МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ "ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ"

Кафедра программирования и информационных технологий

Разработка расширения для PostgreSQL в виде хранилища графовых моделей

Выполнил:

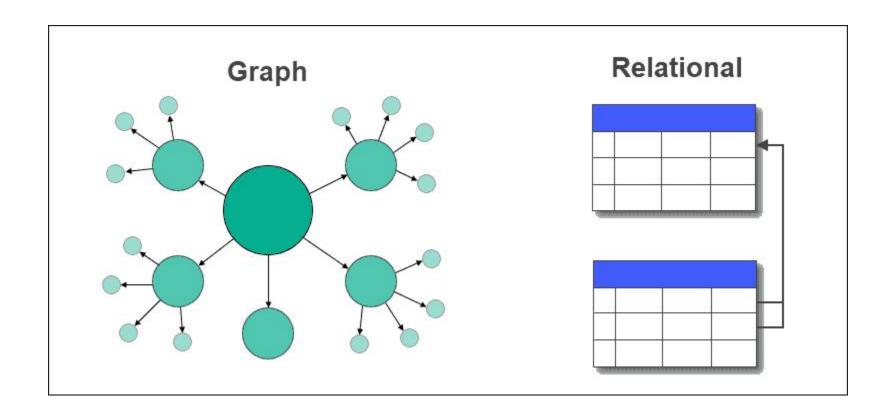
Научный руководитель:

Зав. кафедрой:

Студент Мамонов Д.В. Самойлов Н.К. ст. преподаватель С.Д. Махортов, д.ф.-м.н, профессор

Воронеж 2024

Введение



Постановка задачи

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

Существующие аналоги

Neo4j

Достоинства:

- гибкость модели данных
- простота запросов
- масштабируемость

Недостатки:

- цена
- сложность администрирования
- тяжеловесность из-за среды JVM

Apache AGE

Достоинства:

- гибкость и функциональность
- совместимость с PostgreSQL
- производительность запросов

Недостатки:

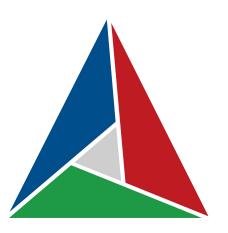
- отсутствие index-free
- сложность настройки и использования

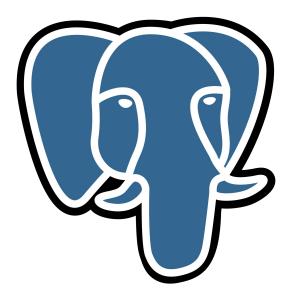
Средства реализации

- C/C++17
- C++ STL
- libpq
- flex
- bison
- CMake

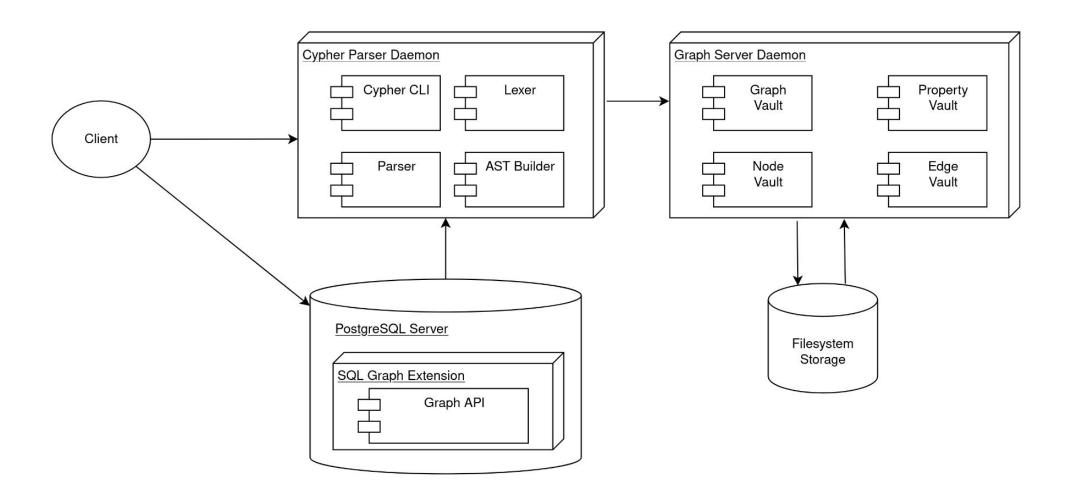






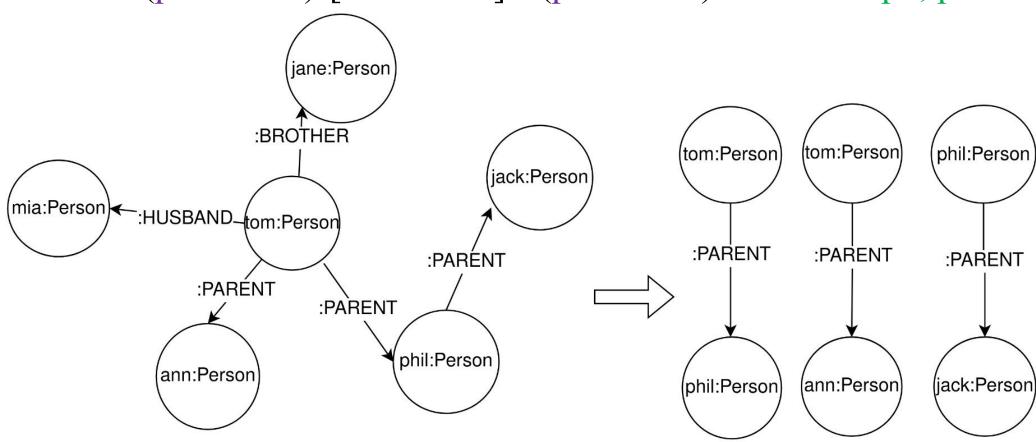


Архитектура



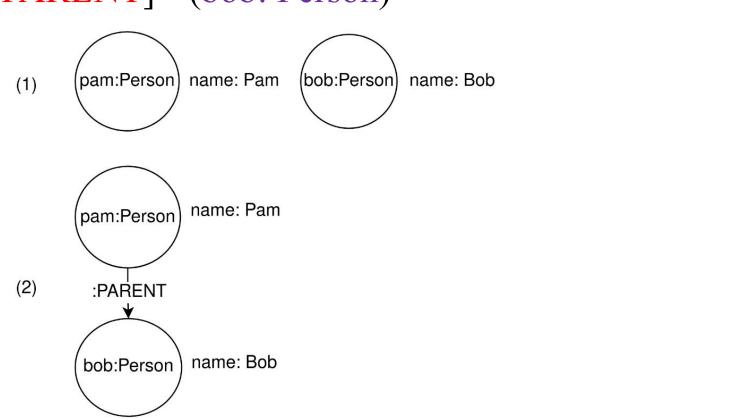
Язык запросов Cypher

MATCH (p1:Person)-[:PARENT]->(p2:Person) RETURN p1, p2



Язык запросов Cypher

CREATE pam: Person{name: "Pam"}, bob: Person{name: "Bob"}, (pam: Person)-[:PARENT]->(bob: Person)



Пример AST-дерева для запроса

MATCH FROM graph_name res = (p1:Person{name: "Pam"})-[p:PARENT]->(p2:Person{name: "Bob"}) RETURN res;

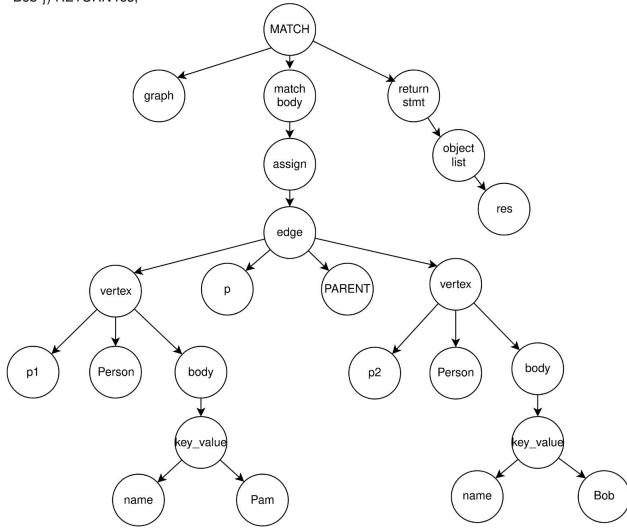


Диаграмма классов AST-дерева

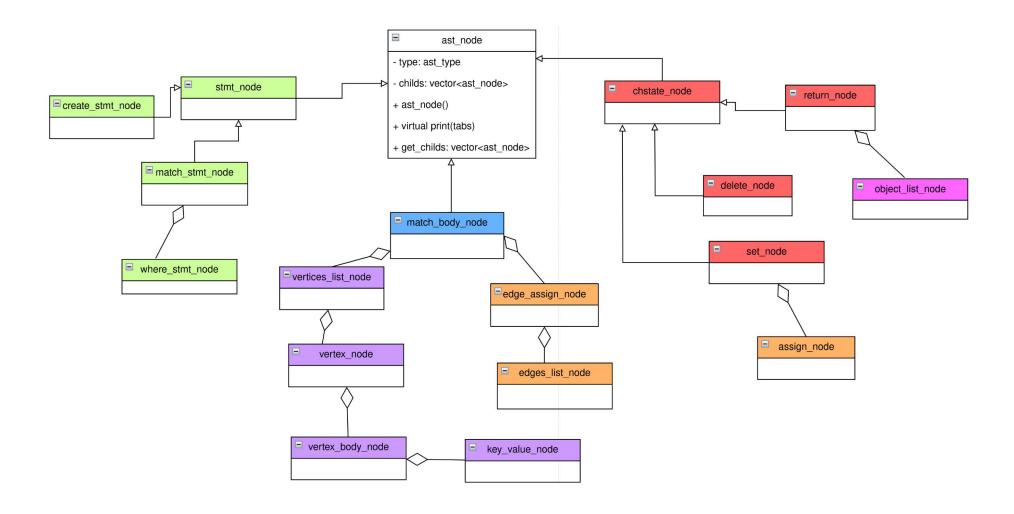
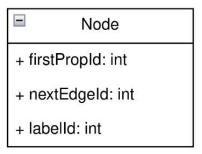


Диаграмма классов сущностей графа

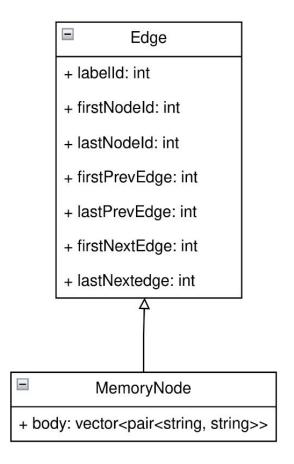


Property

+ keyld: int

+ valueld: int

+ nextPropld: int



Представление сущностей в файлах

Node: 12 bytes

firstPropId: 4 bytes nextEdgeId: 4 bytes labelId: 4 bytes

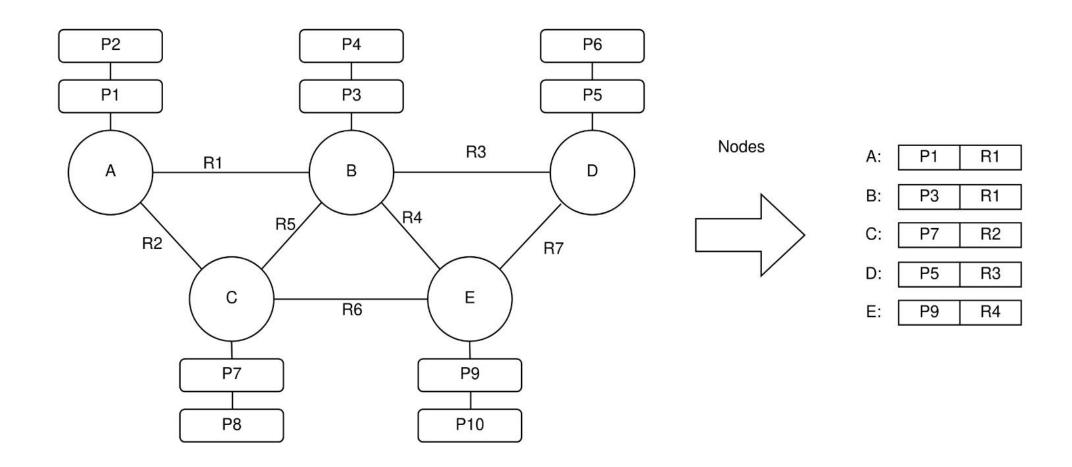
Property: 12 bytes

keyld: 4 bytes valueld: 4 bytes nextPropld: 4 bytes

Edge: 28 bytes

labelld	firstNodeld	lastNodeld	firstPrevEdge	lastPrevEdge	firstNextEdge	lastNextEdge
---------	-------------	------------	---------------	--------------	---------------	--------------

Алгоритм построения графа: узлы



Алгоритм построения графа: ребра

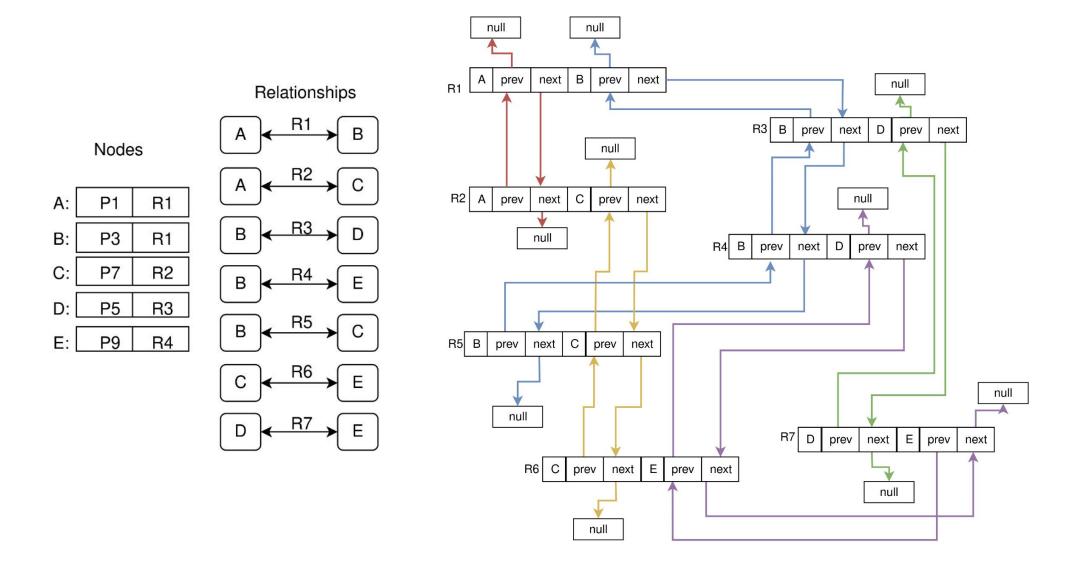
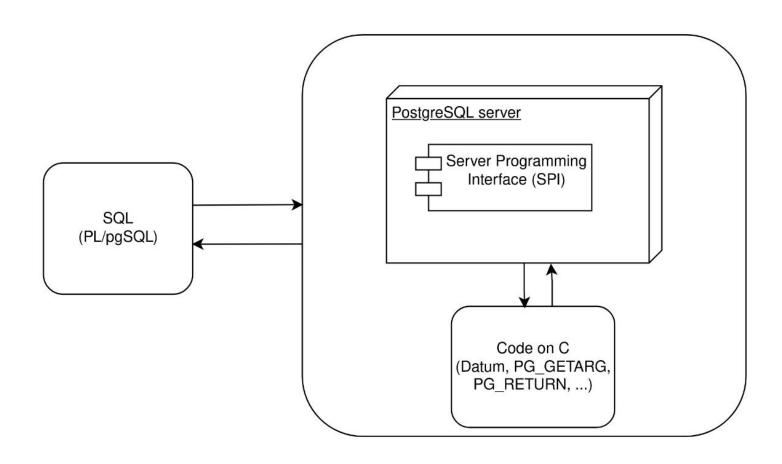


Диаграмма классов хранилища графа

Graph NodeEdgeVault localNodes : vector<MemoryNode> - nodesFile : FILE* localEdges : vector<Edge> **PropVault** - edgesFile : StringVault propVault : PropVault - file : FILE* + NodeEdgeVault(nodeFilename, nodeVault : NodeEdgeVault - vault : StringVault edgeFilename) currentNodeId: int currPropId: int + findNodeById(int id): Node - currentEdgeId : int + PropVault(propFilename, **StringVault** + writeNodeToFile(int id, Node n) - labelVault : StringVault strFilename) - file : fstream + findEdgeById(int id): Edge + Graph(strFilename, propFilename, + isOpen(): bool - db : map<int, string> nodeFilename, edgeFilename) + writeEdgeToFile(int id, Edge e) + currentld(): int - curr id: int + getNodeById(id): MemoryNode + getPrevForEdgeAndNode(nodeld, + getPropById(int id): Property edgeld): Edge + getEdgeByld(id): Edge + StringVault(filename) + getPropsById(int id): + getLastForEdgeAndNode(nodeld, + appendNewNode(vertex, label): vector<Property> + isVaultOpen(): bool edgeld): Edge MemoryNode + setProp(int id, string key, string + currentId(): int value) + appendNewEdge(first, last, label): Edge + findFirstEdgeForNode(nodeld, + getStrById(int id): string + writePropBody(vector<Property> *outEdgeId): Edge + updateNodeBody(id, node) properties) + setStr(int id, string value) + findLastEdgeForNode(nodeld, + deleteNode(id) + appendNewProp(string key, string + append(string value) *outEdgeld): Edge value) + deleteEdge(id)

Архитектура расширения PostgreSQL



Разбор запроса (построение AST)

```
MATCH FROM graph (p1:Person)-[p:PARENT]->(p2:Person) WHERE p2.name = "Bob" RETURN p1, p2;
MATCH query:
  -- object value: graph
  -- match query body:
    -- edge assign statement:
      -- edges list:
        -- edge:
          -- object value: p
          -- label value: PARENT
          -- vertices list:
            -- vertex:
              -- object value: p1
              -- label value: Person
          -- vertices list:
            -- vertex:
              -- object value: p2
              -- label value: Person
  -- where statement:
  -- assign:
    -- object value: p2
   = "Bob"
   -- change statement:
    -- return statement:
      -- object list:
        -- object value: p2
        -- object value: p1
```

Cypher CLI

```
Graph Cypher CLI (version 1.0.0)
> CREATE GRAPH persons (pam: Person{name: "Pam"}, bob: Person{name: "Bob", age: 19}), ((pam: Person)-[p:PARENT]->(bob: Person));
CREATE query
(pam:Person) - [PARENT] -> (bob:Person)
> MATCH FROM persons (pam: Person) RETURN pam;
MATCH query
pam:Person {
    name: Pam
> MATCH FROM persons (:Person)-[p:PARENT]->(:Person) RETURN p;
MATCH query
(pam:Person) - [PARENT] -> (bob:Person)
> EXIT
```

Выполнение запросов в PostgreSQL

```
psql (15.4, cepsep 14.10)
Введите "help", чтобы получить справку.
postgres=# SELECT * FROM cypher("new_graph", "
CREATE (:Person {name: "Zeus"})-[:FATHER_OF]->(:Person {name: "Persey"})
") AS (f);
 a
(0 rows)
postgres=# SELECT * FROM cypher("new_graph", "
MATCH (p1:Person)-[:FATHER_OF]->(p1:Person) RETURN p1, p2
") AS (p);
   | object | name
       p1 | Zeus
       p2
           Persey
(2 rows)
```

postgres=#

Заключение

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

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ "ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ"

Кафедра программирования и информационных технологий

Разработка расширения для PostgreSQL в виде хранилища графовых моделей

Выполнил:

Научный руководитель:

Зав. кафедрой:

Студент Мамонов Д.В. Самойлов Н.К. ст. преподаватель С.Д. Махортов, д.ф.-м.н, профессор

Воронеж 2024