ТЕХНИКАЛЫҚ ҒЫЛЫМДАР
ТЕХНИЧЕСКИЕ НАУКИ
TECHNICAL SCIENCE

CENTRAL ASIAN SCIENTIFIC JOURNAL
ISSN 3135-3061
2026. №4 (32), Том 2. С. 468–479
DOI: https://doi.org/10.5281/zenodo.24344243
Сайт журнала: https://cajournal.kz

УДК 004.65

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ПРОИЗВОДИТЕЛЬНОСТИ И МАСШТАБИРУЕМОСТИ NOSQL И РЕЛЯЦИОННЫХ СИСТЕМ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ ПРИ ОБРАБОТКЕ СТРУКТУРИРОВАННЫХ И ПОЛУСТРУКТУРИРОВАННЫХ ДАННЫХ

А.А. Сахипов
Astana IT University, Астана, Казахстан

Сахипов Айвар Айтуарович — PhD, ассистент профессор, Школа программной инженерии, Astana IT University, Астана, Казахстан.

© А.А. Сахипов, 2026

Аннотация: В статье исследуется производительность документно-ориентированной системы MongoDB и реляционной системы PostgreSQL при обработке структурированных и полуструктурированных данных. Актуальность работы связана с ростом объёмов информации и расширением требований к современным информационным системам, включая скорость выполнения запросов, гибкость хранения и возможность масштабирования. Разработана методика сравнительного экспериментального исследования, основанная на использовании одинаковых наборов данных и сопоставимых сценариев нагрузки. Рассматриваются операции создания, чтения, обновления и удаления записей, выполнение аналитических запросов, влияние индексации и увеличение объёма информационного массива. Для оценки производительности выбраны среднее и медианное время выполнения операций, перцентили P95 и P99, пропускная способность, потребление оперативной памяти и объём дискового пространства. Научная новизна заключается в разработке комплексного подхода к оценке MongoDB и PostgreSQL с учётом взаимосвязи между моделью представления данных, структурой запросов, параметрами индексации и объёмом обрабатываемой информации. Предложена методика, позволяющая определять эффективность применения рассматриваемых СУБД для различных типов информационной нагрузки. Практическая значимость работы состоит в возможности применения полученных методических положений при проектировании образовательных платформ, корпоративных информационных систем и аналитических приложений, работающих со структурированными и полуструктурированными данными.

Ключевые слова: NoSQL, MongoDB, PostgreSQL, реляционные базы данных, производительность баз данных, масштабируемость, индексация, CRUD-операции, полуструктурированные данные, обработка больших данных.

Введение

Развитие цифровых технологий, облачных вычислений, интернета вещей и интеллектуальных информационных систем сопровождается постоянным увеличением объёмов обрабатываемой информации. Современные приложения должны обеспечивать быстрый доступ к данным, поддерживать сложные структуры хранения и сохранять работоспособность при росте пользовательской нагрузки.

Реляционные системы управления базами данных на протяжении нескольких десятилетий остаются основой корпоративных информационных систем. Их архитектура предусматривает организацию информации в виде взаимосвязанных таблиц. Использование

схем данных, первичных и внешних ключей, ограничений целостности и языка SQL позволяет формализовать структуру предметной области и выполнять транзакционные операции с соблюдением установленных требований.

Расширение функциональных возможностей цифровых приложений привело к появлению задач, для которых табличная модель хранения требует дополнительной адаптации. К ним относятся обработка событийных потоков, хранение пользовательских профилей, управление данными веб-сервисов и интеграция информации из различных источников. Структура таких данных может изменяться в процессе эксплуатации приложения.

Для решения подобных задач получили распространение NoSQL-системы управления базами данных. Они используют документную, колоночную, графовую модели и модель ключ-значение. Каждая из них ориентирована на определённые способы организации информации и характер запросов.

При выборе СУБД необходимо учитывать особенности предметной области, требования к целостности информации, интенсивность операций чтения и записи, сложность запросов и возможности масштабирования. Ошибки на этапе проектирования структуры хранения могут привести к увеличению времени выполнения операций и дополнительным затратам на сопровождение программного обеспечения.

Актуальность исследования определяется необходимостью экспериментального сравнения современных реляционных и документно-ориентированных СУБД. Особый интерес представляет оценка их производительности при работе с одинаковыми информационными объектами.

Цель исследования состоит в разработке методики сравнительной оценки производительности и масштабируемости MongoDB и PostgreSQL при обработке структурированных и полуструктурированных данных.

Для достижения поставленной цели сформулированы следующие задачи:

1. Изучить архитектурные особенности реляционной и документно-ориентированной моделей хранения данных.
2. Определить критерии сравнительной оценки MongoDB и PostgreSQL.
3. Разработать структуру тестового набора данных и сценарии экспериментальной нагрузки.
4. Определить методику измерения производительности операций чтения, записи, обновления и удаления.
5. Исследовать влияние индексации и сложности запросов на скорость обработки информации.
6. Сформировать рекомендации по выбору СУБД для различных типов информационных систем.

Объект исследования представляет собой процессы хранения, поиска, изменения и аналитической обработки данных в современных системах управления базами данных.

Предмет исследования охватывает закономерности изменения производительности MongoDB и PostgreSQL в зависимости от модели представления информации, объёма данных, типа операций и конфигурации индексов.

Научная гипотеза состоит в предположении, что производительность рассматриваемых СУБД определяется соответствием модели хранения характеру выполняемой нагрузки. Документная модель может сокращать количество обращений к базе при работе с автономными полуструктурированными объектами. Реляционная модель предоставляет развитые средства обработки взаимосвязанных сущностей и выполнения сложных аналитических запросов.

1. Анализ современных подходов к организации хранения данных

1.1. Архитектурные особенности реляционных систем

Реляционная модель данных основана на математическом аппарате отношений. Информация организуется в виде таблиц, состоящих из строк и столбцов. Каждая таблица

описывает определённую сущность предметной области, а связи между сущностями устанавливаются посредством первичных и внешних ключей.

PostgreSQL представляет собой объектно-реляционную систему управления базами данных с открытым исходным кодом. Она поддерживает транзакции, различные типы индексов, сложные SQL-запросы и работу с полуструктурированными данными через JSONB.

Существенное значение имеет поддержка модели ACID, включающей атомарность, согласованность, изолированность и долговечность транзакций. Эти свойства востребованы в финансовых приложениях, системах управления образовательными процессами и электронном документообороте.

Нормализация структуры базы данных позволяет уменьшить избыточность информации и поддерживать согласованность связанных записей. При этом запросы, объединяющие большое количество таблиц, могут требовать значительных вычислительных ресурсов. Их производительность определяется структурой данных, качеством индексации и возможностями оптимизатора запросов.

1.2. Архитектурные особенности NoSQL-систем

NoSQL представляет собой совокупность подходов к организации баз данных, использующих различные способы представления информации. В зависимости от архитектуры выделяют четыре основные категории.

Документно-ориентированные базы данных. Информация хранится в виде документов, обычно представленных в формате BSON или JSON. Документ может содержать вложенные объекты и массивы. Такая структура позволяет объединять связанные атрибуты в одном информационном объекте.

Базы данных ключ-значение. Данные представлены парами, в которых уникальный ключ обеспечивает доступ к соответствующему значению. Подобные решения применяются для кэширования, хранения пользовательских сессий и быстрого получения объектов.

Колоночные базы данных. Информация организуется с использованием семейств столбцов. Такая архитектура применяется в распределённых системах хранения и обработки больших информационных массивов.

Графовые базы данных. Основными элементами являются вершины и рёбра, представляющие объекты и связи между ними. Графовая модель используется в социальных сетях, рекомендательных системах и приложениях анализа взаимосвязей.

В настоящем исследовании рассматривается документно-ориентированная модель MongoDB. Её архитектура позволяет сопоставить способы хранения одних и тех же информационных объектов в документной и реляционной формах.

1.3. Сравнительная характеристика MongoDB и PostgreSQL

MongoDB представляет собой документно-ориентированную СУБД, в которой информация хранится в коллекциях в виде BSON-документов. Такая организация позволяет объединять связанные атрибуты объекта в единую структуру.

PostgreSQL использует реляционную модель с поддержкой JSON и JSONB. Это позволяет хранить полуструктурированные объекты непосредственно в таблицах и выполнять запросы к их содержимому.

Таблица 1. Сравнительная характеристика MongoDB и PostgreSQL

Критерий	MongoDB	PostgreSQL
Модель данных	Документная	Реляционная
Основная единица хранения	BSON-документ	Строка таблицы
Организация структуры	Гибкая	Схема таблиц
Связи между объектами	Вложенные документы, ссылки	Первичные и внешние ключи
Язык запросов	MongoDB Query API	SQL

Критерий	MongoDB	PostgreSQL
Индексация	Одиночные, составные, текстовые, геопространственные	B-tree, Hash, GIN, GiST, BRIN и другие
Транзакции	Поддерживаются	Поддерживаются
Полуструктурированные данные	Нативная документная модель	JSON и JSONB
Аналитическая обработка	Aggregation Pipeline	SQL, оконные функции, CTE
Масштабирование	Репликация и шардинг	Репликация и различные механизмы масштабирования
Основная область применения	Документные приложения, гибкие структуры данных	Транзакционные и аналитические системы

Сравнение показывает, что обе СУБД обладают широкими функциональными возможностями. Различия проявляются в способах представления информации, организации связей и выполнении запросов. Эти особенности необходимо учитывать при разработке экспериментальной методики.

2. Материалы и методы исследования

2.1. Общая схема экспериментального исследования

Для достижения цели работы разработана методика сравнительного тестирования MongoDB и PostgreSQL. Она предусматривает выполнение одинаковых логических операций над идентичными наборами информации.

Эксперимент включает следующие этапы:

1. Формирование тестового набора данных.
2. Создание соответствующих структур хранения в обеих СУБД.
3. Загрузка идентичной информации.
4. Проверка корректности и полноты загруженных данных.
5. Выполнение тестовых операций.
6. Сбор показателей производительности.
7. Статистическая обработка результатов.

Для обеспечения воспроизводимости эксперимента необходимо зафиксировать версии СУБД, аппаратные характеристики сервера, конфигурацию оперативной памяти, тип накопителя, настройки индексации и параметры подключения.

В качестве экспериментальной среды предлагается использовать локальный сервер, на котором обе СУБД работают на одинаковом оборудовании.

Таблица 2. Предлагаемая конфигурация экспериментальной среды

Параметр	Конфигурация
Операционная система	Ubuntu Linux 24.04 LTS
Процессор	8 вычислительных ядер
Оперативная память	32 ГБ
Накопитель	SSD NVMe
MongoDB	Версия 8.x
PostgreSQL	Версия 17.x или 18.x
Язык тестирования	Python 3.12
Драйвер MongoDB	PyMongo
Драйвер PostgreSQL	psycopg
Формат исходных данных	JSON

Параметр	Конфигурация
Инструмент мониторинга	Prometheus и системные средства Linux

Указанная конфигурация является рекомендуемой экспериментальной средой. В итоговом протоколе необходимо привести фактические характеристики оборудования, точные версии программного обеспечения и параметры конфигурации обеих СУБД.

2.2. Формирование тестового набора данных

Для проведения сравнительного анализа предлагается сформировать синтетический набор данных, моделирующий информационную систему управления пользователями, образовательными курсами и результатами выполнения заданий.

Выбор предметной области обусловлен наличием различных типов информации. К ним относятся фиксированные атрибуты пользователей, характеристики курсов, результаты тестирования и история активности.

Каждая запись содержит следующие логические атрибуты:

– идентификатор пользователя;
– фамилию и имя;
– адрес электронной почты;
– дату регистрации;
– идентификатор курса;
– название курса;
– категорию курса;
– список выполненных заданий;
– результаты тестирования;
– временные отметки активности.

В MongoDB информация хранится в виде документов с вложенными массивами результатов обучения. В PostgreSQL используется нормализованная структура из таблиц users, courses, enrollments и task_results.

Пример документной структуры MongoDB:

```
{
  "_id": 10001,
  "name": "User 10001",
  "email": "user10001@example.com",
  "registration_date": "2026-01-15",
  "courses": [
    {
      "course_id": 101,
      "title": "Database Systems",
      "category": "IT",
      "progress": 85,
      "results": [
        {
          "task_id": 1,
          "score": 90,
          "completed": true
        },
        {
          "task_id": 2,
          "score": 80,
          "completed": true
        }
      ]
    }
```

]
}

Для исследования масштабируемости предусматриваются три объёма тестового массива:

– 100 000 записей;
– 500 000 записей;
– 1 000 000 записей.

Во всех конфигурациях должно сохраняться одинаковое логическое содержание информации. Дополнительно необходимо контролировать распределение значений, количество вложенных объектов и размер записей.

2.3. Сценарии экспериментальной нагрузки

Эксперимент предусматривает пять основных сценариев тестирования.

Сценарий 1. Операции записи.

Измеряется время добавления новых пользователей и связанных с ними данных. В MongoDB используется операция insertMany, в PostgreSQL выполняется пакетная вставка посредством INSERT с подготовленными параметрами.

Сценарий 2. Операции чтения.

Исследуется скорость получения отдельного объекта по идентификатору, поиск по атрибутам и выборка пользователей по заданным условиям.

Сценарий 3. Операции обновления.

Измеряется время изменения отдельных атрибутов и вложенных элементов. Для MongoDB применяются updateOne и updateMany, для PostgreSQL используется UPDATE.

Сценарий 4. Аналитические запросы.

Проводится сравнение выполнения группировки пользователей, вычисления среднего результата тестирования, определения количества завершённых заданий и формирования агрегированных показателей по курсам.

Сценарий 5. Масштабируемость.

Исследуется изменение времени выполнения операций при увеличении объёма информационного массива от 100 000 до 1 000 000 записей.

Для каждого сценария предусматривается выполнение нескольких повторений. Первые два запуска исключаются из итоговой статистики в качестве прогревочных. Основные измерения выполняются не менее 30 раз для каждого сочетания параметров при неизменной конфигурации среды.

2.4. Критерии оценки производительности

Для комплексного анализа используются следующие показатели.

Среднее время выполнения операции:

$T_avg = (1/n)\ \Sigma_{i=1}^{n}\ t_i$ где t_i представляет время выполнения отдельной операции, n обозначает количество измерений.

Медианное время выполнения:

$T_med = Median(t_1, t_2, \ldots, t_n)$

Медиана позволяет уменьшить влияние единичных аномальных измерений.

Пропускная способность:

TPS = N/T

где N представляет количество успешно выполненных операций, T обозначает общее время тестирования.

Коэффициент относительного ускорения:

S = T_PostgreSQL / T_MongoDB

При S>1 MongoDB демонстрирует меньшее время выполнения соответствующей операции. При S<1 преимущество по времени имеет PostgreSQL.

Изменение производительности при увеличении объёма данных:

$K_scale = T_N_2 / T_N_1$ где N_1 и N_2 представляют два различных объёма данных, а T_N_1 и T_N_2 обозначают соответствующее время выполнения одного и того же запроса.

Дополнительно регистрируются перцентили P95 и P99, объём используемой оперативной памяти, размер хранилища, количество прочитанных документов и индексных ключей.

Для статистической обработки результатов необходимо рассчитывать доверительные интервалы и оценивать разброс измерений. При сравнении распределений времени выполнения операций могут применяться критерий Манна-Уитни или t-критерий Стьюдента. Выбор критерия определяется свойствами полученных данных.

2.5. Обеспечение корректности сравнения

Сопоставимость результатов является обязательным условием экспериментального исследования.

Необходимо обеспечить следующие условия:

– одинаковый исходный набор данных;
– одинаковое количество логических записей;
– сопоставимые индексы;
– одинаковый сценарий нагрузки;
– идентичное оборудование;
– одинаковые параметры конкурентного доступа;
– контроль состояния кэша;
– раздельный учёт времени передачи данных и выполнения запросов.

Особое значение имеет индексация. Сравнение индексированной конфигурации одной СУБД с неиндексированной конфигурацией другой приведёт к некорректным выводам. Поэтому рекомендуется провести два самостоятельных эксперимента: без дополнительных индексов и с оптимизированными индексами.

Для анализа выполнения запросов в MongoDB используется механизм explain с режимом executionStats. В PostgreSQL применяется EXPLAIN (ANALYZE, BUFFERS). Полученные планы запросов позволяют изучить используемые алгоритмы, объём обработанной информации и характер обращения к индексам.

3. Результаты исследования и сравнительный анализ

3.1. Архитектурные результаты сравнения

Сопоставление двух систем показывает, что различия между ними проявляются прежде всего в способе представления взаимосвязанных объектов.

Документная модель MongoDB позволяет объединять связанные атрибуты в одном документе. При получении полного объекта пользователя вместе с его курсами и результатами заданий это может сократить количество отдельных операций чтения.

В PostgreSQL нормализованная модель предполагает обращение к нескольким таблицам при получении аналогичной информации. Оптимизатор запросов использует индексы и различные алгоритмы соединения, что позволяет эффективно обрабатывать взаимосвязанные сущности.

Архитектурный анализ показывает, что производительность каждой системы определяется структурой хранимой информации и характером запросов.

3.2. Результаты анализа операций CRUD

Для сравнения основных операций используются унифицированные логические сценарии.

Таблица 3. Сравнительная матрица CRUD-операций

Операция	**MongoDB**	**PostgreSQL**	**Основной фактор влияния**
Создание записи	insertOne / insertMany	INSERT	Размер объекта, пакетная запись
Чтение по ID	findOne	SELECT	Индексация первичного ключа

Операция	MongoDB	PostgreSQL	Основной фактор влияния
Фильтрация	find	SELECT WHERE	Селективность условия
Обновление записи	updateOne	UPDATE	Объём изменяемых данных
Удаление записи	deleteOne	DELETE	Индексация и ограничения
Пакетная запись	insertMany	INSERT	Размер пакета и транзакции
Агрегация	Aggregation Pipeline	GROUP BY	Структура данных и сложность запроса

Архитектурные особенности позволяют выделить несколько закономерностей.

При работе с документами, структура которых соответствует модели приложения, MongoDB позволяет выполнять многие операции без предварительного объединения нескольких сущностей.

PostgreSQL располагает развитыми средствами обработки взаимосвязанных таблиц. Это особенно важно при выполнении запросов, включающих объединение информации, группировку и сложные логические условия.

Операции обновления в обеих системах зависят от размера изменяемого объекта, структуры индексов и характера транзакционной нагрузки. Для получения достоверных результатов необходимо измерять их выполнение в одинаковых условиях.

3.3. Влияние индексации на производительность

Индексация существенно влияет на скорость доступа к информации. Её эффективность определяется структурой индекса, селективностью условий поиска и характером запросов.

В MongoDB поддерживаются одиночные, составные, многоключевые, текстовые и другие типы индексов. PostgreSQL предлагает B-tree, Hash, GIN, GiST, BRIN и дополнительные механизмы индексации.

Для экспериментального исследования рассматриваются три конфигурации:

1. Отсутствие дополнительных индексов.
2. Создание индексов по часто используемым атрибутам.
3. Использование составных индексов для сложных условий поиска.

Таблица 4. План исследования влияния индексации

Конфигурация	MongoDB	PostgreSQL
Базовая	Индекс по _id	Первичный ключ
Поиск пользователя	Индекс email	Индекс по email
Поиск по курсу	Индекс courses.course_id	Индекс course_id
Составной поиск	Составной индекс	Составной B-tree
Аналитические запросы	Индексы по фильтрам	B-tree / GIN по характеру запроса

Эффективность индексации оценивается по времени выполнения запросов, количеству просмотренных документов или строк, числу прочитанных индексных ключей и дополнительным затратам на хранение индексов.

Увеличение их количества может ускорить чтение, однако одновременно возрастает стоимость операций записи и обновления. Поэтому при проектировании базы данных необходимо учитывать общий профиль нагрузки.

3.4. Анализ аналитических запросов

Аналитическая обработка представляет интерес для информационных систем, в которых требуется выполнять группировку и обобщение больших объёмов данных.

В рамках рассматриваемой предметной области выделены три типа запросов:

– определение среднего результата пользователей по каждому курсу;

– определение количества завершённых заданий;

– формирование рейтинга курсов по активности пользователей.

В PostgreSQL эти операции реализуются посредством GROUP BY, JOIN, агрегатных функций и оконных функций.

В MongoDB для аналогичных задач применяется Aggregation Pipeline, включающий стадии $match, $unwind, $group, $sort и $project.

Отдельного внимания требует разворачивание вложенных массивов. При использовании $unwind количество обрабатываемых элементов может существенно превышать количество исходных документов.

В реляционной модели результаты заданий представлены отдельными строками. Это позволяет применять агрегатные функции непосредственно к таблицам. На производительность также влияют количество соединений и качество индексации.

Поэтому при сравнении аналитических запросов необходимо учитывать не только время выполнения, но и объём фактически обработанной информации.

3.5. Влияние объёма данных на масштабируемость

Для исследования масштабируемости предлагается использовать три уровня информационной нагрузки: 100 тысяч, 500 тысяч и 1 миллион логических объектов.

Таблица 5. Матрица экспериментального исследования масштабируемости

Объём данных	MongoDB: время, мс	PostgreSQL: время, мс	MongoDB: память, МБ	PostgreSQL: память, МБ
100 000	Измерить	Измерить	Измерить	Измерить
500 000	Измерить	Измерить	Измерить	Измерить
1 000 000	Измерить	Измерить	Измерить	Измерить

Получение экспериментальных значений является обязательным условием формулирования количественных выводов о масштабируемости.

При увеличении объёма данных может изменяться поведение оптимизаторов запросов, увеличиваться размер индексов и возрастать нагрузка на оперативную память. Дополнительные затраты возникают при обращении к дисковым подсистемам.

Для MongoDB существенное значение имеют организация документов, размер вложенных структур и распределение информации по шардам при использовании горизонтального масштабирования.

В PostgreSQL на производительность влияют структура таблиц, селективность индексов, статистика планировщика, объём буферного кэша и возможности параллельного выполнения запросов.

Полученные измерения позволят определить характер изменения времени выполнения операций при увеличении информационного массива и выявить наиболее чувствительные к объёму данных сценарии.

3.6. Сводная интерпретация результатов

На основании архитектурного анализа сформирована качественная матрица применимости исследуемых систем.

Таблица 6. Предварительная оценка применимости MongoDB и PostgreSQL

Сценарий использования	MongoDB	PostgreSQL
Хранение автономных	Высокая применимость	Высокая применимость

Сценарий использования	MongoDB	PostgreSQL
документов		
Часто изменяемая структура объектов	Высокая применимость	Требуется управление схемой
Сложные межтабличные связи	Возможна реализация, требует дополнительного моделирования	Высокая применимость
Транзакционные операции	Поддерживаются	Поддерживаются
Сложная SQL-аналитика	Ограничения относительно SQL-подхода	Высокая применимость
Обработка вложенных документов	Нативная поддержка	Поддержка JSONB
Горизонтальное распределение	Встроенный механизм шардинга	Требует выбора соответствующей архитектуры
Гибридные структурированные данные	Высокая применимость	Высокая применимость

Таблица отражает архитектурную применимость СУБД и не содержит результатов измерения скорости выполнения операций.

4. Обсуждение результатов

Архитектурный анализ показывает, что выбор между MongoDB и PostgreSQL должен учитывать особенности предметной области и профиль нагрузки информационной системы.

Документная модель MongoDB подходит для приложений, в которых информационные объекты имеют сложную или изменяемую структуру. Она позволяет объединять связанные атрибуты и получать целые документы без выполнения большого количества отдельных запросов.

PostgreSQL предоставляет развитые механизмы работы с взаимосвязанными сущностями, ограничениями целостности и сложными аналитическими операциями. Поддержка JSONB расширяет возможности системы при обработке полуструктурированных данных.

При этом гибкость структуры MongoDB не означает отсутствия требований к проектированию. Необходимо учитывать ограничения размера документа, особенности вложенных массивов и стоимость обновления крупных объектов.

Аналогично реляционная модель требует внимательного проектирования схемы, выбора индексов и оптимизации запросов. Неудачная нормализация или избыточные соединения могут снижать производительность приложения.

Транзакционные механизмы также присутствуют в обеих системах. Поэтому утверждение о принципиальной неспособности NoSQL-систем обеспечивать согласованность информации не соответствует современному состоянию технологий.

Практическое применение рассматриваемых СУБД можно рассмотреть на примере образовательной платформы. PostgreSQL может использоваться для хранения учётных записей, регистраций на курсы и результатов финансовых операций. MongoDB подходит для хранения динамических журналов активности, событий мониторинга и документов с изменяемой структурой.

Подобное разделение позволяет распределить информационную нагрузку между системами с учётом особенностей хранимых объектов. Однако его целесообразность должна подтверждаться анализом требований к интеграции, сопровождению и стоимости эксплуатации.

Полученные методические положения также создают основу для дальнейшего исследования гибридных архитектур, объединяющих реляционные и документные модели.

5. Научная новизна и практическая значимость

Научная новизна исследования состоит в разработке комплексной методики сравнительной оценки документно-ориентированной и реляционной моделей хранения данных. Методика предусматривает совместное исследование производительности CRUD-операций, аналитических запросов, индексации и масштабируемости.

Предложенный подход учитывает взаимосвязь между четырьмя факторами:

1. моделью представления данных;
2. структурой и селективностью запросов;
3. конфигурацией индексации;
4. объёмом обрабатываемой информации.

Дополнительным результатом является формирование матрицы выбора СУБД с учётом характера прикладной нагрузки и особенностей информационных объектов.

Практическая значимость заключается в возможности применения методики при проектировании информационных систем, цифровых образовательных платформ, корпоративных приложений и аналитических сервисов.

Предложенные сценарии тестирования могут использоваться в учебном процессе при изучении современных систем управления базами данных. Они позволяют объединить теоретическое рассмотрение архитектуры SQL и NoSQL с практическим тестированием и анализом производительности.

Заключение

В исследовании рассмотрены архитектурные особенности реляционных и документно-ориентированных систем управления базами данных на примере PostgreSQL и MongoDB.

Разработана методика сравнительного экспериментального исследования, включающая формирование идентичных наборов данных, выполнение CRUD-операций, тестирование аналитических запросов и оценку изменения производительности при увеличении объёма информации. Для анализа определены среднее и медианное время выполнения запросов, перцентили задержки, пропускная способность, использование оперативной памяти и эффективность индексации.

Архитектурный анализ показал, что MongoDB предоставляет развитые возможности представления автономных объектов и полуструктурированных данных. PostgreSQL обеспечивает широкий набор средств управления взаимосвязанными сущностями, транзакциями и аналитическими запросами.

Установлено, что PostgreSQL может использоваться для хранения полуструктурированной информации благодаря поддержке JSONB и соответствующих механизмов индексации. MongoDB, в свою очередь, поддерживает транзакционные операции и механизмы обеспечения согласованности данных.

Производительность рассматриваемых систем определяется структурой хранимой информации, характером запросов, конфигурацией индексов и объёмом информационного массива. Эти факторы необходимо учитывать при выборе СУБД для конкретного приложения.

Количественная проверка сформулированных положений требует проведения описанного эксперимента и статистической обработки измерений. Дальнейшие исследования целесообразно направить на изучение конкурентного доступа, распределённых конфигураций и гибридных архитектур хранения данных.

Список литературы:

1. Codd E.F. A Relational Model of Data for Large Shared Data Banks // Communications of the ACM. – 1970. – Vol. 13, No. 6. – P. 377–387. – DOI: https://doi.org/10.1145/362384.362685.

2. Brewer E.A. Towards Robust Distributed Systems // Proceedings of the 19th Annual ACM Symposium on Principles of Distributed Computing. – 2000. – DOI: https://doi.org/10.1145/343477.343502.

3. Gilbert S., Lynch N. Perspectives on the CAP Theorem // Computer. – 2012. – Vol. 45, No. 2. – P. 30–36. – DOI: https://doi.org/10.1109/MC.2011.389.

4. Chang F., Dean J., Ghemawat S. et al. Bigtable: A Distributed Storage System for Structured Data // Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation (OSDI). – 2006. – P. 205–218. – URL: https://www.usenix.org/legacy/event/osdi06/tech/chang/chang.pdf.

5. DeCandia G., Hastorun D., Jampani M. et al. Dynamo: Amazon's Highly Available Key-value Store // Proceedings of the 21st ACM Symposium on Operating Systems Principles. – 2007. – P. 205–220. – DOI: https://doi.org/10.1145/1294261.1294281.

6. Cooper B.F., Silberstein A., Tam E., Ramakrishnan R., Sears R. Benchmarking Cloud Serving Systems with YCSB // Proceedings of the 1st ACM Symposium on Cloud Computing. – 2010. – P. 143–154. – DOI: https://doi.org/10.1145/1807128.1807152.

7. Makris A., Tserpes K., Spiliopoulos G., Zissis D., Anagnostopoulos D. MongoDB Vs PostgreSQL: A Comparative Study on Performance Aspects // Geoinformatica. – 2021. – Vol. 25. – P. 243–268. – DOI: https://doi.org/10.1007/s10707-020-00407-w.

8. Carvalho I., Sá F., Bernardino J. Performance Evaluation of NoSQL Document Databases: Couchbase, CouchDB, and MongoDB // Algorithms. – 2023. – Vol. 16, No. 2. – Article 78. – DOI: https://doi.org/10.3390/a16020078.

9. Kpekpassi D., Faye D. NoSQL Databases: A Survey // ARIMA. – 2025. – Vol. 43. – DOI: https://doi.org/10.46298/arima.13970.

10. Urnikienė J., Steponavičienė V., Atanasov S. Comparative Read Performance Analysis of PostgreSQL and MongoDB in E-Commerce: An Empirical Study of Filtering and Analytical Queries // Big Data and Cognitive Computing. – 2026. – Vol. 10, No. 2. – Article 66. – DOI: https://doi.org/10.3390/bdcc10020066.

11. Stonebraker M., Çetintemel U. "One Size Fits All": An Idea Whose Time Has Come and Gone // Proceedings of the 21st International Conference on Data Engineering. – 2005. – P. 2–11. – DOI: https://doi.org/10.1109/ICDE.2005.1.

12. Sadalage P.J., Fowler M. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence. – Boston: Addison-Wesley, 2012. – 192 p.

13. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. – Sebastopol: O'Reilly Media, 2017. – 616 p.

14. MongoDB Inc. MongoDB Manual: Explain Results. – URL: https://www.mongodb.com/docs/manual/reference/explain-results/ (дата обращения: 18.11.2026).

15. MongoDB Inc. MongoDB Manual: Aggregation Pipeline. – URL: https://www.mongodb.com/docs/manual/core/aggregation-pipeline/ (дата обращения: 18.11.2026).

16. PostgreSQL Global Development Group. PostgreSQL Documentation: EXPLAIN. – URL: https://www.postgresql.org/docs/current/sql-explain.html (дата обращения: 18.11.2026).

17. PostgreSQL Global Development Group. PostgreSQL Documentation: Indexes. – URL: https://www.postgresql.org/docs/current/indexes.html (дата обращения: 18.11.2026).

18. PostgreSQL Global Development Group. PostgreSQL Documentation: JSON Types. – URL: https://www.postgresql.org/docs/current/datatype-json.html (дата обращения: 18.11.2026).