1. Здравствуйте. Я Мамонов Дмитрий, студент 4 курса 5 группы. Тема моей работы - “Разработка расширения для PostgreSQL в виде хранилища графовых моделей”.
2. В современном мире разработки программного обеспечения в качестве вариантов хранилища данных прочно укрепились реляционные СУБД, такие как PostgreSQL, SQL Server, MySQl и т.д. Но реляционная модель подходит не для всех случаев. Например данные, которые основаны на большом количестве связей между собой, будут храниться в подобных СУБД неэффективно, так как работа с ними подразумевает использование тяжеловесных индексных деревьев и поиск по ним. Для этих данных наилучшим вариантом будет хранилище, основанное на модели графа, которое бы успешно интегрировалось с существующими решениями.
3. Основная задача работы - реализовать расширение для СУБД PostgreSQL, представляющее собой хранилище для объектов в виде графов, со следующими возможностями:

* создание графов с возможностью редактирования узлов и ребер
* получение выборки узлов и ребер с помощью декларативного языка запросов
* предоставление SQL-процедур для работы с хранилищем из PostgreSQL

1. Были рассмотрены существующие аналоги - графовая СУБД Neo4j, написанная на Java и расширение для PostgreSQL Apache AGE, которое реализует графовое хранилище используя механизмы реляционной СУБД. Достоинства и недостатки обоих систем приведены на слайде.
2. На слайде приведены инструменты и технологии, используемые при реализации расширения.

* языки программирования C и C++ 17 стандарта
* Стандартная библиотека шаблонов STL
* библиотека для взаимодействия с сервером PostgreSQL libpq
* генератор лексеров flex
* генератор синтаксических парсеров
* система сборки проектов CMake

1. Система состоит из следующих частей: клиентская часть, представленная расширением для PostgreSQL и отдельным интерфейсом командной строки, через который можно делать запросы к хранилищу напрямую: серверная часть, которая включает в себя следующие компоненты:

* парсер языка запросов, состоящий из лексического анализатора, синтаксического анализатора и построителя абстрактного синтаксического дерева входящего запроса;
* менеджер графовых сущностей, каждый компонент которого отвечает за управление определенными объектами графа или его составляющими в соответствующем файле на диске.

Все представленные компоненты архитектуры системы были реализованы на языке программирования C++ с помощью STL. Расширение было написано на языке C с помощью библиотеки libpq.

1. В качестве языка запросов используется декларативный язык Cypher, являющийся основным для Neo4J и ставший прародителем для современного стандарта языка графовых запросов GQL. Запрос состоит из типа запроса (синий цвет), выбираемых узлов (фиолетовый цвет), типа отношения (красный цвет) и действия с выбираемыми данными (зеленый цвет). На слайде приведена выборка по типу отношения.
2. На данном слайде представлен запрос на создание нового графа.
3. На текущем слайде приведен синтаксический разбор запроса и построение абстрактного синтаксического дерева, основные узлы которого представляют отдельные составляющие запроса, перечисленные выше.
4. Здесь представлена диаграмма классов узлов аст-дерева для программы Bison (смежные синтаксические группы выделены одинаковым цветом).
5. На слайде приведена диаграмма классов частей графа, хранящихся в файлах. Каждая сущность представляет собой структуру определенного размера, так как все поля являются целочисленными.
6. Все сущности хранятся непрерывно в файле в том порядке, в котором они были созданы. Зная размер и номер сущности, можно вычислить смещение, откуда необходимо начинать чтение из файла. На слайде представлено расположение каждой структуры в файле.
7. Для быстрого поиска и обхода по графу используется построение ребер графа в виде двусвязного списка. Сначала для каждого узла определяется его первое свойство и первое ребро по списку, которое с ним соединено.
8. После этого для каждого узла в ребре ищутся первые попавшиеся ребра, в которых узел задействован. Если такие найдены, то между ними устанавливаются двунаправленные связи. В итоге получается множественный двусвязный список. Преимущество такого варианта хранения заключается в быстром поиске соседей для определенного узла и реализации свободного индекса для перемещения между ребрами.
9. На слайде представлена диаграмма классов для файлового хранилища, которое оперирует всеми сущностями. Приведены классы для хранилища узлов графа, его ребер, свойств узлов и общего строкового хранилища, куда сохраняются все наименования, части свойств и метки узлов и ребер.
10. Расширение для СУБД PostgreSQL состоит из файла с хранимыми процедурами, написанными на одном из языков серверного программирования (в основном pl/pgSQL). Для более углубленного взаимодействия с сервером базы данных используется интерфейс SPI на языке C. Пользовательский код, написанный на C, динамически загружается в сервер и связывается через этот интерфейс с SQL-процедурами.
11. На слайде представлена распечатка в командную строку AST-дерева запроса, построенная при помощи генераторов flex и bison.
12. Здесь приведено взаимодействие с графовым хранилищем напрямую через специальный интерфейс командной строки, поставляемый вместе с сервером.
13. На слайде представлена работа с графовыми моделями через SQL-функции, которые предоставляет расширение.
14. Таким образом, в результате выполнения работы было реализовано расширение для СУБД PostgreSQL, представляющее собой хранилище для объектов в виде графов, со следующими возможностями:

* создание графов с возможностью редактирования узлов и ребер
* получение выборки узлов и ребер с помощью декларативного языка запросов
* предоставление SQL-процедур для работы с хранилищем из PostgreSQL

Спасибо за внимание.