

УДК 004.652

Савоськин И. В., студент магистратуры кафедры управляющих электронно-вычислительных машин, МИРЭА, Россия, Москва

Фирсов А. О., студент магистратуры
кафедры промышленной информатики, МИРЭА, Россия, Москва

ИССЛЕДОВАНИЕ СПОСОБОВ ПРИМЕНЕНИЯ NOSQL И РЕЛЯЦИОННЫХ БАЗ ДАННЫХ

Аннотация: В статье рассматриваются два типа баз данных: реляционные базы данных и NoSQL базы данных. Рассматриваются основные принципы каждого из типов баз данных. Описывается существующая классификация. Приводятся и описываются существующие СУБД каждого из типов. Описываются преимущества и недостатки реляционных и NoSQL баз данных. На основе проведенных исследований делаются выводы о сфере использования каждого типа. Рассматривается необходимость в NoSQL базах и возможности замены реляционных баз данных ими.

Ключевые слова: база данных, система управления базами данных, NoSQL, реляционные базы данных, транзакция.

Abstract: The article discusses two types of databases: relational databases and NoSQL databases. The basic principles of each type of database are considered. Describes the existing classification. The existing DBMS of each type is described and described. The advantages and disadvantages of relational and NoSQL databases are described. Based on the research conducted, conclusions are drawn on the scope of use of each type. The need for NoSQL databases and the possibility of replacing relational databases with them is considered.

Key word: database, database management system, NoSQL, relational databases, transaction.

В современном мире особенно актуален вопрос хранения и использования данных. Экономическая и социальная сферы жизнедеятельности общества, неотъемлемой частью которых стали информационные технологии, являются источником огромных объемов данных. Свыше 90% всех данных в мире появились в период 2013-2014 годов и в будущем объем хранимых данных будет лишь увеличиваться. Ожидается, что в 2020 году созданный за год объем информации достигнет 40 зеттабайт против 2.8 в 2012 году [1]. В связи с этим возникает задача оптимальной систематизации и хранения данных в базах данных. С конца 80-х годов 20-го века реляционные базы данных занимают господствующую позицию на рынке [2] и до сих пор остаются самыми востребованными [3]. Однако реляционными базами данных выбор не ограничивается.

Реляционные базы данных основываются на предложенной доктором Эдгаром Коддом в 1970 году реляционной модели. К середине 1980-х годов РСУБД – реляционные системы управления базами данных — получили большую популярность.

Реляционная база данных представляет из себя набор данных с связями между ними. Данные организованы в таблицы, в столбцах которых хранятся типы данных, а в ячейках – значения. Строки при этом являются набором связанных значений, которые относятся к одному объекту и может быть помечена уникальным идентификатором – первичным ключом. Для связи нескольких таблиц используются внешние ключи. Таким образом реляционная БД хранит структурированные данные, структура которых не подвержена частым изменениям.

Правила, которым должны удовлетворять РСУБД, были сформированы Коддом и опубликованы в 1985-м году, хотя все популярные РСУБД не соответствуют многим из правил в связи с их высокой строгостью.

РСУБД, как транзакционные системы, основаны на принципах ACID [4]:

- Атомарность (atomicity) — свойство системы, которое гарантирует, что никакая транзакция не будет выполнена лишь частично. Согласно этому свойству, транзакция в системе является минимальной единицей выполнения, следовательно будут выполнены либо все входящие в транзакцию операции, либо не будет выполнена ни одна.

- Согласованность (consistency) — свойство, при котором транзакции, достигшие своего нормального завершения, не нарушают согласованности базы данных;

- Изолированность (isolation) — свойство системы ограничивать выполнение транзакций для недопущения их влияния на результаты других выполняемых транзакций;

- Надежность (durability) — результаты успешно завершенной транзакции сохраняются и будут доступны даже в случае аварийного прекращения работы.

Следование данным принципам позволяет прогнозировать поведение системы и обеспечить целостность БД.

Из всего вышеперечисленного можно сделать вывод, что основными преимуществами РСУБД являются целостность и структурированность данных, а также уменьшения вероятности неожиданного поведения системы.

Самыми популярными РСУБД являются:

- MySQL – самая используемая СУБД [3]. Используется такими компаниями как Facebook, Wikipedia, Twitter, LinkedIn, Alibaba и др. Из основных преимуществ можно выделить скорость и простоту;

- PostgreSQL – из основных преимуществ данной СУБД можно выделить надежность, легкую расширяемость, а также более подробную и качественную документацию, чем у MySQL;

- Microsoft SQL Server – данная СУБД обладает хорошей производительностью и масштабируемостью, а также удобным инструментарием для администрирования. Главным недостатком является

высокая стоимость, которая не позволяет использовать ее в малых или средних проектах;

- SQLite - на данный момент это одна из самых компактных СУБД. Позволяет хранить все данные в одном файле и, благодаря своему небольшому объему, отличается завидным быстродействием. Однако в ней отсутствуют система пользователей и возможность увеличения производительности;

- Oracle Database – данная система отличается стабильностью, обеспечивает хорошие безопасность и защиту данных. Главным ее недостатком, как и у SQL Server является высокая стоимость.

В последние пять лет растет популярность NoSQL баз данных. Изначально термин появился еще в конце 90-х годов 20-го века. Так называлась база данных, созданная Карло Строззи, которая хранила все данные в виде файлов, а вместо языка SQL для доступа к данным использовала сценарии командной строки (shell script). Однако понятие NoSQL баз данных, в том виде, в каком оно известно сейчас, сформировалось лишь в 2009 году. Название, которое изначально расшифровывалось как «No SQL», то есть «Без SQL», сейчас интерпретируется как «Not only SQL» — «не только SQL», и подразумевает под собой, что поддержка SQL является лишь одним из компонентов системы.

В отличии от реляционных СУБД, NoSQL основаны на принципах BASE [5]:

- Basic Availability - базовая доступность — каждый запрос гарантированно завершается (успешно или безуспешно).

- Soft State - гибкое состояние — состояние системы может изменяться со временем, даже без ввода новых данных, для достижения согласования данных.

- Eventual Consistency - согласованность в конечном счёте — данные могут быть некоторое время рассогласованы, но приходят к согласованию через некоторое время.

Исходя из этого можно понять, что в NoSQL базах более важными критериями являются скорость и гибкость, а не полная целостность данных.

Хранение данных в NoSQL базах данных различается в зависимости от типов:

- Key/value store (хранилища пар «ключ-значение») – представляют из себя хэш-таблицу, содержащую значения и соответствующие им ключи. Например, такие таблицы могут использоваться для хранения изображений или в качестве кэша объектов. Наиболее распространенными представителями являются Redis, Memcached, Riak.

- Документо-ориентированные базы – служат для хранения иерархических систем данных. По сути, представляют из себя усложненный вариант базы предыдущего типа, позволяя работать с хранящимся значением – прочитать или обновить его лишь частично. Чаще всего используются в издательском деле, документальном поиске или в системах управления контентом. Типичными представителями являются MongoDB, CouchDB, Elasticsearch.

- Колоночные базы данных – в отличие от реляционных БД хранят все данные не строках, а в колонках. Каждая колонка является как бы отдельной таблицей, хранящей только свои значения, обычно в отсортированном виде. Каждая колонка при этом хранится в отдельном файле. Это позволяет применять эффективное сжатие, так как каждая колонка содержит только один тип данных. Типичным применением этого типа СУБД является веб-индексирование или задачи с большими данными, но с пониженными требованиями к согласованности. Яркие представители: HBase, Cassandra, Vertica.

- Графовые базы данных – предназначены для обработки графов, например в социальных сетях, построении маршрутов, рекомендации товаров. Примерами таких СУБД являются Neo4j, OrientDB.

Также существуют мультимодельные базы данных, в которые входят две или более из представленных выше категорий.

Из представленных выше описаний типов баз данных следует вывод о том, что NoSQL базы данных являются неструктурированными. Это свойство NoSQL баз данных позволяет производить изменения содержимого базы с минимальными последствиями для кода разрабатываемого проекта. Также к преимуществам NoSQL БД можно отнести скорость работы. К тому же, в отличие от реляционных баз данных, в NoSQL базах можно хранить данные любых типов (в том числе и файлы), при этом скорость их обработки не сокращается. [6]. Еще одним достоинством можно считать скорость разработки – не требуется большого объема подготовительных действий.

Таким образом можно увидеть, что у обоих типов баз данных есть определенные преимущества и недостатки, которые делают каждый из типов более пригодным в том или ином проекте. Так, для хранения большого количества неструктурированных данных подойдет именно NoSQL база, так как она сможет обеспечить скорость работы, легкую масштабируемость при увеличении количества обрабатываемых данных, возможность использования облачных вычислений и хранилищ и более простую работу с ним, так как для узконаправленной работы не нужны серьезные познания. К тому же, в связи с отсутствием структурированности, NoSQL базы отлично подходят для проектов, которые очень быстро расширяются и постоянно меняют хранимые данные. При этом придется пожертвовать универсальностью, так как при переходе на другую СУБД, как реляционную, так и не реляционную, придется все создавать практически с нуля, так как разные NoSQL СУБД обладают разными API; надежностью системы и целостностью данных, так как в первую очередь обеспечивается скорость работы. Поэтому NoSQL БД подойдут для Big Data и документооборота. Также они отлично подходят для работы в связке с SQL базами в качестве кэш-хранилища.

Более же классическим и привычным вариантом все равно остается РСУБД. Во-первых, за счет своей универсальности, ведь все реляционные базы используют SQL, поэтому при необходимости изменение используемой СУБД не повлечет за собой чрезмерных затрат. Хотя реляционные СУБД, при

создании которых не подразумевалось, что придется хранить большие объемы данных, не обладают такой скоростью, как NoSQL базы, они обладают большей надежностью, предсказуемостью работы и целостностью данных за счет жесткого определения того, как именно транзакции взаимодействуют с базой. Структурированность, хотя и является несомненным достоинством данного типа баз, может являться и недостатком в конкретных проектах. Использоваться же реляционные базы данных могут в любых проектах, хотя в некоторых они и уступают NoSQL базам.

Подводя итог, можно сказать, что NoSQL базы данных не станут полноценной заменой реляционных баз данных. NoSQL БД реализуют принципиально иные парадигмы представления данных, и предназначены для работы в связке с SQL БД или же в определенных областях. Не стоит относиться к NoSQL как преемнику реляционных баз данных — скорее как к дополнению, полезному инструменту в определенных ситуациях.

Библиографический список:

1. Глумов, А. А. Теоретический аспект технологической составляющей новой индустриализации: промышленный интернет / А. А. Глумов // Вестник НГИЭИ.— 2018.— № 5.— С. 147-156.
2. OraclePatches [Электронный ресурс]: Режим доступа: <https://qps.ru/8zyo5> – Реляционная модель в сравнении с документоориентированной моделью. – (Дата обращения 28.05.2019).
3. TagLine [Электронный ресурс]: Режим доступа: <https://tagline.ru/database-management-systems-rating/> – Рейтинг систем управления базами данных (СУБД) 2016. – (Дата обращения 28.05.2019).
4. Gray, J. The Transaction Concept: Virtues and Limitations // Proceedings of the 7th International Conference on Very Large Databases. –Cannes, 1981. – P. 144-154.

5. Brewer, Eric A. Towards robust distributed systems // Proceedings of the XIX annual ACM symposium on Principles of distributed computing. — Portland, OR: ACM, 2000. — Vol. 19.

6. Савоськин, И. В. Экспериментальное исследование базы данных инструментального комплекса «Построитель тьюторов» / И. В. Савоськин, А. Д. Павлов // Вестник науки и образования.— 2018.— № 15 (51) часть 1. — С. 20-23.