

УДК 004.65  
DOI: 10.15827/0236-235X.132.714-719

Дата подачи статьи: 10.04.20  
2020. Т. 33. № 4. С. 714–719

## **Сравнительный анализ СУБД для туристической социальной сети**

*Е.Ф. Феоктистов*<sup>1</sup>, студент, *freshstyler@mail.ru*

<sup>1</sup> *Волгоградский государственный технический университет,  
г. Волгоград, 400005, Россия*

Цифровые технологии широко используются во всех сферах жизнедеятельности человека, в том числе и в туризме. Бронирование готового тура, номеров в отеле и авиаперелетов можно сделать, не выходя из дома. Просмотр мобильных приложений с ценами на отели, билеты и туры стал частью ежедневной жизни туриста – даже не собираясь пока в поездку, он часто открывает данные приложения. Этими обстоятельствами и обусловлена идея создания мобильного приложения для туриста. Ниша туристической социальной сети, где можно общаться, планировать поездки, устраивать культурно-сувенирный обмен, пустует. В связи с этим идет разработка туристической социальной сети с рекомендательной системой на основе достаточно простых анкетных данных пользователей: списки сувениров на обмен (что есть, что хотят), ближайшие и совершенные поездки, город проживания и национальность.

Для решения данной задачи необходимо выбрать такую СУБД, которая, во-первых, должна быть масштабируемой в силу возможного большого притока пользователей разных стран, а во-вторых, отвечать современным требованиям, быть надежной и быстродействующей.

В статье анализируются различные виды нереляционных СУБД на опыте их использования в других социальных сетях. Описаны преимущества и недостатки систем при последующем возможном применении в туристической социальной сети. Также произведено тестирование трех графовых СУБД: Virtuoso, Neo4j и Sesame с целью выявления наиболее надежной и быстрой для данной разработки. На основе полученных данных определена лучшая СУБД, прошедшая большую часть тестов с наилучшими показателями времени.

**Ключевые слова:** *БД, социальная сеть, СУБД, туризм, разработка ПО.*

Цифровизация туризма в целом и повсеместное использование цифровых инструментов путешественниками обозначили роль информатики и анализа данных в туризме для понимания планирования поездок, а также для разработки и предоставления эффективных и действенных услуг в данной предметной области. Планирование поездки – это не только выбор места назначения, но и принятие решения о связанных ресурсах, таких как жилье, рестораны, музеи, транспорт или мероприятия [1]. Учитывая растущую роль информатики в туризме, нередко можно встретить термин «умный туризм». Умный туризм определяется как туризм, поддерживаемый интегрированными усилиями в пункте назначения по сбору и агрегации данных, полученных из физической инфраструктуры, социальных связей, правительственных или организационных источников и отдельных людей, в сочетании с использованием передовых технологий для преобразования этих данных [2].

Существует множество мобильных приложений, призванных решать те или иные задачи. Однако свободна ниша так называемой тури-

стической социальной сети, которая собрала бы в себе многие функции приложений, став одним из ведущих приложений умного туризма.

Данная социальная сеть, помимо основных функций, будет включать в себя функциональные возможности для культурно-сувенирного обмена: пользователи смогут составлять списки того, что у них есть (уникальные продукты, сувениры из города проживания, национальные одежды или музыкальные инструменты, возможность показать экскурсию и т.д.), и то, что они хотели бы получить. На основе этих данных будет разработана рекомендательная система, советующая туристам объекты для будущих поездок.

Однако существует проблема хранения этих данных и работы с ними. При проектировании социальной сети прежде всего нужно определиться, как именно будут храниться данные, в каком виде необходимо писать запросы и обращаться к ним. От правильного выбора СУБД зависит дальнейшая разработка приложения.

В последнее время большое распространение получили нереляционные БД NoSQL, кото-

рые используют нетрадиционный подход к хранению данных и доступу к ним. Они подразделяются на несколько видов в зависимости от структуры.

Следует заметить, что обязательное свойство, которым должна обладать выбранная для разработки СУБД, это масштабируемость, то есть способность системы справляться с увеличением рабочей нагрузки при добавлении ресурсов. В социальных сетях это важно, так как всегда сложно оценить рост и темп роста количества пользователей.

В последние годы был разработан ряд новых систем, обеспечивающих хорошую горизонтальную масштабируемость для простых операций чтения или записи БД, распределенных по многим серверам.

БД NoSQL характеризуют следующие ключевые функции:

- возможность горизонтального масштабирования пропускной способности простой операции на многих серверах;
- возможность реплицирования и распределения (разделения) данных по многим серверам;
- простой интерфейс или протокол уровня вызова (в отличие от привязки SQL);
- более слабая модель параллелизма, чем транзакции ACID (Atomicity, Consistency, Isolation and Durability – атомарность, непротиворечивость, изоляция и долговечность) большинства реляционных (SQL) систем БД;
- эффективное использование распределенных индексов и оперативной памяти для хранения данных;
- возможность динамического добавления новых атрибутов в записи данных.

Системы отличаются друг от друга. Они варьируются по функциональности от простейшего распределенного хеширования, поддерживаемого популярным кэшем с открытым исходным кодом memcached, до хорошо масштабируемых секционированных таблиц, поддерживаемых Google BigTable [3].

Важной особенностью всех БД NoSQL является горизонтальное масштабирование. Эта операция обычно называется OLTP (Online Transaction Processing – оперативная обработка транзакций). Она позволяет поддерживать большое количество простых операций чтения и записи в секунду.

Как правило, реляционные БД описывались принципом ACID во многом благодаря тому, что, избавившись от ограничений, заложенных в этом принципе, зародились БД NoSQL, кото-

рые способны достигать куда более высокой производительности и масштабируемости [4].

Принципу реляционных БД ACID часто противопоставляют принцип BASE: доступность, согласованность, последовательность. Доступность достигается за счет репликации. BASE исходит из парадигмы, в которой данные распределяются произвольно и, соответственно, невозможна их синхронизация, точные значения необязательны. Основная задача BASE в том, как обновлять распределенные данные, в то время как сами данные проходят много непредвиденных маршрутов [5].

Прежде всего БД NoSQL делятся на агрегатные и неагрегатные. Часто выделяют четыре типа: БД типа «ключ-значение», документо-ориентированные, со структурой семейства столбцов и графовые. Первые два типа являются сильно агрегатно-ориентированными, так как они состоят из множества агрегатов, каждый из которых имеет ключ или идентификатор, используемый для доступа к данным. Разница между ними в том, что документная база видит структуру агрегата, в то время как «ключ-значение» нет. БД типа «ключ-значение» чаще всего выполняют поиск агрегатов по ключу. В документной можно подать запрос, основанный на внутренней структуре документа.

Модель семейств столбцов представляет собой двухуровневую агрегатную структуру. В результате агрегат разделяется на группы столбцов, интерпретируя их как единицы данных в агрегате-строке. Это накладывает на агрегат структурные ограничения, но позволяет БД использовать эту структуру для улучшения доступа [6].

Популярные социальные сети, в том числе Twitter, Facebook и им подобные, обычно оперируют с конкретными видами запросов, для обработки которых традиционные реляционные модели представления неэффективны. В ряде случаев могут помочь графовые структуры, используемые сегодня в поисковых системах, например, для обработки графов знаний, собирающих информацию по запросу и отображающих ответы вместе со списком найденных сайтов. В графовой структуре используются объекты двух видов: именованные узлы и именованные дуги.

В качестве иллюстрации рассмотрим СУБД, которые работают с данными, представленными в разработанной консорциумом W3C спецификации RDF, имеющей вид триплета: субъект-предикат-объект. Все данные в RDF

можно изобразить в виде ориентированного графа, где объект и субъект – это узлы, а предикат – дуга от субъекта к объекту. Такой способ представления данных удобен для хранения данных социальных сетей и метаданных (вспомогательных данных). Например, если пользователь А следит за новостями пользователя Б, то из узла А идет дуга в узел Б с характеристикой «следить за новостями». Либо если от каждого пользователя какой-либо системы требуется указывать город своего проживания, то это можно отобразить следующим образом: «турист» (субъект) – «едет в» (предикат) – «город» (объект). Наличие двунаправленной связи позволит найти людей, живущих в том же городе, что и определенный пользователь, или всех пользователей, проживающих в данном городе. Поиск соседей или поиск наличия связи между элементами намного проще выполнять именно на графах, поскольку по сравнению с реляционными базами здесь нет необходимости выполнять ресурсоемкие операции объединения таблиц.

Таким образом, в разработке приложения были использованы графовые БД. Во-первых, они обеспечивают хорошую масштабируемость данных, что важно в аспекте проектирования социальной сети. Во-вторых, современные графовые СУБД способны работать со структурами, размеры которых превышают имеющийся объем оперативной памяти. Это позволяет обрабатывать большие потоки данных в реальном времени, что тоже является необходимой особенностью будущего приложения.

Выбор СУБД требует предварительного тестирования для выявления системы, наиболее адекватной запросам сервиса туристической социальной сети. Сегодня существует много различных графовых СУБД, пригодных для работы с социальными сетями, однако каждая имеет свои преимущества и недостатки.

Neo4j – одна из самых распространенных графовых СУБД с открытым исходным кодом, реализованная на Java. Данные хранит в собственном формате, специализированно приспособленном для представления графовой информации. Такой подход в сравнении с моделированием графовой БД средствами реляционной СУБД позволяет применять дополнительную оптимизацию в случае данных с более сложной структурой [7].

Virtuoso – прогрессивная и производительная СУБД, которая, помимо работы с данными в виде RDF, позволяет работать с SQL и XML.

Она обладает весьма гибкими конфигурациями сервера. Кроме того, имеется скрипт для выполнения массовой загрузки, заставляющий систему загружать данные порциями, ускоряя процесс загрузки целиком, а не по одному триплету. Благодаря тому, что Virtuoso написана на языке Си/C++, имеется возможность полностью контролировать все выделенное пространство памяти, не позволяя операционной системе его изменять, причем можно создавать дополнительные индексы для ускорения обработки запросов. В коммерческом издании системы есть возможность одновременного запуска сервера на нескольких машинах для обеспечения безопасности или для ускорения обработки. Каждая машина производит вычисления над своей порцией данных, после чего возвращает результаты заранее определенной главной машине в сети, которая собирает полученные результаты и формирует из них финальный ответ.

Sesame предназначена для хранения данных RDF, обработки SPARQL-запросов и операций с небольшими графами. СУБД не обладает обширным выбором настроек и использует все пространство памяти, которое было выделено JVM при запуске приложения, поэтому в тех случаях, когда приходится работать с большим графом, может понадобиться увеличение объема памяти JVM, если стандартного значения будет недостаточно. Интересной особенностью системы является возможность работы с двумя стратегиями представления графа внутри системы. Первая – Memory Store – предусматривает хранение целиком всего графа в оперативной памяти. Такой подход удобен при работе с небольшими графами, помещающимися в оперативной памяти. Вторая – Native Store – предлагает весь граф размещать на диске. При использовании этой стратегии можно создавать дополнительные индексы, что ускоряет работу [8].

Почти все производители реляционных СУБД избегают широкой огласки результатов тестов конфигураций, построенных на том или ином инструментарии SQL, и единственным открытым источником являются тесты TPC для дорогих высокопроизводительных систем, решающих ограниченный класс задач. Мир систем RDF открыт для исследований, экспериментов и тестов – легко можно найти результаты тестов на разных задачах, на компьютерах разной мощности и архитектуры, а главное, подобрать наиболее подходящий для решения конкретной прикладной задачи инструментальной работы с моделью RDF [9].

В проведенном тестировании использовался тест SP<sup>2</sup>Bench SPARQL Performance Benchmark [10]. Тест построен на модели известной библиотеки DBLP литературы по логическому программированию, однако по своей структуре приближен к социальным связям, используемым в социальных сетях. Тест разработан для оценки скорости SPARQL-запросов. В данном тестировании анализировались следующие СУБД: Virtuoso, Sesame, Neo4j.

Тестирование проходило на разных запросах и на разных размерах БД. Данные были представлены в формате N3. В качестве размера использовались сначала 10 тысяч триплетов, затем 1 миллион триплетов и, наконец, 80 миллионов (см. рисунок). Тест использовал 11 различных запросов, среди них

- простой запрос, время поиска которого не зависит от количества триплетов в БД (показан год публикации журнала);
- запрос с ветвистым обходом графа (выбираются все статьи с определенным свойством);
- объединение результатов (для каждого года возвращается набор авторов, которые выпускались в этом году, но не выпускались ранее);
- сортировка результатов (возвращаются первые 10 электронных изданий прошлого года, начиная с 51, отсортированные в алфавитном порядке) и др.

Аппаратные характеристики устройства для тестирования:

- процессор – Intel Pentium CPU 2117U @ 1.80 GHz;
- размер оперативной памяти – 4,00 Гб;
- операционная система – Windows 10 Pro x64;
- количество ядер – 2.

Результаты тестов по отдельным СУБД представлены в таблице.

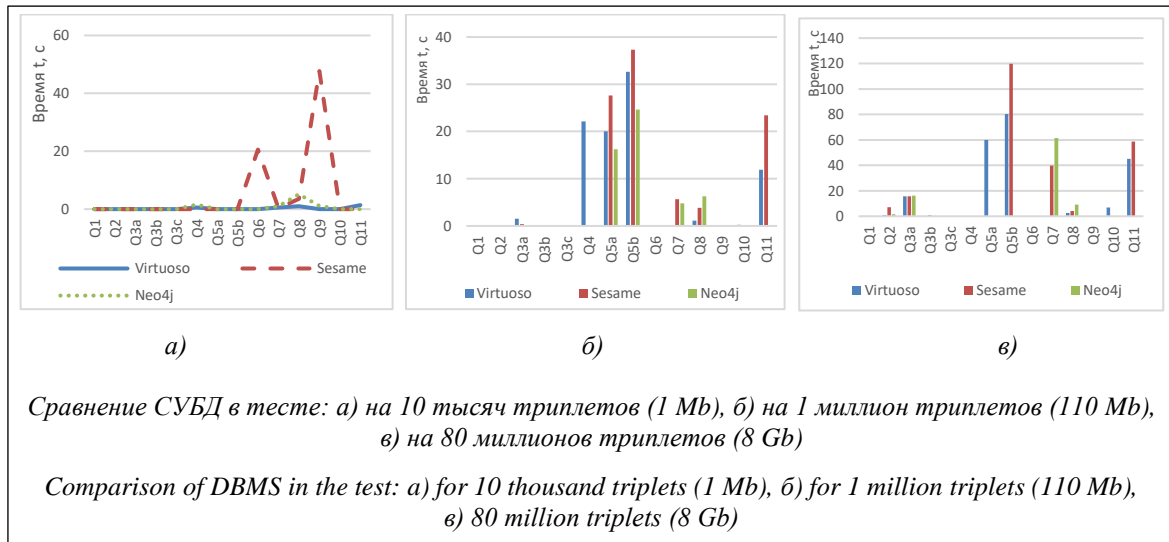
Как видно из данных результатов, меньше всего непройденных тестов оказалось у Virtuoso. СУБД Sesame не справилась с шестью тестами, а Virtuoso не прошла тест query 6, который ищет комплексную информацию и использует операцию объединения. Как уже говорилось, при разработке социальной сети у БД должна быть хорошая способность к масштабируемости, и поэтому важнее, как проходят тесты данные СУБД на больших графах. На них лучший результат опять же демонстрирует Virtuoso.

### Результаты тестов (сек.)

#### Test results (sec.)

Номер теста	10 тыс.	1 млн	80 млн
<b>СУБД Sesame</b>			
Q1	0,002	0,018	0,028
Q2	0,060	0,212	7,256
Q3a	0,031	0,480	15,672
Q3b	X	X	0,762
Q3c	X	X	X
Q4	X	X	X
Q5a	0,003	27,627	X
Q5b	0,002	37,331	119,820
Q6	20,567	X	X
Q7	0,317	5,663	39,820
Q8	3,526	3,821	4,026
Q9	47,700	X	X
Q10	X	X	X
Q11	0,002	23,414	58,799
<b>СУБД Neo4j</b>			
Q1	0,001	0,001	0,020
Q2	0,002	0,005	1,524
Q3a	0,019	X	16,087
Q3b	0,002	X	0,082
Q3c	X	X	X
Q4	1,574	X	X
Q5a	0,003	16,225	X
Q5b	0,003	24,612	X
Q6	X	X	X
Q7	0,988	4,758	61,547
Q8	5,141	6,245	9,147
Q9	1,125	X	X
Q10	X	X	X
Q11	0,005	X	X
<b>СУБД Virtuoso</b>			
Q1	0,002	0,002	0,002
Q2	0,002	0,007	0,900
Q3a	0,015	1,498	15,622
Q3b	0,004	0,060	0,073
Q3c	0,003	0,162	X
Q4	0,565	22,140	X
Q5a	0,003	20,056	59,920
Q5b	0,003	32,621	80,250
Q6	X	X	X
Q7	0,516	X	X
Q8	0,962	1,111	2,562
Q9	0,052	X	X
Q10	0,043	0,212	6,758
Q11	1,395	11,869	45,232

Можно отметить, что единственным невыполненным Virtuoso тестом по времени являлся тест, требующий хранения большого количе-



ства данных, тогда как Sesame не смогла пройти другие более простые тесты. Более того, Virtuoso загрузила БД быстрее, чем Sesame, лишний раз подтверждая превосходство при работе с базами, объем которых превышает доступную оперативную память.

### Заключение

Таким образом, по результатам данного тестирования самой приспособленной СУБД для работы с большими графами является Virtuoso;

Sesame показала лучшую стабильность при работе с небольшим количеством триплетов, а также продемонстрировала лучший результат времени на некоторых тестах. СУБД Neo4j не справилась с большим количеством тестов, несмотря на незначительный выигрыш времени над другими СУБД в тестах Q5a и Q5b, которые тестируют запросы с фильтрами.

Для разработки туристической социальной сети было принято решение использовать СУБД Virtuoso как более надежную и быструю при масштабировании.

Автор выражает благодарность за помощь в работе над написанием статьи кандидату технических наук Земцову А.Н., доценту Волгоградского государственного технического университета.

### Литература

1. Esmaeilia L., Mardania S., Alireza S., Golpayegania S.A.H., Madarb Z.Z. A novel tourism recommender system in the context of social commerce. *Expert Systems with Applications*, 2020, vol. 149. DOI: 10.1016/j.eswa.2020.113301.
2. Koo C., Cantoni L. Special issue on informatics/data analytics in smart tourism. *Information Processing and Management*, 2019, vol. 57, no. 4, pp. 1373–1375. DOI: 10.1016/j.ipm.2019.04.005.
3. Cattell R. Scalable SQL and NoSQL data stores. *SIGMOD Rec.*, 2011, vol. 39, pp. 12–27. DOI: 10.1145/1978915.1978919.
4. Stonebraker M. SQL databases v. NoSQL databases. *Commun. ACM*, 2010, vol. 53, pp. 10–11. DOI: 10.1145/1721654.1721659.
5. Chandra D.G. BASE analysis of NoSQL database. *Future Generation Computer Systems*, 2015, vol. 52, pp. 13–21. DOI: 10.1016/J.FUTURE.2015.05.003.
6. Агрегированные модели и NoSQL базы данных. URL: <https://oracle-patches.com/db/3633> (дата обращения: 29.03.2020).
7. Raj S. *Neo4j High Performance*. UK, Birmingham, Packt Publ., 2015, 192 p.
8. СУБД для социальных сетей. URL: <https://www.osp.ru/os/2014/02/13040051/> (дата обращения: 02.04.2020).
9. RDF – инструмент для неструктурированных данных. URL: <https://www.osp.ru/os/2012/09/13032513> (дата обращения: 02.04.2020).
10. The SP<sup>2</sup>Bench SPARQL Performance Benchmark. URL: <http://dbis.informatik.uni-freiburg.de/index.php?project=SP2B> (дата обращения: 05.04.2020).

## Comparative analysis of DBMS for tourist social network

E.F. Feoktistov<sup>1</sup>, Student, freshstyler@mail.ru

<sup>1</sup> Volgograd State Technical University, Volgograd, 400005, Russian Federation

**Abstract.** Digital technologies are widely used in all spheres of human life, including tourism. In order to book a ready-made tour no longer need to go to a travel agency; Hotel reservations abroad and air travel can also be done without leaving home. Viewing mobile applications with prices for hotels, tickets, and tours has become part of the daily life of a tourist – even when he is not going on a trip, he often opens these applications. So there was a goal to create a mobile application for tourists for daily use. The niche of the tourist social network where tourists could communicate, plan their trips, arrange a cultural souvenir exchange, is empty.

In this regard, a tourist social network is being developed with a recommendation system based on fairly simple personal data of tourists: lists of souvenirs for exchange (what is what they want); upcoming and past trips; the city of residence, and nationality.

There is a problem with choosing the database management system necessary for this problem. Firstly, it should be scalable, in view of the possible large influx of users from different countries. Secondly, it must meet modern requirements, be reliable and fast.

The paper analyzes various types of non-relational DBMS (database management systems), based on the experience of using them in other social networks. Their advantages and disadvantages are described for subsequent possible use in a tourist social network. Three graph DBMSs were also tested: Virtuoso, Neo4j, and Sesame to identify the most reliable and fastest DBMS for this development. As a result, on the basis of the data obtained, the best DBMS was revealed, which passed most of the tests with the best time results.

**Keywords:** database, social network, database management systems, tourism, software development.

**Acknowledgements.** The author expresses his gratitude to Ph.D. (Engineering) A.N. Zemtsov (Associate Professor, Volgograd State Technical University) for his help in writing the paper.

### References

1. Esmaeilia L., Mardania S., Alireza S., Golpayegania S.A.H., Madarb Z.Z. A novel tourism recommender system in the context of social commerce. *Expert Systems with Applications*, 2020, vol. 149. DOI: 10.1016/j.eswa.2020.113301.
2. Koo C., Cantoni L. Special issue on informatics/data analytics in smart tourism. *Information Processing and Management*, 2019, vol. 57, no. 4, pp. 1373–1375. DOI: 10.1016/j.ipm.2019.04.005.
3. Cattell R. Scalable SQL and NoSQL data stores. *SIGMOD Rec.*, 2011, vol. 39, pp. 12–27. DOI: 10.1145/1978915.1978919.
4. Stonebraker M. SQL databases v. NoSQL databases. *Commun. ACM*, 2010, vol. 53, pp. 10–11. DOI: 10.1145/1721654.1721659.
5. Chandra D.G. BASE analysis of NoSQL database. *Future Generation Computer Systems*, 2015, vol. 52, pp. 13–21. DOI: 10.1016/J.FUTURE.2015.05.003.
6. *Aggregated Models and NoSQL Databases*. Available at: <https://oracle-patches.com/db/3633> (accessed March 29, 2020) (in Russ.).
7. Raj S. *Neo4j High Performance*. UK, Birmingham, Packt Publishing, 2015, 192 p.
8. *DBMS for Social Networks*. Available at: <https://www.osp.ru/os/2014/02/13040051/> (accessed April 02, 2020) (in Russ.).
9. *RDF – a Tool for Unstructured Data*. Available at: <https://www.osp.ru/os/2012/09/13032513> (accessed April 02, 2020) (in Russ.).
10. *The SP<sup>2</sup>Bench SPARQL Performance Benchmark*. Available at: <http://dbis.informatik.uni-freiburg.de/index.php?project=SP2B> (accessed April 05, 2020) (in Russ.).

### Для цитирования

Феоктистов Е.Ф. Сравнительный анализ СУБД для туристической социальной сети // Программные продукты и системы. 2020. Т. 33. № 4. С. 714–719. DOI: 10.15827/0236-235X.132.714-719.

### For citation

Feoktistov E.F. Comparative analysis of DBMS for tourist social network. *Software & Systems*, 2020, vol. 33, no. 4, pp. 714–719 (in Russ.). DOI: 10.15827/0236-235X.132.714-719.