**ВВЕДЕНИЕ**

Так уж сложилось, что нынешний век стал весьма необычным в истории человечества. Мы живем в Информационном веке. На самом деле, уже не один десяток лет миром правит информация. Информация стала играть решающую роль при организации и в ходе работы предприятия, являясь движущей силой бизнеса и наиболее ценным активом. Это связано как с развитием информационным технологий, так и с изменением технологии принятия решений. Объёмы информации растут постоянно, и также постоянно растёт их влияние на нас.

Во всем мире организации накапливают или уже накопили в процессе своей деятельности большие объемы данных. Эти коллекции данных хранят в себе большие потенциальные возможности по извлечению новой, аналитической информации, на основе которой можно и необходимо строить стратегию фирмы, выявлять тенденции развития рынка, находить новые решения, обусловливающие успешное развитие в условиях конкурентной борьбы. Для некоторых фирм такой анализ является неотъемлемой частью их повседневной деятельности, но большинство, очевидно, только начинает приступать к нему всерьез.

Однако постоянный и ускоряющийся рост информации и количества устройств хранения этой информации, которые необходимо при этом поддерживать, привели к существенному усложнению анализа имеющихся данных. Для аналитика просмотреть весь этот объем информации без специализированной помощи не представляется возможным. Как следствие, данные часто используются не эффективно или вовсе не используются, что приводит к издержкам, которые мешают дальнейшему развитию предприятия.

Поэтому одной из главных задач любой торговой сети является совершенствование методов анализа данных (продаж), изучение которых помогает принять решения по поводу любых проблем данного бизнеса, в частности прогнозирования поведения потребителей и, как следствие, увеличения прибыли. По полученному анализу можно строить политику предприятия, а именно какие товары предлагать, на какие сделать скидки, каких сотрудников компании поощрить за проделанную работу и т.д.

Предприятие после проведения анализа и построения прогноза сможет сократить свои расходы, так как отвергнет заранее убыточные начинания, сэкономит время на, например, продлении контрактов с сотрудниками, работа которых не рентабельна.

Объектом исследования дипломного проекта являются данные о продажах компании, которые являются неотъемлемым и важным компонентом деятельности любой торговой сети, не зависимо от ассортимента товаров и отрасли обслуживания.

Предмет исследования – современные методы и модели анализа данных, продаж.

Для анализа имеющихся на предприятии данных используются различные программные средства. Однако анализ рынка существующего программного обеспечения в области анализа данных показал, что подавляющая масса программ не удовлетворяет в полном объеме потребностям организаций. И стоимость таких программных средств достаточно велика.

Целью данного дипломного проекта является предоставление достоверной информации о продажах товаров в торговой сети с целью помощи в принятии решений на основе анализа имеющихся данных и успешного прогнозирования для повышения эффективности работы компании.

Основными задачами для достижения поставленной цели являются:

* исследовать методы анализа данных;
* представить обзор существующих систем анализа данных;
* проанализировать деятельность предприятия по использованию систем анализа данных;
* разработать систему поддержки принятия решений на основе анализа данных.

Необходимо создать систему, которая помогает проанализировать имеющийся объем данных предприятия, позволяет просмотреть различные отчеты о состоянии дел за какой-то промежуток времени, составить прогноз о продажах товара в ближайшее время. Важнейшей частью такой системы является интерфейс, который был бы простым, понятным и удобным для пользователя.

Система поддержки принятия решений должна не просто предоставить какую-либо отчетность о деятельности компании, необходимо, чтобы увеличилась эффективность работы сотрудников, в частности аналитика, который непосредственно и занимается анализом полученных данных, которые предоставляются в виде небольших отчетов. Такие отчеты представляют собой агрегированную, скомпонованную информацию о продажах, которая необходима аналитику для корректного анализа.

1. **СОВРЕМЕННЫЕ МЕТОДЫ И ПОДХОДЫ К АНАЛИЗУ ПРОДАЖ В ТОРГОВОЙ СЕТИ ПРЕДПРИЯТИЯ**
   1. **Общие сведения о процессе анализа данных**

По сути, интеллектуальный анализ данных — это обработка информации и выявление в ней моделей и тенденций, которые помогают принимать решения. Принципы интеллектуального анализа данных известны в течение многих лет, но с появлением *больших данных* они получили еще более широкое распространение.

Большие данные привели к взрывному росту популярности более широких методов интеллектуального анализа данных, отчасти потому, что информации стало гораздо больше, и она по самой своей природе и содержанию становится более разнообразной и обширной. При работе с большими наборами данных уже недостаточно относительно простой и прямолинейной статистики. Имея 30 или 40 миллионов подробных записей о продажах, недостаточно знать, что два миллиона из них сделаны в одном и том же месте. Чтобы лучше удовлетворить потребности покупателей, необходимо понять, принадлежат ли эти два миллиона к определенной возрастной группе, и знать их средний заработок.

Эти бизнес-требования привели от простого поиска и статистического анализа данных к более сложному интеллектуальному анализу данных. Для решения бизнес-задач требуется такой анализ данных, который позволяет построить модель для описания информации и в конечном итоге приводит к созданию результирующего отчета. Этот процесс в обобщенном виде показан на рисунке 1.1.

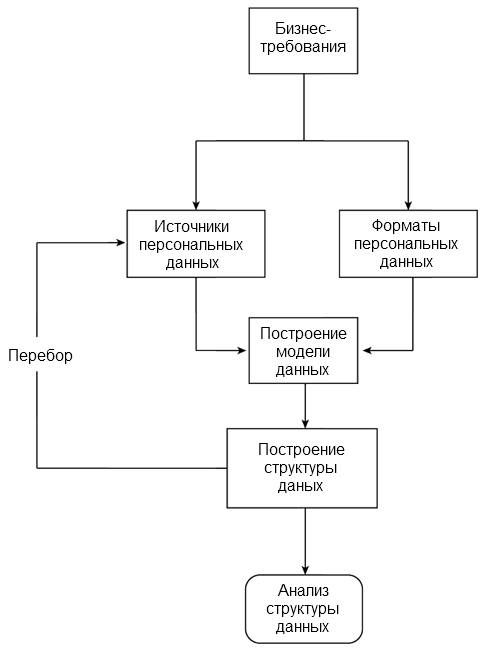


Рисунок 1.1 – Схема процесса анализа данных

Процесс анализа данных, поиска и построения модели часто является итеративным, так как нужно разыскать и выявить различные сведения, которые можно извлечь. Необходимо также понимать, как связать, преобразовать и объединить их с другими данными для получения результата. После обнаружения новых элементов и аспектов данных подход к выявлению источников и форматов данных с последующим сопоставлением этой информации с заданным результатом может измениться. [1]

Рассмотрим данный процесс более подробно.

Первым шагом в анализе является получение **исходной выборки**. На основе этих данных и строятся модели. На этом шаге необходимо активное участие эксперта для выдвижения гипотез и отбора факторов, влияющих на анализируемый процесс. Желательно, чтобы данные были уже собраны и консолидированы. Крайне необходимо наличие удобных механизмов подготовки выборки.

Чаще всего в качестве источника рекомендуется использовать специализированное хранилище данных, агрегирующее всю необходимую для анализа информацию.

Реальные данные для анализа редко бывают хорошего качества. Необходимость предварительной обработки при анализе данных возникает независимо от того, какие технологии и алгоритмы используются. Более того, эта задача может представлять самостоятельную ценность в областях, не имеющих непосредственного отношения к анализу данных.

К задачам **очистки данных** относятся такие как: заполнение пропусков, редактирование аномалий, сглаживание, обнаружение дубликатов и противоречий и прочие.

**Трансформация данных** – последний этап перед, собственно, анализом. Дело в том, что различные алгоритмы анализа требуют специальным образом подготовленные данные, например, для прогнозирования необходимо преобразовать временной ряд при помощи скользящего окна или вычисление агрегируемых показателей.

К задачам трансформации данных относятся: скользящее окно, приведение типов, выделение временных интервалов, преобразование непрерывных значений в дискретные и наоборот, сортировка, группировка и прочее.

**Анализ данных** – это процесс обнаружения в «сырых» данных ранее неизвестных нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности.

Информация, найденная в процессе применения методов анализа, должна быть нетривиальной и ранее неизвестной, например, средние продажи не являются таковыми. Знания должны описывать новые связи между свойствами, предсказывать значения одних признаков на основе других.

Основные методы анализа, а также наиболее популярные подходы будут описаны в пунктах 1.2 и 1.3.

В случае, когда извлеченные знания непрозрачны для пользователя, должны существовать методы постобработки, позволяющие привести их к **интерпретируемому виду**. [2]

На рисунке 1.2 изображен детальный процесс извлечения знаний.



Рисунок 1.2 – Процесс извлечения знаний

Интеллектуальный анализ данных ― это еще и используемые инструменты или программное обеспечение баз данных. Интеллектуальный анализ данных можно выполнить с относительно скромными системами баз данных и простыми инструментами, включая создание своих собственных, или с использованием готовых пакетов программного обеспечения. Сложный интеллектуальный анализ данных опирается на прошлый опыт и алгоритмы, определенные с помощью существующего программного обеспечения и пакетов, причем с различными методами ассоциируются разные специализированные инструменты.

Наиболее оптимальной с точки зрения гибкости, возможностей и простоты использования для анализа данных является аналитическая система, состоящая из хранилища данных, механизмов визуализации и методов построения моделей.

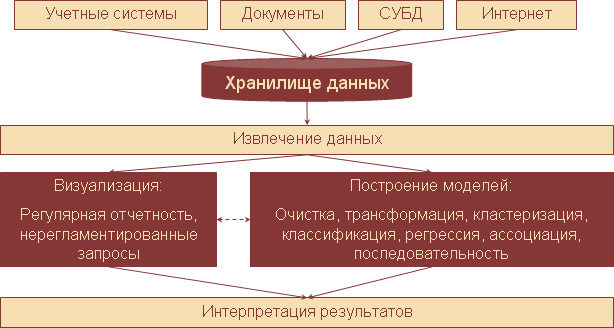


Рисунок 1.3 – Аналитическая система

Хранилище данных консолидирует всю необходимую для анализа информацию, обеспечивает быстрый и удобный доступ к необходимым данным. Полученные из хранилища данные анализируются при помощи комбинирования различных способов визуализации и механизмов построения моделей. [3]

В последнее время стала возможна работа с очень большими наборами данных и кластерная/крупномасштабная обработка данных, что позволяет делать еще более сложные обобщения результатов интеллектуального анализа данных по группам и сопоставлениям данных. Сегодня доступен совершенно новый спектр инструментов и систем, включая комбинированные системы хранения и обработки данных.

Можно анализировать самые разные наборы данных, включая традиционные базы данных SQL, необработанные текстовые данные, наборы "ключ/значение" и документальные базы. Кластерные базы данных, такие как Hadoop, Cassandra, CouchDB и Couchbase Server, хранят и предоставляют доступ к данным такими способами, которые не соответствуют традиционной табличной структуре.

В частности, более гибкий формат хранения базы документов придает обработке информации новую направленность и усложняет ее. Базы данных SQL строго регламентируют структуру и жестко придерживаются схемы, что упрощает запросы к ним и анализ данных с известными форматом и структурой.

Документальные базы данных, которые соответствуют стандартной структуре типа JSON, или файлы с некоторой машиночитаемой структурой тоже легко обрабатывать, хотя дело может осложняться разнообразной и переменчивой структурой. Например, в Hadoop, который обрабатывает совершенно "сырые" данные, может быть трудно выявить и извлечь информацию до начала ее обработки и сопоставления.

* 1. **Основные методы анализа данных**

Существует множество различных методов интеллектуального анализа данных, моделирования запросов, обработки и сбора информации.

Несколько основных методов, которые используются для интеллектуального анализа данных, описывают тип анализа и операцию по восстановлению данных. Рассмотрим некоторые ключевые методы и примеры того, как использовать те или иные инструменты для интеллектуального анализа данных. [4]

**1.2.1 Ассоциация**

Ассоциация (или отношение), вероятно, наиболее известный, знакомый и простой метод интеллектуального анализа данных. Для выявления моделей делается простое сопоставление двух или более элементов, часто одного и того же типа. Например, отслеживая привычки покупки, можно заметить, что вместе с клубникой обычно покупают сливки.

Создать инструменты интеллектуального анализа данных на базе ассоциаций или отношений нетрудно. Например, в InfoSphere Warehouse есть мастер, который выдает конфигурации информационных потоков для создания ассоциаций, исследуя источник входной информации, базис принятия решений и выходную информацию. [На рисунке 1.2](http://www.ibm.com/developerworks/ru/library/ba-data-mining-techniques/#fig2) приведен соответствующий пример для образца базы данных.

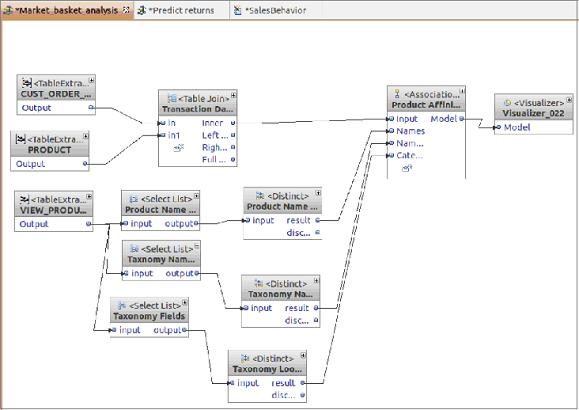


Рисунок 1.4 –Ассоциативный подход

* + 1. **Классификация**

Классификацию можно использовать для получения представления о типе покупателей, товаров или объектов, описывая несколько атрибутов для идентификации определенного класса. Например, автомобили легко классифицировать по типу (седан, внедорожник, кабриолет), определив различные атрибуты (количество мест, форма кузова, ведущие колеса). Изучая новый автомобиль, можно отнести его к определенному классу, сравнивая атрибуты с известным определением. Те же принципы можно применить и к покупателям, например, классифицируя их по возрасту и социальной группе.

Кроме того, классификацию можно использовать в качестве входных данных для других методов. Например, для определения классификации можно применять деревья принятия решений. Кластеризация позволяет использовать общие атрибуты различных классификаций в целях выявления кластеров.

**1.2.3 Кластеризация**

Исследуя один или более атрибутов или классов, можно сгруппировать отдельные элементы данных вместе, получая структурированное заключение. На простом уровне при кластеризации используется один или несколько атрибутов в качестве основы для определения кластера сходных результатов. Кластеризация полезна при определении различной информации, потому что она коррелируется с другими примерами, так что можно увидеть, где подобия и диапазоны согласуются между собой.

Метод кластеризации работает в обе стороны. Можно предположить, что в определенной точке имеется кластер, а затем использовать свои критерии идентификации, чтобы проверить это. График, изображенный на [рисунке 1.3](http://www.ibm.com/developerworks/ru/library/ba-data-mining-techniques/#fig3), демонстрирует наглядный пример. Здесь возраст покупателя сравнивается со стоимостью покупки. Разумно ожидать, что люди в возрасте от двадцати до тридцати лет (до вступления в брак и появления детей), а также в 50-60 лет (когда дети покинули дом) имеют более высокий располагаемый доход.

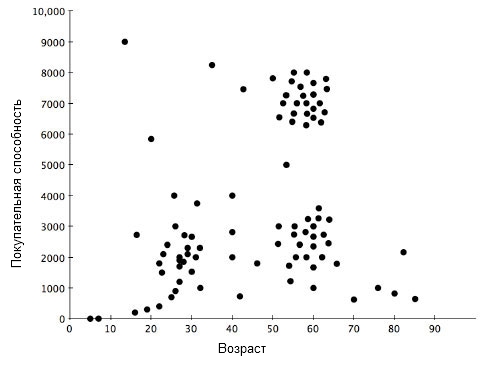


Рисунок 1.5 – Кластеризация

В этом примере видны два кластера, один в районе $2000/20-30 лет и другой в районе $7000-8000/50-65 лет. В данном случае мы выдвинули гипотезу и проверили ее на простом графике, который можно построить с помощью любого подходящего ПО для построения графиков. Для более сложных комбинаций требуется полный аналитический пакет, особенно если нужно автоматически основывать решения на информации о *ближайшем соседе*.

Такое построение кластеров являет собой упрощенный пример так называемого образа *ближайшего соседа*. Отдельных покупателей можно различать по их буквальной близости друг к другу на графике. Весьма вероятно, что покупатели из одного и того же кластера разделяют и другие общие атрибуты, и это предположение можно использовать для поиска, классификации и других видов анализа членов набора данных.

Метод кластеризации можно применить и в обратную сторону: учитывая определенные входные атрибуты, выявлять различные артефакты. Например, недавнее исследование четырехзначных PIN-кодов выявили кластеры чисел в диапазонах 1-12 и 1-31 для первой и второй пар. Изобразив эти пары на графике, можно увидеть кластеры, связанные с датами (дни рождения, юбилеи).

* + 1. **Прогнозирование**

Прогнозирование ― это широкая тема, которая простирается от предсказания отказов компонентов оборудования до выявления мошенничества и даже прогнозирования прибыли компании. В сочетании с другими методами интеллектуального анализа данных прогнозирование предполагает анализ тенденций, классификацию, сопоставление с моделью и отношения. Анализируя прошлые события или экземпляры, можно предсказывать будущее.

Например, используя данные по авторизации кредитных карт, можно объединить анализ дерева решений прошлых транзакций человека с классификацией и сопоставлением с историческими моделями в целях выявления мошеннических транзакций. Если покупка авиабилетов в США совпадает с транзакциями в США, то вполне вероятно, что эти транзакции подлинны.

**1.2.5 Последовательные модели**

Последовательные модели, которые часто используются для анализа долгосрочных данных, ― полезный метод выявления тенденций, или регулярных повторений подобных событий. Например, по данным о покупателях можно определить, что в разное время года они покупают определенные наборы продуктов. По этой информации приложение прогнозирования покупательской корзины, основываясь на частоте и истории покупок, может автоматически предположить, что в корзину будут добавлены те или иные продукты.

**1.2.6 Деревья решений**

Дерево решений, связанное с большинством других методов (главным образом, классификации и прогнозирования), можно использовать либо в рамках критериев отбора, либо для поддержки выбора определенных данных в рамках общей структуры. Дерево решений начинают с простого вопроса, который имеет два ответа (иногда больше). Каждый ответ приводит к следующему вопросу, помогая классифицировать и идентифицировать данные или делать прогнозы.

На [рисунке 1.4](http://www.ibm.com/developerworks/ru/library/ba-data-mining-techniques/#fig4) приведен пример классификации неисправных состояний.

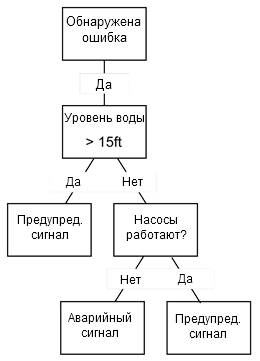


Рисунок 1.6 – Дерево решений

Деревья решений часто используются с системами классификации информации о свойствах и с системами прогнозирования, где различные прогнозы могут основываться на прошлом историческом опыте, который помогает построить структуру дерева решений и получить результат.

**1.2.7 Комбинации**

На практике очень редко используется только один из этих методов. Классификация и кластеризация ― подобные методы. Используя кластеризацию для определения ближайших соседей, можно дополнительно уточнить классификацию. Деревья решений часто используются для построения и выявления классификаций, которые можно прослеживать на исторических периодах для определения последовательностей и моделей.

**1.2.8 Обработка с запоминанием**

При всех основных методах часто имеет смысл записывать и впоследствии изучать полученную информацию. Для некоторых методов это совершенно очевидно. Например, при построении последовательных моделей и обучении в целях прогнозирования анализируются исторические данные из разных источников и экземпляров информации.

В других случаях этот процесс может быть более ярко выраженным. Деревья решений редко строятся один раз и никогда не забываются. При выявлении новой информации, событий и точек данных может понадобиться построение дополнительных ветвей или даже совершенно новых деревьев.

* 1. **Основные подходы к обработке данных**

Интеллектуальный анализ данных опирается на построение подходящей модели и структуры, которые можно использовать для обработки, выявления и создания необходимой информации. Независимо от формы и структуры источника данных, информация структурируется и организуется в соответствии с форматом, который позволяет выполнять интеллектуальный анализ данных с максимально эффективной моделью.

Подумайте о комбинировании бизнес-требований по интеллектуальному анализу данных с выявлением существующих переменных (покупатель, стоимость, страна) и созданием новых переменных, которые можно использовать для анализа данных на подготовительном этапе.

Аналитические переменные для данных, полученных из множества различных источников, можно составить в единую, определенную структуру (например, создать класс покупателей определенных уровней).

В зависимости от источника данных важно выбрать правильный способ построения и преобразования этой информации, каким бы ни был метод окончательного анализа данных. Этот шаг также ведет к более сложному процессу выявления, сбора, упрощения или расширения информации в соответствии с входными данными ([рисунок 1.5](http://www.ibm.com/developerworks/ru/library/ba-data-mining-techniques/#fig5)). [1]

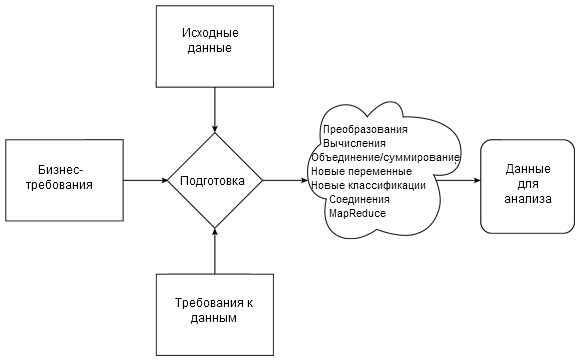


Рисунок 1.7 – Подготовка данных

Источник данных, местоположение и база данных влияют на то, как будет обрабатываться и объединяться информация.

**1.3.1 Опора на SQL**

Наиболее простым из всех подходов часто служит опора на базы данных SQL. Данный подход реализован во многих базах данных, Oracle Database, MS SQL Server, MySQL и т.д.

SQL (и соответствующая структура таблицы) хорошо понятен, но структуру и формат информации нельзя игнорировать полностью. Например, при изучении поведения пользователей по данным о продажах в модели данных SQL (и интеллектуального анализа данных в целом) существуют два основных формата, которые можно использовать: транзакционный и поведенческо-демографический.

Например, по данным о продажах можно выявить тенденции продаж конкретных товаров. Исходные данные о продажах отдельных товаров можно преобразовать в информацию о транзакциях, в которой идентификаторы покупателей сопоставляются с данными транзакций и кодами товаров. Используя эту информацию, легко выявить последовательности и отношения для отдельных товаров и отдельных покупателей с течением времени. Это позволяет вычислять последовательную информацию, определяя, например, когда покупатель, скорее всего, снова приобретет тот же товар.

Из исходных данных можно создавать новые точки анализа данных. Например, можно развернуть (или доработать) информацию о товаре путем сопоставления или классификации отдельных товаров в более широких группах, а затем проанализировать данные для этих групп, вместо отдельных покупателей.

[В таблице 1.1](http://www.ibm.com/developerworks/ru/library/ba-data-mining-techniques/#table1) приведен пример расширения информации.

Таблица 1.1 – Расширенная таблица товаров

| **product\_id** | **product\_name** | **product\_group** | **product\_type** |
| --- | --- | --- | --- |
| 101 | Клубника неупакованная | Клубника | Фрукты |
| 102 | Клубника в коробках | Клубника | Фрукты |
| 110 | Бананы неупакованные | Бананы | Фрукты |

**1.3.2 Документальные базы данных и MapReduce**

Обработка с помощью функции MapReduce многих современных документальных и NoSQL баз данных, таких как Hadoop, нацелена на очень большие наборы данных и информацию, которая не всегда соответствует табличному формату. При работе с программным обеспечением интеллектуального анализа данных эта система может как принести пользу, так и вызвать проблемы.

Основная проблема данных на основе документов ― это неструктурированный формат, который может потребовать дополнительной обработки. Много различных записей могут содержать аналогичные данные. Сбор и согласование этой информации в целях упрощения ее обработки зависит от этапов подготовки и применения MapReduce.

В системе, основанной на MapReduce, на этапе преобразования исходные данные *нормализуются* ― приводятся к стандартной форме. Этот шаг может быть относительно простым (определение ключевых полей или точек данных) или сложным (анализ и обработка информации для создания выборки данных). В процессе преобразования данные приводятся к стандартизированному формату, который можно использовать в качестве базы.

Сокращение ― это суммирование или количественная оценка информации с последующим выводом этой информации в стандартизованную структуру, основанную на итогах, суммах, статистике или других результатах анализа, выбранных для вывода данных.

Запросы к этим данным часто бывают сложными ― даже при использовании специализированных инструментов. Идеальный подход к интеллектуальному анализу данных заключается в использовании этапа MapReduce в рамках подготовки данных.

Например, при выполнении интеллектуального анализа данных методом ассоциации или кластеризации на первом этапе лучше всего построить подходящую статистическую модель, которую впоследствии можно будет применять для выявления и извлечения необходимой информации. Лучше использовать этап MapReduce для извлечения и вычисления этой статистической информации с ее последующим вводом в остальную часть процесса интеллектуального анализа данных, что ведет к созданию структуры, показанной на [рисунке 1.7](http://www.ibm.com/developerworks/ru/library/ba-data-mining-techniques/#fig7).

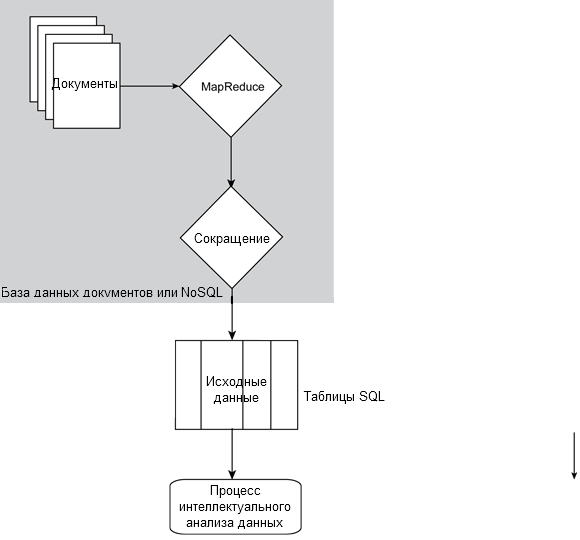


Рисунок 1.8 – Структура MapReduce

В предыдущем примере выполнялась обработка (в данном случае посредством MapReduce) исходных данных в документальной базе данных и преобразовывалась в табличный формат в базе данных SQL для целей интеллектуального анализа данных.

Для работы с этой сложной и даже неструктурированной информацией может потребоваться более тщательная подготовка и обработка. Существуют сложные типы и структуры данных, которые нельзя обработать и подготовить в нужном вам виде за один шаг. В этом случае можно направить выход MapReduce либо для *последовательного* преобразования и получения необходимой структуры данных, как показано на [рисунке 1.8](http://www.ibm.com/developerworks/ru/library/ba-data-mining-techniques/#fig8), либо для *индивидуального* изготовления нескольких таблиц выходных данных.

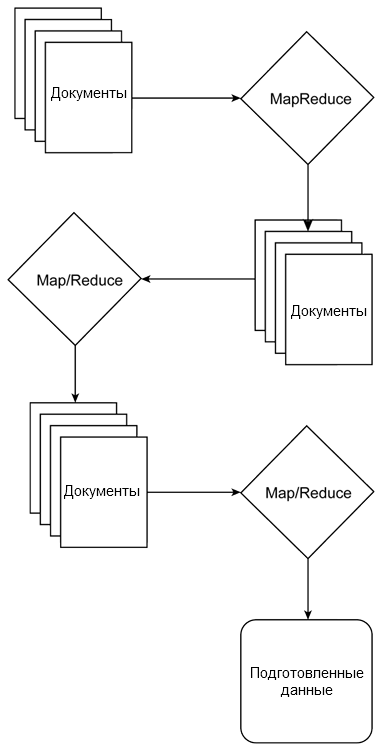


Рисунок 1.9 – Последовательная цепочка вывода результатов обработки MapReduce

Независимо от исходных данных, многие инструменты могут использовать неструктурированные файлы, CSV или другие источники данных. Например, InfoSphere Warehouse в дополнение к прямой связи с хранилищем данных DB2 может анализировать неструктурированные файлы.

* 1. **Понятие Business Intelligence**

Современному бизнесу необходимы аналитические средства. Потребность постоянно уменьшать издержки производства, оптимизировать складские запасы, исследовать рынок и прогнозировать его развитие поддерживают интерес к подобным технологиям. На Западе без консультации с аналитиками не решается ни один серьезный вопрос. У нас же из-за того, что в течение длительного времени даже статистика должна была соответствовать идеологии, к аналитическим технологиям относятся так же скептически, как к гадалкам. И эта точка зрения в некотором смысле справедлива, потому что во многих областях четкие алгоритмы мирно сосуществуют с интуитивными догадками. Однако правильное использование современных аналитических инструментов может дать хорошие результаты, поэтому в последнее время интерес к разработкам в этой области проявляют многие компании, желающие повысить свои показатели.

Business Intelligence (BI) в широком смысле слова определяет:

* процесс превращения данных в информацию и знания о бизнесе для поддержки принятия улучшенных и неформальных решений;
* информационные технологии (методы и средства) сбора данных, консолидации информации и обеспечения доступа бизнес-пользователей к знаниям;
* знания о бизнесе, добытые в результате углубленного анализа детальных данных и консолидированной информации. [5]

В основе технологии BI лежит организация доступа конечных пользователей и анализ структурированных количественных по своей природе данных и информации о бизнесе. BI порождает итерационный процесс бизнес-пользователя, включающий доступ к данным и их анализ, и тем самым проявление интуиции, формирование заключений, нахождение взаимосвязей, чтобы эффективно изменять предприятие в положительную сторону. BI имеет широкий спектр пользователей на предприятии, включая руководителей и аналитиков.

BI объединяет методы анализа данных и подходы к их обработке, которые были описаны выше, в совокупность технологий, программного обеспечения и практик, направленных на достижение целей бизнеса путём наилучшего использования имеющихся данных. Именно поэтому, в отличие от других классов делового ПО, актуальность во внедрении BI-систем в трудные с экономической точки зрения времена не ослабевает, а только усиливается. Развитие средств визуального представления данных, мобильных и облачных технологий сделали BI-инструменты массовыми всего за последние несколько лет. На рисунке 1.9 изображены 15 компонентов BI платформ.

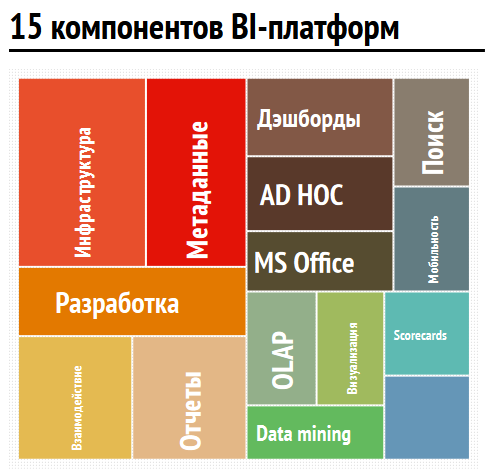
[](http://www.tadviser.ru/index.php/%D0%A4%D0%B0%D0%B9%D0%BB:15_bi_gartner.png)

Рисунок 1.10 – Компоненты BI платформ

Архитектура Business Intelligence определяет компоненты доставки BI-информации и компоненты BI-технологии (рисунок 1.11). После определения профилей использования BI-информации, может быть спроектирована архитектура доставки информации, основанная на этих профилях и на требуемом типе внедрения. Это может быть любая смесь настольных клиентов с сетевым подключением, настольных клиентов и сервера, тонких клиентов на основе Web и других мобильных вычислительных устройств. Архитектура доставки информации определит пользовательские интерфейсы, которые часто являются порталами с возможностью персонализации. [6]

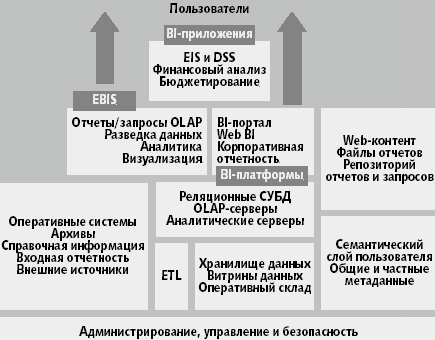


Рис.1.11 Архитектура Business intelligence

Архитектура BI-технологии определяет инфраструктуру и компоненты, необходимые для поддержки внедрения, эксплуатации и администрирования BI-инструментов и приложений, а также связи этих компонентов. Прочная архитектура BI-технологии будет состоять из двух важных слоев: инфраструктуры и прикладных сервисов (или функциональности). Инфраструктурный слой включает информационные ресурсы, администрирование и сети. На этом слое данные собираются, интегрируются и становятся доступными. Хранилище данных является одним из возможных компонентов инфраструктурного слоя. Для использования BI в оперативных системах может потребоваться оперативный склад данных (operational data store, ODS), возможно связанный с корпоративными структурами workflow. Прикладные сервисы включают все BI-сервисы, такие как механизмы запросов, анализа, генерации отчетов и визуализации, а также средства безопасности и метаданные.

Ключевыми преимуществами от внедрения [BI](http://www.tadviser.ru/index.php/BI)-системы на данный момент являются повышение скорости и точности составления отчетов, аналитики и планирования, повышение эффективности процессов принятия решений, увеличение удовлетворенности клиентов. Гораздо реже BI-системы позволяют добиться сокращения операционных расходов или увеличения объемом выручки.

На рисунке 1.12 изображены наиболее часто достигаемые в ходе внедрения BI-систем преимущества.

[](http://www.tadviser.ru/index.php/%D0%A4%D0%B0%D0%B9%D0%BB:Bi_benefits_tadviser_2.png)

Рисунок 1.12 – Преимущества внедрения BI-систем

Проблема, однако, заключается в том, что большая часть аналитического инструментария BI рассчитана на крупные корпорации и стоит баснословных денег. Как правило, это специализированные технологии, поддерживающие работу с огромными массивами данных и требовательные к аппаратной платформе. Например, Microsoft, SAS Institute, Oracle, SAP, Cognos и другие. Для малого и среднего бизнеса практически ничего нет, хотя потребность в анализе они также испытывают. Требования к подобным системам продиктованы здравым смыслом: реализация современных механизмов анализа данных, способность работать с выборками в сотни тысяч записей, интеграция с офисными приложениями, стабильная работа на офисных ПК, возможность использования для анализа данных из разнородных источников, доступная цена, простота использования.

Понятно, что у компаний малого и среднего бизнеса нет очень большого объема данных, но вручную данные также не могут быть обработаны. Но потребность в таких компонентах BI, как Data Mining, AD HOC или OLAP, попросту отпадает, так как нет такого объема данных, которые бы были необходимы для анализа. Да и построение того же качественного OLAP-куба займет не только много времени, но и денег для покупки BI-инструмента, который предназначен для его построения.

С учетом всего вышенаписанного для анализа продаж в торговой сети хватит визуализации данных, поиска данных, отчетов и дэшбордов. Но выбор методов и подходов анализа данных для построения системы поддержки принятия решений на основе анализа данных зависит от целей и потребностей конкретного бизнеса. Поэтому разработка собственно BI-системы должна производиться с учетом особенностей деятельности конкретной торговой сети.

1. **АНАЛИЗ ПРИМЕНЕНИЯ МЕТОДОВ БАЗ ДАННЫХ В ПРОЦЕССЕ АНАЛИЗА ПРОДАЖ В СЛУЖБЕ BI КОМПАНИИ ЭПАМ СИСТЕМЗ**

**2.1 Характеристика и организационная структура ИООО «Эпам Системз»**

Компания «ЭПАМ Системз» – крупнейший поставщик услуг в области разработки программного обеспечения на территории бывшего Советского Союза и в Центральной и Восточной Европе. Созданная в 1993 году, сегодня она имеет 17 представительств в восьми странах мира, в штате более 4500 высококвалифицированных специалистов.

Компания «ЭПАМ Системз» выполняет проекты более чем в 30 странах мира для частных и государственных учреждений.

Сегодня ЕРАМ Systems входит в список приоритетных поставщиков ИТ-услуг компаний самых различных отраслей, среди которых автобизнес, энергетика, нефть и газ, телекоммуникационная и FMCG-индустрии, а также учреждения государственного сектора. Опыт компании для заказчиков данных областей включает разработку, внедрение и интеграцию полнофункциональных решений, а также развитие, тестирование, поддержку и сопровождение как существующих, так и новых ИТ-комплексов.

**Организационная структура** EPAM Systems является линейно-функциональной, так как основана на тесном сочетании линейных и функциональных связей в аппарате управления. Она обеспечивает такое разделение труда, при котором линейные звенья принимают решения и управляют, а функциональные – консультируют, информируют, координируют и планируют хозяйственную деятельность. В основу организации функциональных действий положен линейный принцип. Руководитель функционального отдела является одновременно линейным руководителем непосредственно подчиненных ему работников.

В Беларуси управленческая структура представлена в соответствии с рисунком 2.1:

Рисунок 2.1 - Управленческая структура Epam Systems в Беларуси

Что касается непосредственно структуры производственных единиц, занятых производством программного обеспечения, а также подразделений, обслуживающих процессы в компании, можно представить схему на рисунке 2.2

Рисунок 2.2 – Структура производственных единиц

Встречаются отделы в компании, которые являются локально распределенными (Competency Сenters).

В общем виде производственные отделы объединяются по направлениям работы в рамках региона, а административные и сервисные подразделения – по конкретному офису.

Так как компания организует свою производственную деятельность в качестве проекта, каждый сотрудник имеет одного или нескольких проектных менеджеров. Однако помимо проектного менеджера работник также имеет своего ресурсного менеджера, который отвечает за удовлетворенность сотрудника условиями работы, согласовывает зарплату и должность.

С целью аккумулирования наработок внутри различных предметных областей компания формирует и развивает технологические и отраслевые **центры компетенции**. Данные центры дают возможность сотрудникам доступа к лучшим мировым и отечественным практикам, развивать свои навыки и умения для решения нестандартных и сложных задач, а также использовать решения, накопленные компанией в ходе успешного выполнения проектов. На данный момент в компании действуют следующие центры компетенции:

- Рынки капитала;

- Страхование;

- E-Commerce;

- Тестирование;

- Мобильные решения;

- Облачные технологии;

- Управления контентом и др.

Разработкабизнес-приложений может осуществляться и по модели выделенного центра разработки. В этом случае формируется команду профессионалов, обладающих именно тем набором знаний и технологий, которые необходимы заказчику. Разработчики, аналитики, тестировщики выделяются на длительный срок и работают в тесном сотрудничестве с ИТ-департаментом заказчика. Это дает возможность специалистам компании глубоко изучить особенности бизнеса, внутреннюю инфраструктуру компании и стать экспертом в конкретной области.

**2.2 Анализ деятельности ИООО «Эпам Системз» в сфере BI-решений**

Компетенция EPAM Systems в области разработки корпоративных хранилищ, данных и BI-решений охватывает широкий спектр технологий, предлагаемых ведущими производителями программного обеспечения для интеграции, организации и анализа информации. В таблице 2.1 описаны все возможные BI-решения, предоставляемые компанией.

Таблица 2.1 – BI-решения, предоставляемые компанией

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Источники** | **Интеграция (ETT/ETL)** | **Реляционное хранилище** | **Многомерное хранилище** | **BI-кубы** | **KPI, Аналитика, Отчетность** |
| - Разрозненные данные  - Учетные системы и др. | - Очистка данных  - Консолидация данных | - Репликация данных  - Закрытие периодов  - Версионность данных  - Учетные записи пользователей | - Многомерная структура  - Хранение по структуре «звезда» или «снежинка» | - Многомерная аналитика  - Срезы данных, метрики, показатели | - Отчетность  - Аналитика  - KPI  - DrillDown (детализация данных) |
| **Oracle** | Oracle Data Integrator | Oracle Database | Oracle BI Server | Oracle Essbase | Oracle BI |
| **Microsoft** | MS SQL Server Integration Services | MS SQL Server | MS SQL Server | MS SQL Server Analysis Services | SharePoint Portal |
| **IBM Cognos** | IBM Data Stage | IBM DB2 | Cognos Framework Manager | Cognos PowerCubes | Cognos BI |
| **SAP BW** | SAP NW BW | SAP NW BW (на базе MaxDB, Oracle, SQL Server, DB2) | SAP NW BW | SAP BW Business Explorer (BEx) | SAP BW BEx (Analyzer, Query Designer, Report Designer, Web Application Designer) |
| **SAP BO** | SAP BO Data Services | SAP BO Enterprise | SAP BO Enterprise | SAP BO WebI/ DeskI/ Voyager/ Live Office | SAP BO WebI/ DeskI/ Voyager/ LO/ Crystal reports/ Xcelsius |
| **MicroStrategy** | Любое стороннее решение | Любое стороннее решение | MicroStrategy Intelligence Server | MicroStrategy OLAP Services | MicroStrategy Report Services |

Для обеспечения высокого качества реализации проектов по созданию систем для структурирования информации на базе компании действует Центр компетенции по BI-решениям, объединяющий накопленную экспертизу EPAM по работе с технологиями Business Intelligence.

Благодаря большому опыту работы с различными BI-платформами и инструментами для построения хранилищ данных и непосредственному участию специалистов компании в разработке BI-продуктов для ведущих вендоров ПО (Oracle и SAP), консультанты и инженеры EPAM Systems способны найти оптимальное решение исходя из требований и бизнес-задач каждого заказчика.

Рассмотрим более подробно BI-проекты компании и их деятельность, в частности процесс анализа данных.

Большая часть деятельности BI-проектов и имена самих заказчиков подлежат неразглашению. Это делается потому, что BI работает с данными, данными заказчика, которые для большинства заказчиков BI-решений представляют собой коммерческую информацию, которая имеет огромное значение для их бизнеса и разглашение которой может привести к убыткам со стороны заказчика. Каждый сотрудник Центра компетенции по BI-решениям подписывает договор о неразглашении и конфиденциальности разработки. В связи с этим имена заказчиков преобразуются в Заказчик А, Заказчик Б и т.д.

Рассмотрим цикл разработки BI-решения (процесс анализа) на примере реального Заказчика А.

Заказчик А специализируется в аутсорсер управлении в сфере услуг по водоподготовке и водоочистке для муниципальных и индустриальных заказчиков, является лидером по разработке, проектированию и строительству «под ключ» очистных сооружений. Компания также создает адаптированные технологические решения. В 2010 году товарооборот компании составил 34.8 миллиардов Евро.

Заказчик А оплатил полный цикл разработки BI-решения. Под полным циклом разработки понимается разработка всех компонентов BI-платформ или хотя бы одного из находящихся на одном уровне разработки, например, уровень представления: отчеты, ad-hoc, визуализация, dashboards и т.д. Данные анализируются на всех уровнях разработки, то есть процесс разработки – это и есть процесс анализа, заказчик же работает с конкретной визуализацией (результатом анализа), в данном проекте такой визуализацией являются отчеты.

Цикл разработки начинается с определения бизнес-требований заказчика. Этим процессом занимается бизнес-аналитик, который предоставляет разработчикам уже проанализированные требования заказчика в понятном каждому разработчику виде (инструкции, спецификации, guidelines) и понятия конкретного бизнеса. Заказчик А, хоть и специализируется в аутсорсер управлении в сфере услуг по водоподготовке и водоочистке для муниципальных и индустриальных компаний, данные для анализа предоставил про сотрудников компании, то есть понятиями будет личная информации (имя, фамилия, пол, адрес), должность, отдел, дата приема на работу, дата увольнения, выплаты и т.д. Требования, определенные на начальном этапе, могут меняться в процессе разработки, в отличии от понятий, так как понятия — это названия сущностей, атрибуты этих сущностей, факты, которые добавить в проект после согласования требований и начала разработки будет очень ресурсоемко. Естественно наиболее важные требования обычно постоянны, такие как бюджет, конечное представление решения, логика данных, инструменты, команда разработчиков, хотя и эти параметры могут поменяться с течением времени. Чаще всего меняются требования, относящиеся к представлению данных, особенно изображению данных, шрифтов, картинок и т.д.

Далее определяется инструментарий разработки, то есть с помощью каких программных средств это будет разрабатываться. В большинстве случаев заказчик уже может иметь какое-либо программное обеспечение или уже готовые системы, с помощью которых он бы хотел видеть разработку.

В зависимости от выбранного инструментария набирается команда разработчиков. Такой подход наиболее эффективно и качественно помогает создать правильное решение для конкретных целей бизнеса.

Набранная команда же и разрабатывает BI-решение, то есть занимается анализом имеющихся у заказчика данных.

Разработка BI-решения начинается с проектирования хранилища данных. Этим занимается архитектор DWH (Data Ware House).

Как правило, данные в такие хранилища поступают из OLTP систем. Данные из OLTP-системы копируются в хранилище данных таким образом, чтобы при построение отчётов и OLAP-анализа не использовались ресурсы транзакционной системы и не нарушалась её стабильность. Данные загружаются в хранилище с определённой периодичностью, поэтому актуальность данных может несколько отставать от OLTP-системы.

Хранилище данных не является абстрактным по своей структуре (набору таблиц, полей в них и взаимосвязей между таблицами), а создано на основе модели данных. Модель данных представляет собой описание всех сущностей, объектов базы данных корпоративного хранилища данных и включает в себя: концептуальную модель данных, логическую модель данных и физическую модель хранилища данных. На уровне концептуальной модели определяются сущности и взаимосвязи между ними. На уровне логической модели сущности делятся на бизнес-области, им дается подробное и полное описание, прописываются взаимосвязи. При разработке физической модели базы данных определяется вся структура базы данных - от таблиц и полей в них, до партиций и индексов.

После построения модели хранилища, в котором храниться вся информация по системе: описание сущностей, атрибутов, логика данных, тип данных, комментарии и т.д. - в процесс вступает команда ETL-разработчиков.

ETL-командой пишутся процедуры для загрузки данных из OLTP системы в хранилище. Такой процесс получил название ETL (от англ. Extract, Transform, Load — дословно «извлечение, преобразование, загрузка»), который включает в себя:

- извлечение данных из внешних источников;

- трансформация и очистка данных, чтобы они соответствовали нуждам бизнес модели;

- загрузка данных в хранилище данных.

Кроме того, создается расписание загрузки новых данных в базу, пишутся скрипты для автоматизации процесса.

Процесс преобразования данных изображен на рисунке 2.3.



Рисунок 2.3 – Архитектура хранилища данных (DWH)

Так называемой область стейджа (Stage Area) для хранилищ, данных, это место для временного хранения копии данных из систем источников. Область стейджа требуется, в основном, по причинам временного характера. Все необходимые данные из систем источников должны быть в наличии перед началом интеграции данных в хранилище.

Ввиду разных схем ведения бизнеса, обработки данных, доступных ресурсов аппаратного обеспечения и существующих ограничений пропускной способности сетей для передачи данных, не всегда есть возможность вытащить все необходимые данные из систем источников в один и тот же момент времени.

Также в стэйдже преобразуются данные из различных источников в одну какую-то версию данных, происходят переводы данных. Например, статусы работника в одних источниках данных (системах) обозначались буквами, в других – цифрами, но все это переводят (транслируют) в единый и понятный для всех систем статус.

Не все хранилища требуют использования области стейджа. Для многих хранилищ вполне возможно использование ETL, который будет загружать данные в ХД непосредственно из систем источников.

Сами хранилища состоят из множества таблиц. Большинство из них непосредственно связаны с предметной областью (таблицы фактов и измерений), но есть такие, которые хранят историю изменений, произошедших в базе, а также таблицы («backup» таблицы) предназначенные для возможности «откатить» базу к первоначальному состоянию, то есть вернуть предыдущие значения измерений и фактов.

Аналитики, как основные пользователи хранилища данных, нуждаются в отслеживании изменений значений атрибутов аналитических измерений. Отслеживание изменений значений аналитических измерений в хранилище данных решается путем применения механизма медленно меняющихся измерений (Slowly Changing Dimensions, SCD). Наиболее популярные из них являются SCD Type 1, SCD Type 2 и SCD Type 3.

Медленно меняющиеся измерения в базе данных реализуются в виде обычных таблиц, в которые добавляется ряд служебных столбцов, позволяющий реализовать логику отслеживания изменений данных.

К медленно меняющимся измерениям первого типа (SCD Type 1) относятся те измерения, в которых не поддерживается отслеживание изменений данных во времени, т.е. значения полей записей в случае их изменения просто обновляются.

Действия, производимые с записями таблицы SCD Type 1 в зависимости от их статуса, представлены в таблице 2.2.

Таблица 2.2 – Действия, производимые с записями таблицы SCD Type 1

|  |  |
| --- | --- |
| **Статус записи** | **Действие** |
| На добавление | Записи присваивается следующий по порядку уникальный идентификатор. Запись добавляется в таблицу. |
| На изменение | Запись изменяется. |
| На удаление | Никаких действий над записью не производится. Удалять запись из таблицы нельзя, потому что к ней могут быть «привязаны» фактические данные. |

К медленно меняющимся измерениям второго типа (SCD Type 2) относятся те измерения, в которых поддерживается отслеживание изменений данных во времени следующим образом: старая запись помечается как утратившая актуальность, и добавляется новая запись с тем же идентификатором, но уже с обновленными полями.

К медленно меняющимся измерениям третьего типа (SCD Type 3) относятся те измерения, в которых поддерживается отслеживание изменений данных во времени путем добавления в структуру таблиц полей, хранящих предыдущие значения.

Если SCD Type 2 позволяет отслеживать неограниченное число изменений, то в SCD Type 3 количество отслеживаемых изменений ограничивается количеством дополнительных полей.

Использование различных типов измерений зависит от потребностей конкретного бизнеса.

В хранилищах могут также содержаться таблицы с агрегированными данными, которые необходимы, если в таблицах фактов хранятся данные с очень маленькой детализацией по времени. Детализация по времени, так называемая гранулярность таблицы фактов, может быть час, день, месяц, год и т.д. Если гранулярность таблицы фактов по часам, то для выборки данных из такой таблицы, могут понадобиться часы, ведь в большинстве случаев необходимо просмотреть информацию по операциям, произошедших за месяц. Конечно все зависит от требований анализа, но хорошей практикой является создание таблиц, где уже хранятся пред агрегированные данные, которые наиболее часто используются в запросах к базе.

После загрузки данных в хранилище начинается разработка модели данных для отчетов, так называемые ДатаМарты (DataMarts) (рисунок 2.3), из которых и получаются схемы «звезда» или «снежинка». Поля из таблиц ДатаМартов непосредственно используются для построения отчетов. Механизм вытягивания данных из базы может быть различный в зависимости от инструментария, выбранного заказчиком. Заказчик А купил лицензию на использования IBM Cognos, представленную Framework Manager и Cognos BI.

Cognos Framework Manager подключается напрямую к хранилищу данных (в данном проекте Oracle Database), вытягивает данные из ДатаМартов. Данные проходят через несколько слоев обработки (связываются, группируются, составляются иерархии и т.д) После составляются непосредственно «звезды» или «снежинки», из которых создаются пакеты, на которых строятся отчеты с помощью Cognos BI.

Данным процессом занимается непосредственно разработчик таких отчетов, специалист по IBM Cognos.

После разработки отчетов (аналитической части) может происходит тестирование данных тестировщиками. Данные проходят большой путь, чтобы пользователь увидел их отчете. Данные могут теряться в процессе переход между уровнями, показанными на рисунке 2.3. А ведь согласованность и правильность данных в данном процессе ключевой момент, так как данные нужны пользователям для анализа деятельности компании, сотрудников и т.д.

Естественно после того как разработаны отчеты их тоже необходимо протестировать. Этим могут заниматься как тестировщики, так и сами пользователи.

На этом, в принципе, проект заканчивается для EPAM Systems. Обычно со стороны самого заказчика привлекаются специалисты, проверяющие и тестирующие систему, и решающие возникающие проблемы, хотя иногда этим может заниматься команда EPAM Systems.

Сам процесс анализа представлен в приложении А, на диаграммах подробно представлены входные и выходные данные, а также дана информация о том, что управляет процессом анализа. Из рисунков приложения А видно, что данным процесс является довольно трудоемким и задействует много материальных и трудовых ресурсов.

Каждый заказчик приобретает лицензию на использования разработчиками данного ПО, для работы с данным приобретенным ПО необходимы высококлассные специалисты. Естественно на большие проекты с огромными объемами данных работают большие команды разработчиков, которые включают в себя бизнес-аналитиков, архитекторов DWH, ETL-разработчиков, разработчиков аналитики (отчетов), тестировщиков, проектных менеджеров. Это минимум 6 человек для полного цикла разработки (анализа). Для крупных заказчиков нанять такую команду не сложно, тем более что преимущества полученной системы перевесят затраты на ее разработку. Для компаний малого и среднего бизнеса такие системы не стоят такой разработки, поэтому EPAM Systems не работает с малым и средним бизнесом, хотя это неплохая ниша на рынке. Таких компаний достаточно много и для работы с таким заказчиком не требуется целой команды разработчиков, а буквально одного, двух человек будет достаточно.

Поэтому компанией EPAM Systems было принято решение разработать систему для пред-продажи (pre-sale) для одной торговой сети. Это значит, что оплата за работу будет проведена только, если сторона заказчика будет довольна разработанным решением.

Разработка системы поддержки принятие решений на основе анализа данных на примере продаж торговой сети позволит работать с компаниями малого и среднего бизнеса, кроме того поможет пользователям такой системы проанализировать весь объем данных, имеющийся в их распоряжении.

Использование языка программирования Java в процессе разработки системы поддержки дает возможность поддерживать кроссплатформенность, которая будет важна для любой компании.

Таким образом, разработка система поддержки принятия решений на основе анализа продаж в торговой сети является актуальной, т.к. это позволит EPAM Systems работать с маленькими компаниями и не терять клиентов ввиду невозможности этих клиентов оплатить услуги EPAM Systems. При этом разрабатываемый портал будет полностью соответствовать интересам заказчика.

**3 РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ НА ОСНОВЕ АНАЛИЗА ПРОДАЖ В ТОРГОВОЙ СЕТИ**

**3.1 Торговые сети**

Для разработки системы поддержки принятия решений на основе анализа продаж в торговой сети важно понимать все тонкости такого бизнеса, так как что в одном бизнесе просто сопутствующая информация для другого может представлять большую ценность и являться коммерческой информацией. [10]

Все данные для системы предоставляются заказчиком, но так как проект для торговой сети является предварительным (pre-sale), что было описано в предыдущем пункте, заказчик не согласен показать все свои реальные данные, поэтому данными для анализа будут продажи в торговой сети за 2008 – 2012 года.

Сети магазинов — один из наиболее важных и значительных феноменов розничной торговли. Сеть магазинов — это два или более торговых заведения, находящихся под общим владением и контролем, продающих товары аналогичного ассортимента, имеющих общую службу закупок и сбыта, а возможно аналогичное архитектурное оформление.

Ассортимент товаров является важным признаком классификации розничной торговой сети. Число товарных позиций, представленных покупателю, определяется форматом магазина. Одно из направлений совершенствования розничной торговой сети — ее специализация, которая способствует облегчению труда и росту производительности, оказывает положительное влияние на качество обслуживания покупателей.

В данном проекте будет рассмотрена специализированная розничная сеть.

Для качественного анализа необходима полная информация о товарах, способах продвижения товара, естественно время покупки, каналы продаж, а именно как был продан тот или иной товар, то есть через интернет, в магазине или у дилера, а также необходима информация о самом покупателе, особенно из какой страны покупатель, так как торговые точки могут находиться в разных странах.

Необходимо учесть, что анализ может проводиться по объему продаж как в стоимостном выражении, так и в количественном, то есть анализируется не только выручка от реализации, но и количество проданного товара. Также анализ может проводиться по себестоимости и цене для продажи по каждому продукту, которые зависят от времени. В нашем случае себестоимость и цена для продажи меняется практически каждый день.

Фактами будут выручка от реализации, количество проданного товара, себестоимость и цена для продажи по каждому продукту.

Процесс анализа данных практически идентичен процессу разобранному в предыдущей главе, кроме одного большого различия: не требуется специальные инструменты для анализа, так весь анализ будет происходить в одной системе, как построение отчетов, так и вытягивание данных из хранилища, соответственно нет необходимости в наборе специализированной команды. Поэтому контекстная диаграмма на рисунке А.1 приложения А будет декомпозирована на две составляющие: Разработать документацию, спецификацию и Разработать BI-решение.

**3.2 Спецификация системы поддержки принятия решений**

Целью данного проекта является предоставление достоверной информации о продажах товаров в торговой сети с целью анализа и успешного прогнозирования для повышения эффективности работы компании.

Необходимо создать приложения, которое позволяет просмотреть различные отчеты о состоянии дел за какой-то промежуток времени, а также составить прогноз о продажах товара в ближайшее время. В проекте необходимо создать такой интерфейс, который был бы простым, понятным и удобным для пользователя.

При разработке проекта по построению отчетов для помощи в анализе данных важно не просто предоставить какие-либо отчеты о деятельности компании. Важно, чтобы это увеличило эффективность работы сотрудников в частности аналитика, который непосредственно и занимается анализом полученных данных, которые предоставляются в виде небольших отчетов. Такие отчеты представляют собой агрегированную, скомпонованную информацию о продажах, которая необходима аналитику для корректного анализа.

Реализация такого проекта является довольно сложной, требующей временных, трудовых и научных затрат, и ресурсов. В рамках данного проекта ставится задача о написании начальной версии системы анализа продаж в торговой сети в виде веб-приложения, которую впоследствии можно усовершенствовать и приблизить тем самым к готовому программному продукту, который уже можно в дальнейшем внедрять в эксплуатацию.

Для достижения цели были поставлены и решены следующие задачи:

1. Изучение предметной области. Был разобран и изучен основной процесс предметной области.

2. Создание хранилища данных на основе изученной предметной области, включающей основные структурные элементы предметной области.

3. Выявление основного функционала, который полно и достоверно отображает всю сущность выбранной предметной области. На основе этого построение диаграммы функциональной модели работы системы и диаграммы вариантов использования (Use Case).

4. Написание программной реализации данной предметной области, которое включает в себя следующие основные моменты:

* отображение информации обо всех отчетах, которые могут понадобиться для анализа;
* просмотр конкретного анализа или отчета;
* возможность добавления, удаления и редактирования пользователей, которые имеют право на просмотр информации о продажах.

Кроме того, программный продукт должен отвечать ряду технологических требований:

* приложение должно функционировать на основе клиент-серверной архитектуры для работы в многопользовательском режиме на базе протокола TCP/IP;
* пользовательский интерфейс должен быть реализован при помощи web-приложения; [8, 9]
* архитектура приложения должна легко масштабироваться;
* приложение должно быть кроссплатформенным, кроссбраузерным. В приложении логика и представление должны быть реализованы отдельно друг от друга.

В качестве сервера базы данных был выбран Oracle Database, так как Oracle отвечает всем требованиям к хранилищам данных.

Поставленные задачи были решены следующими способами:

* При изучении предметной области было использовано приложение Process Modeler r7. С помощью данного приложения была построена функциональная модель разрабатываемого продукта.
* Информационная модель системы была построена с помощью Erwin Data Modeler r7. Для разработки схемы базы данных использовалось приложение SQL Developer. Используемый сервер базы данных – Oracle Database.
* При построении диаграммы вариантов использования и других диаграмм данного приложения было использовано приложение Enterprise Architect 7.5.

Выдвинутые требования позволят создать программный продукт, архитектура которого будет соответствовать наиболее эффективному анализу данных предприятия.

Для повышения надежности работы системы необходимо, чтобы разрабатываемая система предоставляла разграничение доступа к информационному наполнению системы. С этой целью были выделены следующие роли: роль администратора и роль пользователя системы.

Администратор системы отвечает за администрирование системы, т.е. управление пользователями системы. Администрирование системы включает следующие группы действий:

* добавление нового пользователя системы;
* изменение информации по текущим пользователям;
* удаление старых пользователей системы.

Пользователь сможет просматривать информацию о продажах, то есть имеет право только на просмотр, но не на изменение.

Функционал системы будет следующим:

* вход в систему как пользователь;
* вход в систему как администратор;
* добавление новых пользователей;
* редактирование информации о пользователях;
* просмотр информации обо всех имеющихся дашбордах. (Дашбордами называются отчеты, данные в которых могут пересекаться);
* просмотр анализа по покупателям торговой сети;
* просмотр анализа по проданным продуктам;
* просмотр анализа продаж за 2008-2012 года;
* просмотр анализа по странам продаж;
* выход из системы.

Данный функционал наглядно отображен на диаграмме вариантов использования на рисунке 3.1.



Рисунок 3.1 – Диаграмма вариантов использования системы

Таким образом, разработанные диаграммы вариантов использования системы, архитектура приложения и программные средства позволят предприятию сократить сроки разработки системы поддержки принятия решений на основе анализа продаж в торговой сети и повысить производительность системы и эффективность работы с информацией.

**3.3 Информационная модель системы поддержки принятия решений на основе анализа продаж в торговой сети и её описание**

Для анализа информации о продажах необходимо обрабатывать не просто большие, а огромные объемы данных, поэтому обычная комплексная система, не подходит, так как выборка данных из такой системы нецелесообразна, для этого требуется большое количество времени, потому что сама система предназначена только для быстрого ввода данных. Такая система отлично будет использоваться для записи информации о конкретных продажах. Нам же необходима система, которая сможет быстро выбирать необходимые для анализа данные, что выражается в скорости выборки данных или производительности системы. Поэтому, с учетом всего выше сказанного, целесообразней будет использовать хранилище данных, которое отвечает всем требованиям производительности.

Хорошая производительность системы достигается за счет денормализации данных, выражающееся в архитектуре витрин данных (дата мартов).

Данные должны пройти все уровни хранилища данных, как было показано на рисунке 2.3 пункт 2.2, поэтому процесс ETL является важным звеном в процессе анализа и будет описан в пункте 3.4.

Хранилища состоят из множества таблиц. Большинство из них непосредственно связаны с предметной областью (таблицы фактов и измерений), но есть такие, которые хранят историю изменений, произошедших в базе. В хранилищах могут также содержаться таблицы с агрегированными данными, которые необходимы, если в таблицах фактов хранятся данные с очень маленькой детализацией по времени. Детализация по времени, так называемая гранулярность таблицы фактов, может быть час, день, месяц, год и т.д. Если гранулярность таблицы фактов по часам, то для выборки данных из такой таблицы, могут понадобиться часы, ведь в большинстве случаев необходимо просмотреть информацию по операциям, произошедших за месяц. Конечно все зависит от требований анализа, но хорошей практикой является создание таблиц, где уже хранятся пред агрегированные данные, которые наиболее часто используются в запросах к базе.

Как уже было написано выше основными составляющими структуры многомерной модели являются таблица фактов и таблицы измерений.

В данном проекте нет необходимости в создании «backup» таблиц и пред агрегированных таблиц. Гранулярность таблиц фактов будет по дням. Соответственно измерение времени будет содержать непрерывные данные по каждому дню.

Таблица фактов является основной таблицей хранилища данных. Как правило, она содержит сведения об объектах или событиях, совокупность которых будет в дальнейшем анализироваться, а также ссылки на таблицы измерений.

В данном проекте целесообразнее будет создать две таблицы фактов, так как совместить факты, которые были выделены в разделе первом не представляется возможным. Такая ситуация происходит, когда между фактами нет никаких зависимостей, т.е. они показывают совершенно разные факты. В нашем случае они зависимы, но при совмещении таблица фактов разрослась бы до огромных размеров, что замедлило бы анализ.

Первая таблица фактов содержит информацию о продажах товара: выручку от реализации, количество проданного товара, уникальные идентификаторы измерений.

Вторая таблица фактов информация о стоимости товара: себестоимость товара и цена для продажи по каждому продукту в зависимости от времени, а также уникальные идентификаторы измерений.

Так же для более качественного анализа создадим механизм отслеживания изменений в таблицах, с помощью медленно меняющихся измерений второго типа (SCD Type 2), про которые было описано в пункте 2.2. Медленно меняющиеся измерения второго типа (SCD Type 2) позволяет отслеживать неограниченное число изменений, что обеспечивает актуальность данных в любой момент времени. Такой механизм реализован для таблиц измерений «Products» («Продукты») и «Customers» («Покупатели»), так как важно знать какие происходили изменения в имеющихся данных о покупателях и продуктах с течением времени. Остальные таблицы измерений содержат неизменяемые либо редко изменяемые данные.

Рассмотрим подробнее сущности, представленные в нашей модели Data Mart, так как пользователь будет видеть только данные из витрины данных (Data Mart).

Таблицы измерений:

«Times» («Время»), данное измерение содержит непрерывные данные о днях, то есть подробное описание каждого дня, начиная с 2002 до 2013 года. Содержит атрибуты:

* конкретная дата, являющаяся натуральным ключом (Time\_Id),
* наименования дня (понедельник, вторник и т.д.) (Day\_Name),
* номер дня в месяце (Day\_Number\_In\_Month),
* номер дня в недели (Day\_Number\_In\_Week),
* номер месяца в году (Calendar\_Month\_Number),
* количество дней в месяце (Days\_In\_Cal\_Month),
* название месяца (январь, февраль и т.д.)(Calendar\_Month\_Desc),
* год(Calendar\_Year),
* количество дней в году (Days\_In\_Cal\_Year).

«Promotions» («Продвижение»), содержит информацию о продвижении товара, то есть каким образом этот товар рекламировался, продвигался. Категория продвижения показывает с помощью чего достигался результат, например, газета или телевидение и т.д. Подкатегория показывает конкретные действия в отношении продвижения, то есть если категорией продвижения является телевидение, то подкатегория может быть программа на телевидении и т.д.

Атрибуты данного измерения:

* идентификатор продвижения (Promo\_Id),
* название продвижения (Promo\_Name),
* название категории продвижения (Promo\_Category),
* идентификатор категории продвижения (Promo\_Category\_Id),
* название подкатегории продвижения (Promo\_Subcategory),
* идентификатор подкатегории продвижения (Promo\_Subcategory\_Id),
* стоимость продвижения (Promo\_Cost),
* дата начала продвижения (Promo\_Begin\_Date),
* дата окончания продвижения (Promo\_End\_Date).

«Channels» («Каналы»), измерение содержит информацию о том как товар был продан, то есть это были прямые или непрямые продажи, через каталог, партнеров, а может с помощью интернета.

Измерение содержит:

* идентификатор канала (Channel\_Id),
* описание канала (Channel\_Desc),
* класс канала (Channel\_Class),
* идентификатор класса канала (Channel\_Class\_Id),

«Products» («Товары»), описывает информацию о товарах, которые были проданы, содержит следующие поля:

* идентификатор товара (Prod\_Id),
* название товара (Prod\_Name),
* описание товара (Prod\_Desc),
* категория товара (Prod\_Category),
* описание категории товара (Prod\_Category\_Desc),
* идентификатор категории товара (Prod\_Category\_Id),
* подкатегория товара(Prod\_Subcategory),
* описание подкатегории товара (Prod\_Subcategory\_Desc),
* идентификатор подкатегории товара(Prod\_Subcategory\_Id),
* весовая категория товара (Prod\_Weight\_Class),
* цена за единицу товара (Prod\_List\_Price),
* минимальная цена (Prod\_Min\_Price).

Поля, отвечающие за механизм отслеживания изменений:

* с какой даты данная запись актуальна (Prod\_Eff\_From),
* с какой даты данная запись актуальна (Prod\_Eff\_To),
* идентификатор валидность записи (Prod\_Valid).

«Customers» («Покупатели»), данное измерение содержит информацию обо всех покупателях, когда-либо купивших товар.

Атрибуты данного измерения:

* идентификатор покупателя (Cust\_Id),
* имя покупателя (Cust\_First\_Name),
* фамилия покупателя (Cust\_Last\_Name),
* пол покупателя(Cust\_Gender),
* дата рождения (Cust\_Year\_Of\_Birth),
* статус покупателя (Cust\_Marital\_Status),
* адрес (Cust\_Street\_Address),
* почтовый индекс (Cust\_Postal\_Code),
* город (Cust\_City),
* идентификатор города (Cust\_City\_Id),
* внешний идентификатор измерения «Countries» («Страны») (Country\_Id),
* провинция (Cust\_State\_Province),
* идентификатор провинции (Cust\_State\_Province\_Id),
* телефон (Cust\_Main\_Phone\_Number),
* доход (Cust\_Income\_Level),
* кредит (Cust\_Credit\_Limit),
* электронный адрес (Cust\_Email).

Поля, отвечающие за механизм отслеживания изменений:

* с какой даты данная запись актуальна (Cust\_Eff\_From),
* с какой даты данная запись актуальна (Cust\_Eff\_To),
* идентификатор валидность записи (Cust\_Valid).

«Countries» («Страны»), содержит информацию о странах, в которые продаются товары, а также о различных регионах земного шара. Это измерение содержит иерархию по странам и регионам, поэтому для более эффективной выборки из этого измерения сделаем его иерархическим, что выражается в архитектуре измерения.

Атрибуты данного измерения:

* идентификатор страны (Country\_Id),
* название страны, (Country\_Name)
* код региона (Country\_Region\_Id),
* название региона (Country\_Region),
* код субрегиона (Country\_Subregion\_Id),
* название субрегиона (Country\_Subregion).

Таблицы фактов.

«Sales» («Продажи»), содержит аддитивные метрики стоимость заказа и количество проданных товаров и внешние ключи (идентификаторы измерений «Times» («Время»), «Customers» («Покупатели»), «Products» («Товары»), «Channels» («Каналы») и «Promotions» («Продвижение»)).

«Costs» («Стоимости»), содержит неаддитивные метрики себестоимость товара и цена для продажи по каждому продукту в зависимости от времени и внешние ключи (идентификаторы измерений «Times» («Время»), «Customers» («Покупатели»), «Products» («Товары»), «Channels» («Каналы») и «Promotions» («Продвижение»)).

Как было написано выше, данные в таблицах фактов хранятся по дням, это значит, что, если есть идентичные продажи, совершенные в один день, то сумма по этим продажам будет просуммирована и в результате получится одна строка с данными, а не, например, пять.

Модель хранилища, описанного выше, отражена на рисунке 3.2.

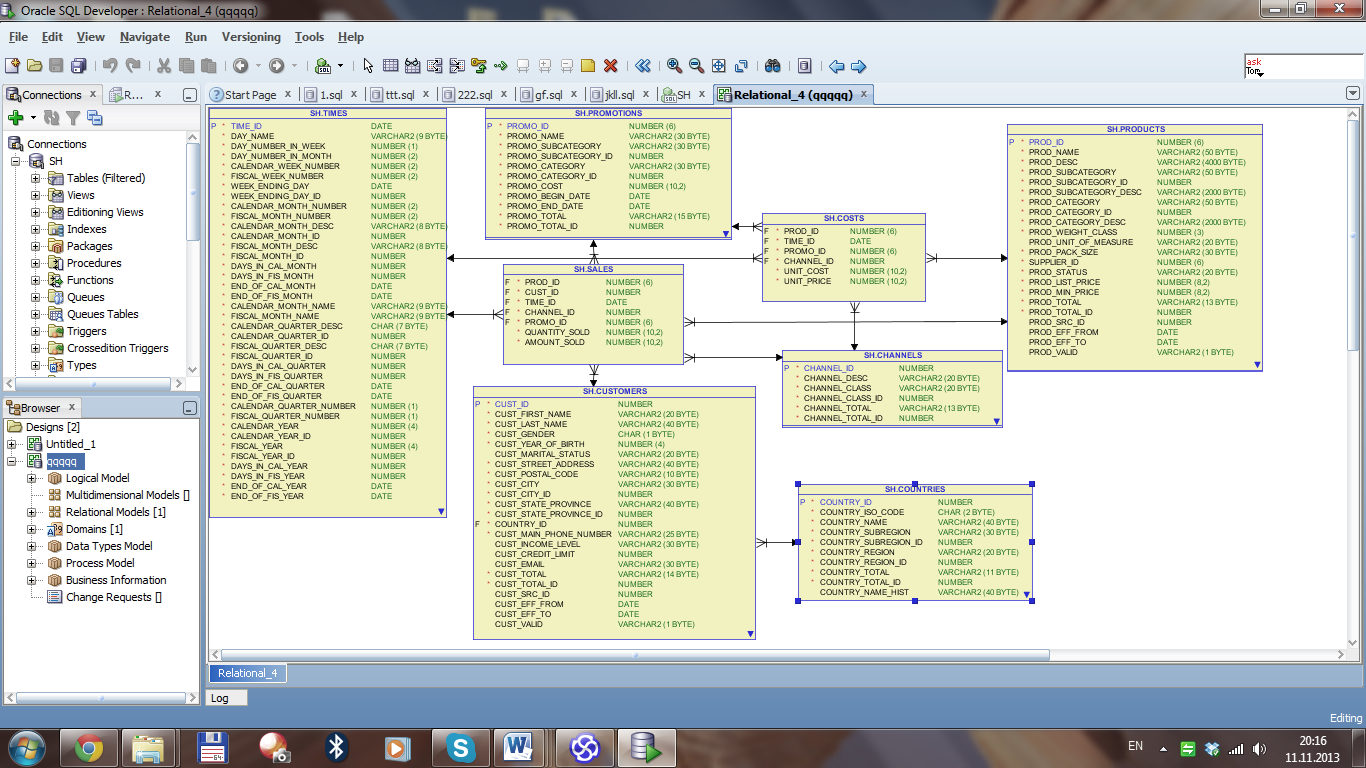


Рисунок 3.2 – Информационная модель хранилища данных

**3.4 Описание процесса ETL**

Данные и структура всех уровней хранилища (рисунок 2.3, пункт 2) перегоняются, генерируются, транслируются, очищаются и связываются друг с другом посредством специальных скриптов (процедур, запросов) написанных на языке запросов SQL.

Данные от заказчика приходят в виде файлов, поэтому для начала необходимо обработать приходящие файлы. Алгоритм обработки показан на рисунке 3.3.

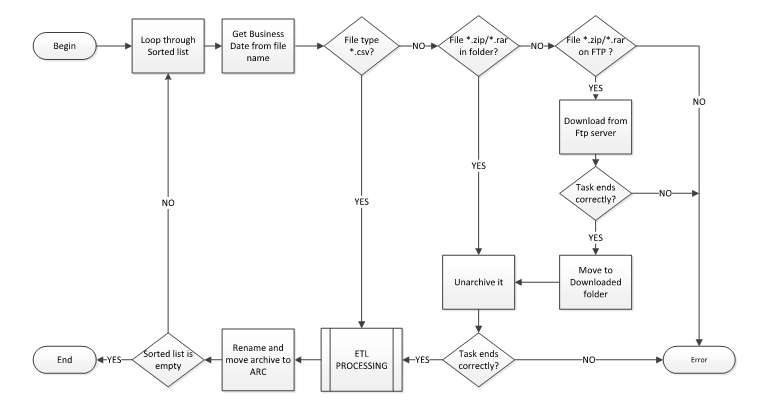


Рисунок 3.3 – Алгоритм обработки файлов, приходящих от заказчика

Из рисунка видно, что если файл в подходящем формате, то начинается ETL PROCESSING. Это значит, что запускаются написанные пакеты, состоящие из процедур. Подробное описание процесса запуска пакетов изображено на рисунке 3.4.

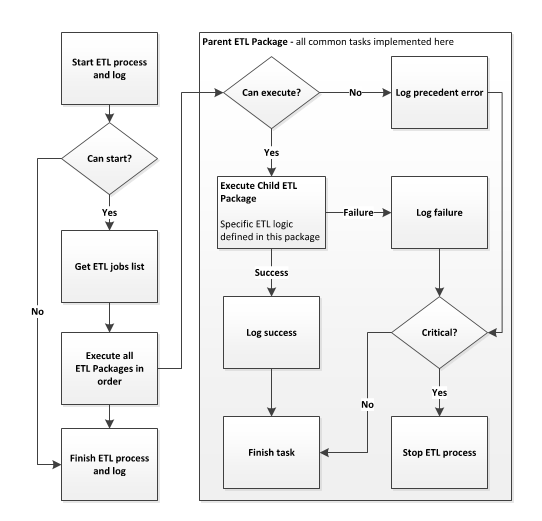


Рисунок 3.4 – Процесс запуска пакетов

После запуска пакетов, данные попадают промежуточную область, а затем – в Stage Area. Такой же процесс запуска «переливает» данные из Stage Area непосредственно в Хранилище данных, а затем в витрины данных (Data Marts), но с другими наборами пакетов соответственно.

Данные из источников данных (файлов) загружаются в промежуточную область со связью 1 к 1, это означает, что сколько было файлов – столько и будет таблиц в промежуточной области, также происходит валидация и на таблицы «навешиваются» бизнес-правила (дата загрузки, бизнес-даты и т.д.)

В наборе пакетов перегоняющих данные в Stage Area происходят разбиение таблиц из промежуточной области, происходит нормализация данных, валидация данных, а также проверки данных на соответствие бизнес-требованиям, строятся связи.

Для перегрузки данных из Stage Area в хранилища происходят более тяжелые трансформации обычных таблиц в таблицы фактов и измерений, происходит связывание таблиц, выставляются ключи (первичные, суррогатные, внешние), проверяется согласованность данных, а также происходит непростой процесс отслеживания измерений. Механизм отслеживания изменений для измерений второго типа показан на рисунке 3.5.

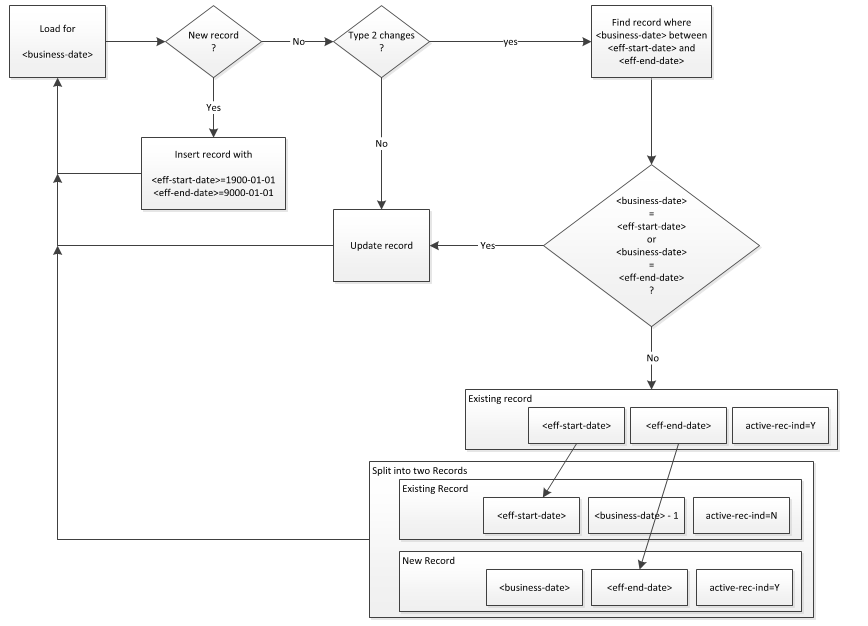


Рисунок 3.5 – Механизм отслеживания изменений SCD 2

В промежуточной области и в Stage Area хранятся только актуальные на данный момент данные, то есть какие данные в файлах на данный момент времени приходят от заказчика, такие и хранятся в этих областях.

Данные в хранилище данных нормализованы, данные накапливаются и никогда не удаляются.

В Data Mart данные перегружаются для денормализации без каких-либо особых трансформаций, кроме, естественно, проставления связей. Денормализованная структура обеспечивает быстроту выборки данных для дальнейшего анализа из системы.

Весь процесс загрузки данных в хранилище данных показан на рисунке 3.6.

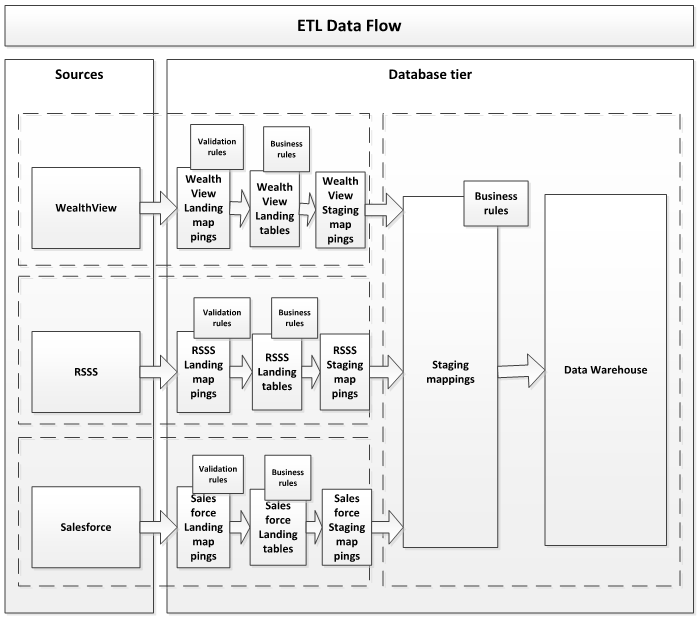


Рисунок 3.6 – Процесс загрузки данных в хранилище данных

**3.5 Модели представления системы поддержки принятия решений на основе анализа продаж в торговой сети и их описание**

Различные представления системы описываются с помощью диаграмм.

Воспользуемся следующими диаграммами:

* диаграмма классов (class diagram);
* диаграмма последовательности (sequence diagram);
* диаграмма состояний (statechart diagram);
* диаграмма компонентов (component diagram);
* диаграмма развертывания (deployment diagram).

Разнообразие диаграмм позволяет отразить различные аспекты системы.

Один такой аспект проектируемой системы, а именно варианты использования системы, был описан выше. Опишем остальные.

**Диаграмма развертывания** показывает конфигурацию обрабатывающих узлов системы и размещенных в них компонентов. Кроме того, диаграмма развертывания показывает наличие физических соединений – маршрутов передачи информации между аппаратными устройствами, задействованными в реализации системы.

Построенная система состоит из сервера базы данных, браузера пользователя и контейнера сервлетов ApacheTomcat. Сервер базы данных представлен версией Oracle Database 11g. Контейнер сервлетов ApacheTomcat содержит приложения Sales.war, откуда и отправляются необходимые запросы на сервер базы данных. Браузер связан по протоколу HTTP с ApacheTomcat, который в свою очередь по протоколу TCP/IP соединяется с сервером базы данных.

Все это отображено на диаграмме развертывания, которая представлена в Приложении Б.

**Диаграмма последовательности –** это диаграмма, на которой показаны взаимодействия объектов, упорядоченные по времени их проявления.

Диаграмма последовательности представлена примере формирования сравнительного анализа продаж по годам.

Пользователь кликает по ссылке «СравнительныйАнализ». В приложении вызывается функция построения команды buildCommand(), в этой функции с помощью паттерна Command вызывается функция performed(), которая в свою очередь осуществляет запрос в базу через функцию getYearAnalis(), которая возвращает результат запроса. При успешном выполнении всех этих функций перед пользователем в браузере появится отчет.

Диаграмма представлена в приложении В.

**Диаграмма классов**

В диаграмме классов разрабатываемой системы анализа продаж в торговой сети представлены использованные паттерны проектирования. Этими паттернами являются паттерн Command (рисунок Г.1), паттерн FactoryMethod (рисунок Г.1) или Factory и паттерн DAO (рисунок Г.2). Более подробно об используемых паттернах будет описано в разделе 6.

Диаграмма классов представлена в приложении Г.

**Диаграмма состояний**

В приложении Д (рисунок Д.1) приведена диаграмма состояний, которая показывает, в каких состояниях может находиться система или ее составная часть и какие действия вызывают изменение ее состояния.

Были выделены основные состояния в принципе самого важного объекта как отчет о каких-то данных или по-другому анализ. Отчет (анализ) может быть в трех основных состояниях: найден, выбран и получен. То есть пользователь из ссылок на разные отчеты ищет ту, которая ему необходима в данный момент. После того как ссылка найдена пользователь кликает на эту ссылку, следовательно отчет выбран. Далее вызываются последовательно различные функции, которые подключаются к базе данных и проводят выборку данных согласно запрашиваемому запросу. После всех манипуляций по выводе данных отчет/анализ отображается в браузере. Анализ получен.

**Диаграмма компонентов** описывает особенности физического представления системы. Диаграмма компонентов позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами.

Разрабатываемое приложение состоит из веб-приложения Sales.war, хранилища данных Sales и драйвера JDBC Oracle – ojdbc6.jar. Веб-приложение состоит из:

* модели (Model), которая представляет собой структуру хранилища данных Sales;
* представления(View), это jsp-страницы, где отображаются данные с помощью css-файлов и js-файлов;
* контроллера(Controller), который использует модель и представление для реализации необходимой реакции системы на действие пользователя;
* команды(Command), которая реализует паттерн проектирования Command.

Данная диаграммы изображена на рисунке Е.1 приложения Е.

* 1. **Описание работы системы поддержки принятия решений на основе анализа продаж в торговой сети и оценка выполнения задач**

Для работы приложения его необходимо запустить, то есть пройти по ссылке на страницу аутентификации пользователя. На данной странице пользователю необходимо ввести логин и пароль. В зависимости от введенной комбинации логина и пароля клиент может войти в систему в качестве Пользователя или Администратора. Права администратора отличаются от прав Пользователя возможностью добавления новых Пользователей и редактированию информации о них. Если вводятся некорректные данные, выводятся сообщения об ошибках.

Страница входа в систему представлена на рисунке 3.7.

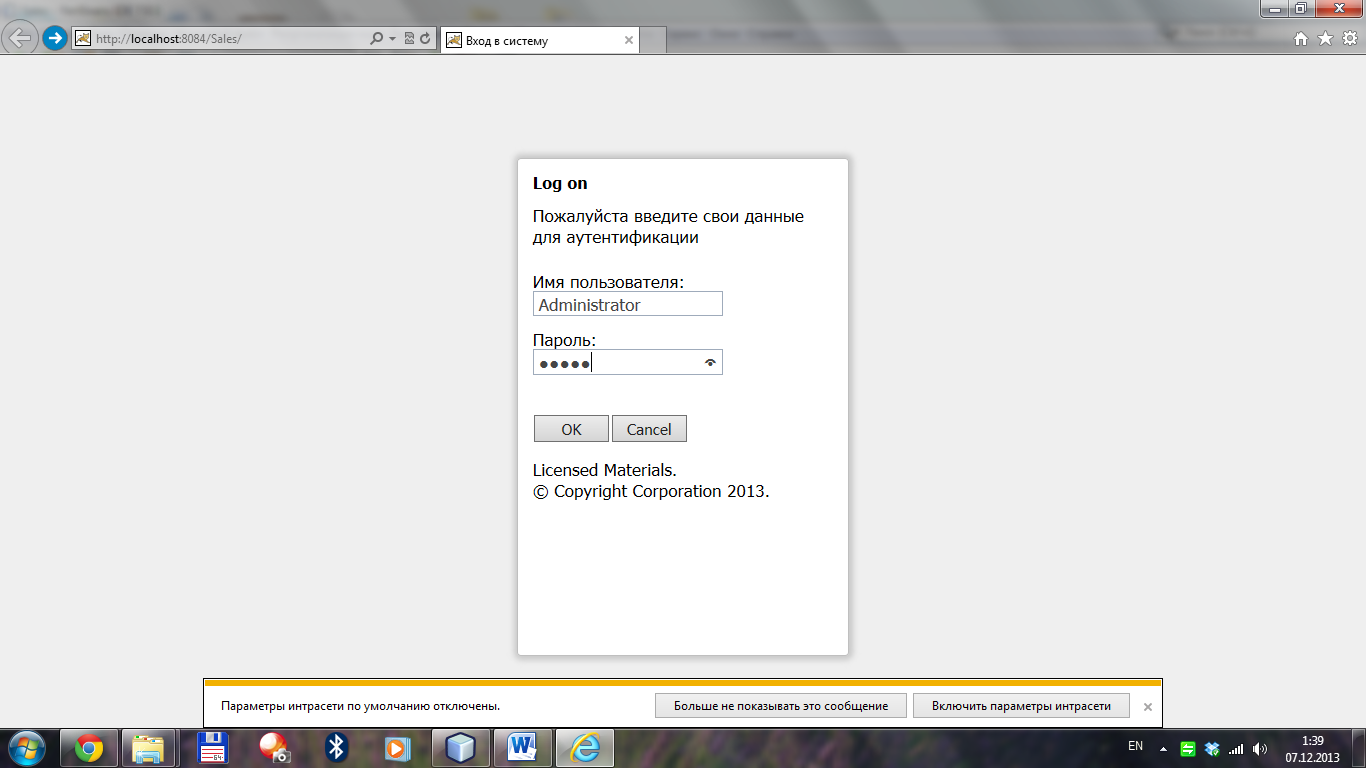


Рисунок 3.7 – Страница для входа в систему

Рассмотрим сначала возможности пользователя Администратор.

Нажимаем на ОК и появляется страница, показанная на рисунке 3.8.

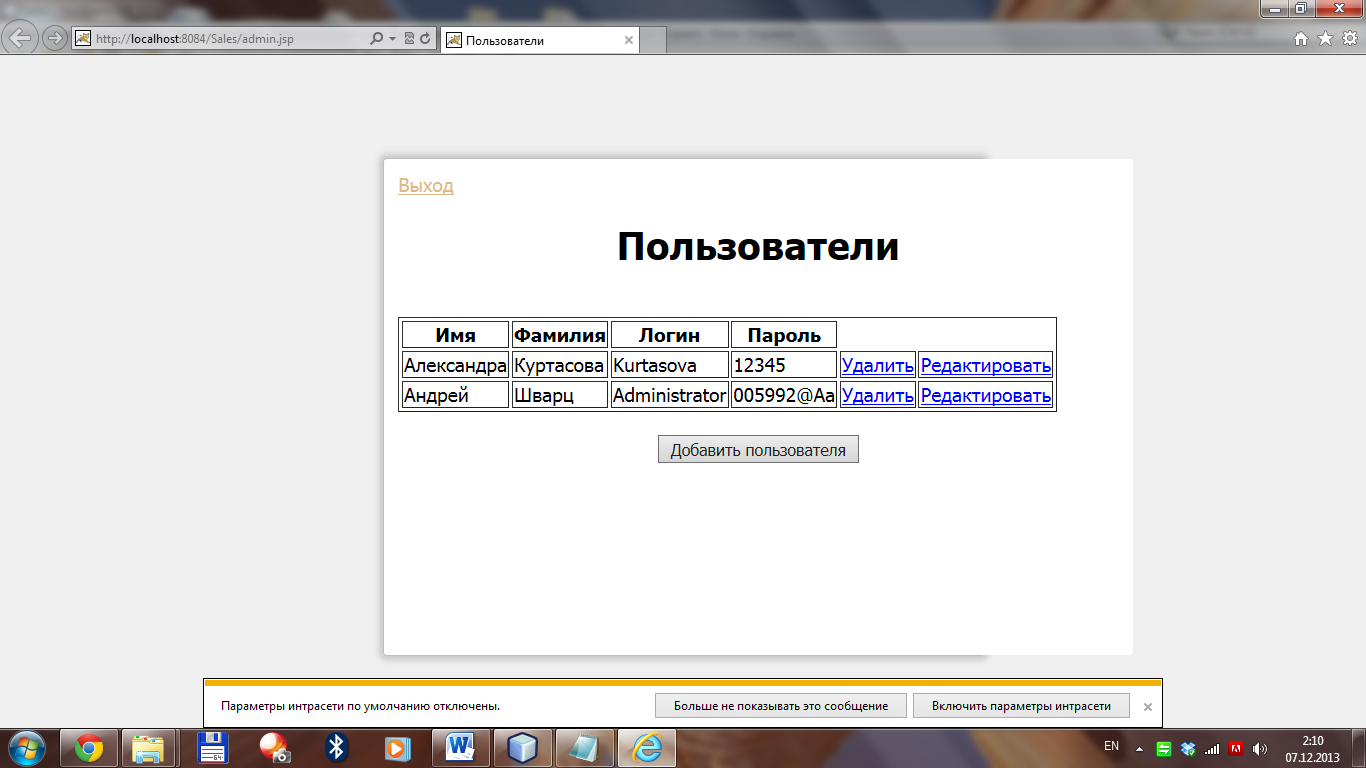


Рисунок 3.8 – Страница администратора

Как уже было написано выше администратор может добавить пользователя, для этого необходимо нажать кнопку «Добавить пользователя» и на новой странице ввести данные в поля ввода (рисунок 3.9) и нажать Добавить, введенная информация появится в таблице (рисунок 3.10); обновлять старые данные или удалять уже существующие, для этого необходимо выбрать необходимую ссылку соответствующую строке, которую нужно либо Удалить, либо Редактировать. При прохождении по ссылке редактировать появляется страница, в которой вы можете изменить данные о пользователе (рисунок 3.11). На рисунке 3.12 показаны данные о пользователях после редактирования. При выборе Удалить строка удаляется из таблицы (рисунок 3.13).

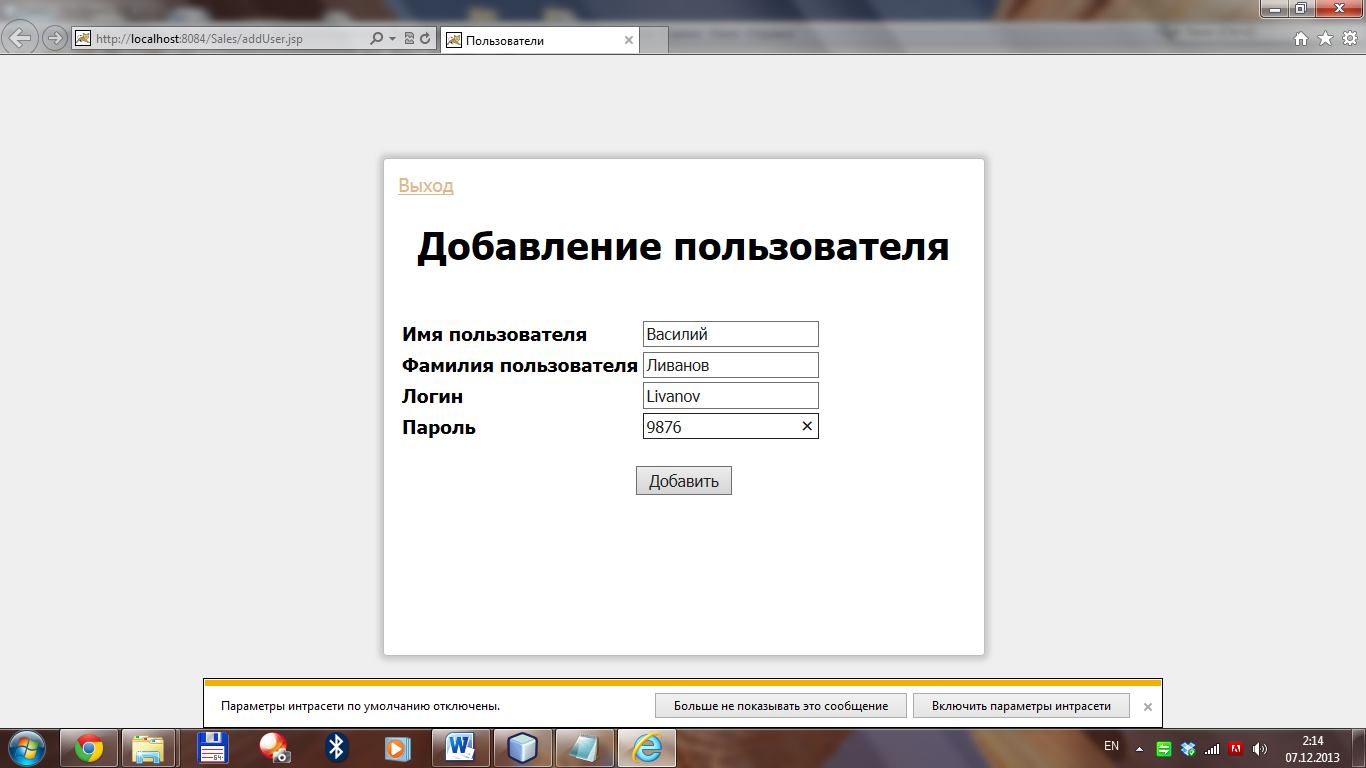


Рисунок 3.9 – Добавление пользователя

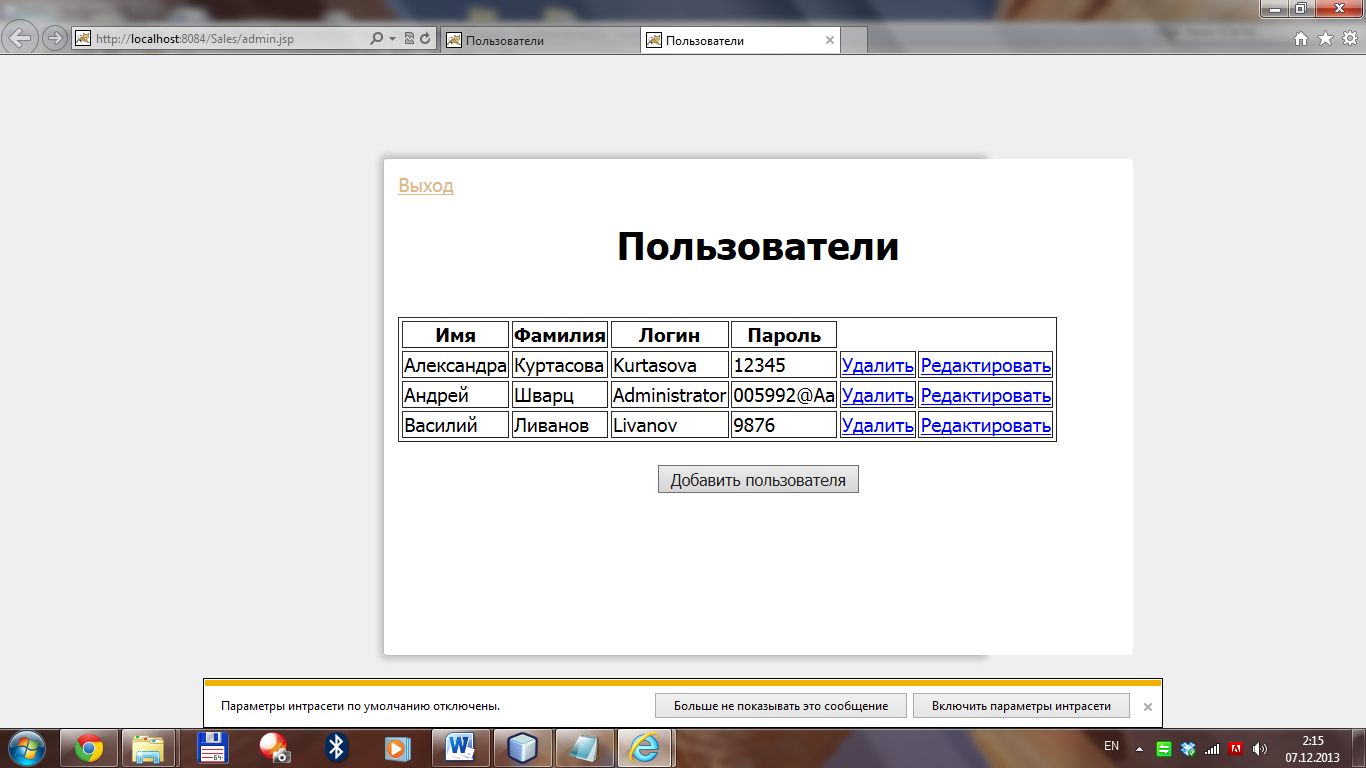


Рисунок 3.10 – Измененный список пользователей системы

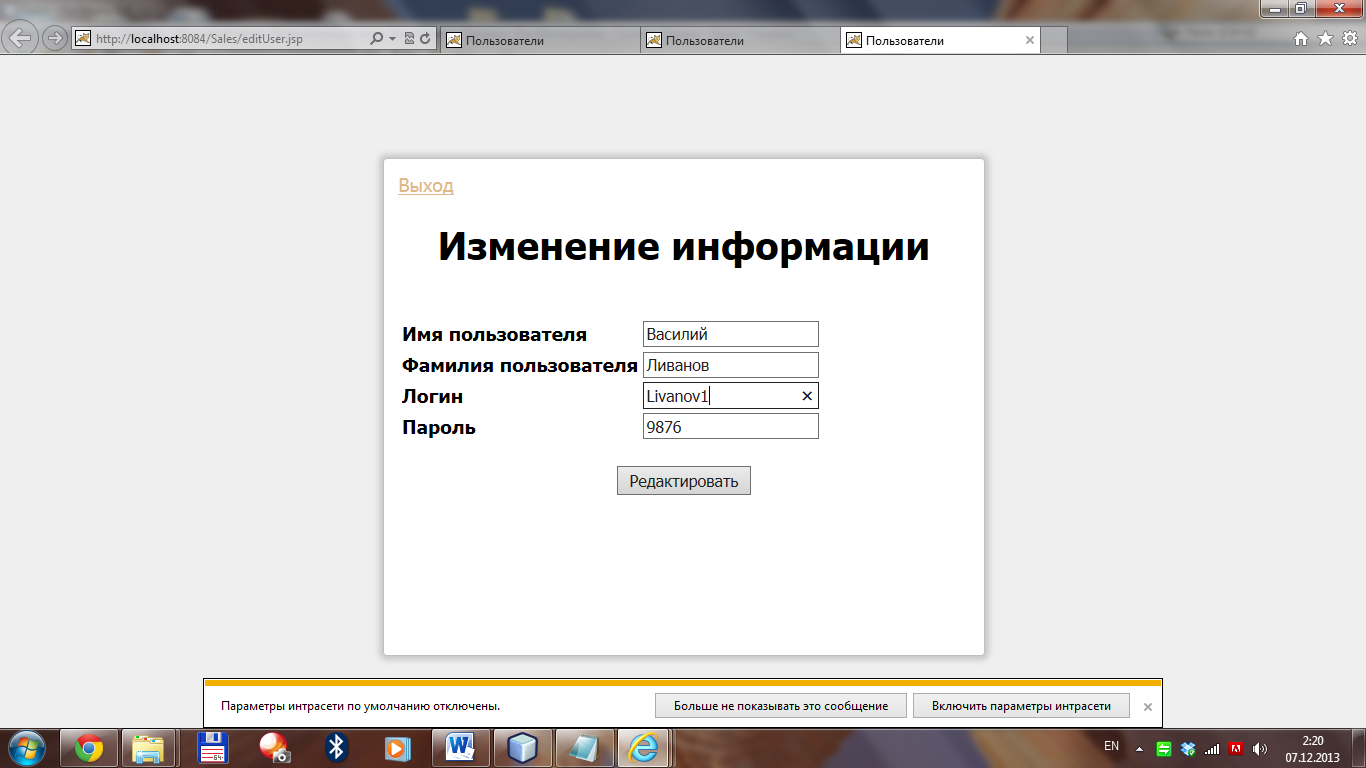


Рисунок 3.11 – Редактирование информации

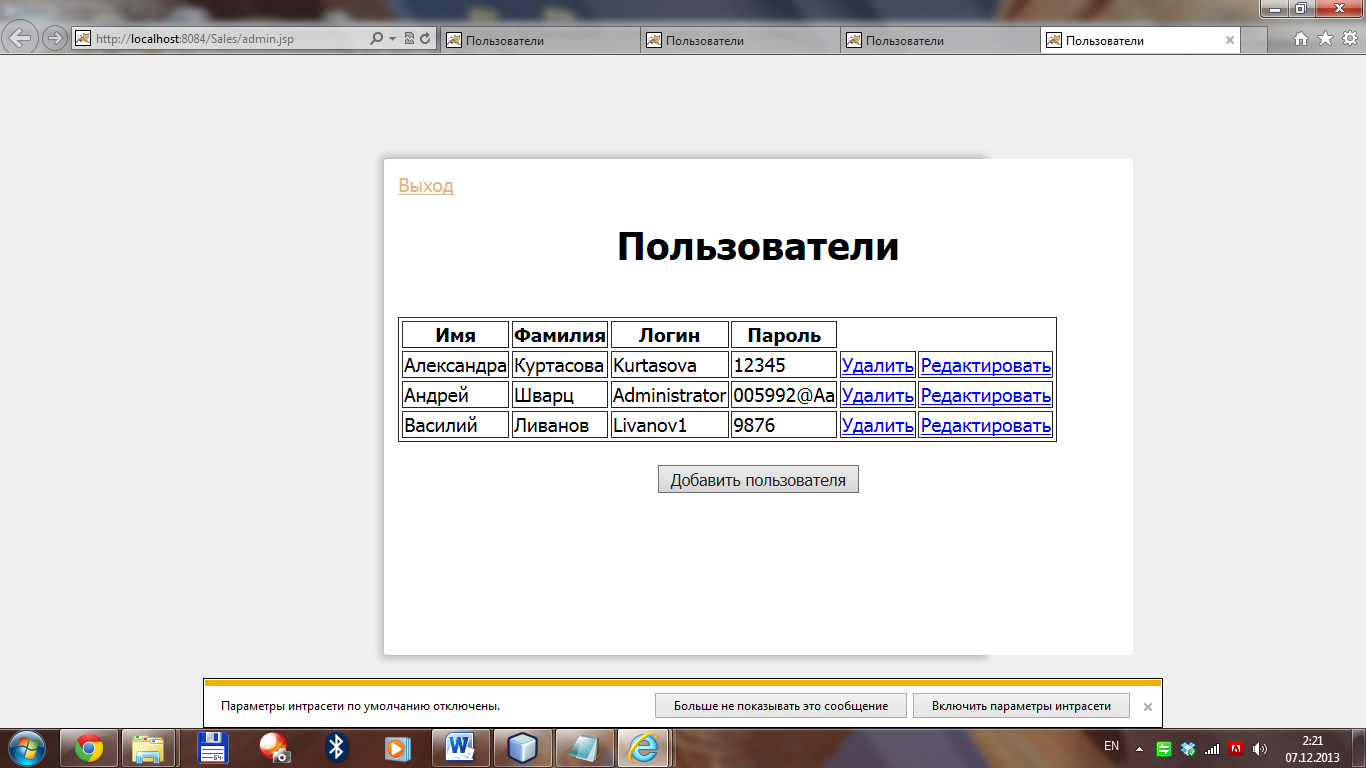


Рисунок 3.12 – Измененный список пользователей системы

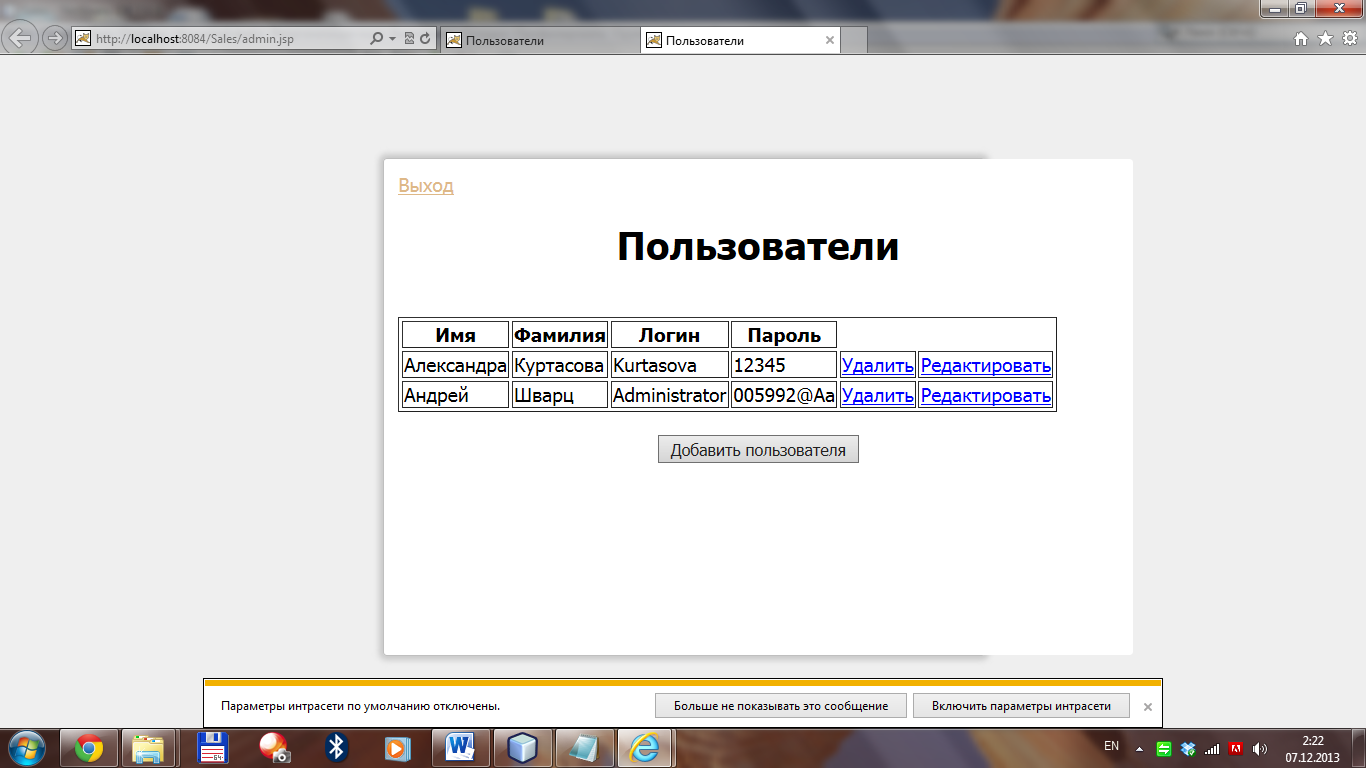


Рисунок 3.13 – Удаление пользователя

Рассмотрим возможности роли Пользователя. После ввода соответствующих логина и пароля появляется страница, показанная на рисунке 3.14, в которой Пользователь может выбрать и просмотреть необходимый ему отчеты или анализы, которые показан на рисунках 8.15-3.18.

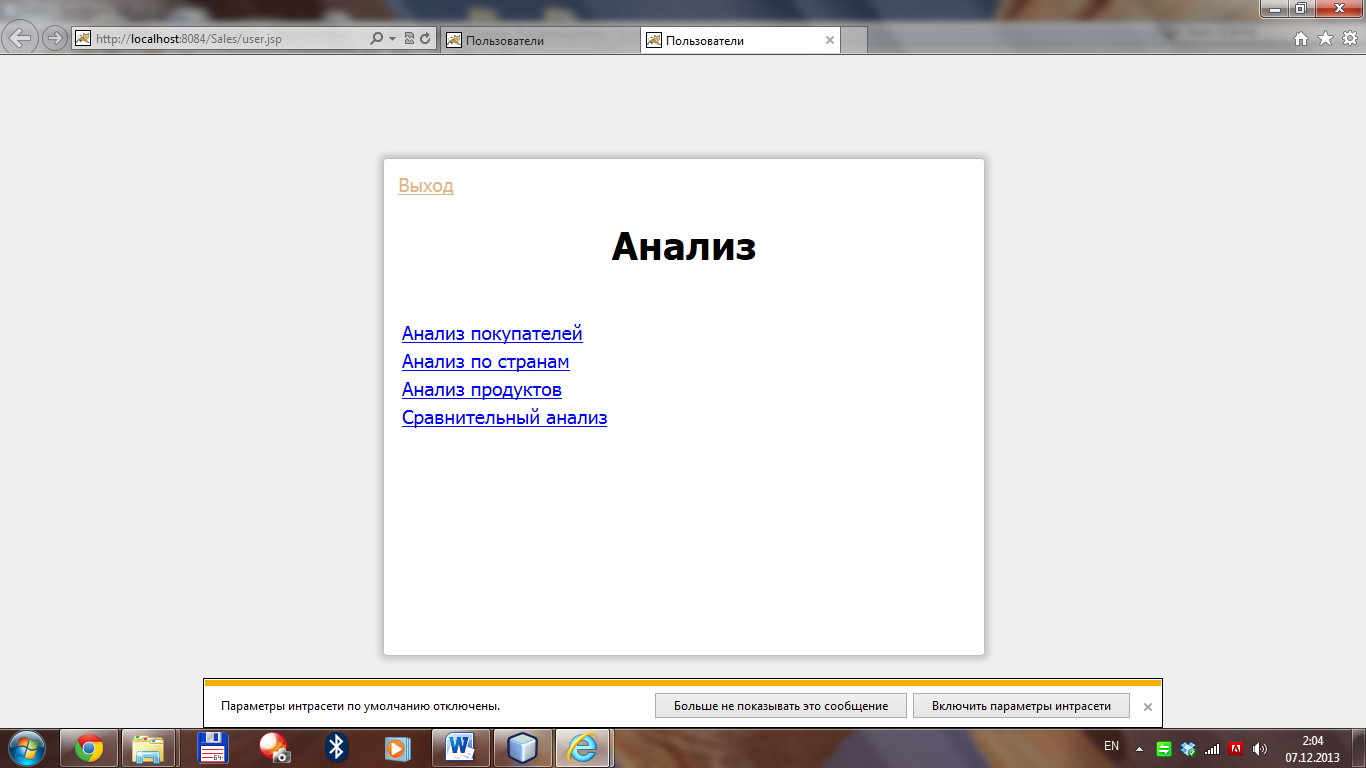


Рисунок 3.14 – Страница Пользователя

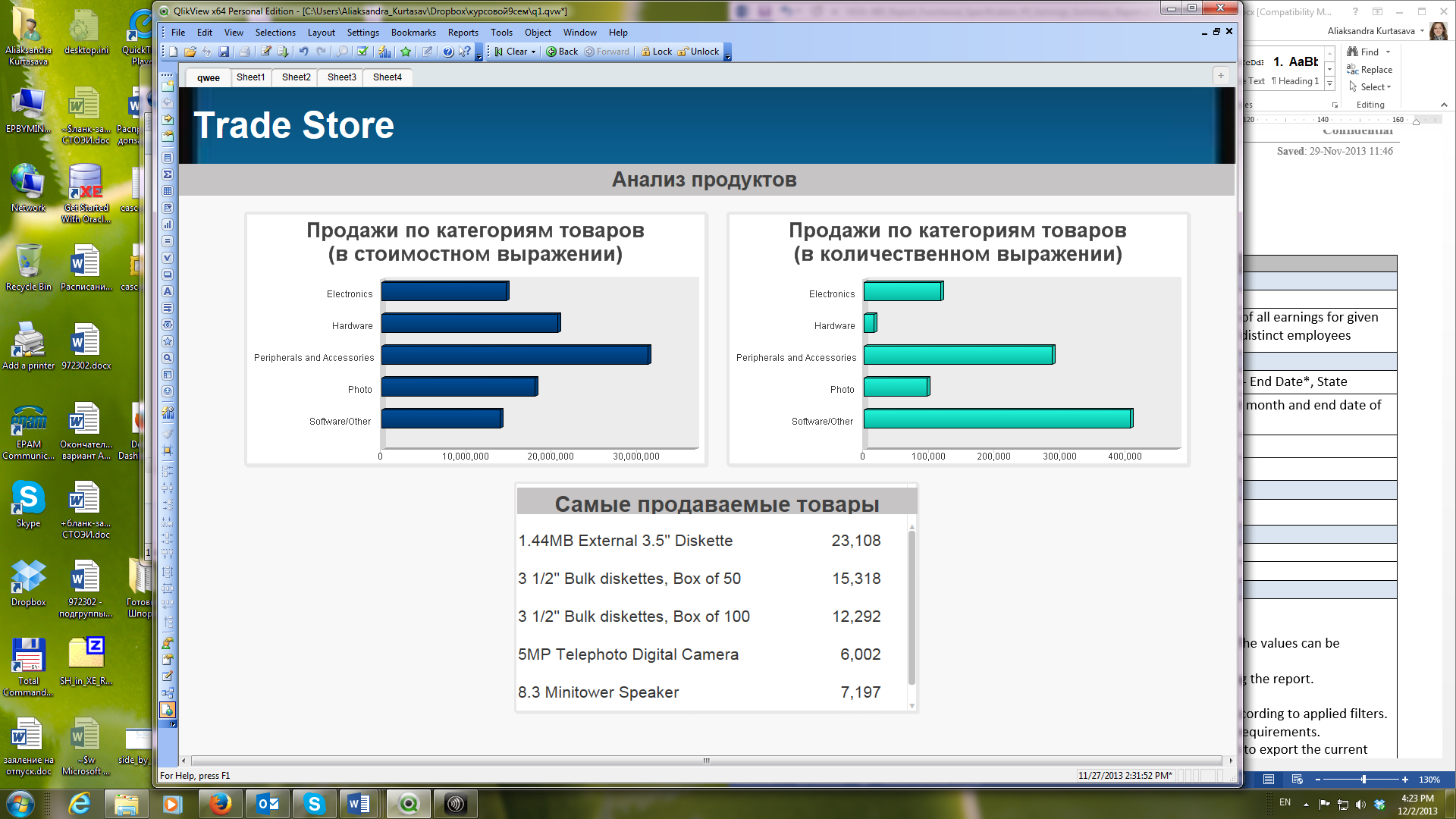


Рисунок 3.15 – Анализ продуктов

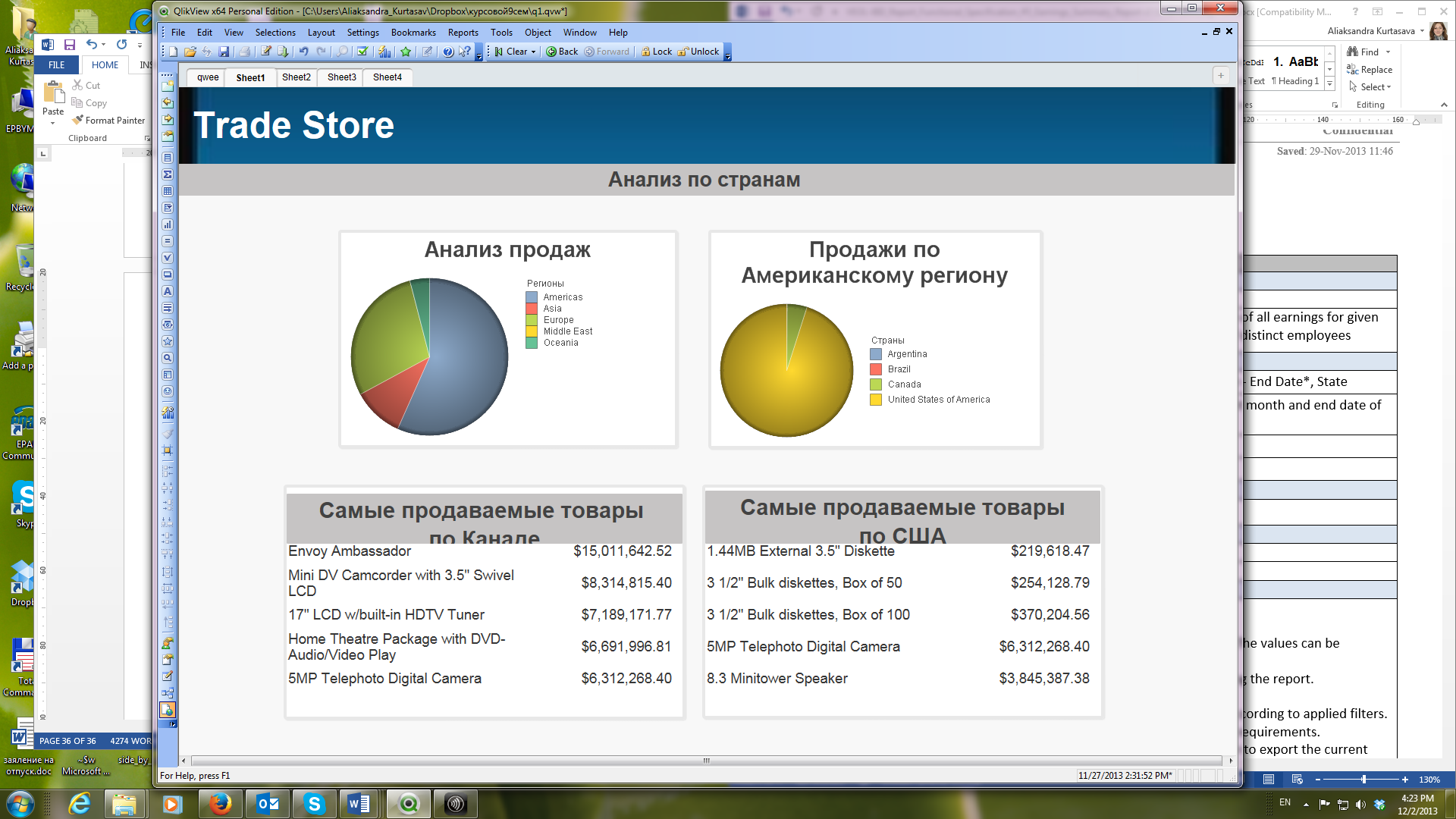


Рисунок 3.16 – Анализ по странам



Рисунок 3.17 – Сравнительный анализ

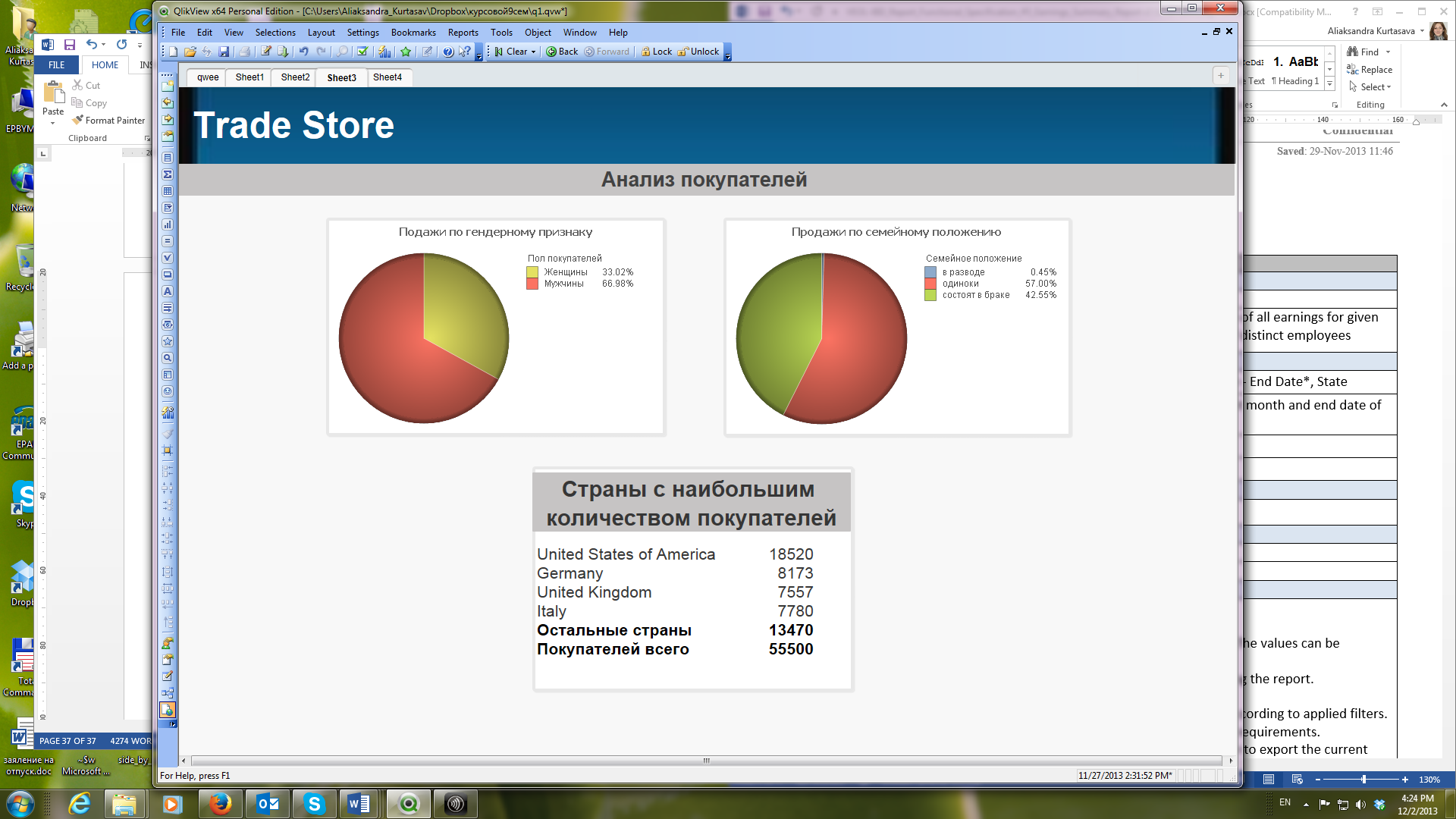


Рисунок 3.18 – Анализ покупателей

После того как пользователь закончит работу с системой, он может выйти из системы, кликнув по ссылке Выход. Система вернется к странице Входа в систему, которая была показана на рисунке 3.7.

В ходе тестирования разработанной системы был выявлен единственный момент, в котором программа вела себя некорректно. Так как при регистрации нового пользователя

При аутентификации, в случае ввода неверного логина/пароля пользователю выводится сообщение об этом. Это представлено на рисунке 9.1.

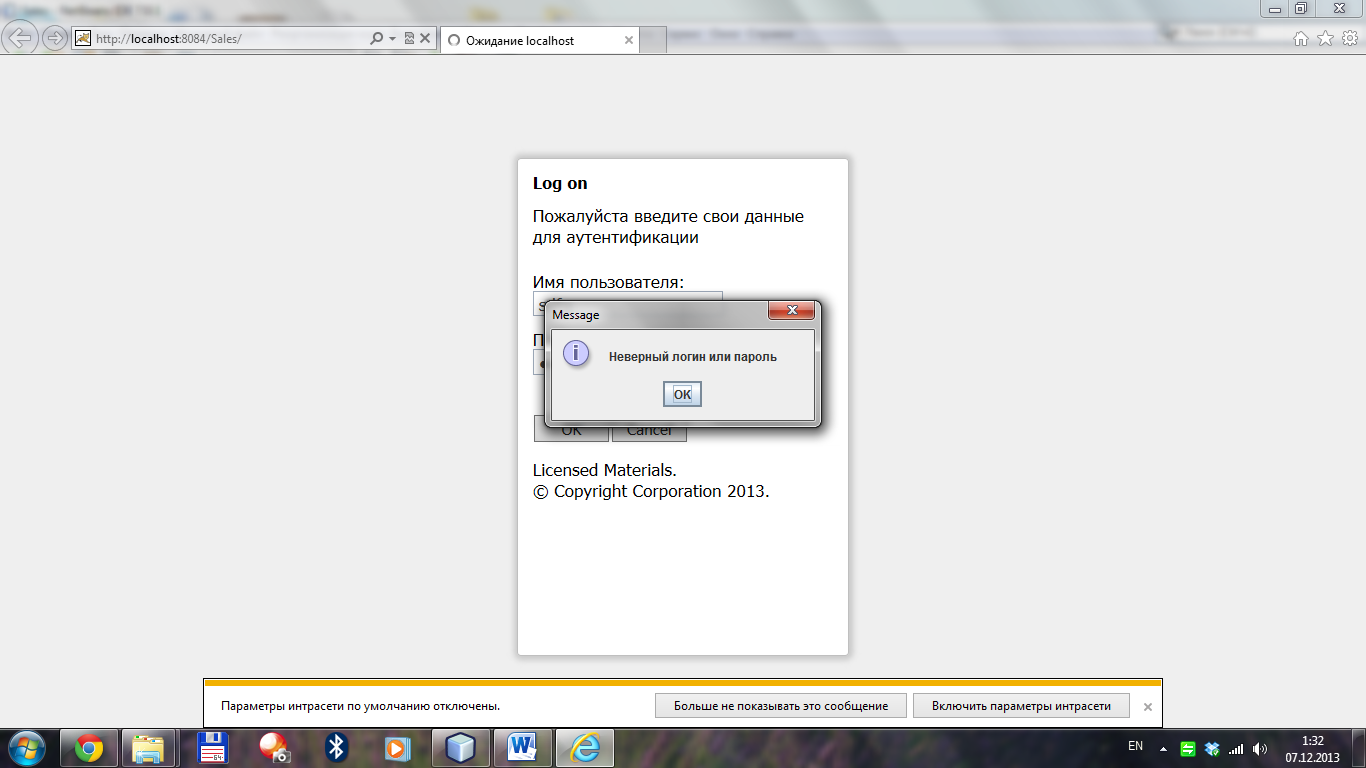


Рисунок 3.19 – Неверный ввод логина/пароля

Программа могла вести себя некорректно в случае добавления нового пользователя, но при регистрации нового пользователя никакие проверки не нужны, так как при регистрации нового пользователя вводится совершенна новая информация, которая не связана с уже имеющейся информацией.

В целом система предназначена для просмотра данных о продажах в торговой сети.

**ЗАКЛЮЧЕНИЕ**

В ходе дипломного проектирования были собраны сведения о существующих методах анализа данных, а также современных подходах к обработке данных и анализ представленных на рынке систем поддержки принятия решений.

Было выявлено, что на рынке программного обеспечения наблюдается большое количество BI-инструментов, а также готовых систем поддержки принятия решений для компании. Однако подавляющая масса программных средств предназначена для крупных компаний с огромными объемами данных и не удовлетворяет в полном объеме потребностям и целям малого бизнеса. И стоимость таких программных средств достаточно велика. Поэтому был сделан вывод, что разработку системы поддержи принятия решений необходимо проводить в соответствии с потребностями конкретного предприятия, в частности на основе анализа продаж торговой сети.

В процессе анализа деятельности коммерческой организации были выявлены особенности работы IT-компании, определены основные требования к системам поддержки принятия решений: предоставление доступа к данным предприятия, предоставление отчетности, высокая производительность системы, выражающаяся в правильной архитектуре, контроль за изменением данных.

На основании выдвинутых требований к системе была определена архитектура приложения, выбрана технология для их разработки. Определена специфика торговой сети. Построена информационная модель представления системы. Разработаны другие модели представления системы.

Разработанная модель системы поддержки принятия решений на основе анализа продаж в торговой сети полностью удовлетворяют поставленной цели и реализуют все необходимые для этого задачи. Они позволят предприятию повысить эффективность обработки данных и провести качественный анализ продаж.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. IBM [Электронный ресурс]. – Электронные данные. – Режим доступа: http://www.ibm.com/developerworks/ru/library/ba-data-mining-techniques
2. Microsoft [Электронный ресурс]. – Электронные данные. – Режим доступа: http://msdn.microsoft.com/ru-ru/library
3. Спирли Э. Корпоративные хранилища данных. Планирование, разработка, реализация. М., 2001.
4. CI [Электронный ресурс]. – Электронные данные. – Режим доступа: http://www.ci.ru/inform05\_02/p\_04.htm
5. OLAP [Электронный ресурс]. – Электронные данные. – Режим доступа: http://www.olap.ru/basic/news/m001031988.asp
6. Проектный опыт. Аналитика и статьи. [Электронный ресурс]. – Электронные данные. – Режим доступа: http://www.prj-exp.ru/dwh/what\_is\_dwh.php
7. IBM [Электронный ресурс]. – Электронные данные. – Режим доступа: http:://www.ibm.com/.
8. Хорстман, М. Java-2. Библиотека профессионала. Том 1 / М. Хорстман . – М.:Вильямс, 2003. – 850с.

[9] Хорстман, М. Java-2. Библиотека профессионала. Том 2 / М. Хорстман . – М.:Вильямс, 2003. – 630с.

[10] Торговые сети [Электронный ресурс]. – Электронные данные – Режим доступа: <http://www.grandars.ru/college/biznes/torgovye-seti.html>.

[11] CA [Электронный ресурс]. – Электронные данные. – Режим доступа: <http://ca.com/>.

[12] Охрана труда: учебное пособие для студ. Учреждений, обеспечивающих получение высш. образования по спец. в областях радиоэлектроники и информатики / Т.Ф. Михнюк. – Минск: ИВЦ Минфина, 2007. – 320с.

[13] СанПиН 9-131 РБ "Гигиенические требования к видеодисплейным терминалам, электронно-вычислительным машинам и организации работы"

[14] Производственная санитария: лабораторный практикум по курсу «Охрана труда» для студентов всех специальностей и форм обучения БГУИР / под.общ.ред. Т.Ф.Михнюка. – Мн. : БГУИР, 2004. – 34с.

[15] Девисилов, В.А. Охрана труда: учебник. / В.А. Девсилов. – 4-е изд., перераб. и доп. – М. : ФОРУМ, 2009. – 496 с.

**ПРИЛОЖЕНИЕ А**

(обязательное)

Функциональная модель процессов предметной области



Рисунок А.1 – Контекстная диаграмма



Рисунок А.2 – Декомпозиция контекстной диаграммы

Продолжение приложения А



Рисунок А.3 – Декомпозиция блока Разработать BI-решение



Рисунок А.4 – Декомпозиция блока Спроектировать хранилище данных

Продолжение приложения А



Рисунок А.5 – Декомпозиция блока Написать ETL-процедуры

**ПРИЛОЖЕНИЕ Б**

(обязательное)

Диаграмма развертывания



Рисунок Б. 1 – Диаграмма развертывания

**ПРИЛОЖЕНИЕ В**

(обязательное)

Диаграмма последовательности



Рисунок В.1 – Диаграмма последовательности

**ПРИЛОЖЕНИЕ Г**

(обязательное)

Диаграмма классов



Рисунок Г.1 – Диаграмма паттерна Command и Factory

Продолжение приложения Г



Рисунок Г.2 – Диаграмма паттерна DAO

**ПРИЛОЖЕНИЕ Д**

(обязательное)

Диаграмма состояний



Рисунок Д.1 – Диаграмма состояний

**ПРИЛОЖЕНИЕ Е**

(обязательное)

Диаграмма компонентов



Рисунок Е. 1 – Диаграмма компонентов