

УДК 004.65

[https://doi.org/ 10.35546/kntu2078-4481.2019.4.13](https://doi.org/10.35546/kntu2078-4481.2019.4.13)

О.А. ЛЯШЕНКО

ГВУЗ «Український державний хіміко-технологічний університет»

ORCID: 0000-0002-9983-5504

О.О. КОНАШКОВ

ГВУЗ «Український державний хіміко-технологічний університет»

ORCID: 0000-0001-6376-9157

Н.А. СОЛОДКА

ГВУЗ «Український державний хіміко-технологічний університет»

ORCID: 0000-0002-7545-4969

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ВЫПОЛНЕНИЯ ЗАПРОСОВ К СЕРВЕРАМ БАЗ ДАННЫХ MYSQL И MONGODB

В данной работе проводится сравнение скорости выполнения запросов к реляционной и нереляционной системам управления базами данных (СУБД). Объектом исследования является определение скорости выполнения запросов в СУБД MySQL и MongoDB. Предмет исследования – реляционная СУБД MySQL 5.7.19 и нереляционная СУБД MongoDB 4.0.6. Цель исследования – сравнение реляционной и нереляционной СУБД в контексте времени выполнения аналогичных запросов к базам данных с одинаковыми данными и структурой на одном оборудовании. Для проведения анализа была использована учебная реляционная база данных «Employees», которая имеет открытый исходный текст и является доступной для скачивания из репозитория GitHub. Для тестирования нереляционной СУБД база данных «Employees» была импортирована в документно-ориентированную СУБД MongoDB. Реляционные таблицы были преобразованы в коллекции, строки – в документы. При этом структура базы данных не была изменена. Как в случае применения реляционного подхода, так и в случае организации нереляционной структуры данных в соответствующих базах данных индексы не использовались. Запросы к базам данных выполнялись в соответствующих консольных инструментах: для MySQL – Command Line Client, для MongoDB – Mongo Shell. В качестве инструмента обработки данных в выбранных СУБД использовались запросы на выборку данных. Запросы для тестирования имели аналитический характер, часть из них содержали агрегатные функции. Измерение времени выполнения запросов производилось при старте сервера СУБД (первый запрос к серверу) и на сервере, к которому уже производились запросы (обычные условия). Время выполнения запросов получено с помощью стандартных, встроенных функций СУБД. В MySQL время выводилось автоматически и с помощью профилирования, в MongoDB – с помощью функции Explain. Выявлено, что время выполнения запросов в обычных условиях меньше на MySQL, чем на MongoDB. Некоторые запросы выполнялись быстрее в MongoDB при условии, что они были первыми к запущенному серверу. Замечено, что время выполнения нескольких параметрических запросов в MySQL разное в зависимости от параметра запроса. С практической точки зрения, для выполнения запросов на выборку данных лучше применять MySQL, чем MongoDB. Данные выводы распространяются на базу данных «Employees» и на подобные ей по структуре и объёму данных.

Ключевые слова: база данных, система управления базой данных, реляционный подход, нереляционный подход, NoSQL, MySQL, MongoDB, время выполнения запросов, сравнение, запрос.

О.А. ЛЯШЕНКО

ДВНЗ «Український державний хіміко-технологічний університет»

ORCID: 0000-0002-9983-5504

О.О. КОНАШКОВ

ДВНЗ «Український державний хіміко-технологічний університет»

ORCID: 0000-0001-6376-9157

Н.О. СОЛОДКА

ДВНЗ «Український державний хіміко-технологічний університет»

ORCID: 0000-0002-7545-4969

ПОРІВНЯЛЬНИЙ АНАЛІЗ ВИКОНАННЯ ЗАПИТІВ ДО СЕРВЕРІВ БАЗ ДАНИХ MYSQL І MONGODB

У даній роботі проводиться порівняння швидкості виконання запитів до реляційної і нереляційної систем управління базами даних (СУБД). Об'єктом дослідження є визначення швидкості виконання запитів в СУБД MySQL і MongoDB. Предмет дослідження – реляційна СУБД MySQL 5.7.19 і нереляційна СУБД MongoDB 4.0.6. Мета дослідження – порівняння реляційної і нереляційної СУБД в

контексті часу виконання аналогічних запитів до баз даних з однаковими даними та структурою на одному обладнанні. Для проведення аналізу була використана навчальна реляційна база даних «Employees», яка має відкритий вихідний текст і є доступною для скачування з репозиторію GitHub. Для тестування нереляційної СУБД база даних «Employees» була імпортована в документно-орієнтовану СУБД MongoDB. Реляційні таблиці були перетворені в колекції, рядки – в документи. При цьому структура бази даних не була змінена. Як у разі застосування реляційного підходу, так і в разі організації нереляційної структури даних у відповідних базах даних індекси не використовувалися. Запити до баз даних виконувалися в відповідних консольних інструментах: для MySQL – Command Line Client, для MongoDB – Mongo Shell. Як інструмент обробки даних в обраних СУБД використовувалися запити на вибірку даних. Запити для тестування мали аналітичний характер, частина з них містили агрегатні функції. Вимірювання часу виконання запитів відбувалося при старті сервера СУБД (перший запит до сервера) і на сервері, до якого вже проводилися запити (звичайні умови). Час виконання запитів отримано за допомогою стандартних, вбудованих функцій СУБД. В MySQL час виводився автоматично і за допомогою профілювання, в MongoDB – за допомогою функції Explain. Виявлено, що час виконання запитів в звичайних умовах менший на MySQL, ніж на MongoDB. Деякі запити швидше виконувалися в MongoDB за умови, що вони були першими до запущеного сервера. Помічено, що час виконання декількох параметричних запитів в MySQL різний в залежності від параметра запиту. З практичної точки зору, для виконання запитів на вибірку даних краще застосовувати MySQL, ніж MongoDB. Дані висновки поширюються на базу даних "Employees" і на подібні їй за структурою і обсягом даних.

Ключові слова: база даних, система управління базою даних, реляційний підхід, нереляційний підхід, NoSQL, MySQL, MongoDB, час виконання запитів, порівняння, запит.

O.A. LIASHENKO

Ukrainian State University of Chemical Technology
ORCID: 0000-0002-9983-5504

O.O. KONASHKOV

Ukrainian State University of Chemical Technology
ORCID: 0000-0001-6376-9157

N.O. SOLODKA

Ukrainian State University of Chemical Technology
ORCID: 0000-0002-7545-4969

COMPARATIVE ANALYSIS OF QUERIES EXECUTION FOR MYSQL AND MONODB DATABASE SERVERS

In this work queries execution speed for relational and non-relational database management systems (DBMS) was compared. Speed of queries execution for MySQL DBMS and MongoDB DBMS is an object of a research. Relational DBMS MySQL 5.7.19 and non-relational DBMS MongoDB 4.0.6 is a subject of the research. Comparison of a relational DBMS and a non-relational DBMS in a context of execution time of analogical queries to databases with the same data and structure using the same computer is a purpose of the research. For analysis a sample relational database «Employees» which has open source code and available for downloading from GitHub repository was used. For a non-relational DBMS testing the database «Employees» was imported to a document-oriented DBMS MongoDB. Relational tables were converted to collections, table rows were converted to documents. In addition the database structure was not changed. In both cases, using relational and non-relational approaches, corresponding databases do not have indexes. Databases queries were executed using corresponding console tools: for MySQL it is Command Line Client, for MongoDB – Mongo Shell. In mentioned DBMS's used data retrieval queries as a tool for data processing. Test queries are analytical, part of them contain aggregate functions. Capturing of time which needed for queries execution were made when DBMS's were started (the first query to the server) and when DBMS's were previously queried for several times (normal conditions). Execution time of the queries was captured using standard, built functions of DBMS's. MySQL showed time automatically and using profiling, MongoDB showed time using Explain function. Was found that execution time of the queries in normal conditions is less in MySQL than in MongoDB. In MongoDB some queries were executed faster in conditions when they were the first queries to the server. Was found that in MySQL execution time of some parametric queries depend from parameters of those queries. From

a practical perspective for data retrieval queries it is better to use MySQL than MongoDB. These findings spread to the database «Employees» and databases which have similar structure and data volume.

Keywords: database, database management system, relational approach, non-relational approach, NoSQL, MySQL, MongoDB, query execution time, comparison, query.

Постановка проблеми

При необходимости выбора СУБД главный вопрос обычно состоит в выборе реляционной (SQL) или нереляционной (NoSQL) структуры. Одним из критериев выбора может служить скорость выполнения запросов. Для данной работы в качестве SQL СУБД рассматривается MySQL 5.7.19, в качестве NoSQL СУБД – MongoDB 4.0.6. MongoDB – документно-ориентированная СУБД с открытым исходным кодом. Она является самой популярной NoSQL базой данных в соответствии с рейтингом DB-Engines Ranking. Один документ эквивалентен строке таблицы из реляционной базы данных, документы хранятся в коллекциях. Коллекции в отличие от реляционных таблиц могут хранить документы разной структуры. Начиная с версии 4.0 MongoDB поддерживает ACID транзакции. Так же MongoDB имеет инструменты для проведения операций аналогичных JOIN в реляционных базах данных.

Проанализировав исследования на данную тему, выявлено, что вопрос сравнения популярных СУБД MySQL и MongoDB по скорости выполнения запросов на выборку данных изучен не достаточно детально. Поэтому исследования в данной работе проведены с целью решения проблемы сравнения реляционной и нереляционной СУБД путём сопоставления скорости выполнения запросов на выборку данных.

Анализ последних исследований и публикаций

В статье [1] приведены результаты исследования производительности СУБД MySQL и MongoDB. По результатам экспериментов СУБД MySQL на небольших объемах данных более чем в два раза быстрее, чем MongoDB. С ростом количества записей скорость работы MySQL значительно падает, на больших объемах данных MongoDB работает быстрее. Производительность реляционной базы, запущенной в четыре потока на больших наборах данных сопоставима с производительностью MongoDB, которая обрабатывает данные последовательно. В статье сравнение СУБД основано на примере решения задачи очистки и трансформации данных для подготовки к Data Mining.

В работе [2] показано, что MongoDB также превосходит MySQL по скорости выполнения операций вставки, выборки данных, удаления элементов по индексу. В статье сравнение СУБД проведено с точки зрения их применения в высоконагруженных системах.

Исследование [3] показало, что на маленьких объёмах данных разница между MySQL и MongoDB не большая, но на больших объёмах данных MongoDB выполняет операции быстрее. В данном исследовании точно не ясно – применяются транзакционные запросы или нет. Основываясь на том, что производительность СУБД сравнивается для предметной области, связанной с системой управления супермаркетом, в данном исследовании применялись запросы транзакционного характера.

В работе [4] проведена разработка методики эталонного тестирования (benchmark) и программного обеспечения для баз данных MySQL и MongoDB. По результатам данного исследования MongoDB быстрее MySQL в запросах на запись и считывание данных. Исследование [4] связано с транзакционной обработкой данных OLTP.

Исследование [5] демонстрирует преимущество MongoDB в записи и считывании данных. Тестовые запросы выполняются к базам данных одной предметной области, но различной структуры.

По результатам, приведённым в [6], MongoDB превосходит MySQL по скорости выполнения операций вставки, выборки данных, удаления элементов. Статья содержит запрос транзакционного уровня сложности и аналитический запрос, при этом они выполняются к базам данных одной предметной области и одинаковой структуры. Время измерялось на уровне выполнения PHP кода, но не уровне баз данных, что сочетает время выполнения самого запроса и программного кода на PHP.

Некоторые аспекты сравнения реляционного и нереляционного подхода на примере СУБД MySQL и MongoDB приведены в работе [7]. Реляционный подход так же рассматривался в исследовании [8]. Нереляционная база данных как альтернатива реляционной рассматривается в исследовании [9] на примере графовой базы данных.

Проведенный анализ литературных источников показал, что вопросы выбора типа структур данных и соответствующих им СУБД является весьма актуальным. Основываясь на анализе приведенных исследований, представляется актуальным провести более детальное изучение реляционной и нереляционной СУБД, связанное с измерением времени выполнения запросов на выборку данных. При этом целесообразно использовать общедоступную базу данных и одинаковые схемы данных, специализированные инструменты СУБД для измерения времени выполнения запросов.

Формулирование цели исследования

Целью данного исследования является измерение и сравнение времени выполнения запросов к реляционной и нереляционной базам данных, при условии, что обе базы данных содержат одни и те же

пользовательские данные и имеют одну и ту же структуру. Для сравнения выбраны СУБД MySQL 5.7 и MongoDB 4.0.6 как широко используемые соответственно реляционная и документно-ориентированная базы данных. Обе СУБД являются продуктами с открытым исходным кодом и работают на разных платформах.

Изложение основного материала исследования

1. Базы данных используемые для тестирования СУБД. Реализация реляционной базы данных

Для сервера реляционных баз данных MySQL разработано несколько баз данных, находящихся в свободном доступе, которые можно использовать в учебных и исследовательских целях. Одна из них – «Employees», которая использована в данной работе. Указанная БД свободно доступна для скачивания из репозитория GitHub, ссылка на который размещена на официальном сайте MySQL.

База данных «Employees» описывает предметную область, связанную с хранением записей о сотрудниках, работающих на определённом предприятии. Она содержит данные о личности сотрудника, его зарплате, отделах предприятия, принадлежности сотрудника к определённому отделу, его должности и данные о том, является ли сотрудник менеджером определённого отдела. На рисунке 1 представлена схема данных для реляционной базы «Employees».

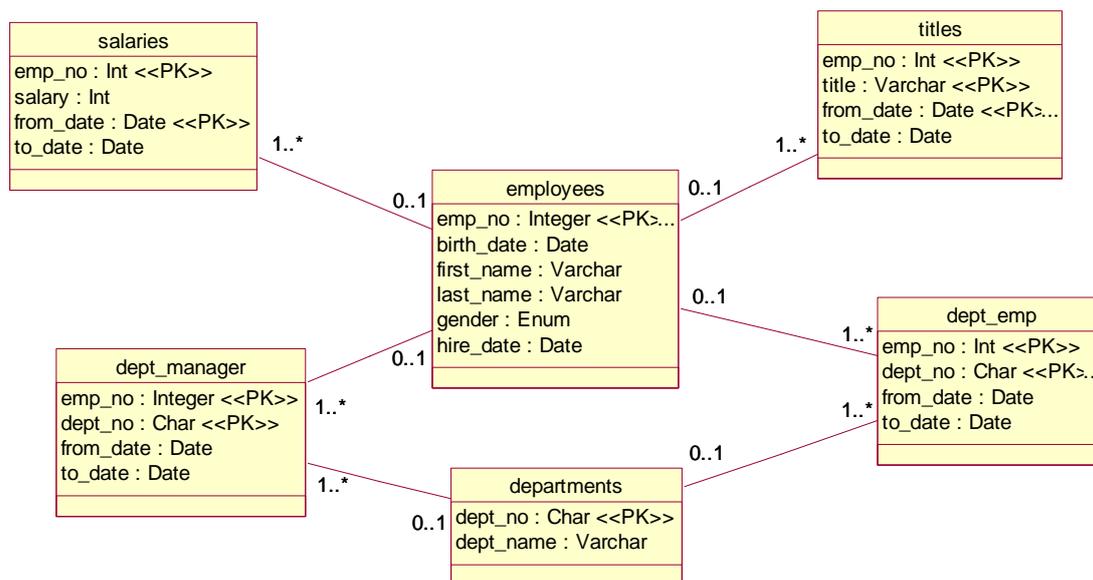


Рис. 1. Схема данных реляционной базы данных «Employees»

Данные о личности сотрудника хранятся в таблице «EMPLOYEES», она включает в себя поля с номером сотрудника (`emp_no`), датой рождения (`birth_date`), именем (`first_name`), фамилией (`last_name`), полом (`gender`), датой найма (`hire_date`). Данная таблица содержит 300 024 записей.

Данные о зарплате сотрудника хранятся в таблице «SALARIES», она включает в себя поля с его номером (`emp_no`), размером зарплаты (`salary`), дату начала получения зарплаты определённого размера (`from_date`) и конечную дату получения данной зарплаты (`to_date`). Таблица содержит 2 844 047 записей.

Данные об отделах предприятия хранятся в таблице «DEPARTMENTS», она включает в себя поля с номером отдела (`dept_no`), названием отдела (`dept_name`). Таблица содержит 9 записей.

Данные о принадлежности сотрудника к определённому отделу хранятся в таблице «DEPT_EMP», она включает в себя поля с номером сотрудника (`emp_no`), номером отдела (`dept_no`), даты начала работы в данном отделе (`from_date`) и окончания работы (`to_date`). Таблица содержит 331 603 записей.

Данные о должности сотрудника хранятся в таблице «TITLES», она включает в себя поля с номером сотрудника (`emp_no`), его должностью (`title`), датами начала работы на данной должности (`from_date`) и окончания работы (`to_date`). Таблица содержит 443 308 записей.

Данные о менеджерах отделов хранятся в таблице «DEPT_MANAGER», она включает в себя поля с номером сотрудника (`emp_no`), номером отдела (`dept_no`), датой начала работы менеджером (`from_date`) и датой окончания (`to_date`). Таблица содержит 24 записи.

2. Базы данных используемые для тестирования СУБД. Реализация документно-ориентированной базы данных

Данные из базы «Employees» можно перенести в СУБД MongoDB, используя схему данных аналогичную реляционной. При необходимости так же возможно свести все таблицы в одну коллекцию

или частично преобразовать схему. Для исследования времени выполнения запросов был выбран вариант переноса данных без изменений схемы данных, так как целью данной работы не является исследование быстродействия СУБД в зависимости от различных способов представления структуры данных. Физически реляционные таблицы были преобразованы в коллекции, строки – в документы с использованием экспорта и импорта файлов формата CSV. При импортировании базы данных «Employees» идентификаторы документов типа ObjectID во всех коллекциях были созданы по умолчанию.

Таким образом, в MongoDB была получена документная база данных аналогичная реляционной.

3. Тесты производительности

Все тесты скорости выполнения запросов были выполнены на персональном компьютере со спецификациями, включающими: процессор Mobile DualCore Intel Core i5-2410M, 2700 MHz; 12 ГБ оперативной памяти DDR3; WDC WD5000BPVT-24HXZT1 ATA Device (500 ГБ, 5400 об/мин, SATA-II); Windows 7 Home Premium.

В тестах использовались следующие версии СУБД: MySQL 5.7.19 для Windows x64; MongoDB 4.0.6 для Windows x64.

В обеих базах данных индексирование полей для запросов не использовалось.

К указанным базам данных был построен и выполнен ряд запросов, приведен текст, описание запросов и время, за которое они были выполнены. Часть запросов была создана с использованием агрегатных функций SQL.

Запросы к СУБД MySQL осуществлялись посредством использования консольного клиента Command Line Client для работы с базой данных MySQL. Для доступа к данным MongoDB использовался интерактивный консольный JavaScript интерфейс Mongo Shell.

Для измерения времени запросы выполнялись как при старте СУБД (первые запросы после запуска сервера), так и путем запуска в обычных условиях (на сервере, на котором уже выполнялись запросы). Для MySQL проводились также измерения времени при выполнении запросов из файла, в MongoDB такой эксперимент не проводился, поэтому данные результаты не приведены в статье.

Время выполнения запроса в консоли MySQL отображается автоматически или при использовании профилирования (set profiling=1). В Mongo Shell для этого используется функция explain("executionStats").

Ниже приведены тексты запросов и результаты времени их выполнения при различных условиях на исследуемых СУБД. Запросы 1–17 являются однотабличными.

Запрос №1. Подсчитать количество записей в таблице «SALARIES».

Запрос для MySQL Command Line Client:

```
select count(*) from salaries.
```

Запрос для Mongo Shell:

```
db.salaries.count().
```

Время выполнения данного запроса приведено в табл. 1.

Таблица 1

Измерение времени выполнения запроса №1

№	MySQL	MongoDB	Условия выполнения запроса
1	1.94 с	0.000 с	Первый запрос после запуска сервера
2	1.61 с	0.000 с	Запуск в обычных условиях
3	1.49 с	0.000 с	Запуск в обычных условиях
4	1.81 с	0.000 с	Запуск в обычных условиях

Запрос №2. Подсчитать количество зарплат, размер которых больше, чем 40 000.

Запрос для MySQL Command Line Client:

```
select count(*) from salaries where salary>40000.
```

Запрос для Mongo Shell:

```
db.salaries.count({"salary":{"$gt:40000}}).
```

Время выполнения данного запроса приведено в табл. 2.

Таблица 2

Измерение времени выполнения запроса №2

№	MySQL	MongoDB	Условия выполнения запроса
1	2.01 с	2.486 с	Первый запрос после запуска сервера
2	0.78 с	1.771 с	Запуск в обычных условиях
3	0.75 с	1.764 с	Запуск в обычных условиях
4	0.76 с	1.766 с	Запуск в обычных условиях

Запрос №3. Подсчитать количество зарплат, размер которых меньше либо равен 50 000.

Запрос для MySQL Command Line Client:

```
select count(*) from salaries where salary<=50000.
```

Запрос для Mongo Shell:

```
db.salaries.count({"salary":{$lte:50000}}).
```

Время выполнения данного запроса приведено в табл. 3.

Таблица 3

Измерение времени выполнения запроса №3

№	MySQL	MongoDB	Условия выполнения запроса
1	1.94 с	2.476 с	Первый запрос после запуска сервера
2	0.73 с	1.824 с	Запуск в обычных условиях
3	0.72 с	1.809 с	Запуск в обычных условиях
4	0.72 с	1.808 с	Запуск в обычных условиях

Запрос №4. Подсчитать количество зарплат, размер которых больше 40 000 и меньше либо равен 50 000.

Запрос для MySQL Command Line Client:

```
select count(*) from salaries where salary>40000 and salary<=50000.
```

Запрос для Mongo Shell:

```
db.salaries.count({"salary":{$gt:40000, $lte:50000}}).
```

Время выполнения данного запроса приведено в табл. 4.

Таблица 4

Измерение времени выполнения запроса №4

№	MySQL	MongoDB	Условия выполнения запроса
1	1.57 с	2.500 с	Первый запрос после запуска сервера
2	0.82 с	1.992 с	Запуск в обычных условиях
3	0.80 с	1.983 с	Запуск в обычных условиях
4	0.81 с	1.990 с	Запуск в обычных условиях

Запрос №5. Подсчитать количество сотрудников, работающих на должности «Assistant Engineer». В базе данных содержится 15 128 таких записей.

Запрос для MySQL Command Line Client:

```
select count(*) from titles where title like "Assistant Engineer".
```

Запрос для Mongo Shell:

```
db.titles.count({"title": "Assistant Engineer"}).
```

Время выполнения данного запроса приведено в табл. 5.

Таблица 5

Измерение времени выполнения запроса №5

№	MySQL	MongoDB	Условия выполнения запроса
1	0.41 с	0.380 с	Первый запрос после запуска сервера
2	0.15 с	0.279 с	Запуск в обычных условиях
3	0.16 с	0.278 с	Запуск в обычных условиях
4	0.15 с	0.273 с	Запуск в обычных условиях

Запрос №6. Подсчитать количество сотрудников, работающих на должности «Engineer». В базе данных содержится 115 003 таких записей.

Запрос для MySQL Command Line Client:

```
select count(*) from titles where title like "Engineer".
```

Запрос для Mongo Shell:

```
db.titles.count({"title": "Engineer"}).
```

Время выполнения данного запроса приведено в табл. 6.

Таблица 6

Измерение времени выполнения запроса №6

№	MySQL	MongoDB	Условия выполнения запроса
1	0.40 с	0.354 с	Первый запрос после запуска сервера
2	0.16 с	0.272 с	Запуск в обычных условиях
3	0.16 с	0.271 с	Запуск в обычных условиях
4	0.16 с	0.271 с	Запуск в обычных условиях

Запрос №7. Отбираются сотрудники, у которых сумма всех зарплат больше либо равна 2 400 000.

Запрос для MySQL Command Line Client:

```
select emp_no, sum(salary) as sum_sal from salaries group by emp_no having sum_sal >= 2400000.
```

Запрос для Mongo Shell:

```
db.salaries.aggregate([{$group: {_id:"$emp_no", totalSalary: {$sum: "$salary"}}}, {$match: {totalSalary: {$gte: 2400000 }}}]).
```

Время выполнения данного запроса приведено в табл. 7.

Таблица 7

Измерение времени выполнения запроса №7

№	MySQL	MongoDB	Условия выполнения запроса
1	2.27 с	4.740 с	Первый запрос после запуска сервера
2	1.07 с	4.050 с	Запуск в обычных условиях
3	1.04 с	4.046 с	Запуск в обычных условиях
4	1.04 с	4.050 с	Запуск в обычных условиях

Запрос №8. Отбираются сотрудники, у которых сумма всех зарплат меньше 41 000.

Запрос для MySQL Command Line Client:

```
select emp_no, sum(salary) as sum_sal from salaries group by emp_no having sum_sal < 41000.
```

Запрос для Mongo Shell:

```
db.salaries.aggregate([{$group: {_id:"$emp_no", totalSalary: {$sum: "$ salary"}}}, {$match: {totalSalary: {$lt: 41000 }}}]).
```

Время выполнения данного запроса приведено в табл. 8.

Таблица 8

Измерение времени выполнения запроса №8

№	MySQL	MongoDB	Условия выполнения запроса
1	1.59 с	4.696 с	Первый запрос после запуска сервера
2	1.09 с	4.065 с	Запуск в обычных условиях
3	1.06 с	4.098 с	Запуск в обычных условиях
4	1.06 с	4.099 с	Запуск в обычных условиях

Запрос №9. Определить число работников (бывших и работающих в данный момент) в отделе под номером «d001».

Запрос для MySQL Command Line Client:

```
select count(*) from dept_emp where dept_no="d001".
```

Запрос для Mongo Shell:

```
db.dept_emp.find({"dept_no":{"$eq":"d001"}}).count().
```

Запросы с девятого по семнадцатый отличаются только номерами отделов (d001-d009). Время выполнения запросов разнится в MySQL. Запросы, которые возвращают большее количество подсчитанных записей, обрабатывались дольше. Похожий результат наблюдался с запросами №5 и №6. Среднее время выполнения запросов с №9 по №17 в обычных условиях и результаты их выполнения приведены в табл. 9.

Таблица 9

Сводная таблица среднего времени выполнения запросов №9–№17 в обычных условиях и результатов их выполнения

№ запроса	№ отдела	MySQL	MongoDB	Количество работников (бывших и работающих в данный момент) в отделе
9	d001	0.01 с	0.202 с	20 211
10	d002	0.01 с	0.202 с	17 346
11	d003	0.01 с	0.204 с	17 786
12	d004	0.023 с	0.202 с	73 485
13	d005	0.03 с	0.203 с	85 707
14	d006	0.01 с	0.202 с	20 117
15	d007	0.02 с	0.202 с	52 245
16	d008	0.01 с	0.201 с	21 126
17	d009	0.01 с	0.204 с	23 580

Запросы №18–№20 являются многотабличными.

Запрос №18. Вывести номера сотрудников, которые были менеджерами отделов, номера и названия этих отделов, начальную и конечную даты, когда работник был менеджером отдела. Время выполнения данного запроса приведено в табл. 10.

Запрос для MySQL Command Line Client:

```
select dm.emp_no, d.dept_no, d.dept_name, dm.from_date, dm.to_date from departments d inner join dept_manager dm on d.dept_no=dm.dept_no.
```

Запрос для Mongo Shell:

```
db.getCollection("departments").aggregate([{"$project": {"_id": NumberInt(0), "d": "$ROOT"}}, {"$lookup": {"localField": "d.dept_no", "from": "dept_manager", "foreignField": "dept_no", "as": "dm"}}, {"$unwind": {"path": "$dm", "preserveNullAndEmptyArrays": false}}, {"$project": {"dm.emp_no": "$dm.emp_no", "d.dept_no": "$d.dept_no", "d.dept_name": "$d.dept_name", "dm.from_date": "$dm.from_date", "dm.to_date": "$dm.to_date"}}]);
```

Таблица 10

Измерение времени выполнения запроса №18

№	MySQL	MongoDB	Условия выполнения запроса
1	0.05 с	0.001 с	Первый запрос после запуска сервера
2	0.0004 с	0.001 с	Запуск в обычных условиях
3	0.0004 с	0.001 с	Запуск в обычных условиях
4	0.0004 с	0.001 с	Запуск в обычных условиях

Запрос №19. Вывести фамилию и имя менеджеров отделов, при этом названия отделов начинается на букву “s”, а так же вывести номера этих отделов, название отделов, начальную и конечную даты, когда работник был менеджером отдела. Время выполнения данного запроса приведено в табл. 11.

Запрос для MySQL Command Line Client:

```
select d.dept_no, d.dept_name, e.first_name, e.last_name, dm.from_date, dm.to_date from employees e join dept_manager dm on e.emp_no=dm.emp_no join departments d on dm.dept_no=d.dept_no where d.dept_name like 'S%';
```

Запрос для Mongo Shell:

```
db.getCollection("employees").aggregate([{"$project": {"_id": NumberInt(0), "e": "$ROOT"}}, {"$lookup": {"localField": "e.emp_no", "from": "dept_manager", "foreignField": "emp_no", "as": "dm"}}, {"$unwind": {"path": "$dm", "preserveNullAndEmptyArrays": false}}, {"$lookup": {"localField": "dm.dept_no", "from": "departments", "foreignField": "dept_no", "as": "d"}}, {"$unwind": {"path": "$d", "preserveNullAndEmptyArrays": false}}, {"$match": {"d.dept_name": /^S.*$/i}}, {"$project": {"d.dept_no": "$d.dept_no", "d.dept_name": "$d.dept_name", "e.first_name": "$e.first_name", "e.last_name": "$e.last_name", "dm.from_date": "$dm.from_date", "dm.to_date": "$dm.to_date"}}]);
```

Таблица 11

Измерение времени выполнения запроса №19

№	MySQL	MongoDB	Условия выполнения запроса
1	0.07 с	23.877 с	Первый запрос после запуска сервера
2	0.0004 с	23.338 с	Запуск в обычных условиях
3	0.0004 с	23.723 с	Запуск в обычных условиях
4	0.0004 с	23.530 с	Запуск в обычных условиях

Запрос №20. Вывести названия отделов (dept_name), в которых зарегистрирован сотрудник с определенным номером (emp_no). Время выполнения данного запроса приведено в табл. 12.

Запрос для MySQL Command Line Client:

```
select emp_no, dept_name from departments join dept_emp on departments.dept_no = dept_emp.dept_no where dept_emp.emp_no = 499992.
```

Запрос для Mongo Shell:

```
db.departments.aggregate ([{"$ project": {"_id": NumberInt (0), "Departments": "$ ROOT"}}, {"$ lookup": {"localField": "departments.dept_no", "From": "dept_emp", "foreignField": "dept_no", "as": "dept_emp"}}, {"$ Unwind": {"path": "$ dept_emp", "preserveNullAndEmptyArrays": false}} {"$ Match": {"dept_emp.emp_no": NumberLong (499992)}}, {"$ project": {"Dept_emp.emp_no": "$ dept_emp.emp_no", "departments.dept_name": "$ departments.dept_name"}}]).
```

Таблиця 12

Измерение времени выполнения запроса №20

№	MySQL	MongoDB	Условия выполнения запроса
1	0.07 с	1.869 с	Первый запрос после запуска сервера
2	0.0004 с	1.793 с	Запуск в обычных условиях
3	0.0004 с	1.778 с	Запуск в обычных условиях
4	0.0004 с	1.795 с	Запуск в обычных условиях

Далее в табл. 13 приведены средние арифметические значения времени выполнения запросов в обычных условиях.

Таблиця 13

Среднее время выполнения запросов в обычных условиях

№ запроса	MySQL Среднее время, с.	MongoDB Среднее время, с.
1	1.637	0.000
2	0.763	1.707
3	0.723	1.817
4	0.81	1.986
5	0.153	0.277
6	0.16	0.273
7	1.05	4.048
8	1.07	4.082
9	0.01	0.202
10	0.01	0.202
11	0.01	0.204
12	0.023	0.202
13	0.03	0.203
14	0.01	0.202
15	0.02	0.202
16	0.01	0.201
17	0.01	0.204
18	0.0004	0.001
19	0.0004	23.617
20	0.0004	1.789

На основе измерения времени тестовых запросов на выборку данных можно сделать вывод, что сервер MySQL 5.7.19 выполнял запросы в обычных условиях быстрее, чем MongoDB 4.0.6. Запросы №5, №6, №13 и №15 выполнялись быстрее в MongoDB при условии, что они были первыми запросами после запуска сервера. Замечено, что время выполнения параметрических запросов №5 и №6, а так же запросов с №9 по №17 включительно в MySQL разное в зависимости от параметра запроса.

Выводы

Проведен сравнительный анализ СУБД MySQL и MongoDB, с помощью измерения времени выполнения запросов на выборку данных. Для этого изначально была использована образцовая реляционная база данных «Employees», находящаяся в открытом доступе. На ее основе построена документно-ориентированная база данных с той же структурой, что и реляционная.

К реляционной базе на языке SQL разработаны запросы на выборку данных, учитывающие семантику пользовательских данных. Такие же запросы разработаны для документной базы данных.

На одном и том же аппаратном обеспечении с использованием стандартных консольных интерфейсов были выполнены запросы к серверам MySQL и MongoDB.

С помощью стандартных функций СУБД были получены результаты измерений времени выполнения запросов, как при старте сервера СУБД, так и на сервере, на котором уже выполнялись запросы.

Результаты тестовых запросов показали, что для рассмотренной базы данных «Employees» и подобных ей по структуре и объёму данных сервер базы данных MySQL выполняет запросы на выборку данных быстрее, чем сервер MongoDB.

Список использованной литературы

1. Шичкина Ю.А. Сравнение производительности реляционных и нереляционных баз данных на примере MySQL и MongoDB / Куприянов М.С., Коблов А.А. // Информационные системы и технологии в моделировании и управлении: материалы всероссийской научно-практической конференции, Ялта, 2017. – С. 213-219.
2. Сокольников А.М. Сравнительный анализ подходов к разработке архитектуры и систем управления базами данных для высоконагруженных WEB-сервисов // Кибернетика и программирование. – 2014. – № 4. – С. 1-13. <http://doi.org/10.7256/2306-4196.2014.4.12800>
3. Dipina Damodaran B. MongoDB vs MySQL: a comparative study of performance in super market management system / Dipina Damodaran B, Shirin Salim, Surekha Mariam Vargese // International Journal of Computational Science and Information Technology (IJCSITY), 2016, vol.4, no.2. – P. 31-38. <https://doi.org/10.5121/ijcsity.2016.4204>
4. Ion Lungu The Development of a Benchmark Tool for NoSQL Databases / Ion Lungu, Bogdan George Tudorica // Database Systems Journal, 2013, vol IV, no. 2. – P. 13-20.
5. Zhu Wei-ping Using MongoDB to Implement Textbook Management System instead of MySQL / Zhu Wei-ping, Li Ming-xin, Chen Huan // 2011 IEEE 3rd International Conference on Communication Software and Networks, United States, 2011. – P. 303-305.: <https://doi.org/10.1109/ICCSN.2011.6013720>
6. Cornelia Gyorodi A Comparative Study: MongoDB vs. MySQL / Robert Gyorodi, George Pecherle, Andrada Olah // 2015 13th International Conference on Engineering of Modern Electric Systems (EMES) Режим доступа: <https://doi.org/10.1109/EMES.2015.7158433>
7. Ляшенко О. А., Коначков О. О. Сравнение реляционной и нереляционной СУБД на примере MYSQL и MONGODB // Матеріали V Міжнародної науково-технічної конференції «Комп'ютерне моделювання та оптимізація складних систем». – Дніпро, 2019. – С. 125-126. <http://doi.org/10.32434/CMOCS-2019>
8. Солодка Н.О., Поліщук Є.О., Ляшенко О.А. Використання графової та реляційної моделей даних при розробці експертних систем // ВІСНИК ХНТУ. – 2018. – № 4(67). – С.246-251.
9. Малініна В.О., Ляшенко О.А. Особливості використання нереляційної бази даних на основі графової моделі даних // Тези доповідей IX Міжнародної науково-технічної конференції студентів, аспірантів та молодих вчених «Хімія та сучасні технології». – Дніпро, 2019. – Т. IV. – С. 28-29.

References

1. Shichkina Ju.A., Kuprijanov M.S., Koblov A.A. Sravnenie proizvoditel'nosti reljacionnyh i nereljacionnyh baz dannyh na primere MySQL i MongoDB [Comparison of the performance of relational and non-relational databases on the example of MySQL and MongoDB]. Materialy vserossijskoj nauchno-prakticheskoy konferencii "Informacionnye sistemy i tehnologii v modelirovanii i upravlenii" [National Sci.-Pract. Conf. proceedings "Information systems and technologies in modelling and management"]. Yalta, 2017, pp. 213-219.
2. Sokol'nikov A.M. Sravnitel'nyj analiz podhodov k razrabotke arhitektury i sistem upravlenija bazami dannyh dlja vysokonagruzhenykh WEB-servisov [Comparative analysis of approaches to the development of architecture and database management systems for heavily loaded WEB-services]. Kibernetika i programmirovanie [Cybernetics and programming]. 2014, no.4, pp. 1-13. <http://doi.org/10.7256/2306-4196.2014.4.12800>
3. Dipina Damodaran B., Shirin Salim, Surekha Mariam Vargese. MongoDB vs MySQL: a comparative study of performance in super market management system. International Journal of Computational Science and Information Technology (IJCSITY), 2016, vol.4, no.2, pp. 31-38. <https://doi.org/10.5121/ijcsity.2016.4204>
4. Ion Lungu, Bogdan George Tudorica, Ion Lungu. The Development of a Benchmark Tool for NoSQL Databases. Database Systems Journal, 2013, vol.IV, no.2, pp. 13-20.
5. Zhu Wei-ping, Li Ming-xin, Chen Huan. Using MongoDB to Implement Textbook Management System instead of MySQL. 2011 IEEE 3rd International Conference on Communication Software and Networks. United States, 2011, pp. 303-305. <https://doi.org/10.1109/ICCSN.2011.6013720>
6. Cornelia Gyorodi, George Pecherle, Andrada Olah. A Comparative Study: MongoDB vs. MySQL. 2015 13th International Conference on Engineering of Modern Electric Systems (EMES). Available at: <https://doi.org/10.1109/EMES.2015.7158433> (Accessed 09 December 2019).
7. Liashenko O.A., Konashkov O.O. Sravnenie reljacionnoj i nereljacionnoj SUBD na primere MYSQL i MONGODB [Comparison of relational and non-relational DBMS on the example of MYSQL and MONGODB]. Materiali V Mizhnarodnoї naukovo-tehnichnoї konferencii "Komp'juterne modeljuvannja ta optimizacija skladnih sistem. [5th Int. Sci.-Techn. Conf. proceedings "Computer

- modeling and optimization of complex systems"]. Dnipro, 2019. pp. 125-126. <https://doi.org/10.32434/CMOCS-2019>
8. Solodka N.O., Polishhuk Є.O., Liashenko O.A. Viktoristannja grafovoї ta reljacijnoї modelej danih pri rozrobci ekspertnih sistem [Using the graph database model and the relation model while developing expert syytems]. VISNIK HNT. [VISNYK of KNTU]. 2018, no. 4(67), pp.246-251.
 9. Malinina V.O., Liashenko O.A. Osoblivosti vikoristannja nereljacijnoї bazi danih na osnovi grafovoї modeli danih [Features of using non-relational database based on graph model of data]. Tezi dopovidej IX Mizhnarodnoї naukovo-tehnichnoї konferencii studentiv, aspirantiv ta molodih vchenih "Himija ta suchasni tehnologii". [Abstracts of 9th Int. Sci.-Techn. Conf of students, graduate students and young scientists "Chemistry and modern technologies"]. Dnipro, 2019, vol. IV, pp. 28-29.