Предметом* является информационная система на основе хранилища данных для принятия решений в области продаж

Основным *методом* исследования в данной работе выступает анализ выявленных проблем, с целью проектирования наиболее подходящего хранилища данных для более эффективного процесса принятия решений в компании.

Целью данной выпускной квалификационной работы является, разработка программного комплекса, предназначенного для более эффективного ведения бизнес-анализа.

Задачи, которые необходимо решить в процессе выполнения работы заключаются в изучении существующих подходов к проектированию хранилищ данных. После рассмотрения вышеизложенной задачи необходимо рассмотреть структурную составляющую компании, то есть провести анализ деятельности компании. Также необходимо провести моделирование моделей данных и спроектировать непосредственное хранилище.

Следующим этапом работы будет разработка хранилища данных и ETL процессов для загрузки в нее данных. В качестве среды для разработки хранилища данных было выбрано BI-решение Microsoft SQL Server 2012.

Работа состоит из введения, трех глав, заключения, списка используемой литературы, приложений.

В первой главе работы даются теоретические сведения о хранилищах данных и системах бизнес-анализа. Ставятся цели и задачи данной работы. Описывается актуальность выбранной темы в текущее время.

Во второй главе работы происходит обоснование выбранного средства реализации хранилища данных. Проектируется и разрабатывается хранилище данных.

В третьей главе работы с помощью методов для определения рентабельности инвестиций дается экономическое обоснование внедряемого решения.

В заключении подводятся общие итоги и даются рекомендации для последующего развития работы.

1 ФОРМИРОВАНИЕ СПИСКА ТРЕБОВАНИЙ К ХРАНИЛИЩУ ДАННЫХ

Хранение данных – задача, которую приходится решать каждой компании. Каждый день повышается объем информации необходимой для принятия важных управленческих решений, поэтому компаниям требуется организовать работу с информацией наилучшим образом. Поэтому возникла необходимость в создании специализированных систем хранения данных и дополнительных средств анализа поступающей информации.

Система хранения данных состоит из набора программного обеспечения и специализированного оборудования, предназначенного для хранения и передачи информации больших объемов.

Целью хранилищ данных является хранение «достоверной» информации, к которой несложно получить доступ людям, ответственным за принятие решений. Подготовка и представление данных – являются основными процессами в этой области. Процесс подготовки информации состоит из ETL (Extract, Transform, Load) процессов. Анализ и составление отчетов из собранной информации происходит после выполнения загрузки в область представления данных.

Использование аналитических систем для анализа данных позволяет лицам ответственным за принятия решений, путем выделения ключевых факторов эффективности, принимать решения, основанные на полученных результатах. Системы анализа данных, данные в которых собираются из всех учетных источников организации, позволяют руководителю видеть полную картину о состоянии бизнеса и избавляет от необходимости составления отчетов в различных системах.

1.1 Описание деятельности ООО «Экран»

Компания «Экран» является обществом с ограниченной ответственностью (ООО). Компания была основана в 2000 году и располагается по юридическому адресу: г. Тихвин, ул. Зайцева д.10.

Основным направлением деятельности компании «Экран» является реализация бытовой и компьютерной техники. В компании существуют несколько направлений работы.

Розничная торговля осуществляется двумя отделами, находящимися по адресам:

- ул. Школьная дом 45. Розничный магазин, также присутствует сервисный отдел;

- ул. Гагарина дом 5. Розничный магазин.

- Сервисный отдел:

Занимается обслуживанием и ремонтом гарантийной и не гарантийной техники для физических и юридических лиц.

Оптовая торговля осуществляется отделом, находящимся по адресу: ул. Советская дом 13.

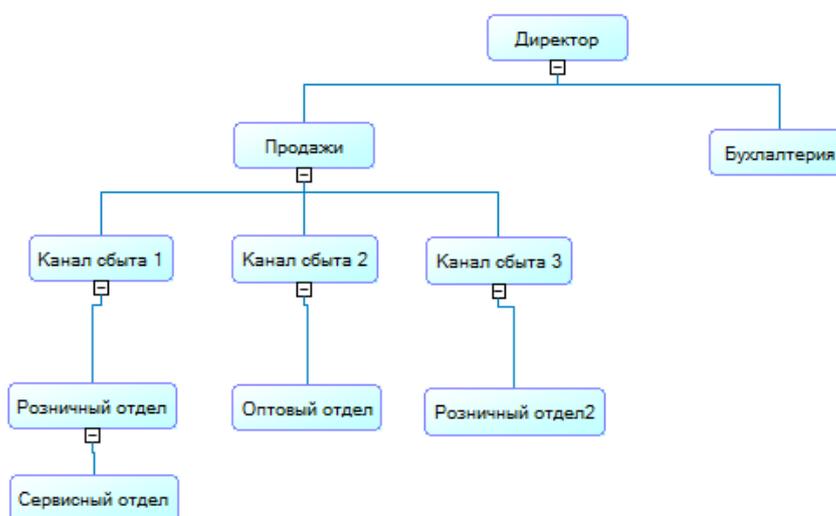


Рисунок 1.1 – Структура организации

Чтобы более тщательно разобраться с рассматриваемой областью и определить, что может понадобиться от разрабатываемой системы, надо дать описание существующих в компании бизнес-процессов [4], подлежащих автоматизации.

На представленной структуре (рис. 1.1), выделим отдел «Продажи» и опишем функции, подходящие для автоматизации.

В структуре данного отдела существуют два отдела сбыта товара – корпоративный и розничный. Корпоративный отдел работает только с оптовыми покупателями. Оптовые покупатели, путем непосредственного присутствия в корпоративном отделе или при помощи факса, формируют заказ. Сформированный заказ отгружается со склада после передачи счета на отгрузку товара. Для наглядности на рисунках 1.2-1.4 изображен процесс бизнес взаимодействия.

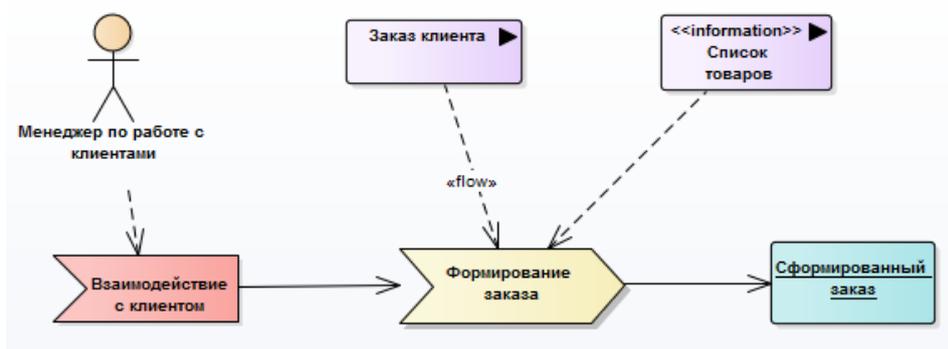


Рисунок 1.2 – Бизнес-процесс «Формирование заказа клиента»

Данный процесс отображает получение от клиента заказа. Полученный результат бизнес-процесса говорит о сформированном клиентом заказе.

На рисунке 1.3 представлен бизнес-процесс расчет с клиентом.

Данный бизнес-процесс предполагает формирование счета на оплату для клиента.

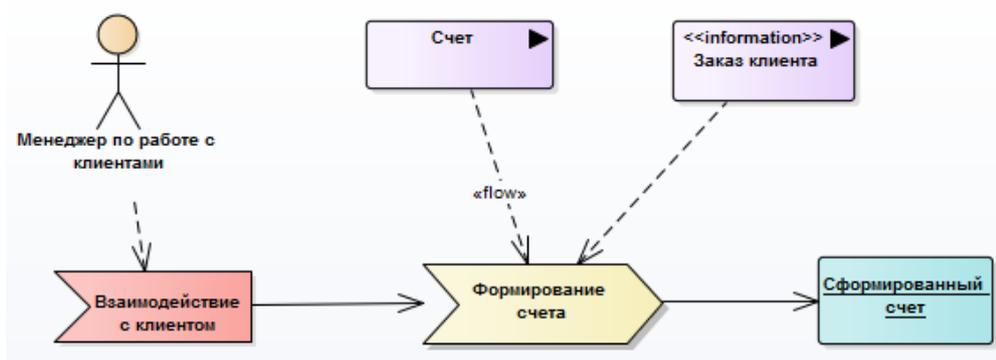


Рисунок 1.3 – Бизнес-процесс «Формирование счета клиента»

Суть представленного бизнес-процесса (рис. 1.4) является формирование заказа для поставки товара. Результатом данного процесса выступает отображение выбранного канала сбыта и данные о заказанном товаре.

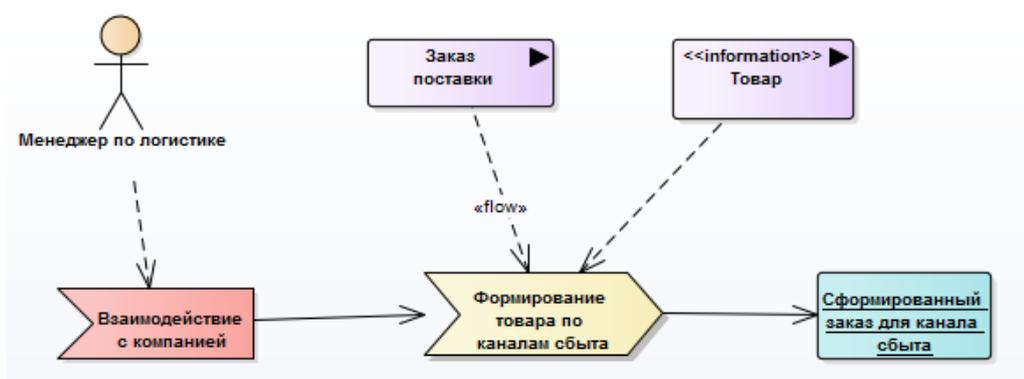


Рисунок 1.4 – Бизнес-процесс «Формирование заказа для поставки»

Из описанного выше способа взаимодействия оптовых покупателей с корпоративным отделом, слабым звеном являются следующие моменты:

- при высокой загрузке сотрудника, не всегда можно вовремя оповестить клиента о готовности заказа,
- при использовании факса тратится большое количество бумаги и времени сотрудника.

Кроме этого, не всегда получается оповестить клиента по разным причинам. В связи с этим клиентам приходится самим связываться с отделом, чтобы узнать о готовности заказа.

В розничных отделах ведется только учет проданных товаров. Учет покупателей не ведется. Документы о проданных товарах заполняются вручную, что в свою очередь увеличивает трудоемкость, с одной стороны. С другой стороны, поскольку данные о продажах поступают с запозданием, то невозможно получить актуальные данные о состоянии бизнеса.

Поэтому для обеспечения эффективности работы компании необходимо разработать информационную систему для анализа бизнес деятельности, с учетом особенностей описанных отделов.

1.2 Анализ существующей технологии анализа бизнес-деятельности

В связи с тем, что для анализа бизнеса-деятельности необходимо использование информационных средств для обработки информации, применение которых позволит более эффективно управлять принимаемыми решениями. Для решения данной задачи необходимо использовать средства

автоматизации. Чтобы более рационально подойти к процессу выбора инструментально решения, необходимо провести анализ существующей технологии, применяемой в компании.

Для анализа рассмотрим бизнес-процесс формирования отчета. Для решения данной задачи необходимо построить модель (AS-IS – «как есть»), которая показывает, как происходит выполнение процесса в данный момент (рис. 1.5).

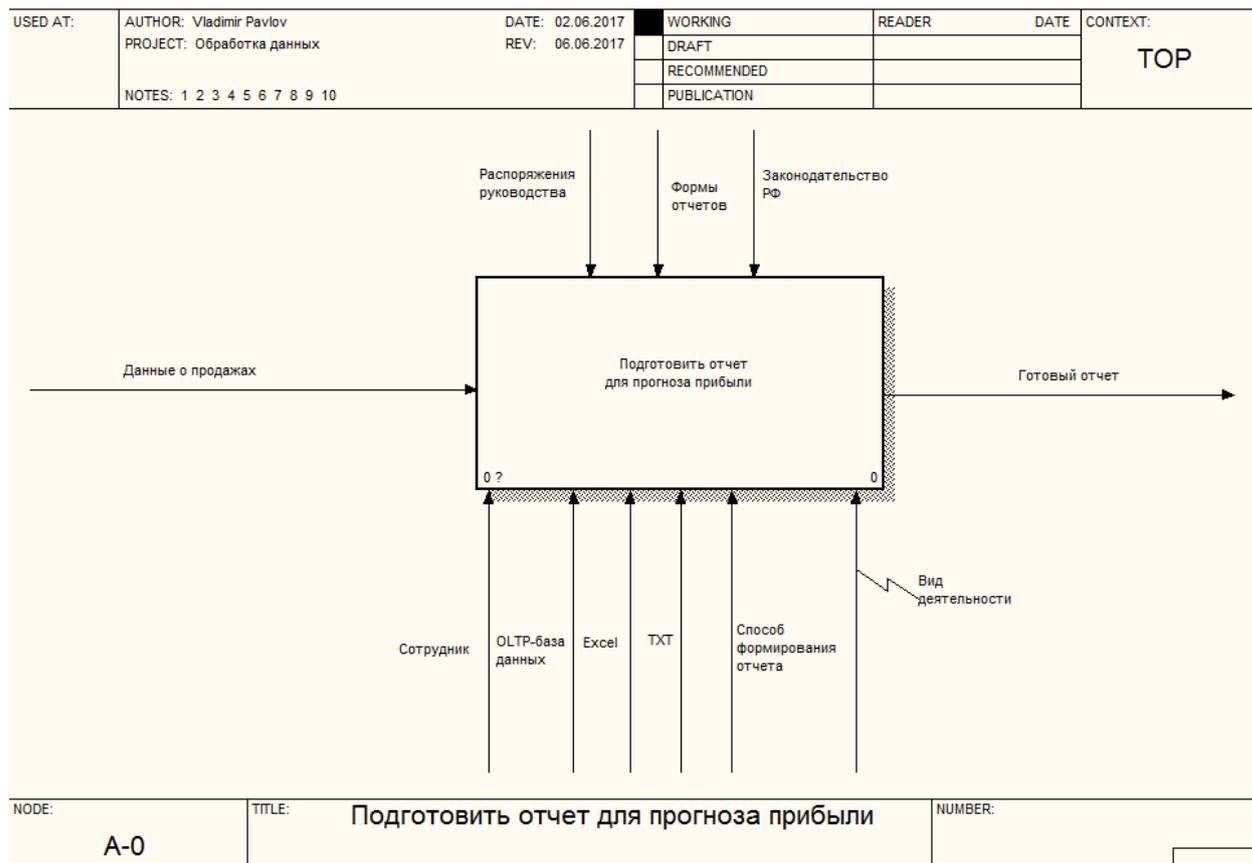


Рисунок 1.5 - Контекстная диаграмма процесса обработки AS-IS

Данная диаграмма представляет собой общее описание процесса, из которого выделяются следующее:

Входы (слева):

- данные о продажах.

Выходы (справа):

- готовый отчет.

Управление (сверху):

- распоряжения руководства;
- формы отчетов;
- законодательство.

Механизмы:

- Excel;
- TXT-документ;
- OLTP-база;
- сотрудник;
- вид деятельности;
- способ формирования отчета.

Данные для анализа собираются из различных источников, с ними работают сотрудники, отвечающие за определенный вид деятельности, поэтому представленные отчеты не имеют комплексной информации. Отчеты были ежедневными, составлялись вручную и требовали дополнительных трудозатрат. Поэтому часто имели неактуальную и противоречивую информацию, которая не в полной мере отображала состояние дел компании.

После описания контекстной диаграммы производится декомпозиция – процесс разбиения системы на подсистемы для достижения нужной степени подробности.

Процесс подготовки отчета (рис. 1.6) подразделяется на:

- подготовку информации для отчета. На данном этапе происходит процесс поиска и сбора информации сотрудником в соответствии с распоряжениями руководства;
- формирование отчета.

На данном этапе сотрудником, ответственным за свой вид деятельности, из источников (OLTP, Excel, TXT), происходит формирование отчета. Способ формирования отчета в данном случае зависит от конкретного отдела (ручной, автоматический).

На этапе сотрудником проводится проверка правильности данных и публикация отчета (рис. 1.6).

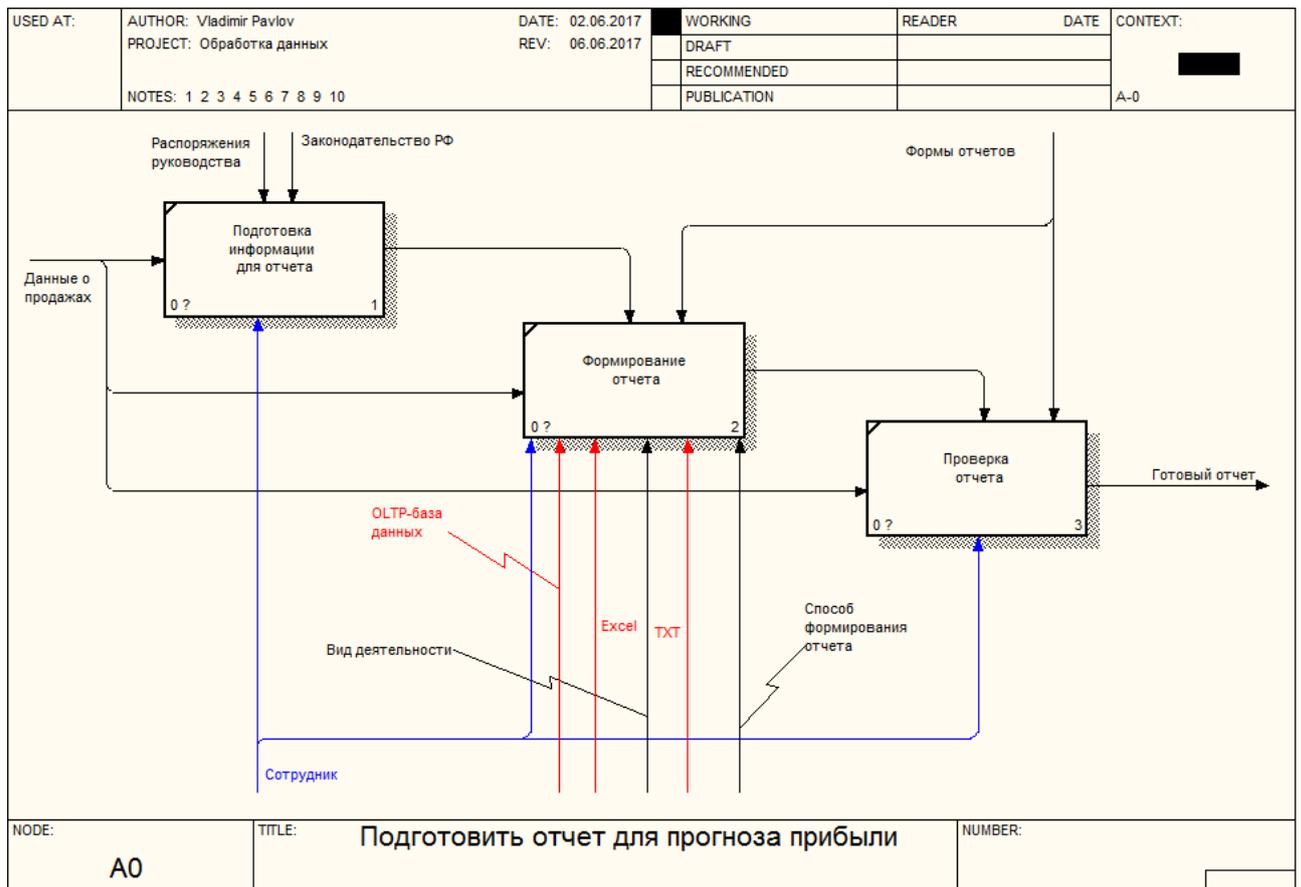


Рисунок 1.6 – Диаграмма декомпозиции IDEF0 AS-IS

Для анализа данных в организации используется технология, представленная компанией Microsoft – Microsoft Excel. В ней реализован достаточный набор возможностей для построения аналитических решений.

С помощью Excel можно обращаться к OLAP-кубам и получать двух и трехмерные сечения на рабочих листах книг Excel в виде сводных таблиц. В Excel включен драйвер источника данных и клиентское программное обеспечение, необходимое для доступа к базам данных, созданных в Microsoft SQL Server OLAP Services, Microsoft SQL Server OLAP Services и Microsoft SQL Server Analysis Services. Источник данных предоставляет доступ ко всем данным в базе данных OLAP или в автономном файле куба. После создания источника данных OLAP на его основе можно создавать отчеты и возвращать данные OLAP в Excel в виде отчета сводной таблицы или отчета сводной диаграммы, или в функции листа, преобразованной из отчета сводной таблицы.

Таким образом, при анализе модели «Подготовить отчет для прогноза прибыли» были выявлены недостатки, связанные с источниками данных и скоростью обработки информации. То есть документы необходимо искать в разных местах вручную. Поскольку информация может храниться на бумаге, для подготовки отчета её необходимо заносить в компьютер.

Проведенный анализ предметной области выявил основные задачи, которые необходимо автоматизировать при разработке информационной системы компании. Рассмотрение существующих решений по информатизации показало, что целесообразно провести индивидуальную разработку информационной системы этого предприятия.

1.3 Формирование требований к новой информационной технологии анализа бизнес деятельности

Для достижения деловых целей при построении систем хранения данных важно определить функциональные и нефункциональные требования. Функциональные требования определяют, что делает система. Нефункциональные требования ограничивают архитектуру. Данные требования нужны при дальнейшем моделировании системы.

Сбор требований – это процесс, включающий мероприятия, необходимые для создания и утверждения документа, содержащего спецификацию системных требований [4].

Процесс сбора требований необходим для создания информационной системы с правильными рабочими характеристиками. Требование - это характеристика или условие, которому должна удовлетворять система [5].

Для удобного просмотра и доступа собранные на разных этапах данные необходимо тщательно документировать, это процедура называется спецификация требований.

Спецификация требований представляет собой документирование требований в структурированном, удобном и доступном всем формате. Она помогает определить программную область разрабатываемого проекта с точки

зрения поведения, процесса и данных. Спецификация требований используется при разработке, тестировании, приемке проекта и связанных с проектом функциях [5].

Сбор требований осуществлялся на основе использования метода интервьюирования.

В процессе интервьюирования были выявлены следующие требования, которым должна отвечать система:

- система должна охватывать основные бизнес-процессы предприятия;
- система должна иметь понятное назначение функций и наглядный результат обработки информации;
- система должна иметь возможность наращивания в программной части;
- пользователи системы ХД должны быть в состоянии получить доступ к приложениям ХД;
- ХД не должно быть доступно за пределами компании;
- ХД должно быть доступным 24 часа в сутки. Простой не более одного часа в месяц;
- если при выполнении ETL-операций произошел сбой, данные в ХД не должны быть повреждены.

Полный список требований представлен в Приложении А.

1.4 Анализ решений для разработки хранилища данных и OLAP

Для того чтобы разобраться зачем понадобилось создание хранилища данных, проведем некие аналогии с реляционной базой данных (рис. 1.7):

- OLTP-системы предназначены для повседневной работы пользователей, а хранилища данных нужны для принятия решений;
- данные во время работы пользователей подвержены постоянным изменениям, тогда как в ХД более стабильно, и данные в него загружаются с регулярной периодичностью (например: ежедневно, ежемесячно);

- хранилища данных очень редко строятся на пустом месте, поэтому чаще всего оперативная база является источником данных для её пополнения, в том числе за счет внешних источников.

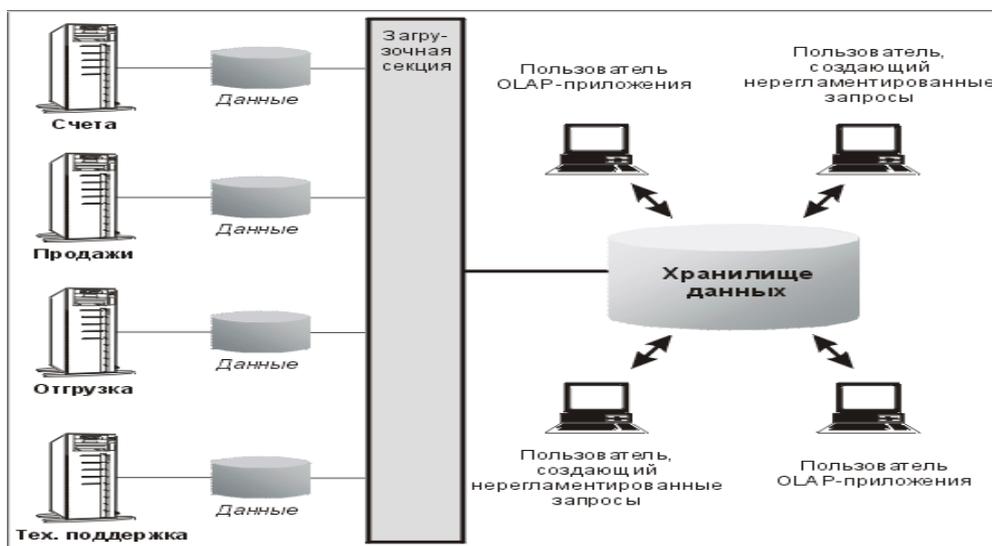


Рисунок 1.7 – Типичное хранилище данных

Хранилище данных предполагает использование следующих компонентов:

- оперативных источников информации;
- средств переноса и трансформации данных;
- реляционного хранилища;
- метаданных;
- OLAP-хранилища;
- средств анализа данных.

Назначение данных компонентов в следующем. Оперативные данные собранные из разных источников очищаются, интегрируются и помещаются в реляционное хранилище. Эти данные доступны для анализа с помощью средств построения отчетов. Затем данные подготавливаются к переносу и трансформации с использованием средств OLAP-анализа. Данные подготовленные подобным образом можно разместить в специальную базу OLAP (OLAP-куб, срез) или оставить в хранилище.

Концепция OLAP представляет собой многомерную базу данных, так называемый куб (рис. 1.8), с помощью которого пользователь может развернуть интересующее его измерение. Для создания таких многомерных баз используются специальные средства, входящие в состав MS SQL Server, Oracle Warehouse Builder и др.

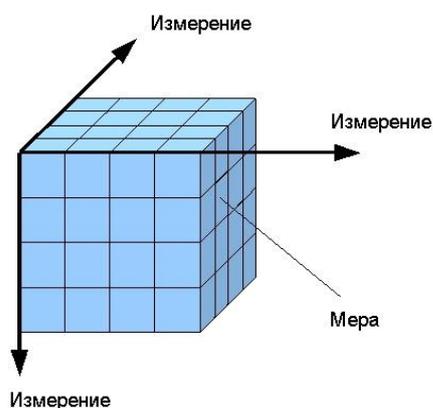


Рисунок 1.8 – OLAP-концепция

На рисунке видно направления именуемые измерения и мера. Измерения располагаются на осях столбцов и строк и представляют собой таблицы аналогичные модели данных, по этим таблицам производится фильтрация данных. Меры представляют собой числовые данные, которые показывают агрегирующие функции.

В концепции OLAP существует три типа OLAP [3,6]:

- MOLAP (Multidimensional OLAP) – OLAP со многими измерениями;
- ROLAP (Relational OLAP) – реляционный OLAP;
- HOLAP (Hybrid OLAP).

В MOLAP-системах используются данные, которые хранятся в многомерном формате базы данных в детальном виде. Использование такого решения позволяет управлять данными как многомерным массивом. Однако в представленном подходе многомерная база становится избыточной, поскольку для работы используются реляционные данные. Поэтому применение MOLAP предпочтительно для обработки небольших наборов данных.

В ROLAP-системах используются данные взятые из транзакционной базы данных для представления в таблицах или многомерном виде извлеченные путем преобразования с помощью метаданных. Агрегаты хранятся в той же базе данных в созданных для этого служебных таблицах. Такой подход позволяет обеспечить хранение большого количества данных. Кроме того, это обеспечивает более актуальное отображение информации для принятия решений. [3]

При создании информационной системы надо определить ряд задач необходимых для решения и выбрать подходящий под описанные условия программный продукт.

Первая задача заключалась в предоставлении пользователям возможности работы не только с определенными запросами, построенными на основе отчетов, но и иметь возможность выбора различных вариантов взаимодействия с помощью фильтров.

Второй задачей было решение для обеспечения высокой работоспособности анализируемых данных.

Таблица 1- Сравнение OLAP-серверов

OLAP-сервер	Разработчик	ROLAP	MOLAP	HOLAP
Microsoft Analysis Services	Microsoft	+	+	+
Essbase	Oracle	+	+	+
TM1	IBM	-	+	-
Mondrian OLAP-Server	Pentaho	+	-	-
Palo	Jedox	-	+	-
Oracle OLAP-Option	Oracle	+	+	+
Microstrategy OLAP Services	Microstrategy	+	+	-
SAS OLAP Server	SAS Institute	+	+	+

Наиболее предпочтительным является решение, предоставленное компанией Microsoft, поскольку ранее обработка данных осуществлялась с помощью программного продукта Microsoft Office и MS Access. Также принято решение использовать подход ROLAP.

1.5 Разработка архитектуры информационной системы для анализа данных

Все системы информационно-аналитического взаимодействия построены на концепции хранилищ данных. В общем виде архитектура такой системы описывается тремя слоями:

- извлечение, преобразование и загрузка;
- хранение данных;
- анализ и обработка данных.

Технология работы системы состоит в следующем. Данные поступающие из различных транзакционных систем, внешних источников, проверяются, обрабатываются, преобразовываются и загружаются в хранилище данных. Пользователи с помощью специализированных инструментов получают доступ к необходимым данным, для выполнения различных операций по построению отчетов, анализа данных, конструирования табличных и графических представлений.

При анализе структуры компании ООО «Экран» были выявлены следующие недостатки архитектуры хранения данных:

1. В качестве хранилища данных используется реляционная база данных. Такой подход является малоэффективным и менее производительным для работы с нетипичными данными – документы, изображения. Поэтому перед развертыванием системы необходимо учесть требования собранные на этапе формирования;

2. Данные поступающие в хранилище являются значимые, проверенные, непротиворечивые и исторически целостные. При этом надо учесть, что

данным необходимо дополнительное время на процесс осуществления извлечения и преобразования.

3. Доступ к данным на этапе анализа предполагает использование «клиент-серверной» и локальный вид взаимодействия с данными с помощью «клиент-приложение-сервер» технологиями.

На рисунке 1.9 представлено архитектурное решение для информационной системы.

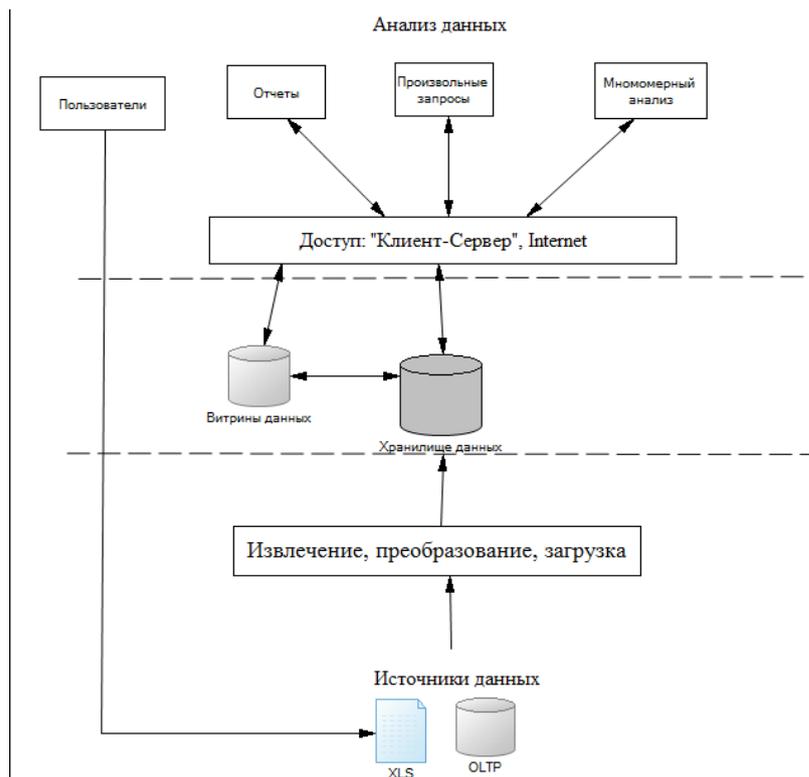


Рисунок 1.9 – Архитектура разрабатываемой информационной системы

Таким образом, в разрабатываемой архитектуре информационной системы предполагается устранение недостатков присутствующих до этого в компании.

1.6 Концептуальное моделирование данных информационной системы для анализа данных

Концептуальная модель данных (CDM) позволяет анализировать концептуальную структуру информационных систем с целью выявления базовых сущностей, их атрибутов и взаимосвязей между ними. [8]

Создание CDM позволяет сосредоточиться на проблемах проектирования, поскольку работа ведется над концептуальным представлением бизнес задачи и позволяет увидеть картину в целом (рис. 1.10).

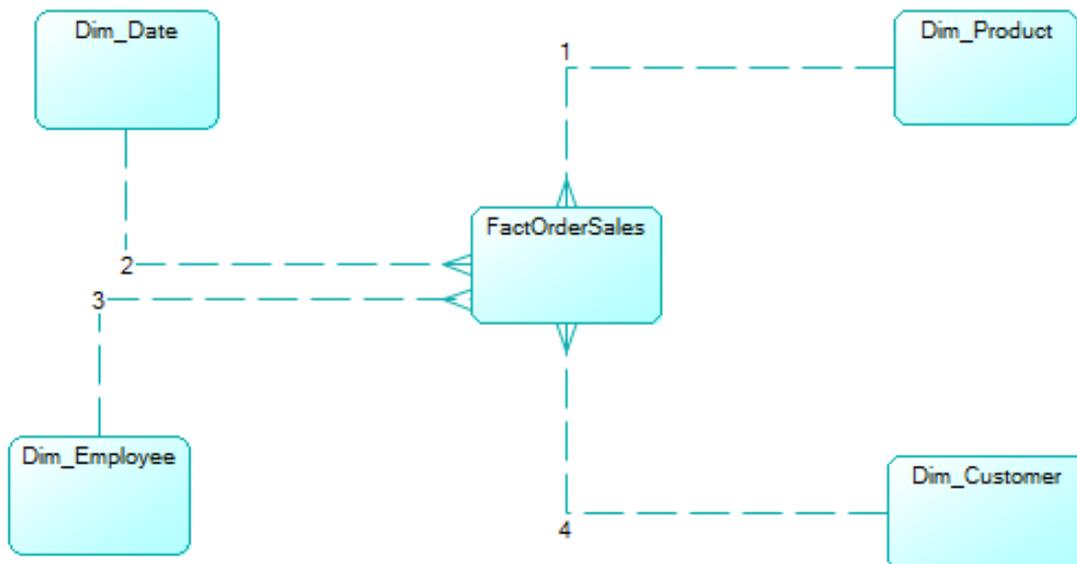


Рисунок 1.10 – Концептуальная модель ХД

Логическое проектирование – процесс создания модели будущей информационной системы, без учета физических аспектов и реального типа СУБД. При моделировании нужно опираться на концептуальную модель будущего ХД, определенного на этапе сбора.

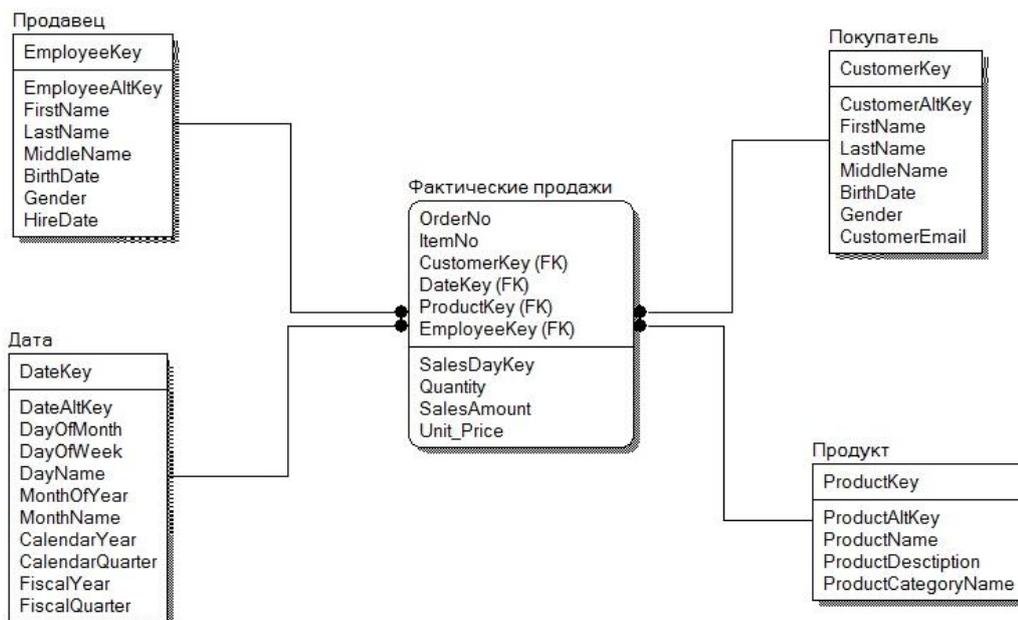


Рисунок 1.11 – Логическая модель ХД

При проектировании базы данных хранилища было создано несколько таблиц для хранения используемой в системе информации.

Вывод по главе

На основании анализа бизнес-процессов были разработаны требования к информационной системе. Был проведен анализ средств для разработки и выбрано решение для реализации проекта. Была разработана концептуальная и логическая модель. На основании логической модели данных для Microsoft SQL Server 2012 разработана физическая модель данных.

2 РЕАЛИЗАЦИЯ ПРИЛОЖЕНИЯ ДЛЯ СИСТЕМЫ АНАЛИЗА ДАННЫХ

2.1 Обоснование выбора платформы для создания информационной системы

Сегодня существует много различных BI-систем представленные на российском рынке – это продукты компаний **Oracle, IBM, Microsoft, SAS** и другие.

Для разработки информационной системы будет использоваться обеспечение представленной компанией Microsoft – SQL Server 2012, поэтому проведем анализ положительных и отрицательных сторон использования представленного продукта.

Таблица 2 - Сравнение BI-решений

Вендор	Достоинства	Недостатки
IBM Cognos BI	Анализ взаимосвязей показателей, Оповещения об изменениях показателей, Построение отчетов, обновляющихся в реальном времени, Четкое распределение прав доступа к отчетности	Высокая стоимость, Малая распространённость, Сложность администрирования
Oracle BI	Интерактивные отчёты, информационные показатели, иерархии, метрики – любой сложности, пространственная аналитика	Высокая стоимость, сложность администрирования,
Microsoft BI	Низкая стоимость, Интеграция с продуктами компании, Высокая производительность, масштабируемость, отказоустойчивость и безопасность	Меньшие возможности по сравнению с Oracle и IBM

Главным преимуществом решения от компании Microsoft является более низкая цена и интеграция со своими программными продуктами поэтому,

применение данного программного продукта позволяет выполнять решения по управлению и анализу данных.

Используемое решение является проприетарным, поэтому перед применением данного продукта, необходимо определиться с лицензией на него.

Компания Microsoft выпускает SQL Server в следующих редакциях:

- Enterprise – для больших ХД;
- Business Intelligence – для расширенной корпоративной и самостоятельной аналитики;
- Standard – для простых баз данных, отчетности и аналитики.

Для SQL Server 2012 предлагается две модели лицензирования: "Сервер+CAL" и "На ядро процессора".

THREE MAIN EDITIONS	LICENSING MODEL	
	Server + Client Access License (CAL)	Core-based
Enterprise		✓
Business Intelligence	✓	
Standard	✓	✓

Рисунок 2.1 – Модели лицензирования

Лицензирование по модели «Сервер + клиентская лицензия» предполагает:

- лицензирование каждого отдельного сервера с помощью лицензии SQL Server.
- лицензирование каждого пользователя или устройства, обращающегося к серверу, с использованием лицензии SQL Server CAL. Лицензии SQL Server 2012 CAL могут использоваться для доступа к любой базе данных SQL Server, лицензированной на сервере, независимо от платформы или редакции.

Этой модели соответствуют редакции Business Intelligence и Standard. Исходя из потребностей компании и запросов заказчика выберем редакцию Standard с моделью лицензирования «Сервер + клиентская лицензия».

SQL Server представляет решение, которое делает работу пользователей компании более безопасной, надежной и продуктивной. Различные инструменты SQL Server 2012 предоставляют разработчикам и пользователям дополнительные возможности, тем самым упрощая работу с разработкой решений для анализа данных и аналитических приложений. Благодаря этому, а также большому набору компонент представленное решение повышает эффективность хранения данных в различных компаниях.

Платформа SQL Server 2012 предоставляет компаниям следующие преимущества:

- благодаря интеграции с Microsoft Office, SQL Server 2012 предоставляет пользователям компании важную, своевременную информацию, тем самым увеличивая продуктивность работы.

- SQL Server 2012 упрощает разработку, внедрение и управление аналитическими приложениями.

BI-решение компании Microsoft основывается на масштабируемой платформе, предназначенной для интеграции данных, построения хранилищ данных, анализа накопленных данных и построения отчетов. В составе SQL Server 2012 входят такие службы как: Integration Services, Analysis Services, Reporting Services.

Таблица 3 - Компоненты BI-решения Microsoft

Компонент	Описание
SQL Server Database Engine	Масштабируемая, высокопроизводительная СУБД, способная хранить большие объемы данных, образующиеся в результате консолидации данных в единое хранилище для анализа и построения отчетов.
SQL Server Integration Services	Платформа для выполнения операций извлечения, преобразования и загрузки, которая обеспечивает заполнение ХД и его синхронизацию с данными из различных источников, которые используются бизнес-приложениями организации.

Продолжение таблицы 3

SQL Server Analysis Services	Обеспечивает возможность построения OLAP-решений, включая возможность расчета ключевых индикаторов производительности (KPI). Применяется также для построения data-mining -решений, которые используют специализированные алгоритмы для выявления трендов и зависимостей в бизнес-данных.
SQL Server Reporting Services	Инструментарий построения отчетов, предназначенный для создания, публикации и распространения детализированных бизнес-отчетов, как для внутренних, так и для внешних целей.

Таким образом, SQL Server является наиболее подходящим средством для решения поставленной задачи.

2.2 Разработка физической модели хранилища данных

Перед непосредственной реализацией нужно определиться с выбором модели применяемой для ХД и OLAP. Исходя из условий, полученных на этапе формулирования требований, целесообразно использовать ROLAP со схемой реализации многомерного представления данных с помощью реляционных таблиц типа «звезда» (рис. 2.2).

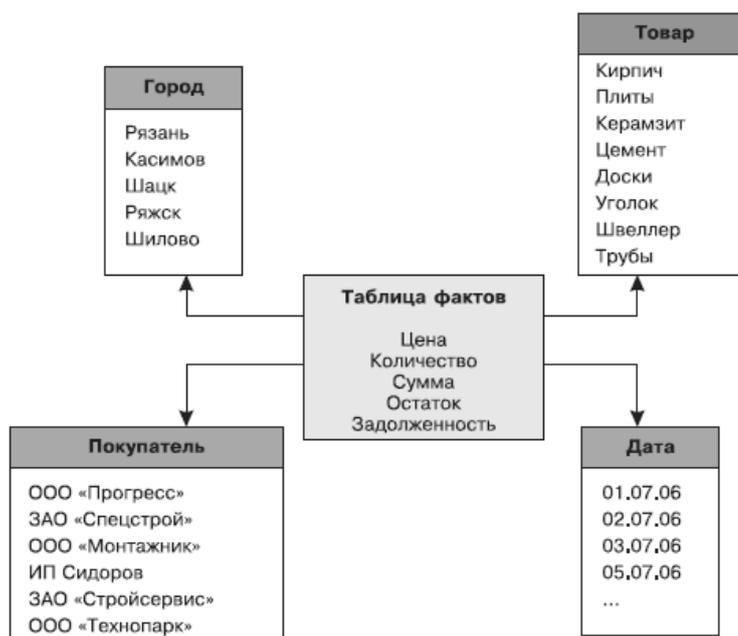


Рисунок 2.2 – Схема «звезда»

Данная схема представляет собой звезду, в центре которой располагается таблица, называемая таблица фактов, которая окружена таблицами именуемыми измерениями. Одна схема «звезда» охватывает одну часть бизнес-деятельности. Таблица фактов соединена с таблицами измерений внешними ключами.

В процессе физического проектирования базы данных в среде MS SQL Server 2012 была создана база данных хранилища EkranDW, состоящая из файлов данных разбитых на файловые группы ekrandw.mdf.

Листинг SQL-скриптов, создающих информационные структуры сервера баз данных хранилища приведен в приложении Б.

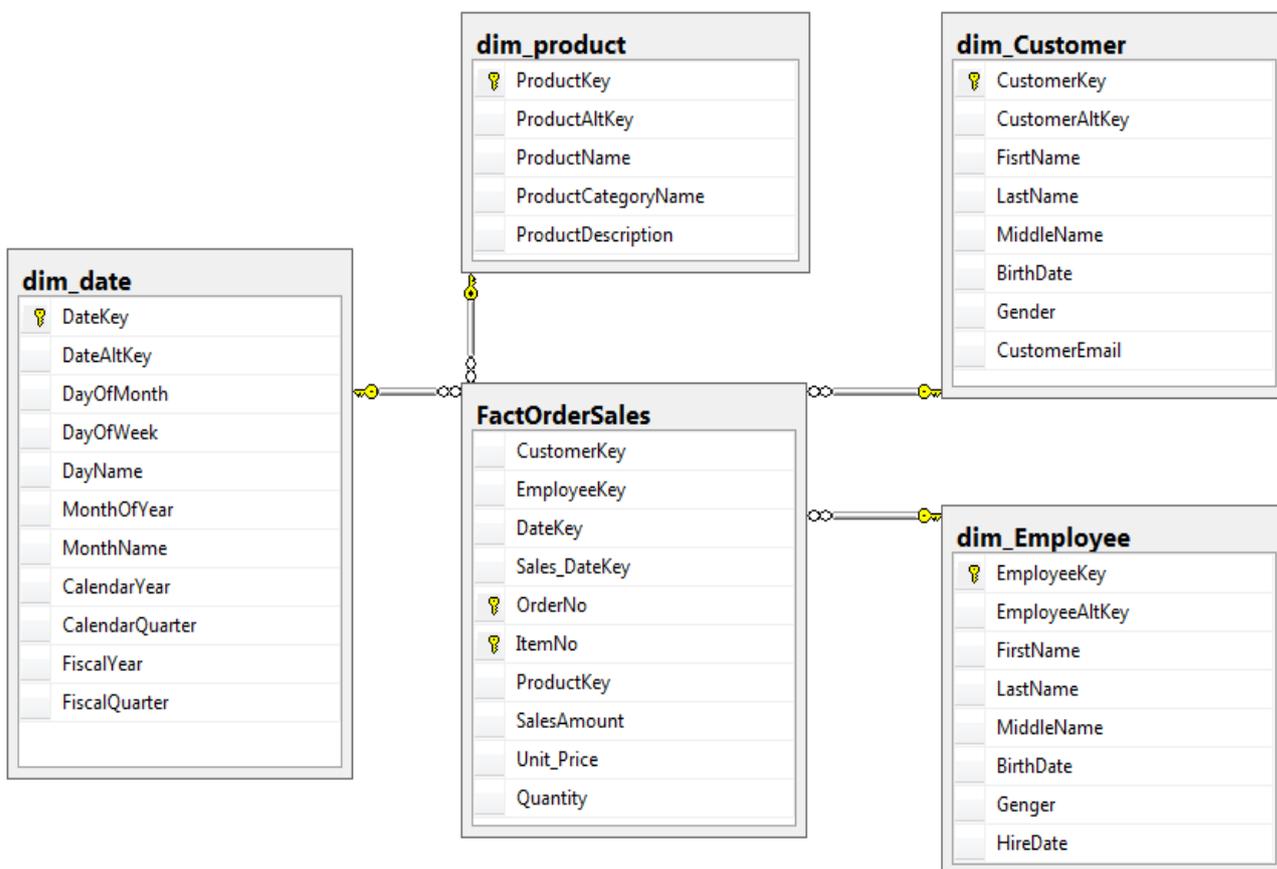


Рисунок 2.3 - Физическая модель данных

Хранилище состоит из таблиц исходной базы данных связанных между собой и именованных запросов к базе данных. Спецификация таблиц в Приложении В.

2.2.1 Определение таблиц измерений

Измерения представляют конкретные данные для мер и определяют по чему будут фильтроваться данные. Терминология ХД в реляционной модели представляет любой столбец как *атрибут*.

Таблицы измерений содержат неизменяемые либо редко изменяемые данные. Эти данные представляют собой по одной записи для каждого члена нижнего уровня иерархии в измерении. Таблицы измерений также содержат как минимум одно описательное поле и, как правило, целочисленное ключевое поле для однозначной идентификации члена измерения.

Каждая таблица измерений должна находиться в отношении «один ко многим» с таблицей фактов.

Скорость роста таблиц измерений должна быть незначительной по сравнению со скоростью роста таблицы фактов; например, добавление новой записи в таблицу измерений, характеризующую товары, производится только при появлении нового товара, не продававшегося ранее.

В таблицу-измерений, крайне полезно, добавлять вычисляемые столбцы. Это связано с тем, что при постоянно растущих данных, замедляется работа ХД. Следовательно, добавление вычисляемых столбцов оказывает положительное влияние на производительность ХД. Вычисляемый столбец – это виртуальный столбец таблицы. Значение столбца определяется выражением.

Таблицы измерений, полученные в результате проектирования, представлены в приложении В.

2.2.2 Определение таблицы фактов

Таблица фактов, содержит фактическую информацию, быстро растет и является самой большой таблицей в ХД, и представляет собой таблицу данных, которые будут анализироваться. Таблицы фактов нужно создавать после создания таблиц-измерений. Это связано с тем, что связи в таблице фактов определяются как «один-ко-многим», следовательно, должна существовать родительская таблица для создания внешнего ключа.

Количество таблиц измерений связанных с таблицей фактов определяет степень гранулярности. Присутствуют два вида гранулярности. Первый вид называется *размерностью*. Она определяется тем, что если отсутствует связанное с таблицей фактов измерение, то становится невозможным получить отчет. Другой вид определяет, что с помощью внешнего ключа измерения можно определить, на каком уровне в таблице фактов находится измерение, так как измерения имеют разную гранулярность.

Таблица фактов является главной таблицей для анализа пользователями, поэтому при её проектировании нужно также учитывать все возможные условия, по которым будут фильтроваться и агрегироваться запросы и не только те, которые были получены на этапе опроса. Для таблицы фактов нужно определить роли и уровни, определенные на этапе сбора требований Ролями в нашем случае, являются клиент и продукт. Уровни определяются количественными величинами или мерами, это цена товара, количество, общая стоимостью товара.

Таблица фактов, объединяется с таблицами измерений с помощью уникального составного ключа с помощью первичных ключей таблиц измерений. Таблица фактов включает в себя несколько числовых полей для получения агрегатных данных. Поскольку таблица фактов, как правило, содержит много записей, хранить в ней текстовые описания неудобно, поэтому их чаще всего помещают в таблицы измерений.

В таблице фактов нет никаких сведений о том, как группировать записи при вычислении агрегатных данных. Например, в ней есть идентификаторы продуктов или клиентов, но отсутствует информация о том, к какой категории относится данный продукт или в каком городе находится данный клиент. Эти сведения, в дальнейшем используемые для построения иерархий в измерениях куба, содержатся в таблицах измерений.

2.3 Реализация хранилища данных

Физическая модель данных создается на основе логической модели. Отличием данного способа является использование конкретной СУБД. Для непосредственной реализации проекта будет использоваться программное обеспечение компании Microsoft SQL Server 2012.

Начальная стадия создания ХД предполагает создание физической модели данных с помощью средств языка Transact-SQL. Реализация модели в данном случае ничем не отличается от реализации баз данных, однако присутствует ряд особенностей.

Первая особенность состоит в том, что ХД содержит не оперативную информацию, то есть данные не обрабатываются в режиме реального времени, отсутствует необходимость в создании журнала транзакций. Исходя из этого, можно выбрать простой режим восстановления. Простой режим восстановления характеризуется поддержкой минимальных требований для хранения журнала транзакций на диске и автоматическим освобождением места в журнале транзакций.

```
CREATE DATABASE EkranDW
ON PRIMARY
(
  NAME = N'EkranDW_fg1', FILENAME = N'C:\sqlserver\disk\data1\EkranDW_fg1.mdf',
  SIZE = 5 GB, FILEGROWTH= 1 GB),
FILEGROUP EkranDW_fg2
(NAME='EkranDW_fg2', FILENAME = N'C:\sqlserver\disk\data2\EkranDW_fg2.ndf',
SIZE = 5 GB, FILEGROWTH= 1 GB),
FILEGROUP EkranDW_fg3
(NAME='EkranDW_fg3', FILENAME = N'C:\sqlserver\disk\data3\EkranDW_fg3.ndf',
SIZE = 5 GB, FILEGROWTH= 1 GB)
LOG ON (name = 'EkranDW_log'
, filename = N'C:\sqlserver\disk\data1\EkranDW_log.ldf'
, size = 1 GB, filegrowth = 512 MB)
go
alter database EkranDW set recovery simple
go
alter database EkranDW set auto_shrink off
go
alter database EkranDW set auto_create_statistics on
go
alter database EkranDW set auto_update_statistics on
go
|
```

Рисунок 2.4 – Скрипт создания базы хранилища

После создания базы данных хранилища, теперь нужно создать таблицы измерений и фактов, спроектированных в логической модели.

```
SQLQuery2.sql - D...PC\Darkoomen (52))* X
CREATE TABLE dim_date
(
    DateKey int not null,
    DateAltKey datetime not null,
    [DayOfMonth] int not null,
    [DayOfWeek] int not null,
    [DayName] nvarchar(15) not null,
    MonthOfYear int not null,
    [MonthName] nvarchar(15) not null,
    CalendarYear int not null,
    CalendarQuarter int not null,
    FiscalYear int not null,
    FiscalQuarter int not null,
    CONSTRAINT PK_DateKey PRIMARY KEY CLUSTERED (DateKey)
ON EkranDW_fg2
) ON EkranDW_fg2
GO
CREATE UNIQUE INDEX DateAltKey
ON dim_date (DateAltKey)
ON EkranDW_fg2
go
```

Рисунок 2.5 – Скрипт Создание измерений

В скрипте создания таблицы выделяем следующие моменты:

- при создании первичного ключа (DateKey) указан его кластерный индекс. Такой подход позволяет улучшать производительность запросов.
- использование файловых групп позволяет определить, где физически хранится кластеризованный индекс первичного ключа и физическая таблица.

Аналогичным образом создаются все остальные таблицы измерений и фактов.

Таким образом, подготавливается пустое хранилище для последующих манипуляций с загрузкой данных.

Загрузка данных в ХД может выполняться ручным и программным способом. При ручном способе выполняется скрипт языка T-SQL.

```
SQLQuery1.sql - D...PC\Darkoomen (52))* X
INSERT INTO dbo.Products
(ProductKey, ProductAltKey, ProductName, ProductDescription, ProductCategoryName)
SELECT P.ProductKey, P.ProductAltKey, P.ProductName, P.ProductDescription, P.ProductCategoryName
FROM EKRAM.dbo.Product AS P
```

Рисунок 2.6 – Ручной вариант загрузки данных

Программная реализация выполняется с помощью объектно-ориентированного подхода, реализованного в SSDT (SQL Server Data Tools) входящего в пакет SSIS.

Рабочее окно программы (рис. 2.7) выглядит следующим образом.

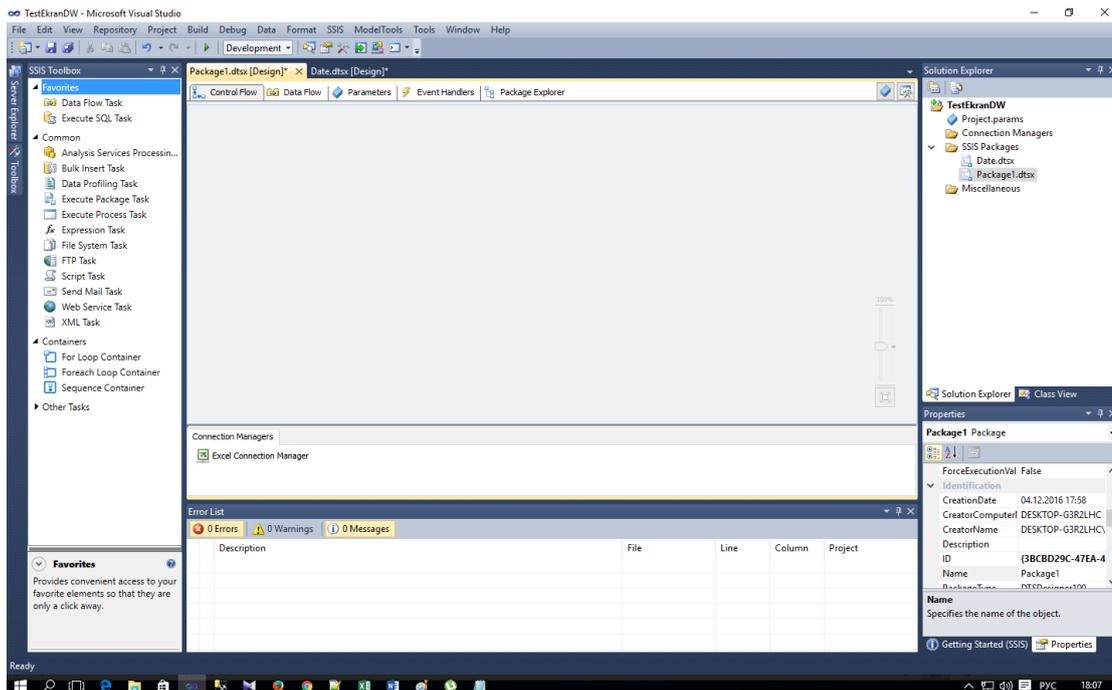


Рисунок 2.7 – Рабочее окно SSIS

Для создания ХД, в наших исходных данных, кроме базы данных, также присутствуют разнообразные «плоские» файлы (Excel). С помощью средств SSIS будет продемонстрирован процесс создания пакета для переноса этих данных в подготовленное ХД.

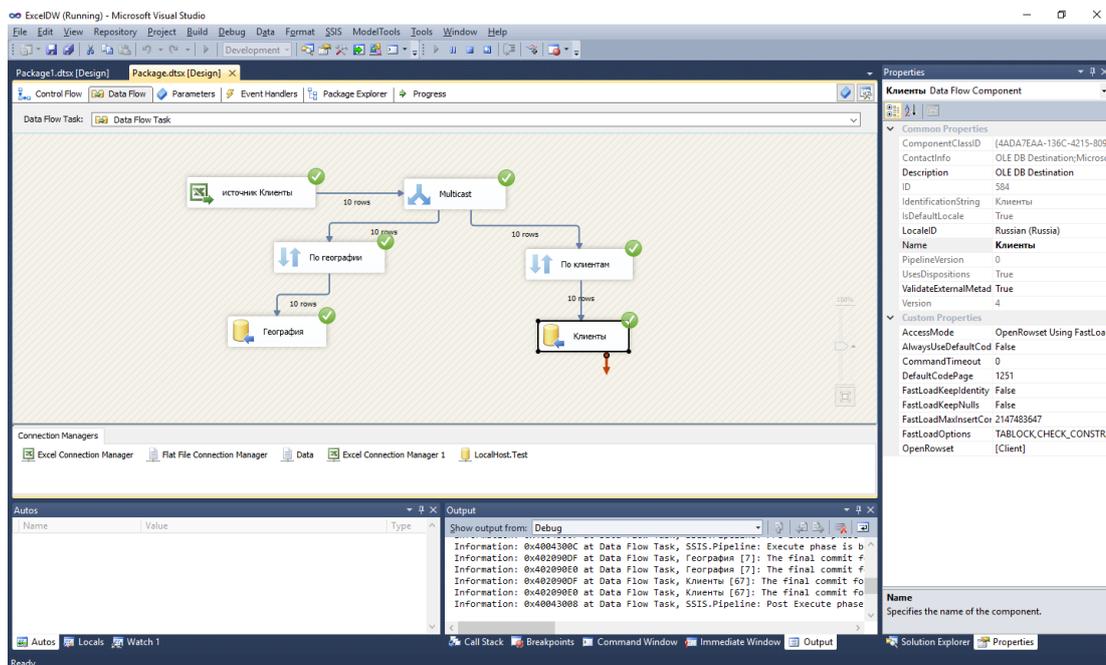


Рисунок 2.8 – Успешный результат передачи данных

После создания базы данных хранилища, на основании этих данных, средствами SSAS была создана структура OLAP-куба, для последующего тестирования.

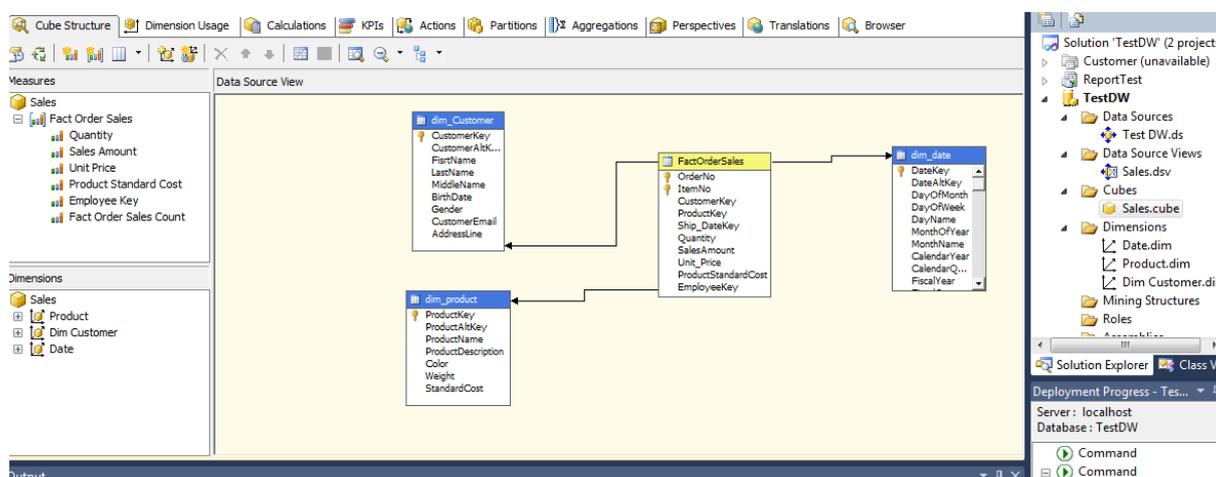


Рисунок 2.9 – Структура OLAP-куба

2.4 Тестирование приложения

После создания и наполнения хранилища данных необходимо провести тестирование работоспособности.

Тестирование – это процесс анализа программного средства и сопутствующей документации с целью выявления дефектов и повышения качества продукта [17]. Тестирование выполняется путем моделирования входных данных в разрабатываемой системе в условиях, приближенных к реальным.

Тестирование ХД будет состоять из этапа проверки SQL-запросов, а также создание тестового куба с помощью средств MS SQL Server Analysis Services (SSAS) и тестирование созданных кубов в клиентских приложениях.

При тестировании учитывалось:

- проверка аппаратного устройства;
- проверка совместимости;
- проверка работоспособности SQL-запросов и средств создания отчетов;
- проверка производительности запросов.

Результаты тестирования показаны в таблице 4 и Приложении Г.

Таблица 4 – Результаты тестирования

Название теста	Проверяемый случай	Результаты теста
Проверка аппаратного устройства	Взаимодействие с пакетами прикладных программ	Взаимодействие осуществляется
Проверка совместимости	Windows 7	Совместима
	Windows 8	Совместима
Проверка работоспособности	Windows 7	Работоспособна
	Windows 8	Работоспособна
Проверка производительности	Скорость загрузки сформированного отчета в браузере	Время загрузки – 2 сек. (SQL Server Profiler)

При разработке аналитических требований к кубу данных можно воспользоваться специализированным языком запросов к многомерным структурам данных – MDX, который применяется IT-специалистами при разработке отчетов и работе с кубом.

```
SELECT NON EMPTY { [Measures].[Quantity] } ON COLUMNS, NON EMPTY { ([Dim Product].[Product Name].[Product Name].ALLMEMBERS ) } DIMENSION PROPERTIES MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS FROM ( SELECT ( { [Dim Product].[Product Alt Key].&[TV], [Dim Product].[Product Alt Key].&[TV-UHD] } ) ON COLUMNS FROM [Sales] ) WHERE ( [Dim Product].[Product Alt Key].CurrentMember ) CELL PROPERTIES VALUE, BACK_COLOR, FORE_COLOR, FORMATTED_VALUE, FORMAT_STRING, FONT_NAME, FONT_SIZE, FONT_FLAGS
```

Product Name	Quantity
LG	281
Panasonic	639
Sony	631

Рисунок 2.10 – Выполнение MDX-запроса

Данный запрос показывает, сколько товаров определенного типа было реализовано.

После развертывания файла проекта на сервере дальнейшее взаимодействие с кубом происходит с помощью специального браузера. Web-отчеты на SQL-сервере, а доступ к ним осуществляется по средствам этого браузера (рис 2.11).

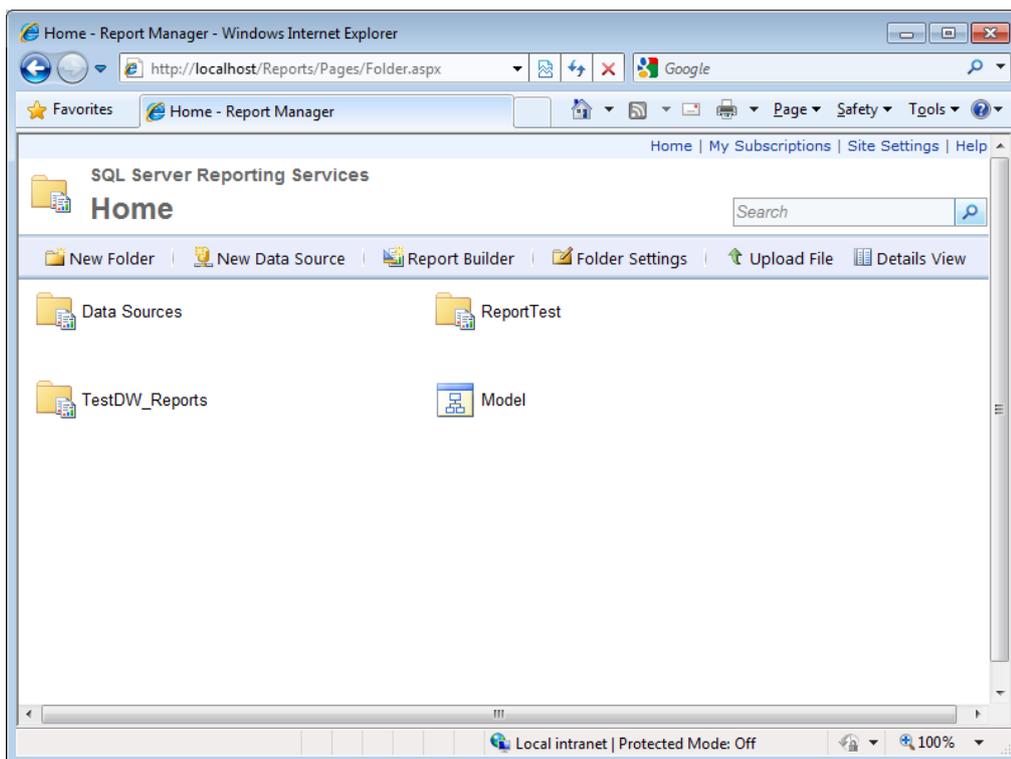


Рисунок 2.11 – Работа с кубом через браузер

С помощью SQL Server Reporting Services были созданы следующие отчеты, представленные в Приложении Г:

- количество продаж группы товаров по годам;
- сколько потрачено денег мужчинами и женщинами за январь 2006 года;

Несмотря на то, что отчеты, реализуемыми средствами Reporting Services, позволяют работать только с отчетами, развернутыми на специальном сервере, данное решение успешно выполняет функции необходимые для работоспособности компании. Для тестирования на стороне клиента используется Microsoft Office Excel.

Вывод по главе

Во второй главе было приведено обоснование выбранного средства разработки, описаны этапы разработки хранилища данных и основных методов его тестирования. На основании логической модели данных была спроектировано хранилище данных, включающее в себя раздел бизнес-деятельности. Данное хранилище было проверено в тестовом режиме путем

выполнения MDX-запросов. Продемонстрирована реализация взаимодействия с системами отчетов в развернутом на сервере хранилище данных.

3. ОЦЕНКА ЭКОНОМИЧЕСКОЙ ЭФФЕКТИВНОСТИ ПРОЕКТА

3.1 Расчет показателей экономической эффективности проекта

Проект автоматизации должен быть рассмотрен с точки зрения эффективности его внедрения. Поскольку решение должно предусматривать снижение затрат и сроков на получение и обработку данных и улучшение качества полученной информации. Что в свою очередь повышает эффективность принимаемых на основе этой информации решений.

Экономическая эффективность проекта складывается из двух составляющих: косвенного эффекта и прямого эффекта:

$$\mathcal{E} = \mathcal{E}_k + \mathcal{E}_n, \quad (3.1)$$

где \mathcal{E} - экономическая эффективность проекта.

Косвенный эффект определяется улучшением качественных характеристик. Прямой эффект отражает сокращение трудовых (3.2-3.4) и стоимостных (3.5-3.7) затрат на обработку данных. На рассматриваемом предприятии при существующей бизнес-организации, операции взаимодействия с клиентами осуществлялись способами, в зависимости от того где находится необходимый отдел. Поэтому, для выяснения разницы до и после внедрения проекта, надо провести анализ показателей необходимых для выполнения данных операций. Формулы, по которым будут рассчитываться показатели трудовых и стоимостных затрат, для определения прямого эффекта описаны ниже.

К трудовым показателям относятся:

- Абсолютное снижение трудовых затрат

$$\Delta T = T_0 - T_1, \quad (3.2)$$

где ΔT – абсолютное снижение трудовых затрат,

T_0 – трудовые затраты на обработку информации по базовому варианту,

T_1 - трудовые затраты на обработку информации по проектируемому варианту.

- коэффициент относительного снижения трудовых затрат

$$K_T = \Delta T / T_0, \quad (3.3)$$

где K_T - коэффициент относительного снижения трудовых затрат,

ΔT - абсолютное снижение трудовых затрат,

T_0 - трудовые затраты на обработку информации по базовому варианту.

– индекс снижения трудовых затрат или повышения производительности труда

$$Y_T = T_0 / T_1, \quad (3.4)$$

где Y_T – индекс снижения трудовых затрат или повышения производительности труда,

T_0 - трудовые затраты на обработку информации по базовому варианту,

T_1 - трудовые затраты на обработку информации по проектируемому варианту.

К стоимостным показателям относятся:

– абсолютное снижение стоимости затрат

$$\Delta C = C_0 - C_1, \quad (3.5)$$

где ΔC - абсолютное снижение стоимостных затрат

C_0 - стоимостные затраты на обработку информации по базовому варианту,

C_1 - стоимостные затраты на обработку информации по проектируемому варианту.

– коэффициент относительного снижения стоимостных затрат

$$K_c = \Delta C / C_0, \quad (3.6)$$

где K_c - коэффициент относительного снижения стоимостных затрат,

ΔC - абсолютное снижение стоимостных затрат,

C_0 - стоимостные затраты на обработку информации по базовому варианту.

– индекс снижения стоимости затрат

$$Y_C = C_0 / C_1, \quad (3.7)$$

где Y_C - индекс снижения стоимостных затрат,

C_0 - стоимостные затраты на обработку информации по базовому варианту,

C_1 - стоимостные затраты на обработку информации по проектируемому варианту.

3.2 Расчет базовых показателей экономической эффективности проекта

Расчет трудовых и стоимостных затрат при базовом варианте обработки информации:

Трудовые затраты и стоимостные затраты:

На основании имеющихся данных определим трудоемкость обработки информации в базовом варианте:

$$T_0 = T_{01} + T_{02} + \dots + T_n, \quad (3.8)$$

где $T_{01} \dots T_n$ – годовая трудоемкость обработки процессов;

$$T_{01} \dots T_n = Q_p / T_{\text{опр}},$$

где Q_p – объем работы в год;

$T_{\text{опр}}$ – общая норма выработки;

$$T_{01} = 8000 / 40 = 200 \text{ часов.}$$

$$T_0 = 200 + 50 + 25 + 125 + 50 + 100 + 100 + 60 + 140 = 850 \text{ час.}$$

Годовые стоимостные затраты в базовом варианте

$$C_0 = C_{01} + C_{02} + \dots + C_n, \quad (3.9)$$

где $C_{01} \dots C_n$ – годовая стоимость обработки процессов;

$$C_{01} \dots C_n = D_{\text{обр}} + D_{\text{рм}} + D_{\text{об}}, \quad (3.10)$$

где $D_{\text{обр}}$ = расходы на заработную плату;

$D_{\text{рм}}$ = расходы на расходные материалы;

$D_{\text{об}}$ = расходы на обслуживание оборудования;

где $D_{\text{обр}} = Z_{\text{ср}} * T_{01} \dots T_n$,

где $Z_{\text{ср}}$ = среднечасовая зарплата.

$$C_{01} = (170 * 200) + 1500 + 6000 = 41500$$

$$C_0 = 41500 + 14500 + 5000 + 23750 + 8500 + 17000 + 15000 + 12500 + 21500 = 159250$$

Годовые стоимостные затраты в базовом варианте составляют 159250 рублей.

На основании данных полученном на предприятии, и проанализировав их были подсчитаны количество исполненных процессов. Для большей наглядности данные занесены в таблицу. (Таблица 5).

3.3 Показатели эффективности от внедрения проекта

Разработанная система должна автоматизировать процесс обработки информации, поэтому участие менеджера в данном процессе отпадает. Для поддержания работоспособности системы потребуется участие системного администратора, который будет загружать необходимые данные.

Данные в систему будут загружаться 7 дней в неделю, максимальное время загрузки информации 10 минут, плановые работы по обслуживанию составляют 24 часа. Общее время для данного процесса равняется 84 часа.

В стоимостные затраты входят следующие данные:

- Средняя заработная плата системного администратора составляет около 150 рублей в час.

$$C_1 = T_1 * Z_{CP}, \quad (3.11)$$

где Z_{CP} - средняя заработная плата системного администратора в час,

$$C_1 = 84 * 150 = 12600 \text{ рублей.}$$

Годовая стоимость обработки информации при проектируемом варианте 12600 рублей.

Абсолютный показатель снижения трудовых затрат:

$$\Delta T = 850 - 84 = 766.$$

Коэффициент снижения трудовых затрат:

$$K_T = 766 / 850 * 100\% = 90,1\%.$$

Абсолютный показатель и коэффициент снижения трудовых затрат составили 766 часов или 90,1%. Это означает, что в случае внедрения проекта произойдет снижение трудовых затрат.

Таблица 5 - Базовые показатели

№	Наименование операций	Оборудование	Ед.изм.	Объем работы в год	Норма Выработки, часов	Трудоемкость	Среднечасовая зарплата	Годовые расходы на расходные материалы	Годовые расходы на поддержание оборудования	Стоимостные затраты
1	Прием сведений о клиентах	Вручную, ЭВМ	операция	8000	40	200	170	1500	6000	41500
2	Прием сведений о поставщиках	Вручную	операция	2000	40	50	150	2000	5000	14500
3	Прием сведений о персонале	Вручную	операция	1000	40	25	160	1000		5000
4	Формирование учетных данных по договорам	Вручную	операция	5000	40	125	150	5000		23750
5	Поиск записей	Вручную, ЭВМ	операция	2000	40	50	150	1000		8500
6	Сортировка	Вручную,	операция	3000	30	100	160	1000		17000

	записей	ЭВМ								
7	Редактирование записей	Вручную, ЭВМ	операция	5000	50	100	130	1000	1000	15000
8	Анализ данных по договорам	Вручную	документ	3000	50	60	200	500		12500
9	Отчеты	Вручную	документ	7000	50	140	150	500		21500
						850		13500	12000	159250

Индекс снижения трудовых затрат:

$$Y_T = 850/84 = 10,1.$$

Индекс изменения затрат по трудовым затратам составил 10,1 это говорит о том, что трудовые затраты до автоматизации в 10,1 раза превышают трудовые затраты после автоматизации.

Абсолютный показатель снижения стоимостных затрат:

$$\Delta C = 159250 - 12600 = 146650 \text{ рублей.}$$

Коэффициент снижения стоимостных затрат:

$$K_C = 146650/159250 * 100\% = 92,1\%.$$

Абсолютный показатель и коэффициент снижения стоимостных затрат составили 146650 рублей и 92,1% соответственно. Это означает, что внедрение системы привело к снижению показателей относительно базового варианта.

Индекс снижения стоимостных затрат:

$$Y_C = 159250/12600 = 12,6.$$

Индекс изменения стоимостных затрат составил 12,6. Это означает превышение стоимостных затрат до автоматизации в 12,6 раз стоимостных затрат после автоматизации.

Для наглядности показатели эффективности занесены в таблицу 6.

Таблица 6 – Показатели эффективности проекта

Показатель	Затраты		Абсолютное измерение затрат	Коэффициент изменения затрат	Индекс изменения затрат
	Базовый вариант	Проектный вариант			
Трудоемкость	T_0	T_1	$\Delta T = T_0 - T_1$	$K_T = \Delta T / T_0 * 100\%$	$Y_T = T_0 / T_1$
	850	84	766	90,1%	10,1
Стоимость	C_0	C_1	$\Delta C = C_0 - C_1$	$K_C = \Delta C / C_0 * 100\%$	$Y_C = C_0 / C_1$
	159250	12600	146650	92,1%	12,6

Кроме рассмотренных показателей также целесообразно рассчитать срок окупаемости проекта по созданию системы:

$$T_{OK} = K_{\Pi} / \Delta C, \quad (3.12)$$

где T_{OK} – срок окупаемости затрат на внедрение программы,

K_{Π} – затраты на внедрение новой программы,

ΔC – абсолютное снижение стоимостных затрат.

К затратам на внедрение новой программы относятся следующие показатели: количество человек участвующих в разработке, число дней отведенное на создание проекта, среднюю плату разработчиков, время работы. Также к затратам можно отнести стоимость приобретения лицензии на внедрение системы.

$$K_{\Pi} = C_{РАЗ} + C_{ВН}, \quad (3.13)$$

где $C_{РАЗ}$ – стоимость разработки,

$C_{ВН}$ – стоимость внедрения.

Так для разработки потребуется 2 разработчика. Срок на разработку 2 месяца (60 дней), с учетом 8-ми часового рабочего дня. Средняя заработная плата 180 рублей. Стоимость лицензии SQL Server 2012 составляет 45000 рублей.

$$C_{РАЗ} = 2 * 30 * 8 * 180 = 172800.$$

$$K_{\Pi} = 172800 + 45000 = 217800.$$

$$T_{OK} = 217800 / 146650 = 1,49.$$

Срок окупаемости затрат на внедрение информационной системы составит около 2 месяцев, это означает, что после истечения 2 месяцев проект окупит вложения и начнет приносить прибыль.

Вывод по главе

Проведённая проверка эффективности создания проекта показала хороший уровень окупаемости в случае его принятия, путем улучшения показателя трудоспособности в связи с улучшением автоматизации деятельности.

ЗАКЛЮЧЕНИЕ

В рамках данного проекта с помощью описанных выше средств была разработана информационная система, представляющая собой хранилище данных, реализованное в соответствии с требованиями к системе. Также была изучена литература по проектированию хранилищ данных.

Разработанная автоматизированная информационная система, охватывающая основные бизнес процессы предприятия была внедрена и используется в организации.

В качестве дальнейшего развития данной системы можно предложить расширение разработанного хранилища данных по мере необходимости в построении других отчетов.

Созданная система - это эффективное средство обработки данных, внедрение которой позволит сократить трудовые затраты на 90,1%, а стоимостные затраты на 146650 рублей в год.

Использование системы позволит оценить динамику продаж за разные временные периоды для различных категорий товаров, отделов продаж, отдельных товаров.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

Научная и методическая литература

1. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining. – СПб.: БХВ-Петербург, 2004. – 336 с
2. Celko Joe. Analytics and OLAP in SQL. Morgan Kaufmann, 2006.
3. Нартова А. PowerDesigner 15. Моделирование данных, 2012. - 480с.
4. Вигерс К. Битти Д. Разработка требований к программному обеспечению, издание третье, 2014. – 737с.
5. Adrienne H. Slaughter. OLAP. 2004
6. Избачков, Ю.С. Информационные системы. Учебник для вузов / Ю.С. Избачков, В.Н. Петров. 2-е изд. – СПб.: Питер, 2005. – 739 с
7. Полубояров В.В. Использование MS SQL Server Analysis Services 2008 для построения хранилищ данных – 2008. -487с.
8. Туманов В.Е. Маклаков С.В. Проектирование реляционных хранилищ данных – 2007. – 333с.
9. Агальцов, В. П. Базы данных : Учебник для вузов / В. П. Агальцов. - М. : ФОРУМ, 2011. - 270 с.
10. Сарка Д. Microsoft SQL Server 2012. Реализация хранилищ данных. – 2014. – 816с
11. Жуковский О.И. Хранилища данных. Учебное пособие. – 2015. – 165с.
12. Советов Б. Я. Архитектура информационных систем : Учебник для вузов / Б. Я. Советов, А. И. Водяхо, В. А. Дубенецкий. - М. : Академия, 2012. - 288 с.
13. Беленькая, М. Н. Администрирование в информационных системах : Учебник для вузов / М. Н. Беленькая, С. Т. Малиновский, Н. В. Яковенко. - М. : Горячая линия - Телеком, 2011. - 400 с.
14. Белов, В. В. Проектирование информационных систем : Учебник для вузов / В. В. Белов, В. И. Чистякова. - М. : Академия, 2013. - 352 с.

15. Брещенко А. Базы данных / А. Брещенко-М: LAP Lambert Academic Publishing, 2011.- 404с.

Электронные ресурсы

16. Microsoft SQL Server 2005. Обзор продукта. [Электронный документ]

17. Описание языка UML. [Электронный документ] (<https://ru.wikipedia.org/wiki/UML>).

18. Бизнес-правила в среде разработки и моделирования. [Электронный документ] (<http://www.interface.ru/home.asp?artId=1752>)

19. Трофимов, С. Определение требований к программному обеспечению. [Электронный документ] (<http://www.caseclub.ru/articles/treb.html>).

20. Nigel Pendse. “What is OLAP? An analysis of what the often misused OLAP term is supposed to mean” [Электронный документ]

21. Nigel Pendse. “OLAP architectures” [Электронный документ]

ПРИЛОЖЕНИЕ А

Спецификация требований к программному обеспечению

Назначение

Эта спецификация требований описывает функциональные и нефункциональные требования для информационной системы предприятия. Этот документ предназначен для команды, которая будет реализовывать, и проверять корректность работы системы.

ОБЩЕЕ ОПИСАНИЕ

Описание продукта

Информационная система предприятия – это новая, которая система позволит сотрудникам:

- уменьшить площадь хранимой информации за счет использования информационных технологий;
- предоставить возможность для эффективного управления анализом бизнес-процессов

Доступ к разработанной информационной системы может осуществляться только тем категориям пользователей, которые связаны с реализацией бизнес-процессов предприятия.

Общие ограничения

Операционная среда-1. Минимальные требования к операционной системе – Microsoft Windows 7.

Ограничения реализации-1. Система должна использовать базу под управлением СУБД MS SQL Server 2012.

Ограничения реализации-2. Приложение должно быть использовано как клиент-серверная система, в которой модули, управляющие внешними устройствами, являются серверами автоматизации.

Специфические требования

Таблица А.1 - Функциональные требования

Требования	Описание
Бизнес-анализ продаж	Система должна позволять пользователю выбирать данные о клиенте и должна предоставить возможность проанализировать "продажи продуктов" по магазинам, отделам, продуктам.
Сроки хранения информации в хранилище.	Хранилище данных будет хранить данные до пяти лет онлайн и пять лет офлайн. Оффлайновые данные должны быть в состоянии быть полученными доступ онлайн с двумя днями уведомления.
Просмотр данных управляющими магазинами	Система позволит управляющим магазином рассмотреть данные просто их собственного магазина. Это связано с тем, что каждый управляющий магазином ответственен за различные магазины..
Анализ	Система позволит пользователям CRM проанализировать результаты кампании, рассматривая следующие меры для кампании
Сохранение для анализа	В аналитических целях CRM хранилище данных сохранит старых клиентов

Требования к производительности

Отклик системы не должен превышать 30 секунд с момента передачи запроса.

Требования к безопасности

– функции системы становятся доступными пользователю только после его аутентификации в системе;

- регистрация новых пользователей в системе осуществляется только администратором системы;
- хранилище данных не должно быть доступно вне сети компании;
- хранилище хранится на центральном сервере.

Атрибуты качества ПО

Доступность-1. Система должна быть доступна 24 часа.

Надежность-1. Система не должна нарушать целостность данных.

ПРИЛОЖЕНИЕ Б

Скрипт создания базы данных хранилища

```
use master
go
IF DB_ID ('EkranDW') IS NOT NULL
DROP DATABASE EkranDW
GO
CREATE DATABASE EkranDW
ON PRIMARY
(
    NAME = N'EkranDW_fg1',FILENAME =
N'C:\sqlserver\disk\data1\EkranDW_fg1.mdf',
    SIZE = 5 GB, FILEGROWTH= 1 GB),
    FILEGROUP EkranDW_fg2
    (NAME='EkranDW_fg2', FILENAME =
N'C:\sqlserver\disk\data2\EkranDW_fg2.ndf',
    SIZE = 5 GB, FILEGROWTH= 1 GB),
    FILEGROUP EkranDW_fg3
    (NAME='EkranDW_fg3', FILENAME =
N'C:\sqlserver\disk\data3\EkranDW_fg3.ndf',
    SIZE = 5 GB, FILEGROWTH= 1 GB)
LOG ON (name = 'EkranDW_log'
, filename = N'C:\sqlserver\disk\data1\EkranDW_log.ldf'
, size = 1 GB, filegrowth = 512 MB)
go
alter database EkranDW set recovery simple
go
alter database EkranDW set auto_shrink off
go
alter database EkranDW set auto_create_statistics on
```

```

go
alter database EkranDW set auto_update_statistics on
go
Создание таблиц фактов и измерений.
CREATE TABLE dim_date
(
DateKey int not null,
DateAltKey datetime not null,
[DayOfMonth] int not null,
[DayOfWeek] int not null,
[DayName] nvarchar(15) not null,
MonthOfYear int not null,
[MonthName] nvarchar(15) not null,
CalendarYear int not null,
CalendarQuarter int not null,
FiscalYear int not null,
FiscalQuarter int not null,
CONSTRAINT PK_DateKey PRIMARY KEY CLUSTERED (DateKey)
ON EkranDW_fg2
) ON EkranDW_fg2
GO
CREATE UNIQUE INDEX DateAltKey
ON dim_date (DateAltKey)
ON EkranDW_fg2
go

CREATE TABLE dim_product
(
ProductKey int not null,
ProductAltKey nvarchar(10) not null,

```

```
ProductName nvarchar(50),
ProductDescription nvarchar(100),
ProductCategoryName nvarchar(50),
CONSTRAINT PK_dim_Product PRIMARY KEY CLUSTERED
```

(ProductKey)

```
ON EkranDW_fg2
```

```
) ON EkranDW_fg2
```

```
GO
```

```
CREATE TABLE dim_Customer
```

```
(
```

```
CustomerKey int not null,
```

```
CustomerAltKey nvarchar(15) not null,
```

```
FisrtName nvarchar(50),
```

```
LastName nvarchar(50),
```

```
MiddleName nvarchar(50),
```

```
BirthDate datetime,
```

```
Gender nvarchar(1),
```

```
CustomerEmail nvarchar(50),
```

```
AddressLine nvarchar(120),
```

```
GeographyKey int not null,
```

```
CONSTRAINT PK_dim_Customer PRIMARY KEY CLUSTERED
```

(CustomerKey)

```
ON EkranDW_fg2
```

```
) ON EkranDW_fg2
```

```
GO
```

```
CREATE TABLE dim_Employee
```

```
(
```

```
EmployeeKey int not null,
```

```
EmployeeAltKey nvarchar(15),
```

```
FirstName nvarchar(50),
```

```

LastName nvarchar(50),
MiddleName nvarchar(50),
BirthDate datetime,
Genger nvarchar(1),
HireDate date,
CONSTRAINT PK_Employee PRIMARY KEY CLUSTERED
(EmployeeKey)
ON EkranDW_fg3
) ON EkranDW_fg3
GO
CREATE TABLE FactOrderSales
(
CustomerKey int not null,
ProductKey int not null,
EmployeeKey int not null,
Sales_DateKey int not null,
OrderNo int not null,
ItemNo int not null,
Quantity int,
SalesAmount money,
Unit_Price money,
CONSTRAINT PK_FactOrderSales PRIMARY KEY CLUSTERED
(OrderNo,ItemNo)
ON EkranDW_fg3,
CONSTRAINT FK_FactOrderSales_dim_Customer
FOREIGN KEY(CustomerKey) REFERENCES
dim_Customer(CustomerKey),
CONSTRAINT FK_FactOrderSales_dim_Employee
FOREIGN KEY (EmployeeKey) REFERENCES
dim_Employee(EmployeeKey),

```

```
CONSTRAINT FK_FactOrderSales_dim_Product
FOREIGN KEY (ProductKey) REFERENCES dim_Product (ProductKey),
CONSTRAINT FK_FactOrderSales_dim_Date
FOREIGN KEY (Sales_DateKey) REFERENCES dim_date (DateKey)
) On EkranDW_fg3
```

ПРИЛОЖЕНИЕ В

Таблицы хранилища данных

Таблица В.1 – Атрибуты управляющих таблиц в проектируемом приложении.

Имя	Тип	Значение
Атрибуты таблицы Dim_Customer		
CustomerKey (PK)	int	Идентификатор строк клиента
CustomerAltKey	nvarchar(15)	Альтернативный код клиента
FirstName	nvarchar	Имя
LastName	nvarchar	Фамилия
MiddleName	nvarchar	Отчество
BirthDate	date	Дата рождения
Gender	nchar	Пол клиента
CustomerEmail	nvarchar	Email адрес
AddressLine	nvarchar	Адрес клиента
Атрибуты таблицы Dim_Employee		
EmployeeKey (PK)	int	Идентификатор строк сотрудника
FirstName	nvarchar	Имя сотрудника
LastName	nvarchar	Фамилия сотрудника
MiddleName	nvarchar	Отчество сотрудника
BirthDate	date	Дата рождения
Gender	nchar	Пол сотрудника
HireDate	date	Дата найма сотрудника
Атрибуты таблицы Dim_Product		
ProductKey (PK)	int	Идентификатор строк продукта
ProductAltKey	nvarchar	Альтернативный номер продукта
ProductName	nvarchar	Название продукта
ProductDescription	nvarchar	Описание продукта
Color	nvarchar	Цвет продукта
Weight	float	Вес
StandardCost	money	Цена продукта
Атрибуты таблицы Dim_Date		
DateKey (PK)	int	Идентификатор строк

		даты
DateAltKey	date	Альтернативная запись даты
DayOfMonth	int	Номер дня месяца
DayOfWeek	int	Номер дня недели
DayName	nvarchar	Название дня недели
MonthOfYear	int	Номер месяца года
MonthName	nvarchar	Название месяца
CalendarYear	int	Календарный год
CalendarQuarter	int	Календарный квартал
FiscalYear	int	Финансовый год
FiscalQuarter	int	Финансовый квартал
Атрибуты таблицы FactOrderSales		
CustomerKey (FK)	int	
ProductKey (FK)	int	
Ship_DateKey (FK)	int	
EmployeeKey (FK)	int	
OrderNo (PK)	nvarchar	
ItemNo (PK)	int	
Quantity	int	Количество
SalesAmount	money	Цена
Unit_Price	money	Цена за единицу товара
ProductStandardCost	money	Цена закупки товара

ПРИЛОЖЕНИЕ Г

Тестирование приложения

Тестирование производительности и работоспособности запросов в MSSAS.

EventClass	EventSubclass	TextData	ConnectionID	NTUserName	ApplicationName	IntegerData	StartTime	CurrentTime	Duration	DatabaseName
Existing Connection			24	Admin			2017-04-17 19:17:52...	2017-04-18 16:50:56...		
Existing Connection			86	Admin	Microsoft Vi...		2017-04-18 16:46:58...	2017-04-18 16:50:56...		
Existing Connection			84	Admin	Microsoft Vi...		2017-04-18 16:46:58...	2017-04-18 16:50:56...		
Existing Connection			90	Admin	Microsoft Vi...		2017-04-18 16:46:58...	2017-04-18 16:50:56...		
Existing Connection			92	Admin	Microsoft Vi...		2017-04-18 16:46:58...	2017-04-18 16:50:56...		
Existing Connection			93	Admin	Microsoft Vi...		2017-04-18 16:46:58...	2017-04-18 16:50:56...		
Existing Connection			94	Admin	Microsoft Vi...		2017-04-18 16:46:58...	2017-04-18 16:50:56...		
Existing Session			74	Admin			2017-04-17 19:17:52...	2017-04-18 16:50:56...	7762417	TestDW
Existing Session			96	Admin	Microsoft Vi...		2017-04-18 16:46:58...	2017-04-18 16:50:56...	237701	TestDW
Existing Session			88	Admin	Microsoft Vi...		2017-04-18 16:46:58...	2017-04-18 16:50:56...	237779	TestDW
Existing Session			21	Admin			2017-04-17 19:17:52...	2017-04-18 16:50:56...	7760008	TestDW
Existing Session			84	Admin	Microsoft Vi...		2017-04-18 16:46:58...	2017-04-18 16:50:56...	237775	TestDW
Existing Session			93	Admin	Microsoft Vi...		2017-04-18 16:46:58...	2017-04-18 16:50:56...	237775	TestDW
Existing Session			92	Admin	Microsoft Vi...		2017-04-18 16:46:58...	2017-04-18 16:50:56...	237677	TestDW
Existing Session			94	Admin	Microsoft Vi...		2017-04-18 16:46:58...	2017-04-18 16:50:56...	750081	TestDW
Existing Session			20	Admin			2017-04-17 19:17:52...	2017-04-18 16:50:56...	7760068	TestDW
Existing Session			22	Admin			2017-04-17 19:17:52...	2017-04-18 16:50:56...	7759672	TestDW
Command Begin	13 - Batch	<batch xmlns="http...	86	Admin			2017-04-18 16:50:56...	2017-04-18 16:50:56...		TestDW
Progress Report Begin	6 - Commit		86	Admin			2017-04-18 16:50:56...	2017-04-18 16:50:56...		TestDW
Notification	8 - Object...		86	Admin		1	2017-04-18 16:50:56...	2017-04-18 16:50:56...		
Progress Report End	6 - Commit		86	Admin			2017-04-18 16:50:56...	2017-04-18 16:50:56...		
Discover Begin	4 - MDSCHE...	<restriction list X...	92	Admin	Microsoft Vi...		2017-04-18 16:50:56...	2017-04-18 16:50:56...		TestDW
Discover End	4 - MDSCHE...	<restriction list X...	92	Admin	Microsoft Vi...		2017-04-18 16:50:56...	2017-04-18 16:50:56...		4
Query Begin	0 - MDXQuery	SELECT NON EMPTY { ...	94	Admin	Microsoft Vi...		2017-04-18 16:50:56...	2017-04-18 16:50:56...		TestDW
Query Subcube	1 - cache ...	0100,0100,000	91	Admin	Microsoft Vi...		2017-04-18 16:50:56...	2017-04-18 16:50:56...		TestDW
Query End	0 - MDXQuery	SELECT NON EMPTY { ...	94	Admin	Microsoft Vi...		2017-04-18 16:50:56...	2017-04-18 16:50:56...		27
AUDIT Login			96	Admin			2017-04-18 16:51:06...	2017-04-18 16:51:06...		
Discover Begin	26 - DISCO...	<restriction list X...	96	Admin			2017-04-18 16:51:06...	2017-04-18 16:51:06...		
Discover End	26 - DISCO...	<restriction list X...	96	Admin			2017-04-18 16:51:06...	2017-04-18 16:51:06...		1

Рисунок Г.1 –Производительность запроса

```
SELECT NON EMPTY { [Measures].[Quantity] } ON COLUMNS, NON EMPTY { ([Product].[Product Name].[Product Name].ALLMEMBERS *
[Product].[Product Alt Key].[Product Alt Key].ALLMEMBERS * [Date].[Calendar Year].[Calendar Year].ALLMEMBERS ) } DIMENSION PROPERTIES
MEMBER_CAPTION, MEMBER_UNIQUE_NAME ON ROWS FROM [Sales] CELL PROPERTIES VALUE, BACK_COLOR, FORE_COLOR,
FORMATTED_VALUE, FORMAT_STRING, FONT_NAME, FONT_SIZE, FONT_FLAGS
```

Рисунок Г.2 – MDX-запрос

Year	Product	Brand	Quantity	
2005	TV	Panasonic	125	
		Sony	130	
	TV-UHD	Panasonic	146	
		Sony	159	
		Общий итог		691
				255
2006	TV	Panasonic	177	
		Sony	165	
	TV-UHD	LG	150	
		Panasonic	191	
		Sony	177	
		Общий итог		1551

Рисунок Г.3 – Тестирование в клиентском приложении