

Саранск 2022

ЗАДАНИЕ

на производственную практику
вид практики

научно-исследовательская работа

Тип практики в соответствии с ОПОП ВО

Студенту Гришунину Андрею Александровичу

Фамилия Имя Отчество

1 курса, 103М группы

Направление подготовки / Специальность 09.04.04 Программная инженерия

код Наименование

Профиль / Специализация Управление разработкой программных проектов

Место прохождения практики г. Саранск, ФГБОУ ВО
«МГУ им. Н. П. Огарёва», кафедра систем автоматизированного
проектирования факультета математики и информационных технологий,
научно-исследовательская лаборатория № 415

Населенный пункт, профильная организация/структурное подразделение Университета

Срок прохождения практики 27.01.2022 – 08.06.2022

начало (дата) – окончание (дата)

Срок представления отчёта студентом на защиту 08.06.2022

дата

1 Цель и задачи практики

Целью прохождения практики «Научно-исследовательская работа» является ознакомление обучающихся с концептуальными основами сбора, обработки, анализа и систематизации научно-технической информации по теме исследования, выбор методик и средств решения задачи; формирование научного мировоззрения на основе знания предметной области темы исследования; воспитание научно-исследовательских навыков.

Задачами практики являются:

- приобретение опыта проведения исследований, посвященных решению актуальной научной проблемы;
- подготовка научно-технических отчетов, обзоров, публикаций по результатам выполненных исследований;
- подготовка материалов, необходимых для выполнения выпускной квалификационной работы.

2 Компетенции студента, формируемые в результате прохождения практики

- Способность выполнить постановку новых задач анализа и синтеза новых проектных решений (ПК-5);
- Понимает существующие подходы к верификации моделей программного обеспечения (ПК-6);
- Способен проектировать трансляторы и интерпретаторы языков программирования (ПК-7);
- Способен проектировать сетевые службы (ПК-8);
- Способен проектировать основные компоненты операционных систем (ПК-9);
- Способен применять, расширять и уточнять знания, умения и навыки в области прикладной математики, фундаментальной информатики и программной инженерии, ориентируясь на выбранную область научной или производственной деятельности (ПК-10).

3 Задание на практику

27.01.2022 Получение задания на практику.

03.02.2022 Определение технических характеристик.

21.02.2022 Формирование требований к разрабатываемой системе.

07.03.2022 Анализ инструментов для реализации системы.

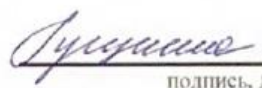
04.04.2022 Анализ СУБД для реализации системы.

09.05.2022 Изучение архитектур для реализации системы.


02.06.2022 Составление отчета о проделанной работе.

08.06.2022 Защита отчета.

Руководитель практики
от университета,
канд. техн. наук, доцент

 04.04.2022 О. А. Гушина
подпись, дата

Задание к исполнению принял

 17.01.2022 А. А. Гришунин
подпись, дата

РЕФЕРАТ

Отчёт по производственной (научно-исследовательская работа) практике содержит 25 с., 1 табл., 5 источн.

ПРОБЛЕМА; ЛПР; ЦЕЛЬ; ОПЕРАЦИЯ; РЕЗУЛЬТАТ; МОДЕЛЬ; УПРАВЛЕНИЕ; РЕШЕНИЕ; УСЛОВИЯ; АЛЬТЕРНАТИВА; КРИТЕРИЙ; НАИЛУЧШЕЕ РЕШЕНИЕ; СИСТЕМЫ ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЙ.

Целью производственной (научно-исследовательская работа) практики является ознакомление обучающихся с концептуальными основами сбора, обработки, анализа и систематизации научно-технической информации по теме исследования, выбор методик и средств решения задачи; формирование научного мировоззрения на основе знания предметной области темы исследования; воспитание научно-исследовательских навыков.

Место проведения практики – кафедра систем автоматизированного проектирования факультета математики и информационных технологий, научно-исследовательская лаборатория № 415.

Во время практики выполнено индивидуальное задание, проделанная работа отражена в дневнике практики.

Получен положительный отзыв руководителя практики от Университета.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	7
1. Создание концепций и требований к разрабатываемой системе	8
2. Определение инструментов разработки для реализации поставленной задачи	12
3. Выбор архитектуры системы.....	18
ЗАКЛЮЧЕНИЕ	24
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	25

ВВЕДЕНИЕ

Принятие решений – каждодневная деятельность человека, часть его повседневной жизни. Простые, привычные решения человек принимает легко, часто автоматически, особо не задумываясь. В сложных и ответственных случаях человек обращается к опытным и знающим людям за подтверждением своего решения, несогласием с ним или за советом: каким могло бы быть другое решение. Такие обращения можно назвать процессом поддержки принятия решения.

Принятие решения в большинстве случаев заключается в генерации возможных альтернатив решений, их оценке и выборе лучшей альтернативы.

Принять "правильное" решение – значит выбрать такую альтернативу из числа возможных, в которой с учетом всех разнообразных факторов будет способствовать достижению поставленной цели. Поскольку человек постоянно сталкивается с таким выбором, можно сделать вывод, что разработка систем для поддержки процесса принятия решений является **актуальной** прикладной задачей.

Целью данной работы является разработка приложения для постановки диагноза на основе теории принятия решений.

Для достижения поставленной цели были сформулированы следующие **задачи:**

- изучить и проанализировать литературу, посвященную исследуемой теме;
- изучить имеющиеся аналоги и описать особенности их функционирования;
- разработать приложение для поддержки процесса принятия решений;
- выявить имеющиеся недостатки разработанной системы и, в случае их обнаружения, внести в неё необходимые коррективы.

1. Создание концепций и требований к разрабатываемой системе

Для дальнейшего выбора системы необходимо сформулировать требования к программному обеспечению.

Требования к программному обеспечению – совокупность утверждений относительно атрибутов, свойств или качеств программной системы, подлежащей реализации.

Изучив предметную область и специфику систем, были сформулированы оптимальные требования, представленные далее.

1.1 Определение технических характеристик

Будущая система должна быть реализована по принципу трёхзвенной архитектуры:

- тонкий клиент на рабочих местах пользователей;
- сервер приложений, который производит обработку запросов с клиентских рабочих мест и реализует всю логику системы;
- сервер базы данных, который обеспечивает хранение и выборки данных.

Преимущества трехзвенной системы по сравнению с клиент-серверной или файл-серверной архитектурой можно выделить следующие достоинства трёхуровневой архитектуры:

- масштабируемость;
- конфигурируемость – изолированность уровней друг от друга позволяет (при правильном развертывании архитектуры) быстро и простыми средствами переконфигурировать систему при возникновении сбоев или при плановом обслуживании на одном из уровней;
- высокая безопасность;
- высокая надёжность;
- низкие требования к скорости канала (сети) между терминалами и сервером приложений;

- низкие требования к производительности и техническим характеристикам терминалов, как следствие снижение их стоимости. Терминалом может выступать не только компьютер, но и, например, мобильный телефон.

Доступ конечных пользователей к системе осуществляется через тонкий клиент посредством web-интерфейса на единой для всех участников информационного обмена БД.

Передача данных между сервером приложений и «тонким клиентом» выполняется с использованием стандартного web-протокола HTTP (HTTPS).

СУБД, используемая для управления оперативными базами данных функциональных компонент, входящих в состав системы, обеспечивает:

- масштабируемость. Отсутствие существенного снижения скорости выполнения пользовательских запросов при пропорциональном росте количества запросов и аппаратных ресурсов, используемых данной СУБД (таких как объем оперативной памяти, количество процессоров и серверов);

- надежность. Минимальную вероятность сбоев, наличие средств восстановления данных после сбоев, инструментов резервного копирования и дублирования данных. Наличие таких функций, как зеркалирование БД, кластеризация с переключением на резервный ресурс, создание снимков БД (повышение эффективности оперативной работы, минимизация простоев и обеспечение постоянной готовности важнейших корпоративных систем);

- управляемость. Простоту администрирования, наличие средств автоматического конфигурирования (средств создания баз данных и их объектов, инструменты описания правил репликации данных между различными серверами, утилиты управления;

- пользователями, группами пользователей и их правами, средства мониторинга событий, средства просмотра планов выполнения запросов, утилиты миграции из других СУБД), единую консоль управления, интеграцию с базовой инфраструктурой;

- безопасность. Безопасные настройки по умолчанию, реализацию политики паролей, гранулярное управление разрешениями;
- эффективность работы. Расширенный набор поддерживаемых языков, возможности расширения, оптимизация доступа к данным, структура приложений.

1.2 Формирование требований к системе

Требования к вычислительной технике средств разработки:

- Процессор с частотой 1,6 ГГц или более мощный;
- Оперативная память 4 ГБ;
- 4 ГБ доступного пространства на жёстком диске;
- жёсткий диск с частотой вращения 5 400 об/мин;
- Видеокарта NVIDIA Geforce GTX 750;
- операционная система Windows 10;
- разрешения экрана 1024x768 или выше.

Требования к средствам разработки системы поддержки принятия решений (СППР):

- PyCharm 2022.1.2;
- Python 3.10.4;
- Google Chrome последней версии.

Системные требования:

Для эксплуатации разрабатываемой системы необходимы следующие условия:

- электропитание технических средств от сети напряжением 220 Вт с частотой 50 Гц.;
- физическая защита аппаратных компонентов системы, носителей данных, резервирование ресурсов.

Функциональные требования:

- Интерфейс системы, и результаты ее работы должны быть представлены русским языком;

- Доступ в СППР должно предоставляться только авторизованным пользователям;

- В разрабатываемой СППР предусмотрены следующие классы пользователей: не авторизованный посетитель, авторизованный пользователь, администратор системы.

Для каждого класса пользователей сформированы свои функциональные возможности.

Требования к дизайну системы:

Интерфейс открытой и закрытой части системы должен быть выполнен в светлой гамме и идентичном дизайне. В закрытой части системы должно быть реализовано вертикальное меню, отражающее перечень разделов системы.

Требования к информационному наполнению системы:

В системе не допускается размещение рекламной информации и информации, направленной на пропаганду войны, разжигание национальной, расовой или религиозной ненависти и вражды, а также иной информации, за распространение которой предусмотрена уголовная или административная ответственность.

Требования к разделению доступа в систему:

Набор разделов системы, доступных авторизованным и не авторизованным пользователям должен быть отличен.

При попытке пользователя, не имеющего данных авторизации, или при вводе некорректных данных авторизации с целью перехода в закрытую часть системы, должно высветиться уведомление о некорректности введенных данных, и форма авторизации должна очистить поля.

2. Определение инструментов разработки для реализации поставленной задачи

2.1 Выбор языка программирования

Первая часть данной работы – создание онтологии для проектирования базы знаний профессий и их характеристик – будет решаться в программе для создания онтологий.

В последнее время количество общедоступных редакторов онтологий перевалило за 100. Но редко можно встретить универсальное и в то же время полезное средство. Рассмотрим существующие программы построения онтологий.

Ontolingua

Кроме собственно редактора онтологий, эта система содержит:

- сетевой компонент Webster, предназначенный для определения концептов;
- сервер, обеспечивающий доступ к онтологиям Ontolingua по протоколу OKBC (Open Knowledge Base Connectivity);
- Chimaera – инструмент для анализа и объединения онтологий.

Protege

Это свободно распространяемая Java-программа, предназначенная для построения (создания, редактирования и просмотра) онтологий той или иной прикладной области. Она включает редактор онтологий, позволяющий проектировать онтологии, разворачивая иерархическую структуру абстрактных и конкретных классов и слотов. На основе сформированной онтологии Protege позволяет генерировать формы получения знаний для введения экземпляров классов и подклассов.

Данный инструмент поддерживает использование языка OWL и позволяет генерировать HTML-документы, отображающие структуру онтологий. Поскольку он использует фреймовую модель представления знаний ОКВС, это позволяет адаптировать его и для редактирования моделей

предметных областей, представленных не в OWL, а в других форматах (UML, XML, SHOE, DAML+OIL, RDF / RDFS и т.п.).

DOE

DOE (Differential Ontology Editor) – простой редактор, который позволяет пользователю создавать онтологии. Процесс спецификации онтологии состоит из трех этапов.

- На первом этапе пользователь строит таксономию понятий и отношений, явным образом очерчивая позицию каждого элемента (понятие) в иерархии. Затем пользователь указывает, в чем специфика понятия относительно его «родителя», и в чем это понятие подобно или отлично от его «братьев». Пользователь может также прибавить синонимы и энциклопедическое определение на нескольких языках для всех понятий.

- На втором этапе две таксономии рассматриваются с разных точек зрения. Пользователь может расширить их новыми объектами или добавить ограничения на области отношений.

- На третьем этапе онтология может быть переведена на язык представления знаний.

OntoEdit

OntoEdit – инструментальное средство, обеспечивающее просмотр, проверку и модификацию онтологии. Оно поддерживает языки представления онтологии OIL и RDFS, а также внутренний язык представления знаний OXML, основанный на XML. Как и Protege, это автономное Java-приложение, но его коды закрыты. Свободно распространяемая версия OntoEdit Free ограничена 50 концептами, 50 отношениями и 50 экземплярами.

Python

Python – интерпретируемый язык программирования, позволяющий упростить отладку программ, что обуславливает сравнительно низкую скорость выполнения программы. Обладает библиотеками для построения онтологии. Является бесплатным для использования и распространения. Поддерживает ООП.

OilEd

OilEd – автономный графический редактор онтологий, разработанный в рамках проекта On-To-Knowledge. Он свободно распространяется по общедоступной лицензии GPL. Инструмент использует для представления онтологий язык OIL. В OilEd отсутствует поддержка экземпляров классов.

WebOnto

WebOnto представляет собой Java-апплет и разработан для просмотра, создания и редактирования онтологий. Для моделирования онтологий он использует язык OCML (Operational Conceptual Modeling Language). Пользователь может создавать различные структуры, в том числе классы со множественным наследованием. Инструмент имеет ряд полезных особенностей: просмотр отношений, классов и правил, возможна совместная работа над онтологией нескольких пользователей.

ODE, WebODE

ODE (Ontological Design Environment) взаимодействует с пользователями на концептуальном уровне, обеспечивает их набором таблиц для заполнения (концептов, атрибутов, отношений) и автоматически генерирует код на языках LOOM, Ontolingua и F-Logic. Данный инструмент получил свое развитие в редакторе онтологий WebODE, который интегрирует все сервисы ODE в единую архитектуру, сохраняя свои онтологии в реляционной БД.

Таким образом, на основе анализа инструментов, используемых при разработке систем поддержки принятия решений, был сделан выбор в пользу языка программирования Python, который оказался наиболее удачным выбором в качестве платформы для реализации системы.

2.2 Выбор СУБД

В идеальном случае СУБД должна обеспечивать реализацию всех возлагаемых на БД задач и укладываться в ограничения, накладываемые техническими средствами. Для окончательного выбора СУБД необходимо проанализировать свойства СУБД, ОС, в которой может работать СУБД,

возможные структуры данных, типы запросов, возможность создания метаданных, интеграции данных, средств экспорта-импорта, возможность использования ранее созданных приложений. Сведем наиболее используемые СУБД и их основные характеристики в одну таблицу.

Таблица 1 – Характеристики СУБД

Компания	СУБД	Краткая характеристика
Borland	InterBase	SQL-совместимая реляционная, ОС - Windows, Linux и Solaris.
IBM	DB2 Universal Database, Informix	Мультимедийная, Web, ОС - Unix, Linux и Windows, аппаратные платформы - zSeries, iSeries, VSE и VM. СУБД для систем масштаба предприятия и рабочей группы, обеспечивает работу с очень крупными БД в условиях дефицита ресурсов. Используемые языки - Java, SQL 2000
InterSystems	Cache	Постреляционная СУБД, включающая сервер многомерных данных и средство обработки запросов на языке SQL. ОС - Unix, Mac OS, Linux, Windows (32- и 64-разрядные версии).
Microsoft	SQL Server	Реляционная СУБД для управления данными в масштабе предприятия, поддерживает технологии XML и Интернет, обладает встроенным средством анализа и извлечения данных, интегрированным с MS Office, ОС - Windows. Используемый язык Transact-SQL, XML
Oracle	Oracle	СУБД для масштабной обработки транзакций (OLTP), хранилищ данных с высокой интенсивностью потока запросов и ресурсоемких Интернет- приложений. ОС - Unix, Windows и Linux. Последняя версия поддерживает Grid-вычисления. Используемые языки Java, Delphi PL/SQL, XML
Microsoft	dBASE III Plus	ОС Windows. Создаваемые ею файлы импортируются СУБД FoxPro, Paradox, MS Access, а также пакетами прикладных программ MS Excel, Surfer, Grapher и др.
Рэлэкс	Линтер	Реляционная СУБД, имеющая сертификат Гостехкомиссии при Президенте РФ на соответствие 2 классу защиты информации от несанкционированного доступа, ОС - Unix, Linux, QNX, VAX/VMS, OpenVMS, DOS, Windows, NetWare, OS/2
ВНИИНС	Паллада	Объектно-ориентированная СУБД, предназначенная для АСУ вооруженных сил, функционирует в среде ОС MCBC и ОЛИВИЯ

Так как приложение ориентировано на интернет технологии, для разработки базы данных выбрана СУБД oracle. Она обладает еще рядом преимуществ, таких как высочайшая надежность, возможность разбиения крупных баз данных на разделы (large-database partition), что дает возможность эффективно управлять гигантскими гигабайтными базами; наличие универсальных средств защиты информации; эффективные методы максимального повышения скорости обработки запросов; индексация по битовому отображению; свободные таблицы (в других СУБД все таблицы заполняются сразу при создании); распараллеливание операций в запросе, а также наличие широкого спектра средств разработки, мониторинга и администрирования.

3. Выбор архитектуры системы

Известно несколько способов построения СППР, и большинство из них основано на технологиях хранилищ и витрин данных. На сегодняшний день можно выделить четыре наиболее популярных типа архитектур систем поддержки принятия решений:

- Функциональная СППР.
- Независимые витрины данных.
- Двухуровневое хранилище данных.
- Трехуровневое хранилище данных.

Функциональная СППР

Функциональная СППР (рисунок 1) является наиболее простой с архитектурной точки зрения. Такие системы часто встречаются на практике, особенно в организациях с невысоким уровнем аналитической культуры и недостаточно развитой информационной инфраструктурой.

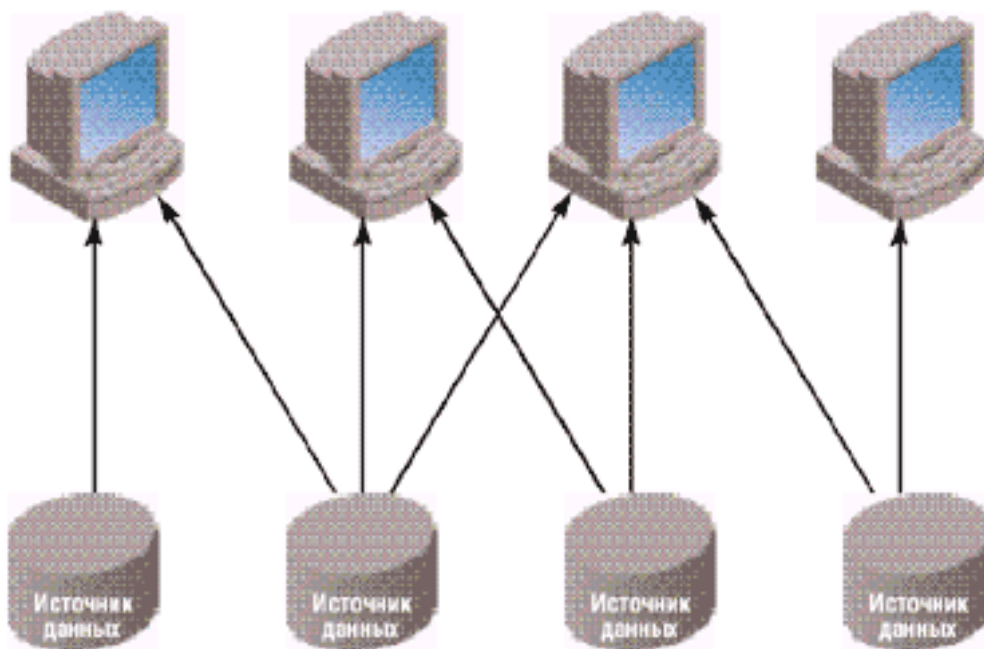


Рисунок 1 – Функциональные СППР

Функциональная СППР характерна тем, что при анализе данных система использует данные из оперативных систем.

Преимущества:

- быстрое внедрение за счет отсутствия этапа перегрузки данных в специализированную систему;
- минимальные затраты за счет использования одной платформы.

Недостатки:

- единственный источник данных, потенциально сужающий круг вопросов, на которые может ответить система;
- оперативные системы характеризуются очень низким качеством данных с точки зрения их роли в поддержке принятия стратегических решений; в силу отсутствия этапа очистки данных данные функциональной СППР, как правило, также обладают невысоким качеством;
- большая нагрузка на оперативную систему — сложные запросы могут привести к остановке работы оперативной системы, что весьма нежелательно.

СППР с использованием независимых витрин данных

Независимые витрины данных (рисунок 2) часто встречаются в крупных организациях с большим количеством независимых подразделений, зачастую имеющих свои собственные отделы информационных технологий.

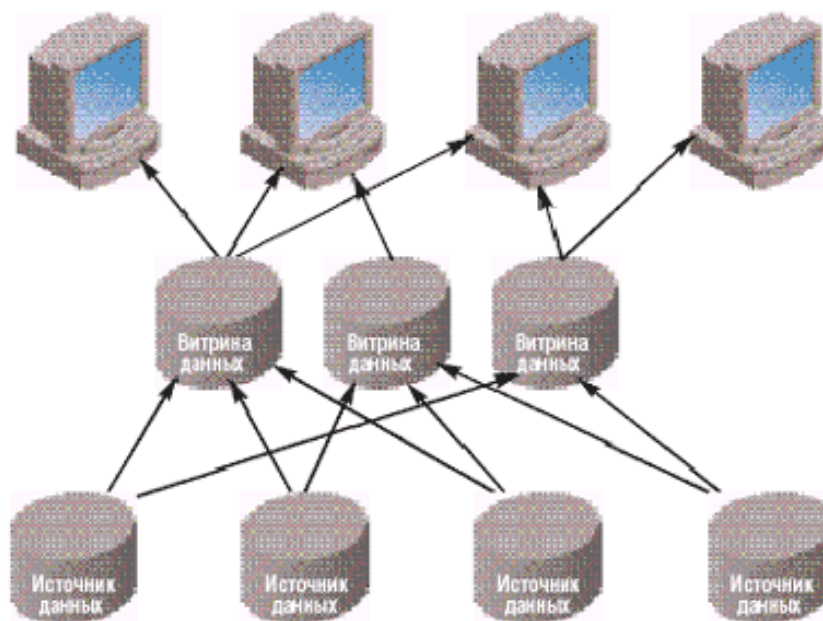


Рисунок 2 – Независимые витрины данных

Преимущества:

- витрины данных можно внедрять достаточно быстро;
- витрины проектируются для ответов на конкретный ряд вопросов;
- данные в витрине оптимизированы для определенных групп пользователей, что облегчает процедуры их наполнения, а также способствует повышению производительности.

Недостатки:

- данные хранятся многократно в различных витринах данных, что приводит к дублированию информации и, как следствие, к увеличению расходов на хранение и возможным проблемам, связанным с необходимостью поддержания непротиворечивости данных;
- потенциально очень сложный процесс наполнения витрин данных при большом количестве источников данных;

СППР на основе двухуровневого хранилища данных

Двухуровневое хранилище данных (рисунок 3) строится централизованно для предоставления информации в рамках компании. Для поддержки такой архитектуры необходима выделенная команда профессионалов в области хранилищ данных. Это означает, что вся организация должна согласовать все определения и процессы преобразования данных.

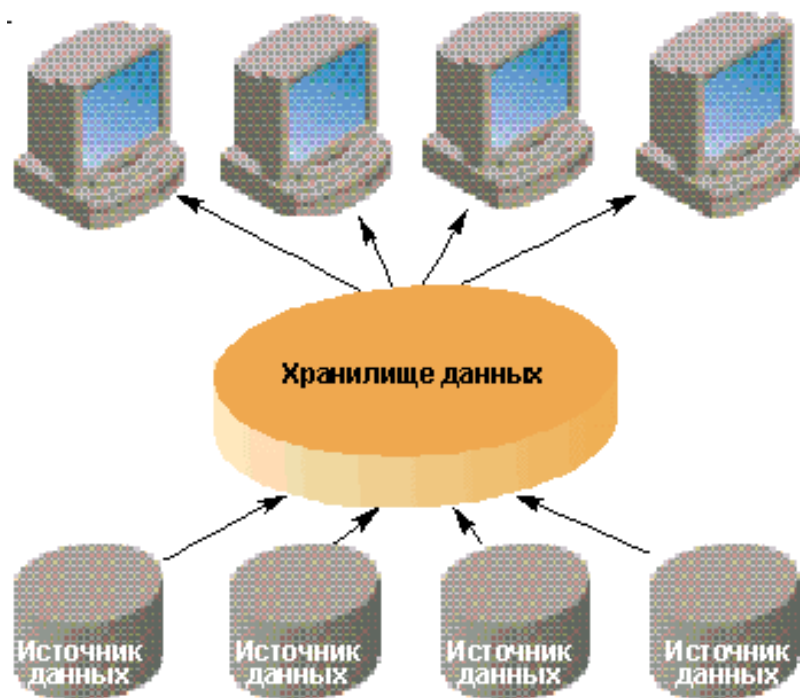


Рисунок 3 –Двухуровневое хранилище данных

Преимущества:

- данные хранятся в единственном экземпляре;
- минимальные затраты на хранение данных;
- отсутствуют проблемы, связанные с синхронизацией нескольких копий данных;
- данные консолидируются на уровне предприятия, что позволяет иметь единую картину бизнеса.

Недостатки:

- данные не структурируются для поддержки потребностей отдельных пользователей или групп пользователей;
- возможны проблемы с производительностью системы;
- возможны трудности с разграничением прав пользователей на доступ к данным.

СППР на основе трехуровневого хранилища данных

Хранилище данных представляет собой единый централизованный источник информации для всего предприятия. Витрины данных отражают подмножества данных из хранилища, организованные для решения задач отдельных подразделений компании. Конечные пользователи имеют возможность доступа к детальным данным хранилища, в случае если данных в витрине недостаточно, а также для получения более полной картины состояния бизнеса.

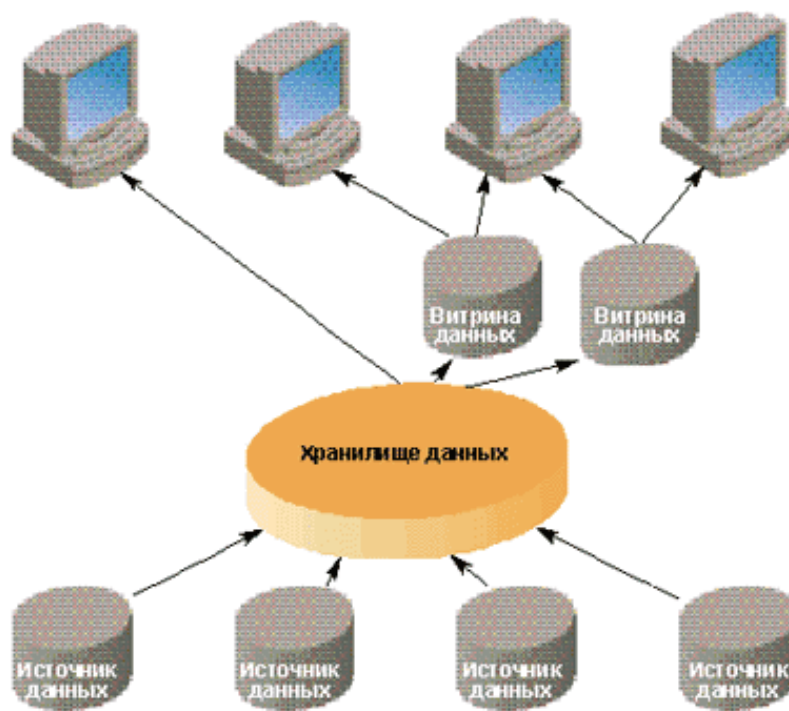


Рисунок 4 – Трехуровневое хранилище данных

Преимущества:

- создание и наполнение витрин данных упрощено, поскольку наполнение происходит из единого стандартизованного надежного источника очищенных нормализованных данных;
- имеется (построена) общая структура данных предприятия;
- витрины данных синхронизированы и совместимы с общей структурой данных предприятия;
- существует возможность сравнительно легкого расширения хранилища и добавления новых витрин данных;
- гарантированная производительность.

Недостатки:

- избыточность данных, ведущая к росту объема хранилища;
- требуется согласованность с принятой архитектурой многих областей с потенциально различными требованиями (например, скорость внедрения иногда конкурирует с требованиями следовать архитектурному подходу).

Таким образом, на основе анализа архитектуры, используемых при разработке систем поддержки принятия решений, был сделан выбор в пользу двухуровневого хранилища данных, который оказался наиболее удачным выбором в качестве архитектуры для реализации системы.

ЗАКЛЮЧЕНИЕ

В процессе выполнения данной работы были приобретены следующие навыки и проделана следующая работа:

- ознакомились и получили опыт создания и применения конкретных информационных технологий и систем для решения реальных задач проектирования и управления программными продуктами;

- обеспечили тесную связь между научно-теоретической и практической подготовкой, создали условия для формирования профессиональных компетенций;

- собрали конкретный материал для выполнения выпускной квалификационной работы в процессе дальнейшего обучения в вузе.

Были выполнены основные задачи производственной практики (научно-исследовательской работы):

- приняли участие в проектах современных фирм-производителей программного обеспечения с учетом профиля подготовки;

- приняли участие в проектах, связанных с использованием современных информационно-коммуникационных технологий организации разработки программного обеспечения;

- изучили новые инструментальные средства;

- подготовили научно-технические отчеты по результатам проведенных работ, выполненных исследований.

Во время практики выполнено индивидуальное задание, проделанная работа отражена в дневнике практики.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Барсегян А. А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining / А.А. Барсегян – Санкт-Петербург: БХВ-Петербург, 284 с. – ISBN 5-94157-991-8. – Текст: непосредственный.
2. Болотова Л.С. Системы и методы искусственного интеллекта: модели и технологии, основанные на знаниях: учебное пособие / Болотова Л.С.– Москва: Финансы и статистика, 663 с. – ISBN 978-5-279-03530- – Текст: непосредственный.
3. Паклин Н. Б. Бизнес-аналитика: от данных к знаниям: учебное пособие / Паклин Н. Б. – Санкт-Петербург: Питер, 704 с. – ISBN 978-5-459-00717-6. – Текст: непосредственный.
4. Матвеев М.Г. Модели и методы искусственного интеллекта: учебное пособие / Матвеев М.Г. – Москва: Финансы и статистика, 448 с. – ISBN 978-5-279-03279-4. – Текст: непосредственный.
5. Практикум по анализу данных на компьютере / И.А. Кацко, Н.Б. Паклин /Под ред. Г.В. Гореловой – Москва: КолосС, 278 с. - 704 с. – ISBN 978-5-459-10391-4. – Текст: непосредственный.

ДНЕВНИК

по производственной практике
вид практики

научно-исследовательская работа

Тип практики в соответствии с ОПОП ВО

Студента Гришунина Андрея Александровича

Фамилия Имя Отчество

1 курса 103М группы

Направление подготовки / Специальность 09.04.04 Программная инженерия

код Наименование

Профиль / Специализация Управление разработкой программных проектов

ЗАПИСИ

о работах, выполненных в период практики

Дата	Содержание/Результаты работы	Отметка о выполнении (выполнено/не выполнено). Замечания руководителя практики
27.01.2022	Получение задания на практику.	выполнено Гришина
03.02.2022	Определение технических характеристик.	выполнено Гришина
21.02.2022	Формирование требований к разрабатываемой системе.	выполнено Гришина
07.03.2022	Анализ инструментов для реализации системы.	выполнено Гришина
04.04.2022	Анализ СУБД для реализации системы.	выполнено Гришина
09.05.2022	Изучение архитектур для реализации системы.	выполнено Гришина
02.06.2022	Составление отчета о проделанной работе.	выполнено Гришина
08.06.2022	Защита отчета.	выполнено Гришина

Руководитель практики

от университета

канд. техн. наук, доцент

Гришина

08.06.2022

О. А. Гущина

подпись, дата

АНКЕТА¹

производственная практика
вид практики

научно-исследовательская работа

Тип практики в соответствии с ОПОП ВО

1 Удовлетворены ли Вы местом прохождения практики? (один ответ)

☒ да ☐ нет

2 Удовлетворены ли Вы качеством разработки методических указаний, содержащихся в программе практики? (один ответ)

☐ в методических указаниях недостаточно информации для составления отчета

☐ методические указания не подходят к профильной организации

☐ не ознакомлен(а) с программой практики

☒ методические указания мне понятны

3 Считаете ли Вы достаточными для выполнения работ, предусмотренных программой практики, те теоретические знания, которые Вы получили в Университете? (один ответ)

☒ да

☐ скорее да, чем нет

☐ скорее нет, чем да

☐ нет

¹ анкета заполняется как в бумажном варианте, так и в электронном виде. Форма интернет-опроса с помощью онлайн-сервиса Google forms размещена по ссылке: <https://forms.gle/LhzEgPos4zFoM1y99>

4 Как Вы оцениваете итоги практики с точки зрения ее результативности?
(один ответ)

- ☒ на практике я еще больше убедился(ась) в правильности выбора профессии
- ☐ практика разочаровала меня в выбранной профессии
- ☐ практика обнаружила пробелы в моей специальной подготовке
- ☐ практика носила формальный характер

5 Оцените степень удовлетворенности местом прохождения практики (материально-техническая оснащенность, кадровый состав) по пятибалльной шкале (1 – очень плохо, 5 –отлично) _____5_____

6 Оцените степень удовлетворенности взаимоотношениями с руководителем практики от Университета по пятибалльной шкале (1 – очень плохо, 5 –отлично) _____5_____

7 Оцените удовлетворенность в целом условиями прохождения практики
(один ответ)

- ☒ полностью удовлетворен(а)
- ☐ удовлетворен(а) частично
- ☐ полностью не удовлетворен(а)

8² Ваши предложения по организации практики:

² при наличии _____