GeekBrains

**Сервис для загрузки файлов и вывода списка уже загруженных файлов пользователя.**

IT-специалист:

Java-программист

Корленков Андрей Владимирович

Москва

2024

**Аннотация**

Целью данной работы является проектирование и разработка сервиса для загрузки файлов и вывода списка уже загруженных файлов пользователя.

В выпускной квалификационной работе выделены основные понятия, связанные с хранилищами данных и основные подходы к их проектированию.

В работе так же подробно изучена сфера интернет хостинга и системы, которая функционирует в существующей компании. Так же проанализирована необходимость в механизме принятия решений и выявлены основные направления принятия решений.

Оглавление

[Введение 4](#_Toc356810086)

[Глава 1. Теоретическая часть 5](#_Toc356810087)

[Цели и задачи 5](#_Toc356810088)

[Актуальность выбранной темы 6](#_Toc356810089)

[Архитектурные подходы 8](#_Toc356810090)

[OLAP-технология 9](#_Toc356810091)

[Загрузка и преобразование данных (ETL) 11](#_Toc356810092)

[Операции манипулирования Измерениями. 13](#_Toc356810093)

[Методологии проектирования 16](#_Toc356810094)

[Хранилище данных с измеренями 19](#_Toc356810095)

[Глава 2. Практическая часть 21](#_Toc356810096)

[Обоснование выбора направления 21](#_Toc356810097)

[Интернет хостинг 22](#_Toc356810098)

[Принятие решений 23](#_Toc356810099)

[OLTP система 24](#_Toc356810100)

[Проектирование хранилища данных 25](#_Toc356810101)

[Глава 3. Результаты 27](#_Toc356810102)

[Основные отчеты 27](#_Toc356810103)

[Преимущества и недостатки 29](#_Toc356810104)

[Заключение 30](#_Toc356810105)

[Список использованной литературы 31](#_Toc356810106)

# Введение

Ни для кого не секрет, что проблема хранения и обработки информации является одной из самых важных на данный момент во всем мире. Особенно актуальной эта проблема является для больших компаний, через которые за один день могут проходить тысячи и миллионы транзакций. Для проблемы хранения информации были придуманы сотни различных решений, что в результате привело к тому, что практически во всех компаниях стоят системы автоматизации повседневной деятельности – OLTP системы.

Но помимо проблемы хранения остается не менее важная часть – обработка и анализ всей информации, прошедшей через OLTP систему. Чаще всего данную проблему невозможно или затруднительно решить средствами существующих OLTP систем, так как в их основе лежат другие принципы, и они решают другие задачи. В результате информация может быть недоступна тем людям, которые отвечают непосредственно за принятие решений. Это приводит к тому, что возникает потребность в системах, которые могли бы использовать информацию из OLTP систем, не нарушая ход их работы, и при этом главной целью которых была бы поддержка принятия аналитических решений. Такими системами являются хранилища данных.

Хранилище данных – спроектированная специальным образом информационная база данных, предназначенная для подготовки отчётов и бизнес-анализа с целью поддержки принятия решений в организации. [1] На данный момент существует два основных подхода к проектированию хранилищ данных[3]. Оба эти подхода имеют как недостатки, так и достоинства и применимы в различных ситуациях. Данные подходы будут проанализированы с точки зрения того, что больше подходит для данной работы.

Процесс принятия решений является неотъемлемой частью управления любым бизнесом[2]. Ежедневно в компаниях принимаются различные решения по дальнейшему развитию и текущим операциям. Некоторые из этих решений могут целиком изменить положение компании на рынке.

Процесс принятия решений важен для любой компании, будь то огромная сеть магазинов по всему миру, или небольшой ларек. Каждый день владельцы и менеджеры решают, какие продукты им покупать, какие услуги предоставлять, какие скидки и кому предложить. Зачастую именно от того, какое решение будет принято, зависит будущее бизнеса. Правильное решение может быть выбрано исключительно исходя из актуальной информации о текущем положении дел в компании. И если для небольших компаний данная информация может быть получена и без внедрения дополнительных систем, то для большого и среднего бизнеса обработка и анализ всего объема информации вручную практически невозможна.

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

Настоящий толчок к развитию хранилища данных получили относительно недавно. Это связанно с увеличением вычислительных мощностей современных процессоров[5]. Еще совсем недавно при разработке хранилища данных приходилось искусственно «ухудшать» архитектуру хранилища, то есть делать её не настолько гибкой и более подверженной изменению желаний конечных бизнес-пользователей. Эта необходимость возникала из-за того, что вычислительных мощностей просто не хватало для полноценной работы хранилища, что приводило к многочисленным сбоям и медленной работе системы.

На данный момент эта проблема решена, и проектировщик хранилищ может не задумываться об искусственном «ухудшении» системы. Это вовсе не значит, что не стоит задумываться о быстроте и стабильности работы хранилища, но решение, которое действительно будет лучшим для конкретного случая действительно может быть внедрено и успешно использоваться не создавая дополнительных нагрузок на систему, которые в конечно итоге могут привести к её нестабильной работе.

Подводя итоги, 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. Существуют современные методики проектирования хранилищ данных, которые позволяют сделать системы более эффективными и удобными для конечного пользователя, а так же менее нуждающиеся в дополнительной доработке при изменении процессов в компании.

Итак, основной **целью** моей выпускной квалификационной работы является изучение сферы интернет хостинга, разработка хранилища данных для поддержки принятия решения в данной сфере и анализ полученных результатов.

Для решения поставленной цели в первую очередь необходимо получить понимание о существующих подходах к проектированию хранилищ данных. После этого, необходимо рассмотреть непосредственно сферу интернет хостинга и существующую в ней OLTP систему.

После этого, на основе собранной информации можно определить основные направления принятия решений в данной сфере и отчеты которые необходимы для принятия этих решений.

Следующим этапом работы будет непосредственно проектирование и разработка хранилища данных и ETL процессов для загрузки в нее данных. На основе построенного хранилища можно будет построить необходимые отчеты. В качестве среды для разработки хранилища данных была выбрана СУБД Microsoft SQL Server 2008 R, а для построения отчетов было выбрано приложение Intellinx.

Завершающим этапом будет анализ полученных результатов.

Итак, окончательный перечень работ выглядит следующим образом:

1. Изучение текущих подходов к проектированию хранилищ данных
2. Анализ основного процесса сферы интернет хостинга – покупки услуги клиентом
3. Выявление потребностей в принятии решений
4. Проектирование структуры хранилища данных
5. Разработка хранилища данных средствами Microsoft SQL Server
6. Проектирование и разработка ETL системы
7. Построение отчетов для принятия решений в Intellinx
8. Анализ полученных результатов

Таким образом, **объектом** данного исследования являются хранилища данных, а **предметом** является разработка хранилища данных для поддержки принятия решений в сфере интернет хостинга.

Основным методом исследования в данной работе выступает системный анализ выявленных проблем, с целью проектирования наиболее подходящего хранилища данных для более эффективного процесса принятия решений в компаний. Другим немаловажным методов исследования является изучение современных российских и зарубежных подходов к проектированию хранилищ данных и использование лучших методик.

Результатом данной работы являются отчеты, построенные с помощью приложения Intellinx, которые позволяют поддерживать процесс принятия решений в компании, предоставляя информацию о текущем положении дел в компании в виде, удобном для бизнес-пользователя

Научно-практическая новизна данной работы заключается в выработке подхода к проектированию хранилища данных для сферы интернет хостинга и непосредственная разработка хранилища для конкретной компании, которая работает в данной сфере. Выработанное решение может служить основой для дальнейшего проектирования хранилищ данных в сфере интернет хостинга в частности и сферы телекоммуникаций в целом.

Данная работа раскрывает тему проектирования и разработки хранилищ данных для поддержки принятия решений в бизнесе. В первой части работы описывается актуальность данной темы на данный момент, ставятся цели и задачи данной работы, а так же даются необходимые теоритические сведения о хранилищах данных.

Во второй части работы происходит анализ сферы интернет хостинга и проектируется и разрабатывается хранилище данных. В третьей части производится анализ проделанной работы. В заключении приводятся дальнейшее применении и пути развития проделанной работы.

# Глава 1. Теоретическая часть

Ключевые понятия

**База данных -** представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью ЭВМ.[14]

**Реляционная база данных –** нормализованная база данных.[14]

**Нормализация –** приведение базы данных в нормальную форму. [14]

**Нормальная форма -** свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение. [14]

**СУБД** – система управления баз данных, совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных[14].

**OLTP система -** обработка транзакций в реальном времени. Способ организации БД, при котором система работает с небольшими по размерам транзакциями, но идущими большим потоком, и при этом клиенту требуется от системы минимальное время отклика.[14]

**Хранилище данных -** предметно-ориентированная информационная база данных, специально разработанная и предназначенная для подготовки отчётов и бизнес-анализа с целью поддержки принятия решений в организации.[4]

**Схема «звезда»** - специальная организация реляционных таблиц, удобная для хранения многомерных показателей. Лежит в основе реляционного OLAP. Модель данных состоит из двух типов таблиц: одной таблицы фактов— центр «звезды» — и нескольких таблиц измерений по числу измерений в модели данных — лучи «звезды». [5]

**Схема «снежинка» -** то же самое что схема «звезда», но таблицы измерений могут соединяться с другими таблицами измерений. [5]

**ETL** - от англ. Extract, Transform, Load, извлечение, преобразование, загрузка. Один из базовых процессов управления хранилищами данных, а также наименование класса утилит автоматизации этого процесса. [6]

**Витрина данных -** срез хранилища данных, представляющий собой массив тематической, узконаправленной информации, ориентированный, например, на пользователей одной рабочей группы или департамента.[4]

**Поток данных -** абстракция, используемая для чтения или записи файлов или передачи данных в единой манере.[6]

Два подхода к хранилищам данных

На сегодняшний день существует два основных подхода к моделям Хранилищ данных. Это так называемая корпоративная информационная фабрика Билла Инмона и Хранилище данных с архитектурой шины Ральфа Кимбалла[3].

В обоих подходах есть общие моменты, в которых оба автора соглашаются друг с другом[8]. Во-первых, безусловно то, что практически любая организация только выиграет от создания хранилища данных для принятия решений. Ни одна организация не сможет существовать без полностью функционирующей OLTP системы. Точно так же нужны и дополнительные аналитические системы, которые дополняют OLTP.

Во-вторых, целью любого хранилища является хранение «правильной» информации, к которой несложно получить доступ людям, ответственным за принятие решений. Два основных компонента этой среды – подготовка данных и её представление. Процесс подготовки информации состоит из ETL процессов и процессов технической поддержки. После того, как вся необходимая информация была собрана, она загружается в область презентации данных, где возможен её последующий анализ, создание отчетов и её использование в различных аналитических приложениях.

Так же, оба подхода говорят о том, что при разработке хранилища необходимо охватить взглядом всю систему целиком. И не смотря на то, что система будет внедряться по частям, важно видеть конечную цель на этапе планирования.

И наконец, оба автора соглашаются в том, что подход, при котором различные хранилища и витрины разрабатываются отдельно и независимо друг от друга заранее неправильный. Такой подход может быть выигрышным в краткосрочном периоде, но в долгосрочном он приведет к многочисленным дублированиям и некорректности данных.

А теперь будут рассмотрены основные различия этих подходов.

### Корпоративная информационная фабрика

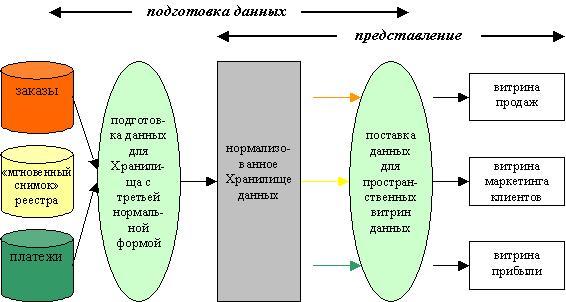


Рисунок 1. Корпоративная информационная фабрика

Как видно из рисунка, разработка данного хранилища начинает со скоординированного извлечения данных из существующей OLTP системы. После этого, данные загружаются в реляционную базу данных с третьей нормальной формой. Получившееся нормализованное Хранилище используется для того, чтобы наполнить информацией витрины данных, т.е. данных, подготовленных для анализа.

При таком сценарии конечные витрины данных создаются для обслуживания бизнес-отделов или для реализации бизнес-функций и используют пространственную модель для структурирования суммарных данных. Атомарные данные тем не менее остаются доступными через нормализованное Хранилище данных.

Итак, отличительные характеристики подхода [10]:

* Использование реляционной модели организации атомарных данных и пространственной - для организации суммарных данных.
* Использование итеративного или "спирального" подхода при создании больших Хранилищ данных, т.е. "строительство" Хранилища не сразу, а по частям. Это позволяет при необходимости вносить изменения в небольшие блоки данных или программных кодов и избавляет от необходимости перепрограммировать значительные объемы данных в Хранилище.
* В хранилище используется третья нормальная форма для атомарных данных, что обеспечивает высокую степень детальности интегрированных данных и, соответственно, предоставляет корпорациям широкие возможности для манипулирования ими и изменения формата и способа представления данных по мере необходимости.
* Хранилище данных – является целым физическим объектом, а не коллекцией витрин данных.

В данном подходе хранилище данных сразу проектируется так, чтобы удовлетворять потребности сразу всех отраслей компании, например маркетинг и продажи. Хранилище создается как единое целое, и витрина данных является не более, чем финальным этапом визуализации всей информации для пользователя.

Основные плюсы данного подхода:

* Достаточно простая реализация хранилища, так как существующая OLTP система обычно тоже нормализована.
* Быстрая доработка витрин данных

Основным минусом является достаточно долгий срок разработки хранилища (от года).

Данный подход применим скорее для очень крупных организаций.

### Хранилище данных с архитектурой шины

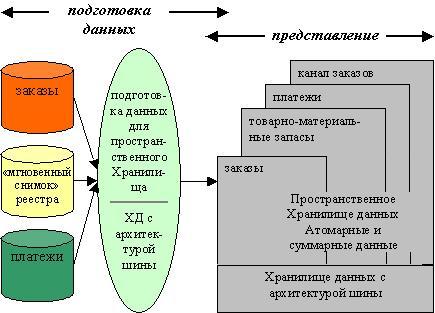


Рисунок 2. Хранилище данных с архитектурой шины

В данной модели, как и в предыдущей, первичные данные преобразуются в информацию, пригодную для использования, на этапе подготовки данных. Ключевым факторами при разработке процесса загрузки являются требования к скорости обработки информации и качеству данных.

В данном подходе вместо централизованного нормализованного хранилища используется множество витрин данных, каждая из которых спроектирована с архитектурой «звезда» или «снежинка».

Хранилище данных не является единым физическим репозиторием (в отличие от подхода Билла Инмона)[9]. Это "виртуальное" Хранилище. Это коллекция витрин данных. В данном подходе каждая витрина данных проектируется отдельно и не особо зависит от остальных. Все витрины соединяются «шиной». Такое хранилище содержит точно такую же информацию как и нормализованное, но она по-особому структурирована для более удобной работы с конечными данными.

Хранилище данных с архитектурой шины обладает следующими характеристиками[10]:

* оно пространственное
* оно включает как данные о транзакциях, так и суммарные данные
* оно включает витрины данных, посвященные только одной предметной области или имеющие только одну таблицу фактов
* оно может содержать множество витрин данных в пределах одной базы данных.

Главное достоинство данной методологии это скорость создания витрин. Первая рабочая витрина может быть создана за пару месяцев и будет полностью функциональной. Основной недостаток может возникнуть, после многочисленных и долгих по времени изменений в хранилище. Вполне вероятно, что после внесений многочисленных изменений, информация будет дублироваться в различных витринах или наоборот одни и те же показатели будут считаться по-разному. Именно поэтому для данного подхода процесс загрузки данных является ключевым и лежит в самой основе. Далее будет дано более подробное описание процесса загрузки данных.

Этот подход больше подходит для малых и средних предприятий, и именно он будет использован в данной работе.

Загрузка и преобразование данных (ETL)

ETL система является основой для хранилища данных. Правильным образом спроектированная ETL система выгружает данные из всех источников, обеспечивает целостность и качество данных, объединяет данные из разных источников так, чтобы их было удобно использовать и, наконец, загружает данные в формате, который подходит для дальнейшей работы с ними на презентационном слое. [6]

Именно от системы загрузки зависит успешность хранилища данных. Несмотря на то, что загрузка происходит в backend и не видна пользователю, работа на ETL системой обычно занимает 70% всего времени разработки хранилища данных.

ETL включает в себя:

* извлечение данных из внешних источников в один, для дальнейшей работы с этими данными
* очистку данных и избавление от возможных ошибок
* сохранение изменений в данных
* структурирование данных для удобства конечных объектов хранилища данных
* загрузку преобразованных данных в целевую систему

Несмотря на кажущуюся простоту, каждый из этапов проекта в реальности достаточно сложен. Во-первых, в качестве внешних источников информации могут выступать различные информационные системы, форматы хранения данных которых и процедуры их извлечения могут существенно разниться. Во-вторых, ETL-процесс не сводится исключительно к техническому преобразованию форматов — данные из разнородных источников должны быть унифицированы и с точки зрения бизнес-правил, единства применяемых систем кодирования информации, классификаторов и справочников. В-третьих, процесс должен учитывать и особенности бизнес-процессов компании, в том числе, функционирования выступающих в качестве источников данных отдельных информационных систем, периодичности обновления данных в них и т. д. [6]

Традиционный процесс ETL представляет собой последовательность шагов, вызываемых в виде пакетного задания. Время цикла процесса ETL ограничивает актуальность данных в хранилище данных; трудно добиться, чтобы это время было меньше нескольких минут. Например, б*о*льшая часть шагов, требующих переработки большого объема данных, выполняется на основе запросов к хранилищу данных: поиск существующих значений в таблице измерений (например, клиентов, совершивших предыдущую покупку) и заполнение сводных таблиц. Система обработки запросов к потоковым данным может кэшировать информацию, требуемую для выполнения этих шагов, снимая излишнюю нагрузку с хранилища данных, в то время как процесс ETL является слишком кратковременным, чтобы в нем было выгодно производить кэширование.

Во время разработки ETL системы необходимо держать в голове сразу два параллельных процесса: планирование и дизайн, и работа с данными. [6]

Процесс планирования и дизайна в общем виде состоит из четырех этапов и может быть представлен следующим образом:

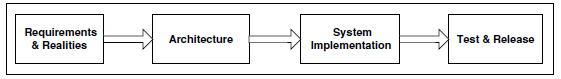


Рисунок 3. Планирование и дизайн

На первом этапе происходит сбор всех существующих требований к системе и оценка возможности их выполнения. На втором этапе разрабатывается непосредственно архитектура процесса загрузки. Третий этап состоит из реализации полученной схемы и последний этап это тестирование получившейся системы и её внедрение.

Работа с данными происходит в виде последовательности различных потоков данных (Data Flow). Она так же может быть представлена в виде четырех основных этапов и в целом отражает сам процесс ETL.

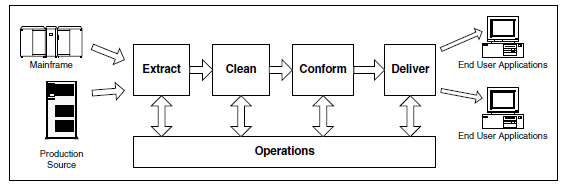


Рисунок 4. Потоки данных

На первом этапе происходит выгрузка данных из существующих источников. На втором этапе данные очищаются – проверяются на наличие ошибок или пропусков. На этом этапе все данные проходят процедуру подтверждения их правильности и соответствия бизнес-требованиям. Далее, происходит интеграция информации из несовместимых источников, так чтобы все одинаковые поля назывались одинаково, и все поля с одинаковым названием означали одно и то же. И на последнем этапе эти данные загружаются в конечное хранилище.

Правильное и последовательное выполнение этих процессов приведет к стабильной и быстрой ETL системе, что в свою очередь улучшит качество всего хранилища данных.

Проектирование хранилища данных с архитектурой шины

Ральф Кимбалл (Ralph Kimball), автор данного подхода, описывал хранилище данных как "место, где люди могут получить доступ к своим данным". Он так же сформулировал основные требования к хранилищам данных[5]:

• **Хранилище данных должно сделать информацию об организации легко доступной.** Содержимое хранилища данных должно быть понятным. Процесс извлечения информации должен быть интуитивно понятным для бизнес пользователя, а не только для разработчика. Бизнес пользователь будет использовать информацию во всех возможным комбинациях и вариациях (срезы данных). Все инструменты для доступа к данным должны быть простыми в использовании. Время отклика для запросов пользователя так же должно быть минимальным.

• **Информация, представленная в хранилище должна быть непротиворечивой.** Пользователь должен доверять информации. Прежде чем информация попадет в общий доступ, она должна быть аккуратно собрана из всех различных источников, очищена и проверена. Если два показателя означают одно и то же они должны называться одинаково и наоборот. Непротиворечивая информация это информация высокого качества.

• **Хранилище должно быть легко изменяемым.** Невозможно избежать изменений, поэтому хранилище данных должно легко перестраиваться при необходимости. Изменение в хранилище не должно влиять на все работающие приложения. Существующая информация и приложения не должны изменяться, если пользователь задает новые вопросы или добавляет новую информацию.

• **Информация в хранилище данных должна быть хорошо защищена.** В хранилище может храниться крайне деликатная информация. В нем как минимум хранится информация о том что и по каким ценам покупает компания. Это означает, что доступ к информации в хранилище должен быть ограничен.

• **Хранилище данных должно быть основой для улучшения и ускорения процесса принятия решений.** Существует только один важных итог работы хранилища данных: решения которые было принято после того, как были получены данные из хранилища. Именно эти решения и являются ценностью, которую привносит хранилище данных. Лучшее описание того, что мы разрабатываем - это система поддержки принятия решений.

• **Для того чтобы хранилище данных можно было назвать успешным, оно должно быть принято на всех уровнях компании.** Неважно создали ли мы элегантное решение, если оно не используется спустя шесть месяцев после обучения. В отличии от внедрения новой OLTP системы, когда у пользователей просто нет выбора, кроме как пользоваться новой системой, использование хранилища не является обязательным. А это означает, что системой будут пользоваться, только если она проста и удобна.

Для проектирования хранилища данных которые отвечают всем требованиям, представленным выше необходимо продумать основные четыре составляющие[5].

1. **Выбрать бизнес-процесс**, который будет моделироваться. Процесс это естественная активность, которая происходит в организации. Её результаты чаще всего записываются в существующую OLTP систему. Лучший способ выбрать процесс это прислушаться к мнению пользователей. Важно отличать бизнес-процесс от бизнес отдела или функции. Так, например правильно будет спроектировать единую систему для процесса заказа услуг, чем две системы для двух отделов продаж и маркетинга. Такой подход к проектированию позволяет построить единую схему и сделать для нее одну загрузку данных и тем самым избежать дублирования информации. Так же этот подход позволяет снизить затраты на процессы загрузки и представления данных.
2. **Правильно выделить самый нижний уровень детализации**, который лежит в основе процесса. Выделение нижнего уровня означает точную формулировку того, что будет подразумеваться под таблицей фактов. Данный шаг процесса проектирования очень легко пропустить, посчитав его слишком очевидным, но этого делать ни в коем случае не стоит. Именно от того, насколько правильно выбрано событие в основе факта зависят дальнейшие возможности по использованию хранилища данных. Может получится так, что на дальнейших шагах проектирования станет понятно, что уровень выбран неправильно. В таком случае, следует еще раз выбрать нижний уровень детализации.
3. **Выбрать все измерения**, которые соответствуют факту. Выбор измерений это ответ на вопрос «Как лучше описать данные, полученные в результате бизнес-процесса?». Измерения это подробные описания каждой строчки, которая входит в факт.
4. **Выделить количественные показатели**, которые будут привязаны к каждой строке в таблице фактов. Факты легко выделить отвечая на вопрос «Что мы измеряем?». Именно эти значения чаще всего интересны бизнес-пользователям в процессе принятия решений. Важно то, что все выбранные показатели должны подходить под элементарное событие, которые мы выбрали на втором шаге.

Итак, в данной главе были рассмотрены основные понятия области хранилищ данных, были рассмотрены основные подходы к проектированию хранилищ данных и выбранный подход был рассмотрен более подробно. Этого достаточно для того, чтобы перейти к проектированию и разработке хранилища данных.

# Глава 2. Практическая часть

Обоснование выбора направления

Сфера интернет хостинга появилась относительно недавно. В России первые компании начали появляться с 1997 года. На данный момент существует порядка 5 крупных компаний и несколько десятков небольших компаний.

Чаще всего все бизнес процессы, связанные с предоставлением услуг клиенту в данной сфере проходят онлайн и только малая часть данного процесса происходит вне сети. В результате клиент выбирает и покупает услуги без непосредственного физического контакта с компанией. Все услуги обычно не являются уникальными, и сразу несколько компаний предлагают одно и то же за примерно одинаковую цену. У клиента редко возникает чувство лояльности к компании, и он легко меняет одного хостинг-провайдера на другого. Это приводит к тому, что без должной аналитики ситуации на рынке в каждый момент времени, компания может не заметить спад в количестве клиентов или заказываемых услуг пока не станет слишком поздно.

Большое количество транзакций, проходящих через хостинг компанию каждый день (около 200 - 500 тысяч) предполагает наличие в ней OLTP системы. Чаще всего, в результате многочисленных слияний и объединений компаний данной сферы эти системы состоят из множество подсистем и могут дублировать одну и ту же информацию.

В результате данная сфера довольно сильно нуждается в централизованных хранилищах данных для поддержки принятия решений.

Интернет хостинг

Хостинг— это сервера и его вычислительных мощностей для физического размещения информации на сервере, постоянно находящемся в сети Интернет. В хостинг также входит услуга колокации - размещение оборудования клиента на территории провайдера. В этом случае провайдер обеспечивает подключение оборудования к своим каналам связи с высокой пропускной способностью.

Обычно услуги хостинга предоставляются пакетами. Это означает, что кроме непосредственно услуги размещения файлов самого сайта на сервере с установленным ПО, в этот пакет так же входят дополнительные услуги. Например, предоставление места для почтовой корреспонденции, баз данных, DNS, файлового хранилища на специально выделенном файл-сервере, а также поддержка функционирования соответствующих сервисов.

Процесс предоставления хостинга клиенту происходит следующим образом:





Рисунок 5. Заказ клиентом услуги

Клиент приходит на сайт компании, предоставляющей услуги и, если он еще не пользовался услугами этой компании, проходит процесс регистрации. При процессе регистрации клиент заполняет всю личную информацию о себе, которая потом заносится в базу. После процесса регистрации он выбирает и заказывает необходимую ему услугу.

Далее он может либо сразу оплатить заказанную им услугу, либо оставить заказ не оплаченным. В любом случае в системе создается новая запись о купленной услуге. Неоплаченный заказ хранится в системе 30 дней, после чего автоматически переходит в раздел удаленных. В течении этого срока клиент может в любой момент оплатить заказанную услугу.

Для оплаты услуги у клиента есть два способа. Первый это оплатить стоимость непосредственно самой услуги, используя электронные или наличные платежи. При данном способе оплаты клиент может заплатить только столько денег, сколько стоит заказанная услуга.

Второй способ – пополнить свой счет в личном кабинете и оплатить услугу с него. Пополнение счета происходит на любую сумму по желанию клиента. При данном способе оплаты в системе считается что клиент заказал еще одну услугу пополнения счета и оплатил её.

Сразу после оплаты клиентом услуги, она переходит в статус активных и он может ей пользоваться. После окончания срока действия услуги клиент может либо продлить ей и тогда будет создана новая услуга с типом «продление», либо не делать ничего и услуга перейдет в статус «закончена», а клиент больше не будет иметь возможности ей пользоваться.

Иметь представление о том, как происходит процесс заказа клиентом новой услуги важно для того, чтобы лучше понимать устройство OLTP системы, а значит для проектирования ETL и хранилища данных.

Принятие решений

Как уже было сказано выше, процесс принятия решений в данной сфере является более чем актуальным. Рынок достаточно насыщен конкурентами, для того, чтобы даже небольшие решения могли менять ситуацию в сторону одной из хостинговых компаний.

Основными направлениями принятия решений являются реклама и маркетинг. Прежде чем вводить новые маркетинговые акции необходимо знать состояние дел на данный момент, это же немаловажно при разработке новой рекламной компании.

Решение о добавлении новой услуги или предоставлении скидки на уже существующее должно быть принято на основе имеющейся информации. А успех любого мероприятия будет оцениваться по тому, насколько улучшились существующие показатели.

Это означает, что важна такая информация, как:

* количественные показатели предпочтений клиентов
  + Количество услуг на клиенте
  + Количество клиентов на услуге
* усредненная информация по данным
* статистика личных данных клиентов
  + распределение клиентов по странам
  + распределение клиентов по возрастам
  + распределение клиентов по полу

Так же, немаловажную роль будут играть возможности переходов из одного отчета в другой и возможность опускаться на более детализированный уровень отчетности.

OLTP система

Ниже представлена схема существующей на данный момент OLTP системы, необходимой для полного функционирования интернет хостинга.

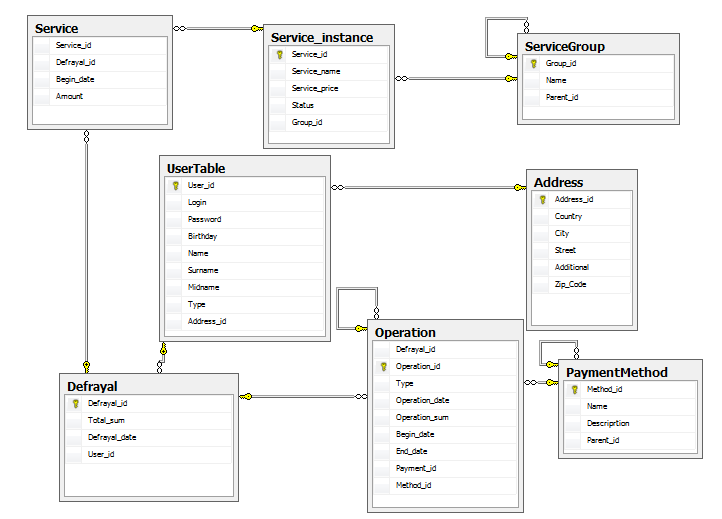


Рисунок 6. OLTP модель

Система состоит из 6 основных таблиц. Таблица UserTable сохраняет логин и пароль клиентов компании, данные об его ФИО и дате рождения, а так же данные о его типе (физическое лицо, юридическое лицо или индивидуальный предприниматель).

С этой таблицей связана таблица Address, в которую записываются данные об адресе клиента. Если несколько клиентов регистрируются на один адрес, то новой записи об адресе создаваться не будет и клиент просто будет связан с этим адресом. Эта система очень удобна для всех статистических данных о месторасположении клиента.

Таблица service\_instance является каталогом всех услуг, которые предоставляет компания, в ней хранится название услуги, стоимость, а так же информация о её статусе (активна, неактивна). С данной таблицей связана таблица Service\_Group.

Она необходима для того, чтобы можно было группировать все услуги по категориям, например, отличать услуги хостинга от услуг колокации. При этом группы имеют структуру дерева, то есть у каждой группы может быть группа родитель и подгруппы. Например, группа хостинг услуг и подгруппа услуг типа «МИР» (самые недорогие услуги с наименьшими возможностями). Древовидная иерархическая структура реализована за счет того, что таблица ссылается сама на себя.

Таблица service является связующей для таблиц service\_instance и defrayal, с её помощью реализуется связь многие-ко-многим. Defrayal хранит все себе общую сумму заказа и идентификатор пользователя, сделавшего заказ.

Таблица Operation служит для хранения платежей и списаний, которые сделал пользователь. Поскольку по российскому законодательству компании не имеют права списывать деньги за услуги, которые еще не были предоставлены, если клиент купит услугу, например, на год деньги сразу за весь год списывать нельзя. Поэтому эти деньги на счету клиента просто блокируются и становятся ему недоступны, а уже, например, каждый месяц нужное количество денег будет списываться.

Поле type служит для хранения типа операции, соответственно Income (платеж клиента), Sale (списание) и Block (блокировка). Таблица ссылается сама на себя - поле payment\_id, для того, чтобы знать к какому платежу было списание, для платежей оно соответственно будет пустым. Таблица связана с defrayal через поле defrayal\_id. Так же в ней хранится дата операции, и срок за который списались деньги (для платежей пустое).

С таблицей Operations так же связана таблица PaymentMethod. В ней содержаться данные о том, как именно расплачивался клиент, будь то кредитная карта или обычный платеж через банк. Структура методов платежей тоже древовидна. Это необходимо потому, что электронные платежи в компанию осуществляются через несколько различных платежных систем. Иерархическая структура позволяет сгруппировать все одинаковые методы платежа из различных систем.

Подводя итоги, можно сказать, что данная схема позволяет производить все основные операции, необходимые для полноценной работы хостинг-компании.

Платформа для реализации хранилища

То, какая СУБД является наиболее удобной для разработки хранилища данных, очень сильно зависит от конкретных условий и требований к системе. Поскольку через существующую OLTP систему проходит до 500 000 транзакций ежедневно, для разработки хранилищ могут подойти только СУБД рассчитанные на высоконагруженные системы. Наиболее используемыми подходящими СУБД на данный момент являются:

* Microsoft SQL Server
* Oracle Database
* IBM Informix

Для текущей OLTP системы в компании на данный момент уже используется Microsoft SQL Server, что является аргументом в пользу использования той же СУБД для хранилища данных, так как позволит более сделать процесс загрузки данных более удобным.

Это не значит, что СУБД хранилища всегда должна совпадать с СУБД существующей системы, тем более, что хранилище данных может собирать информацию из различных источников и различных платформ. Но если такая возможность есть, все же удобнее когда платформы совпадают.

Другим аргументов в пользу Microsoft SQL Server является его ориентированность на построение аналитических приложений и встроенные возможности для ETL процессов и построения OLAP кубов, а так же построения отчетов.

Проектирование хранилища данных

Процесс проектирования любого хранилища, как уже было сказано, делится на следующие составляющие:

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

В части «Интернет хостинг» был описан основной для этой сферы процесс – покупка клиентом услуги. Именно этот процесс лежит в основе хранилища данных.

Элементарным событием этого процесса является непосредственно факт оплаты клиентом услуги (Fact\_Billing). Можно было так же взять факт заказа клиентом услуги в качестве элементарного события, но с точки зрения процесса принятия решений факт покупки более интересен.

Измерениями являются таблицы с записями о клиенте, о дате и об услуге. При этом адрес клиента будет записан непосредственно в информацию о клиенте, так как создание дополнительного измерения в данном случае избыточно.

Показателями будут количество услуг, купленных клиентом, стоимость купленных услуг.

В результате проектирования получилась следующая схема хранилища данных:

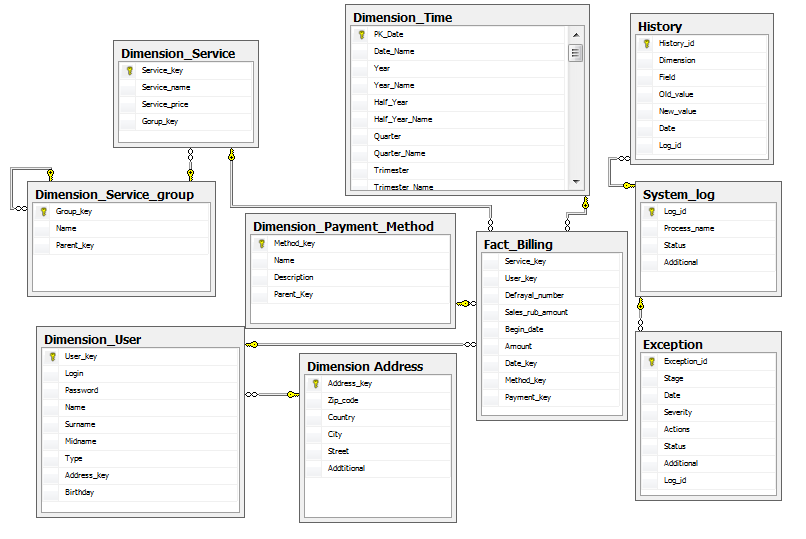


Рисунок 7. Схема хранилища данных

Данная схема позволяет строить все необходимые отчеты для стратегического принятия решений в сфере интернет хостинга.

Схема построена по принципу «Звезда». В центре схемы находится факт оплаты клиентом услуги. У факта есть собственные поля, которые его характеризуют – количество купленных услуг, общая стоимость в рублях, номер списания и номер оплаты, которые по сути являются дегенеративными измерениями (измерениями без дополнительных полей).

От таблицы факта идут три «луча» - таблицы измерений. Данные измерения похожи на измерения исходной системы и выполняют абсолютно те же функции что и таблицы исходной системы. В измерении пользователя и адреса хранится соответствующая информация о клиентах компании. В измерении типа платежа хранится информация о том, как именно был совершен платеж, например через кредитную карту или через банк. А в таблицах услуг и групп услуг хранится более подробная информация о том, что именно оплатил клиент. Единственное новое измерение – измерение времени.

Измерение времени – измерение которые присутствует практически в любой схеме хранилища данных. Особая структура этого измерения позволяет особым образом строить отчеты, фильтруя их например по месяцу, когда произошла покупка.

Таблицы System\_log, History и Exception являются системными таблицами. Первая необходима для легирования всех ETL операции, стадий их выполнения и результатов этого процесса. Остальные две таблицы так же служат для поддержки ETL процесса и будут описаны в соответствующем разделе работы.

Скрипт для создания структуры хранилища данных находится в приложении к диплому.

Проектирование ETL системы

**Автоматизация загрузки**

Уровень автоматизации может быть совершенно различным. Начиная с ручного запуска всех процессов и заканчивая полностью автоматической системой, которая сама распознает что, в каком порядке и когда запускать, ожидает пока остальные части не завершат свою работу и может общаться с супервизорами, сообщая им о критических ошибках и статусах работ.

В данном случае полная автоматизация не является необходимой и будет использован средний уровень автоматизации. То есть все ежедневные запуски необходимых этапов будут проводиться автоматически, но обработка всех возникших ошибок должна проводиться вручную.

**Определение новых записей фактов**

Для таблицы фактов и для всех таблиц OLTP системы, из которых берутся данные для таблицы фактов, существует поле с датой создания записи.

Загрузка новых фактов в хранилище происходит инкрементально, то есть добавляются только новые записи, а старые остаются неизменными. Процесс проверки того, является ли факт новым, происходит по полю даты создания, то есть ежедневно загружаются только записи, созданные в предыдущий день.

Подобный способ загрузки позволяет исключить дублирование записей и ускорить процесс загрузки.

**Определение новых элементов справочников и контроль изменений**

Для справочников поля с датой, по которому можно было бы сказать когда была создана новая запись нет, и к тому же данные в справочниках пользователей и услуг, в отличие от данных в факте, могут меняться – клиент может поменять адрес проживания или фамилию, а услуга может стать неактивной.

Это означает, что данные в этих таблицах должны ежедневно сравниваться с данными в OLTP системах, новые записи должны быть добавлены, а измененные записи отслежены.

Контроль изменений в данных справочников происходит с помощью таблицы, в которой хранятся все изменения всех полей справочников. Таблица выглядит следующим образом:

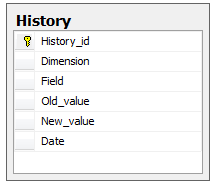


Рисунок 8. Таблица изменений

В данной таблице хранятся поля с названием измерения, в котором были изменения, поля, старого значения, нового значений и даты, когда было внесено изменение. При необходимости все эти данные могут быть просмотрены бизнес-пользователем.

В тот момент, когда процесс обнаруживает, что какое-либо поле в данных справочников было изменено, он записывает все изменения в данную таблицу и меняет сами значения в справочниках.

**Обработка проверок**

На этапе загрузки данных для каждого измерения и для факта проверяется, что все поля не пустые. В случае, если поля пустые вместо пустых значений в хранилище записываются знаки вопроса и выдаются соответствующие ошибки.

В будущем в данную систему было бы целесообразно добавить проверку реальности адреса пользователя и совпадения его страны и города с индексом. Но в задачи данной работы реализация этой проверки не входит.

**Общая модель загрузки**

В целом модель загрузки выглядит следующим образом:



Рисунок 9. Модель загрузки данных

Сначала идет загрузка данных в справочники. Как уже было сказано выше загрузка происходит инкрементально и все данные в исходных таблицах сравниваются с данными в таблицах хранилища данных на предмет изменений или новых записей.

После этого происходит загрузка данных в таблицу фактов. Для записи в таблицу фактов используются данные сразу из трех таблиц OLTP системы: Service, Defrayal и Payment. При этом берутся только уже оплаченные услуги, то есть те, для которых есть записи в таблице платежей. Текст запросов для загрузки данных из таблиц исходной системы находится в приложении.

Помимо таблиц OLTP системы так же используется информация из справочников хранилища данных. Данные из этих таблиц подгружаются с помощью инструмента lookup. Данные в таблицу фактов так же подгружаются инкрементально.

При загрузке данных происходят проверки данных на корректность и обрабатываются ошибки.

Загрузка данных из исходной системы в хранилище данных, а так же все проверки и обработки ошибок реализованы средствами SQL Server Business Intelligence Studio.

**Обработка ошибок**

Все ошибки, обнаруженные в процессе загрузки и очистки данных, записываются в специальную таблицу в хранилище данных. Таблица выглядит следующим образом:

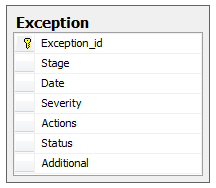


Рисунок 10. Обработка ошибок

Все этапы процесса спроектированы таким образом, что именно в эту таблицу собираются все возникающие ошибки. Столбцами данной таблицы являются идентификатор ошибки, этап, на котором возникла ошибка, время, когда возникла ошибка, потенциальный уровень сложности ошибки, предпринятые действиями по устранению ошибки, статус ошибки и дополнительная информация, например содержание строки с информацией, которая не была загружена.

Такой механизм подразумевает ежедневный мониторинг данной таблицы на предмет появления новых ошибок и последующую их обработку.

Приложение для построения отчетов

Для построения отчетов было выбрано приложение Investigation Center среды Intellinx. Изначально Intellinx это продукт, ориентированный на предотвращение мошенничества и злоупотреблений. Но с дальнейшим развитием данного продукта в нем появилось приложение, которое ориентировано на построение отчетов на основе хранилища данных использующих архитектуру шины.

Приложение Investigation Center это java веб-приложение, которое предоставляет пользователю доступ к средствам разработки и построения отчетов. В результате работы формируются отчеты, которые могут быть легко изменены и дополнены. У конечного бизнес-пользователя так же есть доступ к данному приложению, в котором он может просматривать все существующие отчеты и даже создавать новые.

Данная среда наиболее подходит под существующие задачи данной работы, благодаря своей гибкости и удобства интерфейсов. Её основным преимуществом, помимо удобства в использовании является легкая установка. Приложение достаточно один раз установить на сервере, к которому есть доступ у всех пользователей, а для конечных пользователей достаточно наличия интернет браузера.

# Глава 3. Результаты

Реализованные отчеты

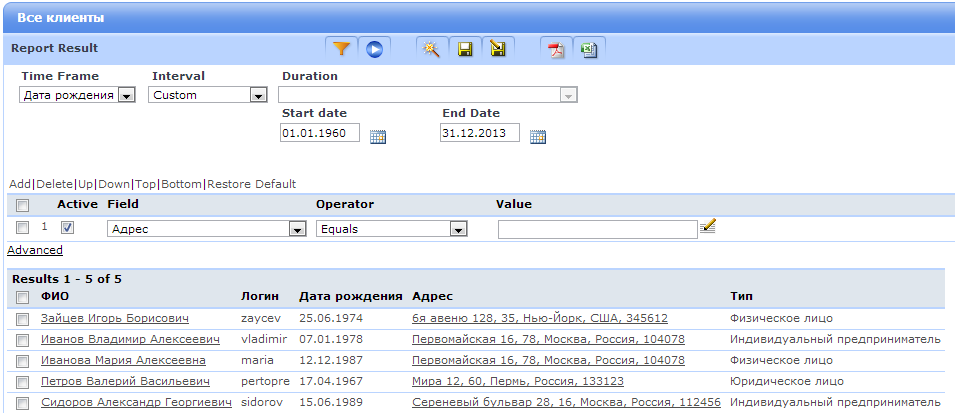
С помощью Intellinx Investigation Center были реализованы следующие отчеты, подотчеты и диаграммы:

Рисунок 11. Отчет по всем клиентам

Данный отчет позволяет просмотреть всех клиентов компании, а так же отфильтровать и отсортировать их по любым выбранным полям. Данный отчет далее будет использоваться как подотчет для других сущностей. При нажатии на ФИО конкретного клиента происходит спуск на более низкий уровень детализации, где например можно посмотреть на все услуги, купленные клиентом:

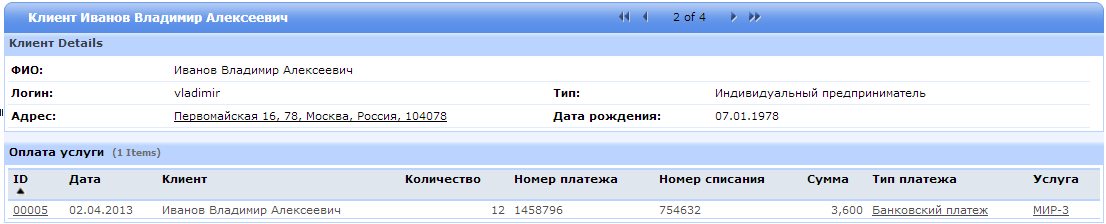


Рисунок 12. Карточка клиента

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

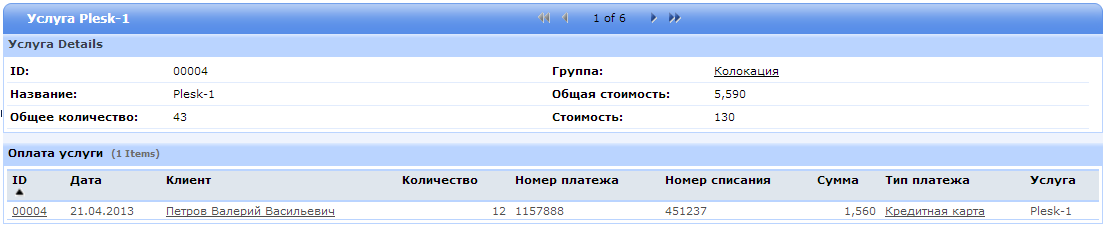


Рисунок 13. Карточка услуги

Так же можно посмотреть на отчет по всем услугам, когда-либо купленным в компании:

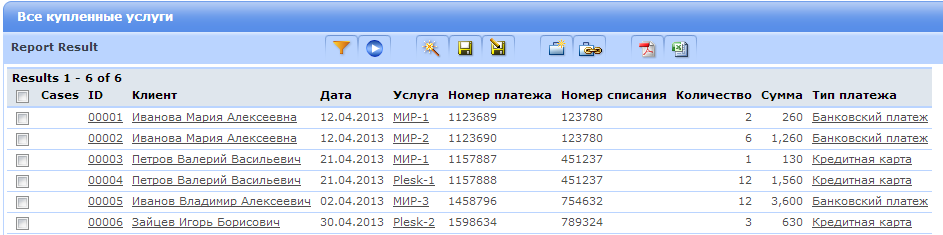


Рисунок 14. Все купленные услуги

И непосредственно отчет по всем существующим в компании с количеством раз, которое была куплена услуга и общей суммой которая была потрачена на эту услугу. Сумма и количество услуг считаются за определенный период и могут быть выбраны пользователем.

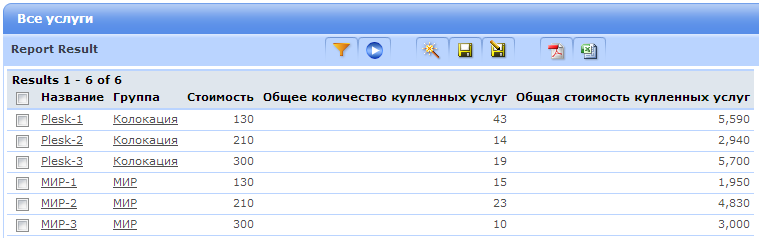


Рисунок 15. Отчет по всем услугам в компании

Так же были построены диаграммы, которые отображают, например суммарную популярность услуг компании в течение одного месяца:

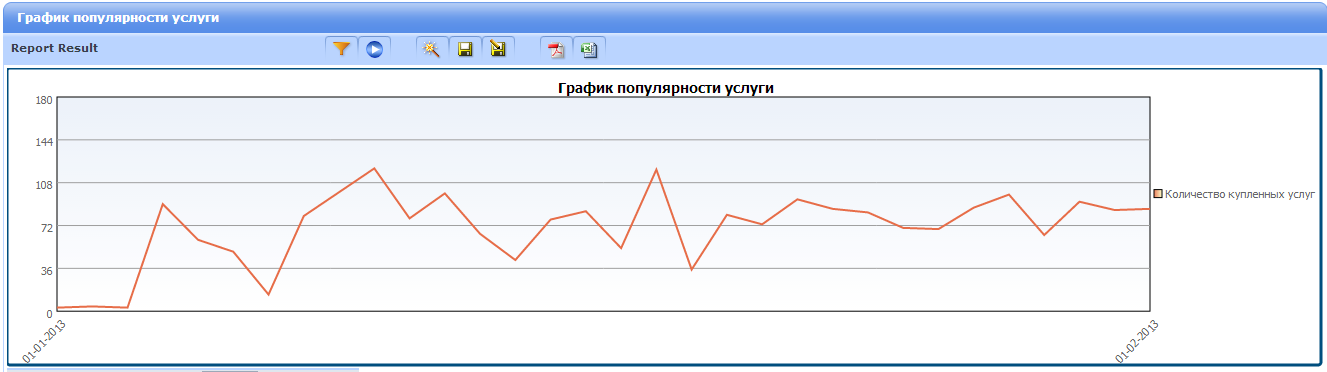


Рисунок 16. Популярность услуг

Или распределение клиентов по возрасту:

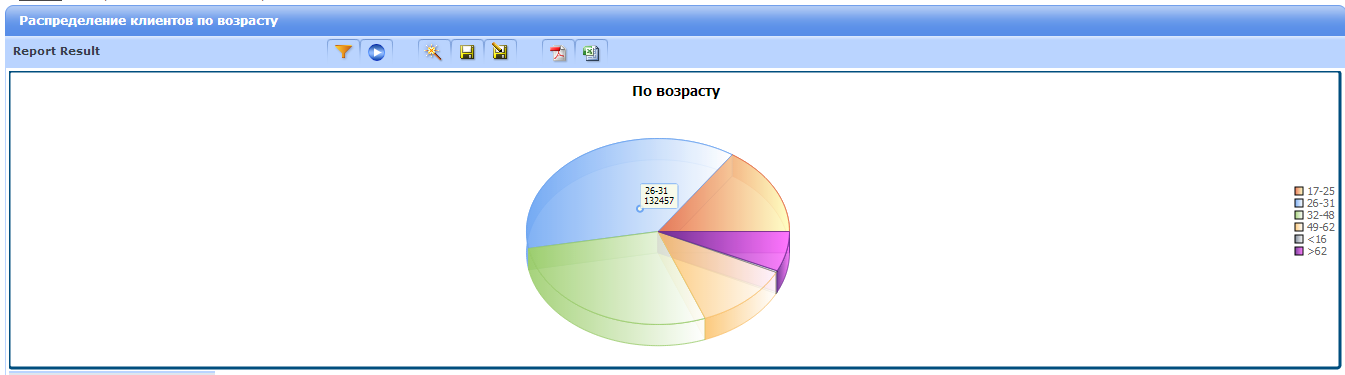


Рисунок 17. Распределение по возрасту

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

Одной интересной вещью, которая может быть реализована с помощью Intellinx Investigation Center, является граф зависимостей между различными сущностями. С его помощью, например, можно посмотреть на все другие услуги, которые покупают клиенты конкретной выбранной услуги.

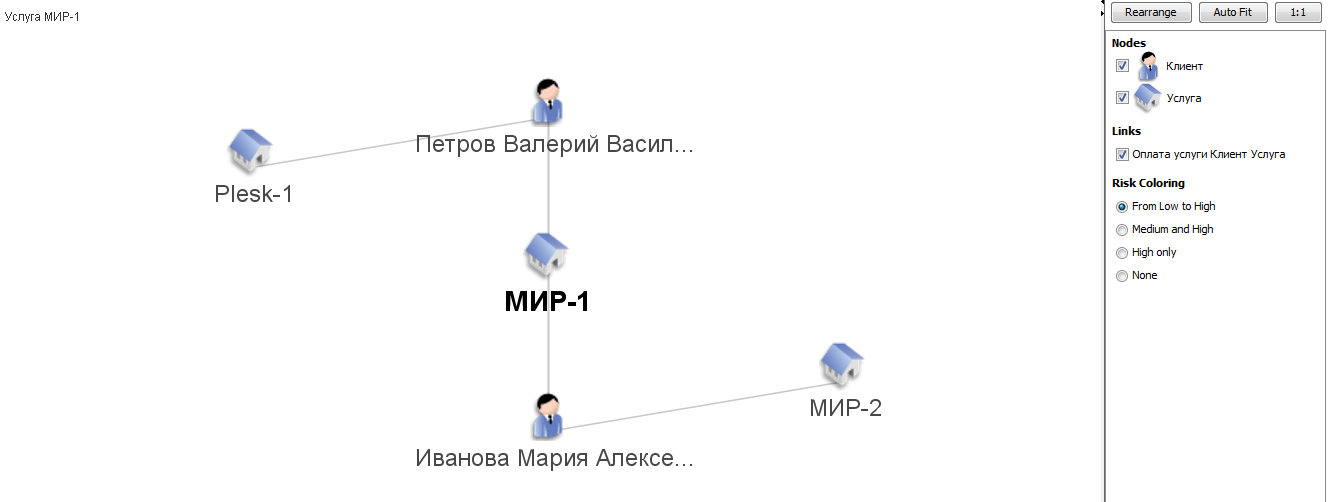


Рисунок 18. Граф клиентов и услуг

Преимущества и недостатки

Основными преимуществами построенного хранилища данных являются:

* Легкость его использования
* Улучшение качества информации за счет очистки данных
* Создание единой модели данных
* Реструктуризация данных для улучшения процесса принятия решений без внесения изменений в существующие системы
* Удобный доступ к данным, необходимым для процесса принятия решений в компании
* Легкость внесения изменений в хранилище при такой необходимости
* Удобная система создания отчетов

Основным недостатком спроектированного и реализованного хранилища данных является его ориентированность хранилища на конкретный бизнес-процесс без рассмотрения остальных процессов компании. Это связано с тем, что рассмотрение остальных бизнес-процессов не входило в рамки данной работы.

# Заключение

Все поставленные задачи были выполнены полностью. В результате была изучена сфера интернет хостинга, была изучена необходимость в системе принятия решений и была спроектирована и разработана модель хранилища данных для конкретного бизнес процесса сферы интернет хостинга.

Так же была изучена литература по построению хранилища данных и были рассмотрены два основных подхода к проектированию его архитектуры и основные плюсы и минусы каждого из подходов. Выбранный подход был рассмотрен более подробно. Помимо этого, был изучен подход к проектированию ETL процессов хранилища данных.

В дальнейшем разработанное хранилище данных может быть расширено за счет необходимости в построение дополнительных отчетов. А так же в него могут быть внедрены более сложные проверки при загрузке данных из OLTP системы, как, например, проверка существования адреса, по которому зарегистрирован клиент.

# Список использованной литературы

1. Golfarelli, M., Rizzi, S. Data warehouse design: Modern principles and methodologies. New York: McGraw-Hill, Inc, 2009.
2. Hillard, R., Information-Driven Business. Indianapolis: Wiley Publishing, Inc, 2010.
3. Imhoff, C., Galemmo N., Geiger, J.G. Mastering Data Warehouse Design Relational And Dimensional Techniques. Indianapolis: Wiley Publishing, Inc, 2003.
4. Inmon W.H., Building the Data Warehouse, 3rd edition, John Wiley & Sons, Inc., New York, 2002
5. Kimball R., Ross M., The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling, 2nd edition, John Wiley & Sons, Inc., New York, 2002
6. Kimball R., Caserta J., The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleanin, Wiley Publishing, Inc., Indianapolis, 2004
7. Mattison R., Data Warehousing and Data Mining for Telecommunications, Artech House, London, 1997
8. Kimball, R., and Ross, M.: Differences of Opinion// KimballGroup Режим доступа от 06 марта 2004 года: http://www.informationweek.com/differences-of-opinion/17800088
9. Ross, M. The bottom-up misnomer. Business Intelligence and Data // Warehouse Articles, режим доступа от сентября 2003 года: http://www.kimballgroup.com/2003/09/17/the-bottom-up-misnomer/
10. Основные подходы к архитектуре Хранилищ данных // Intersoft Lab Режим доступа от мая 2005 года: http://www.iso.ru/print/rus/document6082.phtml
11. ГОСТ Р ИСО МЭК ТО 10032-2007: Эталонная модель управления данными (идентичен ISO/IEC TR 10032:2003 Information technology — Reference model of data management)
12. Грекул В.И., Денищенко Г.Н., Коровкина Н.Л. «Проектирование ин-формационных систем.» — М.: Интернет-университет информационных технологий - ИНТУИТ.ру, 2005.
13. Громов А.И., Моделирование бизнеса // Практическое руководство Методология ARIS – М.: Весть-Метатехнология, 2001
14. Кириллов В.В., Основы проектирования реляционных баз данных. СУБД – СП: БХВ-Петербург, 2008
15. Репин В., Бизнес-процессы. Моделирование, внедрение, управление – М.: Манн, Иванов и Фербер, 2013