Оригинальная статья / Original article

https://doi.org/10.21869/2223-1536-2023-13-1-123-142

Исследование эффективности использования графовых баз данных для анализа больших данных

Р. В. Фаткуллин[1] ✉, Е. В. Кислицын[1]

[1] Уральский технический институт связи и информатики (филиал) Сибирского государственного университета телекоммуникаций и информатики
ул. Репина, д. 15, г. Екатеринбург 620014, Российская Федерация

✉ e-mail: buddhaeye13@gmail.com

Резюме

Цель исследования. *Целью настоящей работы является исследование графовых моделей баз данных и разработка методики сравнительного анализа моделей баз данных. Теоретико-методологическую основу исследования составили фундаментальные научные труды отечественных и зарубежных авторов в области базовых проблем теории баз данных, теории алгоритмов, теории графов, структур и методов обработки данных.*

Методы. *В работе используются методы структурного, сравнительного и контент-анализа, а также статистические методы обработки информации и методы теории графов. В результате проведенных исследований авторы обосновали особенности, преимущества и недостатки использования графовой модели данных.*

Результаты. *Актуальность настоящего исследования обусловлена интенсивным развитием информационных технологий, предназначенных для экономического развития страны, пандемией и геополитической ситуацией в мире. Данные предпосылки ориентируют исследователей к использованию новых методов обработки и анализа данных. Однако оптимизировать процессы обработки больших данных представляется возможным не только с помощью новых мощных алгоритмов, но и с помощью использования принципиально иных структур и моделей данных, отличных от реляционной.*

В работе приведены прикладные примеры использования графовой модели баз данных в различных предметных областях. Разработана методика сравнительного анализа моделей данных применительно к анализу больших данных. Выделены основные пункты проектирования модели данных: масштабирование системы, соответствие требованиям и стандартам, способность изменять структуры модели данных, сложность языка, производительность и скорость обработки данных. Предложенная методика позволила численно оценить эффективность применения графовых моделей.

Заключение. *Теоретическая значимость исследования состоит в развитии методических и технологических подходов к анализу больших данных и формированию структур и баз данных. Практические результаты исследования могут быть полезны крупным ИТ-компаниям, а также финансовому, логистическому и коммерческому секторам, где проблема анализа и исследования больших данных стоит наиболее остро.*

Ключевые слова: *базы данных; система управления базами данных; графовая модель данных; реляционная модель; документная модель.*

Конфликт интересов*: Авторы декларируют отсутствие явных и потенциальных конфликтов интересов, связанных с публикацией настоящей статьи.*

Для цитирования: Фаткуллин Р. В., Кислицын Е. В. Исследование эффективности использования графовых баз данных для анализа больших данных // Известия Юго-Западного государственного университета. Серия: Управление, вычислительная техника, информатика. Медицинское приборостроение. 2023. Т. 13, № 1. С. 123–142. https://doi.org/10.21869/ 2223-1536-2023-13-1-123-142.

Поступила в редакцию 16.01.2023 *Подписана в печать 06.02.2023* *Опубликована 30.03.2023*

Investigation of the Effectiveness of Usage of Graph Databases for Big Data Analysis

Ruslan V. Fatkullin[1] ✉, Evgeny V. Kislitsyn[1]

[1] Ural Technical Institute of Communications and Informatics (branch) of the Siberian State University of Telecommunications and Informatics
15 Repina Str., Ekaterinburg 620014, Russian Federation

✉ e-mail: buddhaeye13@gmail.com

Abstract

The purpose of research. *The purpose of this work is to study graph models of databases and develop a methodology for comparative analysis of database models. The theoretical and methodological basis of the study was the fundamental scientific works of domestic and foreign authors in the field of basic problems of database theory, algorithm theory, graph theory, data processing structures and methods.*

Methods. *The paper uses methods of structural, comparative and content analysis, as well as statistical methods of information processing and methods of graph theory. As a result of the conducted research, the authors justified the features, advantages and disadvantages of using a graph data model.*

Results. *The relevance of this study is due to the intensive development of information technologies intended for the economic development of the country, the pandemic and the geopolitical situation in the world. These prerequisites orient researchers to use new methods of data processing and analysis. However, it is possible to optimize big data processing processes not only with the help of powerful new algorithms, but also with the use of fundamentally different data structures and models other than relational.*

The paper presents applied examples of using the graph model of databases in various subject areas. A method of comparative analysis of data models in relation to big data analysis has been developed. The main points of data model design are highlighted: system scaling, compliance with requirements and standards, the ability to change data model structures, language complexity, performance and data processing speed. The proposed technique made it possible to numerically evaluate the effectiveness of graph models.

Conclusion. *The theoretical significance of the research consists in the development of methodological and technological approaches to the analysis of big data and the formation of structures and databases. The practical results of the study can be useful to large IT companies, as well as to the financial, logistics and commercial sectors, where the problem of big data analysis and research is most acute.*

Keywords: *databases; database management system; graph data model; relational model; document model.*

Conflict of interest: *The Authors declares the absence of obvious and potential conflicts of interest related to the publication of this article.*

For citation: Fatkullin R. V., Kislitsyn E. V. Investigation of the effectiveness of usage of graph databases for big data analysis. *Izvestiya Yugo-Zapadnogo gosudarstvennogo universiteta. Serija: Upravlenie, vychislitel'naja tekhnika, informatika. Meditsinskoe priborostroenie = Proceedings of the Southwest State University. Series: Control, Computer Engineering, Information Science. Medical Instruments Engineering.* 2023; 13(1): 123–142. (In Russ.) https://doi.org/10.21869/2223-1536-2023-13-1-123-142.

Received 16.01.2023 *Accepted 06.02.2023* *Published 30.03.2023*

Введение

С появлением цивилизации человек пытается любыми способами запечатлеть окружающий его мир, используя всевозможные способы хранения информации. В древние времена люди хранили данные у себя в голове и передавали их младшим поколениям с помощью разговорной речи, затем использовались петроглифы, записи на глиняных табличках, печатались первые книги, в счетных машинах стали использовать перфокарты, которые нашли применение в ранних ЭВМ, создавались магнитные ленты, твердотельные накопители, и наконец, разрабатывались способы хранения информации в ДНК.

Все вышеперечисленное представляет собой способы хранения информации (или, другими словами, базу данных). Каждый из этих методов заменял устаревшие варианты хранения данных или являлся совокупностью ранее использовавшихся методов, но, безусловно, все эти способы применялись для определенной цели – хранения информации. С помощью этой информации человек производит анализ, находит закономерности, которые позволяют производить эффективное управление бизнесом, оценивать ситуацию на рынке, обеспечивать поставки продукции и т. д.

В современном мире лидирующее место для обеспечения хранения информации занимают системы, основанные на реляционном подходе. Их суть заключается в хранении данных с помощью таблиц (*relational* с англ. – отношение, зависимость), которые соединены между собой определенными связями и имеют определенные ограничения для обеспечения целостности, корректности и неизбыточности данных.

На рисунке 1 представлены СУБД, наиболее используемые и высоко оцененные пользователями и разработчиками баз данных.

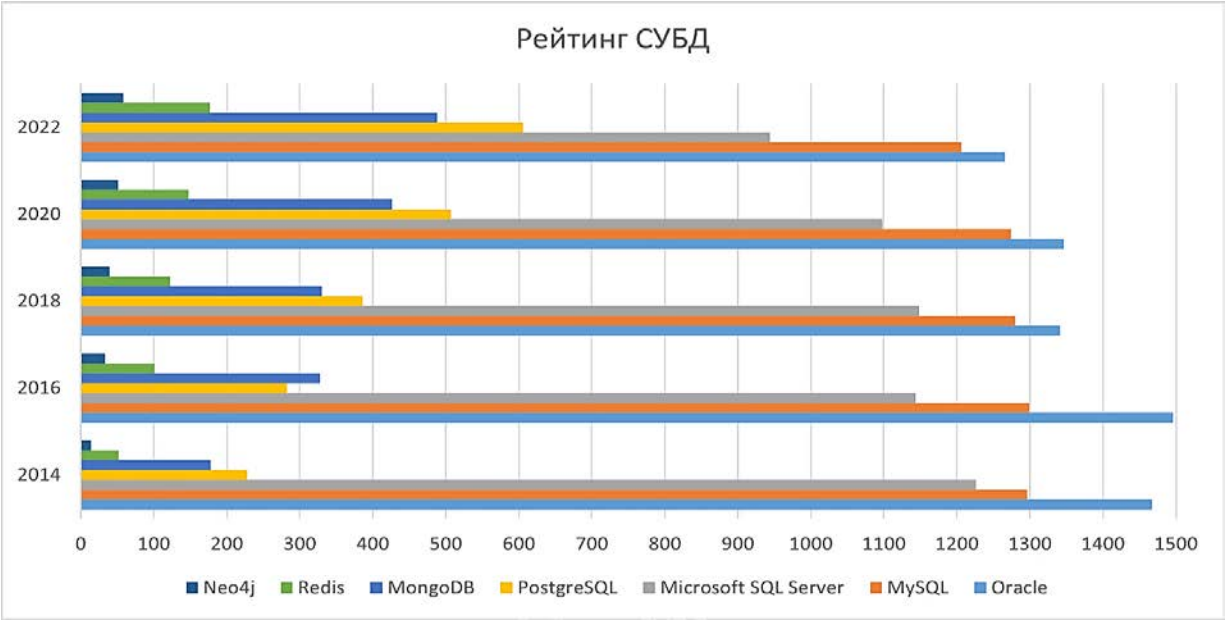

Рис. 1. Рейтинг СУБД [1]

Fig. 1. DBMS rating [1]

Отметим, что в последнее десятилетие активно набирают популярность NoSQL – базы данных (нереляционные СУБД). Например, такие, как MongoDB – документоориентированная СУБД, не требующая описания схемы таблиц; Redis – резидентная СУБД, работающая со структурами данных типа «ключ – значение»; Neo4j – графовая СУБД.

В отечественных научных исследованиях в последние годы все чаще встречаются нереляционные модели баз данных [2; 3], в том числе попытки сравнительного анализа с классическими реляционными базами данных [4]. Чаще всего графовые модели используются в экспертных интеллектуальных системах [5; 6; 7] и методах анализа данных [8]. В частности, ряд работ посвящен проблеме скорости обработки данных в разных моделях [9]. Однако до сих пор отсутствуют работы, связанные с исследованием самой структуры графовых моделей баз данных и обоснованием их применимости для анализа больших данных.

Таким образом, целью настоящего исследования является определение графовой базы данных как мощного инструмента для анализа больших данных и разработка методики сравнительного анализа моделей баз данных.

Материалы и методы

Графовая база данных – это нереляционная база данных, относящаяся к классу NoSQL систем управления баз данных. Аббревиатура NoSQL обозначает Not Only SQL – «Не только SQL», данные СУБД не используют язык запросов SQL, предназначены для неструктурированной информации, частично или полностью отказываются от требований атомарности и согласованности данных. Хотя NoSQL – слишком обширный термин, так как каждый может определять и интерпретировать его по-своему, – это наименование точнее выражает суть развития IT-технологий. В данных системах рассматриваются и используются альтернативные подходы к модели данных, к проектированию, организации, взаимодействию и хранению информации, отличаясь от привычных требований ANSI SQL.

Идея графовых баз данных исходит из математической теории графов, которая описывает и изучает структуру отношений между объектами. Так же как и в математике, в информационном пространстве графовые базы данных имеют вершины (другими словами, точки; *nodes* с англ. – узлы), хранящие информация об объекте, с которым производятся различные изменения или обновления, ребра (или связи, другими словами, *edges* с англ. – грани), которые отображают взаимосвязи между вершинами и описывают семантику и свойства этих отношений, т. е. имеют, как и в привычных графах, направления. И последние – это атрибуты (или по-другому – характеристики, *properties* с англ. – свойства), которые содержат описание вершин и иногда описание ребер [10].

Наглядным примером графовых баз данных в информационном пространстве является любая социальная сеть. В

ней объектами служат пользователи, которые имеют различные свойства (например, это имена пользователей, их увлечения и т. д.), а взаимосвязями или даже взаимоотношениями (это определение лучше подходит для контекста данного примера) является добавление других пользователей в список своих «друзей», тем самым можно наглядно увидеть связь между людьми. Также одним из наиболее распространенных и доступных примеров является карта метро, где станции – это узлы со свойствами в виде названия станции и связями в виде железнодорожных путей. На рисунке 2 представлена упрощенная графовая модель данных, в которой существует несколько объектов, и между ними имеются взаимосвязи.

На текущий момент существуют две основные разновидности модели графовых баз данных: *Property Graph* и модель данных *RDF* (от англ. *resource description framework* – среда описания ресурса).

Первая модель – *Property Graph* – означает, что используются узлы и ребра для хранения данных. В данной модели отсутствует привычная схема данных, которая облегчает моделирование системы, но сам граф можно назвать схемой данных. Такая особенность позволяет использовать полуструктурированные данные, например, если одна из вершин отличается от другой набором свойств, то не приходится полностью изменять дизайн модели и структуру описываемого объекта. Также в этой модели отношения являются явными, т. е. ограничения не задаются в привычном виде с помощью запроса на соединение, и отношения также могут иметь собственные характеристики, например вместимость, длину и т. д.

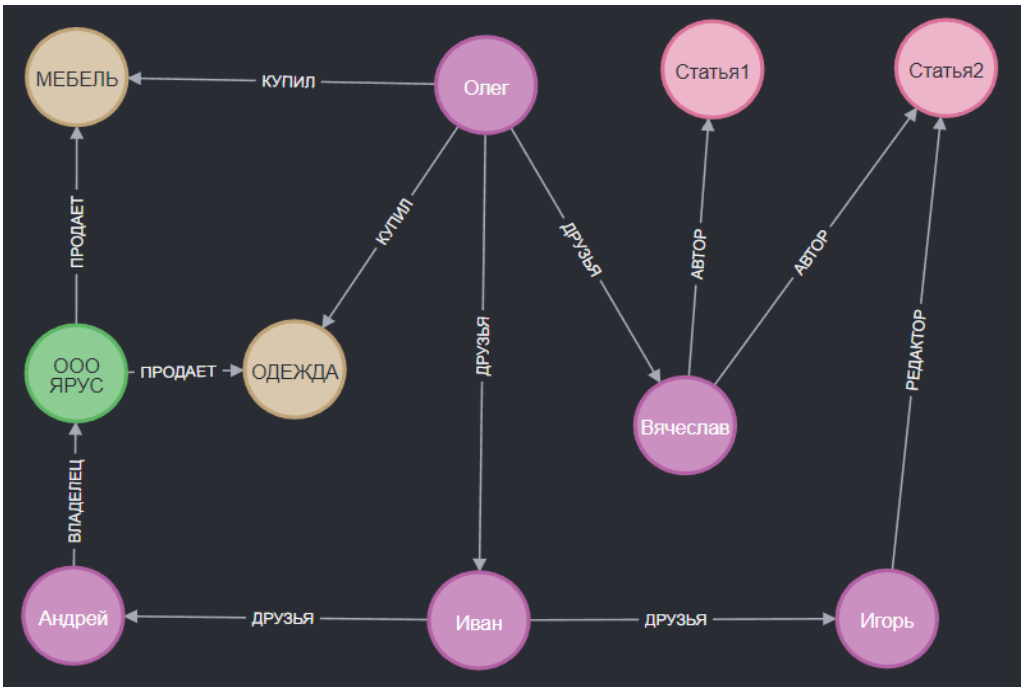

Рис. 2. Графовая модель

Fig. 2. Graph model

Модель данных *RDF* является графовой моделью данных, в которой формат записи графа представлен в виде «субъект – предикат – объект».

Например, если потребуется описать человека с определенными свойствами, то для этого будет использоваться схема, представленная ниже (рис. 3).

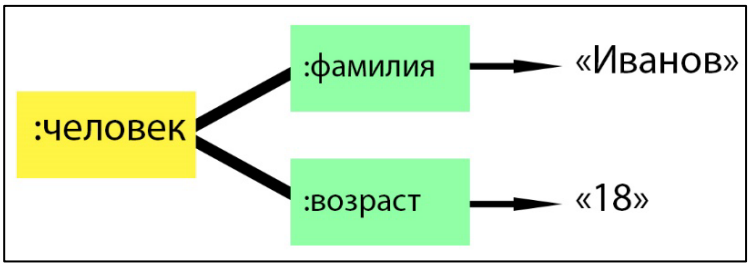

Рис. 3. Пример схемы структуры описания ресурсов (RDF)

Fig. 3. Example of Resource Description Framework Schema (RDF)

Главной функцией данной схемы является возможность открытым способом публиковать данные и разделять их, а также в аналитике, выводах, которые они предоставляют, и в их способности связывать разрозненные источники данных. Данная схема в полной мере помогает отразить опубликованную информацию, сообщать и классифицировать ее.

Графовые базы данных лучше всего предназначены для сложных моделей, где необходимо учитывать семантический контекст и важно качество и ценность данных [11; 12]. Они позволяют выполнять запросы на основе соединений и применять графовые алгоритмы для поиска отношений между узлами, что обеспечивает более эффективный анализ для больших объемов данных [13; 14]. Во время анализа графов алгоритмы анализируют отношения и расстояния между вершинами и важность вершин. Для определения значимости зачастую рассматривают входящие ребра и уровень важности соседних вершин. Например, если исследовать бизнес-процессы или человека в социальных сетях, то алгоритмы позволяют определить, какой объект больше всего связан с другими объектами для того, чтобы обнаружить закономерности или отклонения от стандартного поведения, общие ребра, узлы или пути.

На сегодняшний день информационные системы, которые основаны на графовых базах данных, становятся ключевым решением для управления бизнесом, так как они позволяют обрабатывать сложные и активно развивавшиеся отношения тесно связанных между собой данных, а также в тех случаях, где необходимо учесть взаимосвязи между пользователями какого-либо масштабного интернет-продукта; и способность воспринимать и анализировать огромные графы позволит компаниям занять лидирующую позицию при конкуренции. Графовые базы данных являются на текущий момент одним из лучших способов для отображения и обработки взаимосвязанных данных [15].

Результаты и их обсуждение

Все операции, производимые с помощью компьютеров, связаны с обработкой данных. Количество информации, которая нарастает с каждой секундой, необходимо структурировать и анализировать для составления прогноза поведения информационных систем, рекомендательных сервисов, систем, основанных на маршрутизации, компьютерных сетей, представляющих собой те же графовые базы данных. На сегодняшний день существуют две методики обработки данных – реляционная с использованием SQL и нереляционная NoSQL.

Реляционные базы данных (от англ. *relation* – отношение, связь) основаны на использовании таблиц (отношений), связь между которыми образована на использовании внешних ключей, являющихся идентификатором записи, тем самым другие таблицы ссылаются на этот внешний ключ, что позволяет создавать между ними связь. Главной особенностью является то, что структура реляционной базы данных заранее известна, в ней число строк может быть неограниченно, а вот число атрибутов фиксировано и задается при моделировании базы данных [6].

Основой выбора необходимой модели базы данных служит область, в которой она будет применяться. При выборе баз данных SQL в первую очередь учитывают соответствие ACID (атомарность, непротиворечивость, изолированность, устойчивость) требованиям, и им должна соответствовать любая производимая транзакция. Эти четыре взаимосвязанных элемента предназначены для уменьшения непредсказуемого поведения системы и обеспечения целостности данных.

Также каждая запись в таблице представляет собой отдельный исследуемый объект данных. Один из атрибутов в записи обычно используется для определения и задания уникальности с помощью первичного ключа.

Следующим этапом после проектирования является процесс нормализации, который предназначен для удаления избыточных данных из базы данных и уменьшения повторения информации. Каждая запись должна храниться в единственном экземпляре, и при изменении информации обновления вносятся в одном месте. Таким образом, реляционные базы данных лучше всего подходят для взаимодействия со структурированными данными, структуру которых нет необходимости часто изменять.

На выбор графовой модели базы данных влияет объем обрабатываемой информации и анализ взаимосвязей между объектами [16]. Также большую роль играет динамичность изменяемой структуры модели, которая может модифицироваться в любой момент времени. В графах удобно хранить информацию об объекте любого типа, добавлять новые атрибуты для объекта и данные, которые не являются структурированными. Графовые базы данных обеспечивают гибкий поиск взаимосвязей и их анализа, основанных на «силе» и качестве отношений, которые позволяют исследовать и обнаруживать связи и шаблоны в таких областях, как большие данные, хранилища данных, Интернет вещей,

в социальных и сложных данных транзакций для выявления мошенничества в бизнесе и банковской сфере. Из-за того, что графовые базы данных явно хранят взаимосвязи, запросы и алгоритмы, использующие связность между вершинами, выполняются намного быстрее, чем операции, направленные на соединения таблиц.

Документоориентированные базы данных хранят информацию об объекте в одной сущности, состоящей из массивов, коллекций, скалярных значений или вложенных документов, и каждый хранящийся объект может отличаться от остальных. Данные сущности называются «документами», что является ключевой концепцией документных баз данных. Документы имеют иерархическую древовидную структуру, хранят данные в качестве наборов «ключ-значение» и имеют XML- или JSON-формат извлечения информации [2].

Отличием от реляционных баз данных является гибкость при проектировании и изменение структуры модели данных, нет необходимости хранить пустые свойства объектов, в отличие от реляционных баз данных, где необходимо создать дополнительный столбец и указывать значение NULL, также атрибуты не имеют фиксированной размерности и типизации [17].

Для того чтобы выбрать определенную модель данных, необходимо обозначить основные критерии, используемые для сравнительного анализа. Интегральный показатель эффективности модели базы данных основывается на следующих критериях:

– масштабирование;

– требования;

– структура модели данных;

– сложность языка;

– применение в области больших данных.

Рассмотрим подробно каждый критерий.

1. *Масштабирование*. При увеличении объема данных увеличивается и проблема скорости обработки данных, что становится главным аспектом информационной системы. Масштабирование предполагает возможности системы справляться с увеличением производительности при помощи добавления новых ресурсов. Существуют два вида масштабирования – вертикальное и горизонтальное. В первом случае предполагается замена или добавление компонентов для наращивания мощностей сервера, во втором – добавляются новые серверы или разделение информации на несколько серверов. Данная проблема является основным показателем: чем больше клиентов системы, тем больше данных они предоставляют и тем больше создают запросов, качество и скорость которых влияет на получаемую прибыль.

2. *Требования*. ACID – это набор стандартов и требований к информационной системе, которые должны обеспечивать надежную и предсказуемую работу системы для выполнения транзакций в базе данных. Данные стандарты важны для финансовых операций и других сфер деятельности, где внутрь системы вводят большие массивы персональной информации, данные о финан-

совых операция и т. д. BASE – это модель данных, которая описывает концепцию нетранзакционных систем, обладает альтернативными решениями управления и хранения неструктурированных данных и позволяет применять нестандартные способы репликации баз данных.

3. *Структура модели данных*. При разработке базы данных необходимо формировать схему данных, которая будет отображать отношения и атрибуты предметной области, данный процесс является ключевым при разработке программного продукта, так как на основе разработанной концепции формируется вся база данных. Неправильно разработанная схема данных или частое изменение свойств усложняют разработку модели. В зависимости от хранящихся данных, которые могут быть структурированными или неструктурированными, зависит выбираемый тип модели данных. Тип модели влияет на скорость обработки взаимосвязей между данными, что сильно связано со структурой модели данных.

4. *Сложность языка*. Сложность языка влияет на количество специалистов, которые смогут обучиться синтаксису языка и использовать его в своих проектах. Увеличение сложности языка уменьшает количество специалистов, которые способны разбираться в необходимой предметной области, но тем самым повышает оплату их труда.

5. *Применение в области больших данных*. На текущий момент многие компании взаимодействуют с областью больших данных, которые влияют практически на все сферы жизни, и их быстрый и качественный анализ помогает спрогнозировать дальнейшее поведение компаний для увеличение своей прибыли. Это также формирует конкуренцию среди компаний, занимающихся разработкой программного обеспечения, которые нацелены на создание прибыльного продукта и формирование имиджа экспертной компании.

Первоначальной областью сравнения рассматривается масштабируемость баз данных. В реляционных базах данных используется вертикальная масштабируемость, т. е. при возрастании нагрузки на сервер наращивается мощность компонентов (ЦП, ОЗУ, СХД). Однако при достижении критического значения компонентов необходимо осуществлять горизонтальное масштабирование, чтобы избежать падения производительности и поддерживать согласованность данных. Это не значит, что реляционные базы данных не подходят для больших объемов данных, так как у них присутствует поддержка кластеризации. В случае с графовыми базами данных используется горизонтальная и вертикальная масштабируемость. В первом случае для этого добавляется больше серверов, распределяя поступающий трафики, позволяя повышать поступающий объем информации, во втором – увеличивается мощность самого оборудования путем добавления или улучшения компонентов сервера. Документные базы данных так же, как и графовые базы данных, рассчитаны на масштабируемость, и важная особенность заключается в том, что практически у всех до-

кументоориентированных СУБД существуют встроенные функции репликации и шардинга. Еще одним способом масштабирования является фрагментация данных по определенному полю. Такие данные динамически перемещаются между узлами для балансировки нагрузки. Например, можно ориентироваться на местоположение пользователей, где пользовательские данные конкретного региона хранятся и обрабатываются во фрагментах, обслуживаемых этим регионом.

Следующей областью сравнения является набор требований, которым следует модель базы данных. Как было сказано ранее, в реляционных базах данных соблюдаются требования ACID. Под атомарностью подразумевается полное выполнение транзакции или полное невыполнение, любое количество используемых операторов должно представлять собой неделимую логическую единицу. Согласованность обеспечивает сохранение логики базы данных и поддержку целостности данных. Изолированность влияет на поведение транзакций, они должны выполняться без оказывания влияния на промежуточные операции другой транзакции. Устойчивость обеспечивает успешное сохранение данных.

Графовые базы данных также придерживаются требования ACID, но в графах стремятся к обеспечению гибкости и масштабируемости системы, поэтому для них существует свой набор свойств под названием CAP (Consistency, Availability, Partition) и концепция BASE (Basically Available, Soft state, Eventual consistency). Теорема CAP основана на понятиях «согласованность», «доступность» и «устойчивость». Под согласованностью подразумевается, что во всех вычислительных узлах в один момент времени данные не противоречат друг другу. Доступность означает, что запрос к распределенной системе завершается корректным откликом, и сервер системы всегда ответит на запрос. Устойчивость дает возможность распределения базы данных по физическим узлам.

Концепция BASE является альтернативой для ACID-требований и предполагает базовую доступность (каждая транзакция гарантированно завершается или не выполняется), гибкое состояние (свойства и состояние системы могут изменяться в течение времени, даже без добавления новых данных) и согласованность в конечном счете (согласованность данных происходит через некоторое время, некоторое время данные могут быть несогласованными) [2].

Благодаря этим свойствам графовая модель данных может использоваться для систем, где необходима высокая пропускная способность и низкая задержка.

Документоориентированные базы данных также придерживаются теоремы CAP и стараются повысить уровень доступности с помощью репликации. В данном случае документные базы данных обеспечивают доступ к информации на разных узлах, которые существуют в виде асинхронных горизонтальных репликациях. Таким образом, запрос поступает на главный узел и распределя-

ется между зависимыми. Если же он выходит из строя, то реплики выбирают главного между ними, так как некоторые узлы могут оказаться предпочтительнее за счет близости по отношению к другим серверам или иметь больший объем оперативной памяти.

Документные модели данных не обеспечивают гарантии ACID, но некоторые отдельные операции являются атомарными. В то же время документоориентированные модели поддерживают атомарные, долговременные обновления отдельных документов и последовательное чтение. Вариантом выполнения транзакции служат следующие шаги: атомарно изменить состояние документа или документов, выполнить операцию и удостовериться, что система находится в допустимом состоянии и, если не возникло ошибок, отметить транзакцию как завершенную, в противном случае вернуть документ или документы в исходное состояние. Все это формирует концепцию отмены транзакций.

Также необходимо рассмотреть процесс взаимосвязей данных в каждой модели. Реляционные базы данных тоже имеют взаимосвязи, но применяются только на этапе моделирования как средство объединения таблиц. При увеличении объема информации общая структура данных становится сложной и все менее формализованной, реляционная модель нагружается многочисленными соединениями таблиц, требуя увеличения объема логики для проверки пустых значений. Увеличение взаимосвязей приводит к увеличению производимых объединений, которые снижают производительность и затрудняют внесение в базу данных изменений и обновлений. Выполняя операцию *JOIN*, база данных производит сравнение и объединение данных, согласно принципам Эйлера, находя перекрытие множеств. Таким образом, производительность объединения зависит и снижается при добавлении новых отношений (для реляционных баз данных отношения – это таблицы), а также при увеличении объема информации в таблицах. В графовой базе данных запись является узлом, и ключевое отличие заключается в том, что в процессе соединения узла с другим объектом создается отношение между этими вершинами.

Документоориентированные модели данных направлены на хранение структурированных, неструктурированных и денормализованных данных. В этой модели реализована система, при которой данные хранятся в JSON подобных форматах документов. Информация записывается в виде пар из ключей и значений – как в таблице, в которой есть идентификаторы и соответствующие им данные. Такая структура позволяет разместить сложно структурированную информацию. Каждый экземпляр документной модели данных имеет несколько баз данных, и каждая база данных может иметь несколько коллекций. Если сравнивать с реляционными базами данных, то сущность реляционной базы данных соответствует сущности документной базы данных, схемы в реляционных базах данных соответствуют

базам данных документоориентированных баз данных, а таблицы реляционных баз данных – это коллекции в документных базах данных [3].

В реляционной базе данных во время выполнения запроса происходит разбор, как записи объединены друг с другом, а в графовой базе данных нет необходимости искать соотношение между данными, надо просто просматривать связи узлов.

Стоит отметить, что базы данных NoSQL создавались для расширения возможностей взаимодействия и обработки информации, а не для замены распространенных реляционных баз данных, и тем самым язык запросов графовых баз данных и реляционных баз данных имеет схожий синтаксис. Существуют два языка запросов – *SPARQL* и *Cypher*, использование которых зависит от выбранной системы управления баз данных. Для работы с реляционными базами данных используется язык запросов *SQL*, для которого существует различные виды диалектов: PL/SQL, T-SQL, PL/pgSQL и др., которые также зависят от выбранной СУБД. Отличительной особенностью языка *Cyhper* является использование ASCII-символов, которые отвечают за поиск и направление взаимосвязей узлов. Но в отличие от оператора *SELECT*, в *Cypher* применяется инструкция *MATCH*, которая является шаблоном для поиска и задает критерии для информации, а *RETURN* определяет, какие узлы, атрибуты и взаимосвязи для совпавших данных должны быть возвращены [18].

Например, запрос

```
MATCH (p:Person {name: "Иван"}) -[:`ДРУЗЬЯ`]->(b) RETURN p,b
```

вернёт для узла с меткой «Пользователь» и именем «Иван» всех имеющихся друзей у данного объекта. Результат представлен ниже (рис. 4, 5).

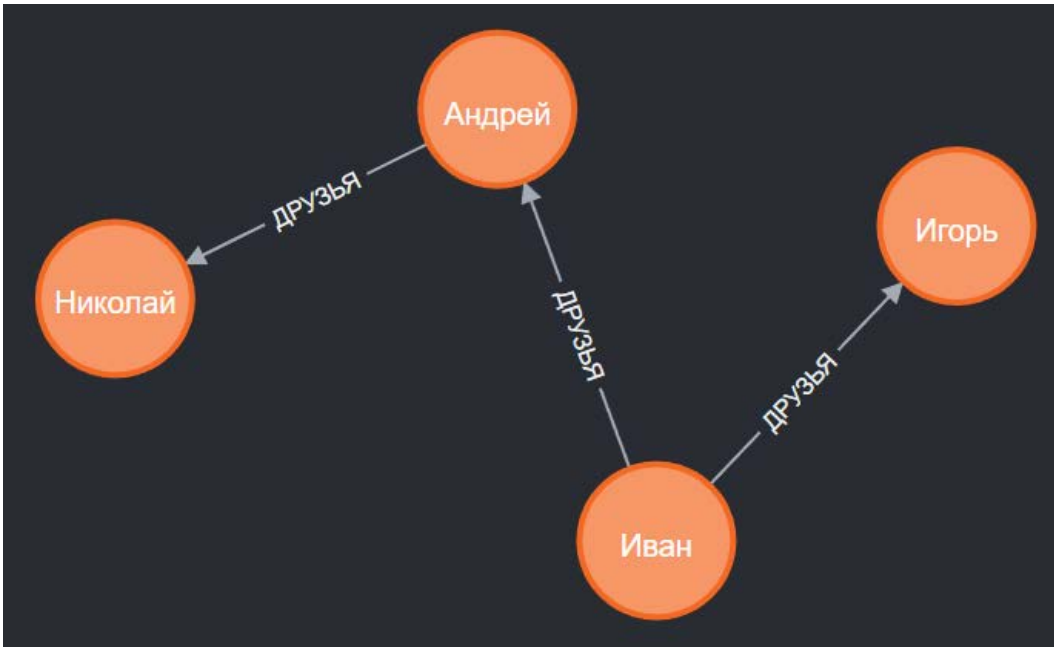

Рис. 4. Граф с пользователями

Fig. 4. Graph with users

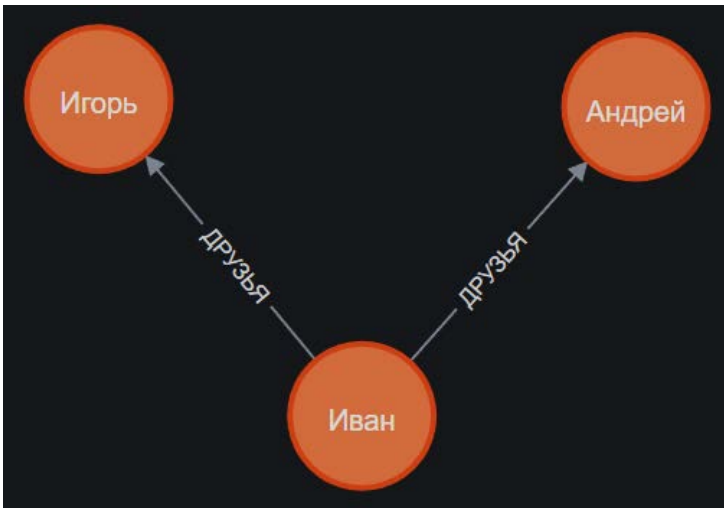

Рис. 5. Результат запроса

Fig. 5. Query result

Большие данные, или *Big Data*, – это массивы данных, которые отличаются от обычных данных объемом, разнообразием, так как они содержат структурированные, неструктурированные и полуструктурированные данные, скоростью генерации информации и обработки. Источниками больших данных является концепция Интернет вещей, камеры видеонаблюдения, социальные сети, фиксация действий колл-центра, параметры метеорологических станций, статистика медицинских данных и т. д.

И такую информацию необходимо обрабатывать и анализировать для эффективного управления всеми сферами жизни. Существует много методов обработки больших данных: Data Mining, машинное обучение, искусственные нейронные сети и др., но также используются реляционные базы данных и NoSQL-методы, куда входят рассматриваемые в данной статье графовые базы данных.

Термин «большие данные» обозначает не только массивы данных, но и технологии их обработки, так как стандартные способы, используемые для обработки обычных баз данных, теряют эффективность и могут привести к ошибочным результатам. Поэтому для их обработки и анализа используются специализированные программные комплексы, направленные на горизонтальную масштабируемость, которая требует больших усилий реализации для реляционных баз данных.

И как упоминалось ранее, реляционные базы данных плохо приспособлены к изменению структуры модели, что трудно учитывать для постоянно видоизменяющихся больших данных, и необходимо достаточно подробно и точно моделировать информационную систему, что усложняет внесение изменений, требующее значительных временных затрат. Все это также влияет на скорость поиска информации, который в графовых базах данных быстрее, чем в реляционных. В реляционной модели поисковой запрос проходит через наборы данных и выясняет, какие их них содержат совпадающую информацию. Для этого используются массивные

JOIN запросы SQL, графовый язык же находит узлы и взаимосвязи между ними. Это также влияет на объем, занимаемый программным кодом, что тоже немаловажно. На рисунке 6 представлен пример, который демонстрирует размерность запросов [19].

Документные базы данных так же, как и графовые базы данных, входят в NoSQL-область, что позволяет использовать их для обработки больших данных, а также способ хранения неструктурированной информации в виде документов с различной структурой данных позволяет эффективно их использовать и проводить анализ в реальном времени [20; 21].

На основе вышеперечисленных методов сформирована таблица сравнения моделей данных, оценивающая их по десятибалльной системе.

Графовые базы данных не предназначены только для хранения и управления большими данными, но неплохо для этого подходят, так как обладают хорошей горизонтальной и вертикальной масштабируемостью и им не требуется строгая структура данных. Они предлагают другой способ проведения анализа. Хранение данных в виде взаимосвязанных объектов облегчает поиск связанных данных при проведении прогнозной аналитики или прогнозов на основе искусственного интеллекта. Кроме этого, графовые базы данных лучше визуализируют информацию. И хотя с момента появления первых баз данных человек привык видеть различные таблицы, четко отображаемые взаимосвязи между узлами легче представить и отобразить визуально с помощью блок-схем.

Таким образом, было определено, что графовые базы данных – это класс баз данных NoSQL, основанных в первую очередь на построении взаимосвязей между данными, что является хорошим решением для работы с неструктурированными данными и подходит для хранения сложно связанных данных. Кроме этого, используют собственный язык в зависимости от выбираемой СУБД, но синтаксис которого понятен для пользователей, уже взаимодействовавших с SQL-языком. В графовых базах данных производительность не зависит от количества имеющихся узлов, так как нет привычного объединения таблиц, а используется навигация по узлам и ребрам, а в документоориентированных – поиск и обработка данных происходит по значениям «ключа».

```
MATCH (:Person {name: "Tom Hanks"})-->(:Movie)<-[:ACTED_IN]-(coActor:Person)
RETURN coActor.name
```

```
WITH tom_movies AS (
    SELECT movie.movie_id FROM movie
    INNER JOIN person_movie ON movie.movie_id = person_movie.movie_id
    INNER JOIN person ON person_movie.person_id = person.person_id
    WHERE person.name = "Tom Hanks")
SELECT person.name FROM person
INNER JOIN person_movie ON tom_movies = person_movie.movie_id
INNER JOIN person ON person_movie.person_id = person.person_id
INNER JOIN involvement ON person_movie.involve_id = involvement.involve_id
WHERE involvement.title = "Actor"
```

Рис. 6. Сравнение размеров кода

Fig. 6. Comparison of code sizes

Таблица. Сравнительный анализ моделей баз данных

Table. Comparative analysis of database models

Область сравнения	Реляционные	Графовые	Документные
Масштабирование (горизонтальное)	Необходимо создавать индексы и учитывать источник данных 6/10	Используется master-slave технология 9/10	Имеют встроенные функции масштабирования 10/10
Масштабирование (вертикальное)	10/10	10/10	10/10
Соответствие требованиям	Полностью придерживаются ACID требованиям 9/10	Частично соблюдают требования ACID и придерживаются концепции BASE 8/10	Следуют свойствам теоремы CAP, только некоторые транзакции соблюдают требования атомарности 6/10
Изменение структуры модели данных	Структуру сложно изменять после этапа проектирования 5/10	Нет структуры данных, что позволяет хранить любую информацию 10/10	Нет структуры данных, что позволяет хранить любую информацию 10/10
Сложность языка	Доступность материалов для изучения и большая база пользователей 5/10	Небольшое количество материалов, узконаправленные пользователи 8/10	Небольшое количество материалов, узконаправленные пользователи 9/10
Применение в области больших данных	Необходимо структурировать информацию 6/10	Возможность хранения структурированной и неструктурированной информации 9/10	Возможность хранения структурированной и неструктурированной информации 9/10
Скорость обработки данных	Необходимо учитывать особенности конструкций JOIN, операторов WHERE и HAVING 6/10	Обработка данных происходит по взаимосвязям между объектами 10/10	Обработка данных происходит по значениям «ключа» 10/10

Для сравнительного анализа были выделены основные пункты проектирования модели данных, к которым относятся: масштабирование системы, соответствие требованиям и стандартам, способность изменять структуры модели данных, сложность языка, производительность и скорость обработки данных, а также актуальная область развития информационных технологий – это

область применения выбранной модели системы в сфере больших данных.

Данный анализ применялся для реляционных, документных и графовых баз данных, и по результатам сравнения можно сделать вывод, что у каждой модели есть свои преимущества в определенной области.

Таким образом, для проектирования и создания эффективной аналитической системы необходимо точно определиться с исследуемой предметной областью и выбрать наиболее подходящую модель данных.

Выводы

Перед информационным пространством всегда был и остается вопрос анализа данных, количество которых не будет уменьшаться, а, наоборот, возрастает с каждым днем, что создает серьезную проблему обработки данных, и проблема эффективных технологий для решения данного вопроса становится очень важной.

Нельзя точно сказать и обобщить, какая модель – реляционная или графовая – будет лучше, все зависит от сферы применения. Необходимо четко оценивать, какие данные необходимо обработать, что они из себя представляют и т. д., чтобы выбрать ту или иную систему управления базой данных.

Реляционные базы данных рассчитаны на структурированные данные, они оптимальны для рутинных запросов и анализа информации, обеспечивая стабильность обработки транзакций, но страдают во время запросов на объединение. Графовые базы данных предназначены для больших объемов неструктурированных данных, делающих упор на взаимосвязи между объектами, но каждая СУБД имеет свой собственный язык манипулирования данными.

Это означает, что каждая из этих моделей отлично показывает себя в конкретной сфере, между ними нет конкурентной борьбы, а, наоборот, их совместное применение использует плюсы и минусы каждой стороны.

Список литературы

1. Рейтинг СУБД. URL: https://db-engines.com/en/ranking (дата обращения: 17.12.2022).

2. Миронов В. В., Юсупова Н. И., Шакирова Г. Р. Ситуационно-ориентированные базы данных: концепция, архитектура, XML-реализация // Вестник Уфимского государственного авиационного технического университета. 2010. Т. 14, № 2 (37). С. 233–244.

3. Салибекян С. М., Петрова С. Б. Объектно-атрибутная модель представления пространственно-временных отношений между объектами // Прикладная информатика. 2016. Т. 11, № 3 (63). С. 103–115.

4. Абрамский М. М., Тимерханов Т. И. Сравнительный анализ использования реляционных и графовых баз данных в разработке цифровых образовательных систем // Вестник Новосибирского государственного университета. Серия: Информационные технологии. 2018. Т. 16, № 4. С. 5–12.

5. Гасанов Э. Э. О сложности хранения и поиска информации // Интеллектуальные системы. 2006. Т. 10, № 1-4. С. 273–302.

6. Засядко Г. Е., Карпов А. В. Проблемы разработки графовых баз данных // Инженерный вестник Дона. 2017. № 1 (44). С. 24.

7. Плетнев А. А. Информационно-графовая модель динамических баз данных и ее применение // Интеллектуальные системы. Теория и приложения. 2014. Т. 18, № 1. С. 111–140.

8. Ломов П. А. Применение графовых СУБД в задачах анализа данных // Труды Кольского научного центра РАН. 2019. Т. 10, № 9-9. С. 137–145.

9. Дубровин А. С., Огородникова О. В. Моделирование работы графовых систем управления базами данных (СУБД) при решении задач анализа продолжительности времени обработки информации // Вестник Воронежского института ФСИН России. 2022. № 3. С. 49–54.

10. Bruggen R. V. Learning Neo4j. United Kingdom Livery Place: Birmingham B3 2PB, Published by Packt Publishing Ltd., 2014. 222 p.

11. Осипов Д. Л. Технологии проектирования баз данных. М.: ДМК Пресс, 2019. 498 с.

12. Сьоре Э. Проектирование и реализация систем управления базами данных. М.: ДМК Пресс, 2021. 466 с.

13. Прамодкумар Дж. С., Фаулер М. NoSQL: новая методология разработки нереляционных баз данных. М.: И. Д. Вильяме, 2013. 192 с.

14. Фрэнкс Б. Укрощение больших данных: как извлекать знания из массивов информации с помощью глубокой аналитики. М.: Манн, Иванов и Фербер, 2014. 352 с.

15. Jordan G. Practical Neo4j. 1st ed. United Kingdom: Published by Apress, 2015. 393 p.

16. Уорд Б. Инновации SQL Server 2019. Использование технологий больших данных и машинного обучения. М.: ДМК Пресс, 2020. 408 с.

17. Harrison G. Next Generation Databases. 1st ed. United States, CA: Published by Apress, 2015. 244 p.

18. Kemper C. Beginning Neo4j. United States: Published by Apress, 2015. 162 p.

19. Робинсон Я., Вебнер Д., Эифрем Э. Графовые базы данных. 2-е изд. М.: ДМК Пресс, 2016. 256 с.

20. Редмонд Э. Семь баз данных за семь недель. Введение в современные базы данных и идеологию NoSQL. М.: ДМК Пресс, 2018. 384 с.

21. Кравченко Ю. А. Задачи семантического поиска, классификации, структуризации и интеграции информации в контексте проблем управления знаниями // Известия Южного федерального университета. Технические науки. 2016. № 7 (180). С. 5–18.

References

1. Reiting SUBD [DB-Engines Ranking]. Available at: https://db-engines.com/en/ranking. (accessed 17.12.2022)

2. Mironov V. V., Yusupova N. I., Shakirova G. R. Situatsionno-orientirovannye bazy dannykh: kontseptsiya, arkhitektura, XML-realizatsiya [Situationally oriented databases: concept, architecture, XML implementation]. *Vestnik Ufimskogo gosudarstvennogo aviatsionnogo tekhnicheskogo universiteta = Bulletin of the Ufa State Aviation Technical University*, 2010, vol. 14, no. 2 (37), pp. 233–244.

3. Salibekyan S. M., Petrova S. B. Ob"ektno-atributnaya model' predstavleniya prostranstvenno-vremennykh otnoshenii mezhdu ob"ektami [Object-attribute model of representation of space-time relations between objects]. *Prikladnaya informatika = Applied Informatics*, 2016, vol. 11, no. 3 (63), pp. 103–115.

4. Abramsky M. M., Timerkhanov T. I. Sravnitel'nyi analiz ispol'zovaniya relyatsionnykh i grafovykh baz dannykh v razrabotke tsifrovykh obrazovatel'nykh sistem [Comparative analysis of the use of relational and graph databases in the development of digital educational systems]. *Vestnik Novosibirskogo gosudarstvennogo universiteta. Seriya: Informatsionnye tekhnologii = Bulletin of Novosibirsk State University. Series: Information Technologies*, 2018, vol. 16, no. 4, pp. 5–2.

5. Hasanov E. E. O slozhnosti khraneniya i poiska informatsii[On the complexity of storing and searching information]. *Intellektual'nye sistemy = Intelligent Systems*, 2006, vol. 10, no. 1-4, pp. 273–302.

6. Zasyadko G. E., Karpov A. V. Problemy razrabotki grafovykh baz dannykh [Problems of graph database development]. *Inzhenernyi vestnik Dona = Engineering Bulletin of the Don*, 2017, no. 1 (44), p. 24.

7. Pletnev A. A. Informatsionno-grafovaya model' dinamicheskikh baz dannykh i ee primenenie [Information graph model of dynamic databases and its application]. *Intellektual'nye sistemy. Teoriya i prilozheniya = Intelligent Systems. Theory and Applications*, 2014, vol. 18, no. 1, pp. 111–140.

8. Lomov P. A. Primenenie grafovykh SUBD v zadachakh analiza dannykh [Application of graph DBMS in data analysis problems]. *Trudy Kol'skogo nauchnogo tsentra Rossiiskaya Akademiya Nauk = Proceedings of the Kola Scientific Center of the Russian Academy of Sciences*, 2019, vol. 10, no. 9-9, pp. 137–145.

9. Dubrovin A. S., Ogorodnikova O. V. Modelirovanie raboty grafovykh sistem upravleniya bazami dannykh (SUBD) pri reshenii zadach analiza prodolzhitel'nosti vremeni obrabotki informatsii [Modeling of graph database management systems (DBMS) in solving problems of analyzing the duration of information processing time]. *Vestnik Voronezhskogo*

instituta FSIN Rossii = Bulletin of the Voronezh Institute of the Federal Penitentiary Service of Russia, 2022, no. 3, pp. 49–54.

10. Bruggen R. V. Learning Neo4j. United Kingdom Livery Place, Birmingham B3 2PB, Published by Packt Publishing Ltd, 2014. 222 p.

11. Osipov D. L. Tekhnologii proektirovaniya baz dannykh [Database design technologies]. Moscow, DMK Press Publ., 2019. 498 p.

12. S'ore E. Proektirovanie i realizatsiya sistem upravleniya bazami dannykh [Design and implementation of database management systems]. Moscow, DMK Press Publ., 2021. 466 p.

13. Pramodkumar J. S., Fauler M. NoSQL: novaya metodologiya razrabotki nerelyatsionnykh baz dannykh [NoSQL: a new methodology for the development of non-relational databases]. Moscow, I. D. Williams Publ., 2013. 192 p.

14. Frenks B. Ukroshchenie bol'shikh dannykh: kak izvlekat' znaniya iz massivov informatsii s pomoshch'yu glubokoi analitiki [Taming big data: how to extract knowledge from arrays of information using deep analytics]. Moscow, Mann, Ivanov and Ferber Publ., 2014. 352 p.

15. Jordan G. Practical Neo4j. 1st ed. United Kingdom, Published by Apress, 2015. 393 p.

16. Uord B. Innovations of SQL Server 2019. Ispol'zovanie tekhnologii bol'shikh dannykh i mashinnogo obucheniya [The use of big data technologies and machine learning]. Moscow, DMK Press Publ., 2020. 408 p.

17. Harrison G. Next Generation Databases. 1st ed. United States, CA, Published by Apress, 2015. 244 p.

18. Kemper C. Beginning Neo4j. United States, Published by Apress, 2015. 162 p.

19. Robinson Ya., Webner D., Eifrem E. Grafovye bazy dannykh [Graph databases]. 2nd ed. Moscow, DMK Press Publ., 2016. 256 p.

20. Redmond E. Sem' baz dannykh za sem' nedel'. Vvedenie v sovremennye bazy dannykh i ideologiyu NoSQL [Seven databases in seven weeks. Introduction to modern databases and NoSQL ideology]. Moscow, DMK Press Publ., 2018. 384 p.

21. Kravchenko Yu. A. Zadachi semanticheskogo poiska, klassifikatsii, strukturizatsii i integratsii informatsii v kontekste problem upravleniya znaniyami [Tasks of semantic search, classification, structuring and integration of information in the context of knowledge management problems]. *Izvestiya Yuzhnogo federal'nogo universiteta. Tekhnicheskie nauki = Proceedings of the Southern Federal University. Technical Sciences*, 2016, no. 7 (180), pp. 5–18.

Информация об авторах / Information about the Authors

Фаткуллин Руслан Владиславович,
преподаватель кафедры информационных систем и технологий, Уральский технический институт связи и информатики (филиал) Сибирского государственного университета телекоммуникаций и информатики,
г. Екатеринбург, Российская Федерация,
e-mail: buddhaeye13@gmail.com

Ruslan V. Fatkullin, Lecturer of the Department of Information Systems and Technologies, Ural Technical Institute of Communications and Informatics (branch) of the Siberian State University of Telecommunications and Informatics,
Ekaterinburg, Russian Federation,
e-mail: buddhaeye13@gmail.com

Кислицын Евгений Витальевич, кандидат экономических наук, доцент, доцент кафедры информационных систем и технологий, Уральский технический институт связи и информатики (филиал) Сибирского государственного университета телекоммуникаций и информатики,
г. Екатеринбург, Российская Федерация,
e-mail: johnkevek@mail.ru

Evgeny V. Kislitsyn, Cand. of Sci. (Economics), Associate Professor, Associate Professor of the Department of Information Systems and Technologies, Ural Technical Institute of Communications and Informatics (branch) of the Siberian State University of Telecommunications and Informatics,
Ekaterinburg, Russian Federation,
e-mail: johnkevek@mail.ru