

## СТАТЬИ

УДК 004.65:004.43

**СРАВНИТЕЛЬНЫЕ ХАРАКТЕРИСТИКИ SQL И NOSQL СУБД, ВЛИЯЮЩИЕ НА РАЗРАБОТКУ ПРИЛОЖЕНИЙ БАЗ ДАННЫХ**

**Куприянчик Е.М., Зудилова Т.В., Ананченко И.В., Осипов Н.А., Осетрова И.С.**  
*ФГАОУ ВО «Национальный исследовательский университет информационных технологий, механики и оптики», Санкт-Петербург, e-mail: kleila800@gmail.com, zudilova@ifmo.spb.ru, anantchenko@yandex.ru, nikita@ifmo.spb.ru, irina@ifmo.spb.ru*

Рассматриваются характеристики реляционных (SQL) и нереляционных (NoSQL) СУБД для определения параметров, существенно влияющих на разработку приложений баз данных. Цель выполненного исследования – проанализировать характеристики реляционных и нереляционных СУБД, влияющие на проектирование и разработку программного обеспечения, дать рекомендации о применимости систем. Выполнено проектирование схем баз данных для бизнес-процесса «Создание публикаций на веб-сайте» при помощи реляционного и нереляционного подходов на примерах двух СУБД: реляционной системы – MySQL и нереляционной системы – MongoDB. Две рассматриваемые СУБД с открытым исходным кодом обладают кроссплатформенностью, что является их преимуществом перед другими аналогичными системами. В работе проводится сравнение характеристик рассматриваемых СУБД. В частности, при проектировании и разработке схем баз данных рассматриваются параметры, которые могут повлиять на разработку программного обеспечения. Такое сравнение видится полезным для разработчиков будущих систем, так как выбор СУБД является одним из важнейших моментов в реализации проекта. В случае неверного выбора СУБД может существенно увеличиться время на разработку приложений, зачастую из-за ошибочных действий на этапе проектирования базы данных может пострадать ее функционал. Кроме того, неверный выбор СУБД может повлиять на дальнейшее масштабирование системы. От выбора СУБД зависят и этапы разработки запросов к данным, а именно извлечение, добавление и модификация данных, поэтому вопрос о выборе СУБД весьма актуален и должен рассматриваться до начала разработки системы.

**Ключевые слова:** БД, СУБД, реляционные СУБД, SQL, NoSQL, MySQL, MongoDB, проектирование

**COMPARATIVE CHARACTERISTICS OF SQL AND NOSQL DBMS AFFECTING THE DEVELOPMENT OF DATABASE APPLICATIONS**

**Kupriyanchik E.M., Zudilova T.V., Ananchenko I.V., Osipov N.A., Osetrova I.S.**  
*ITMO National Research University, Saint Petersburg, e-mail: kleila800@gmail.com, zudilova@ifmo.spb.ru, anantchenko@yandex.ru, nikita@ifmo.spb.ru, irina@ifmo.spb.ru*

The characteristics of relational (SQL) and non-relational (NoSQL) DBMS are considered to determine the parameters that significantly affect the development of database applications. The purpose of the study is to analyze the characteristics of relational and non-relational databases that affect the design and development of software, to make recommendations on the applicability of systems. Database schemas were designed for the business process “Creating publications on a website” using relational and non-relational approaches using the examples of two DBMS: a relational system – MySQL and a non-relational system – MongoDB. The two open-source DBMS under consideration are cross-platform, which is their advantage over other similar systems. The paper compares the characteristics of the considered DBMS. In particular, when designing and developing database schemas, parameters that may affect software development are considered. Such a comparison seems useful for developers of future systems, since the choice of a DBMS is one of the most important moments in the implementation of the project. If the DBMS is chosen incorrectly, the time for application development may significantly increase, often due to erroneous actions at the database design stage, its functionality may suffer. In addition, the wrong choice of DBMS can affect the further scaling of the system. The stages of data query development depend on the choice of DBMS, namely: data extraction, addition and modification, therefore, the question of choosing a DBMS is very relevant and should be considered before starting system development.

**Keywords:** Database, DBMS, relational DBMS, NoSQL, MySQL, MongoDB, design

В настоящее время практически невозможно найти автоматизированную систему без использования СУБД. От выбора определенного типа СУБД будет зависеть перечень задач и качество их решения при автоматизации задач на предприятии.

Наиболее часто в качестве СУБД выбирается реляционная система, которая идеально решает задачи обеспечения целостности данных и многие другие, но она уступает при работе с большими данными. При этом использование NoSQL баз дан-

ных (БД) позволяет работать с большими данными. Однако при использовании реляционных СУБД требуется тщательная подготовка на этапе проектирования системы, так как структура данных должна быть определена заранее. В свою очередь NoSQL СУБД не требуют такого тщательного процесса проектирования баз данных, потому что они работают с гибкими схемами, благодаря которым приложение можно реализовывать поэтапно. Цель выполненного исследования – про-

анализировать характеристики реляционных и нереляционных СУБД, влияющие на проектирование и разработку программного обеспечения, дать рекомендации о применимости систем.

*Сравнение характеристик SQL И NoSQL СУБД*

На данный момент наиболее используемыми типами баз данных являются реляционные (SQL) и нереляционные (NoSQL) БД. Как видно из табл. 1, самыми распространенными в 2022 г., согласно рейтингу DB-Engines, являются реляционные СУБД: Oracle, MySQL и Microsoft SQL Server [1].

В табл. 2 представлены популярные в 2022 г. нереляционные системы, где самыми популярными СУБД являются: MongoDB, Redis и Cassandra [1].

Рассмотрим характеристики, влияющие на разработку БД, при проектировании схем баз данных в реляционной и нереляционной СУБД на примерах двух систем: MySQL и MongoDB.

Основное различие между двумя системами: MySQL и MongoDB заключается в том, что при реляционном подходе необходима тщательная подготовка перед разработкой БД [2]. При проектировании реляционной схемы базы данных для бизнес-процесса «Создание публикаций на веб-сайте» проводится анализ предметной области, логическое и физическое проектирование. Также определяются сущности, необходимые для данного бизнес-процесса, их атрибуты, типы данных и связи между сущностями. В ходе выполнения этапов проектирования получена реляционная модель данных, представленная на рис. 1.

**Таблица 1**

Рейтинг популярности реляционных СУБД

Рейтинг		СУБД	Модель данных
июль 2022	июль 2021		
1.	1.	Oracle	Реляционная
2.	2.	MySQL	Реляционная
3.	3.	Microsoft SQL Server	Реляционная
4.	4.	PostgreSQL	Реляционная
5.	5.	IBM Db2	Реляционная
6.	7.	Microsoft Access	Реляционная
7.	6.	SQLite	Реляционная
8.	8.	MariaDB	Реляционная
9.	16.	Snowflake	Реляционная
10.	10.	Microsoft Azure SQL Database	Реляционная

**Таблица 2**

Рейтинг популярности нереляционных СУБД

Рейтинг		СУБД	Модель данных
июль 2022	июль 2021		
1.	1.	MongoDB	Документоориентированная
2.	2.	Redis	Ключ-значение
3.	3.	Cassandra	Широкий столбец
4.	4.	Amazon DynamoDB	Ключ-значение
5.	5.	Neo4j	Графовая
6.	6.	HBase	Широкий столбец
7.	7.	Microsoft Azure Cosmos DB	Документоориентированная
8.	8.	Couchbase	Документоориентированная
9.	9.	Memcached	Ключ-значение
10.	10.	Firebase Realtime Database	Документоориентированная

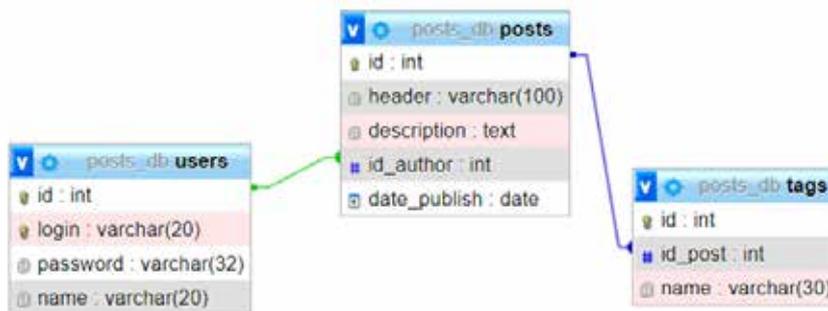


Рис. 1. Реляционная схема базы данных на примере MySQL



Рис. 2. Сопоставление таблиц MySQL с документами MongoDB

При использовании нереляционного подхода приложение можно проектировать сразу же без такой тщательной подготовки.

В MongoDB базы данных состоят из коллекций, которые являются эквивалентами таблиц в реляционных СУБД. В коллекциях хранятся данные в виде документов. Документ эквивалентен строке в таблице. Только в строке данные представлены в виде набора колонок, а в документе данные представлены в виде JSON-структуры (известной как BSON в MongoDB). И если в реляционных БД в строке имеются столбцы, то в MongoDB – поля. На рис. 2 представлены таблицы из MySQL и их отображение в MongoDB.

Еще одно из отличий MongoDB от MySQL заключается в том, что MongoDB обладает динамической схемой данных [3]. То есть один документ может иметь два поля, а второй – пять. Это позволяет легко добавлять и удалять поля. При использовании MySQL схему данных необходимо определить заранее, чтобы достичь согласованности. Также отметим, что при не-

реляционном подходе на тип данных поля не накладываются ограничения. Другими словами, одно и то же поле в одном документе может иметь данные типа int (целое значение), а в другом содержать тип array (массив).

Рассмотрим пример. В коллекции «users» можно создать документы с разной структурой. На рис. 3 представлены документы с разной структурой в одной коллекции.

```
{
  "_id": ObjectId("5ad774d8511dbd925ffc8b8d"),
  "login": "EKupriyanchik",
  "password": "225522",
  "name": "Elenak"
}

{
  "_id": ObjectId("5ad774d8511dbd925ffc8b87"),
  "login": "DSelez",
  "password": "334411dd",
  "name": "Selez",
  "age": 22,
  "favorite_themes": ["Телепередачи", "Кулинария"]
}
```

Рис. 3. Документы с разной схемой

Во второй документ добавили поля `age` и `favorite_themes`. При этом второе поле имеет несколько значений. При реляционном подходе в таблицу, которая хранит информацию о пользователях, необходимо было бы добавить поле `age`, а также создать как минимум одну таблицу для описания любимых тем пользователя. Таким образом, в виде документов можно эффективно реализовывать различные схемы, при этом при использовании реляционного подхода потребовалось бы несколько таблиц. Например, при проектировании в MySQL получилось три таблицы, а MongoDB позволяет несколько объектов объединить в один документ (рис. 4).

Еще одно отличие состоит в отображении связей. Если в реляционных СУБД используются первичные и внешние ключи, то в MongoDB нет прямого сопоставления, но используются ссылки на документы или вложенные документы (рис. 5).

Однако, так как в MongoDB нет ограничений и правил на внешние ключи, то необходимо аккуратно относиться к связям и операциям над ними. Например, поле `id_author` в документе из коллекции «posts» является просто полем, которое содержит

данные, и вся логика, связанная с их интерпретацией, должна быть реализована в приложении. То есть, если в `id_author` вставить ссылку на несуществующий документ из коллекции «users», то в MongoDB не возникнет ошибка о том, что соответствующая запись с таким идентификатором в коллекции «users» не была найдена. При этом в реляционной СУБД возникнет ошибка.

Для манипуляции данными MySQL использует язык SQL, а MongoDB – JavaScript [4]. Несмотря на разницу используемых подходов для работы с данными MongoDB отлично справляется со всеми типовыми задачами.

MongoDB обладает еще одним существенным преимуществом перед MySQL, заключающемся в функциях репликации и сегментирования, так как это решает проблему с масштабируемостью, характерную для реляционных систем [5]. При использовании реляционного подхода в основном реализуется возможность вертикального масштабирования.

Основные характеристики, влияющие на процесс разработки приложений баз данных, представлены в табл. 3.



Рис. 4. Объединение таблиц в один документ на примере MongoDB

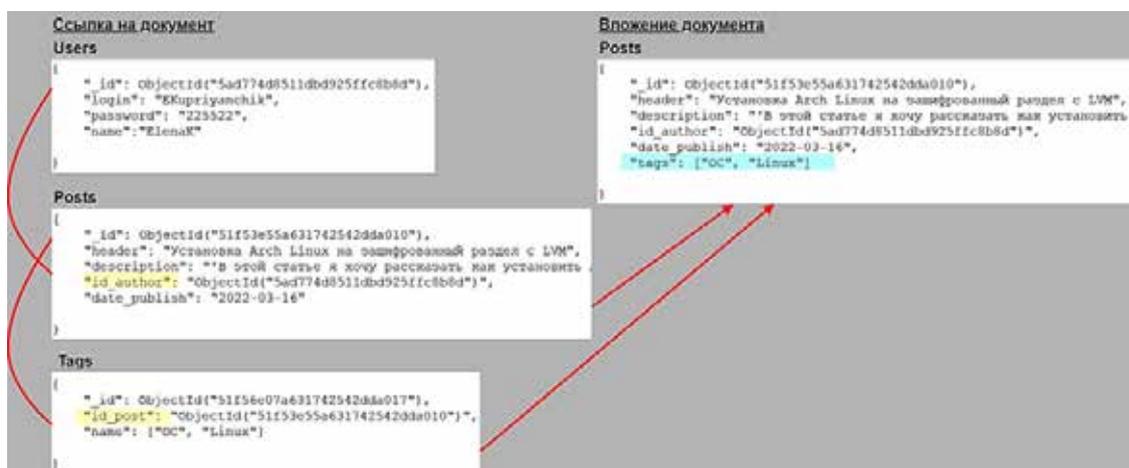


Рис. 5. Варианты связей между документами в MongoDB

Таблица 3

Сравнительные параметры реляционной MySQL и нереляционной MongoDB СУБД

Параметр	СУБД	
	MySQL	MongoDB
Динамическая схема	–	+
Создание связей СУБД	С помощью первичных и внешних ключей	С помощью ссылок на документы или вложенных документов
Объединение таблиц	+	Необязательно, в зависимости от схемы
Формат данных	Таблицы, состоящие из строк со столбцами	Коллекции, состоящие из документов с полями
Язык для манипуляции данными	SQL	JavaScript
Запросы на создание/изменение таблиц/ документов, чтение, вставку данных	+	+
Масштабируемость	Вертикальная	Горизонтальная

СУБД MySQL лучше выбрать при следующих условиях:

- имеются логические требования к данным, которые могут быть определены заранее;
- необходимо обеспечивать целостность данных;
- имеется достаточный опыт работы и хорошая техническая поддержка проекта;
- может понадобиться быстрый переход с одной СУБД на другую.

При выборе MongoDB целесообразно придерживаться следующих рекомендаций:

- при проектировании базы данных для обеспечения гибкости системы необходимо учитывать требования пользователей;
- объединение объектов в один документ следует производить при их совместном использовании, в других вариантах их желательно хранить в разных документах;
- допускается частичное дублирование данных, так как это может уменьшить время вычислений;
- объединение данных лучше выполнять при записи самих данных, а не при их чтении;
- необходима оптимизация схемы базы в соответствии с наиболее частыми запросами данных.

### Заключение

Представлен сравнительный анализ характеристик реляционных и нереляционных СУБД, влияющих на проектирование и разработку программного обеспечения, даны рекомендации о применимости систем. Анализ проводился на примерах двух СУБД: MySQL и MongoDB. Результат анализа будет полезен для разработчиков приложений, поскольку позволяет первоначально выбрать определенную технологию, однако позже при уточнении требований к системе, может использоваться и другой

подход. Показаны характеристики СУБД, по которым можно определить, какая система лучше подойдет для того или иного проекта. MySQL лучше выбрать для проекта, если имеется предопределённая структура и заданные схемы баз данных. Иначе следует выбрать MongoDB, потому что данная СУБД отлично подходит для быстрорастущих проектов без определенной схемы данных. Сравнение характеристик SQL и NoSQL СУБД, влияющих на разработку приложений баз данных, актуально, так как в настоящее время происходит импортозамещение и переход с иностранных сервисов на российские. Иностранные продукты заменяются либо на ПО с открытым исходным кодом, либо на существующие российские аналоги, либо создаются новые отечественные системы. Также согласно указу Президента РФ, уделяется достаточное внимание обеспечению мер для развития информационных технологий [6].

### Список литературы

1. DB-Engines Ranking [Электронный ресурс]. URL: <https://db-engines.com/en/ranking> (дата обращения: 04.07.2022).
2. Илюшечкин В.М. Основы использования и проектирования баз данных: учебник для академического бакалавриата. М.: Юрайт, 2019. 213 с.
3. Якушин А.Ю., Муковозов А.М., Исмоилов М.И. Сравнительный анализ реляционной базы данных и документоориентированной NoSQL базы данных в разрезе их применения при создании локального чата/мессенджера // Инновационная наука. 2018. № 4. С. 73–84.
4. Филиппов А.Н., Поволоцкий Я.А. Применение нереляционной СУБД MongoDB в САПР ТП. СПб.: Университет ИТМО, 2018. 46 с.
5. Шеннон Брэдшоу, Йон Брэзил, Кристина Ходоров. MongoDB: полное руководство. Мощная и масштабируемая система управления базами данных / Пер. с англ. Д.А. Беликова. М.: ДМК Пресс, 2020. 540 с.
6. Указ Президента РФ от 02.03.2022 № 83 «О мерах по обеспечению ускоренного развития отрасли информационных технологий в Российской Федерации» [Электронный ресурс]. URL: <http://publication.pravo.gov.ru/Document/View/0001202203020001> (дата обращения: 04.07.2022).