УДК 004.65 **DOI 10.52167/1609-1817-2024-131-2-350-358**

Ж.М. Алибиева[1], Н.К. Мукажанов[1], Л.Ш. Черикбаева[2],
А.С. Еримбетова[1], Д.А. Байымбетов[1]
[1]Satbayev University, Алматы, Казахстан
[2] Казахский национальных университет имени Аль-Фараби, Алматы, Казахстан
E-mail: aigerian@mail.ru

СРАВНЕНИЕ ВОЗМОЖНОСТЕЙ NOSQL КОЛОНОЧНОЙ БАЗЫ ДАННЫХ

Аннотация. На сегодняшний день объемы аналитических данных достигли критических масштабов, что ставит под сомнение традиционные методы их хранения, основанные на реляционных базах данных, которые не всегда могут эффективно справляться с такими объемами. Решения NoSQL открывают новые перспективы для обработки аналитических данных, особенно в контексте использования многомерных моделей. Исследования проводимые в данной статье посвящена сравнению возможностей применения NoSQL базы данных для аналитических систем с выбором СУБД ClickHouse. В работе представлен краткий сравнительный анализ преимуществ данной программы. В практической части рассмотрены методы создания многомерной модели данных на основе не реляционных баз данных. В статье приведен ClickHouse на примере создания OLAP с последующим сравнением возможностей с СУБД PostgreSQL. В заключении анализируются архитектура и компоненты OLAP системы.

Ключевые слова. Колоночные базы данных, хранение больших данных, обработка больших данных, аналитические системы, многомерные модели, аналитические данные.

Введение.

Во всех организациях на сегодняшний день хранение и обработка данных играют ключевую роль в их деятельности. Одним из важных аспектов повышения производительности информационных систем является скорость обработки данных [1]. Проблему этого аспекта решает использование NoSQL базы данных, способная обрабатывать аналитические запросы в реальном времени на больших объемах структурированных данных. Использование NoSQL, в частности, рассмотрение СУБД ClickHouse, способствует существенному улучшению производительности информационных систем организации и обеспечивает оперативную аналитику в режиме реального времени [2, 3]. ClickHouse постоянно улучшает скорость обработки запросов и объем занимаемого дискового пространства, что снижает нагрузку на систему. Это обеспечивает возможность оперативного получения запрошенных данных для дальнейшего анализа в короткие сроки и в реальном времени.

Материалы и методы.

Используя колоночные базы данных можно создать многомерные аналитические системы, оно направлено на обеспечение эффективного хранения и анализа больших объемов данных. В таких системах данные организуются с использованием колоночной модели, в отличие от традиционной строковой модели[4].

Многомерные аналитические системы строятся с внедрением дополнительных функции и инструментов на основе колоночной базы данных, специально разработанных для анализа больших данных [5]. Эта модель часто применяется в области бизнес-аналитики и обработки больших объемов данных, поскольку она обеспечивает удобный анализ данных по различным подходам и выявление закономерностей и трендов. Кроме

того, многомерные модели данных дают возможность проводить разнообразные методы аналитических исследований, как кумулятивный анализ, анализ по сегментам и регионам, анализ частоты и другие [6, 7].

В многомерной модели данных выделяются следующие основные концепции: гиперкубы, измерения, элементы, ячейки, меры или значения, рисунок 1.

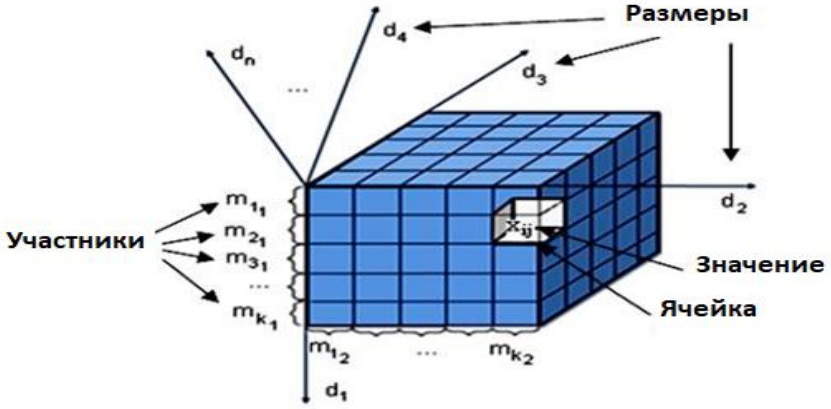

Рисунок 1 – Многомерный OLAPкуб

Формирование OLAP-куба в ClickHouse модели может базироваться на структурах данных, оптимизированных для использования аналитических операций, например, агрегация и фильтрация данных. Одной из самой часто используемой моделью формирования OLAP-куба в колоночных NoSQL базах данных является модель столбцов (columnar model). В этой модели данные хранятся не по строкам, а по столбцам, что обеспечивает эффективное хранение и обработку данных при выполнении агрегации и аналитических операций. Ниже приведен общий подход к формированию колоночных NoSQL базы данных в OLAP-кубах:

Измерения (Dimensions) – это атрибуты или характеристики данных, используемые OLAP-кубе для организации и агрегации данных. В каждом измерении имеются связанные с ними уровни детализации.

Факты (Facts) – это числовые значения, меняющиеся в OLAP-кубе, то есть они агрегируют и анализируются, и представляют количественные метрики.

Иерархии измерений (Dimension Hierarchies) – это измерения в OLAP-кубе, имеют иерархию, описывающую уровни детализации этого измерения.

Сначала на концептуальном уровне определяется многомерная модель, по следующей формуле $H\{F, D\}$, где:

$F = \{F1,..., Fk\}$ – конечное множество фактов,

$D = \{d1,..., dn\}$ – конечный набор изменений,

Гиперкуб при реализации, для измерения и описания его элементов использует «Теорию последовательности». Следовательно, согласно теории последовательности необходимо применить следующие значения:

D - сочетание размеров, если учесть что участники размерных комбинаций это d_i, то $d_i \in D$, это $D=\{d_1,d_2,d_3,...,d_n\}$. Выводы n – число размерности, $d_1,d_2,d_3,...,d_n$ – размещающиеся на оси гиперкуба участники комбинации измерений.

У каждого члена в размерной комбинации есть внутренние элементы, они размещены в соответствии координатным осям гиперкуба. Исходя из этого, в соответствии с измерениями гиперкуба реализуются следующие значения:

$M_d = \{m1_i, m2_i, ..., mk_i\}$ – внутренняя комбинация размеров или комбинация элементов измерения, $i = 1...n$ – комбинации значений в измерениях. Естественно каждое $d1 = \{m_{1_1}, m_{2_1}, ..., m_{k_1}\}$, $d_2 = \{m_{1_2}, m_{2_2}, ..., m_{k_2}\}$, $d_3 = \{m_{1_3}, m_{2_3}, ..., m_{k_3}\}, ..., d_n = \{m_{1_n}, m_{2_n}, ..., m_{k_n}\}$ измерение, это комбинации внутренних элементов. Отсюда, ki - значение внутренних элементов, каждого измерения. Если M – участник измерений куба, то через него можно показать связь всех внутренних членов измерений (∪): M∪Md1 ∪Md2 ∪ Md3 ∪ ...∪ Mdn. Значения каждого члена измерения: d1 – комбинация элементов Md1 измерения, значение члена измерения d2 – комбинация элементов Md2 измерения, в значении k2 = |Md2| членов, d3 - участник Md3 измерения, значение k3= | Md3| членов, ... , dn элемент Mdn измерения, значение $k_n = |M_{d_n}|$ членов [9].

Факты, $F=F^E$, определяются как (N^F, M^F), отсюда:

- N^F – имя факта

- $M^F=\{f1(m1),\ldots, fv(mv)\}$ представляет собой связанную с функцией агрегирования fi, набор мер.

Агрегированные значения соответствуют точкам пересечения элементов гиперкуба. Для формирования агрегационных значений гиперкуба при разработке структуры, учитываются различные метрики, такие как общая сумма агрегированных значений гиперкуба (sum), максимальное значение (max), среднее значение (avg), минимальное значение (min), отклонение (variation), дисперсия и другие.

Колоночные структуры данных создаются: для хранения и обработки соответствующих данных в отдельных колоночных структурах данных с отдельными измерениями, фактами и оптимизацией. С помощью колоночных структур данных можно эффективно сжимать, фильтровать данные, а также выполнять агрегацию по столбцам.

В базе данных ClickHouse, OLAP-куб формируется на основе данных, данные хранятся в таблицах. Таблица представляет собой многомерную структуру, она облегчает анализ данных и интерактивное исследование. Модель формирования базы данных ClickHouse для OLAP-куба состоит из нескольких этапов [10]:

1) Исходных данные: должны быть подготовлены заранее, чтобы их использовать для формирования OLAP-куба. Подготовка это - очистка, преобразование формата и другое.

2) Структура OLAP-куба: на основе анализа бизнес-процессов и требовании к данным определяются структуры OLAP-куба. Определение структуры это – поставленные факты, измерения, иерархии и другие параметры.

3) Таблицы для OLAP-куба: создаются после определения структуры чтобы хранить данные. Таблица создается командой CREATE TABLE, в таблице определяются поля и типы данных.

4) Данные в OLAP-кубе: данные берутся из исходной таблицы в базу данных ClickHouse командой INSERT INTO. Для введения данных используется синтаксис SELECT INTO, она позволяет вводить и выбирать только уже определенные поля из исходной таблицы.

5) Индексы и материализованные представления: для улучшения работы и быстродействия запросов OLAP-куба создаются материализованные представления и индексы. Индексы нужны для быстрого нахождения нужных данных в таблице, а материализованные представления – хранят результаты запросов, для ускорения их выполнения.

6) Запросы: OLAP-куб анализируется с помощью запросов, они включают в себя различные операции, например, группировка, фильтрация, агрегирование и другое.

Результаты.

Для оценки производительности ClickHouse были проведены различные тесты. Тестирование проводится для сравнения производительности ClickHouse с другими системами управления базы данных, например, Apache Impala, Apache Spark, Apache Hive по быстродействию обработки запросов, масштабируемости запросов и использования ресурсов базы данных [11]. Тестирование проводилась в зависимости от одного из конкретных случаев применения, размера используемых данных и от трудности запросов. Но почти во всех случаях ClickHouse показывает отличное быстродействие, чем другие системы, особенно когда обрабатываются больших объемы данных и аналитика в реальном времени. ClickHouse особенно эффективно обрабатывает запросы за секунды, в результате этого колоночной структуре хранения больших данных, ускоряет агрегацию и анализ использования больших объемов данных [12]. PostgreSQL это БД с открытым исходным кодом и с широким функционалом, является реляционной базой, поддерживающей разные типы данных, с обширным функционалом обработки транзакций, репликацией и масштабированием. Регулярно применяется в больших проектах для обработки и хранения объемных наборов структурированных данных. Основное отличие между ClickHouse и PostgreSQL – это обработка данных. PostgreSQL лучше использовать для транзакции данными, и он поддерживает ACID-транзакции и обеспечивает целостность. ClickHouse ориентирован на обработку аналитических запросов и поддерживает некоторый набор транзакций [13]. Другое не менее важное отличие это эффективность работы. ClickHouse благодаря своей структуре может обрабатывать большие запросы за секунды, за счет этого его можно использовать при выборе задач аналитики. У PostgreSQL те же общие назначения и он может обрабатывать транзакционные и аналитические запросы, но он является менее эффективным при работе с большими объемами данных. У него богатый функционал и он выполняет сложные операции, но эффективность ниже по сравнению с ClickHouse для обработки аналитических запросов [14].

Обсуждение.

В данном исследовании мы сравнили нагрузочное тестирование ClickHouse и PostgreSQL. У обеих систем цели различаются и специфика использования тоже различны, все это мы описали выше. Выбор PostgreSQL или же ClickHouse зависит от конкретного требования проекта. Для этого нужно просто знать для хранение и обработка каких структурированных данных нужно их применять, знать для каких транзакции их использовать, в нашем случае хорошим выбором будет PostgreSQL. В то время как для работы с большими данными, значимыми требованиями аналитической обработки, ClickHouse очень хорошо подходить. В PostgreSQL создание и изменение таблицы, индексов, схемы базы данных и других данных осуществляется через DDL (Data Definition Language). DDL определяет типы данных, структуру данных, ограничения данных и другие данные. Основное отличие заключается в том, что в PostgreSQL хранение данных осуществляется в отдельных таблицах, и для выполнения представлений и сложных запросов требуется собирать данные с использованием VIEW с помощью команды UNION ALL. Это существенно влияет на производительность [15].

В рамках данного исследования требуется провести тестирование скорости для определения числа звонков и минуты разговора в период с 01.03.2023 по 15.03.2023 по исходящим вызовам коллекторского направления с использованием префиксов 9000, 9001, 9005, 9006, 9007, 8765, 9012, 9013 запросов.

В результате выполнения запроса в ClickHouse было получено 7 921 307 звонков и 3 574 826 минут разговора за указанный период. Запрос был обработан за 25 секунд (25709 миллисекунд).

В PostgreSQL было получено 9 372 663 звонков и 3 659 855 минут разговора за тот же период. Однако запрос в PostgreSQL занял значительно больше времени, равно 292 секундам (292586 миллисекундам).

Учитывая, что на обе тестируемое СУБД тест проводился в момент минимальной нагрузки коллекторной службы, результаты показывают, что время выполнения запроса в ClickHouse было быстрее в 10 раз, в сравнений с PostgreSQL. Также отмечается разница в данных по времени разговоров и количеству вызовов между ClickHouse и PostgreSQL. Это обусловлено тем, что в PostgreSQL не учитываются дубликаты, которые могут снизить точность биллинга. Удаление дубликатов в PostgreSQL представляет собой сложную задачу, так как их появление трудно предсказать. В ClickHouse снижение дубликатов достигается с помощью расширенного функционала SQL, такого как команды FINAL и PREWHERE, которые оптимизируют время выполнения запроса [16].

Следующим этапом исследования будет сравнение объема данных в тестируемых СУБД ClickHouse и PostgreSQL. Например, данные по устройству GW2 за период с 01.03.2023 по 15.03.2023 будут использованы для анализа.

В процессе тестирования было выполнено перемещение данных из таблицы GW2 за период с 01.03.2023 по 15.03.2023 в другую таблицу gw2222 и определено число записей. В PostgreSQL таблица gw2222 содержит 18 443 139 записей.

Затем был выполнен запрос для определения размера таблицы gw2222 в схеме old cdr. Результатом запроса стал объем таблицы gw2222, равный 10 GB.

Аналогично, в ClickHouse данные были перемещены из общей таблицы "cube" в другую таблицу GW2 для GW2 устройства за период с 01.03.2023 по 15.03.2023. В ClickHouse таблица GW2 содержит 18 5242 945 записей.

После этого был выполнен запрос для определения размера gw2222 таблицы в ClickHouse, который показал, что таблица GW2 занимает 1.4 GB.

Результаты теста показали, что объем данных в ClickHouse меньше восемь раз весят чем в СУБД PostgreSQL. Это объясняется по следующим причинам:

1) Колоночное хранение данных: В ClickHouse используется колоночное хранение данных, это позволяет эффективно хранить и сжимать используемую информацию. В отличие от реляционных СУБД, тут все данные сохраняются по строкам, ClickHouse преобразует данные по столбцам. Оно помогает эффективно использовать алгоритмы сжатия, потому что повторяющиеся значения обычно содержится в столбцах, и оно позволяет более компактно сохранить данные.

2) Компрессия данных: В ClickHouse встроенная поддержка сжатия данных. ClickHouse при записи данных, автоматически использует сжатие для каждого столбца, такие как ZSTD или LZ4. Это позволяет значительно на диске снизит размер данных и уменьшит требования к данным.

3) Оптимизация структуры индексов: В ClickHouse используется специфические конструкции индексов, например, Bitmap Index, Bloom Filter более эффективным образом фильтруют и ищут данные. Индексы по сравнению с другими СУБД занимают меньше места.

4) Лишные данные и метаданные: В ClickHouse специализируется на сохранении и обработке больших данных, и они могут удалить и не сохранять ненужные данные и общую системную информацию. Это тоже приводить к уменьшению общего объема данных.

В итоге этих улучшений ClickHouse обеспечивает более сжатое сохранение данных, и это позволяет сэкономить место и улучшает производительность при работе с аналитическими запросами. Кроме того, благодаря этим возможно значительно повысить период времени, в течение которого данные хранятся. Например, при необходимости в СУБД PostgreSQL для хранения данных нужен будет хранилище размером 2 ТБ за

полгода, то в ClickHouse можно повысить срок до двух лет, даже с увеличением количества звонков.

На заключительном этапе создается многомерный гиперкуб. Для таблицы общих данных по звонкам выбираются нужные столбцы с временными маркерами, отображающие время звонков в секундах и минутах, с номерами абонентов, с временем начала и окончания соединения, а также датой. Создается диаграмма, показывающая количество звонков в течение дня, где на вертикали отображаются время звонков, а по горизонтали сам звонок. Кроме того, есть возможность добавления временных фильтров и отображения итоговых данных по количеству звонков и продолжительности в минутах

В результате получается представление, которое позволяет проводить расчеты биллинга в реальном времени для исходящих вызовов на сотовом операторе Beeline, рисунок 2.

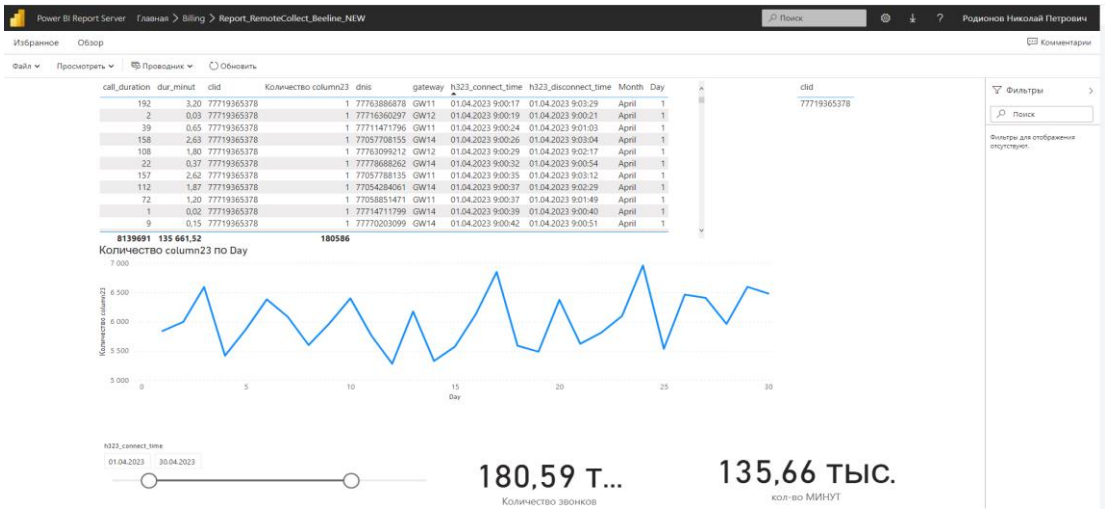

Рисунок 2 – Расчет биллинга по мобильному оператору Beeline

Заключение.

В итоге проведенного исследования было выявлено, что при обработке больших объемов данных и выполнении аналитических запросов колоночная СУБД ClickHouse обладает значительными преимуществами. ClickHouse обеспечивает высокое быстродействие и эффективность при агрегировании и анализе, что делает его привлекательным выбором при использовании колоночных структур данных. ClickHouse способен эффективно применить эффективные параметры для сохранения и обеспечивает скоростной доступ к информации. Выбор между этими двумя системами ClickHouse и PostgreSQL зависит от конкретного требования проекта. Результаты и выводы по исследованиям:

1) *Возможности колоночной БД:* в итоге установлено, что ClickHouse представляет собой мощную систему для аналитической обработки данных.

2) *Многомерные модели данных:* в данной статье предлагается разработанная многомерная модель данных. Примером для исследования является оператор мобильной связи Beeline. При работе с такой моделью данных ClickHouse обеспечивает эффективную работу, которая дает возможность выполнять сложные запросы и проводить анализ.

3) *Повышение скорости запросов:* Тестирование в ходе исследования показало, что скорость ClickHouse быстрее в 10 раз выполняет запросы, чем PostgreSQL.

4) *Объем данных:* Тест показало, что в ClickHouse объем данных меньше в 8 раз, чем в PostgreSQL. Это дает возможность увеличить срок хранения данных в 4 раза.

ЛИТЕРАТУРА

[1] Ramesh Sharda, Dursun Delen, Efraim Turban. Business Intelligence: A Managerial Perspective on Analytics (3rd Edition). Publisher: Pearson, ISBN-13: 9780133051056, 416 pages, 2019.

[2] Tadviser.ru. Большие данные (Big Data) (2017), Available at: https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%91%D0%BE%D0%BB%D1%8C%D1%88%D0%B8%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5_(Big_Data)

[3] Dan Sullivan. NoSQL for Mere Mortals. Publisher: Addison -Wesley Professional, ISBN-13: 9780134023212, 542 pages, 2015.

[4] Sadalage P. J., Fowler M. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Publisher: Addison -Wesley Professional. ISBN-13: 9780321826626, 192 pages, 2012.

[5]Provost F., Fawcett T. Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking. Publisher: O'Reilly Media, ISBN-13: 9781449361327, 413 pages, 2013.

[6] Chatfield C. The Analysis of Time Series: An Introduction (Sixth Edition). Publisher: CRC Press. ISBN-13: 9780203491683, 352 pages, 2016.

[7] Hastie T., Tibshirani R., & Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd edition), Publisher: Springer, ISBN-13: 978-0387848570, 767 pages, 2016.

[8] Codd, E. F. Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate. Technical Report, E. F. Publisher: Codd & Associates, 1993.

[9] R.K.Uskenbayeva, Y.I.Cho, G.B.Bektemyssova, N.K.Mukazhanov, D.K. Kozhamzharova, B.K. Kurmangaliyeva. Multidimensional indexing structure development for the optimal formation of aggregated indicators in OLAP hypercube, 14th International Conference on Control, Automation and Systems (ICCAS 2014) Oct. 22-25, 2014 in KINTEX, Gyeonggi-do, Korea.

[10] Chodorow K., Dirolf M. MongoDB: The Definitive Guide. Publisher: O'Reilly Media, ISBN-13: 9781449381561, 216 pages, 2013.

[11] Radcliffe D. NoSQL for Dummies. Publisher: For Dummies, ISBN-13: 9788126554904, 464 pages, 2015.

[12] Joshi H. Redis Cookbook: Practical Techniques for Fast Data Manipulatio, Publisher:O'Reilly Media, ISBN-13: 978-1449305048, 71 pages, 2011.

[13] Boncz P., et al. advances in Graph Database Research (2020). Available at: https://www.researchgate.net/profile/Peter-Boncz.

[14] Zaharia M. Learning Spark: Lightning-Fast Big Data Analysis. Publisher: Oreilly & Associates Inc, ISBN-13: 978-1449358624, 254 pages, 2015.

[15] Yandex. ClickHouse Documentation. Available at: https://cloud.yandex.com/en/docs/managed-clickhouse/qa/clickhouse.

[16] Farouk R., El-Sayed, H., El-Kassas S., & El-Den D. Building OLAP cubes: challenges and approaches. Journal of Database Management, 2020. 31(3), 38-54.

REFERENCES*

[1] Ramesh Sharda, Dursun Delen, Efraim Turban. Business Intelligence: A Managerial Perspective on Analytics (3rd Edition). Publisher: Pearson, ISBN-13: 9780133051056, 416 pages, 2019.

[2] Tadviser.ru. Большие данные (Big Data) (2017), Available at: https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%91%D0%BE%D0%BB%D1%8C%D1%88%D0%B8%D0%B5_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5_(Big_Data)

[3] Dan Sullivan. NoSQL for Mere Mortals. Publisher: Addison -Wesley Professional, ISBN-13: 9780134023212, 542 pages, 2015.

[4] Sadalage P. J., Fowler M. NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence, Publisher: Addison -Wesley Professional. ISBN-13: 9780321826626, 192 pages, 2012.

[5]Provost F., Fawcett T. Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking. Publisher: O'Reilly Media, ISBN-13: 9781449361327, 413 pages, 2013.

[6] Chatfield C. The Analysis of Time Series: An Introduction (Sixth Edition). Publisher: CRC Press. ISBN-13: 9780203491683, 352 pages, 2016.

[7] Hastie T., Tibshirani R., & Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd edition), Publisher: Springer, ISBN-13: 978-0387848570, 767 pages, 2016.

[8] Codd, E. F. Providing OLAP (On-line Analytical Processing) to User-Analysts: An IT Mandate. Technical Report, E. F. Publisher: Codd & Associates, 1993.

[9] R.K.Uskenbayeva, Y.I.Cho, G.B.Bektemyssova, N.K.Mukazhanov, D.K. Kozhamzharova, B.K. Kurmangaliyeva. Multidimensional indexing structure development for the optimal formation of aggregated indicators in OLAP hypercube, 14th International Conference on Control, Automation and Systems (ICCAS 2014) Oct. 22-25, 2014 in KINTEX, Gyeonggi-do, Korea.

[10] Chodorow K., Dirolf M. MongoDB: The Definitive Guide. Publisher: O'Reilly Media, ISBN-13: 9781449381561, 216 pages, 2013.

[11] Radcliffe D. NoSQL for Dummies. Publisher: For Dummies, ISBN-13: 9788126554904, 464 pages, 2015.

[12] Joshi H. Redis Cookbook: Practical Techniques for Fast Data Manipulatio, Publisher:O'Reilly Media, ISBN-13: 978-1449305048, 71 pages, 2011.

[13] Boncz P., et al. advances in Graph Database Research (2020). Available at: https://www.researchgate.net/profile/Peter-Boncz.

[14] Zaharia M. Learning Spark: Lightning-Fast Big Data Analysis. Publisher: Oreilly & Associates Inc, ISBN-13: 978-1449358624, 254 pages, 2015.

[15] Yandex. ClickHouse Documentation. Available at: https://cloud.yandex.com/en/docs/managed-clickhouse/qa/clickhouse.

[16] Farouk R., El-Sayed, H., El-Kassas S., & El-Den D. Building OLAP cubes: challenges and approaches. Journal of Database Management, 2020. 31(3), 38-54.

Жибек Алибиева, PhD, Satbayev University, Алматы, Қазақстан, alibievajibek@gmail.com

Нуржан Мукажанов, PhD, Satbayev University, Алматы, Қазақстан, mukazhan@mail.ru

Ляйля Черикбаева, PhD, әл-Фараби атындағы Қазақ ұлттық университеті, Алматы, Қазақстан, Cherikbayeva.Lyailya@gmail.com

Айгерим Еримбетова, PhD, т.ғ.к., Satbayev University, Алматы, Қазақстан, aigerian@mail.ru

Даулет Баймбетов, магистр, Satbayev University, Алматы, Қазақстан, daucloudlab@gmail.com

NOSQL БАҒАНДЫҚ ДЕРЕКТЕР ҚОРЫНЫҢ МҮМКІНДІКТЕРІН САЛЫСТЫРУ

Аңдатпа. Қазіргі кезде аналитикалық деректер көлемі критикалық өлшемдерге жете бастады, сол себепті деректерді дәстүрлі сақтау тәсілдерінен ауытқуымыз керек, яғни қолдануды жеңілдету мақсатында, деректердің реляциялық базасы негізіндегі ағымдық шешімдерді алу мүмкіндіктері керек болады, олар көп көлемдегі деректермен жұмыс жасайды. NoSQL осы деректерді өңдеудің дәстүрлі әдісінен ауытқу мүмкіндігін бере алады, көпөлшемді деректерді қарастыруға жақсы келеді. Жұмыстың зерттеу бөлімі аналитикалық жүйелер үшін NoSQL қолданбасының нұсқаларын зерттеуге арналған. ДҚБЖ ретінде ClickHouse таңдалды. Бағдарламаның пайдасын қысқаша салыстырмалы талдау зерттеу болып табылады. Практикалық бөлімде реляциялық емес мәліметтер қорынан көпөлшемді деректер моделін құру тәсілдерін қолданудың шолулары келтірілген. Атап айтқанда, ClickHouse жүйесінде OLAP құру және PostgreSQL ДҚБЖ салыстыру. Қорытындыда ClickHouse ДҚБЖ көмегімен OLAP жүйесінің жұмыс процесі мен құрамдас бөліктері талданды.

Түйінді сөздер. Бағандық деректер қоры, мәліметтерді сақтау, мәліметтерді өңдеу, аналитикалық жүйелер, көпөлшемді моделдер, аналитикалық деректер.

Zhibek Alibiyeva, PhD, Satbayev University, Almaty, Kazakhstan, alibievajibek@gmail.com
Nurzhan Mukazhanov, PhD, Satbayev University, Almaty, Kazakhstan, mukazhan@mail.ru
Lyailya Cherikbayeva, PhD, al-Farabi Kazakh National University, Almaty, Kazakhstan, Cherikbayeva.Lyailya@gmail.com
Aigerim Yerimbetova, PhD, candidate of technical sciences, Satbayev University, Almaty, Kazakhstan, aigerian@mail.ru
Daulet Baiymbetov, master, Satbayev University, Almaty, Kazakhstan, daucloudlab@gmail.com

COMPARISON OF THE CAPABILITIES OF NOSQL COLUMNAR DATABASE

Abstract. Today, the volume of analytical data has reached critical proportions, which calls into question traditional methods of storing it, based on relational databases, which cannot always effectively cope with such volumes. NoSQL solutions open up new perspectives for processing analytical data, especially in the context of using multidimensional models. The research part of this work is devoted to studying the possibilities of using NoSQL for analytical systems with the choice of the ClickHouse DBMS. The paper provides a concise comparative analysis of the advantages of this program. The practical section explores methods for constructing a multidimensional data model using non-relational databases. Specifically, it includes an example of OLAP creation in ClickHouse, followed by a comparison with PostgreSQL DBMS. The concluding part delves into the architecture and components of the operational workflow of the OLAP system, illustrated by the utilization of ClickHouse DBMS..

Keywords. columnar databases, data storage, data processing, analytical systems, multidimensional models, analytical data.

****