Оглавление

[Цель работы 2](#_Toc164257343)

# Цель работы

* Изучить модель представления данных и способы работы с графовыми БД;
* Освоить методы создания графовой БД и языки запросов к ней;
* Получить навыки работы с графовой СУБД Neo4j.

# Задание

OrientDB

* 1. Краткое описание, категория, особенности + ссылки
  2. Описание модели данных, система типов + ссылки
  3. Функциональные возможности + ссылки
  4. Языки запросов + примеры DDL, DML (по основным возможностям) + ссылки

# Ход работы

## Краткое описание, категория, особенности + ссылки

OrientDB - это система управления базами данных NoSQL с открытым

исходным кодом, написанная на Java. Это мультимодельная база данных,

поддерживающая графическую, документальную, ключ/значение и объектную

модели. Он поддерживает режимы без схемы, с полной схемой и со смешанной

схемой. Он имеет систему профилирования безопасности, основанную на

пользователях и ролях, и поддерживает запросы с помощью Gremlin наряду с

расширенным SQL для обхода графа [21].

категория

Многомодельная база данных — это база данных, предназначенная для

поддержки нескольких моделей данных в одной системе хранения данных. Это

означает, что такая система может хранить, индексировать и запрашивать

данные в нескольких моделях. Этот тип базы данных обеспечивает единый

интерфейс для обеспечения согласованности, безопасности и доступа к

данным, а также устраняет необходимость сложных преобразований и

миграций между различными базами данных [1].

Особенности

Многомодельность: OrientDB поддерживает графовую, документную и объектную модели данных.

Безсхемность, схематичность и смешанный режимы работы.

Механизмы индексации на основе B-дерева и Extendible hashing.

Быстрый обход отношений один-ко-многим и быстрая добавление/удаление связей.

Поддержка RB+Tree (красно-черное дерево) и B+-Tree (B-дерево)

## Описание модели данных, система типов + ссылки

Описание модели данных

В настоящее время существует четыре основные модели баз данных:

ключ-значение, семейство столбцов, документальные и графовые [2].

Базы данных ключ-значения имеют хранилище ключ-значение. Они

позволяют разработчику приложения хранить данные без схемы. Эти данные

обычно состоят из строки, которая представляет ключ, и фактических данных,

которые считаются значением в отношениях «ключ-значение». Сами данные

обычно представляют собой своего рода примитив языка программирования

(строка, целое число, массив) или объект, который создаётся привязками

языков программирования к хранилищу ключ-значение. Это заменяет

необходимость в фиксированной модели данных и делает строгими требования

к правильному форматированию данных без данных [3].

Графовая модель базы данных — это модель, в которой структуры

данных для схемы и/или экземпляров моделируются как направленный,

возможно, помеченный граф или обобщение структуры данных графа, где

манипулирование данными выражается с помощью графо-ориентированных

операций и конструкторов типов, а соответствующие ограничения целостности

могут быть определены в структуре графа [4].

База данных семейства столбцов - это база данных NoSQL, которая

хранит данные с использованием столбцового подхода, в отличие от

реляционных, которые упорядочивают данные по строкам. Данные, хранящиеся

4

в базе данных семейства столбцов, выбираются вертикально, что делает

частичное чтение более эффективным, поскольку загружается только набор

атрибутов строки [5].

Вместо хранения данных в фиксированных строках и столбцах базы

данных документов используют гибкие документы. Документ – это запись в

базе данных документов. Документ обычно хранит информацию об одном

объекте и любых связанных с ним метаданных. Документы хранят данные в

парах поле-значение. Значения могут быть различных типов и структур,

включая строки, числа, даты, массивы или объекты

## Функциональные возможности + ссылки

Батч

Кластеры

Графы

Индексы

## Языки запросов + примеры DDL, DML (по основным возможностям) + ссылки

В OrientDB таблицы представляются в виде классов, строки - записи в

таблице, а свойства класса - поля. В основном OrientDB использует

документальную модель, но также можно использовать графовую модель.

Для создания классов используется следующая команда:

CREATE CLASS <class\_name>

Для добавления полей в класс, можно использовать следующую команду:

INSERT INTO <class\_name> (<field\_name>) VALUES (field\_value)

Для удаления записей используется команда DELETE:

DELETE FROM <target-name>

Для изменения значения используется UPDATE:

UPDATE Profile SET nick = 'Luca'

Для выбора результатов используется SELECT [22]:

SELECT name, age FROM Account

Так же написать про графы и что там ещё в доке

CREATE CLASS

ALTER CLASS

DROP CLASS

CREATE PROPERTY

CREATE DATABASE

ALTER DATABASE

ALTER PROPERTY

DROP PROPERTY

CREATE CLUSTER

ALTER CLUSTER

# Список используемой литературы

1. Neo4j – Текст. Изображение: электронные // Neo4j: [сайт]. – URL: Дистрибутив и документация – https://neo4j.com/ (дата обращения: 15.12.2018)
2. Neo4j Docs: [сайт]. – URL: https://neo4j.com/docs/cypher-manual/current/clauses/remove/ (дата обращения: 01.05.2022)