

Лабораторна робота № 6. Аналітика даних

МЕТА РОБОТИ: отримати навички аналізу даних за допомогою SQL

ПОСТАНОВКА ЗАДАЧІ

За допомогою застосунку, розробленого в попередній лабораторній роботі, заповнити базу даних даними з датчиків мобільного телефону. Виконати аналітичні запити до збережених даних.

ПОРЯДОК ВИКОНАННЯ

1. Здійснити запис даних з датчиків мобільного телефону в достатньому для здійснення аналітики обсязі. Кількість даних можна регулювати налаштовуючи належним чином частоту відправки даних з датчиків.

2. Використовуючи **безперервні агрегати**, оптимізувати запити до даних акселерометра та створити графік, що показує **середнє значення прискорення** по одній з осей за кожну хвилину.

Рекомендації! Приклад створення безперервних агрегатів можна знайти в лекціях. В залежності від проміжку часу, протягом якого здійснювався запис даних, часовий інтервал для обчислення середнього значення прискорення може бути замінений на менший, наприклад, 5 чи 10 секунд.

3. Обчислити **середні значення прискорення** по тій самій осі, що й у попередньому пункті, за допомогою відповідної **віконної функції**.

4. **Згладжування даних (Ковзне середнє):** Обчислити ковзне середнє значення прискорення по тій самій осі, що й у попередньому пункті, за допомогою віконних функцій та фрейму (ковзного вікна).

5. **Згладжування даних за часовим вікном:** Обчислити ковзне середнє прискорення, але не за кількістю записів, а за фіксованим проміжком часу, наприклад, за останні 10 секунд. Це більш точно відображає реальні дані, де інтервал між вимірами може бути нерівномірним.

6. **Виявлення моментів пікової активності:** Знайти 5 найвищих показників прискорення за весь період для кожного пристрою.

Приміка! Врахувати, що абсолютне значення прискорення може бути обчислене як квадратний корінь із суми квадратів всіх трьох компонент

вектора прискорення: $a = \sqrt{a_x^2 + a_y^2 + a_z^2}$.

7. Виявлення аномалій (Anomaly Detection): На основі даних акселерометра виявити аномальні події, наприклад, падіння телефону чи сильний удар по телефону.

Рекомендації!

- 1) Щоб імітувати падіння телефону достатньо швидко взмахнути рукою, що тримає телефон, або підкинути телефон (і бажано зловити, що уникнути можливих небажаних наслідків як для телефона так і для його власника 😊).
- 2) Для виявлення аномалій рекомендується використовувати віконні функції AVG() та STDDEV(). Для цього достатньо знайти точки, де показники виходять за межі 2-3 стандартних відхилень, що свідчитиме про аномальну подію.

8. Обчислити ковзну швидкість, використовуючи дані прискорення.

Пояснення:

- 1) Врахуємо, що швидкість — це інтеграл від прискорення по часу:
$$v_x = \int a_x dt.$$

- 2) Оскільки дані з датчиків акселерометру дискретні (вимірюються в окремі моменти часу), інтегрування перетворюється на підсумовування:
$$v_x = \int_{t_0}^{t_n} a_x dt \approx \sum_{i=0}^n \tilde{a}_x(t_{i+1} - t_i),$$
 де $\tilde{a}_x$ — середнє значення прискорення на проміжку $[t_i, t_{i+1}]$. Оскільки швидкість представлена трьома компонентами вектора, то модуль вектора швидкості може бути обчислений як квадратний корінь із суми квадратів всіх трьох компонент вектора:
$$v = \sqrt{v_x^2 + v_y^2 + v_z^2}.$$

- 3) Приклад SQL-запиту для обчислення швидкості:

```
WITH time_deltas AS (  
  -- 1. Обчислюємо різницю в часі між послідовними записами  
  SELECT time, x_accel, y_accel, z_accel,  
         (EXTRACT(EPOCH FROM time) - EXTRACT(EPOCH FROM LAG(time, 1) OVER  
         (ORDER BY time))) AS time_delta  
  FROM accel_data  
  WHERE $__timeFilter(time)  
,  
velocities_components AS (  
  -- 2. Обчислюємо швидкість по кожній осі  
  SELECT time, time_delta,  
         SUM(x_accel * time_delta) OVER (ORDER BY time) AS velocity_x,  
         SUM(y_accel * time_delta) OVER (ORDER BY time) AS velocity_y,  
         SUM(z_accel * time_delta) OVER (ORDER BY time) AS velocity_z  
  FROM time_deltas  
  WHERE time_delta IS NOT NULL  
)  
-- 3. Обчислюємо модуль (загальну величину) вектора швидкості
```

```

SELECT time,
  -- Модуль швидкості: sqrt(vx^2 + vy^2 + vz^2)
  SQRT(POW(velocity_x, 2)+POW(velocity_y, 2)+POW(velocity_z, 2) ) AS total_velocity
FROM velocities_components
ORDER BY time;

```

LAG(): Ця віконна функція дозволяє отримати значення time з попереднього рядка, що необхідно для обчислення time_delta.

SUM() OVER (ORDER BY time): Це кумулятивна сума. Вона підсумовує значення (x_accel * time_delta) для кожного рядка та всіх попередніх рядків. Фізично це і є інтегрування, тобто розрахунок швидкості.

7*. Обчислити **ковзну відстань**, використовуючи дані прискорення.

Примітка! При виконанні цього завдання варто врахувати, що пройдена відстань — це інтеграл швидкості по часу: $s_x = \int v_x dt$. А абсолютне значення пройденої відстані на кожному відрізку шляху може бути обчислене як квадратний корінь з суми квадратів компонент вектора $\vec{s}$: $s =$

$$\sqrt{s_x^2 + s_y^2 + s_z^2}.$$

9. Побудувати графіки в Grafana для всіх розрахованих величин.

10. Оформити звіт, до якого додати скріншоти, що демонструють виконання всіх етапів завдання. До звіту включити також відповіді на контрольні запитання.

КОНТРОЛЬНІ ЗАПИТАННЯ

1. Що таке неперервні агрегати в контексті TimescaleDB? В чому їх перевага над віконними функціями?
2. Що таке фрейми в контексті віконних функцій? Які типи фреймів ви знаєте?
3. Які віконні функції ви використали в цій лабораторній роботі? Які ще віконні функції ви знаєте?