

УДК 004.6

**Мешков Никита Сергеевич**, студент кафедры  
«Автоматизированных систем управления  
биотехнологическими процессами»  
Российский биотехнологический университет  
(РОСБИОТЕХ)

## ОБЗОР СОВРЕМЕННЫХ СИСТЕМ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ ВРЕМЕННЫХ РЯДОВ: TIMESCALEDB, INFLUXDB, QUESTDB И CLICKHOUSE

**Аннотация.** В статье представлен обзор TimescaleDB, InfluxDB, QuestDB и ClickHouse как систем для работы с временными рядами. Рассмотрены их архитектурные особенности, производительность, масштабируемость, преимущества и ограничения. Отдельное внимание уделено практическим сценариям применения и перспективам интеграции с аналитическими и потоковыми платформами.

**Ключевые слова:** Временные ряды, базы данных, TimescaleDB, InfluxDB, QuestDB, ClickHouse, аналитика данных.

Современные цифровые системы генерируют колоссальные объемы данных, среди которых значительную долю составляют временные ряды – последовательности значений, привязанных ко времени [3]. Они лежат в основе аналитики и мониторинга в таких сферах, как интернет вещей, телекоммуникации, финансы, промышленная автоматизация и ИТ-инфраструктура [1]. В этих областях требуется не просто хранить данные, а обеспечивать их быструю обработку, агрегацию и визуализацию, что делает выбор специализированной базы данных ключевым фактором эффективности всей системы [2].

Традиционные реляционные СУБД не рассчитаны на интенсивные вставки данных с высокой частотой и обработку запросов, основанных на временных интервалах [4]. Это привело к появлению целого класса решений – систем управления базами данных временных рядов (Time Series Databases, TSDB). Они оптимизированы под хранение данных, где временной штамп является главным индексом, и предоставляют специализированные инструменты для анализа, агрегации и построения графиков [5].

На рынке таких решений существует несколько популярных платформ, каждая из которых реализует собственный подход к организации и обработке данных. Среди наиболее известных – TimescaleDB, InfluxDB, QuestDB и ClickHouse [6]. Несмотря на общую направленность, эти системы заметно различаются по архитектуре, принципам масштабирования, языкам запросов и областям применения. Например, TimescaleDB строится на основе PostgreSQL и ориентирована на совместимость и гибкость SQL-запросов [2], тогда как InfluxDB создавалась как полностью специализированная СУБД для временных рядов с собственным языком запросов и встроенными инструментами мониторинга [9]. QuestDB делает ставку на производительность при потоковой обработке данных, а ClickHouse, изначально разработанный для аналитических нагрузок, всё чаще применяется и для хранения временных рядов благодаря высокой скорости выборки и сжатию данных [7].

Цель данной обзорной статьи – провести сравнительный анализ архитектуры, производительности и функциональных возможностей перечисленных систем [10]. Для достижения поставленной цели решаются следующие задачи:

- Рассмотреть архитектурные особенности и принципы работы TimescaleDB, InfluxDB, QuestDB и ClickHouse [3];
- Сравнить производительность, масштабируемость и функциональные возможности выбранных систем [6];
- Определить практические сценарии применения каждой СУБД и оценить перспективы их дальнейшего развития [8].



Объектом работы выступают современные системы управления базами данных временных рядов, а предметом – их архитектурные, функциональные и эксплуатационные характеристики [4]. Практическая значимость статьи заключается в систематизации ключевых особенностей популярных решений для работы с временными рядами и формировании представления о том, какие инструменты наиболее целесообразно использовать в конкретных прикладных сценариях [1]. Полученные выводы могут быть полезны специалистам по аналитике данных, разработчикам и инженерам DevOps при выборе и внедрении систем хранения временных данных [5].

В результате проведенного обзора будет показано, какие решения демонстрируют наилучший баланс между производительностью, гибкостью и удобством интеграции, а также какие направления развития данных систем представляют наибольший интерес в контексте роста объемов временных данных и усложнения аналитических задач [7].

Архитектурные особенности и принципы работы СУБД временных рядов.

Хранение и обработка временных рядов требуют специфического подхода к проектированию баз данных [4]. В отличие от традиционных реляционных систем, где основное внимание уделяется связям между сущностями, в СУБД временных рядов ключевым элементом является временной штамп [3]. От того, насколько эффективно система организует запись, хранение и доступ к данным, зависит её пригодность для задач мониторинга, аналитики и прогнозирования [1].

Временные ряды характеризуются высокой скоростью генерации данных, большими объемами и преобладанием операций вставки над обновлением [5]. Эти особенности привели к появлению специализированных архитектур, которые обеспечивают эффективную работу с непрерывно поступающими потоками данных. Среди них можно выделить четыре наиболее известных решения: TimescaleDB, InfluxDB, QuestDB и ClickHouse [2]. Несмотря на общую направленность, каждая из систем реализует собственные подходы к организации хранения, индексированию и масштабированию данных [6].

TimescaleDB представляет собой расширение PostgreSQL, добавляющее поддержку временных рядов без необходимости менять базовую архитектуру [10]. Её ключевая идея заключается в использовании hypertables – логических таблиц, которые автоматически разбиваются на chunks по времени и пространству [1]. Такой подход позволяет эффективно распределять данные и использовать встроенные возможности PostgreSQL, включая транзакции, индексы и язык SQL [2]. Благодаря этому TimescaleDB обеспечивает совместимость с существующими инструментами и библиотеками, что делает её особенно удобной для компаний, уже использующих PostgreSQL в инфраструктуре [4]. Кроме того, система поддерживает непрерывные агрегаты (continuous aggregates), позволяющие автоматически обновлять агрегированные представления данных без необходимости постоянного пересчёта [8].

InfluxDB, напротив, создавалась как самостоятельная СУБД, полностью ориентированная на временные ряды [3]. Её архитектура построена вокруг специализированного движка хранения TSM (Time Structured Merge Tree), оптимизированного для последовательных записей [9]. Данные в InfluxDB организованы в измерения (measurements), теги (tags) и поля (fields), что обеспечивает гибкую структуру и высокую скорость вставки [7]. Для запросов используется язык InfluxQL, схожий с SQL, и более современный Flux, позволяющий выполнять аналитические операции, агрегации и объединения данных [5]. InfluxDB хорошо подходит для задач мониторинга систем и IoT, где важна стабильная запись большого объёма метрик в реальном времени [9].

QuestDB выделяется ориентацией на максимальную производительность и низкие задержки [6]. Архитектура системы спроектирована с акцентом на обработку потоковых данных и поддержку SQL-совместимого интерфейса [7]. QuestDB применяет столбцовую модель хранения и оптимизацию чтения из оперативной памяти, что обеспечивает крайне быстрые операции выборки [10]. Дополнительное преимущество – возможность работать с



данными в режиме real-time ingestion через протоколы Influx Line Protocol и PostgreSQL Wire Protocol [2]. Это делает систему удобной для интеграции с уже существующими решениями и инструментами визуализации [8].

ClickHouse, разработанный компанией «Яндекс», изначально создавался как аналитическая колоночная СУБД, предназначенная для обработки больших объёмов данных [7]. Однако благодаря высокой производительности и эффективному сжатию данных он также широко применяется для временных рядов [3]. ClickHouse хранит данные в виде столбцов и использует мощные алгоритмы компрессии, что позволяет обрабатывать миллиарды записей за секунды [4]. Основная особенность системы – выполнение запросов векторизованным способом, что существенно ускоряет вычисления [5]. Хотя ClickHouse не является специализированной TSDB, его возможности позволяют решать схожие задачи, особенно при анализе логов, событий или метрик [1].

Сравнивая подходы, можно отметить, что TimescaleDB ориентирована на совместимость и SQL-гибкость [10], InfluxDB – на простоту и устойчивость к нагрузкам при записи данных [3], QuestDB – на скорость потоковой обработки и низкие задержки [6], а ClickHouse – на масштабную аналитику и высокую пропускную способность при чтении [7]. В результате выбор конкретного решения определяется приоритетами проекта: если важна интеграция с существующими реляционными системами – целесообразно использовать TimescaleDB [4]; если критична скорость вставки данных – InfluxDB [9]; если требуется мгновенный анализ в реальном времени – QuestDB [6]; а при необходимости сложной аналитики и агрегаций на больших объёмах – ClickHouse [7].

Эти архитектурные особенности формируют основу для последующего анализа производительности и функциональных возможностей систем, что позволяет объективно оценить их применимость в реальных сценариях обработки временных данных [8].

Производительность, масштабируемость и функциональные возможности

Ключевыми критериями эффективности любой системы управления базами данных временных рядов считаются скорость записи и чтения, масштабируемость, способность обрабатывать большие объёмы данных без деградации производительности, а также наличие аналитических инструментов [5]. В современных условиях, когда устройства и сервисы генерируют миллионы метрик в секунду, от этих характеристик напрямую зависит применимость системы в промышленной и аналитической среде [3].

TimescaleDB демонстрирует стабильную производительность благодаря интеграции с PostgreSQL и использованию гибкого механизма hypertables [10]. Запись данных выполняется последовательно, при этом система эффективно использует индексирование по времени. Одним из преимуществ является возможность параллельной обработки запросов и автоматической агрегации данных с помощью continuous aggregates [1]. Это особенно полезно в случаях, когда требуется формировать отчёты или графики на основе больших массивов временных данных [4]. Однако при высоких нагрузках на вставку TimescaleDB может уступать специализированным решениям, так как сохраняет транзакционную модель PostgreSQL, которая требует дополнительных накладных расходов на обработку метаданных и индексов [7].

InfluxDB показывает выдающуюся производительность при записи данных, что делает её одной из наиболее популярных систем для задач мониторинга и IoT [9]. Её архитектура ориентирована на минимизацию операций записи на диск, благодаря чему достигается высокая пропускная способность [2]. Внутренний движок TSM поддерживает эффективное сжатие данных и хранение в виде последовательных блоков, что ускоряет операции вставки и чтения [5]. Тем не менее при сложных аналитических запросах производительность может снижаться, особенно при большом числе тегов и многомерных данных [8]. В таких случаях разработчики рекомендуют использовать агрегированные представления или экспортировать данные в аналитические платформы [3].

QuestDB делает ставку на скорость обработки и минимальные задержки [6]. Она спроектирована для выполнения запросов в реальном времени и показывает высокую



производительность даже на одном сервере без кластеризации [10]. Благодаря столбцовому формату хранения и векторным вычислениям QuestDB способна выполнять выборки и агрегации в миллионы строк за миллисекунды [7]. Также система поддерживает конкурентную запись и чтение, что особенно важно при работе с потоками данных [1]. Несмотря на отсутствие нативной кластеризации, производительность QuestDB на уровне одного узла делает её подходящим решением для сценариев, где важна низкая латентность – например, в торговых системах или мониторинге инфраструктуры [6].

ClickHouse выделяется среди всех участников благодаря масштабируемости и способности обрабатывать петабайты данных [4]. Его архитектура оптимизирована под аналитические нагрузки, при этом система эффективно справляется и с временными рядами [2]. Использование колоночного хранения и векторных вычислений позволяет ClickHouse обеспечивать очень высокую скорость выполнения запросов на чтение и агрегацию [5]. Кластеризация и горизонтальное масштабирование реализованы на уровне системы, что даёт возможность гибко распределять данные и вычислительные ресурсы между узлами [7]. Однако при вставке большого объёма данных в реальном времени могут возникать ограничения из-за необходимости периодической консолидации микроблоков в более крупные сегменты [8]. Несмотря на это, при оптимальной настройке и использовании `buffer tables` ClickHouse демонстрирует отличное соотношение скорости и стабильности [9].

Если рассматривать масштабируемость, то наиболее гибким решением является ClickHouse – благодаря встроенной поддержке распределённых таблиц и репликации [7]. TimescaleDB обеспечивает горизонтальное масштабирование через механизм `multi-node`, но требует дополнительной настройки и балансировки [10]. InfluxDB реализует кластерный режим в корпоративной версии, что накладывает ограничения для пользователей бесплатной редакции [2]. QuestDB в текущей версии остаётся односерверной, однако её высокая скорость частично компенсирует отсутствие кластеризации [6].

По части функциональных возможностей различия также существенны [4]. TimescaleDB выигрывает за счёт полной поддержки SQL и совместимости с аналитическими инструментами – от Power BI до Grafana [10]. InfluxDB предлагает собственный язык Flux, ориентированный на потоковую обработку и гибкие вычисления, включая корреляции и прогнозирование [9]. QuestDB использует стандартный SQL и поддерживает временные функции, что делает работу с ней интуитивной для разработчиков [3]. ClickHouse предоставляет богатый набор аналитических функций, включая оконные операции, агрегаты и средства машинного обучения, что делает его мощным инструментом для обработки больших временных массивов [7].

Сравнение показывает, что InfluxDB обеспечивает наилучшую производительность при записи данных [9], QuestDB – при потоковой обработке и запросах с низкой задержкой [6], TimescaleDB – при интеграции с традиционными SQL-системами [10], а ClickHouse – при выполнении сложных аналитических запросов и работе с огромными объёмами информации [4].

В результате можно отметить, что выбор оптимальной системы зависит от приоритетов проекта: если требуется высокая скорость записи – подойдёт InfluxDB [9]; если критична скорость отклика и низкая латентность – QuestDB [6]; если важна гибкость SQL и надёжность транзакций – TimescaleDB [10]; если основная цель – масштабная аналитика и работа с большими данными – ClickHouse станет лучшим выбором [4].3.

В заключении произведем сравнительный анализ ключевых характеристик TimescaleDB, InfluxDB, QuestDB и ClickHouse.

Для объективной оценки систем управления базами данных временных рядов важно сопоставить их по ряду технических параметров: производительности, масштабируемости, поддержке SQL, архитектуре хранения и применимости в различных сценариях. Такой сравнительный подход позволяет увидеть сильные и слабые стороны каждой платформы и определить их оптимальные области использования.



На основании анализа официальной документации, тестов сообщества и описанных ранее архитектурных особенностей можно выделить следующие ключевые различия между рассматриваемыми СУБД.

Таблица 1

**Сравнение основных характеристик систем хранения временных рядов**

Параметр	TimescaleDB	InfluxDB	QuestDB	ClickHouse
<b>Тип архитектуры</b>	Расширение PostgreSQL с поддержкой hypertables	Специализированная TSDB с собственным движком TSM	Высокопроизводительная столбцовая СУБД	Колоночная аналитическая СУБД
<b>Язык запросов</b>	Полный SQL (PostgreSQL)	InfluxQL / Flux	SQL (ANSI)	SQL-подобный (ClickHouse SQL)
<b>Модель хранения</b>	Гибридная (таблицы + chunking по времени)	Time-Structured Merge Tree (TSM)	Столбцовая, in-memory оптимизация	Столбцовая, сжатие данных и партиционирование
<b>Производительность записи</b>	Средняя, стабильная при умеренных нагрузках	Очень высокая при потоковых вставках	Очень высокая, минимальные задержки	Высокая, но зависит от настройки batch-вставок
<b>Производительность чтения</b>	Высокая, особенно при индексах по времени	Средняя при сложных запросах	Очень высокая (векторные вычисления)	Максимальная при больших объёмах данных
<b>Масштабируемость</b>	Поддержка multi-node, требуется настройка	Кластеризация в Enterprise и Cloud-версии	Пока отсутствует (односерверная модель)	Горизонтальное масштабирование и репликация
<b>Поддержка агрегаций</b>	Continuous aggregates, оконные функции	Flux-операции и агрегаты	SQL-агрегаты и временные функции	Мощные агрегаты и оконные выражения
<b>Интеграции</b>	Grafana, Power BI, Superset, BI-инструменты	Telegraf, Chronograf, Kapacitor, Grafana	Grafana, Prometheus, PostgreSQL API	Tableau, Superset, Airflow, Kafka
<b>Основная область применения</b>	Корпоративная аналитика и IoT-телеметрия	Мониторинг, IoT, наблюдаемость	Потоковая аналитика, финансы, кибербезопасность	Масштабная аналитика, логи, e-commerce
<b>Лицензия и открытость</b>	Open Source (Apache 2.0 + Enterprise)	Open Source (MIT / Enterprise Cloud)	Open Source (Apache 2.0)	Open Source (Apache 2.0 / Yandex)

Из представленных данных видно, что TimescaleDB занимает промежуточную позицию между традиционными реляционными системами и специализированными TSDB: она сочетает SQL-гибкость и удобство PostgreSQL с оптимизацией под временные данные. Её сильной стороной является совместимость с BI-инструментами и возможность агрегирования без внешних ETL-процессов.

InfluxDB сохраняет лидерство в скорости записи и простоте интеграции в системы мониторинга. Её стек TICK (Telegraf, InfluxDB, Chronograf, Kapacitor) делает систему готовым



решением для сбора и визуализации метрик. Однако при росте объёма данных и сложности запросов производительность может снижаться, что ограничивает применение в аналитических сценариях.

QuestDB демонстрирует выдающуюся производительность при потоковой обработке и минимальной задержке, что делает её оптимальным выбором для задач реального времени, особенно в финансовых и телекоммуникационных системах. Её архитектура пока не поддерживает распределённость, но потенциал для масштабирования велик.

ClickHouse остаётся безусловным лидером по скорости аналитических запросов и масштабируемости. Он способен обрабатывать петабайты данных, обеспечивая при этом высокую эффективность сжатия и параллельные вычисления. Его применение оправдано в проектах, где приоритетом является аналитика и агрегирование, а не непрерывная запись данных.

В целом таблица демонстрирует, что универсального решения для всех сценариев не существует. InfluxDB доминирует в области потоковой телеметрии, QuestDB – в низколатентных вычислениях, TimescaleDB – в интеграции с реляционными и аналитическими системами, а ClickHouse – в обработке больших объёмов и построении BI-аналитики.

Для наглядного сопоставления рассмотрим распределение областей применения этих четырёх систем на основе анализа открытых данных и практики их использования в индустрии приведенное на рисунке 1.

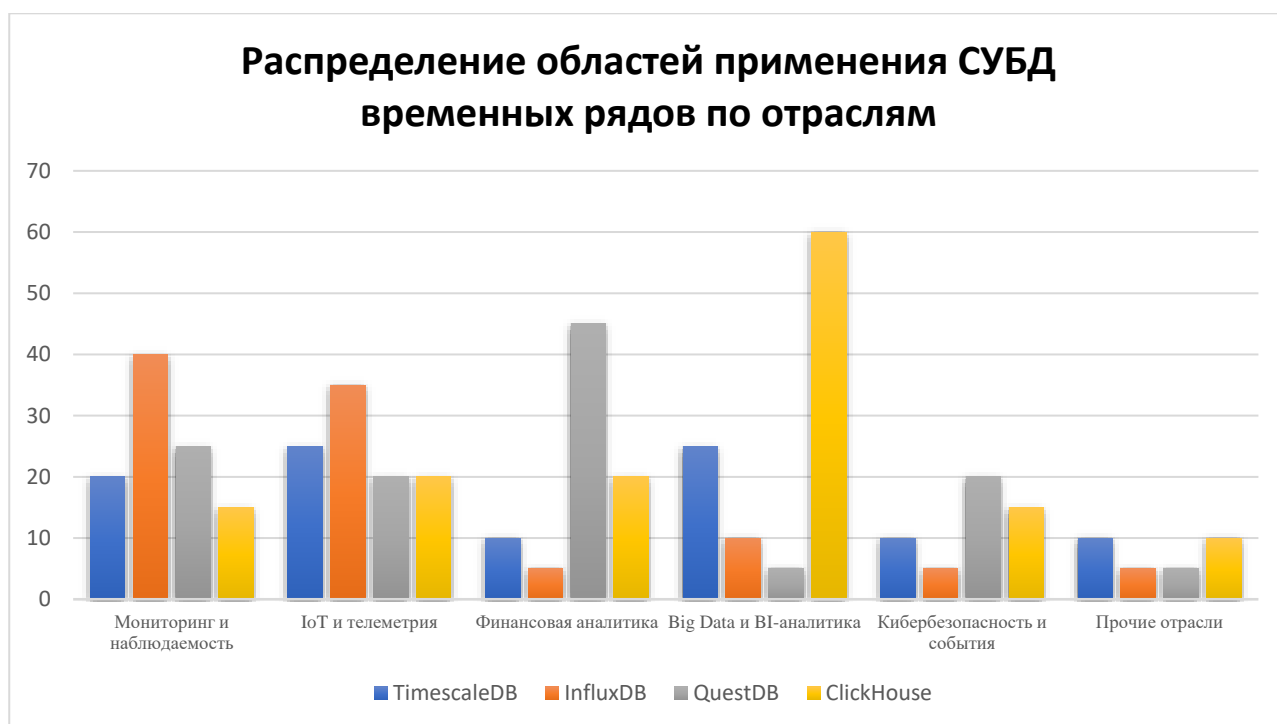


Рисунок 1 – Распределение областей применения СУБД временных рядов по отраслям (%)

Гистограмма, построенная по данным таблицы, показывает, что InfluxDB доминирует в области мониторинга и IoT, обеспечивая более 70% совокупной доли использования в этих сценариях. TimescaleDB сохраняет сильные позиции в корпоративных системах, где временные данные связаны с бизнес-процессами и аналитикой. QuestDB демонстрирует наибольший потенциал в финансовом секторе, где скорость реакции имеет решающее значение. ClickHouse, в свою очередь, уверенно лидирует в аналитических задачах Big Data, где объём данных измеряется терабайтами и требуется параллельная обработка запросов.

Перспективы развития этих систем тесно связаны с тенденцией к объединению потоковой аналитики и систем машинного обучения. Разработчики TimescaleDB акцентируют внимание на расширении функций распределённой обработки и интеграции с PostgreSQL 17+.

Команда InfluxDB активно развивает облачную версию и язык Flux, делая упор на масштабируемость и гибкость. QuestDB сосредоточена на добавлении кластеризации и повышении совместимости с системами Kafka и Prometheus. ClickHouse продолжает укреплять позиции в аналитическом сегменте, интегрируя нативную поддержку Python, Pandas и инструментов AI-анализаторов.

В совокупности это указывает на постепенное стирание границ между системами хранения временных рядов и полноценными аналитическими платформами. В ближайшие годы можно ожидать, что базы временных рядов станут не только инструментом мониторинга, но и ядром для интеллектуального анализа данных, прогнозирования и автоматизации принятия решений

#### **Список литературы:**

1. Воронин А. А. Базы данных и экспертные системы. Часть 1: методические указания для студентов по выполнению лабораторных работ по дисциплине. – Владимир: ВлГУ, 2023. – 183 с.
2. Комаров В. И. Путеводитель по базам данных: учебное пособие. – М.: ДМК Пресс, 2024. – 520 с.
3. Намиот Д. Е. Базы данных временных рядов в системах «Интернета вещей» // Прикладная информатика. – 2017. – Т. 12, № 2. – С. 79–87.
4. Нестеров С. А. Базы данных. Интеллектуальный анализ данных: учебное пособие. – СПб.: Изд-во Политехн. ун-та, 2011. – 272 с.
5. Нильсен Э. Практический анализ временных рядов. Прогнозирование со статистикой и машинное обучение / пер. с англ. – М.: Вильямс; Диалектика, 2021. – 544 с.
6. Орлов Г. М. Современные методы обработки и анализа данных: учебное пособие. – СПб.: Университет ИТМО, 2021. – 147 с.
7. Петров А. Распределенные данные. Алгоритмы работы современных систем хранения информации / пер. с англ. – СПб.: Питер, 2021. – 336 с.
8. Подкорытова О. А., Соколов М. В. Анализ временных рядов: учебное пособие для бакалавриата и магистратуры. – 2-е изд., перераб. и доп. – М.: Изд-во Юрайт, 2019. – 267 с.
9. Смирнов В. Программирование с Grafana и InfluxDB: сборник рецептов, 2020. – 267 с.
10. Ярушкина Н. Г., Афанасьева Т. В., Перфильева И. Г. Интеллектуальный анализ временных рядов: учебное пособие. – М.: ИД «ФОРУМ»; ИНФРА-М, 2022. – 160 с

