

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий

(наименование института полностью)

Кафедра Прикладная математика и информатика

(наименование)

09.04.03 Прикладная информатика

(код и наименование направления подготовки, специальности)

Информационные системы и технологии корпоративного управления

(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ)

на тему «Математическое и программное обеспечение системы поддержки принятия решений на основе витрин больших данных»

Студент

И.В. Борсук

(И.О. Фамилия)

(личная подпись)

Руководитель

д.т.н., доцент, С.В. Мкртычев

(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

Оглавление

Введение.....	4
Глава 1 Анализ современного состояния исследований в области разработки математического и программного обеспечения систем принятия управленческих решений на основе витрин данных.....	9
1.1 Модели систем поддержки принятия решений на основе витрин данных.....	12
1.2 Модели систем поддержки принятия решений на основе бизнес-анализа больших данных.....	18
Глава 2 Анализ методологий и технологий построения систем поддержки принятия решений на основе витрин больших данных	23
2.1 Методы построения витрин данных	23
2.2 Технологии построения BI-систем для анализа больших объемов данных.....	32
2.3 Технология проектирования программного обеспечения BI-системы	35
2.4 Методологии проектирования программного обеспечения BI-системы	39
Глава 3 Математическое обеспечение системы поддержки принятия решения на основе витрин больших данных	46
3.1 Логическое моделирование системы поддержки принятия решения на основе витрин больших данных	46
3.2 Разработка логической модели данных системы поддержки принятия решения на основе витрин больших данных	52
Глава 4 Программное обеспечение системы поддержки принятия решения на основе витрин больших данных	55
4.1 BI-система MS Power BI.....	57
4.2 BI-система Tableau.....	59
4.3 BI-система QlikView.....	60

4.4 Апробация выбранных проектных решений и оценка эффективности VI-системы.....	62
Заключение	67
Список используемой литературы и используемых источников.....	69

Введение

«В современном мире успех предприятия и компании напрямую зависит от того, как быстро ее менеджмент реагирует на изменения конъюнктуры рынка с целью увеличения прибыли и обеспечения конкурентоспособности.

Менеджеры компании должны отслеживать тенденции рынка, идентифицировать конкурентов и угрозы, оценивать риски, оценивать свои ресурсы и принимать обоснованные управленческие решения.

Информация является необходимым производственным ресурсом для принятия эффективных управленческих решений. Менеджерам необходимо, чтобы эта информация была преобразована, предварительно обработана и соответствующим образом организована для быстрого доступа, анализа и принятия решений» [10].

Следует отметить, что принятие решений на основе данных (Data-driven decision making) - это процесс принятия решений в организации, основанный на фактических данных, а не только на интуиции или наблюдении.

Для решения вышеперечисленных задач используются аналитические системы поддержки принятия решений, относящиеся к классу BA/BI (Business Analytics/Business Intelligence) - систем.

Современные предметно-ориентированные BI – системы строятся на основе концепции витрин данных [23].

Витрина данных – это простая форма хранилища данных, ориентированная на конкретное направление бизнеса или подразделение предприятия. Благодаря витрине данных разработчики могут быстрее получать доступ к данным, поскольку им не нужно тратить время на поиск в более сложном хранилище данных или ручное агрегирование данных из разных источников [39].

В современных условиях предприятия и компании накопили значительные объемы данных и имеют доступ к еще большим объемам внешних данных, которые позволяют рассматривать последние как большие данные (Big Data).

В этой связи возникает проблема проектирования VI – систем, компонентами которых будут витрины больших структурированных массивов данных.

Совершенно очевидно, что высокая эффективность таких систем может быть достигнута за счет применения при их разработке специализированного математического и программного обеспечения.

Разработка математического и программного обеспечения, позволяющего повысить эффективность систем поддержки принятия управленческих решений, построенных на основе витрин больших данных, представляет научно-практический интерес.

Актуальность темы исследования обусловлена необходимостью исследования и разработки математического и программного обеспечения, позволяющего повысить эффективность систем поддержки принятия управленческих решений на основе витрин больших данных.

Объектом настоящего исследования является система поддержки принятия решений.

Предметом исследования является математическое и программное обеспечение системы поддержки принятия решений.

Целью работы является исследование и разработка математического и программного обеспечения системы поддержки принятия решений, построенной на основе витрины больших данных.

Для достижения поставленной цели необходимо решать следующие задачи:

- провести анализ современного состояния исследований проблемы разработки математического и программного обеспечения систем поддержки принятия управленческих решений на основе витрин больших данных;
- провести анализ методологических подходов к построению систем поддержки принятия управленческих решений на основе витрин больших данных;

- разработать математическое и программное обеспечение систем поддержки принятия управленческих решений на основе витрин больших данных;
- выполнить апробацию предлагаемых проектных решений и оценить их эффективность.

Гипотеза исследования: применение разработанного в рамках диссертационного исследования математического и программного обеспечения обеспечит повышение эффективности системы поддержки принятия управленческих решений, построенной на основе витрин больших данных.

Методы исследования. В процессе исследования будут использованы следующие положения и методы: системный анализ, методологии и технологии проектирования ВА/ВІ – систем, методы и технологии аналитической обработки больших данных.

Новизна исследования заключается в разработке математического и программного обеспечения, которое обеспечит повышение эффективности системы поддержки принятия управленческих решений, построенной на основе витрин больших данных.

Практическая значимость исследования заключается в возможности применения разработанного математического и программного обеспечения в процессе реализации системы поддержки принятия управленческих решений, построенной на основе витрин больших данных.

Теоретической основой диссертационного исследования являются научные труды российских и зарубежных ученых, занимающихся проблемами проектирования информационно-аналитических систем и систем анализа больших данных.

Основные этапы исследования: исследование проводилось с 2019 по 2021 год в несколько этапов.

На первом (констатирующем) этапе формулировалась тема исследования, выполнялся сбор информации по теме исследования из

различных источников, проводилась формулировка гипотезы, определялись постановка цели, задач, предмета исследования, объекта исследования и выполнялось определение проблематики данного исследования.

Второй этап – поисковый. В ходе проведения данного этапа осуществлялся анализ методологий проектирования BI-систем на основе витрин данных, были разработаны модели и алгоритмы системы поддержки принятия решений на основе витрин больших данных, опубликована научная статья по теме исследования в научном сборнике.

На третьем этапе осуществлялась апробация системы поддержки принятия решений на основе витрин больших данных, произведена оценка ее эффективности, сформулированы выводы о полученных результатах по проведенному исследованию.

На защиту выносятся:

- математическое и программное обеспечение системы поддержки принятия решений на основе витрин больших данных;
- результаты апробации и оценки эффективности системы поддержки принятия решений на основе витрин больших данных.

По теме исследования опубликована 1 статья: Борсук И.В. Модель функционирования ETL-процесса // Вестник научных конференций. 2021. №10-3(74). С. 19-20.

Диссертация состоит из введения, четырех глав, заключения и списка литературы.

Во введении обоснована актуальность темы исследования, представлены объект, предмет, цели, задачи и положения, выносимые на защиту диссертации.

В первой главе дан анализ современного состояния исследований в области разработки математического и программного обеспечения систем принятия управленческих решений на основе витрин данных. Рассмотрены модели систем поддержки принятия решений на основе витрин данных и анализа больших данных.

Вторая глава посвящена анализу методологий и технологий построения систем поддержки принятия решений на основе витрин больших данных.

Третья глава посвящена разработке математического обеспечения системы поддержки принятия решения на основе витрин больших данных. Описан процесс логического моделирования системы поддержки принятия решения на основе витрин больших данных. Описан процесс разработки логической модели данных системы поддержки принятия решения на основе витрин больших данных.

В четвертой главе описан процесс разработки программного обеспечения системы поддержки принятия решения на основе витрин больших данных. Представлены результаты апробации выбранных проектных решений и оценки эффективности VI-системы.

В заключении приводятся результаты исследования.

Работа изложена на 73 страницах и включает 30 рисунков, 7 таблиц, 40 источников.

Глава 1 Анализ современного состояния исследований в области разработки математического и программного обеспечения систем принятия управленческих решений на основе витрин данных

«Система поддержки принятия решений (СППР) – компьютерная автоматизированная система, целью которой является помощь людям, принимающим решение в сложных условиях для полного и объективного анализа предметной деятельности.

Математическое и программное обеспечение информационной системы (ИС) – это совокупность математических методов, моделей, алгоритмов и программ для реализации целей и задач ИС, а также нормального функционирования комплекса технических средств» [4].

Согласно концепции компании Gartner, в технологии ВІ выделяются следующие направления:

- «ВІ-сервисы (ВІ services) – это предложения для проектирования, разработки и развертывания корпоративных процессов, а также для интеграции, поддержки и управления соответствующими технологическими приложениями и платформами. К ним относятся бизнес-приложения и инфраструктурные приложения для платформ бизнес-аналитики, аналитические потребности и инфраструктура хранилищ данных. Решения включают такие области, как корпоративное управление производительностью (СРМ) и аналитика, в дополнение к традиционной платформе бизнес-аналитики, хранилищам данных / инфраструктуре данных и областям качества данных» [24].
- ВІ-платформы (ВІ platforms), которые позволяют предприятиям создавать приложения бизнес-аналитики, предоставляя возможности в трех категориях: анализ (OLAP), доставка информации (отчеты и информационные панели) и платформенная интеграция (управление метаданными бизнес-аналитики и среда разработки).

Основные требования, которые предъявляются к современным BI-системам:

- оперативный бизнес-анализ (в реальном времени). Сегодня из-за конкурентного давления предприятий возросла потребность в бизнес-аналитике, близкой к реальному времени, которая называется операционной бизнес-аналитикой. Цель операционной бизнес-аналитики – уменьшить задержку между временем анализа данных и временем сбора данных. Уменьшение времени отклика позволяет системе предпринимать подходящие действия при возникновении события. С помощью оперативной реализации бизнес-аналитики компании могут обнаруживать закономерности или временные тенденции в потоке операционных данных;
- ситуационный бизнес-анализ, обеспечивающий ситуационную осведомленность. В компаниях позиционирование бизнес-аналитики важно там, где быстрая смена позиций, обычно внешние тенденции в бизнесе, повлияли на бизнес. Однако эти внешние данные, которые в основном поступают из внутренней сети компании, внешнего поставщика или Интернета, не структурированы. Более того, эти неструктурированные данные должны быть объединены с другими структурированными данными из локального хранилища данных компании для поддержки маркировки решений в реальном времени. Например, компания может пожелать узнать, оставляют ли ее пользователи и клиенты положительные или отрицательные комментарии о своих новых продуктах. Анализируя эти комментарии, компании могут незамедлительно направлять свои комментарии команде разработчиков, чтобы сделать продукт более конкурентоспособным и квалифицированным. Другой пример – для компании важно знать, повлияли ли стихийные бедствия на их контрактных поставщиков. Это создает возможность для руководителей компаний принимать соответствующие меры по

снижению потерь;

- самостоятельная бизнес-аналитика, позволяющая конечным пользователям создавать анализы и аналитические запросы без участия ИТ-отдела. В такой BI-системе пользовательский интерфейс приложений должен быть простым в использовании и интуитивно понятным, не требующим от пользователя технических знаний в области хранилищ данных. Кроме того, пользователю должен быть разрешен доступ или расширение источников данных, организованных ИТ-отделом, а также нетрадиционных источников.

Согласно этой же концепции, одним из ключевых компонентов BI-систем являются хранилище и витрина данных.

«Хранилище данных (Data warehouse, DW) - это архитектура хранения, предназначенная для хранения данных, извлеченных из систем транзакций, хранилищ операционных данных и внешних источников. Затем хранилище объединяет эти данные в агрегированную сводную форму, подходящую для анализа данных в масштабах предприятия и создания отчетов для предварительно определенных бизнес-потребностей.

Витрина данных (Data mart, DM) содержит аналогично изменяющиеся во времени и предметно-ориентированные данные, но со связями, подразумевающими использование данных в измерениях, при этом факты четко отделены от данных измерений, что делает их более подходящими для отдельных категорий анализа» [23].

Отличительные характеристики витрины данных:

- витрина данных ориентирована только на запросы пользователей, относящиеся к конкретной бизнес-цели;
- витрины данных обычно не содержат оперативные данные;
- витрины данных содержат меньшие объемы данных по сравнению с хранилищами данных и их легче реализовать.

Принципы построения BI-приложений на основе витрин данных изложены в работах Р. Кимбалла [35].

Проблематике проектирования и применения BI-систем для поддержки принятия решений в различных предметных областях посвятили свои работы В. Белов, А.Н. Косенков, С.В. Мкртычев, Е. Никуличев, В. Sudrajat, I. Ahmad, S. Azhar, P. Lukauskis, N. Houari, В. Н. Far и другие.

Некоторые из работ вышеперечисленных авторов рассмотрены в следующих разделах.

1.1 Модели систем поддержки принятия решений на основе витрин данных

Рассмотрим модели систем поддержки принятия решений, построенные на основе витрин данных.

В работе [2] описана структура типового BI-решения, построенного на основе витрин данных, представленная на рисунке 1.

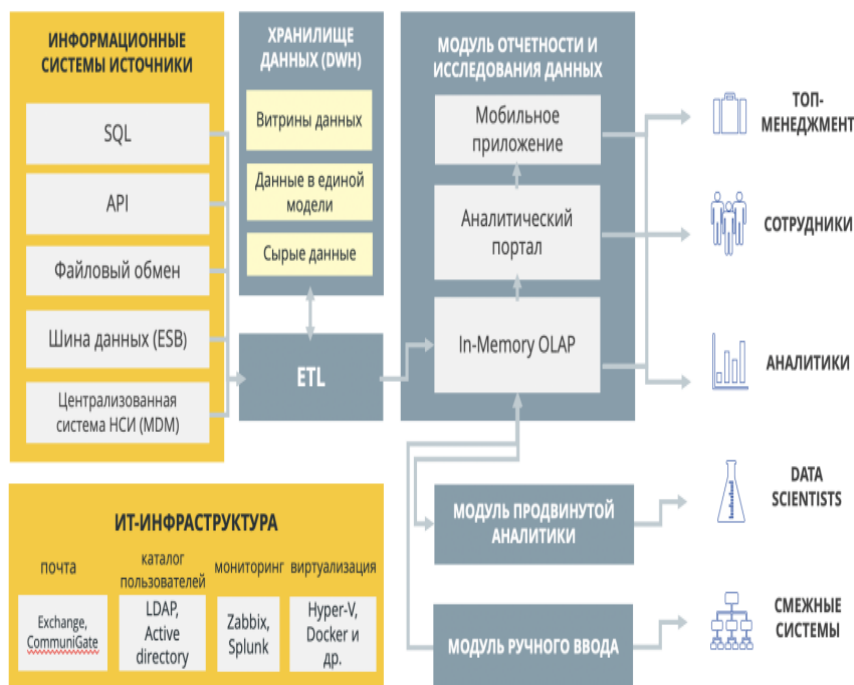


Рисунок 1 - Типовая структура BI-решения на основе витрин данных

Данные, необходимые для работы BI-платформы, извлекаются из различных ИТ-систем с помощью ETL-инструментов и поступают в специальные хранилища данных.

Преобразования данных на уровне хранилища данных, как правило, оркестрируются ETL-инструментом.

Конечным результатом преобразования данных на уровне хранилищ данных являются витрины данных – очищенные наборы данных, к которым пользователи получают прямой доступ с помощью OLAP инструментов [18].

В работе [29] описана информационно-аналитическая система управления андеррайтингом, в которой используется витрина страховых данных (Insurance data mart).

Структурная схема системы представлена на рисунке 2.

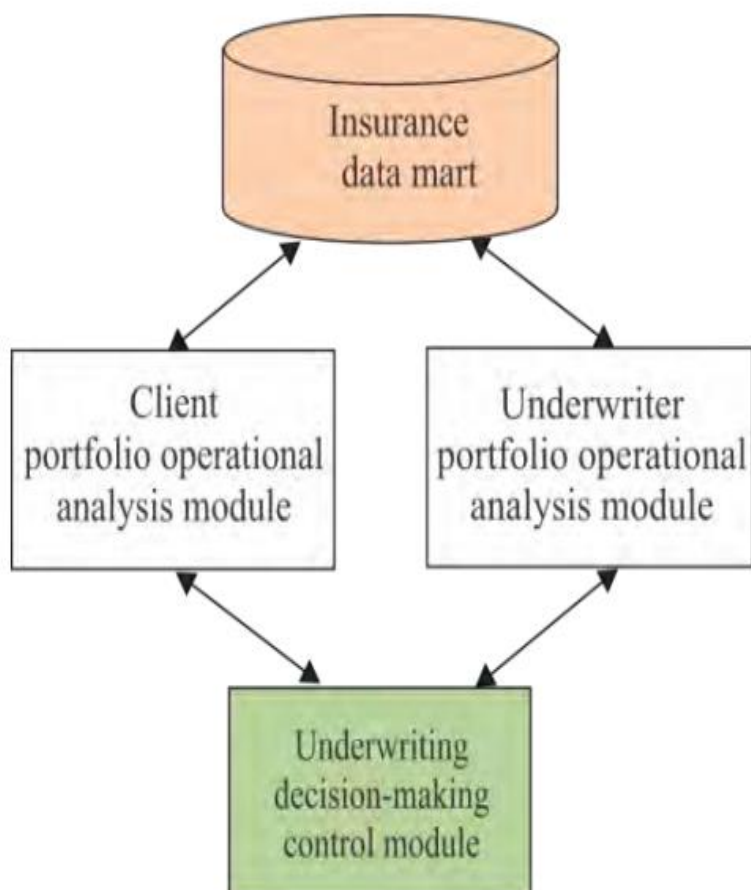


Рисунок 2 - Структурная схема информационно -аналитической системы управления андеррайтингом

Витрина страховых данных построена на основе технологии реляционных OLAP (ROLAP), использующей схему «звезда», как показано на рисунке 3.

Такая модель обеспечивает лучшую производительность запросов при поддержке принятия решений.

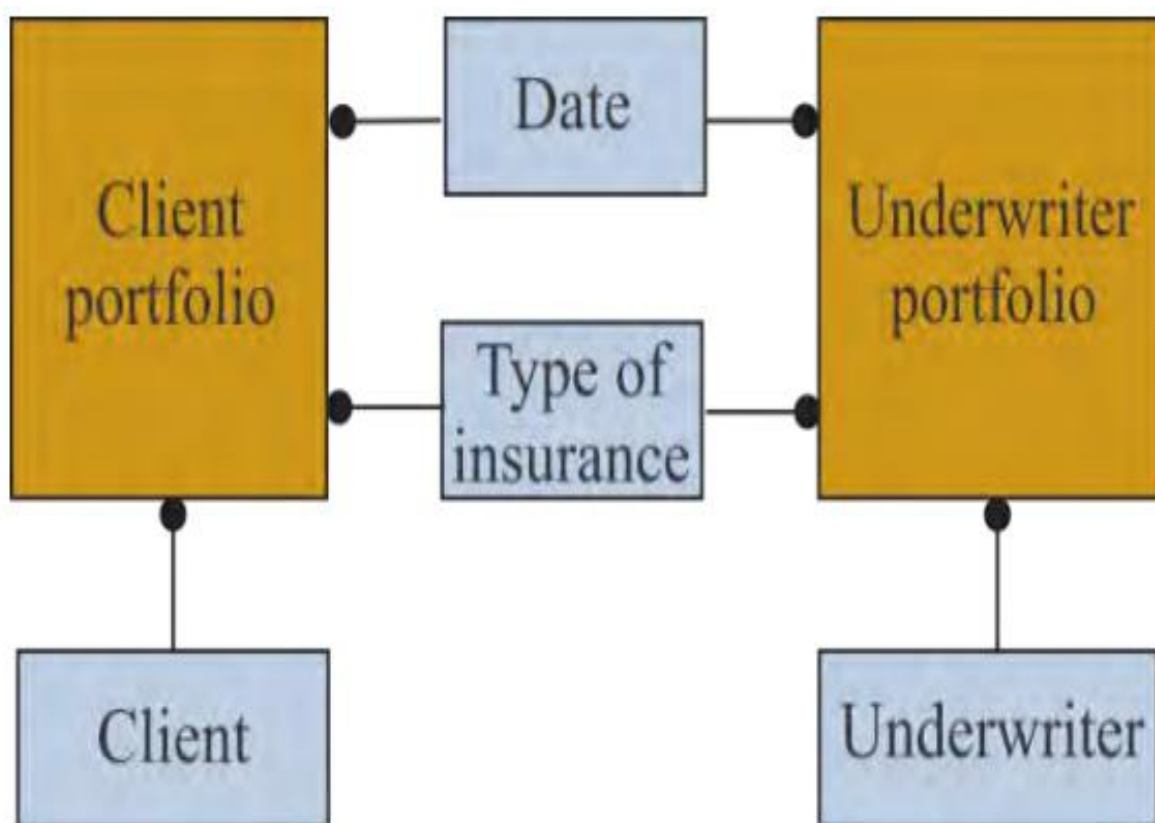


Рисунок 3 – Архитектура витрины страховых данных (Портфель клиента и Портфель Андеррайтера - таблицы фактов; Дата, Клиент, Андеррайтер, Тип страхования - таблицы измерений)

В работе [34] представлены и обсуждаются преимущества разработки и реализации СППР на основе витрины данных.

Предлагается архитектура эффективной СППР телекоммуникационной компании, построенной на основе СУБД MS SQL server (рисунок 4).

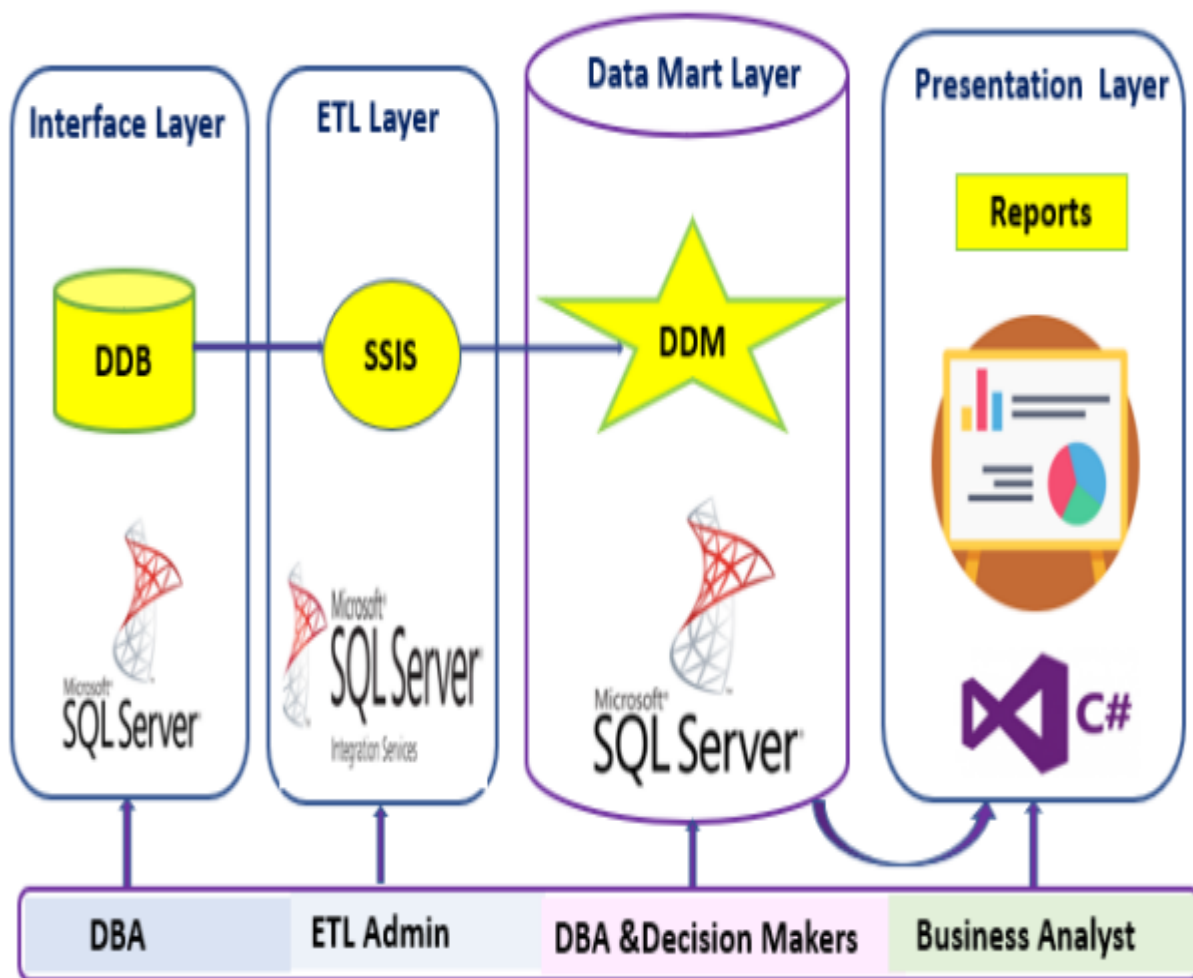


Рисунок 4 - Архитектура эффективной СППР для телекоммуникационной компании на основе витрины данных

В работе [32] обсуждаются проблемы, связанные с системами хранилище данных (DW)/ витрины данных (DM), и предлагается новая архитектура, основанная на мультиагентной технологии, для извлечения информации и знаний из распределенных источников данных для поддержки принятия решений.

Предлагаемая структура использована в жизненном цикле реального проекта EPC (Engineering Procurement and Construction).

Архитектура СППР на основе мультиагентной технологии представлена на рисунке 5.

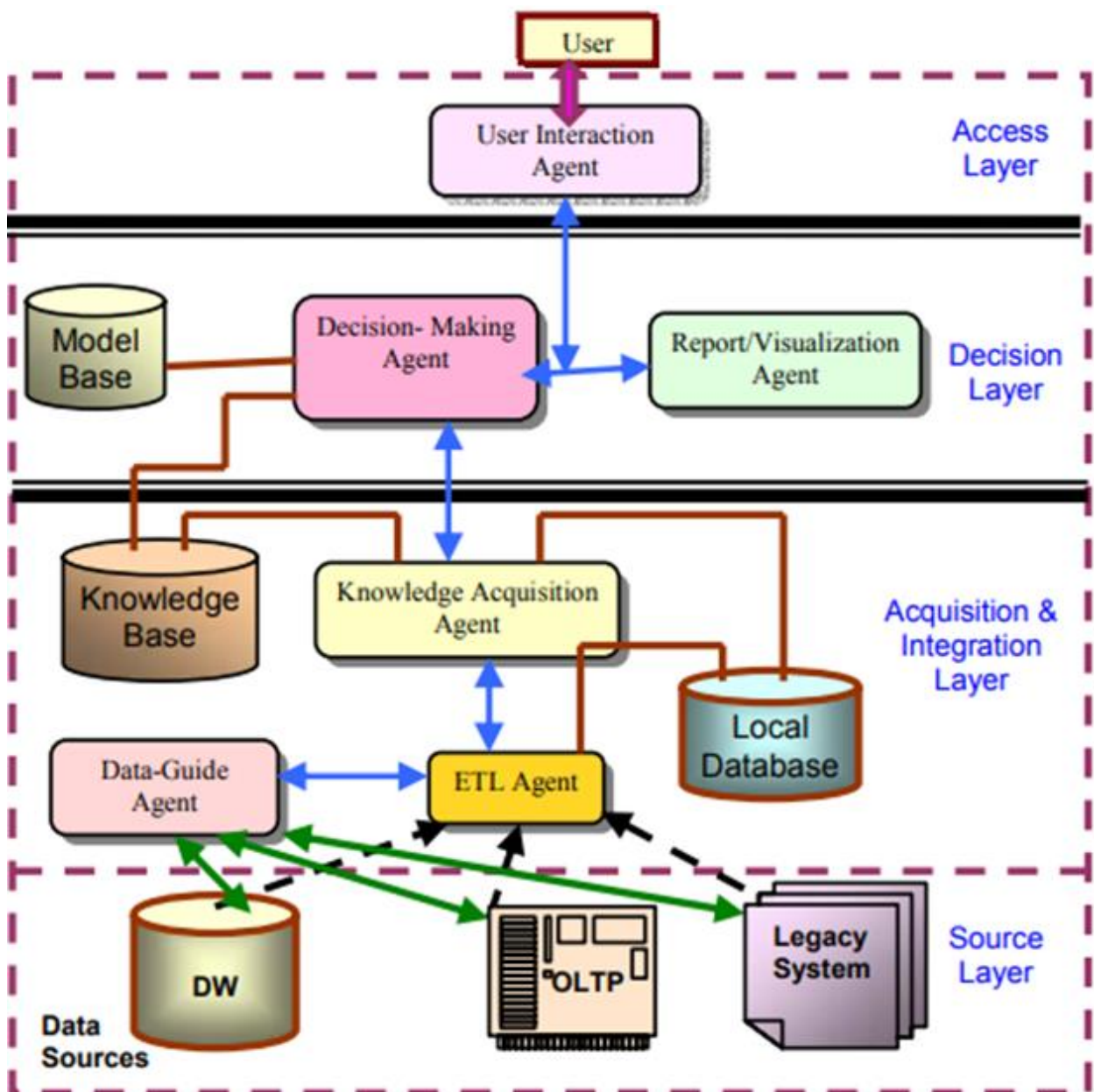


Рисунок 5 – Архитектура СППР на основе мультиагентной технологии

В предлагаемом решении хранилище данных DW состоит из набора зависимых витрин данных.

В работе [25] представлена СППР строителей (застройщиков) при выборе участка для строительства жилого дома.

Концептуальная модель СППР показана на рисунке 6.

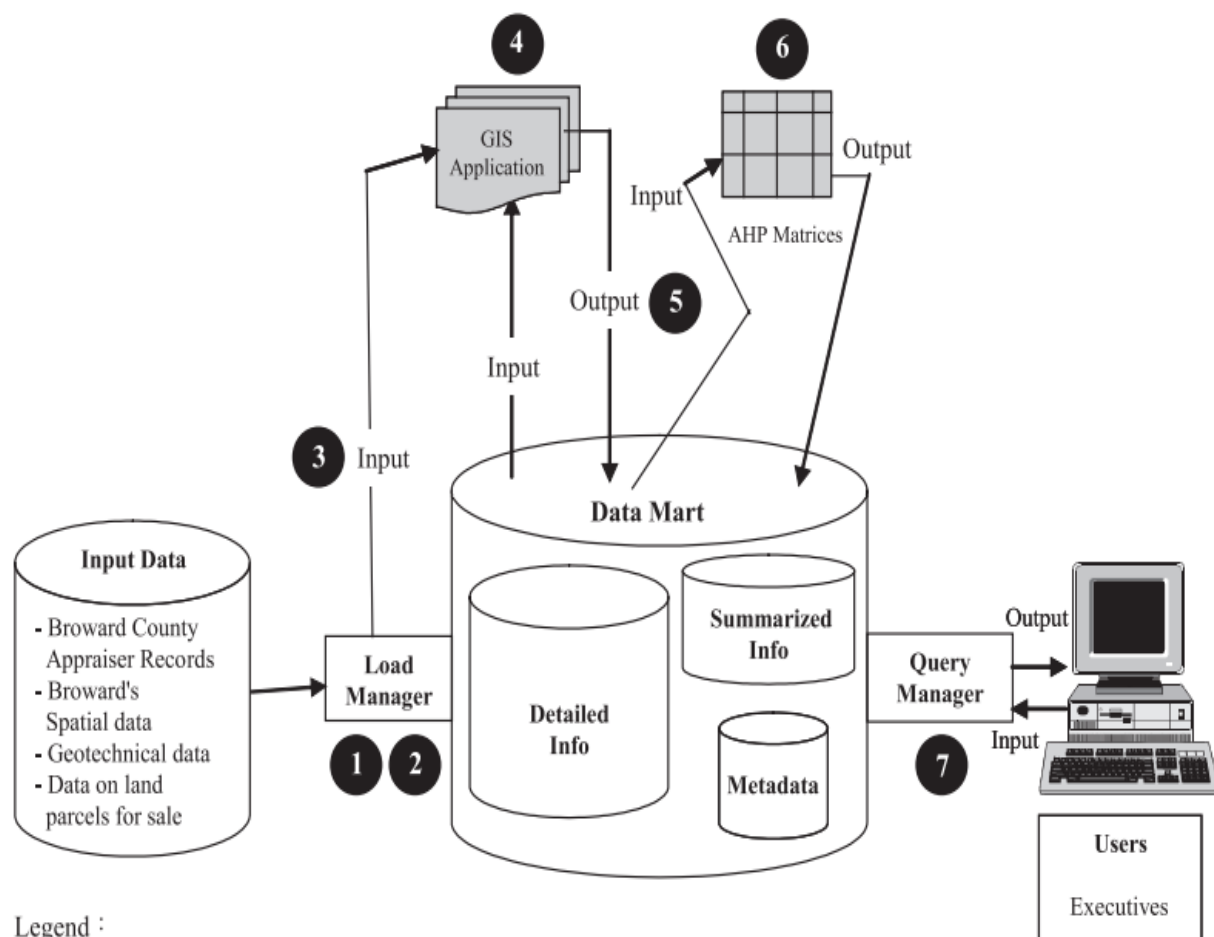


Рисунок 6 - Концептуальная модель СППР при выборе участка для строительства жилого дома

Приложение ГИС экспортирует список участков и их макетов в витрину данных (Data mart) для дальнейшего анализа и принятия решения.

В решении используется подход к созданию хранилища данных, в котором отдельные витрины данных, назначаемые отдельным бизнес-единицам или процессам, разрабатываются и позже интегрируются в хранилище данных в масштабе предприятия.

Главное преимущество данного подхода – относительная простота реализации.

Следует отметить, что рассмотренные выше решения не предназначены для анализа больших массивов данных.

1.2 Модели систем поддержки принятия решений на основе бизнес-анализа больших данных

В теории бизнес-аналитики выделены следующие проблемы традиционных BI-решений в контексте применения для анализа больших данных [16]:

- отсутствие возможностей анализа по требованию: опытным пользователям бизнес-аналитики сегодня не нужно ждать ответов на более сложные бизнес-задачи. Новые пользователи нуждаются в возможностях самообслуживания для связывания и анализа конкретных наборов данных в зависимости от их собственного понимания, для любых целей и в любое время;
- необходим прогнозный анализ: возможности исторических отчетов позволяют получить только одну часть бизнес-анализа – понимание того, что происходило в прошлом. Компании обращаются к прогнозной аналитике или пониманию будущего, чтобы спрогнозировать будущее и действительно руководствоваться данными. С помощью прогнозных моделей компании могут использовать закономерности и прогнозы, чтобы получить следующие практические шаги, используя свои данные;
- анализ смешанных типов данных: традиционные платформы бизнес-аналитики в основном ориентированы на структурированные данные, но сегодня пользователям требуется возможность просмотра и анализа полуструктурированных, неструктурированных данных и сторонних данных. В последние годы огромное количество производимой информации увеличилось, отчасти из-за новых технологий интеллектуального анализа данных, Интернета вещей (IoT), распространения датчиков данных и средств автоматизированного сбора данных. Теперь опытным пользователям бизнес-аналитики и специалистам по обработке данных необходим доступ к неиспользованным данным в разных форматах, чтобы

смешивать типы данных и создавать собственные алгоритмы, в которых по запросу доступны аналитические данные для принятия точных и быстрых решений.

Так, в работе [7] обращается внимание на необходимость разработки инструментов, сочетающие функциональную мощь ВІ-систем (а лучше – превосходящие ее) со способностью к обработке огромных объемов информации.

В качестве одного из способов получить такой инструмент предлагается создание логической витрины данных.

Структурная схема ВІ-системы на основе логической витрины данных представлена на рисунке 7.



Рисунок 7 - Структурная схема ВІ-системы на основе логической витрины данных

В предлагаемом решении витрина данных не только компонует и связывает информацию из различных источников, но и делает логические выводы на ней в соответствии с заданными правилами.

Автоматизация получения логических выводов – одно из главных практических преимуществ семантики.

В работе [20] отмечается, что одним из самых популярных методов построения аналитических платформ для больших данных предполагает использование концепции озер данных.

Озеро данных (Data Lake) – это централизованное хранилище, которое позволяет хранить структурированные и неструктурированные данные в любом масштабе. Данные можно хранить в исходном виде без предварительного структурирования и запускать различные типы аналитики – от панелей мониторинга и визуализаций до обработки больших данных, аналитики в реальном времени и машинного обучения для принятия более эффективных решений [17].

Следует учесть, что озера данных сами по себе мало выполняют автоматические операции очистки и передачи данных, позволяя принимать данные более эффективно, но они перекладывают основную ответственность за подготовку и анализ данных на бизнес-пользователей.

Как показывает практика, подход на основе озера данных может предложить недорогое решение за счет использования распределенной файловой системы (например, HDFS) для эффективного хранения различных типов данных и анализа их в исходной структуре.

Вместе с тем озеро данных требует дополнительные ресурсы для реализации и высокого уровня экспертизы – по-настоящему извлечь из них пользу могут только высококвалифицированные аналитики.

Для решения этой проблемы используются витрины данных, представляющие собой среды хранения данных узкоспециализированной информации, ориентированные на запросы сотрудников определенного

отдела, вектор работы организации.

Авторами представлено исследование форматов хранения больших данных на платформе Apache Hadoop, используемых для создания витрин данных.

На рисунке 8 представлена схема вычислительного эксперимента для исследования форматов хранения больших данных на платформе Apache Hadoop.

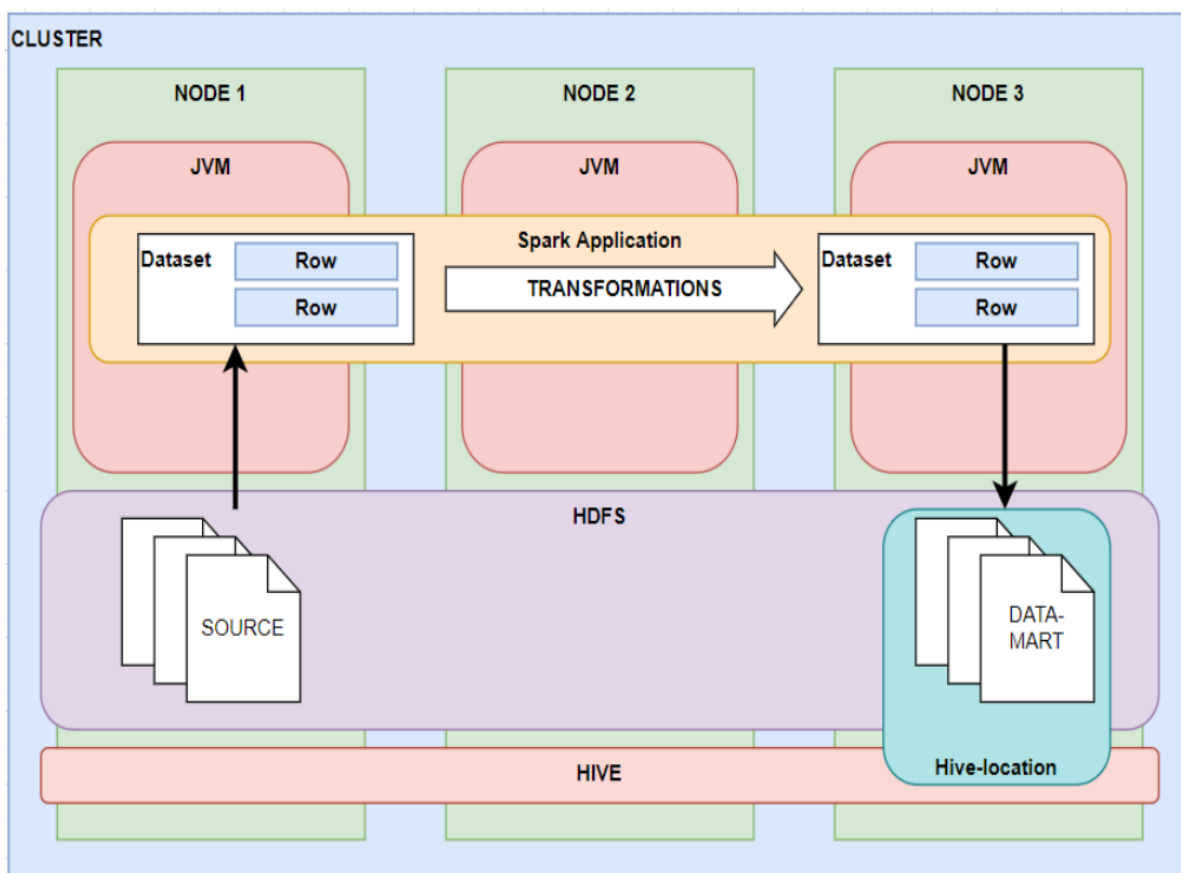


Рисунок 8 – Схема вычислительного эксперимента для исследования форматов хранения больших данных на платформе Apache Hadoop

В качестве основных критериев эффективности авторы выбраны такие характеристики, как время, необходимое для создания витрины данных, а также ее объем.

Эксперимент показал, что наиболее эффективным является формат Apache Parquet.

Вместе с тем необходимо констатировать недостаточность работ, посвященных проблеме разработки BI-систем на основе хранилищ и витрин больших данных.

Выводы по главе 1

В главе 1 магистерской диссертации представлен анализ современного состояния исследований в области разработки математического и программного обеспечения систем принятия управленческих решений на основе витрин данных.

Результаты проделанной работы позволили сделать следующие выводы:

- согласно концепции компании Gartner, одним из ключевых компонентов BI-систем является хранилище данных;
- витрина данных - это небольшая коллекция данных из хранилища данных, которая используется для бизнес-анализа в одном подразделении, в то время как хранилище данных представляет собой набор всех данных компании, которые используются для анализа в целом;
- в некоторых работах авторы обращают внимание на необходимость разработки инструментов, сочетающих функциональную мощь BI-систем со способностью к обработке больших объемов данных. Для достижения данной цели предлагается использовать различные технологии, основанные на применении витрин данных.

Вместе с тем проведенный анализ позволил констатировать недостаточность работ, посвященных проблеме разработки BI на основе хранилищ и витрин больших данных, что подтверждает актуальность темы настоящего исследования.

Глава 2 Анализ методологий и технологий построения систем поддержки принятия решений на основе витрин больших данных

2.1 Методы построения витрин данных

Выделяют следующие типы витрин данных:

- независимые витрины данных;
- зависимые витрины данных и оперативное хранилище данных;
- логические витрины данных.

Рассмотрим методы построения каждого из вышеперечисленных типов витрин данных.

2.1.1 Методы построения зависимых витрин данных

Зависимая витрина данных получает данные из централизованного хранилища данных. Это улучшает централизацию данных. Кроме того, если необходимо разработать одну или несколько физических витрин данных, можно настроить их как зависимые [21].

Чтобы сформировать хранилище данных, определенный набор данных агрегируется из хранилища данных, реструктурируется, а затем загружается в витрину данных, где его можно запросить.

Зависимая витрина данных в основном представляет собой комбинацию корпоративного (централизованного) хранилища данных (ЦХД) и согласованных данных.

Централизованное интегрированное хранилище данных предоставляет конечным пользователям информацию для поддержки принятия решений.

Зависимая витрина данных и архитектура хранилища операционных данных называется подходом «концентратор и луч», в котором ЦХД выступает в качестве концентратора, а системы исходных данных и витрина данных являются входными и выходными периферийными устройствами.

Концепция централизованного, или корпоративного, хранилища данных основана на подходе Б. Инмона, согласно которому хранилище данных

является централизованным хранилищем всех корпоративных данных.

При таком подходе организация сначала создает единое централизованное хранилище корпоративных данных. Затем создаются зависимые витрины размерных данных на основе модели хранилища.

Это нисходящий подход к построению хранилища данных (рисунок 9).



Рисунок 9 – Архитектура ЦХД

На основе корпоративного хранилища данных (КХД) построена архитектура «слоенного пирога» или LSA (Layered Scalable Architecture) [15].

По сути LSA реализует логическое деление структур с данными на несколько функциональных уровней.

Данные копируются с уровня на уровень и трансформируются при этом, чтобы в итоге предстать в виде согласованной информации, пригодной для анализа.

LSA-архитектура хранилища данных представлена на рисунке 10.

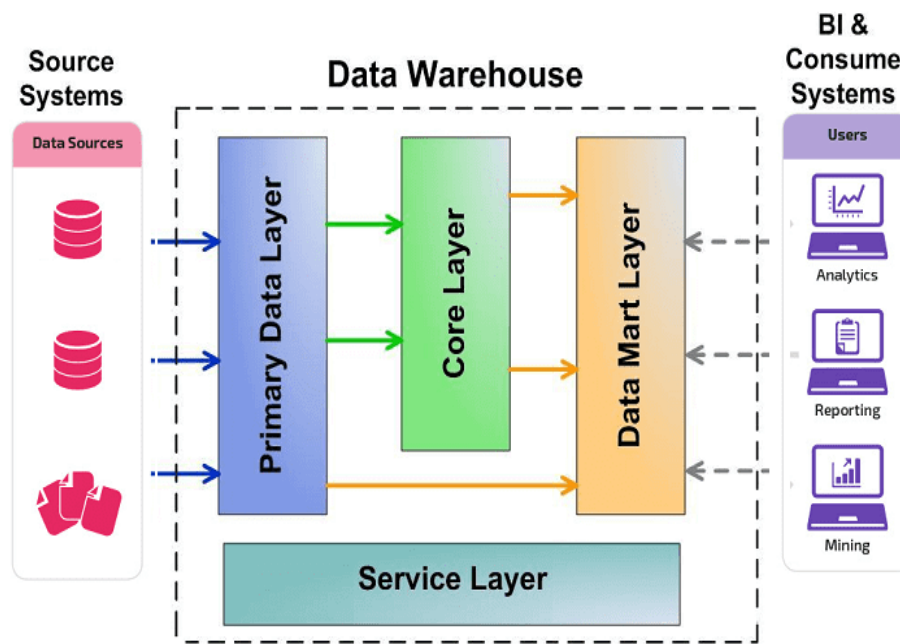


Рисунок 10 - LSA-архитектура хранилища данных

«Классически LSA реализуется в виде следующих уровней:

- операционный слой первичных данных (Primary Data Layer или стейджинг), на котором выполняется загрузка информации из систем-источников в исходном качестве и сохранением полной истории изменений. Здесь происходит абстрагирование следующих слоев хранилища от физического устройства источников данных, способов их сбора и методов выделения изменений;
- ядро хранилища (Core Data Layer) – центральный компонент, который выполняет консолидацию данных из разных источников, приводя их к единым структурам и ключам. Именно здесь происходит основная работа с качеством данных и общие трансформации, чтобы абстрагировать потребителей от особенностей логического устройства источников данных и необходимости их взаимного сопоставления. Так решается задача обеспечения целостности и качества данных» [15];
- аналитические витрины (Data Mart Layer), где данные преобразуются к структурам, удобным для анализа и использования в BI-дэшбордах

или других системах-потребителях. Когда витрины берут данные из ядра, они называются регулярными. «Если же для быстрого решения локальных задач не нужна консолидация данных, витрина может брать первичные данные из операционного слоя и называется соответственно операционной. Также бывают вторичные витрины, которые используются для представления результатов сложных расчетов и нетипичных трансформаций. Таким образом, витрины обеспечивают разные представления единых данных под конкретную бизнес-специфику;

- сервисный слой (Service Layer) обеспечивает управление всеми вышеописанными уровнями. Он не содержит бизнес-данных, но оперирует метаданными и другими структурами для работы с качеством данных, позволяя выполнять сквозной аудит данных (data lineage), использовать общие подходы к выделению дельты изменений и управления загрузкой. Также здесь доступны средства мониторинга и диагностики ошибок, что ускоряет решение проблем» [15].

В LSA-архитектуре реализовано озеро данных VI-системы для анализа больших данных в Тинькофф-банке (рисунок 11).

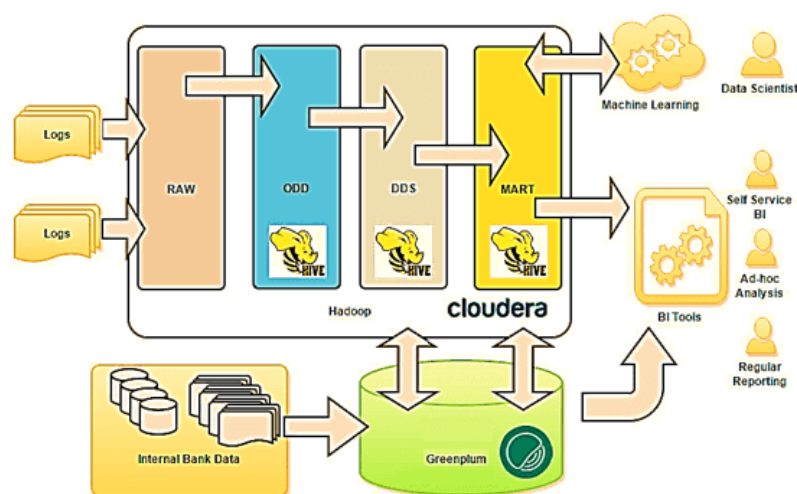


Рисунок 11 - LSA-архитектура корпоративного Data Lake в Тинькофф-банке

Следует отметить, что данная технология широко применяется в решениях корпорации SAP [14].

Зависимые витрины данных обеспечивают все преимущества витрин данных, а также позволяют осуществлять физический контроль данных по мере их извлечения из хранилища данных.

Поскольку зависимые витрины используют хранилище данных в качестве основы, они обычно считаются лучшим решением, чем независимые витрины, но они занимают больше времени на обработку запросов и дороже в реализации.

2.1.2 Методы построения независимых витрин данных

Независимая витрина данных – это автономная система, созданная без использования хранилища данных, которая фокусируется на одной предметной области или бизнес-функции.

Данные извлекаются из внутренних или внешних источников данных, обрабатываются, затем загружаются в репозиторий витрины данных, где хранятся до тех пор, пока они не понадобятся для бизнес-аналитики (рисунок 12).

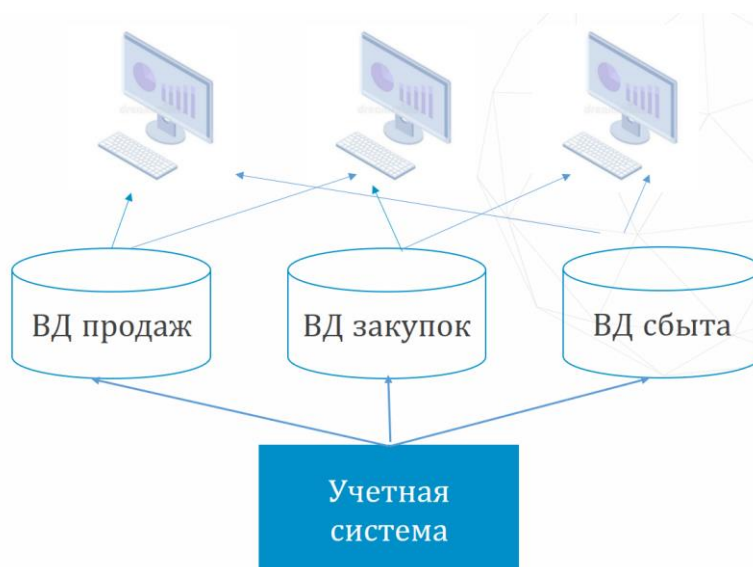


Рисунок 12 – Архитектура независимых витрин данных

Концепция независимых витрин данных принадлежит Р. Кимбаллу. Она опирается на восходящий подход к построению хранилищ данных.

При восходящем подходе сначала создаются витрины данных, чтобы предоставить отчеты и аналитические возможности для конкретных бизнес-процессов.

Затем эти витрины данных можно интегрировать для создания комплексного хранилища данных.

В результате создается многомерное хранилище данных, начинающееся с критически важных витрин данных, которые быстро настраиваются для обслуживания аналитических потребностей отделов или направлений бизнеса.

При таком подходе хранилище данных представляет собой объединение витрин данных, однако единого источника достоверной информации не существует, поскольку данные не интегрируются перед составлением отчетов.

Независимые витрины данных обычно самые простые и быстрые в реализации, и их окупаемость может быть почти мгновенной. Некоторые корпорации начинают с нескольких витрин данных, прежде чем принять решение о создании настоящего хранилища данных.

Однако у этого подхода есть несколько проблем:

- хотя независимые витрины данных имеют очевидную ценность, они не являются настоящим решением для всего предприятия и со временем могут стать очень дорогостоящими по мере увеличения их объемов;
- независимые витрины данных не обеспечивают исторической глубины настоящего хранилища данных.

Поскольку витрины данных предназначены для обработки определенных типов запросов от определенного типа пользователей, они часто не подходят для анализов типа «что, если», как, например, хранилища данных.

Независимые витрины данных используются, например, в компаниях, имеющих разветвленную филиальную сеть.

2.1.3 Методы построения логических витрин данных

Логическое хранилище данных (Logical data warehouse, LDW) - это архитектура управления данными, в которой архитектурный уровень располагается поверх традиционного хранилища данных, обеспечивая доступ к множеству разнообразных источников данных, представляя пользователям один логический источник данных.

По сути, это аналитическая архитектура данных, которая оптимизирует как традиционные источники данных (базы данных, корпоративные хранилища данных, озера данных и т. д.), так и другие источники данных (приложения, файлы больших данных, веб-службы и облака) для удовлетворения любых аналитических задач.

Этот термин был введен в обращение в 2009 году и продолжает набирать обороты на рынке, поскольку объемы и сложность данных становятся растущей проблемой для многих компаний [28].

Логическая витрина данных (Logical data mart, LDM)- это форма зависимой витрины данных, которая создается виртуально из физического хранилища данных.

Данные представляются пользователям витрины с использованием ряда представлений SQL, которые создают впечатление, что в основе данных, доступных для анализа, лежит физическая витрина данных.

Иными словами, логические витрины данных - это не физически разделенные базы данных, а различные реляционные представления хранилища данных. Данные перемещаются в хранилище данных, а не в отдельную промежуточную область.

Новую витрину данных можно создать быстро.

Логические витрины данных всегда в актуальном состоянии.

По мнению специалистов компании Teradata, «возможно, идеальный подход к внедрению концепции витрины данных в хранилище данных - это создание одной или нескольких зависимых логических или виртуальных

витрин данных» [38].

Архитектура логической витрины данных в концепции компании Teradata показана на рисунке 13.

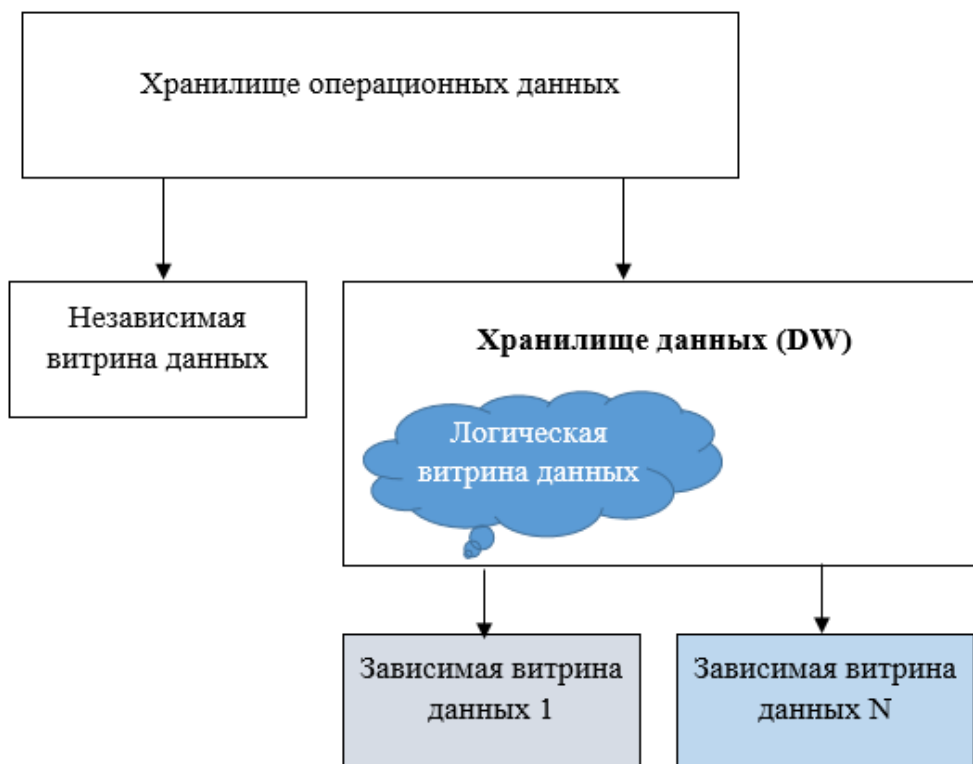


Рисунок 13 – Архитектура логической витрины данных

Используя систему тщательно сконструированных представлений о детализированных данных хранилища, можно создать несколько виртуальных витрин данных для конкретных пользователей или отделов, которые будут предоставлять такую же тщательно подобранную информацию, как и физические витрины данных, без необходимости в больших загрузках данных, очистки и другие необходимых преобразований.

Логические витрины данных преодолевают большинство ограничений независимых витрин данных. У данных нет исторического предела, и запросы типа «что, если» вполне осуществимы.

Главный недостаток логических витрин данных - отсутствие

физического контроля над данными. Поскольку данные в хранилище не агрегированы и не распределены по размерам, производительность по сравнению с логическим магазином обычно не так хороша, как по сравнению с независимым магазином. Однако использование параллелизма в логической витрине может преодолеть некоторые ограничения непреобразованных данных.

Следует также отметить, что логические витрины должны использоваться в хорошо спроектированном хранилище данных.

Для сравнения вышеописанных методов на предмет применения при разработке витрин больших данных составим таблицу 1.

Критерии оценивания:

- 0 – полное несоответствие требованиям;
- 1 – значительное несоответствие требованиям;
- 2 – незначительное несоответствие требованиям;
- 3 – полное соответствие требованиям.

Таблица 1 – Сравнение методов построения витрин данных

Характеристики	Зависимые витрины данных	Независимые витрины данных	Логические витрины данных
Затраты на реализацию для больших объемов данных	2	3	3
Избыточность данных	2	4	3
Большие объемы исторических данных	5	1	4
Скорость обработки запросов	2	4	3
Итого	11	12	13

Таким образом, при разработке витрин больших данных представляется целесообразным использование методов построения логических витрин данных.

2.2 Технологии построения BI-систем для анализа больших объемов данных

Как показал анализ известных решений, при построении BI-систем для анализа больших данных для самых разных задач, обычно используется технология ROLAP.

Реляционная онлайн-аналитическая обработка (Relational OLAP, ROLAP) – это форма OLAP, которая выполняет динамический многомерный анализ данных, хранящихся в реляционной базе данных, а не в многомерной базе данных [36].

Принцип работы такой модели достаточно прост: при активизации аналитической функции формируется SQL-запрос, который уходит на тот или иной бэкэнд. При BI-система может быть интегрирована с любой СУБД, поддерживающей язык SQL (например, реляционные СУБД MySQL, PostgreSQL и другие) [22].

Модель интеграции BI-системы с реляционной СУБД представлена на рисунке 14.

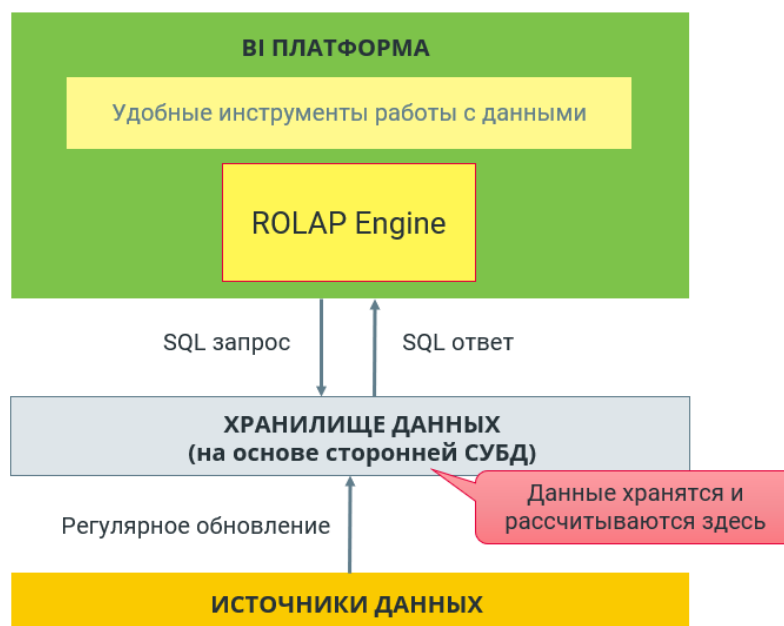


Рисунок 14 – Модель интеграции BI-системы с реляционной СУБД

Преимущество данной технологии заключается в том, что для работы ВІ-системы нет ограничений по объему данных.

Недостаток – относительно невысокая скорость выполнения запросов, что негативно сказывается на производительности ВІ-системы.

Для решения данной проблемы используются различные подходы, в том числе применение специализированных колоночных СУБД, например, ClickHouse или Vertica.

Однако колоночные СУБД используют собственный усеченный диалект SQL, что существенно снижает функциональные возможности такого решения.

Более перспективным представляется решение, основанное на применении СУБД, в которых реализована технология In-Memory.

«Технология In-Memory – это технология выполнения компьютерных вычислений полностью в памяти компьютера, например, в оперативном запоминающем устройстве (ОЗУ).

Этот термин обычно подразумевает крупномасштабные сложные вычисления, которые требуют специального системного программного обеспечения для выполнения вычислений на компьютерах, работающих вместе в кластере.

В качестве кластера компьютеры объединяют свою оперативную память, поэтому расчет по существу выполняется между компьютерами и использует общее пространство ОЗУ всех компьютеров вместе.

Обработка в памяти устраняет все медленные обращения к данным и полагается исключительно на данные, хранящиеся в ОЗУ» [26].

Общая производительность обработки не снижается из-за задержки, обычно наблюдаемой при доступе к жестким дискам или твердотельным накопителям.

Обработка в памяти часто выполняется с помощью технологии, известной как сетки данных в памяти.

Такие СУБД имеют преимущество в скорости по сравнению с традиционными дисковыми СУБД.

Многие существующие реляционные СУБД поддерживают механизм In-Memory, например, поздние версии MS SQL Server, Oracle Database и другие.

Модель BI-системы, использующей движок In-Memory OLAP, представлена на рисунке 15.

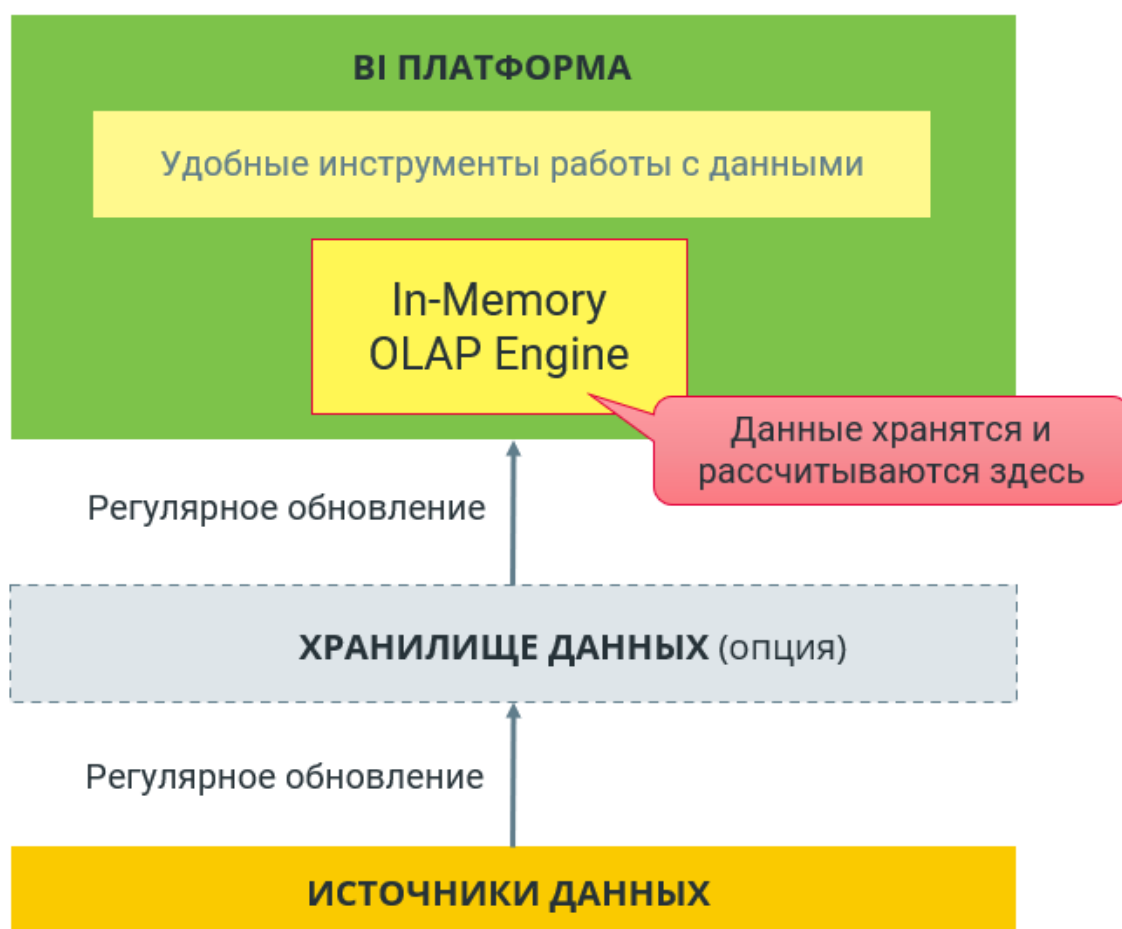


Рисунок 15 – Модель BI-системы, использующей движок In-Memory OLAP

По результатам представленного анализа выбираем в качестве технологии построения BI-системы для анализа больших данных технологию, основанную на применении реляционной СУБД, поддерживающей механизм In-Memory.

2.3 Технология проектирования программного обеспечения ВІ-системы

Между жизненным циклом транзакционных систем и жизненным циклом ВІ-систем есть некоторые существенные различия, которые зависят от характеристик СППР, но для разработки используются одни и те же традиционные методы и этапы [19]:

Диаграмма деятельности процесса проектирования программного обеспечения ВІ-системы показана на рисунке 16.

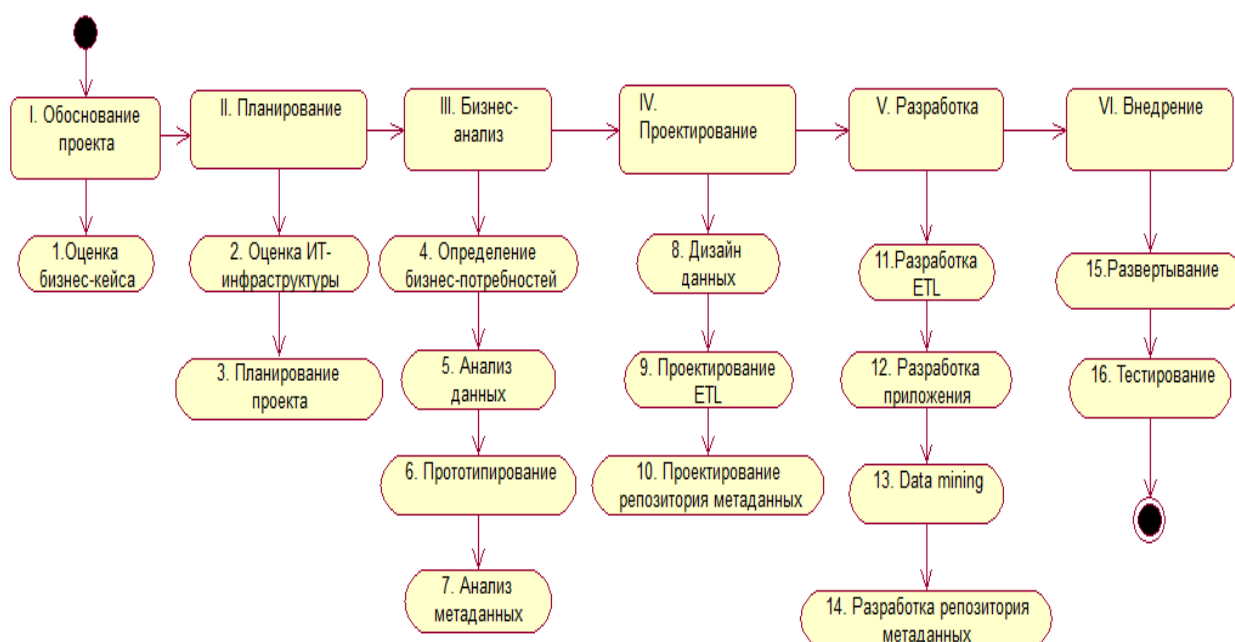


Рисунок 16 - Диаграмма деятельности процесса проектирования программного обеспечения ВІ-системы

Процесс проектирования ВІ-системы состоит из 6 стадий, включающих 16 этапов.

Рассмотрим стадии данного процесса.

Стадия I: Обоснование. Данная стадия состоит из нижеследующих этапов.

Этап 1: Оценка бизнес-кейса.

Определяются бизнес-потребности и возможности организации. Затем команда разработчиков предлагает первоначальное решение, оправданное с учетом затрат и выгод. Составляется предварительный отчет.

Стадия II: Планирование. Данная стадия состоит из нижеследующих этапов.

Этап 2: Оценка инфраструктуры предприятия.

На этом этапе оцениваются возможности организации по поддержанию и выполнению проекта ВІ-системы с точки зрения ИТ-инфраструктуры, компонентов, устройств, сети, а также будущих потребностей в оборудовании.

На этом этапе строится ИТ-инфраструктура организации.

Этап 3: Планирование проекта.

Данный этап включает динамическое планирование проекта, которое приводит к быстрым изменениям в технологиях, организационных и бизнес-потребностях, людских ресурсах и команде разработчиков. План проекта подробный, прогрессивный, на каждом этапе и шаге есть контрольные точки, тестовые документы и отчеты.

Стадия III: Бизнес-анализ. Данная стадия состоит из нижеследующих этапов.

Этап 4: Определение бизнес-потребностей.

На данном этапе организуются собеседования и встречи с руководителями и менеджерами, а также определяются бизнес-потребности и функциональные требования.

Этап 5: Анализ данных.

Этот этап включает в себя определение и проектирование источников данных, разработку подробных диаграмм «сущность-связь» (ER-диаграмм) с атрибутами и ссылками между данными.

Разрабатывается логическая модель данных ВІ-системы.

Этап 6. Создание прототипа приложения.

Первоначальный прототип создается и тестируется для соответствия потребностям бизнеса. После тестирования результаты оцениваются с учетом

положительных и отрицательных аспектов.

Этап 7: Анализ метаданных.

На данном этапе создаются метаданные, источники данных отображаются в структуре метаданных. CASE-средства используются для процесса разработки и отображения.

Стадия IV: Проектирование системы. Данная стадия состоит из нижеследующих этапов.

Этап 8: Дизайн данных.

На этом этапе детализируется и уточняется логическая модель данных, а также проектируется физическая модель данных. Модель данных для обработки и хранения выбирается из следующих вариантов: реляционная, объектно-ориентированная и многомерная модель.

Этап 9: Проектирование процесса ETL.

Этот этап является наиболее сложным во всем цикле и зависит от качества источников данных.

«Рекомендуется, чтобы процесс был построен в единой среде, объединяющей все модули организации, а не отдельно для каждого отдела.

Правило должно быть таким: используйте один скоординированный процесс ETL.

Этап 10: Проектирование репозитория метаданных хранилища или витрины данных.

Если используется predetermined решение для репозитория метаданных, то на этом этапе оно настраивается в соответствии с требованиями проекта, в противном случае репозиторий метаданных разрабатывается в терминах логической модели метаданных в зависимости от модели данных: реляционной, объектно-ориентированный или многомерный» [19].

Также является важным и сложным этапом.

Стадия V: Разработка. Данная стадия состоит из нижеследующих этапов.

Этап 11: Разработка ETL.

Для построения ETL используются инструменты, процедуры и операторы фильтрации. Фильтрация и преобразование данных зависит от качества источников данных. Эти источники разные: файлы, базы данных, электронная почта, Интернет, нетрадиционные источники.

Этап 12: Разработка приложения.

После проверки прототипа создание окончательного приложения может оказаться простым процессом. Перестраиваются шаблоны процедур и интерфейсы. Предоставляются права и привилегии пользователям.

Этап 13: Data mining (интеллектуальный анализ данных).

Исполнительные системы должны реализовать возможности интеллектуального анализа данных, чтобы добиться успеха и выполнить требования менеджера.

Этот этап включает проверку алгоритмов, методов интеллектуального анализа данных, таких как кластеризация, методы прогнозирования и организации.

Этап 14: Разработка репозитория метаданных.

Если репозиторий метаданных должен быть создан, то разрабатываются словарь метаданных и доступ к данным через интерфейсы.

Стадия VI: Внедрение системы. Данная стадия состоит из нижеследующих этапов.

Этап 15: Развертывание системы.

Это процесс доставки, в ходе которого команда разработчиков организует тренинги для менеджеров, готовит окончательную документацию и техническую поддержку, завершает процесс загрузки данных и настройку приложения.

Этап 16: Тестирование системы.

После внедрения системы делаются предварительные выводы и оцениваются затраты. Команда разработчиков составляет окончательный отчет, в котором описываются характеристики системы, а также некоторые

части, которые необходимо улучшить или перестроить.

2.4 Методологии проектирования программного обеспечения ВІ-системы

Как показал анализ, при проектировании ВІ-систем широко применяется методология объектно-ориентированного анализа и проектирования, основанная на принципах объектно-ориентированного подхода (ООП) и языке визуального моделирования UML.

Данная методология используется на стадии логического проектирования ВІ-систем.

Логическое проектирование определяет функции и особенности информационной системы и отношения между ее компонентами.

Логическое проектирование включает выходные данные, которые должны быть произведены информационной системой, входные данные, необходимые для системы, и процесс, который должен быть произведен системой, независимо от того, как задачи будут выполняться физически.

Логическое проектирование определяет то, что должно происходить в информационной системе, а не то, как это должно быть выполнено.

Другими словами, логические проекты не касаются реальных методов реализации. При подготовке логического проекта ИС системный аналитик определяет потребности пользователя на уровне детализации, который фактически определяет поток информации в систему и из нее и необходимые источники данных. Результатом логического проектирования является логическая модель и программная архитектура ИС.

Логическое моделирование – это обязательный процесс перед написанием программного кода.

Логическое моделирование является неотъемлемой частью крупных программных проектов, а также полезно для средних и даже небольших проектов.

Понятие логической модели ИС тесно связано с ее логической архитектурой. Следует добавить, что одной из основных задач логического моделирования является разработка логической модели данных ИС.

В процессе логического моделирования широко применяются паттерны (шаблоны) проектирования.

Одним из таких шаблонов является паттерн «Модель-Представление-Контроллер» (Model-View-Controller, MVC) [30].

Архитектурный шаблон MVC в основном применяется для интерактивных программных систем.

Компонент Model содержит основные функции и данные, компонент View предоставляет информацию пользователю.

Компонент контроллера обрабатывает ввод данных пользователем с помощью проверки.

View и Controller вместе составляют пользовательский интерфейс.

Механизм распространения изменений обеспечивает согласованность между пользовательским интерфейсом и моделью.

Архитектура паттерна MVC представлена на рисунке 17.

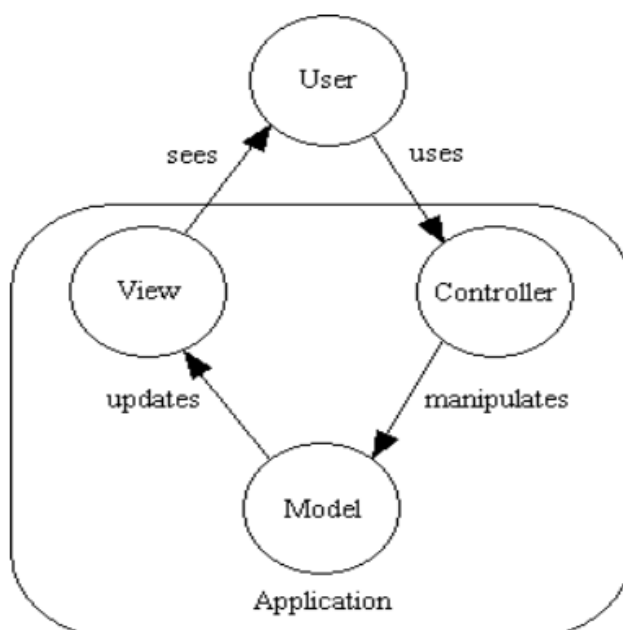


Рисунок 17 – Архитектура паттерна MVC

По мнению авторов работы [27], применение MVC при разработке программного обеспечения ВІ-системы обеспечит следующие преимущества:

- «эффективность. Системы бизнес-аналитики, использующие MVC, очень популярны и эффективны с точки зрения хранения разнородных данных в хранилище данных и синхронизации представлений;
- простота. Разделение задачи на независимые компоненты, а именно на модель, представление и контроллер, снижает сложность архитектуры ВІ-системы;
- масштабируемость. Бизнес-приложения становятся хорошо масштабируемыми, так что новые компоненты бизнес-приложений могут быть добавлены или удалены без ущерба для функциональности ВІ-системы;
- гетерогенность. ВІ-системы неоднородны, поэтому могут работать в разных вычислительных средах;
- поддержка. ВІ-системы поддерживают синхронизированные представления и согласованность данных, когда они реализованы с использованием архитектурного паттерна MVC» [27].

Поскольку архитектура MVC отделяет модель от представления, для одной модели можно создать несколько представлений.

При реализации архитектуры ВІ для любой организации требуется наличие нескольких представлений для представления данных в желаемой форме (например, в виде графиков, круговых диаграмм и т.д.).

Разным типам заинтересованных сторон (менеджерам, аналитикам, разработчикам) могут потребоваться разные представления данных в разных форматах.

Архитектура «Модель-Представление-Контроллер» помогает создавать несколько представлений, сохраняя неизменными модель и контроллер.

2.2.2 Методология объектно-структурного моделирования

Если рассматривать системы ВІ как разновидность систем управления процессом обучения, то при выборе методологического подхода к их построению следует использовать методологии, основанные на интеграции различных методологических подходов.

Одной из таких методологий является методология объектно-структурного подхода.

В работе [29] предлагается методология объектно-структурного подхода к построению информационных систем управленческого учета (ИСУУ), организованных по модели Workflow, которая состоит из нижеследующих стадий.

«I. Объектно-структурное моделирование ИСУУ.

Объектно-структурная модель (ОСМ) системы N-этапной обработки учетной информации имеет вид ориентированного графа $O(W,S,D)$, изображенного на рисунок 18.

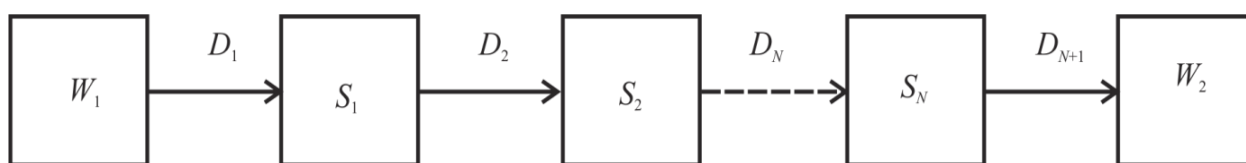


Рисунок 18 – Объектно-структурная модель системы N-этапной обработки учетной информации

На рисунке:

$W = \{W_1, W_2\}$ – узлы, обозначающие объекты класса “Склад”.

$S = \{S_1, S_2, \dots, S_N\}$ – множество узлов, обозначающие объекты класса “Этап (Склад+Агрегат+Контролер)”.

$D = \{D_1, D_2, \dots, D_{N+1}\}$ – множество дуг, нагруженных элементами информационного потока.

Физическим представлением ОСМ является бизнес-транзакция.

II. Формализация элементов ОСМ.

Заключается в представлении объектно-структурной модели ИСУУ в виде системы взаимодействующих автоматов, управляющих статусом обрабатываемого элемента материального (информационного) потока, который описывается как конечный автомат (КА) в соответствии с его жизненным циклом, представленным в таблице 2» [29].

Таблица 2 – Жизненный цикл продукта в многоэтапном производстве

Статус	Описание
1	Сырье
2	Полуфабрикат
3	Готовая продукция

III. Разработка UML паттернов проектирования ИСУУ.

Паттерны проектирования строятся в нотации UML на основе объектных моделей конечного автомата, сгруппированных по суперклассам - классам технологической онтологии конкретной предметной области.

Примеры паттернов проектирования ИСУУ представлены на рисунке 19.

WAREHOUSE	AGGREGATE	CONTROLLER
ID itemBalance	ID itemStatus	ID itemStatusControlResult
+receiptItem() +expenseItem()	+changeItemStatus()	+controlItemStatus()

Рисунок 19 – Паттерны проектирования ИСУУ

Преимуществами предлагаемой методологии проектирования ИСУУ

являются:

- универсальность объектно-структурных моделей ИСУУ;
- простота адаптации объектно-структурных моделей ИСУУ к специфике управленческого учета в конкретной организации;
- простота интеграции ИСУУ в корпоративную информационную систему (КИС) предприятия.

«Для хранения агрегированных данных и результатов анализа информации (например, страховых резервов, убыточности клиентов или финансовых результатов агентов) создаются таблицы фактов, которые связывается по схеме «звезда» или «снежинка» с таблицами-классификаторами или справочниками базы данных информационной системы.

Пример диаграммы классов типовой подсистемы анализа страховой информации приведен на рисунке 20.

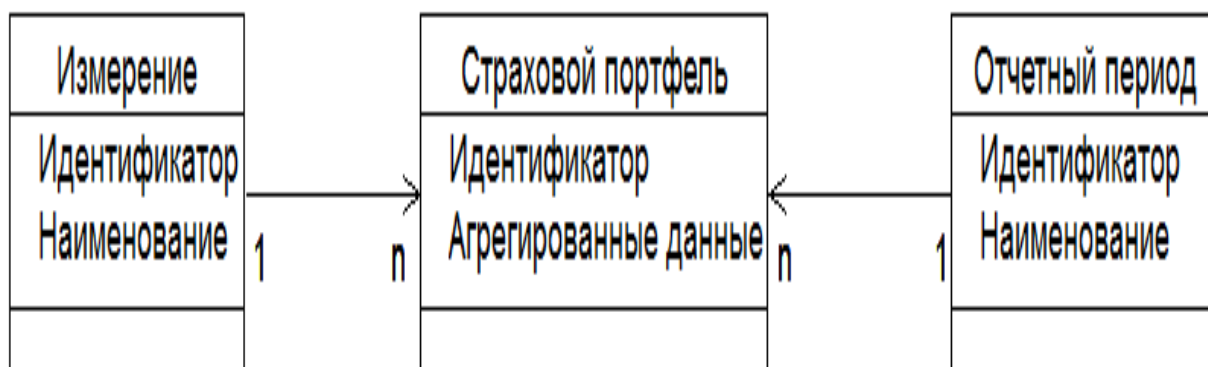


Рисунок 20 - Диаграмма классов типовой подсистемы анализа страховой информации

Ключевым объектом представленной диаграммы является объект класса «Страховой портфель», представляющий собой репозиторий данных.

На основе соответствующего шаблона проектирования разрабатываются программные объекты - информационные эксперты, которые предназначены для аккумуляции и расчета аналитических

данных в системах поддержки принятия решений» [8].

Применение данного шаблона при проектировании витрин данных обеспечит существенное снижение затрат.

Выводы по главе 2

В главе 2 магистерской диссертации представлен анализ методологий и технологий построения систем поддержки принятия решений на основе витрин больших данных.

Результаты проделанной работы позволили сделать следующие выводы:

- как показал анализ, при разработке витрин больших данных представляется целесообразным использование методов построения логических витрин данных;
- наиболее перспективной для построения BI-систем для анализа больших данных представляется технология, основанная на применении реляционной СУБД, поддерживающей механизм In-Memory OLAP;
- технология проектирования BI-системы состоит из 6 стадий, включающих 16 этапов. Наиболее трудоемкими считаются этапы проектирования ETL и репозитория метаданных хранилища или витрины данных.

Применение шаблона «Репозиторий данных» объектно-структурного подхода при проектировании витрин данных обеспечит существенное снижение затрат.

Глава 3 Математическое обеспечение системы поддержки принятия решения на основе витрин больших данных

3.1 Логическое моделирование системы поддержки принятия решения на основе витрин больших данных

Для разработки математического обеспечения СППР используем объектно-ориентированный и объектно-структурный подходы к проектированию информационно-аналитических систем.

Как было отмечено выше, при построении логической модели ИС используется методология объектно-ориентированного проектирования.

Следует констатировать, что при реализации проектов разработки математического и программного обеспечения на основе ООП наилучшие результаты достигнуты при применении методологии RUP и CASE-средства Rational Rose.

«Методология RUP (Rational Unified Process) – это гибкая методология разработки программного обеспечения.

В методологии RUP для успешного процесса разработки необходимы три составляющие: процесс, нотация и набор утилит. Процесс описывает, что мы делаем, в каком порядке и каким образом; нотация является средством общения; набор утилит помогает автоматизировать процесс и управлять им» [6].

RUP разделяет жизненный цикл проекта разработки программного обеспечения ИС на четыре фазы: начало, уточнение, построение, внедрение.

На каждом из этапов выполняются все шесть основных дисциплин разработки: бизнес-моделирование, требования, анализ и проектирование, реализация, тестирование и развертывание.

Основная цель RUP – создание высококачественного программного обеспечения с предсказуемым бюджетом и временными рамками.

Следует отметить, что для построения логической модели ИС необходимо разработать базовые диаграммы языка UML, отражающие различные аспекты исследуемой ИС.

Для определения функциональных требований применяется диаграмма вариантов использования UML [12].

Эта диаграмма позволяет описать сложные процессы относительно простыми средствами, а главное делает их понятными для всех заинтересованных лиц.

Диаграмма вариантов использования наглядно представляет взаимодействие между основными сервисами (бизнес-прецедентами), которые предоставляет исследуемый бизнес-процесс, и теми, кому эти сервисы предоставлены (бизнес-субъекты или акторы).

Как инструмент визуального моделирования и бизнес-анализа Rational Rose позволяет системному аналитику отслеживать бизнес-цели и сопоставлять их с системными требованиями, что существенно повышает эффективность процесса формирования последних.

На этапе управления требованиями RUP необходимо, чтобы все прецеденты и участники были определены, и было разработано большинство описаний прецедентов.

Акторами BI-системы являются ETL, Источник больших данных, Лицо, принимающее решение (ЛПР), Инструмент BI, OLAP-система.

Варианты использования (прецеденты) представлены в таблицах 3-6.

Таблица 3 - Описание прецедента: Извлечение, трансформация и загрузка данных

Прецедент: Извлечение, трансформация и загрузка данных
ID: 1
«Краткое описание: Извлечение, трансформация и загрузка данных в хранилище данных
Главный актер: ETL
Второстепенный актер: Источник больших данных
Предусловие: нет
Основной поток: ETL извлекает данные из Источника больших данных, преобразует их и загружает в хранилище данных» [12]

Продолжение таблицы 3

Прецедент: Извлечение, трансформация и загрузка данных
Постусловие: нет
Альтернативные потоки: нет

Таблица 4 – Описание прецедента: Создание логической витрины данных

Прецедент: Создание логической витрины данных
ID: 2
«Краткое описание: создание логической витрины данных
Главный актер: OLAP-система
Второстепенный актер: нет
Предусловие: нет
Основной поток: OLAP-система формирует логической витрины данных из хранилища данных
Постусловие: нет
Альтернативные потоки: нет» [12]

Таблица 5 - Описание прецедента: Анализ данных

Прецедент: Анализ данных
ID: 3
«Краткое описание: Выполнение анализа данных
Главный актер: Инструмент BI
Второстепенный актер: нет
Предусловие: создание логической витрины данных
Основной поток: Инструмент BI выполняет анализ данных
Постусловие: представление результатов анализа
Альтернативный поток: нет» [12]

Таблица 6 - Описание прецедента: Принятие управленческого решения

Прецедент: Принятие управленческого решения
ID: 4
«Краткое описание: принятие управленческого решения
Главный актер: ЛПР
Второстепенный актер: Инструмент BI
Предусловие: представление результатов анализа
Основной поток: ЛПР принимает управленческое решение на основе результатов анализа
Постусловие: нет
Альтернативный поток: нет» [12]

Диаграмма вариантов использования VI-системы на основе логической витрины данных показана на рисунке 21.

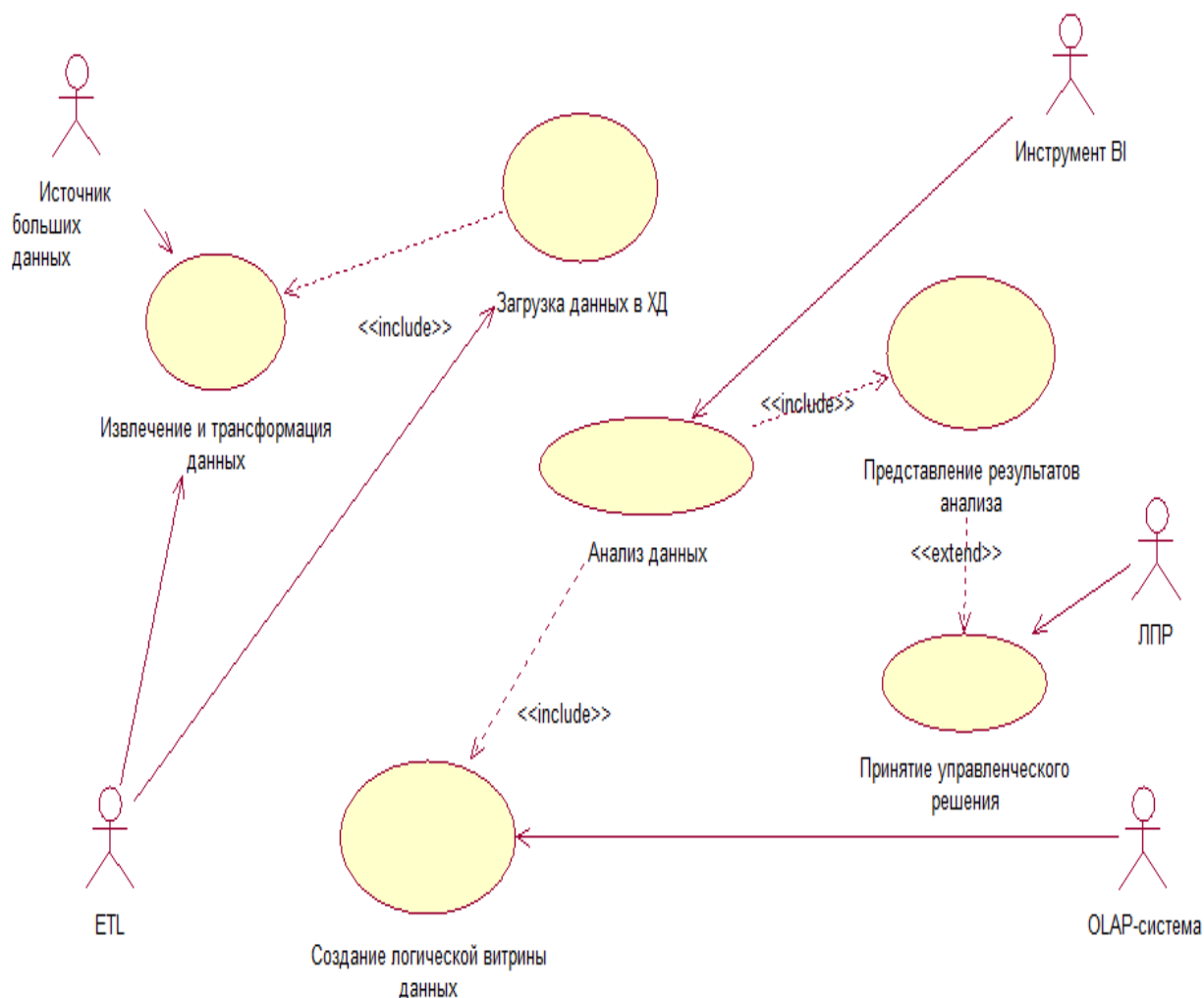


Рисунок 21 - Диаграмма вариантов использования VI-системы на основе логической витрины данных

Рассмотрим процесс разработки метамодели объектов VI-системы.

Метамодель - это модель, описывающая другую модель или транзитивное отношение между двумя моделями.

Для разработки метамodelей рекомендуется использовать подход MDA (Model Driven Architecture), предложенный группой OMG [33].

Выделяются три основные категории объектов VI-системы [37]:

- репозитории данных, представляющие элементы архитектуры хранилища данных;
- объекты данных, представляющие модель данных определенного репозитория;
- объекты-презентации, например, статический отчет или дашборд.

Для разработки метамодели объектов ВІ-системы используем диаграмму классов.

Диаграммы классов описывают внутреннюю структуру классов и отношения между классами. Сюда входят отношения классов, а также атрибуты и поведения, связанные с каждым классом. Диаграммы классов прекрасно иллюстрируют наследование и составные отношения.

На диаграмме классов UML представлены:

- классы в объектно-ориентированной системе;
- атрибуты и методы классов;
- связи между классами.

Не представлены на диаграмме классов UML:

- детали того, как классы взаимодействуют друг с другом
- алгоритмические детали, то есть, как конкретное поведение класса реализовано.

Диаграммы классов отлично подходят для:

- обнаружение связанных данных и атрибутов;
- получения быстрого представления о важных объектах в системе;
- выяснения достаточности или недостаточности классов;
- определения, являются ли отношения между объектами слишком сложными или простыми;
- выявление зависимостей между классами или объектами.

Вместе с тем диаграммы классов не позволяют решить такие вопросы, как обнаружение алгоритмического (не управляемого данными) поведения и понимание общего потока управления приложением.

Метамодель ВІ-системы на основе логической витрины данных

показана на рисунке 22.

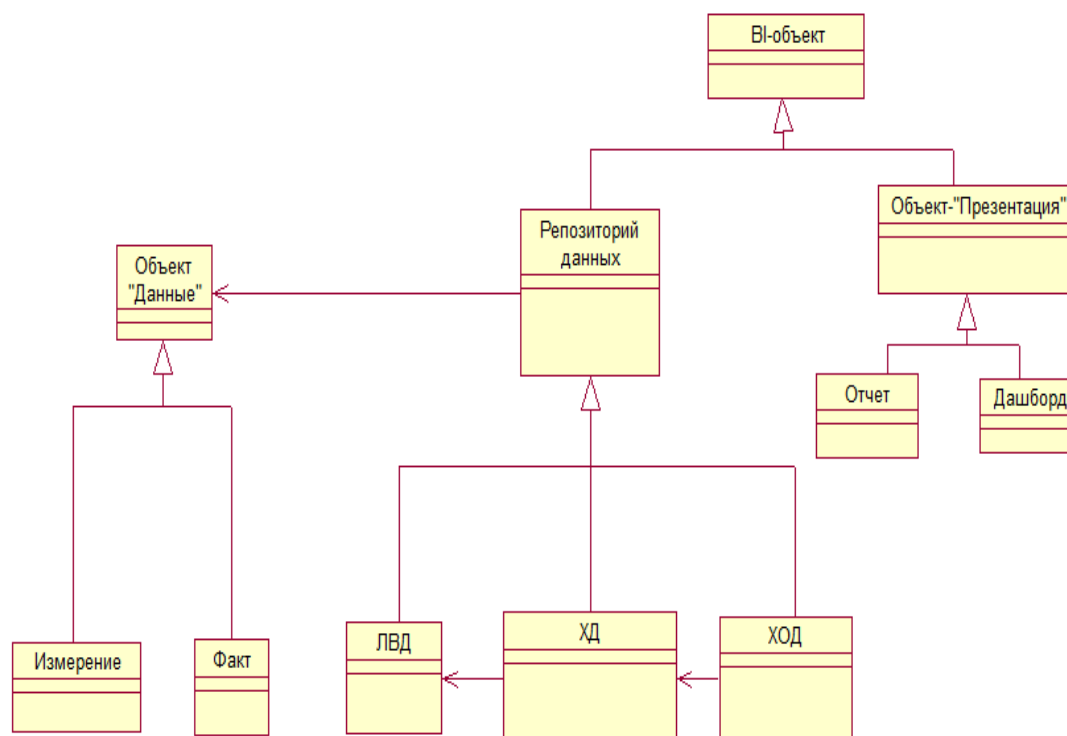


Рисунок 22 – Мета модель BI-системы на основе логической витрины данных

Здесь:

ХД – хранилище данных;

ХОД – хранилище операционных данных;

ЛВД- логическая витрина данных.

Как следует из диаграммы, объекты BI, выбранные для модели, зависят от целевой аудитории и уровня детализации модели.

В обзорной модели бизнес-процесса, подходящей для менеджеров, можно показать ХД или отдельные витрины данных в целом.

В более подробной метамодели для разработчиков подпроцессы можно описать как доступ к отдельным сущностям и фактам.

Кроме того, лица, принимающие решения, часто получают соответствующие данные в форме отчетов, например, отчета о продажах за прошедший финансовый год, который также может иметь отношение к

моделированию бизнес-процессов.

3.2 Разработка логической модели данных системы поддержки принятия решения на основе витрин больших данных

Моделирование данных анализирует объекты данных и определяет отношения между ними.

Оно генерирует теоретическое представление объектов данных - поставщиков или клиентов в базах данных, а также того, как хранить объекты в системе, определяя правила для отношений между таблицами [40].

Моделирование данных определяет правила преобразования данных и подготавливает данные для анализа больших данных.

ETL-процесс применяет эти правила и проверяет их на наличие аномалий. Это гарантирует, что данные готовы для анализа. Затем он загружает эти данные в хранилище данных.

В эпоху больших данных для точного анализа данных аналитики должны понимать взаимосвязь между частями информации и тем, как данные передаются в системе.

Следовательно, моделирование данных BI-системы имеет решающее значение для агрегирования данных между разнородными инструментами и платформами.

Рассмотрим процесс моделирования данных логической витрины данных.

Логическая витрина данных построена по модели ROLAP, схема «снежинка» [1].

Для разработки таблицы фактов использован шаблон «Репозиторий данных».

Репозиторий данных описывается с помощью кортежа:

$$SP = \langle GP, XP, YP, ZP, zp_0, vp, fp \rangle, \quad (1)$$

где GP – поток управления;

XP, YP – входной и выходной потоки строго структурированных и упорядоченных последовательностей значений атрибутов данных;
 ZP – множество состояний репозитория;
 zr_0 – начальное состояние репозитория;
 $vr \in VP$ – оператор переходов репозитория;
 $fr \in FP$ – оператор выходов репозитория.

«Внешние события, вызывающие изменение состояния репозитория ZP , инициируются движениями потоков XP (добавление данных) и YP (удаление данных).

Внутренние события репозитория инициируются управляющим потоком GP (например, изменение условий страхования или кредитования, прекращение действия договора и т.п.)» [8].

Пример логической модели данных виртуальной витрины данных кредитной организации, созданной с помощью представления данных (View) в хранилище больших данных, показан на рисунке 23.

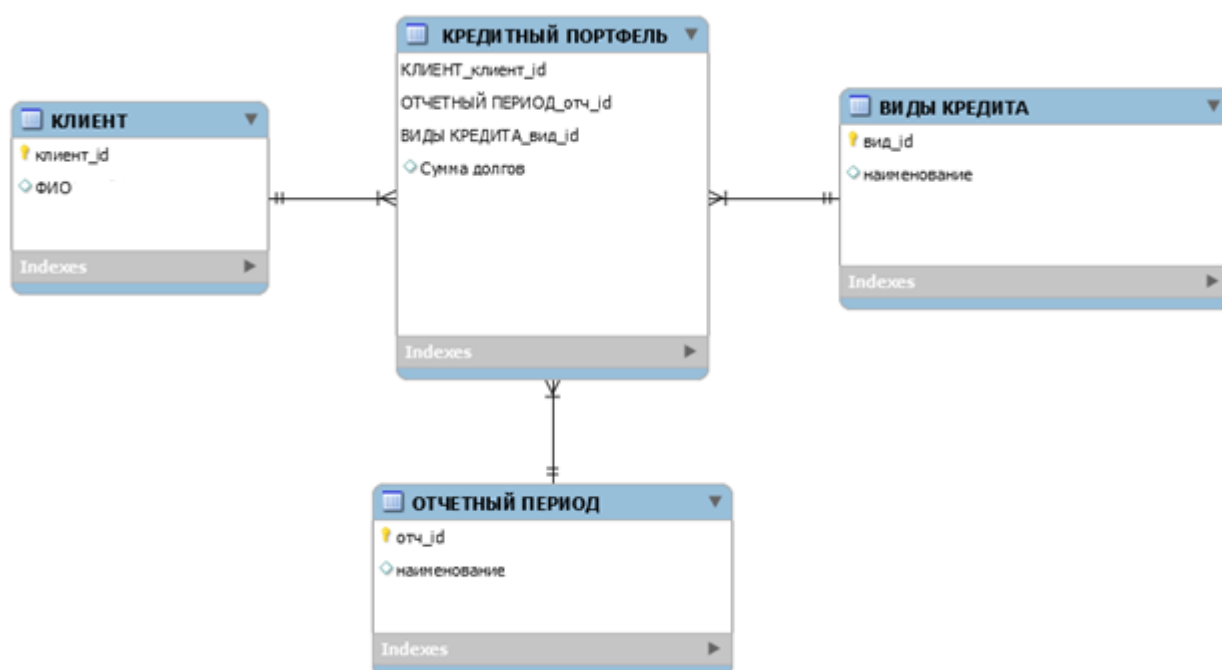


Рисунок 23 – Логическая модель данных логической витрины данных кредитной организации

Таблица фактов модели «Кредитный портфель» создана на основе вышеописанного шаблона «Репозиторий данных».

Выводы по главе 3

В главе 3 магистерской диссертации представлена разработка математического обеспечения системы поддержки принятия решений на основе витрин больших данных.

Результаты проделанной работы позволили сделать следующие выводы:

- для построения логической модели СППР необходимо разработать базовые диаграммы языка UML, отражающие различные аспекты системы;
- для определения функциональных требований применяется диаграмма вариантов использования UML;
- для разработки метамodelей рекомендуется использовать подход MDA (Model Driven Architecture), предложенный группой OMG;
- моделирование данных BI-системы имеет решающее значение для агрегирования данных между разнородными инструментами и платформами.

Логическая витрина данных построена по модели ROLAP, схема «снежинка». Для разработки таблицы фактов использован шаблон «Репозиторий данных».

Глава 4 Программное обеспечение системы поддержки принятия решения на основе витрин больших данных

«Для разработки программной архитектуры СППР используем диаграмму компонентов UML.

Диаграммы компонентов используются для визуализации организации и отношений между компонентами в системе. Эти диаграммы также используются для создания исполняемых систем.

Диаграммы компонентов также можно описать как статическое представление системы. Статическая реализация представляет собой организацию компонентов в определенный момент.

Диаграмма одного компонента не может представить всю систему, но набор диаграмм используется для представления целого» [6].

На рисунке 24 представлена программная архитектура СППР на основе логической (виртуальной) витрины данных.

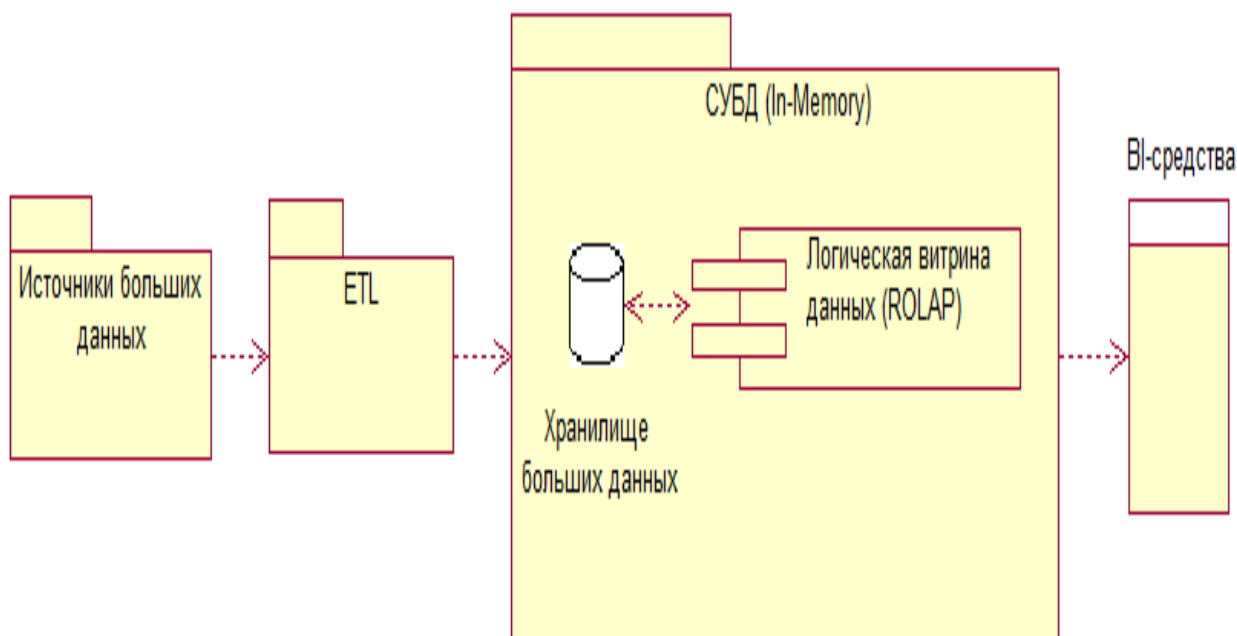


Рисунок 24 – Программная архитектура СППР на основе логической витрины данных

В настоящее время популярен подход к проектированию программного обеспечения BI-систем, основанный на применении готовых ИТ-решений.

«Основные преимущества данного подхода:

- меньшие затраты. Существует множество готовых BI-систем для анализа больших данных, наиболее популярными из которых сегодня считаются MS Power BI, Qlik и Tableau, а также их различные бесплатные и коммерческие аналоги в том числе комплексные DaaS-сервисы. Их можно использовать как локально, так и в облачном варианте по подписной модели;
- высокая производительность. Готовые решения предназначены для массового использования, они уже содержат наборы алгоритмов для обработки и анализа больших данных, которые наиболее востребованы на практике;
- простота понимания. Встроенные модули визуализации готовых BI- и DaaS-решений наглядно демонстрируют наиболее значимые для бизнеса показатели, такие как количество посещений, конверсия, уровень продаж, совокупные расходы и доходы, а также в разрезе отдельных статей. Интерфейс понятен не только ИТ-специалисту, но и руководителю и различным специалистам;
- практичность. BI-системы и DaaS-решения для аналитики больших данных не только наглядно отображают наиболее важные бизнес-показатели, но и автоматизируют множество процессов бэк-офиса, нужных для создания понятного представления. При этом большинство готовых решений являются расширяемыми, предоставляя API или визуальные редакторы для создания собственных функций, скриптов и прочих уникальных настроек» [5];
- доступность. Помимо того, что готовые BI- и DaaS-решения не требуют вложения существенных инвестиций, для их использования необязательно наличие команды Data-профессионалов (инженер, аналитик, архитектор, разработчик, Data Scientist), а также

полноценной Big Data инфраструктуры.

Таким образом, внедрение BI/DaaS можно рассматривать как начальный этап цифровизации без глубокого реинжиниринга корпоративной деятельности, который реализует так называемую «эволюционную стратегию» постепенного улучшения.

Рассмотрим и сравним наиболее популярные решения BI-систем на предмет применения в качестве СППР на основе витрин больших данных.

4.1 BI-система MS Power BI

«MS Power BI - это коллекция программных служб, приложений и соединителей, которые взаимодействуют друг с другом, чтобы превратить разрозненные источники данных в согласованные, визуально иммерсивные и интерактивные аналитические данные.

Данные пользователя могут быть представлены в виде таблицы Excel или коллекции облачных и локальных гибридных хранилищ данных.

Power BI позволяет легко подключаться к источникам данных, визуализировать и выделять наиболее важную информацию и делиться ею с отдельными или всеми пользователями.

Решение состоит из нескольких элементов, которые работают вместе:

- классическое приложение Windows - Power BI Desktop;
- веб-служба SaaS (программное обеспечение как услуга) - служба Power BI;
- мобильные приложения Power BI для Windows, iOS и Android» [31].

На рисунке 25 представлена функциональная архитектура системы Power BI.

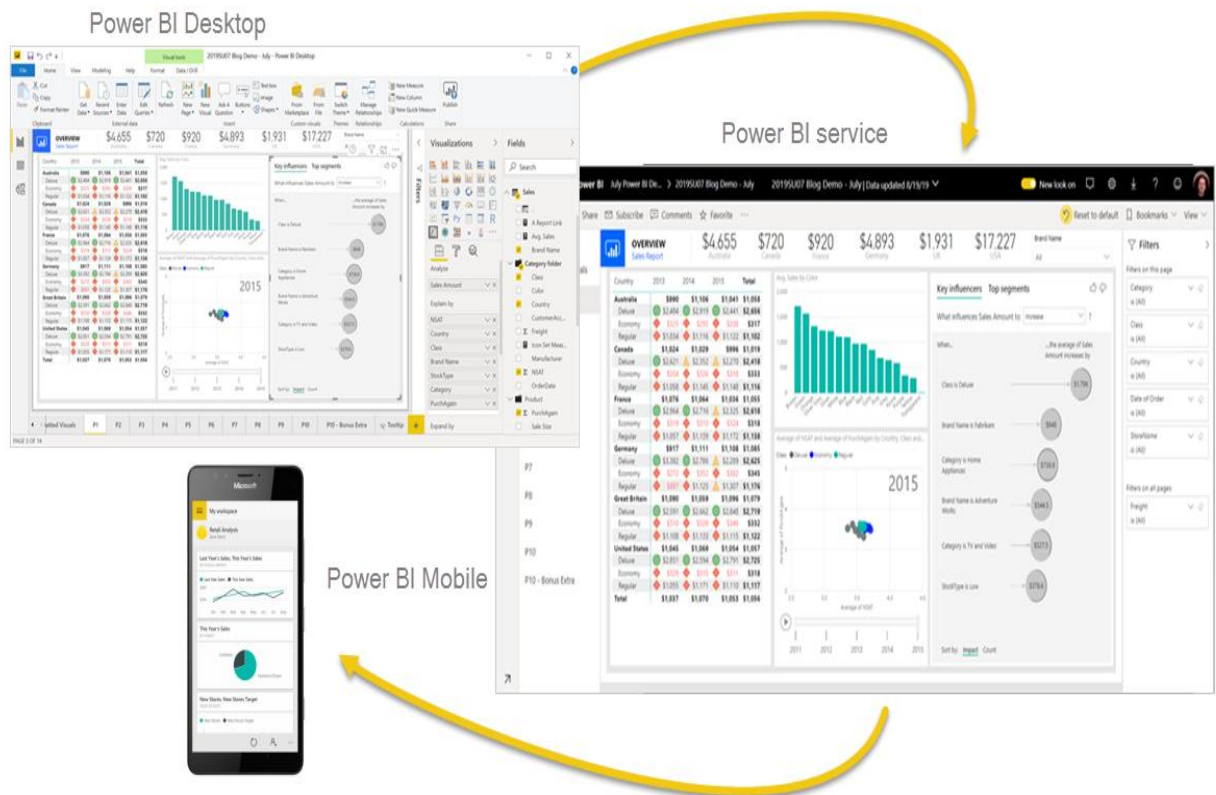


Рисунок 25 - Функциональная архитектура системы Power BI

По мнению вендора, основными преимуществами Power BI являются:

- простота подключения, моделирования и визуализации данных, возможность создавать запоминающиеся отчеты, персонализированные с учетом ключевых показателей эффективности и бренда компании;
- получение максимальной отдачи от инвестиций в большие данные, подключившись ко всем источникам данных с помощью шкалы для анализа, обмена и распространения информации организации, сохраняя при этом точность, согласованность и безопасность данных.
- простота интеграции с приложениями Microsoft Office, такими как Microsoft Teams и Excel.

Power BI позволяет получить быстрые ответы на вопросы, связанные с бизнесом клиента, с помощью искусственного интеллекта.

4.2 BI-система Tableau

Tableau - это самая широкая и глубокая платформа для сквозных данных и аналитики.

Tableau обеспечивает ответственное использование данных и улучшает бизнес-результаты с помощью полностью интегрированного управления данными, визуальной аналитики и повествования данных, а также совместной работы - и все это с помощью ведущего в отрасли решения Einstein от Salesforce [9].

Живая визуальная аналитика способствует неограниченному исследованию данных. Интерактивные информационные панели помогут пользователю мгновенно раскрыть скрытые идеи. Tableau использует естественную способность людей быстро определять визуальные закономерности (рисунок 26).

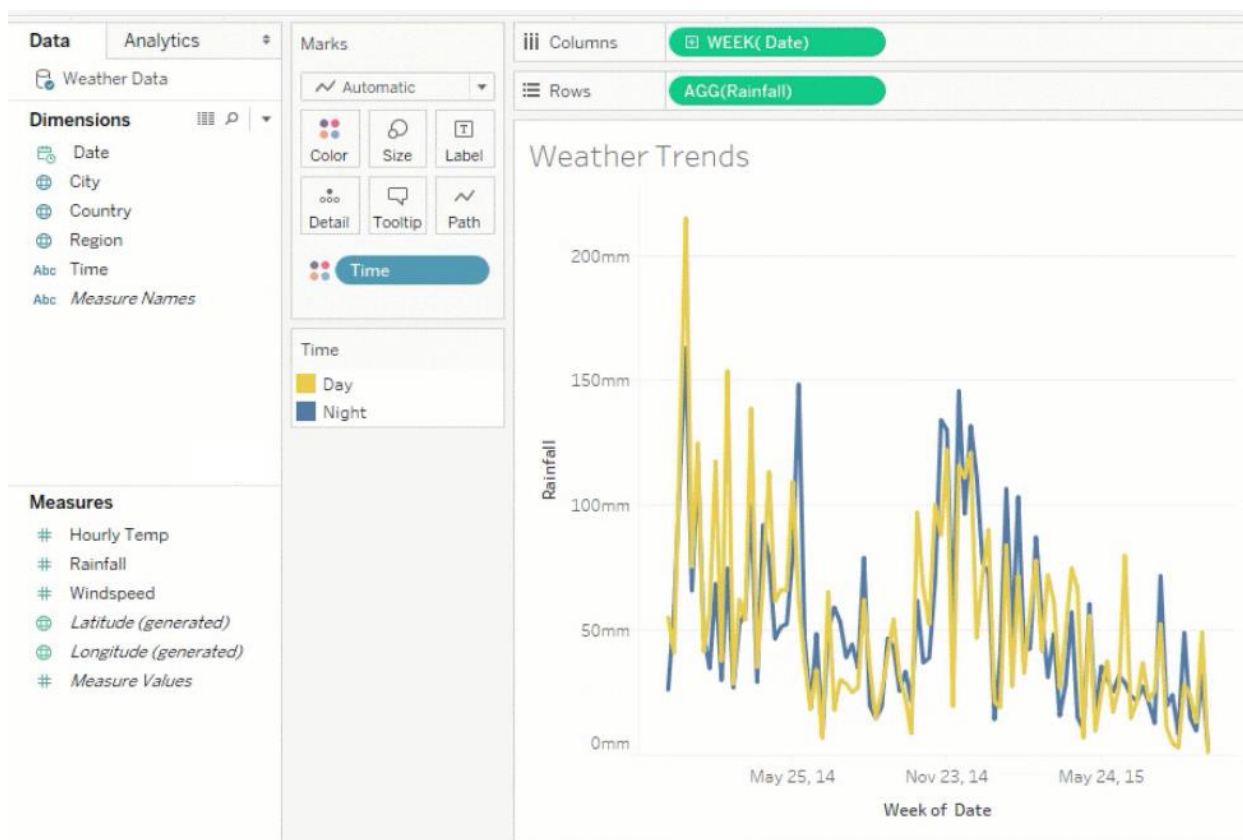


Рисунок 26 – Скриншот визуализации результатов анализа

Tableau предлагает самый широкий выбор и гибкость, чтобы продолжать расти вместе с компанией-владельцем по мере развития стратегии компании в области технологий и данных.

Система позволяет подключаться к данным локально или в облаке - будь то большие данные, база данных SQL, электронная таблица или облачные приложения, такие как Google Analytics и Salesforce.

Преимущества решения:

- анализ больших данных в реальном времени, в режиме In-memory;
- безопасное совместное использование и совместная работа;
- возможность подключения мобильных устройств;
- постоянное усовершенствование решения.

Обеспечивает доступ и объединение разрозненных данных без написания кода.

Опытные пользователи могут сводить, разделять и управлять метаданными для оптимизации источников данных.

4.3 BI-система QlikView

«По мнению вендора, инновационная система бизнес-анализа QlikView – это платформа для продвинутого бизнес-анализа, позволяющая создать источник «единой правды» для принятия взвешенных, точных и оперативных управленческих решений.

Это аналитическое программное обеспечение из категории BI, которое позволяет пользователям, не имеющим специальных технических навыков, формировать отчеты и анализировать информацию о деятельности своего предприятия» [11].

Архитектура системы представлена на рисунке 27.



Рисунок 27 - Архитектура BI-системы QlikView

Стратегия компании Qlik в отношении архитектуры и платформ позволяет создавать автоматизированные конвейеры данных и безопасно предоставлять их аналитику в гибком масштабе.

«QlikView располагает мощным ETL-инструментом, который имеет более двухсот встроенных функций для фильтрации, объединения и выполнения сложных операций над данными для загрузки их напрямую из различных источников» [11].

Платформа QlikView легко адаптируется к требованиям пользователя благодаря гибкой масштабируемости для удовлетворения пиковых требований постоянно растущего объема данных и сложности сценариев использования.

QlikView предлагает современную облачную аналитику на основе

искусственного интеллекта, которая дает всем сотрудникам компаний делать более важные анализы и принимать более обоснованные решения.

Следует отметить, что решения QlikView широко представлены на отечественном рынке программного обеспечения для финансовой сферы.

Для выбора решения используем таблицу сравнения характеристик BI-систем (таблица 7).

Таблица 7 – Сравнительный анализ характеристик BI-систем

Характеристики	Power BI	Tableau	QlikView
Производительность анализа больших объемов данных	2	2	2
Визуализация данных	2	3	2
Доступ к данным	3	3	2
Наличие бесплатной версии	3	1	0
Итого	10	9	6

Таким образом, по результатам сравнительного анализа выбираем в качестве готового ИТ-решения для анализа на основе больших данных BI-систему Power BI.

4.4 Апробация выбранных проектных решений и оценка эффективности BI-системы

Для проверки функциональных возможностей BI-системы на основе логической витрины данных, построенной на платформе решения Power BI используем в качестве источника данных представление базы данных «Кредиты», созданной средствами СУБД MS SQL Server 2017 [13].

Конструктор и скрипт тестового представления данных viewDM1 представлены на рисунке 28.

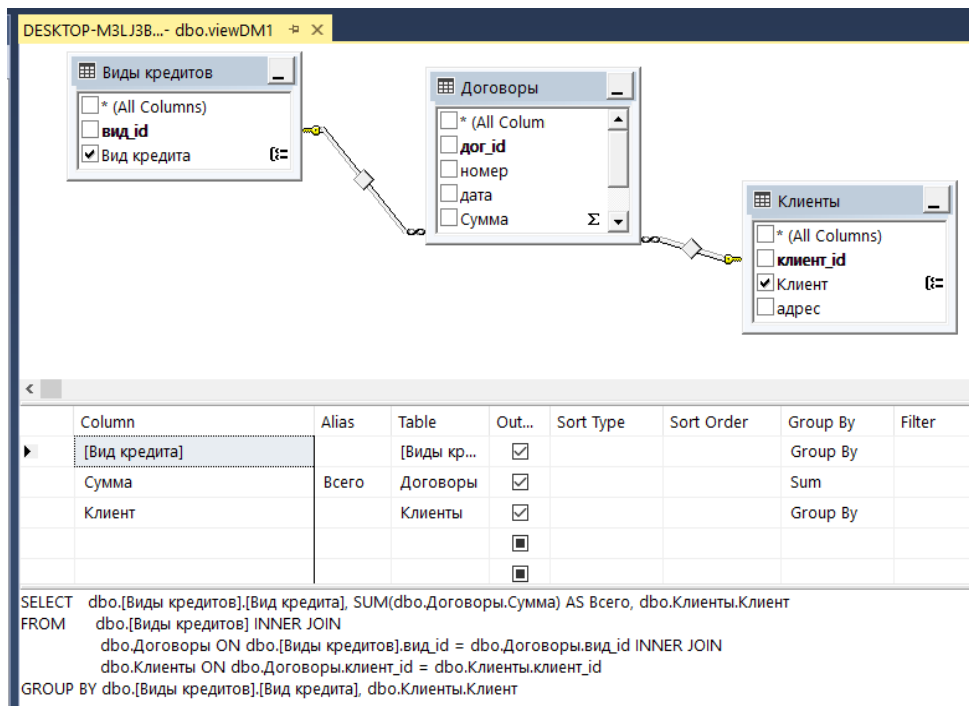


Рисунок 28 - Конструктор и скрипт представления данных viewDM1

Следует отметить, что таблицы БД «Кредиты» оптимизированы для работы в памяти с помощью механизма In-Memory СУБД MS SQL Server 2017.

После запуска программы Power BI был выбран режим подключения к источнику данных SQL Server с последующей загрузкой представления витрины данных viewDM1 (рисунок 29).

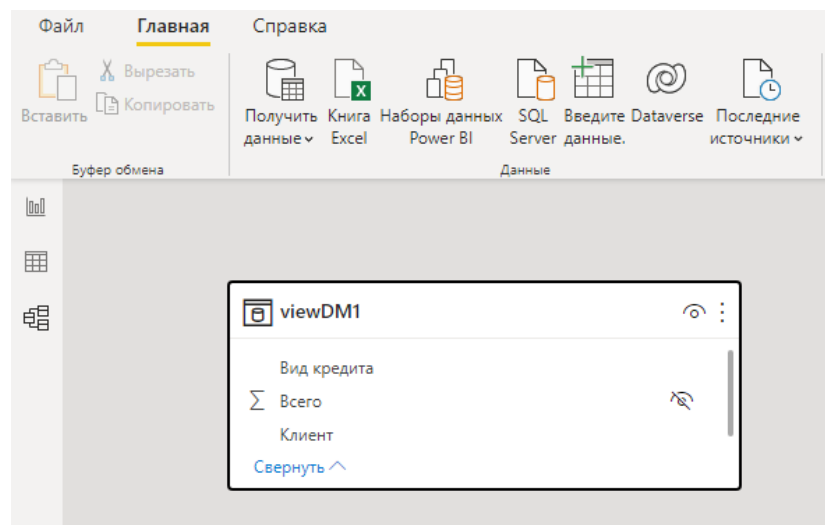


Рисунок 29 – Окно модели данных Power BI

Далее сформирован отчет с визуализацией на основе гистограммы.

Средствами Power BI проведен анализа данных, окно результатов которого показано на рисунке 30.



Рисунок 30 – Окно формирования отчета и анализа данных Power BI

Представленные результаты анализа данных подтверждают работоспособность ПО BI-системы.

Для оценки эффективности BI-системы используем показатель эффективности управления ИС.

Эффективность управления – это многоплановое понятие, которое включает следующие компоненты:

- целевую эффективность управления;
- экономическую эффективность управления;
- функциональную эффективность управления.

«Целевая эффективность управления характеризует степень достижения цели действий управляемого объекта при фактическом уровне реализации функций управления.

Под экономической эффективностью управления понимается степень полезной отдачи от выделенных средств на разработку, эксплуатацию системы и осуществление управления.

Под функциональной эффективностью управления понимается степень реализации органом управления возлагаемых на него функций.

Величина показателя функциональной эффективности управления $K_{\text{эу}}$ может вычисляться с помощью следующей формулы:

$$K_{\text{эу}} = \frac{\sum_{i=1}^n P_{yi}}{n}, \quad (2)$$

где n - количество функций управления, реализуемых ИС;

P_{yi} - вероятность выработки ИС эффективного управляющего воздействия при реализации i -й функции управления» [3].

Для ИС с высокой эффективностью управления величина данного показателя должна превышать значение 0,5.

Оценим эффективность ВІ-системы на основе показателя функциональной эффективности управления.

В рассматриваемом случае ВІ-системы поддерживает следующие функции:

- создание представления витрины данных;
- формирование аналитического отчета;
- анализ данных.

Как показывает практика, только на первую функцию может негативно повлиять человеческий фактор.

Пусть вероятность выработки эффективного управляющего воздействия для данной функции равна 0,5.

Тогда значение показателя функциональной эффективности управления будет равно:

$$K_{\text{эу}} = 2,5/3 = 0,83 \quad (3)$$

Таким образом, коэффициент эффективности управления ВІ-системы $K_{\text{эу}} > 0,5$, что свидетельствует о ее высокой функциональной эффективности управления.

Выводы по главе 4

В главе 4 магистерской диссертации представлены разработка программного обеспечения системы поддержки принятия решений на основе витрин больших данных и анализ ее эффективности.

Результаты проделанной работы позволили сделать следующие выводы:

- в настоящее время популярен подход к проектированию программного обеспечения ВІ-систем, основанный на применении готовых ИТ-решений;
- как показал сравнительный анализ, лучшими характеристиками для анализа на основе витрин больших данных обладает ВІ-система Power ВІ.

Коэффициент эффективности управления ВІ-системы $K_{\text{эу}} > 0,5$, что свидетельствует о ее высокой функциональной эффективности управления.

Заключение

Информация является необходимым производственным ресурсом для принятия эффективных управленческих решений. Менеджерам необходимо, чтобы эта информация была преобразована, предварительно обработана и соответствующим образом организована для быстрого доступа, анализа и принятия решений. Для решения данных задач используются аналитические системы поддержки принятия решений, относящиеся к классу BI (Business Intelligence) – систем.

В этой связи возникает задача проектирования BI – системы, компонентами которой будут витрины больших структурированных массивов данных.

Магистерская диссертация посвящена актуальной проблеме исследования и разработки математического и программного обеспечения, позволяющего повысить эффективность систем поддержки принятия управленческих решений на основе витрин больших данных.

Выполненные в работе научные исследования представлены следующими основными результатами:

- проанализировано современное состояние проблемы разработки математического и ПО СППР на основе витрин больших данных. В некоторых работах авторы обращают внимание на необходимость разработки инструментов, сочетающих функциональную мощь BI-систем со способностью к обработке больших объемов данных. Для достижения данной цели предлагается использовать различные технологии, основанные на применении витрин данных. Вместе с тем анализ позволил констатировать недостаточность работ по данной проблеме, что подтверждает актуальность темы магистерской диссертации;
- проведен анализ методологий и технологий построения СППР на основе витрин больших данных. Как показал анализ, при разработке витрин больших данных представляется целесообразным

использование методов построения логических витрин данных. Наиболее перспективной для построения VI-систем для анализа больших данных является технология, основанная на применении реляционной СУБД, поддерживающей механизм In-Memory;

- разработаны математическое и ПО СППР. Для построения логической модели СППР разработаны базовые диаграммы языка UML, отражающие различные аспекты системы. Для разработки метамodelей использован подход MDA. В настоящее время популярен подход к проектированию программного обеспечения VI-систем, основанный на применении готовых ИТ-решений. Как показал сравнительный анализ, лучшими характеристиками для анализа на основе витрин больших данных обладает VI-система Power VI. Апробация VI-системы и проведенные расчеты подтвердили эффективность предлагаемого решения СППР.

Таким образом, в работе решена актуальная научно-практическая проблема исследования и разработки математического и программного обеспечения, обеспечивающего повышение эффективности системы поддержки принятия управленческих решений, построенной на основе витрин больших данных.

Гипотеза исследования подтверждена.

Список используемой литературы и используемых источников

1. Алексеева Т.В. [и др.]. Информационные аналитические системы : учебник. М. : Московский финансово-промышленный университет «Синергия», 2013. 384 с.
2. Вахмянин И. Глава по BI в учебнике CDTO v4 [Электронный ресурс]. URL: <https://4cio.ru/content/4CDTO/%D0%92%D0%B0%D1%85%D0%BC%D1%8F%D0%BD%D0%B8%D0%BD%20-%D0%93%D0%BB%D0%B0%D0%B2%D0%B0%20%D0%BF%D0%BE%20BI%20%D0%B2%20%D1%83%D1%87%D0%B5%D0%B1%D0%BD%D0%B8%D0%BA%20CDTO%20v4.pdf> (дата обращения: 04.10.2021).
3. Вдовин В. М., Суркова Л. Е., Шурупов А.А. Предметно-ориентированные экономические информационные системы [Электронный ресурс]: учебное пособие. М. : Дашков и К, 2016. 386 с. URL: <https://znanium.com/catalog/product/415090> (дата обращения: 28.10.2021).
4. Виды обеспечения информационных систем [Электронный ресурс]. URL: <https://seniga.ru/stat/230-2010-10-22-13-37-05.html> (дата обращения: 04.10.2021).
5. Вичугова А. 5 причин, почему аналитика больших данных иногда выгоднее модного Data Science: взгляд со стороны бизнеса [Электронный ресурс]. URL: <https://www.bigdataschool.ru/blog/bi-analytics-big-data-vs-data-science.html> (дата обращения: 28.10.2021).
6. Леоненков А. В. Объектно-ориентированный анализ и проектирование с использованием UML и IBM Rational Rose [Электронный ресурс] : учебное пособие. М. : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. 317 с. [Электронный ресурс]. URL: <https://www.iprbookshop.ru/97554.html> (дата обращения: 16.10.2021).
7. Логическая витрина для доступа к большим данным [Электронный ресурс]. URL: <https://habr.com/ru/post/280051/> (дата обращения: 04.10.2021).
8. Мкртычев С.В. Автоматизированная система управления

эффективностью работы страховых агентов // Современные проблемы науки и образования. 2014. № 6. URL: <http://www.science-education.ru/120-15365> (дата обращения: 04.10.2021).

9. Платформа Tableau [Электронный ресурс]. URL: <https://www.tableau.com/products/our-platform> (дата обращения: 28.10.2021).

10. Прокопенко Н. Ю. Аналитические информационные системы поддержки принятия решений [Электронный ресурс] : учебное пособие. Нижний Новгород : Нижегородский государственный архитектурно-строительный университет, ЭБС АСВ, 2020. 143 с. URL: <https://www.iprbookshop.ru/107361.html> (дата обращения: 04.10.2021).

11. Сайт компании Qlik [Электронный ресурс]. URL: <https://www.qlik.com/us/> (дата обращения: 28.10.2021).

12. Самуйлов С. В. Объектно-ориентированное моделирование на основе UML [Электронный ресурс] : учебное пособие. Саратов: Вузовское образование, 2016. 37 с. URL: <http://www.iprbookshop.ru/47277.html> (дата обращения: 04.09.2021).

13. СУБД MS SQL Server [Электронный ресурс]. URL: <https://www.microsoft.com/en-ie/sql-server/sql-server-downloads> (дата обращения: 28.10.2021).

14. Тоноян С.А., Высочанский В.А. Методика проектирования корпоративного хранилища данных на базе платформы sap Net Weaver Business warehouse // Вестник МГТУ им. Н.Э. Баумана. Серия «Приборостроение». 2016. №4 (109). С.33-48.

15. Что такое КХД и как его связать с Big Data [Электронный ресурс]. URL: <https://www.bigdataschool.ru/blog/lisa-data-warehouse-architecture.html> (дата обращения: 04.10.2021).

16. A.A.A. Gad-Elrab Modern Business Intelligence: Big Data Analytics and Artificial Intelligence for Creating the Data-Driven Value [Электронный ресурс]. URL: <https://www.intechopen.com/chapters/76332> (дата обращения: 04.10.2021).

17. Analytics on AWS [Электронный ресурс]. URL:

<https://aws.amazon.com/ru/big-data/datalakes-and-analytics/what-is-a-data-lake/>
(дата обращения: 04.10.2021).

18. B. Sudrajat “Analysis and Design of Data Mart Decision Support Systems at PT Marlindo Tirta Nusantara”, Journal Publications & Informatics Engineering Research Volume 3, Number 2, April 2019.

19. Bara A. et al. A model for Business Intelligence Systems’ Development, Informatica Economică vol. 13, no. 4/2009, pp. 99-107.

20. Belov V., Kosenkov A.N., Nikulchev E. Experimental Characteristics Study of Data Storage Formats for Data Marts Development within Data Lakes. Appl. Sci. 2021, 11, 8651.

21. Bhargava S. “DWH-Performance Tuning Using Metadata Driven Approach”, Dissertation Submitted In partial fulfillment For the award of the Degree of Master of Technology, 2011.

22. Business Intelligence на больших данных — наш опыт интеграции [Электронный ресурс]. URL: <https://habr.com/ru/company/visiology/blog/536732/> (дата обращения: 04.10.2021).

23. Data Warehouse, Data Marts and Online Analytical Processing (OLAP) [Электронный ресурс]. URL: <https://www.managementstudyguide.com/data-warehouse-data-marts-and-olap.htm> (дата обращения: 04.10.2021).

24. Gartner, Inc. [Электронный ресурс]. URL: <https://www.gartner.com> (дата обращения: 04.10.2021).

25. I. Ahmad, S. Azhar, P. Lukauskis “Development of a decision support system using data warehousing to assist builders/developers in site selection”, Automation in Construction 13 (2004) 525 – 542.

26. In-Memory Processing [Электронный ресурс]. URL: <https://hazelcast.com/glossary/in-memory-processing/> (дата обращения: 04.10.2021).

27. Kalelkar M. et al. Implementation of Model-View-Controller Architecture Pattern for Business Intelligence Architecture, International Journal of Computer

Applications (0975 – 8887) Volume 102– No.12, September 2014.

28. Logical Data Warehouse [Электронный ресурс]. URL: <https://www.tibco.com/reference-center/what-is-a-logical-data-warehouse> (дата обращения: 04.10.2021).

29. Mkrtychev, S.; Enik, O. Automated Underwriting Control in a Regional Insurance Company, Proceedings of the International Conference on Trends of Technologies and Innovations in Economic and Social Studies on October 02, 2018.

30. Model-View-Controller design pattern [Электронный ресурс]. URL: <https://help.hcltechsw.com/commerce/9.1.0/developer/concepts/csdmvcdespat.html> (дата обращения: 04.09.2021).

31. MS Power BI [Электронный ресурс]. URL: <https://powerbi.microsoft.com/en-us/desktop/> (дата обращения: 28.10.2021).

32. N. Houari and B. H. Far “An Intelligent Project lifecycle Data Mart-based Decision Support System”, Canadian Conference on Electrical and Computer Engineering 2004 (IEEE Cat. No.04CH37513).

33. Object Management Group [Электронный ресурс]. URL: <https://www.omg.org/> (дата обращения: 28.10.2021).

34. R. A. Jaleel and T.M.J. Abbas “Design and Implementation of Efficient Decision Support System Using Data Mart Architecture”, Proc. of the 2 nd International Conference on Electrical, Communication and Computer Engineering (ICECCE) 12-13 June 2020, Istanbul, Turkey.

35. R. Kimball “The Data Warehouse Toolkit: the Complete Guide to Dimensional Modeling”, 2002.

36. Relational online analytical processing (ROLAP) [Электронный ресурс]. URL: <https://searchoracle.techtarget.com/definition/relational-online-analytical-processing> (дата обращения: 04.10.2021).

37. Stefanov V., List B., Korherr B. (2005) Extending UML 2 Activity Diagrams with Business Intelligence Objects. In: Tjoa A.M., Trujillo J. (eds) Data Warehousing and Knowledge Discovery. DaWaK 2005. Lecture Notes in Computer Science, vol 3589. Springer, Berlin, Heidelberg.

38. Teradata Database Design [Электронный ресурс]. URL: <https://docs.teradata.com/r/w4DJnG9u9GdDlXzsTXyItA/lipw7BdwMhNUKxZAVfKDBg> (дата обращения: 04.10.2021).

39. What is a Data Mart? [Электронный ресурс]. URL: <https://www.oracle.com/autonomous-database/what-is-data-mart/> (дата обращения: 04.10.2021).

40. Why ETL Data Modeling is Critical in 2021 [Электронный ресурс]. URL: <https://www.xplenty.com/blog/etl-data-modeling-why-is-it-important/> (дата обращения: 28.10.2021).