МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение
высшего образования
«Южно-Уральский государственный университет»
(национальный исследовательский университет)
Высшая школа электроники и компьютерных наук
Кафедра системного программирования

РАБОТА ПРОВЕРЕНА

Рецензент
Директор ООО «ВОРТЕКСКОД»

_____ А.В. Воробьев

“___”_____ 2017 г.

ДОПУСТИТЬ К ЗАЩИТЕ

Заведующий кафедрой, д.ф.-м.н., профессор

_____ Л.Б. Соколинский

“___”_____ 2017 г.

РАЗРАБОТКА СИСТЕМЫ АНАЛИТИЧЕСКОЙ ОБРАБОТКИ ДАННЫХ ДЛЯ СИСТЕМЫ БРОНИРОВАНИЯ БАНКЕТНЫХ КОМПЛЕКСОВ

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
ЮУрГУ – 02.03.02.2017.12-012-1881.ВКР

Научный руководитель
к.ф.-м.н., доцент кафедры СП
_____ А.Т. Латипова

Автор работы,
студент группы КЭ-402
_____ Т.В. Матвиенко

Ученый секретарь
(нормоконтролер)

_____ О.Н. Иванова

“___”_____ 2017 г.

Челябинск-2017
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ОГЛАВЛЕНИЕ

ГЛОССАРИЙ 4
ВВЕДЕНИЕ 5
1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 8
1.1. Теория бизнес-анализа 8
1.2. Консолидация данных 9
1.3. Методики прогнозирования 12
2. ТРЕБОВАНИЯ К СИСТЕМЕ 16
2.1. Функциональные требования 16
2.2. Варианты использования системы 16
3. ПРОЕКТИРОВАНИЕ СИСТЕМЫ 18
3.1. Компоненты системы 18
3.2. Схема хранилища данных 21
4. РЕАЛИЗАЦИЯ СИСТЕМЫ 24
4.1. Выбор средств реализации системы 24
4.2. Реализация компонента конструктора наборов данных 24
4.3. Реализация компонента каталога наборов данных 24
4.4. Реализация компонента конструктора предварительной обработки набора данных 25
4.5. Реализация компонента визуализации отчета 29
4.6. Реализация компонента построения прогноза 29
4.7. Реализация каталога прогнозов 30
4.8. Создание отчетов, прогнозов и визуализаций 31
5. ТЕСТИРОВАНИЕ 32
ЗАКЛЮЧЕНИЕ 33
ЛИТЕРАТУРА 34

ГЛОССАРИЙ

1. Бизнес-аналитика (Business Intelligence) – мультидисциплинарная область, находящаяся на стыке информационных технологий, баз данных, алгоритмов интеллектуальной обработки информации, математической статистики и методов визуализации, которая включает в себя инструменты и приложения для поиска, анализа, моделирования и доставки информации, необходимой для принятия решений [18].

2. Модель – объект или описание объекта, системы для замещения (при определенных условиях) одной системы (то есть оригинала) другой системой для лучшего изучения оригинала или воспроизведения каких-либо его свойств [14].

3. KDD (Knowledge Discovery in Databases) – процесс получения из данных знаний в виде зависимостей, правил, моделей, обычно состоящий из таких этапов, как выборка данных, их очистка и трансформация, моделирование и интерпретация полученных результатов [9].

4. Data Mining – обнаружение в сырых данных ранее неизвестных, нетривиальных, практически полезных и доступных интерпретации знаний, необходимых для принятия решений в различных сферах человеческой деятельности [9].

5. Dataiku Data Science Studio – это специализированное программное решение (или набор решений), которое содержит в себе все инструменты для осуществления процесса извлечения закономерностей из «сырых» данных: средства консолидации информации в едином источнике (хранилище данных), извлечение, преобразование, трансформацию данных, алгоритмы Data Mining, средства визуализации распространения результатов среди пользователей, а также возможности «конвейерной» обработки новых данных [4].

ВВЕДЕНИЕ

Актуальность темы

Современное общество подошло к порогу, когда одновременно сконцентрировались огромные массивы информации (в корпорациях, в государственных департаментах, в службах социального обеспечения и т.д.) в виде баз данных. Значительно возросли мощности вычислительной техники, систем и средств связи и передачи данных [5]. Интернет стал всеобщим единым хранилищем данных и знаний для человечества. Четко сформировалась потребность в новых возможностях взаимодействия с данными и со средствами их обработки [3].

Мировой рынок аналитических приложений и систем достигал $10,2 млрд в 2011 году. Однако с тех пор, как компании получили возможность собирать и хранить значительные объемы информации, ситуация коренным образом изменилась: в стране начался экспоненциальный рост спроса на системы бизнес-аналитики [6]. Сегодня среди других сегментов рынка программного обеспечения бизнес-аналитика занимает второе место по темпам роста. Исследование Gartner показывает, что бизнес-аналитика остается технологическим приоритетом для большинства руководителей, вопреки колебаниям финансового рынка, а также покупательскую активность в отношении аналитических систем проявляют бизнес-пользователи. В 2016 году объем глобального рынка аналитических платформ и приложений достиг $16,9 млрд, увеличившись на 5,2% в сравнении с 2015-м, свидетельствуют данные аналитической компании Gartner [8]. Этому немало способствовали снижение сложности и стоимости внедрения таких систем и примеры успешных решений в области корпоративных аналитических систем. Сегодня очевидно, что без анализа данных в условиях конкурентной борьбы не обойтись.

Часто бизнес-аналитика ограничивается представлением в понятной форме отчетов, что действительно помогает руководителю оперативно принять решение. Но с повышением сложности задач одним только визуальным анализом не обойтись. На помощь приходит прогнозное моделирование и извлечение

знаний. От умения производить, искать, анализировать, классифицировать, обобщать, распознавать, перерабатывать, представлять информацию и принимать решения напрямую зависят качество жизни человека и общества, их информационная и общественная безопасность.

Цель и задачи работы

Целью работы является разработка системы, позволяющей производить автоматический сбор данных из внешних источников, их анализ и обработку для системы бронирования банкетных комплексов.

Для достижения поставленной цели необходимо решить следующие задачи:

– провести анализ предметной области;

– спроектировать систему;

– реализовать компоненты системы;

– провести тестирование системы.

Структура и объем работы

Работа состоит из глоссария, введения, 5 глав, заключения и библиографии. Объем работы составляет 35 страниц, объем библиографии – 18 источников.

Содержание работы

Первая глава, «Анализ предметной области», описывает предметную область, в рамках которой выполняется данная работа.

Вторая глава, «Требования к системе», описывает функциональные и нефункциональные требования к системе.

Третья глава, «Проектирование системы», описывает этап проектирования разрабатываемой системы.

Четвертая глава, «Реализация системы», описывает реализацию системы с технической точки зрения.

Пятая глава, «Тестирование системы», описывает суть и результаты проведенных тестовых испытаний системы.

В заключении описываются основные результаты, полученные при выполнении дипломной работы.

1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Бизнес-данные, которые предприятия и организации аккумулируют в базах данных и прочих источниках, накапливаются специально для решения задач анализа только в исключительных случаях. Предприятия и организации предоставляют собранные данные для более эффективного достижения коммерческих целей, проведения финансового анализа, составления отчетности и принятия решений. Особенности бизнес-данных влияют на сам процесс анализа и на подготовку и систематизацию данных [10].

1.1. Теория бизнес-анализа

Анализ данных – прежде всего, средство проверки гипотез и решения задач исследователя. Универсальным способом изучения, с целью обнаружения зависимостей, прогнозирования, и группировки данных, является построение моделей [14].

В традиционном понимании модель представляет собой результат отображения изученной структуры на малоизученную. При таком ***аналитическом подходе*** к моделированию при изучении системы исследователь отталкивается от модели, что не пригодно при решении сложных задач анализа данных. Если модель использует множественные допущения, то представляет собой теоретическую зависимость в функциональном виде, например, величина ожидаемого спроса на будущий месяц рассчитывается как среднее от продаж за предыдущие 3 месяца, что не является корректным решением.

При использовании вышеописанного традиционного аналитического подхода в бизнесе неизбежно возникает диссонанс методов анализа и реальностью, которую они призваны отражать. В последние годы получил распространение ***информационный подход*** к моделированию, ориентированный на использование данных. Факторы, определяющие явления в бизнес-процессах, столь многообразны и многочисленны, что почти никогда не удается создать традиционную модель, удовлетворяющую таким же условиям, что делает бизнес-процессы трудно формализуемыми. В информационном подходе известна лишь структура модели (линейная регрессия, дерево решений), а параметры

модели подстраиваются под данные, описывающие поведение реального объекта, то есть объект рассматривается как «черный ящик». Процесс настройки модели является итеративным. Корректировка параметров модели происходит через получение обратной связи – отклонения результата моделирования от реальности.

Таким образом, при информационном подходе модель постепенно «подстраивается» под действительность, поскольку исходной точкой выбраны данные, характеризующие исследуемый объект.

Противоположно аналитическому подходу, модели, полученные информационным подходом, учитывают специфику моделируемого объекта, что важно для бизнес-процессов. По этой причине именно информационный подход лежит в основе современных промышленных технологий и методов анализа данных: Knowledge Discovery in Databases, Data Mining, машинного обучения.

1.2. Консолидация данных

Информационный подход к получению «модели от данных» предполагает детальный подход к качеству исходных данных, поскольку неверные, аномальные, пустые и зашумленные данные могут привести к получению неверных выводов и моделей, не имеющих какого-либо отражения действительности. Приступая к анализу бизнес-данных, необходимо было выполнить ряд шагов для доведения данных до приемлемого уровня качества и информативности. Для достижения высокой скорости выполнения аналитических запросов используется интегрированное хранилище в виде особой структуры (набора данных), что также обеспечивает целостность и непротиворечивость данных.

Основой консолидации является процесс ETL (extraction, transformation, loading) [15]. Процесс ETL решает следующие задачи:

– извлечение данных из разнотипных внешних источников;

– их преобразование к виду, пригодному для хранения в определенной структуре;

– загрузка в соответствующее хранилище данных.

Схема принципа работы процесса консолидации данных изображена на рис. 1.

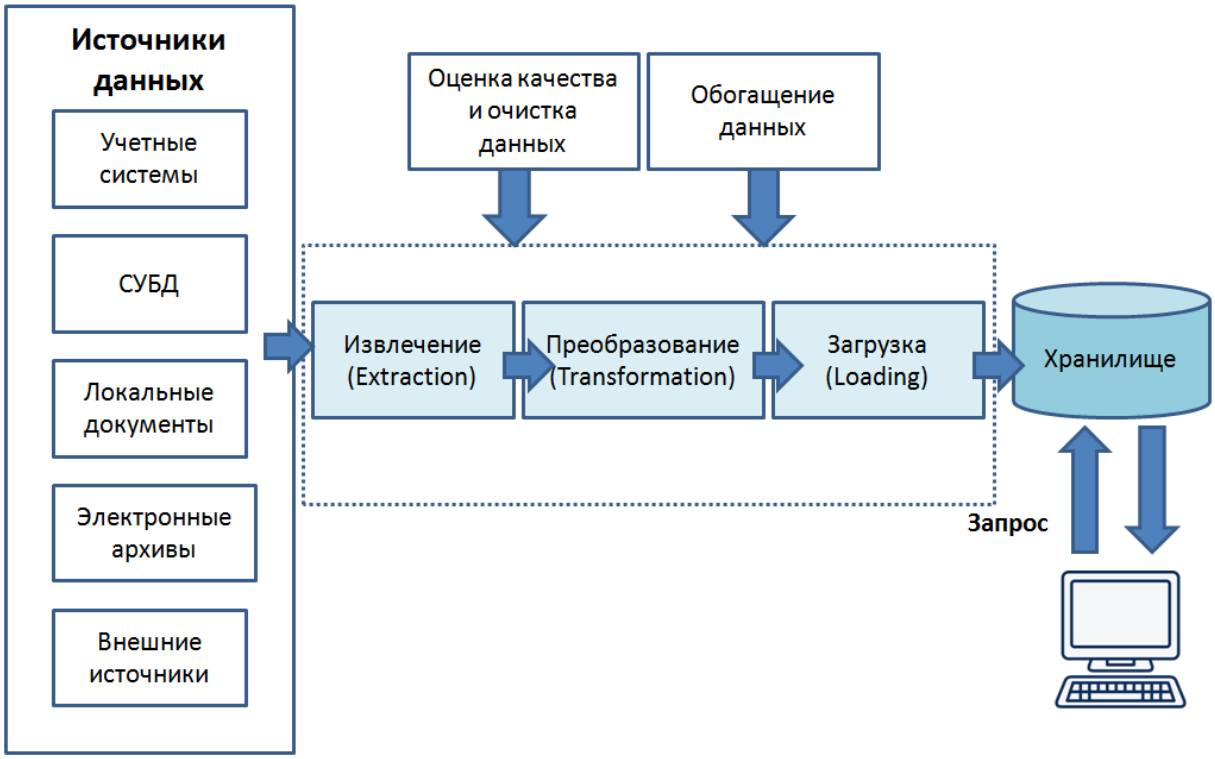

Рис. 1. Процесс консолидации данных

Системы оперативной обработки информации OLTP (On-Line Transaction Processing) были наиболее распространенным классом систем хранения данных в течение долгого времени. Они не пригодны для решения задач интеллектуального анализа данных, поскольку ориентированы на быстрое обслуживание простых запросов большого числа пользователей. Новым классом, появившимся с пониманием преимуществ аналитической обработки, являются информационные систем поддержки принятия решений (СППР). В своей работе такие системы ориентированы исключительно на получение знаний, необходимых в области управления.

Принцип работы информационной системы поддержки принятия решений изображен на рис. 2.

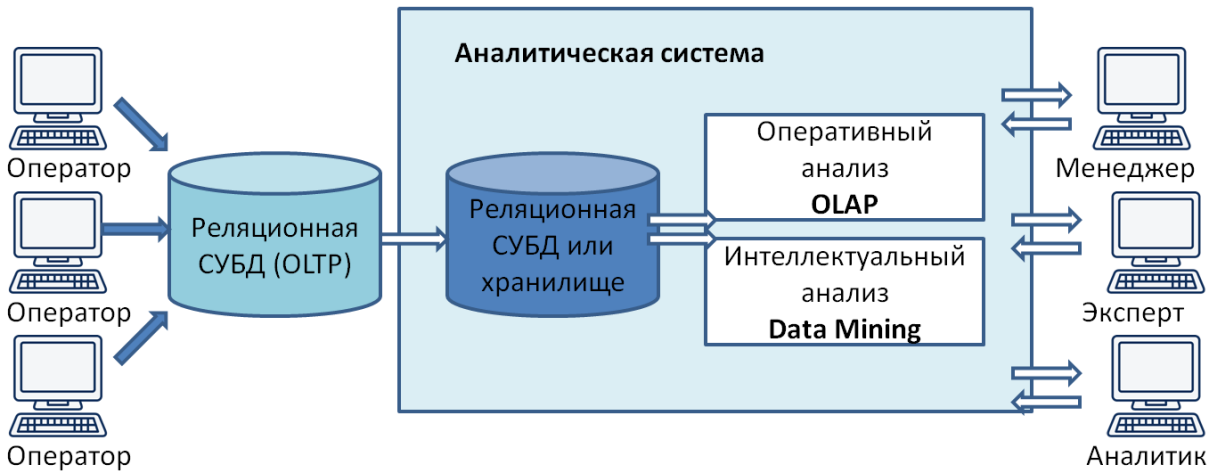

Рис. 2. Структура информационной СППР

Структура и функционирование специализированного хранилища данных специально оптимизируются для работы с аналитической платформой, что делает их использование наиболее эффективным решением. Основным преимуществом хранилища данных является семантический слой: пользователь имеет возможность формировать аналитические запросы с использованием терминов предметной области. Также в OLTP очень высок процент вносимых изменений в условиях повседневной работы, динамика смены данных также является важным отличием. В специализированных хранилищах данных, соответствующий регламент определяет процесс обновления данных, и ранее загруженные данные не удаляются. Данные хранятся в разных степенях обобщенности для решения разных задач анализа: в исходном, в детализированном, в агрегированном виде. Это существенно повышает скорость выполнения запроса, но увеличивая в разы объем хранимых данных.

Используя совместно с хранилищем данных аналитического ПО, можно решать широкий спектр задач, основными подходами к применению хранилищ данных являются:

– регулярная отчетность – подготовка стандартизованных отчетов, запрашиваемых с определенной частотой;

–возможность получить быстрый ответ на нерегламентированные вопросы;

– интеллектуальный анализ: нахождение закономерностей, построения моделей и прогнозов.

1.3. Методики прогнозирования

С точки зрения технологии анализа данных, прогнозирование может рассматриваться как определение некоторой неизвестной величины по набору связанных с ней значений и выполняется с помощью таких задач, как регрессия, классификация и кластеризация. Заявки – это процессы, распределенные во времени. Таким образом, прогнозирование в исследуемой области связано именно с анализом временных рядов – предсказанием будущих значений временного ряда [1].

Временной ряд имеет две составляющие: закономерную (детерминированную) и случайную (стохастическую). Закономерная составляющая вычисляется в соответствии с определенной функцией. Стохастическая составляющая отображает влияние на процесс случайных факторов.

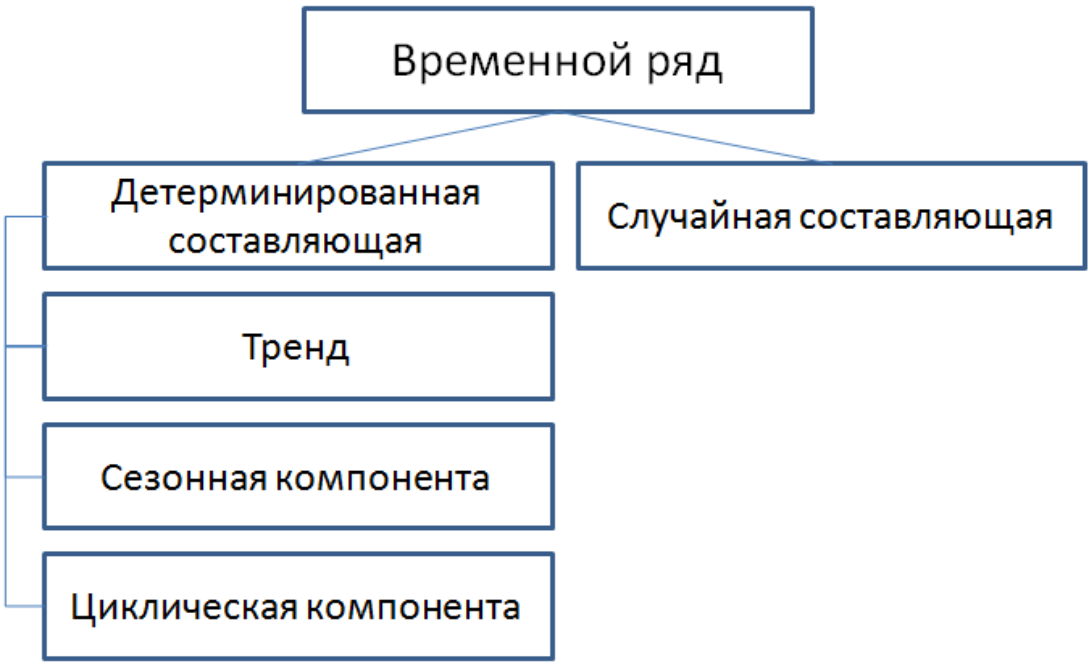

Рис. 3. Композиционная структура временного ряда

Значения временного ряда являются результатом взаимодействия детерминированной и случайной составляющих. Взаимодействие может быть как

аддитивным (результатом является сумма составляющих), так и мультипликативным (результатом является произведение составляющих).

Поведение временных рядов описывают три типовые структуры: тренд, сезонная компонента и циклическая компонента.

Детерминированная составляющая ряда с учетом указанных компонент может быть записана в виде:

$$d_i = t_i + s_i + c_i,$$

где: t_i – тренд,

s_i – сезонная компонента,

c_i - циклическая компонента,

$i = 1 \ldots n$ – номер временного отсчета.

Тренд наиболее часто описывают при помощи следующих моделей:

1). Простая линейная модель:

$$t_i = a + b \cdot i,$$

где $i = 1 \ldots n$ – номер отсчета времени (элемента ряда).

Линейная модель полезна при решении реальных задач анализа, поскольку многие бизнес-процессы линейны по своей природе.

2). Полиномиальная модель:

$$t_i = a + b_1 \cdot i + b_2 \cdot i^2 + \ldots + b_n \cdot i^n$$

В большинстве реальных задач степень полинома не превышает 5.

3). Экспоненциальная модель:

$$t_i = \exp(a + b \cdot i)$$

Используется при равномерном увеличении темпов роста.

4). Логистическая модель:

$$t_i = \frac{a}{1 + b \cdot e^{-ki}},$$

где: k – константа, управляющая крутизной логистической функции.

Логистическая модель описывает процессы с непостоянными темпами роста.

Сезонная компонента детерминированной составляющей временного ря-

да отражает свойственную бизнес-процессам повторяемость во времени. Неверно утверждать, что сезонная компонента описывается периодической функцией. Сезонная компонента ясно прослеживается на графике временного ряда, однако регулярные всплески продаж не будут отображены с абсолютной точностью.

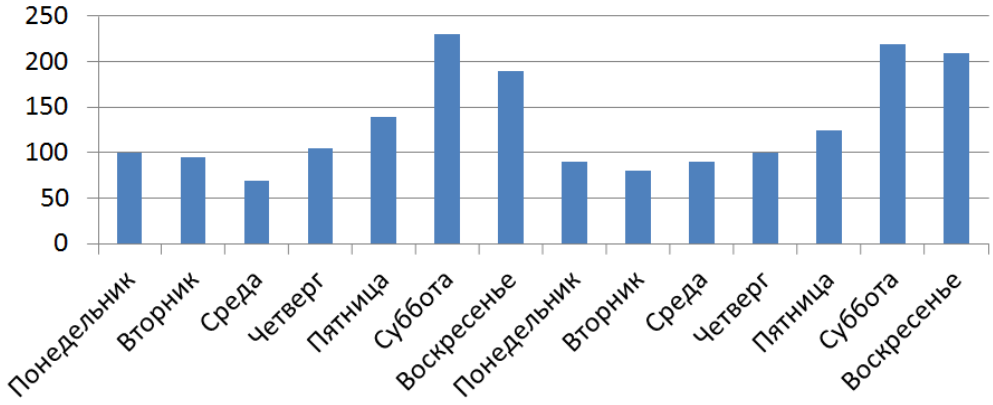

Рис. 4. Сезонная компонента временного ряда

Циклическая компонента временного ряда располагается между детерминированной и случайной составляющими временного ряда. Циклическая компонента отображает изменения, которые не являются достаточно протяженными, чтобы относиться к тренду, или регулярными, чтобы их можно было отнести к сезонной компоненте.

Выделение во временных рядах циклической компоненты является сложной задачей, поэтому, если имеются данные о взаимосвязанных с рассматриваемым временных рядах, используют их.

В данной работе прогнозом является классификация заявок по набору связанных с ними значений. Для поиска решения используется классификационная методика ***деревья решений***, в которой правила принятия решения извлекаются непосредственно из исходных данных.

Дерево решений (дерево классификаций) – модель иерархии, в узлах которой на основании правила проверяется определенный атрибут. В результате формируются дочерние узлы, удовлетворяющие или не удовлетворяющие правилу в родительском узле. Каждый лист содержит объекты одного класса.

Пример дерева решений приведен на рис. 5.

Классический алгоритм построения следующий: начиная с корня, объекты делятся на подмножества в соответствии с правилом, для каждого из которых формируется дочерний узел. Алгоритм повторяется до выполнения одного из условий остановки.

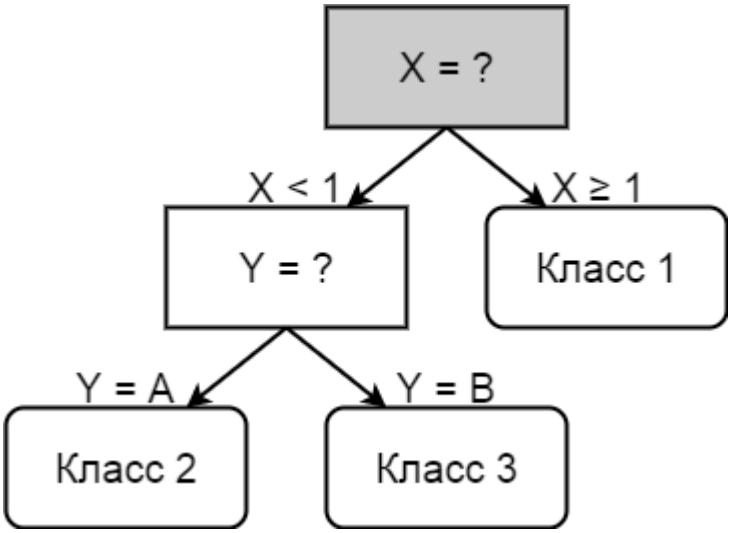

Рис.5. Пример дерева решений.

2. ТРЕБОВАНИЯ К СИСТЕМЕ

Система аналитической обработки данных для системы бронирования банкетных комплексов анализирует существующую базу заявок и строит прогноз выручки на будущие периоды для оценки эффективности работы отдела продаж в текущем периоде.

2.1. Функциональные требования

Разрабатываемая система аналитической обработки данных должна удовлетворять следующим функциональным требованиям:

1) система аналитической обработки данных должна визуализировать обработанную информацию;

2) система аналитической обработки данных должна предоставлять возможность сформировать отчет по выбранным данным;

3) система аналитической обработки данных должна прогнозировать динамику количества заявок и объема выручки на текущий год.

2.2. Варианты использования системы

В ходе выполнения выпускной квалификационной работы бакалавра мною была разработана диаграмма вариантов использования [13] системы аналитической обработки данных, которая изображена на рис. 6. Диаграмма вариантов использования — диаграмма, отражающая отношения между актёрами и прецедентами и являющаяся составной частью модели прецедентов, позволяющей описать систему на концептуальном уровне. Варианты использования обычно применяются для спецификации внешних требований к системе

Диаграмма включает в себя следующих основных актеров, взаимодействующих с системой:

1) *пользователь* – это актер, использующий систему аналитической обработки;

2) *администратор* – это актер, обеспечивающий работоспособность и отсутствие сбоев в периодическом обновлении данных, которые обрабатывает система.

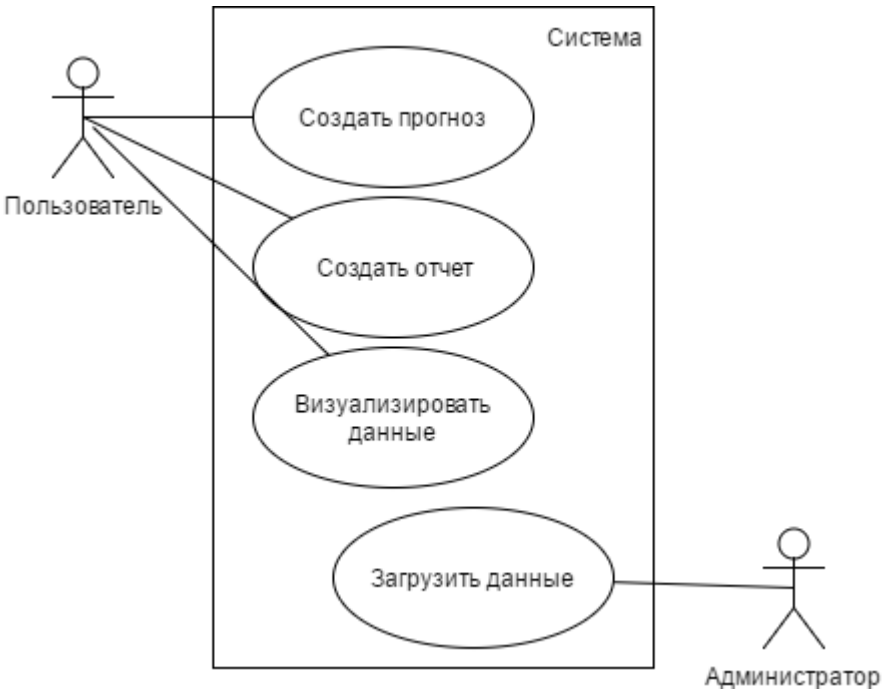

Рис. 6. Диаграмма вариантов использования

Можно определить следующие основные варианты использования системы аналитической обработки данных.

Пользователь:

– «создать прогноз» – запускает генерацию прогноза на основании выбранных данных;

– «создать отчет» – формирует отчет с определенными параметрами;

– «визуализировать данные» – выбирает источник данных для графического отображения.

Администратор:

– «загрузить данные» – осуществляет запуск процедуры обработки и записи в хранилище данных новой информации.

3. ПРОЕКТИРОВАНИЕ СИСТЕМЫ

3.1. Компоненты системы

На рис. 7 изображена диаграмма компонентов системы аналитической обработки данных.

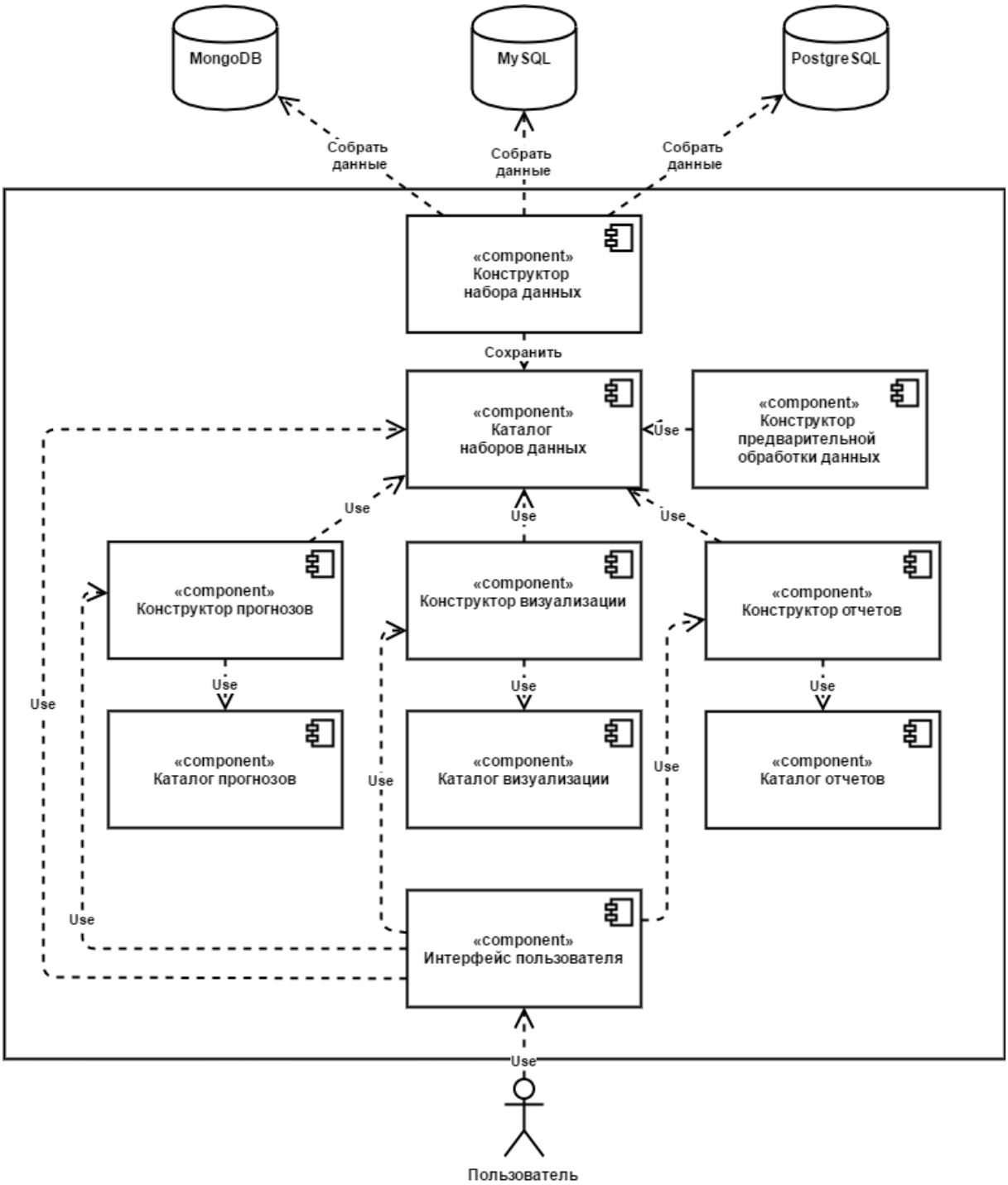

Рис. 7. Компоненты системы аналитической обработки данных

Диаграмма показывает разбиение программной системы на структурные компоненты и зависимости между компонентами.

Система включает в себя следующие компоненты:

1) интерфейс пользователя – компонент, предназначенный для взаимодействия пользователя с системой;

2) конструктор набора данных – компонент сбора данных из внешних СУБД;

3) каталог наборов данных – компонент хранения данных, прошедших предварительную обработку;

4) конструктор предварительной обработки данных – компонент, осуществляющий обработку, слияние, обогащение, группировку и разбиение входных данных;

5) конструктор отчетов – компонент создания отчета на основе выбранных наборов данных;

6) каталог отчетов – компонент, реализующий хранение созданных отчетов;

7) машинное обучение – компонент для осуществления процесса формирования прогноза;

8) каталог прогнозов – компонент для хранения готовых наборов данных прогнозов;

9) конструктор визуализации – компонент, позволяющий представить выбранную информацию каталогов в наглядном виде;

10) каталог визуализации – компонент, хранящий ранее сформированные визуальные представления.

Компонент «Конструктор набора данных» должен осуществлять извлечение актуальных данных из внешних источников, в том числе реляционных и нереляционных СУБД. Предварительная и аналитическая обработка производится только для объектов, которые есть в каталоге наборов данных, поэтому необходимо обеспечить корректное добавление новых наборов данных. На основе компонентной диаграммы системы аналитической обработки данных была

разработана диаграмма последовательности действий добавления новых наборов данных в каталог наборов данных.

Диаграмма последовательности – диаграмма, используемая в языке UML, на которой для некоторого набора объектов на единой временной оси показан жизненный цикл какого-либо определённого объекта и отправка запросов и получение ответов.

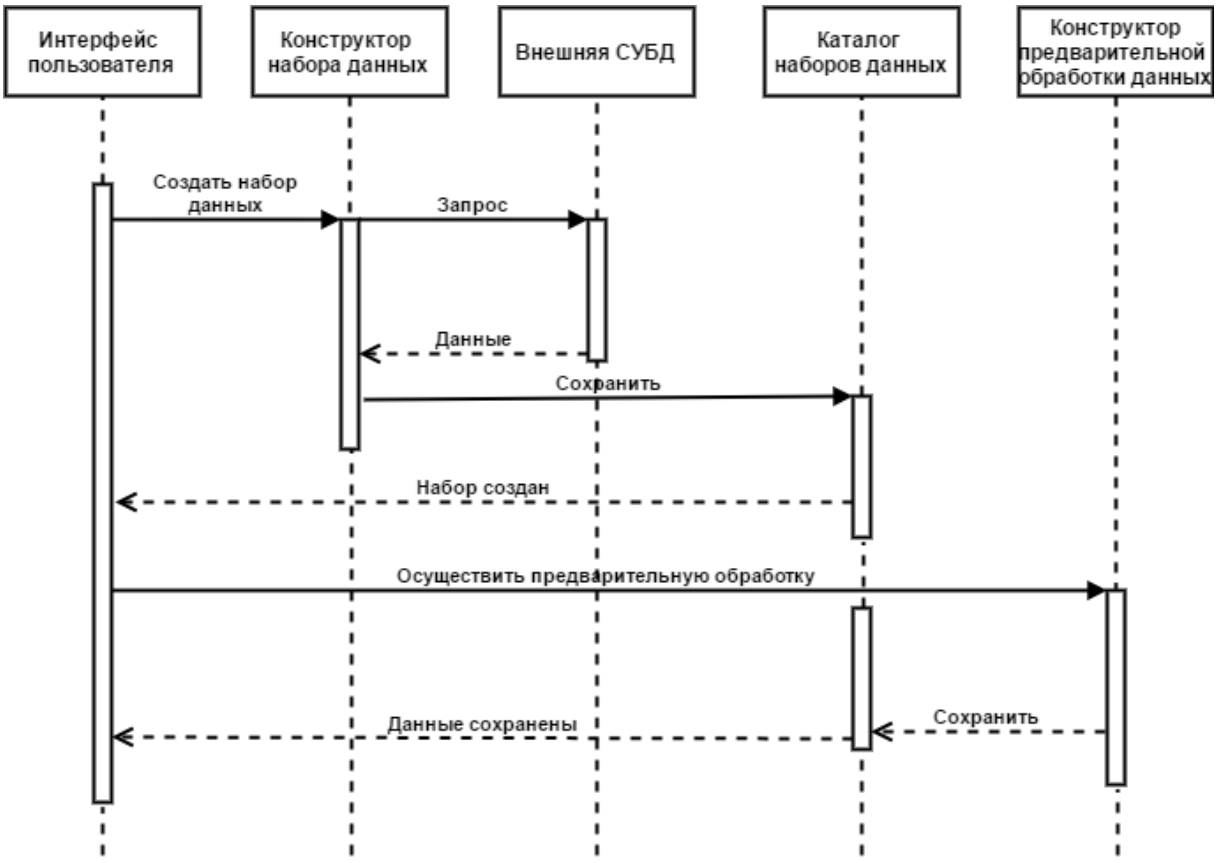

Рис. 8. Диаграмма последовательности действий добавления новых данных в каталог наборов данных

Компонент «Конструктор прогнозов» должен обеспечивать возможность получения прогноза на основании выбранных элементов каталога данных и заданных параметрах. Машинное обучение – класс методов, характерной чертой которых является не прямое решение задачи, а обучение в процессе применения решению множества сходных задач. Для построения таких методов используются средства математической статистики, численных методов, методов оптимизации, теории вероятностей, теории графов, различные техники работы с

данными в цифровой форме. В данной работе используется готовый алгоритм дерева решений, представленный в платформе Dataiku DSS. На основе компонентной диаграммы была разработана диаграмма последовательности действий создания нового прогноза.

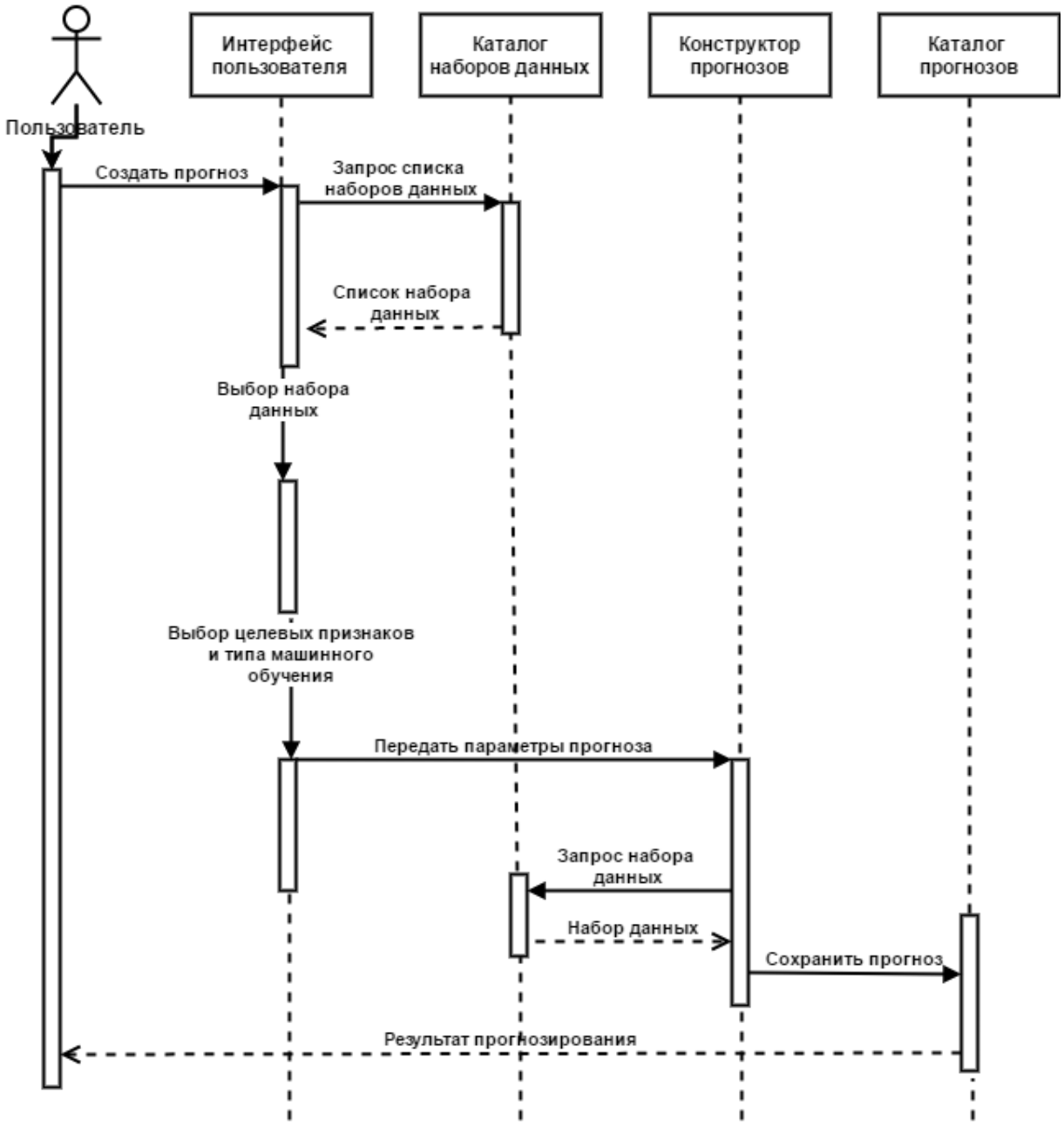

Рис. 9. Диаграмма последовательности действий создания нового прогноза

3.2. Схема хранилища данных

Хранилище данных является одним из главных компонентов аналитической системы, т.к. обработка хранящейся именно в нем информации составляет основу работы системы. Схема базы данных каталога наборов данных с исход-

ной информацией при использовании реляционной СУБД изображена на рис. 10. Диаграмма построена с использованием стандарта Бахмана [2].

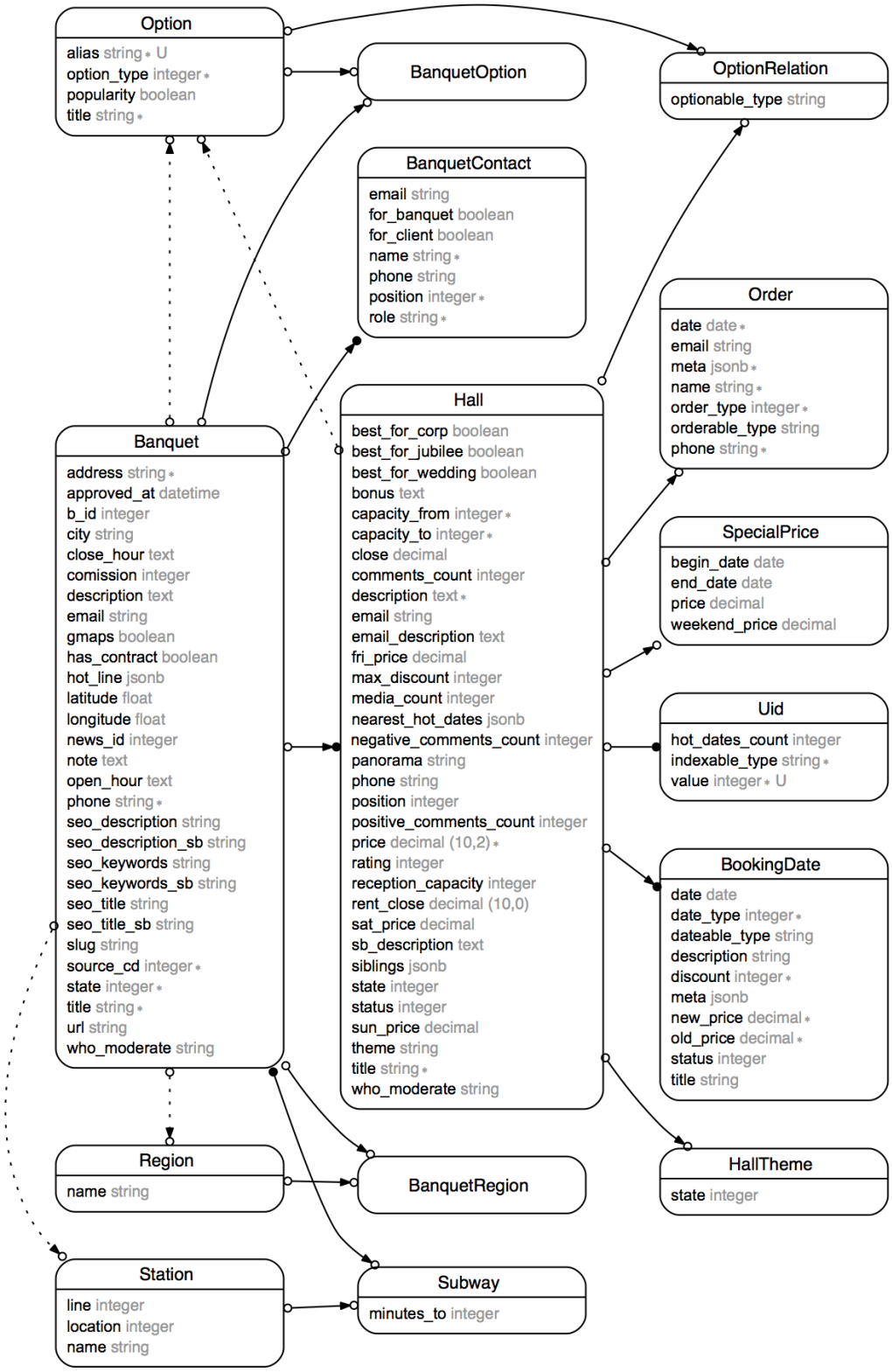

Рис. 10. Схема исходной реляционной СУБД

Схематичное изображение структуры исходной базы данных заявок, оформленная с помощью стандарта JSON Schema представлена на рис. 11. В качестве СУБД для хранения используется нереляционная документ-ориентированная система MongoDB.

Рис.11. Структура исходной БД заявок

4. РЕАЛИЗАЦИЯ СИСТЕМЫ

4.1. Выбор средств реализации системы

В качестве аналитической платформы была выбрана Dataiku Data Science Studio, которая обладает всеми необходимыми инструментами:

– средства консолидации информации в едином источнике (хранилище данных);

– извлечение, преобразование, трансформацию данных;

– алгоритмы Data Mining;

– средства визуализации.

Для разработки компонентов использовался язык программирования Python 3.6.1 [12, 17].

4.2. Реализация компонента конструктора наборов данных

Компонент конструктора наборов данных реализован в виде класса DatasetConstructor, основными методами которого являются:

– конструктор класса, получающий данные об источнике данных и запросе для получения информации;

– метод generate() формирует JSON-документ с конфигурацией набора данных, а также осуществляет первичное наполнение набора данных при помощи метода update();

– метод update() используется для выполнения запроса к источнику данных.

4.3. Реализация компонента каталога наборов данных

Компонент каталога наборов данных реализован в виде класса DatasetCatalogue, основными методами которого являются:

– конструктор класса;

– метод list() возвращает список наборов данных в системе;

– метод delete() используется для удаления набора данных.

4.4. Реализация компонента конструктора предварительной обработки набора данных

Компонент предварительной обработки набора данных реализован в виде класса DatasetProcessor, формирующего JSON-конфигурацию для обработчика данных, основными методами которого являются:

– конструктор класса, получает и сохраняет данные об исходном наборе данных для предварительной обработки;

– метод addStep() добавляет шаг предварительной обработки. Шаг может представлять собою объект в JSON-нотации, описывающий применение готовой хранимой процедуры с параметрами (например, удаление колонок, конкатенацию строк, преобразование JSON-содержимого), обработку написанной пользователем GREL-формулы или выполнение пользовательского кода на языке программирования Python;

– метод deleteStep() используется для удаления шага;

– метод generate() возвращающий JSON-документ с описанием обработки данных;

– метод run() выполняет при помощи соответствующих типу шага процессоров шаги обработки из JSON-документа и формирующий новый обработанный набор данных.

В рамках данной работы были реализованы несколько готовых хранимых процедур, процессор для обработки пользовательского кода на языке программирования Python при помощи интерпретатора и процессор для обработки пользовательских формул на базе системы OpenRefine [11]. Пользовательский код на языке программирования Python должен быть функцией, получающей в качестве параметра словарь записи из набора данных и возвращающей новую запись в виде словаря. Пользовательские формулы должны быть оформлены по стандарту GREL [7]. Результат обработки полученного из исходного набора данных JSON-документа процессором предварительной обработки представляет собой новый набор данных.

```
{
                "type": "CreateColumnWithGREL",
                "params": {
                  "column": "intermediary_sum",
                  "expression": "if(state\u003d\u003d\"done\" ||
        state\u003d\u003d\"completed\", if(intermediary_id\u003e0,
        if(state\u003d\u003d\"done\", money/2, complitedHallMoney/2), 0),
        0)"
                },
                "preview": false,
                "disabled": false
              }
```


Рис. 12. Фрагмент JSON-конфигурации шагов предварительной обработки с применением GREL-формулы

До обработки исходного набора данных процессором предварительной обработки набор данных имеет следующий список полей:

– updatedAt: Date;

– hits: Integer;

– checkedHalls: Array;

– comment: Text;

– comments: Text;

– complitedHall: Integer;

– createdAt: Date;

– customer: Object;

– editors: Array;

– filters: Object;

– halls: Array;

– history: Array;

– id: Integer;

– letters: Array;

– merged: Boolean;

– money: Integer;

– sentHalls: Array;

– sentToClientHalls: Array;

– state: Text;

– stateUpdatedAt: Date;

– manager: Integer;

– reason: Text;

– complitedHallMoney: Integer;

– complitedHallData: Object;

– deletedHalls: Array;

– restoredFrom: Data;

– intermediaryId: Integer;

– visitedHalls: Array;

– deletedData: Object;

– intermediary_money: Integer;

– postponedTo: Date.

Созданный после обработки исходного набора данных процессором предварительной обработки набор данных имеет следующий список полей:

– completedHall: Integer;

– filters_date: Date;

– people_max: Integer;

– people_min: Integer;

– id: Integer;

– money: Integer;

– state: Text;

– reason: Text;

– completedHallMoney: Integer;

– completedHallData: Object;

– intermediaryId: Integer;

– intermediary_sum: Decimal;

– visitedHalls: Array;

– deletedData: Object;

– intermediary_money: Integer;

–join_price: Decimal;

– expected_profit: Decimal;

–remains: Integer;

–join_capacity_to: Integer;

–join_capacity_from: Integer;

–join_id: Integer;

–join_rating: Text;

–join_banquet_id: Integer;

–join_open_hour: Text;

–join_close_hour: Text;

–join_address: Text;

–join_latitude: Decimal;

–join_longtitude: Decimal;

–join_region_id: Integer.

```
{
      "type": "ColumnRenamer",
      "params": {
        "renamings": [
          {
            "from": "filters.date.$date",
            "to": "filters_date"
          }
        ]
      },
      "preview": false,
      "disabled": false
      }
```

Рис. 13. Фрагмент JSON-конфигурации шагов предварительной обработки с применением хранимой процедуры для переименования колонки

```
# -*- coding: utf-8 -*-

import json
import sys

def process(row):
    history = json.loads(unicode(row['history'], errors='ignore'))
    result = history[0]
    completes = [x for x in history if 'state' in x and (x['state'] ==
'processed' or x['state'] == 'to_completed')]
    if len(completes) > 0:
      last_complete = max(completes, key=lambda
last_complete:last_complete['date']['$date'])
      return last_complete['date']['$date']
    else:
      return ""
```

Рис. 14. Исходный код пользовательской Python-функции, используемой в качестве шага для предварительной обработки данных в наборе

4.5. Реализация компонента визуализации

Компонент визуализации реализован в виде класса ReportVisualizer и обладает следующими основными методами:

– конструктор класса получает и сохраняет информацию об исходном наборе данных для визуализации;

– метод generate() получает следующие параметры: движок визуализации, способ визуализации, набор фильтров для визуализации, состав осей и целевые переменные для визуализации и формирует JSON-документ с описанием конфигурации для визуализации.

4.6. Реализация компонента построения прогноза

Компонент построения прогноза реализован в виде класса PredictionsConstructor, основными методами которого являются:

– конструктор класса получает и сохраняет данные об исходном наборе данных;

– метод createPrediction() получает в качестве параметров указание алгоритма классификации (например: ‘decisionTree’ для построения дерева реше-

ний), список целевых переменных, настройки признаков. Результатом выполнения данного метода является размещение нового прогноза в каталоге прогнозов PredictionsCatalogue.

Платформа Dataiku DSS предоставляет закрытую реализацию алгоритма дерева решений. В результате наблюдений было установлено, что для принятия решения алгоритм проверяет заданные атрибуты и согласно их значениям делит элементы набора данных на подмножества. Предполагаемая упрощенная схема дерева решений представлена на рис. 15.

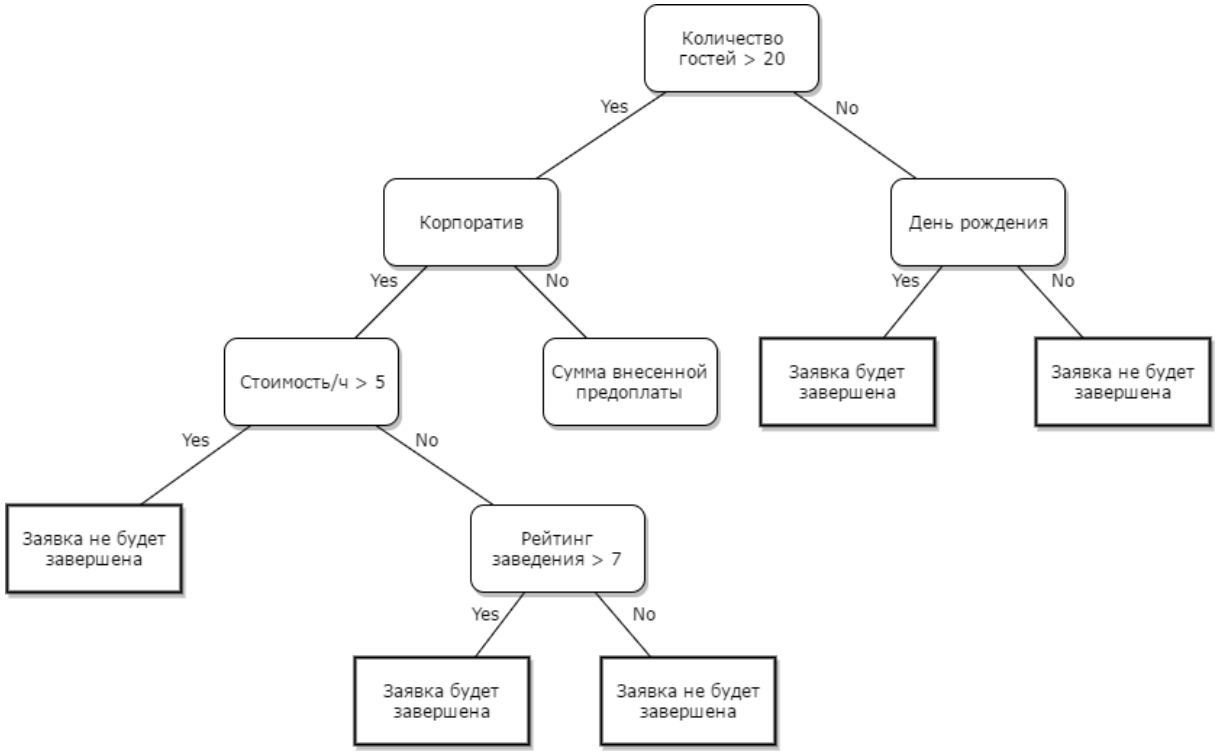

Рис. 15. Схема дерева решений по состоянию заявки

4.7. Реализация каталога прогнозов

Компонент каталога прогнозов реализован в виде класса PredictionsCatalogue, основными методами которого являются:

– конструктор класса;

– метод list() используется для получения списка прогнозов;

– метод delete() используется для удаления прогноза.

4.8. Создание отчетов, прогнозов и визуализаций

С помощью реализованных компонентов в системе были созданы отчеты о текущем состоянии БД заявок, эффективности менеджеров компании, количестве и динамике поступающих заявок, выручке от обработки заявок. На базе полученных отчетов при помощи компонента построения прогнозов был сформирован прогноз о конверсии заявок. Полученный прогноз предназначен для поддержки решений, связанных с оценкой текущей эффективности обработки базы заявок и предполагаемой выручки. Результат предоставлен на рис. 16.

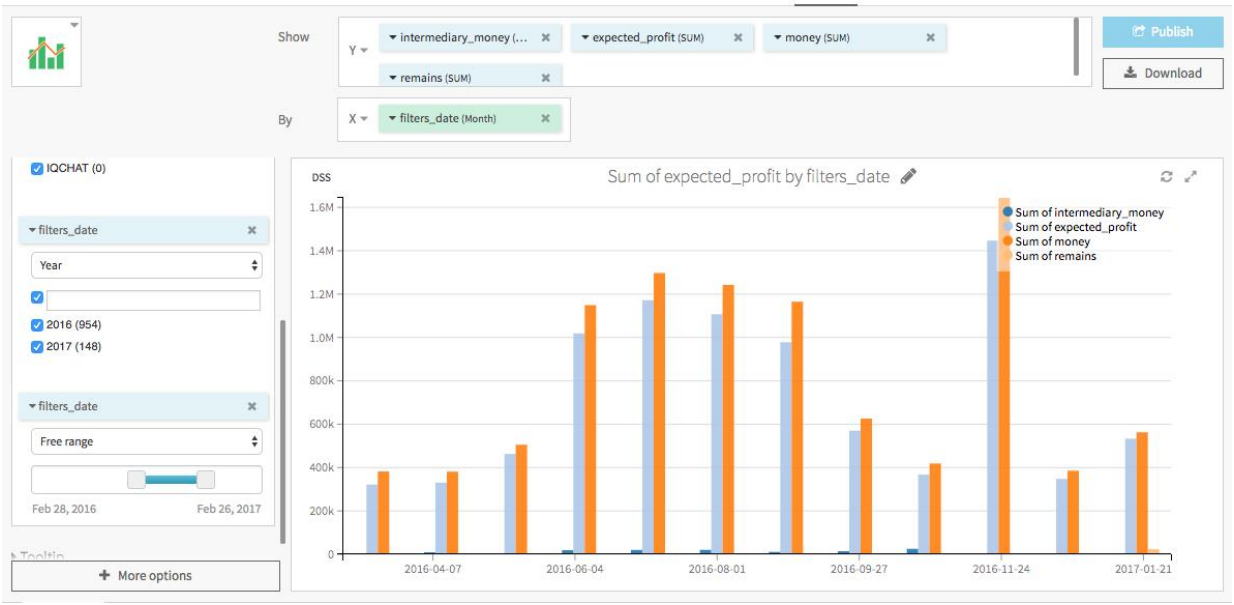

Рис. 16. Пример визуализации прогноза выручки и фактической выручке компании по месяцам

5. ТЕСТИРОВАНИЕ

Для тестирования системы использовалось функциональное тестирование.

Функциональное тестирование – это тестирование программного обеспечения в целях проверки реализуемости функциональных требований, то есть способности программного обеспечения в определенных условиях решать задачи, нужные пользователям [16]. Результаты функционального тестирования совпали с ожидаемыми (требуемыми). Примеры результатов тестирования приведены в таблице.

Табл. Результаты функционального тестирования.

№	Содержание теста	Ожидаемый результат	Тест пройден?
1	Нажатие на кнопку «Update Data»	Уведомление об успешном добавлении в хранилище новых данных (до и после обработки)	Да
2	Нажатие на кнопку «Create Income Prediction»	Уведомление об успешном создании прогноза на основании указанных параметров	Да
3	Нажатие на кнопку «Visualize Data»	Открытие страницы с готовой настраиваемой визуализацией	Да

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работы бакалавра мною была реализована система аналитической обработки данных для системы бронирования банкетных комплексов и решены следующие задачи:

– проведен анализ предметной области;

– спроектирована система аналитической обработки данных;

– выбраны средства разработки;

– реализованы компоненты каталогов для хранения наборов данных;

– разработаны компоненты для предварительной обработки данных и создания отчетов и прогнозов;

– проведено тестирование системы.

ЛИТЕРАТУРА

1. Aggarwal C. C. On k-anonymity and the curse of dimensionality. Proceedings of the 31st international conference on Very large data bases, VLDB Endowment, 2005. – P. 901-909.
2. Bachman C. W. et al. System and method of constructing models of complex business transactions using entity-set variables for ordered sets of references to user data : пат. 5249300 США. – 1993.
3. Chen H., Chiang R. H. L., Storey V. C. Business intelligence and analytics: From big data to big impact. MIS quarterly, 2012. – Т. 36. – №. 4. – P. 1165-1188.
4. Dataiku Data Science Studio. [Электронный ресурс] URL: https://www.dataiku.com/ (дата обращения: 28.04.2017).
5. Dreslinski R., Wieckowski M., Blaauw D., Sylvester D. Near-threshold computing: Reclaiming moore's law through energy efficient integrated circuits. Proceedings of the IEEE, 2010. – P. 253-266.
6. Elbashir M. Z., Collier P. A., Davern M. J. Measuring the effects of business intelligence systems: The relationship between business process and organizational performance. International Journal of Accounting Information Systems, 2008. – Т. 9. – №. 3. – P. 135-153.
7. GREL Functions. [Электронный ресурс] URL: https://github.com/OpenRefine/OpenRefine/wiki/GREL-Functions (дата обращения: 25.04.2017).
8. High-Tech Tuesday Webinar: Uncovering Growth Opportunities in the Transforming Business Analytics Market. [Электронный ресурс] URL: https://www.gartner.com/doc/3537817 (дата обращения: 20.03.2017)
9. Larose D. T. Discovering knowledge in data: an introduction to data mining. – Canada: John Wiley & Sons, 2014. – 336 p.
10. Laursen G. H. N., Thorlund J. Business analytics for managers: Taking business intelligence beyond reporting. – Canada: John Wiley & Sons, 2016. – 288 p.

11. OpenRefine. [Электронный ресурс] URL: http://openrefine.org/ (дата обращения: 25.04.2017).

12. Python. [Электронный ресурс] URL: https://www.python.org/ (дата обращения: 30.04.2017).

13. Арлоу Д., Нейштадт А. UML-2 и Унифицированный процесс. Практический объектно-ориентированный анализ и проектирование. – М.: Символ, 2008. – 621 с.

14. Барсегян А. А. Анализ данных и процессов. – Спб.: БХВ-Петербург, 2009. – 512 с.

15. Барсегян А. Технологии анализа данных: Data Mining, Text Mining, Visual Mining, OLAP. – Спб.: БХВ-Петербург, 2008. – 375 с.

16. Кем К., Фолк Д., Нгуен Е. К. Тестирование программного обеспечения. Фундаментальные концепции менеджмента бизнес-приложений. – Киев: ДиаСофт, 2001. – 538 с.

17. Маккинли У. Python и анализ данных. – М.: ДМК Пресс, 2015. –482 с.

18. Паклин Н. Б., Орешков В. И. Бизнес-аналитика: от данных к знаниям. – СПб.: Питер, 2009. – 624 с.