

М. М. Абрамский, Т. И. Тимерханов

*Казанский (Приволжский) федеральный университет
ул. Кремлевская, 35, Казань, 420008, Россия*

ma@it.kfu.ru, timurgrunge@gmail.com

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ИСПОЛЬЗОВАНИЯ РЕЛЯЦИОННЫХ И ГРАФОВЫХ БАЗ ДАННЫХ В РАЗРАБОТКЕ ЦИФРОВЫХ ОБРАЗОВАТЕЛЬНЫХ СИСТЕМ

Рассмотрены вопросы выбора варианта реализации базы данных при разработке цифровых образовательных систем. Проведен сравнительный анализ реляционного и графового подходов к хранению базовых сущностей. Продемонстрирован ряд преимуществ графовой модели в контексте удобства разработки и эффективности реализации запросов.

Ключевые слова: графовые базы данных, реляционные базы данных, Neo4j, цифровые образовательные системы.

Введение

В настоящее время лидирующее положение среди средств хранения данных занимают системы управления базами данных (СУБД), основанные на реляционном подходе¹ (рис. 1): на трех лидирующих позициях – реляционные СУБД Oracle, MySQL и MS SQL, на 4-м месте – также реляционная СУБД PostgreSQL; самая распространенная нереляционная СУБД MongoDB занимает 5-е место, на 6-м месте вновь находится реляционная БД DB2.

Реляционный подход был сформулирован в 1969–1970 гг. Э. Ф. Коддом [1]. Он заключается в хранении данных в таблицах (*реляциях*) с определенными между ними связями и ограничениями для поддержки целостности, корректности и избыточности данных.

Однако за последние несколько лет² начала расти доля рынка СУБД, реализующих альтернативные модели хранения данных, которые объединяют в класс нереляционных (NoSQL, “not only SQL”). Такие модели имеют преимущество перед реляционными БД при работе со специфическими структурами данных, в частности они могут превосходить реляционные СУБД в работе со слабоструктурированными данными или с данными, явным образом связанными с визуальным представлением информации (графы, диаграммы и др.). На рис. 2 представлена динамика роста популярности использования всех нереляционных СУБД независимо от модели, например, MongoDB – документоориентированная СУБД, Redis – СУБД, реализующая модель «ключ – значение», Neo4j – графовая БД.

¹ DB-Engines Ranking 2018. URL: <https://db-engines.com/en/ranking>

² Ranking of the most popular database management systems worldwide, as of February 2018. URL: <https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems>

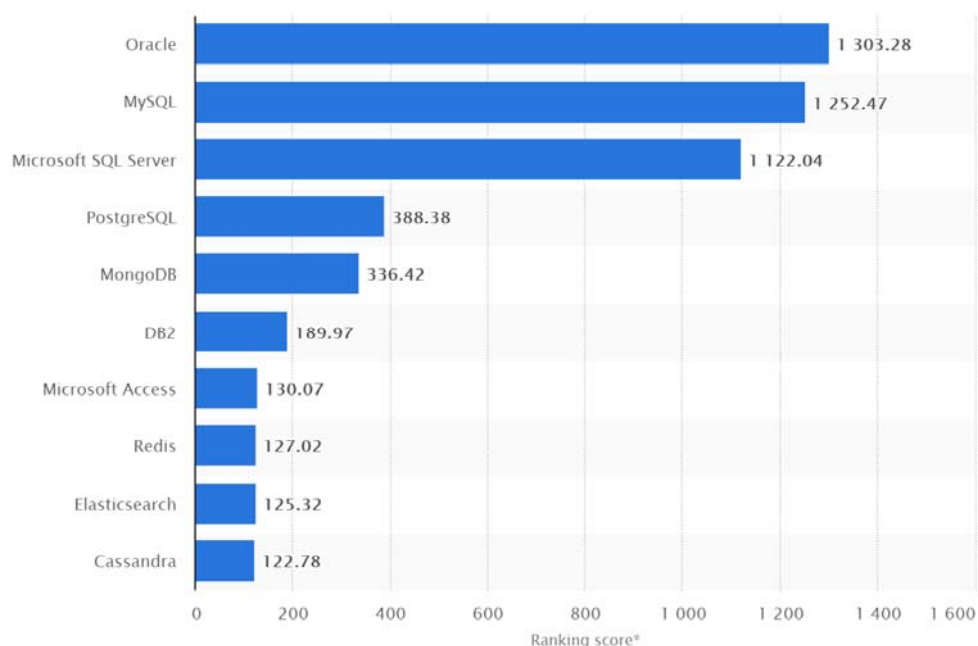


Рис. 1. Доли рынка, занимаемые системами управления базами данных
(URL: <https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems>)

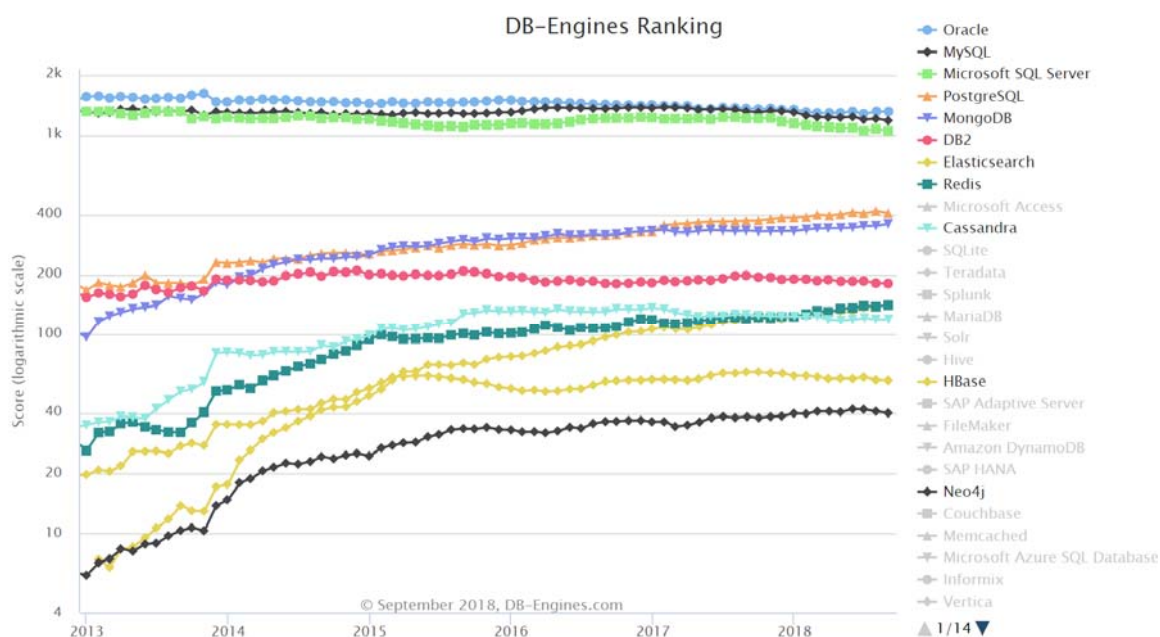


Рис. 2. Динамика использования СУБД с 2013 по 2018 г.
(URL: <https://www.statista.com/statistics/809750/worldwide-popularity-ranking-database-management-systems>)

Вопрос выбора подхода к организации базы данных актуален при разработке цифровых образовательных систем и сред. В последние годы в сфере образования наблюдается тенденция к индивидуализации [2], что означает организацию учебного процесса с учетом индивидуальных особенностей каждого обучающегося. В [3] показана возможность реализации такой индивидуализации с помощью специальных цифровых инструментов, которые анализируют перечень компетенций, достижений и характеристик обучающегося и генерируют его индивиду-

дуальную траекторию, представляющую собой граф. Предложенный способ генерации основан на обработке образовательных программ вуза. Сущность «образовательная программа» с точки зрения данных может быть представлена в виде связанных ориентированных графов, описывающих зависимости между учебными дисциплинами, компетенциями, модулями и др.

Авторами данной статьи было сделано предположение, что при работе с сущностями предметной области графовые базы данных могут иметь преимущество в эффективности работы и удобстве разработки в сравнении с реляционными. В данной работе рассмотрены реализации хранения данных, характерных для цифровых образовательных систем, в реляционной (PostgreSQL) и графовой (Neo4J) системах управления базами данных. Сравнительный анализ реляционных и графовых баз данных проводился в некоторых работах (например, [4]), нашей же целью является сравнительный анализ именно удобства использования реляционной и графовой моделей при разработке цифровых образовательных систем в контексте модели данных.

Общие сущности образовательных систем

Нами были проанализированы образовательные программы высшего и дополнительного образования и выявлены сущности баз данных, характерные для каждой образовательной системы. Любая информационная система, которую можно назвать образовательной, хранит данные, характерные для таких систем.

Отметим, что работа посвящена разработке образовательных систем с технической точки зрения, поэтому, несмотря на различия в образовательной терминологии, некоторые разные понятия были рассмотрены с точки зрения данных как одна и та же сущность. Именно поэтому ниже приведены все подходящие термины, а в скобках – вариант, выбранный в рамках данной статьи.

- **Модуль / предмет / дисциплина (модуль).** Содержит информацию об учебном предмете: название, цель, методы обучения. Термин *модуль* может также подразумевать некий обособленный раздел дисциплины, который можно изучать отдельно, например, модуль «Язык SQL» в курсе «Базы данных».

- **Образовательный результат / компетенция / знание, умение, навык (компетенция).** То, что может / чем владеет обучающийся по окончании образования. Если рассматривать высшее образование, то образовательными результатами являются компетенции ФГОС и связанные с ними знания, умения и навыки. Программы дополнительного образования, корпоративного образования, переподготовки могут также определять конкретные требования к сотрудникам, прошедшим обучение.

- **Направление подготовки / позиция / специальность (направление).** То, на что ориентировано образование – на получение диплома по конкретному направлению или, в случае дополнительного или корпоративного образования, на некоторую специализацию или соответствие некоторой вакансии на рынке труда.

Между выделенными сущностями можно определить следующие типы отношений:

- **Модуль – Модуль, связь «пререквизит».** Описывает зависимость между двумя модулями, когда изучение одного должно предшествовать изучению другого.

- **Модуль – Модуль, связь «иерархия».** Описывает зависимость между двумя предметами, когда один модуль является частью другого модуля (как показано выше с модулем «Язык SQL» в модуле «Базы данных»).

- **Модуль – Компетенция.** Компетенции / знания / образовательные результаты, получаемые при изучении предмета.

- **Направление – Компетенции.** Компетенции / знания / умения / навыки, которые необходимо освоить в рамках обучения по данному направлению.

Реализация модели данных в реляционных БД

Структурным аспектом реляционной модели данных являются отношения (таблицы) [1]. Для описания структуры реляционной базы данных, а также управления (создание, модифика-

ция, удаление) данными в уже существующем хранилище используется декларативный язык программирования Structured Query Language (SQL) ³.

Таблицы реляционных баз данных можно рассматривать в двух аспектах. С одной стороны, они могут пониматься как структуры для хранения выделенных сущностей. На этом понимании основан, в том числе, подход Object-Relational Mapping (объектно-реляционное отображение) ⁴, когда структура таблицы понимается как некоторый класс, столбец – как атрибут класса, а строка – как объект класса.

С другой стороны, реляционный подход к БД предполагает, что таблица хранит сами отношения между сущностями. Так, например, можно рассмотреть отношение «Подписка» между пользователями в социальных сетях. Подписка является не сущностью в привычном понимании, а отношением между двумя экземплярами сущности «Пользователь» (Пользователь1 подписан на Пользователь2).

Схема реляционной базы данных для модели данных представлена на рис. 3. Таблицы Module (модуль), Major (направление), Competence (компетенции) содержат сущности, описанные выше. Все связи являются отношением типа «многое-ко-многим» (Many-To-Many), что традиционно реализуется введением так называемых «промежуточных таблиц» для хранения этих связей. Кроме внешних ключей, эти таблицы могут содержать дополнительные данные. Например, таблица modules-major может содержать информацию о семестрах, в которых изучается модуль для конкретной образовательной программы. Отметим, что в рамках этой статьи отдельная спецификация таких данных не требуется.

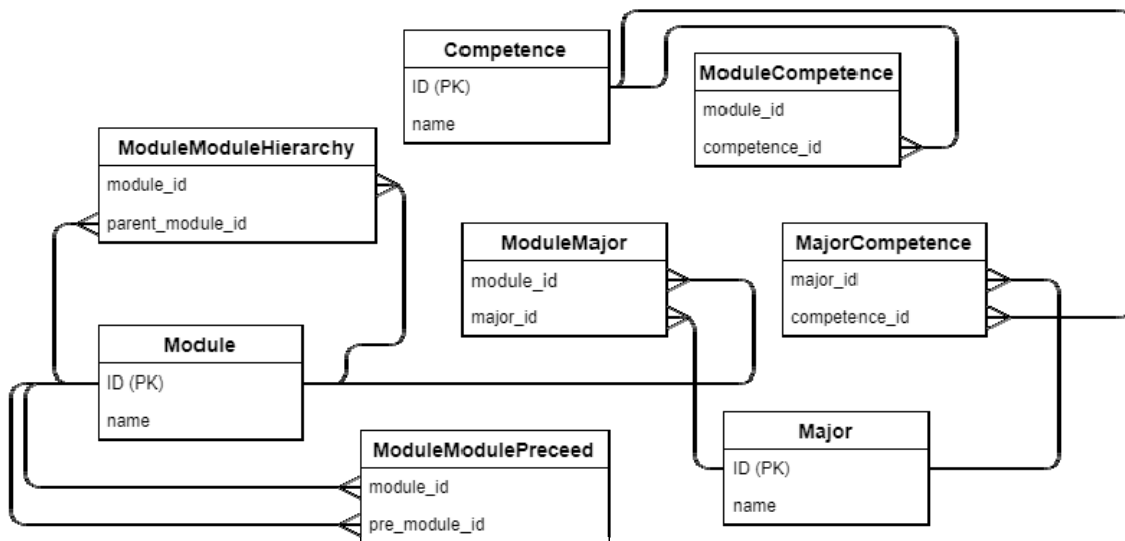


Рис. 3. Схема реляционной БД

Использование реляционного подхода для баз данных образовательных систем имеет несколько недостатков. Во-первых, реляционная схема не различает таблицы, хранящие сущности, и таблицы, хранящие связи между объектами. Во-вторых, визуализация результатов запросов к реляционным БД опять же выглядит как таблица, которая не является естественным способом визуализации данных, носящих графовый характер.

Не уменьшая достоинств реляционных БД, для повышения качества и удобства разработки цифровых образовательных систем необходимо обратиться к другим моделям хранения дан-

³ Information Technology – Database Language SQL. URL: <http://www.contrib.andrew.cmu.edu/~shadow/sql/sql1992.txt>

⁴ Mapping Objects to Relational Databases: O/R Mapping In Detail. URL: <http://www.agiledata.org/essays/mapping-Objects.html>

ных, имеющим собирательное название NoSQL (“not only SQL”). Они призваны не целиком заменить существующие реляционные решения, а дополнить их там, где они недостаточно гибки и удобны. Создатели нереляционных решений отмечают более высокую производительность при использовании специфических моделей данных и легкость работы с ними [5].

Одна из классификаций NoSQL базы данных приведена в [6]. Основываясь на модели данных, можно разделить нереляционные хранилища на 4 группы:

- 1) Базы данных NoSQL на основе модели «ключ – значение» (Redis, MemcacheDB и т. п.);
- 2) хранилища колонок (Cassandra, HBase);
- 3) документоориентированные базы данных (MongoDB, Couchbase);
- 4) графовые СУБД (OrientDB, Neo4J).

Графовые БД являются одним из наиболее популярных и актуальных подвидов нереляционных хранилищ [5]. Такие БД оперируют представлениями данных в виде графов и позволяют эффективно совершать привычные для них операции.

Использование графовых баз данных в цифровых образовательных системах

Для реализации хранения модели данных, описанной выше, была выбрана Neo4j – самая популярная на сегодняшний день графовая СУБД [6]. В основе ее работы лежит аппарат ориентированных графов. Узлы (nodes) хранят основную информацию об объектах и сгруппированы по видам с помощью специальных меток (labels). Также узлы связаны друг с другом отношениями (ребрами), в метках которых можно хранить данные⁵. Для манипуляций с данными используется язык Cypher, синтаксис которого достаточно близок к современным языкам разработки.

Как было отмечено ранее, представленная выше модель данных может быть описана ориентированным графом, что означает естественную возможность реализации в Neo4j. Для основных сущностей определены три метки: Модуль, Направление и Компетенция. Вершины разных меток могут хранить разный набор атрибутов.

Определены следующие связи между сущностями:

Связь	Описание
:ПРЕДШЕСТВУЕТ	Описывает зависимость между двумя модулями, когда один модуль необходим для освоения другого
:СОДЕРЖИТСЯ	Описывает зависимость между двумя модулями, когда один модуль является составной частью другого
:ИМЕЕТ_НАПРАВЛЕНИЕ	Связывает предмет и направление подготовки. Ребро также содержит свойства – первый и последний семестр преподавания предмета
:РАЗВИВАЕТ_КОМПЕТЕНЦИИ	Связывает предмет и компетенции. Описывает компетенции, которые являются результатом освоения предмета. Также связывает направление подготовки и компетенции. Описывает компетенции, которыми должен обладать выпускник данного направления

Для сравнительного анализа удобства использования реляционного и графового подходов в контексте разработки цифровых образовательных систем обратимся к конкретным примерам типовых запросов, которые могут быть использованы при работе с такими системами.

Запрос 1. Получить все модули, которые непосредственно (длина связей = 1) предшествуют (связь “пререквизит”) модулю “Управление программными проектами”.

⁵ Neo4J Basics. URL: <https://neo4j.com/product/>.

Реализация на языке SQL:

```
SELECT m.name FROM modules m
  INNER JOIN modules_modules_preceed m2 ON m.id = m2.module_id
  INNER JOIN modules m3 ON m2.pre_module_id = m3.id WHERE m3.name='Управление про-
граммными проектами'
```

Реализация на языке Cypher:

```
MATCH (steps)-[rels:ПРЕДШЕСТВУЕТ]->(m:Module) WHERE m.name='Управление программными
проектами' RETURN m, steps;
```

Запрос 2. Получить все модули, которые предшествуют (связь “пререквизит”) данному (за все время обучения).

Для сокращения запроса считаем, что id модуля заранее известен и равен 0.

Реализация на языке SQL:

```
WITH RECURSIVE traverse AS (
  SELECT pre_module_id FROM ModuleModulePreceed
  WHERE module_id = 0
  UNION ALL
  SELECT ModuleModulePreceed.from_subject_id
  FROM ModuleModulePreceed
  INNER JOIN traverse
  ON ModuleModulePreceed.module_id =
    traverse.pre_module_id
)
SELECT distinct pre_module_id, s2.name FROM traverse INNER JOIN Module s2
  ON traverse.pre_module_id = s2.id;
```

Реализация на языке Cypher:

```
MATCH (steps)-[rels:ПРЕДШЕСТВУЕТ*]->(m:Module) WHERE m.id=0 RETURN DISTINCT steps;
```

Приведенные примеры показывают, что обе модели данных эмулируют обход графа. Однако в силу своей структуры реляционная модель для прохода по графу требует явного рекурсивного прохода с операцией соединения на каждом уровне, при этом осуществляется многократный проход по одному и тому же набору записей. Графовые БД, опираясь на представление графа, естественным образом осуществляют переход от вершины к ее смежным вершинам за один шаг. Обратим внимание, что запросы 1 и 2 практически не отличаются. Это связано с тем, что Cypher позволяет указать максимальную и минимальную глубины связей, которые надо пройти. В данном случае эти значения не указаны – возвращаются все предки.

Помимо выигрыша в удобстве проектирования запросов к базе данных и в количестве операций, Neo4J предоставляет удобный интерфейс визуализации данных в запросах, встраиваемый автоматически в каждую БД. Так, выполняя запрос на получение предмета с названием «Системное программное обеспечение», можно также получить информацию о связанных узлах (рис. 4).

Представленная визуализация при работе с графовой БД позволяет видеть данные в естественном для них представлении, что облегчает восприятие данных и позволяет быстрее проверять корректность запросов.

Заключение

Проведенный анализ позволяет говорить о преимуществе в цифровых образовательных системах в ряде случаев графовых баз данных перед реляционными в контексте проектирования

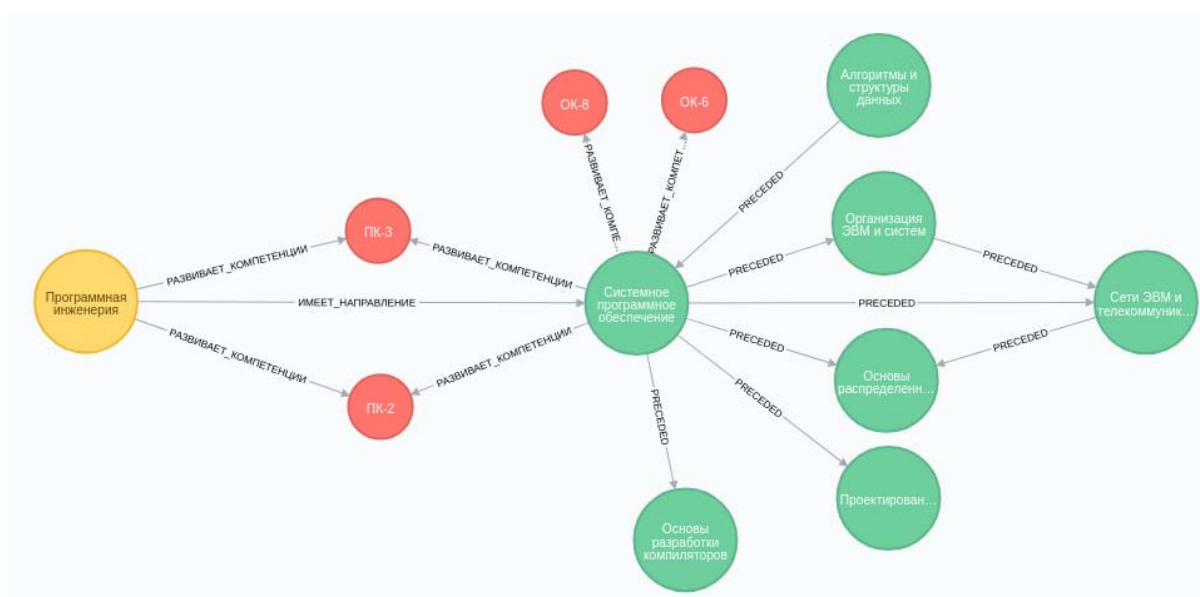


Рис. 4. Фрагмент структуры графовой базы данных

запросов, скорости их выполнения и удобства работы с самой базой данных. Возможно применение графовых БД с опорой на большое количество различных связей между объектами в образовательных системах, что позволяет рассматривать такие объекты как графы.

Еще раз подчеркнем, что графовая структура воспроизводима в реляционных базах данных, но для эффективной работы с «графовыми» данными в таких БД потребуются больше времени, более сложные запросы, а также, возможно, использование сторонних библиотек. Чем больше запрос использует графовую структуру, тем он становится сложнее и медленнее [7]. В дальнейших исследованиях предполагается продолжить анализ NoSQL-средств для работы с данными в цифровых образовательных средах, а также выполнить разработку специализированных расширений для графовых баз данных, позволяющих более эффективно разрабатывать соответствующие приложения в области образования.

Список литературы

1. Дейт К. Дж. Введение в системы баз данных. М.: Вильямс, 2006. 92 с.
2. Cheng Yin Cheong. New Paradigm for Re-engineering Education: Globalization, Localization and Individualization. Springer, 2005. 26 с.
3. Абрамский М. М., Батырова Э. Ф., Марданова А. Р., Ахметзянова Т. А. Генерация индивидуальных образовательных траекторий и расписания обучения в парадигме индивидуализации образования // Электронные библиотеки. 2018. Т. 21 (3–4). С. 129–145.
4. Мизюков Г. С., Сергеева Е. А., Чернов А. В. Технологии графовых баз данных для информационной поддержки электронного портфолио студентов // Успехи современной науки и образования. 2016. Т. 3, № 8. С. 57–63.
5. Бартнев М. В., Вишняков И. Э. Использование графовых баз данных в целях оптимизации анализа биллинговой информации // Инженерный журнал: наука и инновации. 2013. Вып. 11. URL: <http://engjournal.ru/catalog/it/hidden/1058.html>
6. Antonios Makris. A classification of NoSQL data stores based on key design characteristics // Procedia Computer Science. 2016. No. 97. P. 94–103.
7. Vicknair C., Macias M., Zhao Zhendong, Nan Xiaofei, Chen Yixin, Wilkins D. A Comparison of a Graph Database and a Relational Database // Proc. of the 48th Annual Southeast Regional Conference. 2010. Article No. 42.

M. M. Abramskiy, T. I. Timerkhanov

*Kazan Federal University
35 Kremlevskaya Str., Kazan, 420008, Russian Federation*

ma@it.kfu.ru, timurgrunge@gmail.com

COMPARATIVE ANALYSIS OF RELATIONAL AND GRAPH DATABASES USE IN DIGITAL EDUCATION SYSTEMS DEVELOPMENT

Questions on database implementation model choice for development of digital education systems are discussed. The comparative analysis of relational and graph approaches for main entities data storing. Several advantages of graph model in case of development usability and query efficiency are shown.

Keywords: Graph Databases, Relational Databases, Neo4j, Digital Education Systems.

References

1. Deit K. J. *Vvedenie v systemy baz dannykh*. Moscow, Viliams Publ., 2006, 92 p. (in Russ.)
2. Cheng Yin Cheong. *New Paradigm for Re-engineering Education: Globalization, Localization and Individualization*. Springer, 2005, 26 p.
3. Abramskiy M. M., Batyrova E. F., Mardanova A. R., Ahmetzyanova T. A. Generatsiya individual'nykh obrazovatel'nykh traektorij i raspisaniya obucheniya v paradigme individualizatsii obrazovaniya. *Elektronnye biblioteki*, 2018, Tvol. 21 (3–4), p. 129–145. (in Russ.)
4. Mizyukov G. S., Sergeeva E. A., Chernov A. V. Tekhnologii grafovykh baz dannykh dlya informatsionnoy podderzhki elektronnoho portfolio studentov. *Uspekhi sovremennoy nauki i obrazovaniya*, 2016, vol. 3, no. 8, p. 57–63. (in Russ.)
5. Bartenev M. V., Vishnyakov I. E. Ispol'zovanie grafovykh baz dannykh v tselyakh optimizatsii analiza billingovoy informatsii. *Inzhenernyj zhurnal: nauka i innovatsii*, 2013, no. 11. URL: <http://engjournal.ru/catalog/it/hidden/1058.html/> (in Russ.)
6. Antonios Makris. A classification of NoSQL data stores based on key design characteristics. *Procedia Computer Science*, 2016, no. 97, p. 94–103.
7. Vicknair C., Macias M., Zhao Zhendong, Nan Xiaofei, Chen Yixin, Wilkins D. A Comparison of a Graph Database and a Relational Database. *Proc. of the 48th Annual Southeast Regional Conference*, 2010, article no. 42.

Received 11.09.2018

For citation:

Abramskiy M. M., Timerkhanov T. I. Comparative Analysis of Relational and Graph Databases Use in Digital Education Systems Development. *Vestnik NSU. Series: Information Technologies*, 2018, vol. 16, no. 4, p. 5–12. (in Russ.)

DOI 10.25205/1818-7900-2018-16-4-5-12