Добрый день уважаемая комиссия, представляю вам проект, который я разработал – «Графовая базы данных для системы управления проектами».

Поскольку каждая из существующих систем управления проектами предлагает свой набор инструментов и сервисов для удобной и эффективной работы над проектами и зачастую в рамках одного проекта используется несколько таких систем, становится актуальной задача их интеграции.

Для интеграции данных о проектах из существующий систем, мною и моим научным руководителем было принято решение использовать NoSQL подход, а именно графовую базу данных. Выбор пал именно на граф ориентированную модель данных, поскольку данные проекта тесно связаны между собой в отношениях и могут углубляться в несколько уровней. Также хотелось отойти от обыденного реляционного подхода где таблицы имеют строго заданные поля и строгую схему. Было необходимо разработать гибкую структуру данных, которая давала бы возможность изменять себя по мере появления требований к интегрированной системе и что немало важно, разработанная база данных должна предоставлять высокую производительность при работе со связанными данными. Для реализации таких задач была выбрана графовая СУБД Neo4j.

**Слайд 2. Цель и задачи ВКР**

При разработке графовой базы данных для системы управления проектами были поставлены следующие задачи:

1. Разработка графовой модели данных для системы управления проектами с целью интеграции данных о проектах из существующих систем;
2. Создание тестовых скриптов для заполнения графовой базы данных;
3. Создание запросов для извлечения информации о проектах с целью проверки работоспособности разработанной модели;
4. Уменьшение времени отклика на запросы пользователей интегрированной системы за счет использования графовой модели данных и СУБД Neo4j.

**Слайд 3. Интегрируемые системы управления проектами**

Исходя из предложенного перечня систем управления проектами, в графовую базу данных интегрируются данные о проектах таких систем как:

1. GitHub – крупнейший веб сервис для совместной разработки проектов основанный на системе контроля версий Git. Сервис является бесплатным для публичных проектов, и насчитывает более 12 миллионов человек участвующих в 31 миллионе проектов;
2. Bitbucket – распределенная система контроля версий, созданная для профессиональных команд. Она упрощает совместную работу команды над проектом благодаря гибким моделям развертывания. Единственное Git-решение, которое хорошо масштабируется и подходит для команд любых размеров;
3. Jira – это коммерческая, настраиваемая система отслеживания ошибок для управления проектами. Может использоваться для поддержки клиентов, поддерживает командную работу в режиме реального времени, предоставляет высокий уровень безопасности, API и множества плагинов. Поддерживает email-интеграцию и русский интерфейс;
4. GitLab – является веб-менеджером репозитория Git, предназначенным для написания кода, его развертывания и тестирования. Обеспечивает управление репозиторием с тонкоструктурным контролем доступа, обзорами кода, отслеживанием ошибок, лентами активности и вики-системами.

Перечисленные системы управления проектами базируются на использовании репозитория файлов Git.

Git – это система контроля версий (VCS) для отслеживания изменений в компьютерных файлах и координации работы с этими файлами среди множества людей.

**Слайд 4. Преимущества графовой СУБД Neo4j**

Поскольку разработка графовой базы данных осуществлялась на СУБД Neo4j, хочу отметить некоторые ее преимущества, а именно:

1. Гибкость, которая дает возможность хранить одни и те же объекты с различными свойствами, что является удобным при интеграции систем;
2. Ощутимый прирост производительности при работе со связанными данными;
3. Современный способ развертывания, как локальный, так и в облаке;
4. Возможность расширения, означает что можно добавлять и дополнять новые виды взаимосвязей, новые узлы, новые метки, а также новые подграфы в существующую структуру и что самое важное – это происходит без нарушения существующих запросов и функционала приложения или системы;
5. Графический интерфейс визуализации данных;
6. Поддержка транзакций ACID – набор свойств гарантирующих надежную работу транзакций: атомарность, согласованность, изолированность, долговечность;
7. Собственный декларативный графовый язык запросов Cypher, который позволяет выразительно и эффективно строить запросы и обновлять графовые хранилища;
8. Облачного сервис «Neo4j Browser Sync». Бесплатный сервис, позволяющий разработчикам сохранять и синхронизировать свои наиболее часто используемые скрипты и настройки.

**Слайд 5. Разработка модели графовой базы данных**

Разработанная графовая база данных для системы управления проектами хранит такие данные как:

1. наименования и описания проектов, их заказчиков и руководителей;
2. наименования и описания задач, подзадач проекта;
3. исполнителей (разработчиков) назначенных на задачи;
4. совершённые изменения и их описание на этапе выполнения задач;
5. URL адреса репозиториев интегрируемых систем в которых были созданы задачи или совершены их изменения на этапе выполнения.

**Слайд 6. Схема разработанной графовой БД**

Схема разработанной графовой базы данных выглядит следующим образом (рисунок схемы).

Вершины графовой базы данных хранят такие данные как:

* ФИО заказчиков (Customer);
* наименования проектов (Project);
* статусы проектов (Project\_status);
* типы проектов (Project\_type);
* коды проектов (Project\_code);
* ФИО руководителей проектов (Master);
* наименования задач (Issue);
* типы задач (Issue\_type);
* приоритеты задач (Issue\_priority);
* статусы задач (Issue\_status);
* ФИО разработчиков (Developer);
* совершённые изменения и их описание (Commit);
* наименования и URL адреса репозиториев интегрируемых систем (Repository).

Отношения между вершинами, обеспечивают общее понимание разработанной графовой модели данных для системы управления проектами. К примеру, вершины Master (руководитель проекта), Project (проект), Issue (задача) и Developer (разработчик) связаны между собой такими отношениями как: Creates (создает), Consists\_of (состоит из), Appoints (назначает) и Executes\_the (выполняет). Следуя схеме, нетрудно понять, что руководитель создает проект и задачи, назначает разработчиков, которые будут выполнять конкретную задачу или задачи и сам проект состоит из задач.

Разработанная таким образом графовая модель, гарантирует извлечение необходимых связанных данных при корректно построенных запросах, позволяет легко отслеживать внесенные изменения при выполнении задач проекта и контролировать состояние проектов в целом. Далее, я привел несколько результатов выполнения различных запросов что бы продемонстрировать работу базы данных.

**Слайд 7. Результаты запросов**

После того как я импортировал созданные тестовые скрипты для заполнения графовой базы данных, первым запросом я решил сделать полную выборку хранящихся данных в базе. Результат вы можете наблюдать на слайде.

**Слайд 8. Результаты запросов**

Вторым запросов я извлек сведения о текущих проектах, хранящихся в базе, а именно такие сведения как: проекты, их заказчики, руководители, задачи и общая характеристика проекта, выраженная в его статусе, коде и типе.

**Слайд 9. Результаты запросов**

Третьим запросов я извлек сведения о конкретной задаче, а именно: назначенного на задачу разработчика, ссылку на репозиторий в котором задача была создана, статус, приоритет и тип задачи.

**Слайд 10. Результат запросов**

При управлении проектами, важной информацией является история изменений задач, то есть как проходит их выполнение и на какой стадии находятся назначенные разработчики. Данный слайд демонстрирует результат запроса на извлечение истории изменений конкретной задачи.

**Слайд 11. Охрана труда**

В виду того, что взаимодействие с разработанной графовой базой данных будут осуществлять системные администраторы, была произведена оценка условий труда системного администратора по факторам тяжести и напряженности трудового процесса. Разработанная графовая база данных для системы управления проектами не меняет класс условий труда системного администратора, но некоторые показатели тяжести труда улучшились за счет уменьшения времени отклика на запросы к базе данных и количества манипуляций при формировании запроса. Разработанная графовая база данных позволила сократить количество стереотипных движений, что привело к экономии времени и соответственно к уменьшению совершаемых движений. Таким образом, улучшились следующие показатели тяжести трудового процесса: физическая динамическая нагрузка с 175 кг × м до 91 кг × м, суммарная масса груза в течении каждого часа рабочей смены с 218,75 кг до 113,75 кг, стереотипные движения с 10 000 до 5 200 движений, статическая нагрузка одной рукой с 875 кг × с до 455 кг × с за рабочую смену. Разработанная графовая база данных для системы управления проектами не повлияла на показатели напряженности трудового процесса.

**Слайд 12. Обоснование экономической эффективности ВКР**

На выполнение выпускной квалификационной работы было затрачено 40 дней.

При расчете стоимости, учитывались расходы на материалы, связь, источники информации и расходы на электроэнергию. Определены расходы на оплату труда дипломника, дипломного руководителя и консультантов. Также были рассчитаны затраты на амортизацию оборудования. Общая стоимость ВКР составила 13 688,91 руб.

Экономия средств при использовании данной разработки составляет 15 000 руб./год.

Период окупаемости разработки составил приблизительно 11 месяцев.

**Заключение**

В заключении хочу еще раз акцентировать внимание на том, что разработанная модель данных предоставляет возможность интегрирования записей из существующих систем управления проектами. Также были реализованы скрипты для заполнения графовой БД и созданы запросы для извлечения основной информации о проектах. Ввиду использования графовой СУБД Neo4j, разработанная база данных обеспечивает высокую производительность и эффективное масштабирование.

Дальнейшее развитие работы предполагает разработку веб-интерфейса и API для интегрирования проектов других систем управления проектами в разработанную графовую базу данных. На этом я завешаю свой рассказ, всем спасибо за внимание!!!