

## ФУНКЦІОНАЛЬНІ МОЖЛИВОСТІ СИСТЕМ КЕРУВАННЯ БАЗАМИ ДАНИХ ЧАСОВИХ РЯДІВ

Ветлужських М.В., Шинкаренко В. І.

*Український державний університет науки і технологій, Україна*

**Анотація.** Представлена робота надає всебічний огляд можливостей баз даних часових рядів, з акцентом на InfluxDB – провідну базу даних часових рядів, відому своєю ефективністю та підтримкою складних аналізів за допомогою мови запитів Flux, включно з агрегацією даних, фільтрацією, виявленням трендів та прогнозуванням. Практичне застосування InfluxDB демонструється через різні приклади, показуючи її універсальність у виконанні складних операцій з даними, таких як фільтрація даних, виявлення значних змін у даних та виконання стохастичного аналізу. Ці функціональності висвітлюють потенціал InfluxDB у широкому спектрі застосувань, від телеметрії IoT до фінансового аналізу, підкреслюючи критичну роль баз даних часових рядів у сучасних стратегіях управління та аналізу даних.

**Ключові слова:** часовий ряд, бази даних, аналіз часових рядів

**Вступ.** Наряду з відомими та широко розповсюдженими системами керування реляційними базами даних (реляційними СКБД), мають застосування і інші підходи збереження даних які вирізняються способом їх збереження і обробки та спеціалізацією.

Реляційні бази даних та бази даних часових рядів (Time series database) відрізняються по кільком основним аспектам. Реляційні бази даних орієнтовані на зберігання інтерреляційних даних у таблицях з рядками та стовпцями. Вони підтримують складні SQL-запити для обробки даних і ідеально підходять для додатків, які потребують складних транзакцій, таких як банківські системи або системи управління ресурсами підприємства. З іншого боку, бази даних часових рядів спеціалізуються на зберіганні і аналізі даних, які змінюються з часом, з використанням часових міток для ефективного виконання запитів за часовими інтервалами. Вони оптимальні для моніторингу, відстеження та аналізу даних в реальному часі, наприклад, для фінансових котирувань, телеметрії IoT або метеорологічних даних.

**Основна частина.** Розглянемо функціональні можливості СКБД часових рядів InfluxDB, яку було обрано за рейтингом [1], де ця СКБД займає перше

місце. З'ясуємо як застосувати мову запитів Flux для аналізу, включно з агрегацією, фільтрацією, виявленням трендів і прогнозуванням. Можливості InfluxDB продемонструємо на прикладах для кращого розуміння її потенціалу.

Запити в InfluxDB можна робити за допомогою мов Flux або InfluxQL. InfluxQL являє собою мову схожу на SQL, а Flux – це альтернативна мова запитів, що використовує функціональну парадигму. Будемо розглядати лише приклади на мові запитів Flux. Детальніше щодо різниці функціональності запитів представлено у [2].

Для проведення дослідження потрібні дані. InfluxDB надає можливість загрузки історичних даних за допомогою Line Protocol або CSV, також є можливість наповнення даними у реальному часі за допомогою бібліотек для багатьох мов програмування та спеціального агента Telegraf [3], який має багато готових плагінів для розповсюджених задач.

InfluxDB має багато засобів попередньої обробки даних. До них входять агрегація, фільтрація та інші.

Агрегація даних важлива для зменшення кількості даних, що аналізуються, та отримання суттєвих метрик з великих наборів даних. У InfluxDB присутня підтримка агрегації даних на ділянках за допомогою функції `aggregateWindow` та багатьох спеціальних функцій таких як `min`, `max`, `mean`, `stddev` та інших. Також є можливість використання індивідуально розроблених функцій. Приклад використання функції `aggregateWindow` :

```
from(bucket: "OverviewBucket")
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
|> filter(fn: (r) => r["_measurement"] == "electric_production")
|> filter(fn: (r) => r["_field"] == "production")
|> aggregateWindow(every: 1y, fn: sum, createEmpty: false)
|> yield()
```

Цей запит Flux до InfluxDB вибирає дані про виробництво електроенергії з кошика "OverviewBucket" за заданий часовий діапазон, фільтрує їх за конкретними вимірами та полями, що стосуються виробництва електроенергії, агрегує сумарні значення продукції по роках без створення пустих агрегатів, і повертає ці агреговані результати.

Фільтрація даних є ключовою операцією для вибірки специфічних даних з бази даних часових рядів. Це дозволяє точно вказати, які дані потрібно аналізувати, зменшуючи обсяг даних, які потребують обробки, та підвищуючи ефективність запитів. Фільтрація може бути застосована до вимірювань, тегів, полів, значень, а також до часових діапазонів. У прикладі нижче можна побачити фільтрацію даних по вівторкам:

```
import "date"
from(bucket: "OverviewBucket")
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
|> filter(fn: (r) => r["_measurement"] == "electric_production")
|> filter(fn: (r) => r["_field"] == "production")
|> filter(fn: (r) => date.weekDay(t: r._time) == 2)
```

Цей запит імпортує бібліотеку "date" для роботи з датами, він вибирає дані про виробництво електроенергії з кошика "OverviewBucket" за заданий часовий діапазон, використовуючи фільтрацію для обмеження вибірки лише записами, що стосуються виробництва електроенергії. Останній фільтр додатково обмежує дані до тих записів, що були зроблені у вівторок (0 – неділя, 6 – субота).

Для візуального аналізу даних InfluxDB призначений вбудований веб-інтерфейс, який дозволяє користувачам створювати налаштовані панелі керування з різними типами віджетів для візуалізації даних, включаючи графіки, гістограми, карти тепла та багато іншого.

Одним із найпоширеніших методів аналізу часових рядів є пошук сплесків. Для того, щоб реалізувати його засобами InfluxDB, потрібно використати функцію difference, яка знаходить різницю до попереднього запису.

Приклад пошуку сплесків:

```
import "math"
from(bucket: "OverviewBucket")
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)
|> filter(fn: (r) => r["_measurement"] == "electric_production")
|> filter(fn: (r) => r["_field"] == "production")
// Використання віконного агрегування для поділу даних на 1-годинні
інтервали
|> aggregateWindow(every: 1h, fn: mean)
// Додавання нового поля, що показує зміну середнього значення між поточним
і попереднім значеннями
```

```
|> difference(nonNegative: false, columns: ["_value"])  
// Фільтрація результатів для виявлення значних змін  
|> filter(fn: (r) => math.abs(x: r["_value"]) > 11)
```

Цей запит Flux, починаючи з імпорту бібліотеки "math", вибирає дані про виробництво електроенергії з кошика "OverviewBucket" за вказаний часовий діапазон. Фільтрує записи за вимірами та полями, що стосуються виробництва електроенергії, і використовує віконне агрегування для поділу даних на 1-годинні інтервали, розраховуючи середнє значення в кожному з них. Після цього застосовується функція difference для створення нового поля, що показує зміну середнього значення між поточним і попереднім значеннями. Завершується запит фільтрацією результатів, щоб виявити значні зміни, тобто ті, що перевищують абсолютне значення 11, використовуючи функцію math.abs.

Ступінь стохастичності в контексті аналізу даних часто пов'язаний з оцінкою рівня випадковості або непередбачуваності в часових рядах. Реалізацію стохастичного аналізу [4] за допомогою InfluxDB наведено далі:

```
import "math"  
// Вибірка даних за вибраний період  
data = from(bucket: "OverviewBucket")  
|> range(start: v.timeRangeStart, stop: v.timeRangeStop)  
|> filter(fn: (r) => r["_measurement"] == "electric_production")  
|> filter(fn: (r) => r["_field"] == "production")  
// Обчислення мінімального та максимального значення за період  
minValue = (data |> min() |> findRecord(idx: 0, fn: (key) => true))._value  
maxValue = (data |> max() |> findRecord(idx: 0, fn: (key) => true))._value  
// Обчислення стохастичного осцилятора  
stochasticOscillator = data  
|> map(fn: (r) => ({  
    _time: r._time,  
    _value: ((r["_value"] - minValue) / (maxValue - minValue)) * 100.0  
}))  
// Виведення результатів  
stochasticOscillator |> yield()
```

В подальшому планується порівняння прогнозів часових рядів за розробленою конструктивно-продукційних моделлю [5] та за методами реалізованими у InfluxDB на наявних банках даних.

**Висновок.** Завдяки своїй високій продуктивності, гнучкості та підтримці мови запитів Flux, InfluxDB надве широкі можливості для агрегації даних, фільтрації, виявлення трендів і прогнозування. Через демонстрацію різноманітних прикладів використання показано, як InfluxDB може бути застосована до різних сценаріїв, від телеметрії IoT до фінансового аналізу, підкреслюючи її значення як інструменту для вирішення складних завдань аналізу даних.

### ЛІТЕРАТУРА / REFERENCE

1. Time Series Database Management Systems Ranking. URL: <https://db-engines.com/en/ranking/time+series+dbms> (Accessed: 21th March 2024).
2. InfluxQL and Flux Parity – InfluxData Documentation. URL: <https://docs.influxdata.com/influxdb/cloud/reference/syntax/flux/flux-vs-influxql/#influxql-and-flux-parity> (Accessed: 22nd March 2024).
3. Telegraf: Time Series Data Collection – InfluxData. URL: <https://www.influxdata.com/time-series-platform/telegraf/> (Accessed: 1st March 2024);
4. Stochastic Oscillator – Investopedia. URL: <https://www.investopedia.com/terms/s/stochasticoscillator.asp> (Accessed: 2nd March 2024).
5. Shynkarenko V., Zhadan A. Modeling of the Deterministic Fractal Time Series by One Rule Constructors, 2020 IEEE 15th International Conference on Computer Sciences and Information Technologies (CSIT), Zbarazh, Ukraine, 2020, pp. 336-339, doi: 10.1109/CSIT49958.2020.9321923.

## FUNCTIONAL CAPABILITIES OF TIME SERIES

### DATABASE MANAGEMENT SYSTEMS

Michael Vietluzhskykh, Viktor Shynkarenko

**Abstract.** *This work provides a comprehensive overview of the capabilities of time series databases, with a special emphasis on InfluxDB – a leading time series database known for its efficiency and support for complex analyses through the Flux query language, including data aggregation, filtering, trend detection, and forecasting. The practical application of InfluxDB is demonstrated through various examples, showcasing its versatility in performing complex data operations, such as data filtering, identifying significant changes in data, and conducting stochastic analysis. These functionalities highlight the potential of InfluxDB across a broad range of applications, from IoT telemetry to financial analysis, underscoring the critical role of time series databases in modern data management and analysis strategies.*

**Keywords:** *fractals, graph, constructive-synthesizing modeling, computer modeling, iterative algorithms, pattern formation*