

УДК 004.658

## ПРЕИМУЩЕСТВА ИСПОЛЬЗОВАНИЯ ГРАФОВЫХ БАЗ ДАННЫХ ПРИ РАЗРАБОТКЕ ПРИКЛАДНЫХ ПРИЛОЖЕНИЙ

**К. К. Отраднов**, старший преподаватель кафедры телекоммуникаций, институт радиоэлектроники и информатики, ФГБОУ ВО «МИРЭА – Российский технологический университет», Москва, Россия;  
e-mail: const.otradnov@yandex.ru

**А. С. Алёшкин**, к.т.н., доцент кафедры КБ-3 «Безопасность программных решений», институт кибербезопасности и цифровых технологий, ФГБОУ ВО «МИРЭА – Российский технологический университет», Москва, Россия;  
e-mail: antony@testor.ru

**В. Н. Калинин**, педагог ДО, кафедры прикладной информатики и интеллектуальных систем в гуманитарной сфере, РУДН, Москва, Россия;  
e-mail: vkalininz@mail.ru

*Приведен обзор достоинств графовых баз данных, показаны различия между графовыми и реляционными базами данных, показано использование графовых баз данных в различных отраслях (в том числе для анализа медианпространства и анализа сетей транспорта), а также показаны используемые для реализации программных решений библиотеки и отражена модель (диаграмма) реляционной и графовой баз данных, необходимые для анализа данных социальной сети.*

**Ключевые слова:** графовые базы данных, проектирование графовых баз данных, Python, Neo4j, социальные сети, транспортные сети.

**DOI:** 10.21667/1995-4565-2023-86-73-85

### Введение

В настоящее время существует множество различных видов баз данных, каждый из которых имеет свои преимущества и недостатки. Одними из относительно новых подходов к проектированию баз данных являются подходы с использованием графовых баз данных (БД). Они отличаются от реляционных БД тем, что данные представляются в виде связанного графа, где узлы представляют данные об объектах, а ребра представляют данные о связях между ними. В этой статье мы рассмотрим преимущества графовых БД перед реляционными БД, а также продемонстрируем примеры их применения в различных отраслях.

### Теоретическая часть

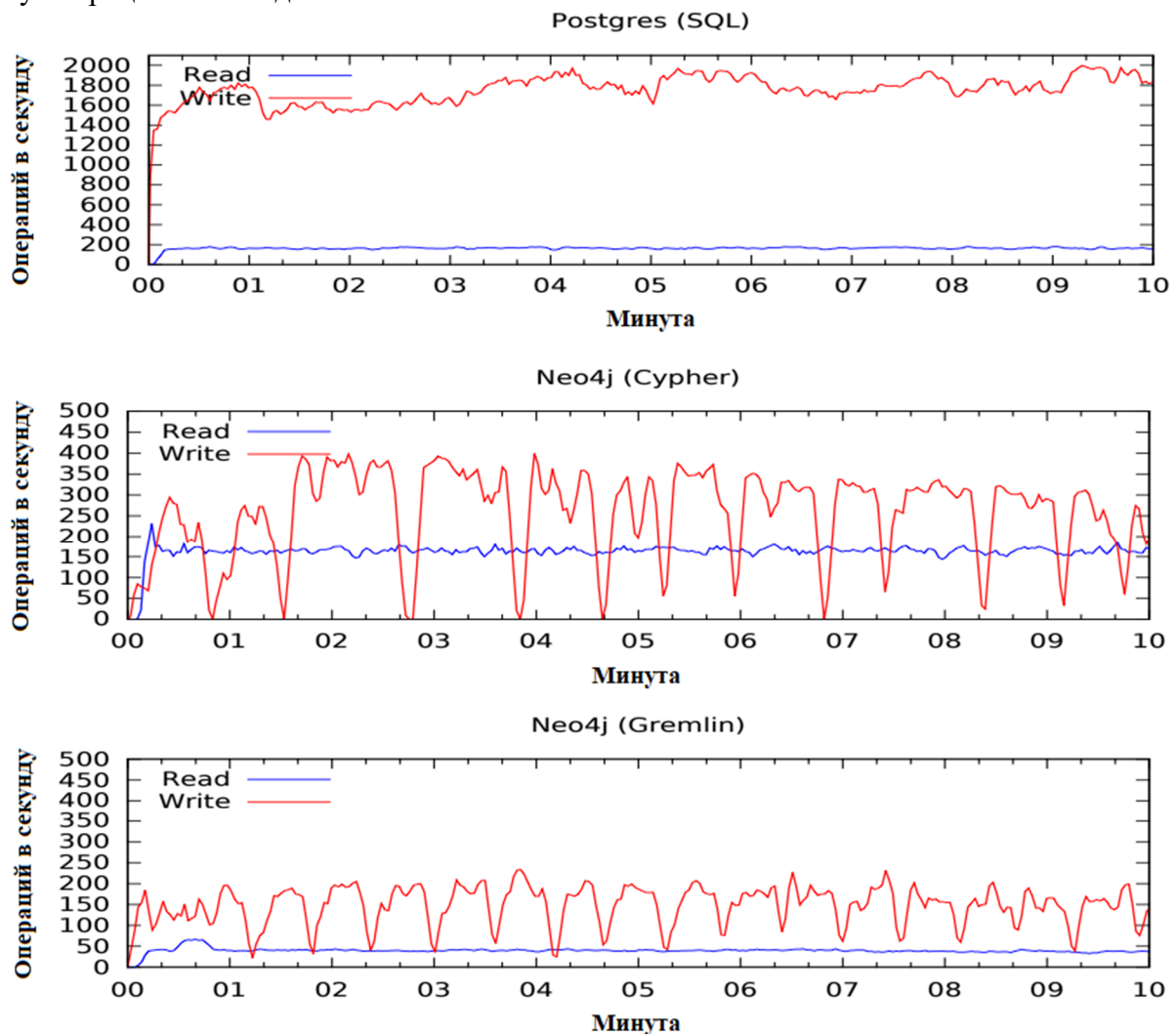
В работе [1] авторы показывают, что графовые БД предоставляют эффективный способ хранения и анализа сложных клинических данных, позволяя организовать такие данные в виде графа, и предоставляют возможность использовать методы работы с графами для обучения и аналитики, что имеет важное значение для медицинской отрасли.

Кроме того, рассматривая возможности графовых БД, авторы в работе [4] отмечают, что данные из реального мира могут быть представлены в виде графа, и это предоставляет множество возможностей для проведения исследований и анализа таких данных.

Более того, графовые базы данных становятся все более важными и распространенными в связи с развитием технологий, а также в связи с повышающейся сложностью современных сетей данных, таких как Интернет вещей. Это подтверждается работой [5], где авторы показывают, что графовые базы данных благодаря своей способности представления данных в виде графов, а также их гибкости в адаптации к изменяющимся структурам данных получают широкое распространение в сравнении с традиционными таблицами.

С развитием онлайн-социальных сетей возникает растущая потребность в обработке графовых данных, но вопрос о замене реляционных БД на графовые БД остается открытым.

Сравнительный анализ приложений, использующих специализированные графовые БД, провели авторы статей [3, 10]. В статье [3] была приведена архитектура бенчмаркинга, которая представляет собой попытку сравнить производительность различных систем для работы с графовыми данными, и результаты такого сравнения показывают, что специализированные графовые базы данных не всегда обеспечивают существенное преимущество. На рисунке 1 представлены данные по бенчмаркингу реляционной базы данных PostgreSQL и графовой базы данных Neo4j (с языком запросов Cypher и с языком запросов Gremlin). Предложенная модель экспериментальной оценки графовой базы данных Neo4j получает развитие в работах [12, 13]. Исходя из показанных данных, реляционная БД на базе СУБД PostgreSQL показывает сравнимое число операций чтения в единицу времени и отличается в большую сторону по числу операций записи данных.



**Рисунок 1 – Бенчмаркинг (Neo4j и Postgres)**  
**Figure 1 – Benchmarking (Neo4j and Postgres)**

Рассмотрев работу [9], можно сделать вывод о том, что выбор между графовыми базами данных, такими как Neo4j и OrientDB, зависит от характеристик приложения и сложности данных. Возможность индексации данных играет важную роль в ускорении запросов в графовых БД, и она может существенно влиять на производительность. При сравнении СУБД Neo4j и OrientDB было выявлено, что каждая из них имеет свои преимущества и недостатки, а конкретный выбор должен зависеть от применимых требований и обрабатываемого объема данных (сравнение представлено в таблице).

**Таблица 1 – Сравнение Neo4j и OrientDB**  
**Table 1 – Comparison of Neo4j and OrientDB**

Глубина	OrientDB	Neo4j
1	0,010	0,010
2	0,010	0,018
3	0,072	0,135
4	2,544	1,600

В работе [8] авторы проводили исследование в решении проблемы неэффективных запросов в графовых БД и предложили использовать новую систему выделения идентификаторов (ID) на основе графовых БД, используемых в качестве кэша. При этом результаты запросов не хранятся в памяти, а идентификаторы результатов используются для улучшения производительности базы данных и экономии памяти. Технология выделения ID и их быстрого использования в запросах вносит существенное улучшение производительности, особенно при обработке больших объемов данных.

В работе [7] авторами было предложено три новых решения для поиска подграфов (подвыборка данных) в больших БД малых графов с использованием метода "Фильтр-Затем-Проверка" (FTV). Новые решения основаны на существующих методах GGSX и CT-Index, которые считаются передовыми, что позволяет осуществлять быструю навигацию и выборку по узлам в графовой базе данных.

Ценная мысль работы [6] заключается в том, что, исследуя внутренние ограничения целостности в графовых БД, которые возникают при концептуальном моделировании, можно добавить механизмы контроля, которые позволяют поддерживать на уровне СУБД проверки ссылочной целостности. Графовые БД часто не имеют явных конструкций для определения таких ограничений целостности, и авторы предлагают расширение языка запросов для графовых баз данных, чтобы удовлетворить эти требования. Они также обращают внимание на важную особенность графовых баз данных — свободу от индексации, что облегчает нативную обработку данных.

Многие авторы отмечают что преимущества использования графовых БД можно выделить во многих областях, где графовые базы данных не только не уступают реляционным базам, но и в несколько раз могут превосходить их по различным параметрам, причем использование графовых баз данных является довольно распространённой практикой. Так, графовые БД могут использоваться в области биомедицины [2] или же для комплексного сетевого анализа [11].

На текущий момент решения, поддерживающие графовые представления данных, активно развиваются, и их преимущества заключаются в эффективном представлении, а также в обработке связей, гибкости в моделировании и представлении данных, и, что зачастую критично, в высокой производительности при выполнении операций над графовыми данными

### **Обзор графовых баз данных и их преимущества**

Графовые базы данных (Graph Databases) представляют собой специализированный тип баз данных (относящийся к типу NoSQL), который обеспечивает эффективное оперирование, моделирование, визуализацию и хранение данных в виде графовой структуры, состоящей из узлов (вершин) и рёбер (связей) между ними.

Графовые базы данных предоставляют мощный инструмент для работы с данными, ориентированными на связи, что обуславливает их широкое применение в различных отраслях для решения сложных задач анализа и моделирования.

Одним из основных различий между графовыми и реляционными базами данных является способ организации данных. В реляционных БД данные организованы в виде таблиц, где каждая строка представляет отдельную запись, а каждый столбец - отдельное свойство этой записи. Связи между таблицами устанавливаются с помощью внешних ключей.

В графовых БД данные организованы в виде графа, где узлы представляют объекты, а ребра – связи между ними. Каждый узел может иметь несколько свойств, а каждое ребро может иметь свойства, описывающие характер связи между узлами.

Более того, главное отличие заключается в способе обработки запросов. В реляционных БД запросы выполняются с помощью языка SQL, который позволяет выбирать данные из таблиц, объединять таблицы и выполнять другие операции. В графовых БД запросы выполняются с помощью языка запросов к графу, который позволяет выбирать узлы и ребра, а также выполнять различные операции над ними (например, переход к узлу-потомку или переход в начало ребра).

Кроме того, одним из преимуществ графовых баз данных перед реляционными является возможность эффективно работать с данными, в которых задана сложная или же неоднородная структура связей. Например, графовые базы данных могут использоваться для анализа социальных сетей, где каждый узел представляет пользователя, а ребра – какие-то связи между ними. Другим примером является использование в области логистики и транспорта, где графовые данные могут применяться для оптимизации маршрутов и управления логистическими сетями. Особенно интересны и эффективны приложения графовых БД к случаям, когда данные содержат сложные и взаимосвязанные структуры [14].

Область применения графовых баз данных обширна, они используются:

- 1) для моделирования социальных сетей и анализа связей между пользователями, друзьями, подписчиками, сообществами и т.д.;
- 2) для построения персональных рекомендаций, формируемых рекомендательными системами, выполняемых на основе анализа предпочтений (предыдущей истории) и интересов пользователя;
- 3) для оптимизации маршрутов доставки, управления поставками (логистика) и анализа сетей транспорта;
- 4) для моделирования биологических сетей, молекулярных взаимодействий и генетических структур;
- 5) для моделирования и анализа связей между устройствами и событиями в сетях IoT.

### **Этапы проектирования графовых баз данных**

Проектирование графовых баз данных (Graph Database Design) является процессом создания базовой структуры и организации данных для эффективного хранения, доступа и анализа данных в графовом представлении. На этапе проектирования пользователь описывает граф, в котором задаёт основные сущности (которые будут организованы как узлы в графовой БД) и задаёт возможные рёбра - связи между узлами.

Проектирование графовой базы данных состоит из следующих этапов.

1. Определение сущностей и связей: здесь выполняется выбор узлов в моделируемом графе, определяются свойства для таких узлов. Это могут быть персоналии (ФИО, дата рождения, регион проживания и т. д.), места (координаты, описания, фотографии и т. д.), продукты (название, EAN13, код, цена и т. д.), события (источник, дата, описание, автор и т. д.) и другие варианты. Определяются связи (рёбра) между сущностями и характеристики таких связей. Например, для социальной сети связями могут быть "дружба", "подписка" и т. п. Для связи персоналии и продукта могут быть указаны дата покупки, персональная скидка, номер чека и другие данные.

2. Разработка схемы графа: здесь выполняется создание схемы графа, определение типов узлов и связей, фиксируются типы данных для атрибутов (свойств). Необходимо определить, какие атрибуты будут храниться в узлах, а какие — в связях, для того чтобы оптимизировать доступ к этим данным в последующих запросах.

3. Учет запросов и использования: выполняется построение типичных запросов к построенной графовой БД и моделируются операции, которые будут выполняться в графовой БД.

Данный этап помогает в том числе определить, какие индексы и структуры данных нужны для оптимизации запросов.

### Отличия в проектировании реляционных БД и графовых БД

Как говорилось ранее, графовые базы данных и реляционные базы данных отличаются основными структурами данных и способами организации получения информации.

Рассмотрим процесс проектирования базы данных для сбора данных из социальной сети. В социальной сети присутствуют такие объекты, как:

- 1) сообщество;
- 2) пост;
- 3) комментарий;
- 4) пользователь.

Кроме того, важно помнить о том, что:

- 1) сообщество состоит из различного количества постов;
- 2) под постами пишут комментарии;
- 3) один пользователь может написать сколько угодно комментариев;
- 4) пользователь может "дружить" (быть подписчиком) с другим пользователем.

Такие данные при проектировании в реляционной структуре БД порождают создание четырёх таблиц для хранения каждого типа объекта социальной сети, а также пятую промежуточную таблицу для организации взаимосвязи типа "друг" между отдельными пользователями с типом связи «многие-ко-многим».

Диаграмма построенной реляционной БД представлена на рисунке 2.

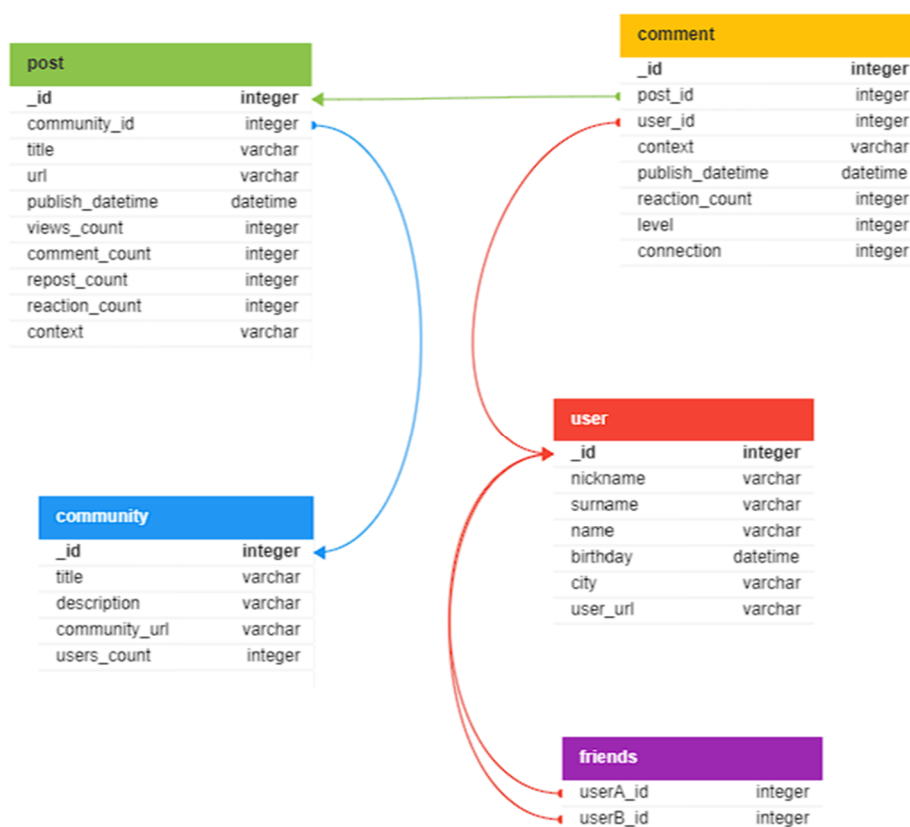


Рисунок 2 – Проектирование реляционной базы данных для сбора данных из социальной сети  
Figure 2 – Designing relational database to collect data from social network

В отличие от реляционной БД для графовой БД нам необходимо определить:

- 1) типы объектов (узлы) и их метаданные;

2) взаимосвязи узлов между собой.

Для наших возможных данных определим 4 типа узлов (рисунок 3) и зафиксируем их данные (для графовых БД жесткая структура по составу полей не является обязательной, но приветствуется, так как позволяет унифицировать хранимые данные):

- 1) узел – сообщество;
- 2) узел – пост;
- 3) узел – комментарий;
- 4) узел – пользователь.

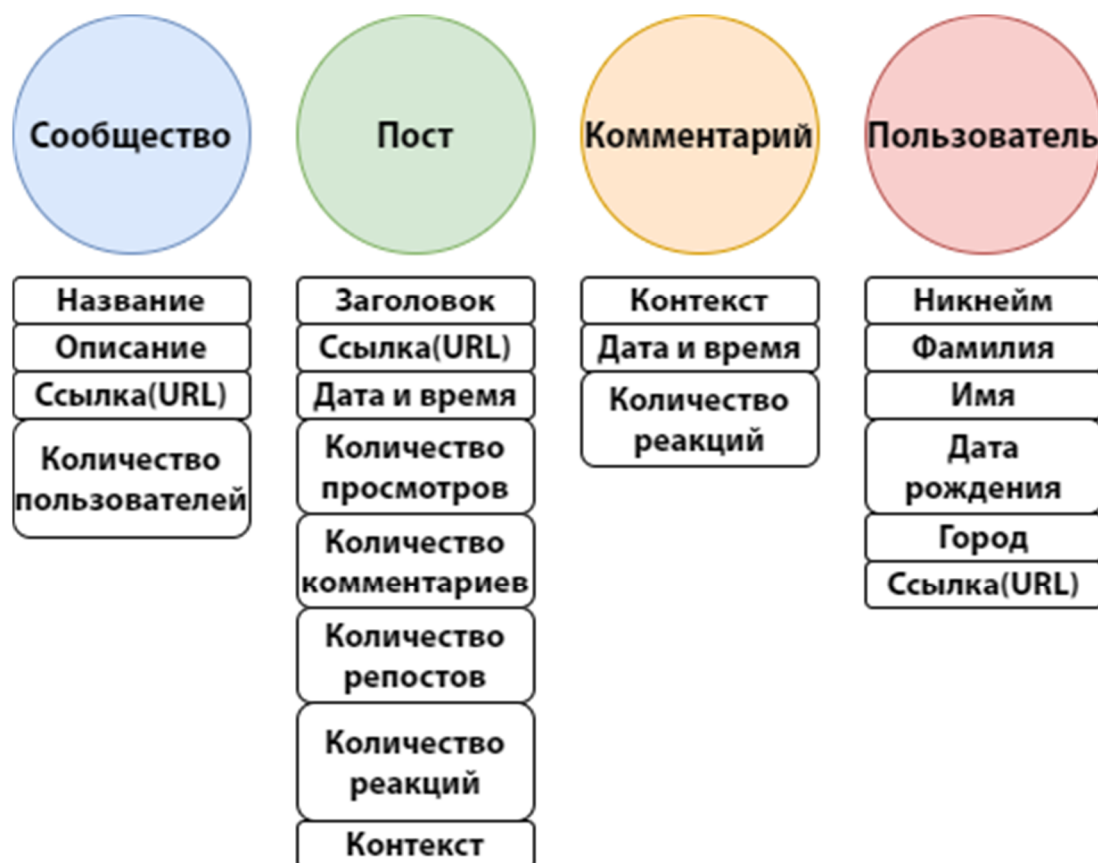


Рисунок 3 – Типы объектов и их метаданные в графовой базе данных  
Figure 3 – Object types and their metadata in a graph database

Также определяется 4 типа взаимосвязей (рисунок 4):

- 1) сообщество – пост;
- 2) пост – комментарий;
- 3) комментарий – пользователь;
- 4) пользователь – пользователь.

Результатом моделирования структуры данных для социальной сети станет граф, представленный на рисунке 5.

Отметим, что формирование данных в графовом виде позволяет нам получить некоторые дополнительные свойства данных, которых не было в реляционной структуре (которые мы могли бы получить, дополнительно усложнив реляционную структуру данных). Например, мы можем связать один и тот же пост с несколькими сообществами, отражая в данных такое понятие как «репост», или дублирование сообщений. Точно так же мы можем выделять полностью идентичные комментарии и таким образом находить все посты всех сообществ (и пользователей, оставивших такие комментарии), где комментарии повторяются, то есть за-дублированы.

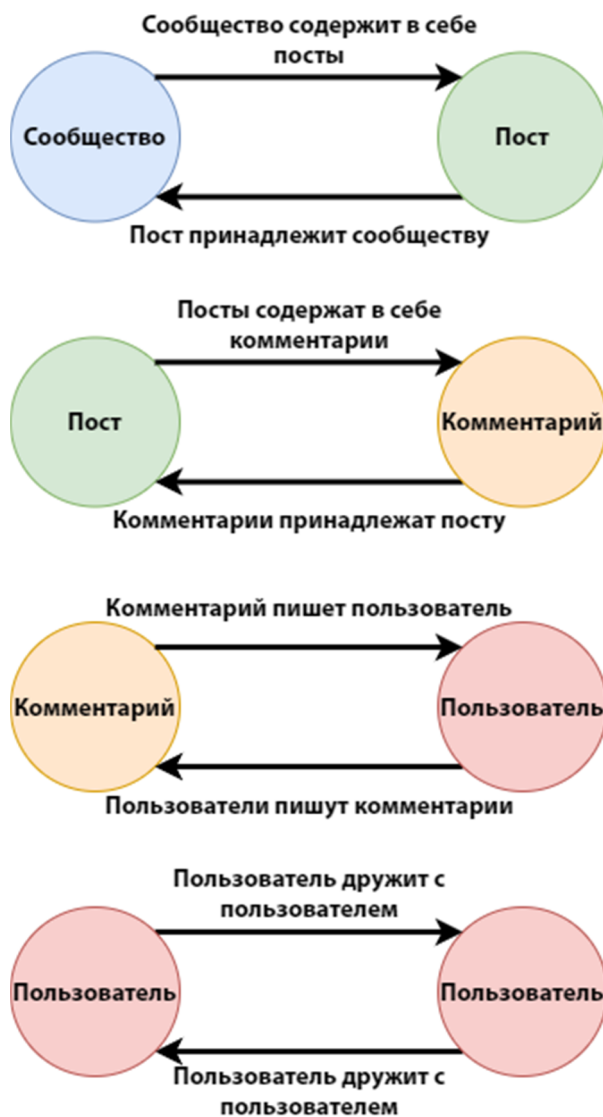


Рисунок 4 – Взаимосвязи в графовой базе данных  
Figure 4 – Relationships in a graph database

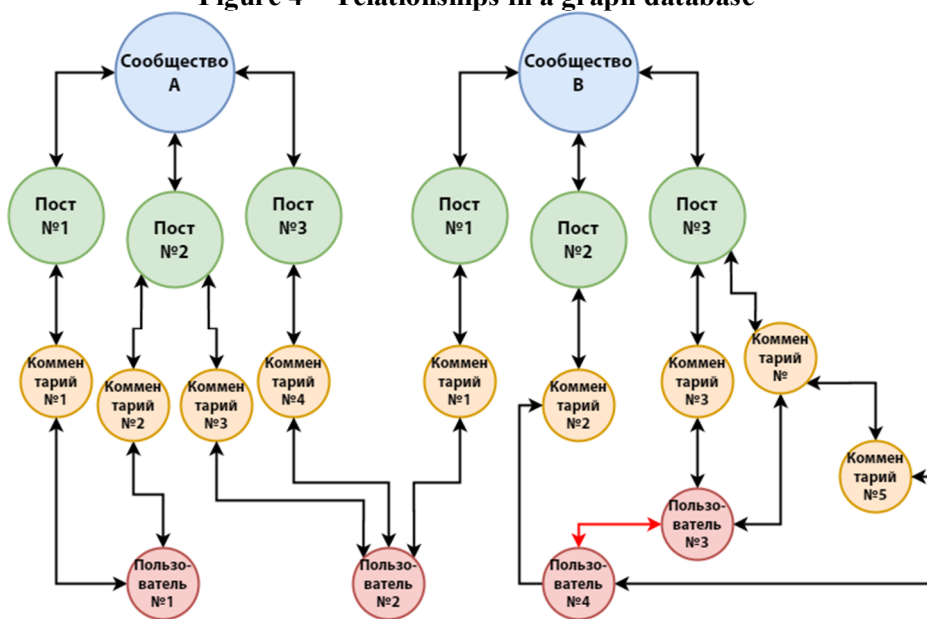


Рисунок 5 – Модель графовой базы данных  
Figure 5 – Graph database model

### **Графовые базы данных для анализа социальных сетей**

Графовые базы данных являются мощным инструментом анализа социальных сетей. Они специально разработаны для хранения и обработки данных в графовом виде, что делает их эффективным инструментом моделирования и анализа связей между различными сущностями, такими как пользователи социальных сетей и/или их взаимодействия.

К примеру, графовые БД применимы для анализа социальных сетей в качестве:

1) средства хранения структуры социальной сети: графовая БД может эффективно хранить информацию о пользователях, их профилях и взаимосвязях между ними, такими как дружба, подписки, лайки и комментарии. Каждый пользователь представлен узлом графа, а связи между ними – ребром графа;

2) подходящей среды для использования графовых алгоритмов: графовые БД предоставляют богатый набор алгоритмов для анализа графовых структур. Такие алгоритмы включают в себя поиск кратчайших путей, выявление центральных узлов (важных пользователей), обнаружение сообществ (групп пользователей с похожими интересами) и решение многих других задач. Кроме того, различные подходы могут быть таким образом унифицированы, даже несмотря на то, что данные могут быть представлены с различным заполнением и набором отдельных полей;

3) основы для построения рекомендательной системы (на основе предпочтений пользователей): графовые БД могут использоваться для создания рекомендательных систем, которые предлагают пользователям контент или возможных друзей на основе их предыдущих действий и интересов, указанных в социальной сети;

4) обнаружения аномалий, спама и мошеннических действий среди пользователей тематических групп социальной сети: графовые алгоритмы могут помочь выявлять аномалии в структуре сети, такие как необычная активность или попытки мошенничества через анализ структуры связей между узлами графа (например, пользователями). Примером здесь может являться взаимодействие взрослого пользователя и несовершеннолетнего пользователя для случаев, когда у взрослого пользователя нет никаких связей с несовершеннолетним (не родители/не семья, не учитель, не входит в друзей-друзей и прочие условия);

5) средства визуализации сети: графовые БД позволяют создавать наглядные визуализации социальных сетей, что облегчает понимание и визуальное представление структур и взаимосвязей между отдельными элементами сети. Например, визуализация комментариев может показать охват (окупаемость бюджета) рекламного поста на странице сообщества.

### **Графовые базы данных для логистики и анализа сетей транспорта**

Графовые БД также оказываются очень полезными в области логистики и анализа сетей транспорта. Они способны эффективно моделировать и анализировать сложные сети дорог, маршрутов, транспортных узлов и грузоперевозок. К примеру:

1) моделирование транспортной сети: графовая БД позволяет хранить информацию о дорогах, железнодорожных путях, портах, аэропортах и других элементах транспортной инфраструктуры в виде графовых узлов и ребер. Это позволяет легко определить оптимальные маршруты для грузоперевозок и пассажирских перевозок;

2) маршрутизация трафика и оптимизация сети: графовые БД могут использоваться для поиска оптимальных маршрутов для грузовых перевозок и общественного транспорта. Алгоритмы поиска кратчайших путей (например, алгоритм Дейкстры или A\*), доступные в графовых БД, позволяют быстро вычислять оптимальные маршруты с учетом различных параметров, таких как расстояние, время и стоимость, типы применяемых узлов и связей;

3) мониторинг трафика и управление грузоперевозками: графовые БД могут использоваться для отслеживания местоположения грузов и управления логистическими операциями. Информация о перемещении грузов может быть связана с графовой структурой для более эффективного управления и мониторинга;



4) анализ сети и прогнозирование нагрузки: графовые БД позволяют проводить анализ сетей транспорта, выявлять узкие места, прогнозировать потребности в транспортных ресурсах и позволять разрабатывать стратегии для оптимизации сети;

5) системы мониторинга транспорта: графовые БД могут быть использованы для создания систем отслеживания и управления парками транспортных средств.

### **Архитектура приложений и используемые библиотеки**

Для работы с графовыми БД и последующей визуализации графа (отображения результата) можно воспользоваться популярным языком программирования Python, версии 3.8 и выше [15-17]. Более того, для языка программирования Python существует несколько популярных библиотек:

1) `py2neo` [18] – это Python-библиотека, предназначенная для работы с графовой базой данных Neo4j. Продукт Neo4j – это популярная графовая СУБД, которая использует графовую модель данных для хранения и обработки информации. Библиотека `py2neo` предоставляет удобные средства для создания, обновления, удаления и выполнения запросов к данным в Neo4j из Python. Она облегчает работу с базой данных и предоставляет объектно-ориентированный API для работы с узлами и отношениями. Кроме того, библиотека поддерживает асинхронное взаимодействие с Neo4j, что позволяет асинхронно выполнять запросы и эффективно работать с большими объемами данных;

2) `NetworkX` [19] – это библиотека Python для работы с графами и выполнения анализа сетей. Она предоставляет удобные инструменты для создания, манипулирования и анализа графовых структур. `NetworkX` разработана с упором на простоту использования и понимания. Она предоставляет интуитивно понятный API для работы с графами. Библиотека поддерживает различные типы графов, такие как направленные и ненаправленные графы, мультиграфы (графы, в которых могут существовать несколько ребер между одной и той же парой узлов) и взвешенные графы (с ребрами, имеющими веса);

3) `matplotlib` [20] – это популярная библиотека Python для создания статических, интерактивных и анимированных графиков и визуализации данных. Она предоставляет разнообразные инструменты для создания графиков и диаграмм, что делает ее незаменимой для визуализации данных и исследования научных результатов. `Matplotlib` предоставляет простой и понятный интерфейс для создания графических изображений. Она позволяет легко настраивать внешний вид графиков и добавлять к ним различные элементы, такие как легенды и подписи. Библиотека поддерживает широкий спектр типов графиков, включая линейные графики, столбчатые диаграммы, точечные графики, гистограммы, круговые диаграммы и многие другие;

4) `Gephi` [21]-`Gephi` – это мощное программное обеспечение с открытым исходным кодом для визуализации и анализа сложных сетей на графовых данных. Оно предоставляет средства для визуализации и исследования связей между данными. `Gephi` обладает привычным многим пользователям пользовательским интерфейсом, который позволяет создавать, настраивать и визуализировать графовые структуры без необходимости программирования. `Gephi` способна обрабатывать и визуализировать большие и сложные сети, что делает ее полезной для анализа социальных сетей, транспортных сетей, биологических сетей и других сложных систем. Отдельно стоит отметить поддержку различных форматов данных, включая формат `GraphML` и возможность расширения функциональности программы через механизм использования сторонних модулей-плагинов;

Для работы с графовыми БД и последующей визуализации графа (отображения результата) можно воспользоваться популярным языком программирования Python, версии 3.8 и выше [15-17]. Более того, для языка программирования Python существует несколько популярных библиотек:

1) `py2neo` [18] – это Python-библиотека, предназначенная для работы с графовой базой данных Neo4j. Продукт Neo4j – это популярная графовая СУБД, которая использует графовую

вую модель данных для хранения и обработки информации. Библиотека `py2neo` предоставляет удобные средства для создания, обновления, удаления и выполнения запросов к данным в `Neo4j` из `Python`. Она облегчает работу с базой данных и предоставляет объектно-ориентированный API для работы с узлами и отношениями. Кроме того, библиотека поддерживает асинхронное взаимодействие с `Neo4j`, что позволяет асинхронно выполнять запросы и эффективно работать с большими объемами данных;

2) `NetworkX` [19] – это библиотека `Python` для работы с графами и выполнения анализа сетей. Она предоставляет удобные инструменты для создания, манипулирования и анализа графовых структур. `NetworkX` разработана с упором на простоту использования и понимания. Она предоставляет интуитивно понятный API для работы с графами. Библиотека поддерживает различные типы графов, такие как направленные и ненаправленные графы, мультиграфы (графы, в которых могут существовать несколько ребер между одной и той же парой узлов) и взвешенные графы (с ребрами, имеющими веса);

3) `matplotlib` [20] – это популярная библиотека `Python` для создания статических, интерактивных и анимированных графиков и визуализации данных. Она предоставляет разнообразные инструменты для создания графиков и диаграмм, что делает ее незаменимой для визуализации данных и исследования научных результатов. `Matplotlib` предоставляет простой и понятный интерфейс для создания графических изображений. Она позволяет легко настраивать внешний вид графиков и добавлять к ним различные элементы, такие как легенды и подписи. Библиотека поддерживает широкий спектр типов графиков, включая линейные графики, столбчатые диаграммы, точечные графики, гистограммы, круговые диаграммы и многие другие;

4) `Gephi` [21]-`Gephi` – это мощное программное обеспечение с открытым исходным кодом для визуализации и анализа сложных сетей на графовых данных. Оно предоставляет средства для визуализации и исследования связей между данными. `Gephi` обладает привычным многим пользователям пользовательским интерфейсом, который позволяет создавать, настраивать и визуализировать графовые структуры без необходимости программирования. `Gephi` способна обрабатывать и визуализировать большие и сложные сети, что делает ее полезной для анализа социальных сетей, транспортных сетей, биологических сетей и других сложных систем. Отдельно стоит отметить поддержку различных форматов данных, включая формат `GraphML` и возможность расширения функциональности программы через механизм использования сторонних модулей-плагинов;

5) `Plotly` [22] – это библиотека `Python` для создания интерактивных графиков и визуализации данных. Она предоставляет средства для создания высококачественных графиков и диаграмм, которые можно легко внедрять в веб-приложения и дашборды. `Plotly` специализируется на создании именно интерактивных графиков, что позволяет пользователям взаимодействовать с представленными данными. Взаимодействия включают в себя приближение, перемещение, выделение, а также дополнительные интерактивные функции. Библиотека поддерживает разнообразные типы графиков, включая линейные графики, столбчатые диаграммы, точечные графики, гистограммы, круговые диаграммы, тепловые карты и другие.

### Заключение

Разработка графовых БД представляет собой важное направление в области управления данными и привлекает внимание как исследователей, так и инженеров по всему миру. В данной статье мы рассмотрели ключевые аспекты разработки графовых БД, что приводит нас к следующим выводам:

1) с развитием таких явлений, как интернет вещей и социальные сети, для приложений, которые требуют работы со связанными между собой данными, графовые БД становятся все более востребованными. Такие системы позволяют эффективно моделировать и анализировать сложные отношения между различными (возможно неструктурированными или слабо структурированными) данными;

2) графовые БД предоставляют уникальные возможности для моделирования данных, представляя информацию в виде вершин и ребер, что идеально подходит для задач, связанных с анализом связей и сетей;

3) графовые БД разрабатываются с учетом эффективной обработки запросов, особенно касающихся поиска путей и анализа связей. Это делает их отличным выбором для приложений, требующих быстрого доступа к подобным данным.

В современных условиях разработка графовых СУБД и связанных с ними систем продолжается, предоставляя новые возможности для решения разнообразных задач в области анализа данных и управления информацией. Отдельно отметим, что выбор конкретной СУБД и применяемых инструментальных средств зависит от специфики проекта и его требований.

*Работа выполнена при финансовой поддержке Российского научного фонда (РНФ), грант № 23-21-00153*

### Библиографический список

1. Daniel Walke, Daniel Micheel, Kay Schallert, Thilo Muth, David Broneske, Gunter Saake, Robert Heyer, The importance of graph databases and graph learning for clinical applications, Database, 2023, baad045, <https://doi.org/10.1093/database/baad045>.
2. Santiago Timón-Reina, Mariano Rincón, Rafael Martínez-Tomás, An overview of graph databases and their applications in the biomedical domain, Database, Volume 2021, 2021, baab026, <https://doi.org/10.1093/database/baab026>.
3. Anil Pacaci, Alice Zhou, Jimmy Lin, and M. Tamer Özsu. 2017. Do We Need Specialized Graph Databases? Benchmarking Real-Time Social Networking Applications. In Proceedings of the Fifth International Workshop on Graph Data-management Experiences & Systems (GRADES'17). Association for Computing Machinery, New York, NY, USA, Article 12, 1–7. <https://doi.org/10.1145/3078447.3078459>.
4. Pokorný, J. (2015). Graph Databases: Their Power and Limitations. In: Saeed, K., Homenda, W. (eds) Computer Information Systems and Industrial Management. CISIM 2015. Lecture Notes in Computer Science(), vol. 9339. Springer, Cham. [https://doi.org/10.1007/978-3-319-24369-6\\_5](https://doi.org/10.1007/978-3-319-24369-6_5).
5. S. Paul, A. Mitra and C. Koner. A Review on Graph Database and its representation. 2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC), Nagercoil, India, 2019, pp. 1-5, doi: 10.1109/ICRAECC43874.2019.8995006.
6. Jaroslav Pokorný, Michal Valenta, Jiří Kovačič, Integrity constraints in graph databases, Procedia Computer Science, vol. 109, 2017, pp. 975-981, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2017.05.456>.
7. David Luaces, José R.R. Viqueira, José M. Cotos, Julián C. Flores, Efficient access methods for very large distributed graph databases, Information Sciences, vol. 573, 2021, Pages 65-81, ISSN 0020-0255, <https://doi.org/10.1016/j.ins.2021.05.047>.
8. Wei Jiang, Hai-Bo Hu, Liu-Gen Xu, Query Acceleration of Graph Databases by ID Caching Technology, DOI: 10.11989/JEST.1674-862X.80904163.
9. Steve Ataky, Marcela Xavier Ribeiro, Lucas Cesar Ferreira, Marilde Terezinha Prado Santos, Evaluation of Graph Databases Performance through Indexing Techniques, DOI: 10.5121/ijaia.2015.6506.
10. Macak, M.; Stovcik, M. and Buhnova, B. (2020). The Suitability of Graph Databases for Big Data Analysis: A Benchmark. In Proceedings of the 5th International Conference on Internet of Things, Big Data and Security - IoTBDS; ISBN 978-989-758-426-8; ISSN 2184-4976, SciTePress, pp 213-220. DOI: 10.5220/0009350902130220.
11. Liu Chunjiang, Li Shuying, Hu Hanlin, Fang Shu, Graph Databases for Complex Network Analysis DOI: 10.11925/infotech.2096-3467.2021.1168.
12. Kotiranta, P.; Junkkari, M.; Nummenmaa, J. Performance of Graph and Relational Databases in Complex Queries. Appl. Sci. 2022, 12, 6490. <https://doi.org/10.3390/app12136490>
13. Monteiro, J., Sá F., Bernardino J. Experimental Evaluation of Graph Databases: JanusGraph, Nebula Graph, Neo4j, and TigerGraph. Appl. Sci. 2023, 13, 5770. <https://doi.org/10.3390/app13095770>
14. Робинсон Я., Эфрем Э., Вебер Д., Графовые базы данных. Новые возможности для работы, М.: ДМК-Пресс, 2016 г. ISBN: 978-5-97060-201-0

15. **Маккинли У.** Python и анализ данных. Пер. с англ. Слинкин А.А. М.: ДМК Пресс, 2015. 482 с. ISBN 978-5-97060-315-4.
16. **Хеллман Д.** Стандартная библиотека Python 3. Справочник с примерами. Диалектика. 2-е изд. 2019. 1375 с.
17. **Персиваль Г.** Python. Разработка на основе тестирования. 2018. 624 с. ISBN: 978-5-97060-594-3.
18. **py2neo** [Электронный источник] URL: <https://py2neo.org/2021.1>.
19. **NetworkX** [Электронный источник] URL: <https://networkx.org/>.
20. **matplotlib** [Электронный источник] URL: <https://matplotlib.org/>.
21. **Gephi** [Электронный источник] URL: <https://gephi.org/>.
22. **Plotly** [Электронный источник] URL: <https://plotly.com/python/>.

UDC 004.658

## ADVANTAGES OF USING GRAPH DATABASES IN APPLICATION

**K. K. Otradnov**, Senior Lecturer, Department of Telecommunications, Institute of Radio Electronics and Informatics, Federal State Budgetary Educational Institution of Higher Education «MIREA – Russian Technological University», Moscow, Russia;  
e-mail: const.otradnov@yandex.ru

**A. S. Aleshkin**, assistant professor, Department KB-3 «Security of Software Solutions», Institute of Cybersecurity and Digital Technologies, Federal State Budgetary Educational Institution of Higher Education «MIREA – Russian Technological University», Moscow, Russia; e-mail: antony@testor.ru

**V. N. Kalinin**, additional education teacher, Department of Applied Informatics and Intelligent Systems in the Humanities, RUDN University, Moscow, Russia;  
e-mail: vkalininz@mail.ru

*The article provides an overview of the advantages of graph databases, the differences between graph and relational databases, the use of graph databases in various industries (including for media space analysis and transport network analysis), the libraries used and the model (diagram) of a relational and graph database are given to analyze social network data.*

**Keywords:** graph databases, graph database design, Python, Neo4j, social networks, transport networks.

**DOI:** 10.21667/1995-4565-2023-86-73-85

### References

1. **Daniel Walke, Daniel Micheel, Kay Schallert, Thilo Muth, David Broneske, Gunter Saake, Robert Heyer.** The importance of graph databases and graph learning for clinical applications. *Database*. 2023, baad045, <https://doi.org/10.1093/database/baad045>.
2. **Santiago Timón-Reina, Mariano Rincón, Rafael Martínez-Tomás,** An overview of graph databases and their applications in the biomedical domain. *Database*, vol. 2021, baab026, <https://doi.org/10.1093/database/baab026>.
3. **Anil Pacaci, Alice Zhou, Jimmy Lin, and M. Tamer Özsu.** 2017. Do We Need Specialized Graph Databases? Benchmarking Real-Time Social Networking Applications. *In Proceedings of the Fifth International Workshop on Graph Data-management Experiences & Systems (GRADES'17)*. Association for Computing Machinery, New York, NY, USA, Article 12, 1–7. <https://doi.org/10.1145/3078447.3078459>.
4. **Pokorný, J.** (2015). Graph Databases: Their Power and Limitations. In: Saeed, K., Homenda, W. (eds). *Computer Information Systems and Industrial Management. CISIM 2015*. Lecture Notes in Computer Science, vol 9339. Springer, Cham. [https://doi.org/10.1007/978-3-319-24369-6\\_5](https://doi.org/10.1007/978-3-319-24369-6_5).
5. **S. Paul, A. Mitra and C. Koner.** A Review on Graph Database and its representation, *2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC)*, Nagercoil, India, 2019, pp. 1-5, doi: 10.1109/ICRAECC43874.2019.8995006.

6. **Jaroslav Pokorný, Michal Valenta, Jiří Kovačič.** Integrity constraints in graph databases. *Procedia Computer Science*, vol. 109, 2017, pp. 975-981, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2017.05.456>
7. **David Luaces, José R.R. Viqueira, José M. Cotos, Julián C. Flores.** Efficient access methods for very large distributed graph databases. *Information Sciences*. Volume 573, 2021, pp. 65-81, ISSN 0020-0255, <https://doi.org/10.1016/j.ins.2021.05.047>.
8. **Wei Jiang, Hai-Bo Hu, Liu-Gen Xu.** *Query Acceleration of Graph Databases by ID Caching Technology*. DOI: 10.11989/JEST.1674-862X.80904163.
9. **Steve Ataky, Marcela Xavier Ribeiro, Lucas Cesar Ferreira, Marilde Terezinha Prado Santos,** *Evaluation of Graph Databases Performance through Indexing Techniques*. DOI: 10.5121/ijaia.2015.6506.
10. **Macak, M.; Stovcik, M. and Buhnova, B.** (2020). The Suitability of Graph Databases for Big Data Analysis: A Benchmark. In *Proceedings of the 5th International Conference on Internet of Things, Big Data and Security - IoTBDS*; ISBN 978-989-758-426-8; ISSN 2184-4976, SciTePress, pp. 213-220. DOI: 10.5220/0009350902130220.
11. **Liu Chunjiang, Li Shuying, Hu Hanlin, Fang Shu.** *Graph Databases for Complex Network Analysis*. DOI: 10.11925/infotech.2096-3467.2021.1168.
12. **Kotiranta, P.; Junkkari, M.; Nummenmaa, J.** Performance of Graph and Relational Databases in Complex Queries. *Appl. Sci.* 2022, 12, 6490. <https://doi.org/10.3390/app12136490>.
13. **Monteiro, J., Sá, F., Bernardino J.** Experimental Evaluation of Graph Databases: JanusGraph, Nebula Graph, Neo4j, and TigerGraph. *Appl. Sci.* 2023, 13, 5770. <https://doi.org/10.3390/app13095770>
14. **Robinson Ian, Eifrem Emil, Weber Jim.** *Graph Databases. New opportunities for work*. 2016 ISBN: 978-5-97060-201-0.
15. **Makkinli U.** *Python i analiz dannykh* (Python and data analysis). Per. s angl. Slinkin A.A. Moscow: DMK Press, 2015. 482 p. ISBN 978-5-97060-315-4.
16. **Khellman D.** *Standartnaya biblioteka Python 3. Spravochnik s primerami. Dialektika* (Python 3 Standard Library. Reference with examples. Dialectics). 2-nd Ed. 2019. 1375 p.
17. **Persival G.** *Python. Razrabotka na osnove testirovaniya* (Python. Test Driven Development). 2018. 624 p. ISBN: 978-5-97060-594-3.
18. **py2neo** [Electronic source] URL: <https://py2neo.org/2021.1/>
19. **NetworkX** [Electronic source] URL: <https://networkx.org/>
20. **matplotlib** [Electronic source] URL: <https://matplotlib.org/>
21. **Gephi** [Electronic source] URL: <https://gephi.org/>
22. **Plotly** [Electronic source] URL: <https://plotly.com/python/>