

Н.Е. Тимофеева, К.А. Дмитриева

СРАВНИТЕЛЬНЫЙ АНАЛИЗ РЕЛЯЦИОННОЙ
И НЕРЕЛЯЦИОННОЙ МОДЕЛИ ХРАНЕНИЯ
СЛУЖЕБНОЙ ИНФОРМАЦИИ ЦЕНТРАЛИЗОВАННОЙ
РАСПРЕДЕЛЕННОЙ БАЗЫ ДАННЫХ

Рассматриваются модели представления данных в базах данных. Предложена логическая схема словаря распределенной базы данных. Проведен сравнительный анализ времени отклика словаря на часто используемые операции при разных моделях данных. Получено экспериментальное подтверждение эффективности использования нереляционной модели данных для словаря и сделаны выводы.

Ключевые слова: база данных, реляционные базы данных, NoSQL, документоориентированные базы данных, сравнительный анализ, планируемый эксперимент, словарь распределенной базы данных.

N.E. Timofeeva, K.A. Dmitrieva

COMPARATIVE ANALYSIS OF RELATIONAL
AND NoSQL MODELS OF SERVICE INFORMATION
STORAGE IN CENTRALIZED
DISTRIBUTED DATABASE

The models of data representation in databases are considered. The comparative analysis of the dictionary response time to frequently used operations in different data models is carried out. Experimental confirmation of the efficiency of using NoSQL data model for the dictionary is obtained and conclusions are drawn.

Keywords: database, relational databases, NoSQL, document-oriented databases, comparative method, experimental design techniques, DDB dictionary.

Введение

В современном мире хранение информации является неотъемлемой частью любого процесса. Так как доступ к хранимой информации должен осуществляться с минимальной задержкой по времени, то сегодня приходится выбирать и создавать решения, которые не только удовлетворяли бы потребностям пользователей, но и отвечали современным требованиям быстродействия системы.

В настоящее время наравне с классическими реляционными базами данных (БД) для хранения и управления информацией используют и нереляционные. Благодаря особенностям нереляционных БД их использование является эффективным решением для различных задач, в которых данные имеют слабоструктурированный или вовсе не структурированный формат.

Тимофеева Н.Е., Дмитриева К.А. Сравнительный анализ реляционной...

Для хранения служебной информации в программном комплексе «НЕКА» предусмотрен словарь. Так как словарь программного комплекса не представляет собой строго структурированные данные, то логическую структуру словаря можно представить разными моделями данных. Чтобы определить наиболее подходящую для этого модель, необходимо провести сравнительный анализ с использованием реляционной и нереляционной модели для хранения служебной информации.

В статье приводится сравнительный анализ быстродействия системы с использованием реляционной и нереляционной модели данных для формирования словаря.

Модели данных БД

Центральным звеном в проектировании любой БД, в том числе словаря распределенной базы данных (РБД), является выбор модели данных. От того, насколько правильно была выбрана модель, зависит производительность системы в целом. Так как на данный момент не существует однозначного определения термина «*модели данных*», то, говоря о модели данных, будем понимать совокупность трех компонентов, которые были формально определены Э. Коддом в докладе «Модели данных в управлении базами данных»:

- 1) множество информационных конструкций, допускаемых этой моделью;
- 2) множество допустимых операций над данными;
- 3) множество ограничений, наложенных на информационные конструкции [11].

Для более удобного восприятия информации разделим все множество моделей данных в БД на два типа: модели, которые предоставляют способы четкого структурирования данных, и модели, использующие неструктурированный подход (NoSQL).

К базам данных структурированного типа относят иерархическую, сетевую и реляционные модели [8]. Первые две в настоящее время не используются ввиду сложности их организации на физическом уровне [8, 12]. Реляционные же БД, напротив, получили большое распространение в информационном мире.

Реляционная модель – это совокупность взаимосвязанных таблиц, каждая из которых содержит информацию об объектах определенного типа: строка таблицы – данные об одном объекте, столбцы таблицы – различные характеристики этих объектов [2].

Неструктурированными данными будем называть информацию, которая либо не имеет заранее определенной структуры данных, либо не организована в установленном порядке. **Нереляционный способ структуризации данных** заключается в избавлении от ограничений при хранении и использовании информации. Его отличают исключение излишнего усложнения, высокая пропускная способность и неограниченное горизонтальное масштабирование [5].

Нереляционные БД принято разделять на четыре вида: «ключ-значение», документо-ориентированные, колоночные и графовые модели [1]. Каждая из моделей служит для определенного класса задач [6, 10].

Для выбора модели словаря необходимо сформировать представление о данных, хранение которых будет на него возлагаться. Словарь должен хранить служебную информацию о существующих в РБД узлах. Набор атрибутов при этом для каждого узла может отличаться. Также в словаре должна храниться информация о таблицах, их местонахождении, внешних связях, а также о возможной фрагментации и репликации.

Таким образом, очевидно использование реляционной системы управления базами данных (СУБД), однако стоит учитывать, что, помимо очевидных плюсов данной модели: табличный вид представления данных, строгие правила проектирования, базирующиеся на реляционной алгебре, интуитивно понятный механизм связей таблиц и др. [2], – есть и значительное ограничение: для организации узлов с разным количеством исходных данных необходимо будет записывать основные параметры в одну таблицу, а дополнительные поля – во множество связанных таблиц. Такой подход может сильно усложнить логику программного комплекса, затруднить дальнейшую работу с данными и увеличить время выполнения запроса [3].

Другим подходом к организации словаря РБД является использование документоориентированной модели, так как она позволяет хранить данные в виде коллекций документов, состоящих из набора полей, которые в общем случае необязательно будут одинаковыми, а в случае с реляционной моделью уберет излишне усложненную структуру. Кроме того, документоориентированная модель позволяет создавать вложенные документы и использовать сложные типы (массивы, строки и т.д.) [Там же]. Однако при проектировании логической схемы стоит учитывать, что связей между коллекциями должно быть минимальное количество (в идеале их вообще не должно быть) во избежание усложнения поиска нужной информации и увеличения времени на эту операцию.

Логическая структура словаря

Логическая модель БД (логическая структура БД) – это графическое представление структуры БД с учетом принимаемой модели данных, независимое от конечной реализации БД и аппаратной платформы [4]. Другими словами, она показывает, что хранится в БД (объекты, их атрибуты и связи между ними), какую модель данных предполагается использовать, но не опирается на конкретную реализацию СУБД.

В **реляционной модели** логическая структура представляет собой набор таблиц (отношений), каждая из которых является либо самим объектом, либо взаимосвязью между ними [2] (рис. 1). Графически ее представляют в виде прямоугольников, в которых отражаются собственное имя объекта и его атрибуты. На связи между объектами указывают стрелки.

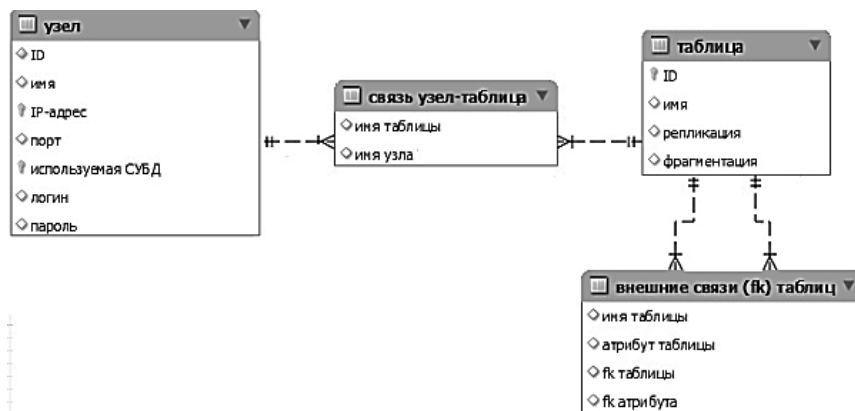


Рис. 1. Структура реляционного словаря

Тимофеева Н.Е., Дмитриева К.А. Сравнительный анализ реляционной...

В отличие от реляционных БД **документоориентированные** СУБД оперируют абстрактным понятием «документ». Хотя каждая их реализация определяет это понятие по-разному, в общем они все подразумевают инкапсуляцию и кодирование сохраняемой информации в некотором стандартном формате. Самые распространенные: XML, YAML, JSON, BSON [7]. Графически документы представляют прямоугольниками с наборами операндов, а связи, если такие имеются, – стрелками. Логическая схема нереляционного словаря представлена на рисунке 2.

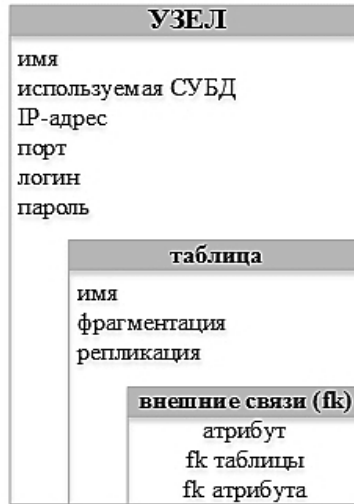


Рис. 2. Структура нереляционного словаря

Определение исследуемых операций

В процессе работы комплекс «НЕКА» многократно производит операции над словарем. Например, при добавлении нового узла в словарь добавляется служебная информация об узле. То же происходит и при добавлении таблицы. При запросе выборки вначале будет прочитано и отфильтровано содержимое словаря, и только после этого получена необходимая информация. Таким образом, производительность комплекса в целом зависит от быстродействия словаря.

Наиболее используемыми операциями над словарем, как уже отмечалось, являются чтение, запись и удаление служебной информации. Однако не стоит забывать об организации соединения комплекса со словарем перед осуществлением какой-либо операции или, если позволяет техническая реализация СУБД, одновременного подключения соединения на весь цикл работы.

Учитывая все перечисленные нюансы работы словаря, выделим четыре группы операций для проведения исследования:

- 1) установление соединения со словарем;
- 2) операции записи служебной информации в словарь;
- 3) операции удаления служебной информации из словаря;
- 4) операции чтения служебной информации из словаря.

Сравнительный анализ быстродействия реляционного и нереляционного словарей

Важнейшей частью научных исследований является построение математических моделей и численного эксперимента, результаты которого требуют дальнейшей обработки. Как правило, для решения этой задачи используют статистические методы планирования эксперимента, повышающие эффективность исследования, основанного на экспериментальном подходе, а также выявлении свойств исследуемых объектов и проверке справедливости гипотез.

Для осуществления сравнительного анализа согласно выбранным критериям был проведен планируемый эксперимент и построены регрессионные модели.

Стоит отметить, что быстродействие выбранных операций зависит от исходного объема словаря. Однако от количества пользователей время выполнения операций зависеть не будет, так как к словарю в централизованной РБД обращается только модуль управления программным комплексом. Вследствие этого исследование проводилось с использованием алгоритма тестирования на производительность [9] при количестве пользователей, равном 1.

В рамках проводимого эксперимента будем рассматривать в качестве реализации реляционного словаря СУБД MySQL 5.7 и PostgreSQL 9.3, в качестве нереляционного – MongoDB 3.6. Сравнительный анализ проводился с использованием полнофакторного эксперимента. На рисунках 3–9 приведены графики зависимостей реляционного и нереляционного словарей, где t – время выполнения соединения со словарем; N – исходный объем словаря.

Время соединения программного комплекса со словарем не зависит от его исходного объема. Из графика (см. рис. 3) видно, что соединение с MongoDB устанавливается быстрее, чем с другими рассматриваемыми СУБД. Наихудшим быстродействием обладает СУБД PostgreSQL.

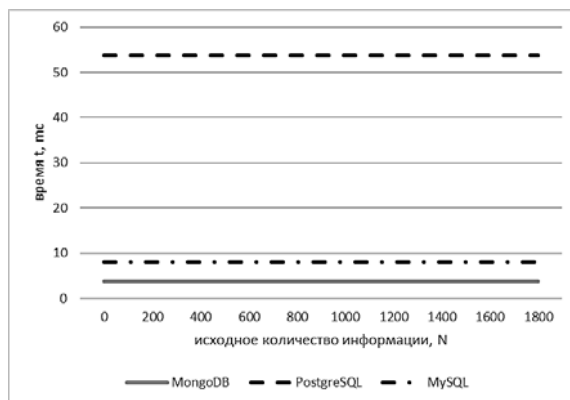


Рис. 3. Зависимости регрессионных моделей СУБД MongoDB $t(N) = 3,8$, MySQL $t(N) = 7,94$ и PostgreSQL $t(N) = 53,8$

Время выполнения операции записи служебной информации не зависит от исходного объема словаря (см. рис. 4–5), кроме более сложных случаев для нереляционной СУБД MongoDB (см. рис. 5). Это связано с тем, что при записи таблицы вначале выполняется выборка с уточнением информации о том, к какому «документу-узлу» данная таблица принадлежит, а после происходит сама запись. Самый плохой результат показала СУБД MySQL.

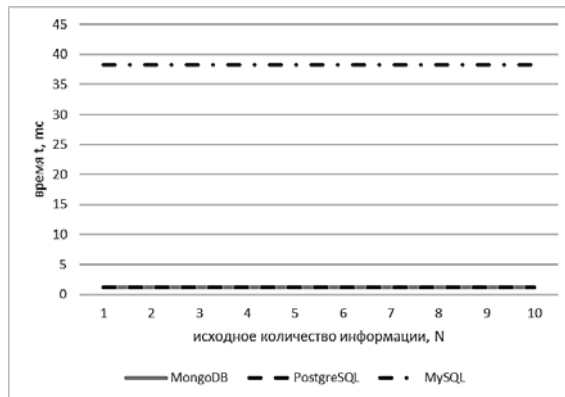


Рис. 4. Зависимости регрессионных моделей СУБД MongoDB $t(N) = 1,76$, MySQL $t(N) = 38,29$ и PostgreSQL $t(N) = 1,2$

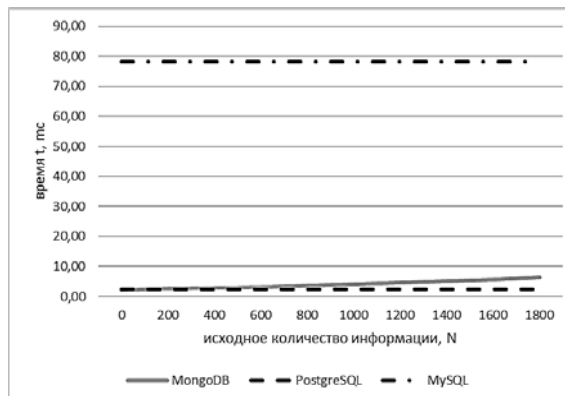


Рис. 5. Зависимости регрессионных моделей СУБД MongoDB $t(N) = 0,0000005x^2 + 0,001x + 2,252$, MySQL $t(N) = 78,26$ и PostgreSQL $t(N) = 2,38$

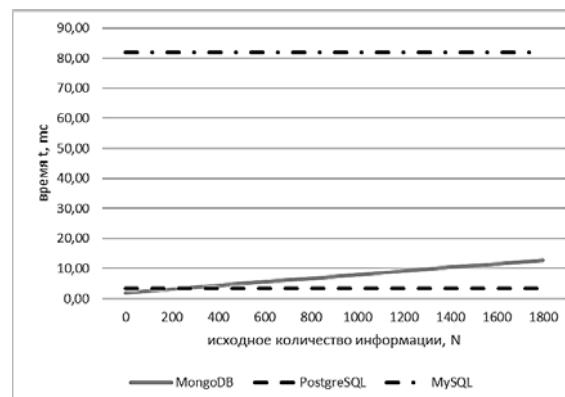


Рис. 6. Зависимости регрессионных моделей СУБД MongoDB $t(N) = 0,006x + 1,97$, MySQL $t(N) = 81,99$ и PostgreSQL $t(N) = 3,32$

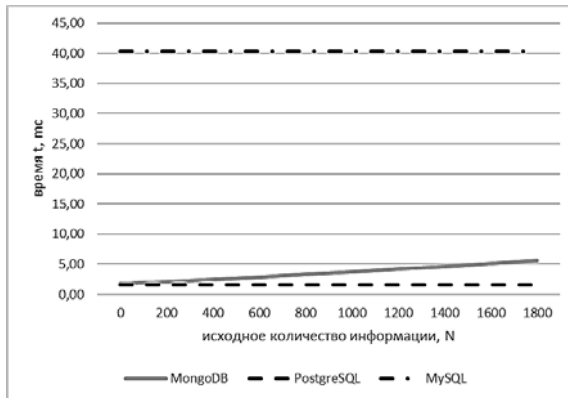


Рис. 7. Зависимости регрессионных моделей СУБД

MongoDB $t(N) = 0,0000002x^2 + 0,002x + 1,78$, MySQL $t(N) = 40,29$ и PostgreSQL $t(N) = 1,56$

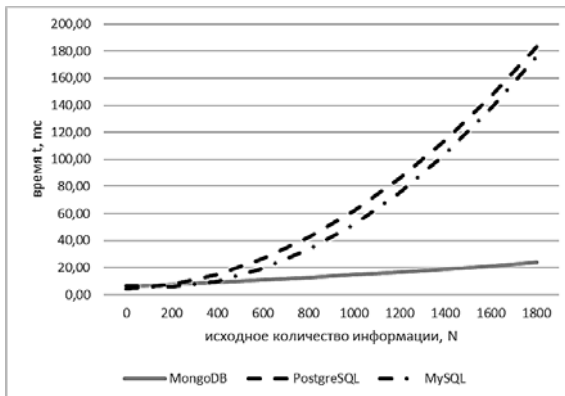


Рис. 8. Зависимости регрессионных моделей СУБД MongoDB $t(N) = 0,000001x^2 + 0,007x + 6,05$, MySQL $t(N) = 0,00006x^2 - 0,15x + 6,24$ и PostgreSQL $t(N) = 0,0001x^2 + 0,006x + 4,25$

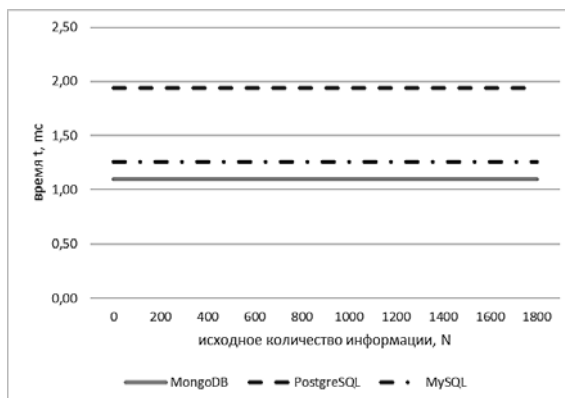


Рис. 9. Зависимости регрессионных моделей СУБД

MongoDB $t(N) = 1,1$, MySQL $t(N) = 1,26$ и PostgreSQL $t(N) = 1,94$

Тимофеева Н.Е., Дмитриева К.А. Сравнительный анализ реляционной...

Зависимость между удалением служебной информации и исходным объемом словаря для СУБД MongoDB квадратичная (см. рис. 6), а в более сложных случаях – линейная (см. рис. 7). У других исследуемых СУБД данная операция не зависит от выбранного параметра (см. рис. 6–7). Наиболее медленное выполнение удаления служебной информации было выявлено у СУБД MySQL.

Считывание всей служебной информации – довольно тяжелый процесс для любой СУБД, время выполнения которого будет расти в квадратичной зависимости от объема словаря (см. рис. 8). Наиболее быстро данную операцию выполняют СУБД MongoDB. У этой СУБД также самый малый коэффициент увеличения времени выполнения от объема словаря.

Как видно из графиков (см. рис. 9), время выполнения чтения узконаправленной информации не зависит от исходного объема. Под **узконаправленной информацией** понимаем выборку, состоящую из 1–2 строк, например определение узла, на котором находится таблица, наличие внешних связей у таблицы и т.д. MongoDB обладает самым высоким показателем быстродействия чтения узконаправленной информации. Наихудший результат показала СУБД PostgreSQL.

Заключение

По итогам проведенного исследования полученных регрессионных моделей времени выполнения операций словарем можно сказать, что у каждой СУБД есть проблемные зоны: MongoDB – удаление информации, PostgreSQL – соединение со словарем и чтение узконаправленной информации, MySQL – запись и удаление информации. Стоит отметить, что из исследуемых СУБД не рекомендуется использовать в качестве словаря РБД СУБД PostgreSQL, так как чтение со словаря происходит довольно часто, а, как было отмечено ранее, у данной СУБД этот параметр является слабой стороной.

СУБД MongoDB справляется быстрее со всеми исследуемыми операциями, кроме удаления. Поскольку данная операция используется намного реже, чем остальные, то есть основания полагать, что данную СУБД целесообразнее использовать в качестве словаря РБД. Также БД типа NoSQL предоставляют возможность параллельной обработки данных средствами парадигмы MapReduce [5], что впоследствии еще увеличит быстродействие словаря и всего комплекса в целом.

Литература

1. Гладкий М.В. Модель распределенных вычислений MapReduce // Труды БГТУ. Серия 6: Физико-математические науки и информатика. 2016. № 6 (188). С. 194–198.
2. Дейт К.Дж. Введение в системы баз данных / пер. с англ. 8-е изд. М.: Вильямс, 2005. 1328 с.
3. Дьяконов А.В., Козлова Ю.Б. О современных тенденциях хранения данных в документоориентированных СУБД // Актуальные проблемы авиации и космонавтики. 2015. № 11.
4. Змитрович А.И., Апанасович В.В., Скакун В.В. Базы данных и знаний: учеб. пособие. М.: Изд. центр БГУ, 2007.
5. Клеменков П.А., Кузнецов С.Д. Большие данные: современные подходы к хранению и обработке // Труды ИСП РАН. 2012. Т. 23. С. 143–158.
6. Кузнецов С.Д., Посконин А.В. Системы управления данными категории NoSQL // Программирование. 2014. Т. 40, № 6. С. 34–47.
7. Курцев Ю.В., Цапко Г.П. Виды NoSQL и их сравнение с реляционными базами данных // Молодежь и современные информационные технологии: сборник трудов XII Все-

российской научно-практической конференции студентов, аспирантов и молодых ученых (Томск, 12–14 ноября 2014 г.): в 2 т. Томск: Изд-во ТПУ, 2014. Т. 2. С. 42–43. URL: <http://www.lib.tpu.ru/fulltext/c/2014/C04/V2/015.pdf> (дата обращения: 01.02.2019).

8. Нестеров С.А. Базы данных: учебник и практикум для академического бакалавриата. М.: Юрайт, 2016. 230 с.

9. Тимофеева Н.Е., Гераськин А.С., Полулях К.А. Исследование и построение моделей нагруженного тестирования СУБД для повышения скорости и производительности распределенной вычислительной системы // Вестник Волгоградского государственного университета. Серия 1: Математика. Физика. 2017. № 1 (38). С. 75–89.

10. Фаулер М., Садаладж П.Дж. NoSQL: новая методология разработки нереляционных баз данных. М.: Вильямс, 2013. 192 с.

11. Codd E.F. Data Models in Database Management // Data Abstraction, Databases, and Conceptual Modelling: Workshop Proceedings (Pingree Park, Co., June 1980) / ed. by M.L. Brodie, S.N. Zilles. (ACM SIGART Newsletter № 74, January 1981; ACM SIGMOD Record 11(2), February 1981; ACM SIGPLAN Notices 16(1), January 1981).

12. Types of Database Models // Database Management System. URL: <http://www.databasemanagementsystems.com/types-of-database-models/> (date of the application: 01.02.2019).

Literatura

1. Gladkiy M.V. Model raspredelennykh vyichisleniy MapReduce // Trudy BGTU. Seriya 6: Fiziko-matematicheskie nauki i informatika. 2016. № 6 (188). S. 194–198.

2. Deyt K.Dzh. Vvedenie v sistemy baz dannykh / per. s angl. 8-e izd. M.: Vilyams, 2005. 1328 s.

3. Dyakonov A.V., Kozlova Yu.B. O sovremennykh tendentsiyah hraneniya dannykh v dokumentoorientirovannykh SUBD // Aktualnyye problemy aviatsii i kosmonavtiki. 2015. № 11.

4. Zmitrovich A.I., Apanasovich V.V., Skakun V.V. Bazy dannykh i znaniy: ucheb. posobie. M.: Izd. tsentr BGU, 2007.

5. Klemenkov P.A., Kuznetsov S.D. Bolshie dannie: sovremennyye podhody k hraneniyu i obrabotke // Trudy ISP RAN. 2012. T. 23. S. 143–158.

6. Kuznetsov S.D., Poskonin A.V. Sistemy upravleniya dannyimi kategorii NoSQL // Programirovanie. 2014. T. 40, № 6. S. 34–47.

7. Kurtsev Yu.V., Tsapko G.P. Vidy NoSQL i ih sravnenie s relyatsionnyimi bazami dannykh // Molodezh i sovremennyye informatsionnyye tehnologii: sbornik trudov XII Vserossiyskoy nauchno-prakticheskoy konferentsii studentov, aspirantov i molodykh uchenykh (Tomsk, 12–14 noyabrya 2014 g.): v 2 t. Tomsk: Izd-vo TPU, 2014. T. 2. S. 42–43. URL: <http://www.lib.tpu.ru/fulltext/c/2014/C04/V2/015.pdf> (data obrashcheniya: 01.02.2019).

8. Nesterov S.A. Bazy dannykh: uchebnyk i praktikum dlya akademicheskogo bakalavriata. M.: Yurayt, 2016. 230 s.

9. Timofeeva N.E., Geraskin A.S., Polulyah K.A. Issledovanie i postroenie modeley nagruzoch-nogo testirovaniya SUBD dlya povysheniya skorosti i proizvoditelnosti raspredelennoy vyichislitelnoy sistemy // Vestnik Volgogradskogo gosudarstvennogo universiteta. Seriya 1: Matematika. Fizika. 2017. № 1 (38). S. 75–89.

10. Fauler M., Sadaladzh P.Dzh. NoSQL: novaya metodologiya razrabotki nerelyatsionnykh baz dannykh. M.: Vilyams, 2013. 192 s.

11. Codd E.F. Data Models in Database Management // Data Abstraction, Databases, and Conceptual Modelling: Workshop Proceedings (Pingree Park, Co., June 1980) / ed. by M.L. Brodie, S.N. Zilles. (ACM SIGART Newsletter № 74, January 1981; ACM SIGMOD Record 11(2), February 1981; ACM SIGPLAN Notices 16(1), January 1981).

12. Types of Database Models // Database Management System. URL: <http://www.databasemanagementsystems.com/types-of-database-models/> (date of the application: 01.02.2019).