|  |
| --- |
| ИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ |
| ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ |
| **«МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»** |
| **(МОСКОВСКИЙ ПОЛИТЕХ)** |
| Факультет информационных технологий |

Кафедра «Прикладная информатика»

Форма обучения: очная

**Лабораторная работа №6**

**Сравнение производительности запросов в реляционных и нереляционных базах данных**

**по дисциплине**

**«Хранилища данных»**

Студент (А. Костромин)

(личная подпись) (И. Фамилия)

**Москва 2021**

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc87920753)

[АНАЛИТИЧЕСКАЯ ЧАСТЬ 4](#_Toc87920754)

[1.1 Базы данных 4](#_Toc87920755)

[1.2 SQL против NoSQL 4](#_Toc87920756)

[1.3 Особенности SQL 6](#_Toc87920757)

[1.4 Базы данных NoSQL 7](#_Toc87920758)

[ЭКСПЕРИМЕНТАЛЬНАЧ ЧАСТЬ 9](#_Toc87920759)

[2.1 Эксперимент для 10 000 записей 9](#_Toc87920760)

[2.2 Эксперимент для 100 000 записей 12](#_Toc87920761)

[ЗАКЛЮЧЕНИЕ 14](#_Toc87920762)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 15](#_Toc87920763)

# ВВЕДЕНИЕ

Данные – самое главное богатство компании. Они являются источником для анализа, оценки и принятия решений. Сложное управление данными требует сложных структур хранения и принципов доступа. В данной работе рассматриваются архитектуры хранения баз данных, принципы и различия, описываются основные операции манипулирования - вставка, обновление, удаление и выборка. Для оценки были выбраны реляционные базы данных, такие как Oracle, MySQL и MS SQL, которые последовательно сравниваются в разделе производительности с нереляционными ориентированными базами данных – Mongo, Redis, GraphQl и Cassandra.

В настоящее время такие компании, как Google, Facebook, Twitter и другие, генерируют данные с помощью различных приложений. Кроме того, эти приложения генерируют огромное количество неструктурированных данных, что затрудняет их обработку. Данные очень важны для компаний, что также требует их анализа и хранения в определенных структурах. Реляционная база данных может работать только со структурированными данными, поэтому существует необходимость в системе управления базами данных NoSql, которая может работать с неструктурированными данными [1].

# АНАЛИТИЧЕСКАЯ ЧАСТЬ

## Базы данных

База данных – это набор структурированных данных или информации, хранящихся в компьютерной системе, так что компьютерная программа или человек могут использовать язык поиска для извлечения этой информации. Полученная таким образом информация может быть использована в процессе принятия решений.

Компьютерная программа, используемая для управления данными и запросами, называется СУБД (система управления базами данных). Компьютерные ученые могут классифицировать системы управления базами данных в соответствии с моделями баз данных, которые они поддерживают.

Первые упомянутые базы данных (реляционные) стали доминировать в 1980-х годах. Они поддерживают использование строк и столбцов в ряде таблиц. Значительное большинство из них также используют Sql для записи и запроса данных. Нереляционные базы данных стали популярными в 2000-х годах, их называют NoSql, поскольку они используют другой язык запросов.

## SQL против NoSQL

В данном пункте сравним базы данных, основанные на SQL и NoSQL. Наиболее существенное различие между этими концепциями заключается в том, что база данных SQL является реляционной и содержит внешние ключи. Напротив, база данных No SQL является необратимой и поэтому не определяет отношения. В Таблице 1 показаны различные особенности баз данных SQL и NoSQL [3].

Таблица 1

Особенности SQL и NoSQL

|  |  |  |
| --- | --- | --- |
| Свойство | SQL | NoSQL |
| Способ хранения данных | Таблицы | Документы, ключевое значение |
| Организация данных | Предопределенная схема | Динамическая схема |
| Масштабируемость (увеличение производительности) | Вертикальный (больше оперативной памяти, более мощный процессор) | Горизонтальный (больше серверов, экземпляров) |
| Язык запросов | Стандартизированный Sql | Собственный язык запросов |
| Взаимодействие с данными | Внешние ключи | Вложенные документы |
| Безопасность | Транзакции, согласованность, изоляция | Не существует |

На первый взгляд может показаться, что SQL перевешивает преимущества NoSQL, но это может оказаться не так. NoSQL не предлагает нам никаких функций безопасности, но больше считывает и записывает данные. Поэтому он используется для приложений BigData, где ожидаются терабайтные данные. NoSQL также является идеальным выбором, если нам нужно реализовать сложную полнотекстовую поисковую систему, которая учитывает похожие слова или другие грамматические правила. NoSQL также решает проблему, когда мы не знаем схем баз данных. Напротив, нам нужно решать проблему согласованности данных на стороне приложения, и в этом заключается ее недостаток. Еще одним недостатком является то, что база данных NoSQL не поддерживает транзакционную обработку [4].

Существует гораздо больше типов баз данных NoSQL, чем только те, которые имеют дело с полнотекстовым поиском. Они различаются как по стилю хранения, так и по использованию. Когда речь идет о полнотекстовом поиске, мы говорим о базах данных документов. Мы используем базу данных ключ-значение, потому что при работе с кэшем нам необходимо где-то хранить данные. Среди наиболее известных распространены Redis или Memcached [5].

## Особенности SQL

В Таблице 2 представлены различия между БД SQL.

Таблица 2

Различия в базах данных SQL

|  |  |  |  |
| --- | --- | --- | --- |
| Продукт/характеристика | Microsoft SQL | Oracle RDBMS | MySQL |
| Типовые применения | SharePoint, SCOM, SCCM, WSUS | OBI, SAP | Joomla, WordPress, MyBB, phpBB, Drupal |
| Операционная система | Windows Server, Windows Client | Windows, Unix, Linux | Windows, Unix, Linux, Mac и многое другое |
| Драйверы | ODBC, JDBC, ADO.NET, OLEDB, Microsoft Visual Studio | - | ODBC, JDBC, ADO.NET, Microsoft Visual Studio |
| Лицензирование | Закрытый исходный код, проприетарный | Закрытый исходный код, проприетарный | GNU-GPL с открытым исходным кодом |
| Стандартизация | ANSI-SQL | - | ANSI-SQL |
| Транзакции | Да | Да | Использование механизма хранения данных InnoDB |
| Частичный индекс | Да | Да | Нет |
| Схема | Да | Да | Нет |
| Вычисляемые столбцы | Да | Да | Нет |
| Активная/активная кластеризация | Только для чтения на втором узле | Да (RAC) | Нет |
| Планирование работы | Да | Да (Oracle Scheduler) | Да (планировщик событий) |
| История | 1989 | 1979 | 1995 |

## Базы данных NoSQL

Разработчикам нужны решения, соответствующие реалиям современных данных и итеративной практике разработки программного обеспечения. В последние годы базы данных NoSQL появились как ответ на ограничения традиционных реляционных баз данных, а также как обеспечение производительности, масштабируемости и гибкости, необходимых в современных приложениях [4].

Большинство аспектов этих технологий NoSQL сильно различаются и имеют мало общего, за исключением того, что они не используют реляционную модель данных. Существует четыре типа систем управления базами данных NoSQL:

* Key-Value Store – имеет большую хэш-таблицу ключей и значений (например, Riak, Amazon S3 Dynamo);
* хранилище на основе документов – хранит документы, состоящие из помеченных элементов (например, CouchDB);
* хранилище на основе столбцов – каждый блок хранилища содержит данные только из одного столбца (пример, HBase, Cassandra).
* основанная на графике – сетевая база данных, которая использует ребра и узлы для представления и хранения данных (например - Neo4J).

Рассмотрим немного подробнее два из четырех типов БД, которые будут использоваться в экспериментах со сравнением.

1. **Хранилище на основе документов**

Данные, представляющие собой набор пар ключ-значение, сжимаются в виде хранилища документов, очень похожего на хранилище ключей-значений, с той лишь разницей, что хранимые значения (называемые "документами") обеспечивают определенную структуру и кодировку управляемых данных. Некоторые распространенные стандартные кодировки данных БД: XML, JSON (JavaScript Object Notation), BSON (это двоичная кодировка объектов JSON).

Примеры документоориентированных БД:

* MongoDB;
* CouchBase.

1. **Базы данных NoSQL на основе графиков**

Графовая база данных – разновидность баз данных с реализацией сетевой модели в виде графа и его обобщений. Графовую модель данных обычно рассматривают как обобщение RDF-модели или сетевой модели данных. Основными элементами модели являются узлы и связи.

Графовые базы данных применяются для моделирования социальных графов (социальных сетей), в биоинформатике, а также для семантической паутины. Для задач с естественной графовой структурой данных графовые СУБД могут существенно превосходить реляционные по производительности, а также иметь преимущества в наглядности представления и простоте внесения изменений в схему базы данных.

Примеры графовых БД:

* Neo4j;
* GraphQL.

# ЭКСПЕРИМЕНТАЛЬНАЧ ЧАСТЬ

Для проведения сравнения были выбраны представители как реляционных, так и нереляционных баз данных. Среди реляционных БД: Oracle, MySQL, а среди нереляционных: MongoDB как представитель документоориентированной БД и Neo4j как представитель графовой БД.

Данными для эксперимента являются сведения о прибытии и отправлении всех коммерческих рейсов в США с октября 1987 года по апрель 2008 года. Это большой набор данных: всего около 120 миллионов записей, занимающих 1,6 гигабайта сжатого пространства и 12 гигабайт в несжатом виде. Для каждого представителя баз данных данные были предварительно переведены к соответствующему виду.

## Эксперимент для 10 000 записей

В первом эксперименте было проведено сравнение времени работы операций insert, update, delete и select. Эксперимент проводился при размере диапазона 10 000 записей в БД. Результаты проведения сравнения представлены в Таблице 3.

Таблица 3

Производительность запросов к базам данных с 10 000 записями в миллисекундах

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Тип/операция** | **Oracle** | **MySql** | **Mongo** | **Neo4j** |
| **insert** | 0.076 | 0,093 | 0.005 | 0.008 |
| **update** | 0.077 | 0,058 | 0.008 | 0.007 |
| **delete** | 0.059 | 0.025 | 0.01 | 0.017 |
| **select** | 0.025 | 0,093 | 0.009 | 0.010 |

Рассмотрим визуализацию получившихся результатов в виде графиков. На Рисунках 1-4 представлено сравнение скорости работы выбранных баз данных для каждой операции в отдельности.

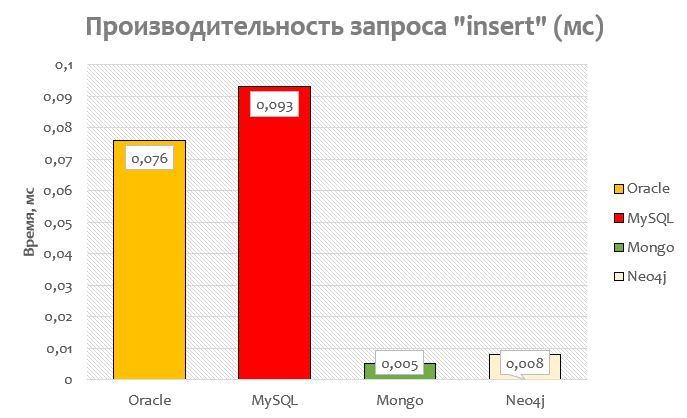


Рисунок 1 – Производительность запроса insert

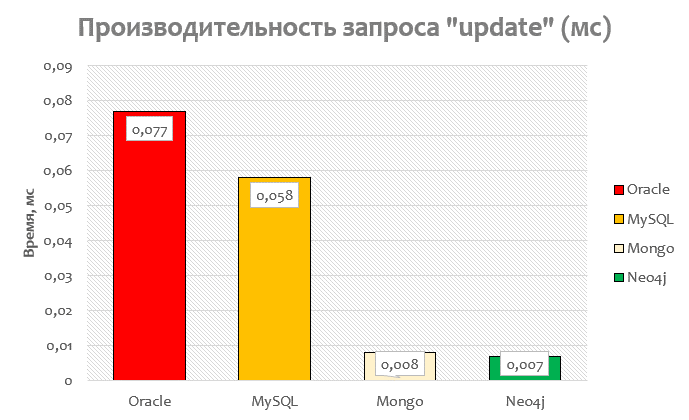


Рисунок 2 – Производительность запроса update

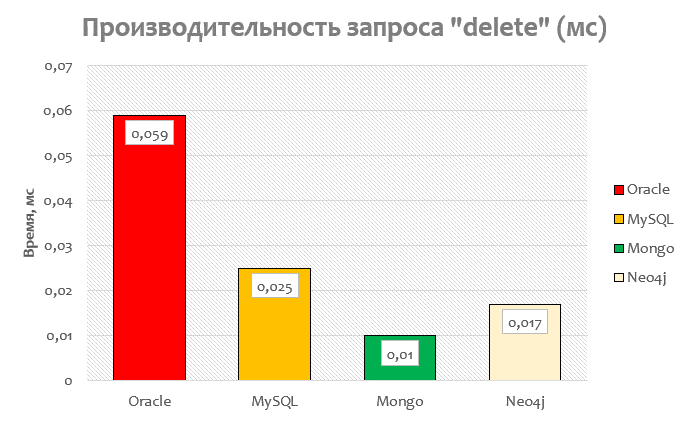


Рисунок 3 – Производительность запроса delete

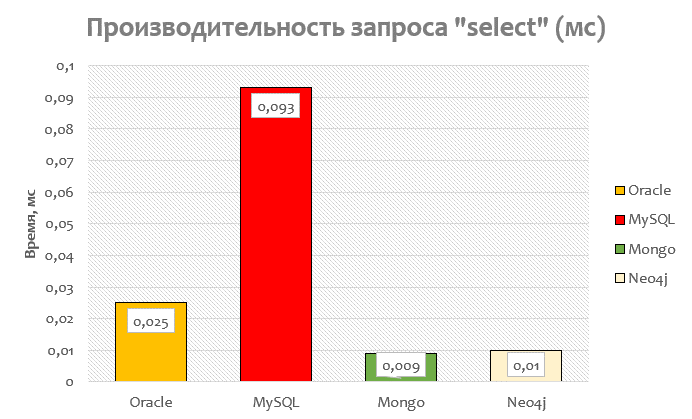


Рисунок 4 – Производительность запроса select

На Рисунках 1 и 4 видно, что время выполнения запроса select и insert в реляционных базах данных (Oracle, MySql) больше нежели в нереляционных базах данных (MongoDB, Neo4j). Такой результат был ожидаем. Он обусловлен способом хранения данных. В NoSQL, таких как MongoDB или Neo4j, данные хранятся в виде плоских коллекций, где эти данные дублируются снова и снова, а отдельный фрагмент данных практически никогда не разделяется на части, вместо этого он хранится в виде сущности. Это означает, что операции чтения или записи в отдельную сущность стали более доступными и быстрыми.

С другой стороны, в реляционных базах данных данные необходимо разбить на несколько небольших логических таблиц, чтобы избежать дублирования и избыточности. Для этого необходима нормализация. Нормализация обеспечивает правильное и эффективное управление данными. Разделение данных на несколько связанных таблиц, связанное с нормализацией, снижает производительность обработки данных в базах данных отношений с использованием SQL, и поэтому время, необходимое для получения значения из баз данных отношений, намного выше, чем из баз данных без отношений.

Аналогичную ситуацию мы видим и на Рисунках 2 и 3, где сравнивается скорость работы операций Delete и Update. Но здесь стоит отметить, что в нереляционных БД процесс удаления занимает больше времени, чем остальные операции.

## Эксперимент для 100 000 записей

Во втором эксперименте было проведено аналогичное сравнение времени работы операций insert, update, delete и select. Изменением служит количество записей в БД: эксперимент проводился при размере диапазона 100000 записей в БД. Результаты проведения сравнения представлены в Таблице 4.

Таблица 4

Производительность запросов к базам данных с 100 000 записями в миллисекундах

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Тип/операция** | **Oracle** | **MySql** | **Mongo** | **Neo4j** |
| **insert** | 0.091 | 0.038 | 0.005 | 0.008 |
| **update** | 0.092 | 0.068 | 0.009 | 0.012 |
| **delete** | 0.119 | 0.047 | 0.015 | 0.018 |
| **select** | 0.062 | 0.067 | 0.009 | 0.011 |

Результаты второго эксперимента получились аналогичными. Нереляционные БД (MongoDB, Neo4j) справляются с запросами быстрее. Помимо этого, здесь также стоит отметить, что с увеличением количества данных, скорость работы реляционных БД (Oracle, MySql) заметно снижается. Нереляционные БД продолжают работать примерно с той же скоростью.

Для наглядности на Рисунке 5 приведена визуализация полученных результатов на графике.

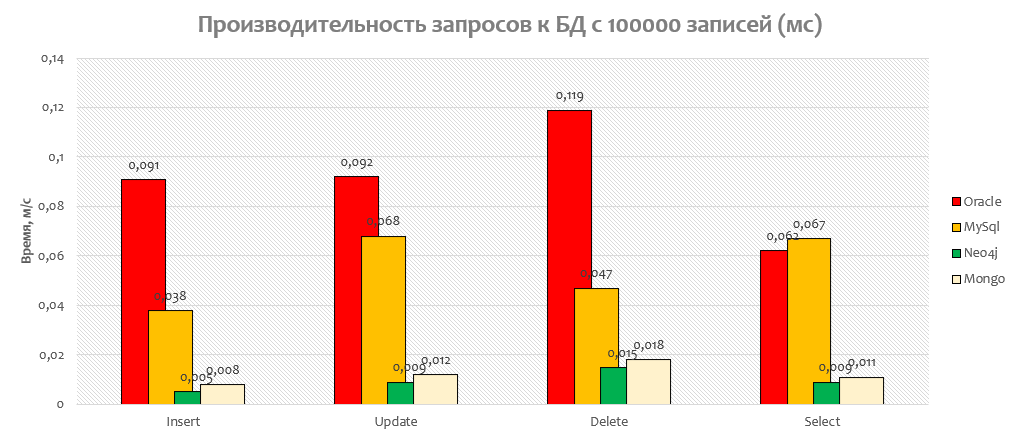


Рисунок 5 – Производительность запросов к БД с 100 000 записей

# ЗАКЛЮЧЕНИЕ

В данной лабораторной работы было проведено сравнение представителей нереляционных (MongoDB, Neo4j) и реляционных баз данных (Oracle, MySql). Для экспериментов использовался одинаковый датасет, приведенный к необходимому виду для импорта в БД. Было проведено два эксперимента, в которых сравнивалась скорость выполнения следующих команд: insert, update, delete и select.

В результате проведения экспериментов можно сделать следующие выводы:

* когда сравниваются две группы, разница между реляционными и нереляционными базами данных во времени запроса не является огромной при 10 000 записей в БД;
* при 100 000 записей в базах данных разница во времени работы между двумя группами баз данных становится более заметна;
* при проектировании БД необходимо оценивать не только скорость работы базы данных, но и другие характеристики, такие как наличие нормализации, исходные данные и др. Только в комплексе можно определить БД, которая лучше всего подойдет для поставленной задачи.

# СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Книга «Data mining. Извлечение информации из Facebook, Twitter, LinkedIn, Instagram, GitHub» – Хабр – [Электронный ресурс]. Режим доступа: https://habr.com/ru/company/piter/blog/466729/ (Дата обращения: 16.11.2021)
2. Что такое База Данных – Oracle СНГ – [Электронный ресурс]. Режим доступа: https://www.oracle.com/ru/database/what-is-database/ (Дата обращения: 16.11.2021)
3. Lourenço, J., Cabral, B., Carreiro, P., Vieira, M. and Bernardino, J. (2015). Choosing the right NoSql database for the job: a quality attribute evaluation. Journal of Big Data, 2(1).
4. Mehmood, N., Culmone, R. and Mostarda, L. (2017). Modeling temporal aspects of sensor data for MongoDB NoSql database. Journal of Big Data, 4(1).
5. Sánchez-de-Madariaga, R., Muñoz, A., Lozano-Rubí, R., Serrano-Balazote, P., Castro, A., Moreno, O. and Pascual, M. (2017). Examining database persistence of ISO/EN 13606 standardized electronic health record extracts: relational vs. NoSql approaches. BMC Medical Informatics and Decision Making, 17(1).