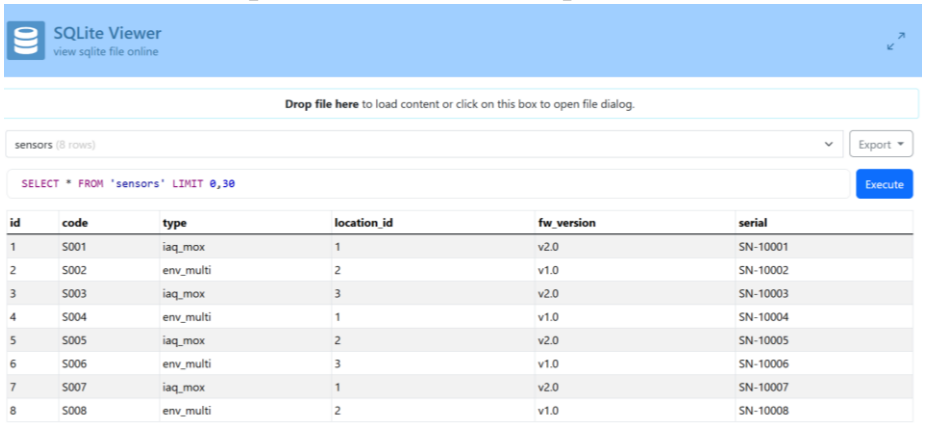
Лабораторна робота №5

# Дослідження вимірювальної інформації у базі SQLite

**Мета:** ознайомитись з принципами зберігання вимірювальних даних у базах SQLite, освоїти аграгацію, фільтрацію індексацію та часові запити

## Хід роботи

1. Завантажити файл measurement.sqlite;
2. Відкрити онлайн SQLite Viewer за адресою: https://inloop.github.io/sqlite-viewer/ та завантажити файл measurement.sqlite




| id | code | type | location_id | fw_version | serial |
|---|---|---|---|---|---|
| 1 | S001 | iaq_mox | 1 | v2.0 | SN-10001 |
| 2 | S002 | env_multi | 2 | v1.0 | SN-10002 |
| 3 | S003 | iaq_mox | 3 | v2.0 | SN-10003 |
| 4 | S004 | env_multi | 1 | v1.0 | SN-10004 |
| 5 | S005 | iaq_mox | 2 | v2.0 | SN-10005 |
| 6 | S006 | env_multi | 3 | v1.0 | SN-10006 |
| 7 | S007 | iaq_mox | 1 | v2.0 | SN-10007 |
| 8 | S008 | env_multi | 2 | v1.0 | SN-10008 |

Рис 5.1

3. Ознайомитись з базою та на сайті: https://dbdiagram.io/d побудувати діаграму бази даних за допомогою файлу dbdiagram.txt.

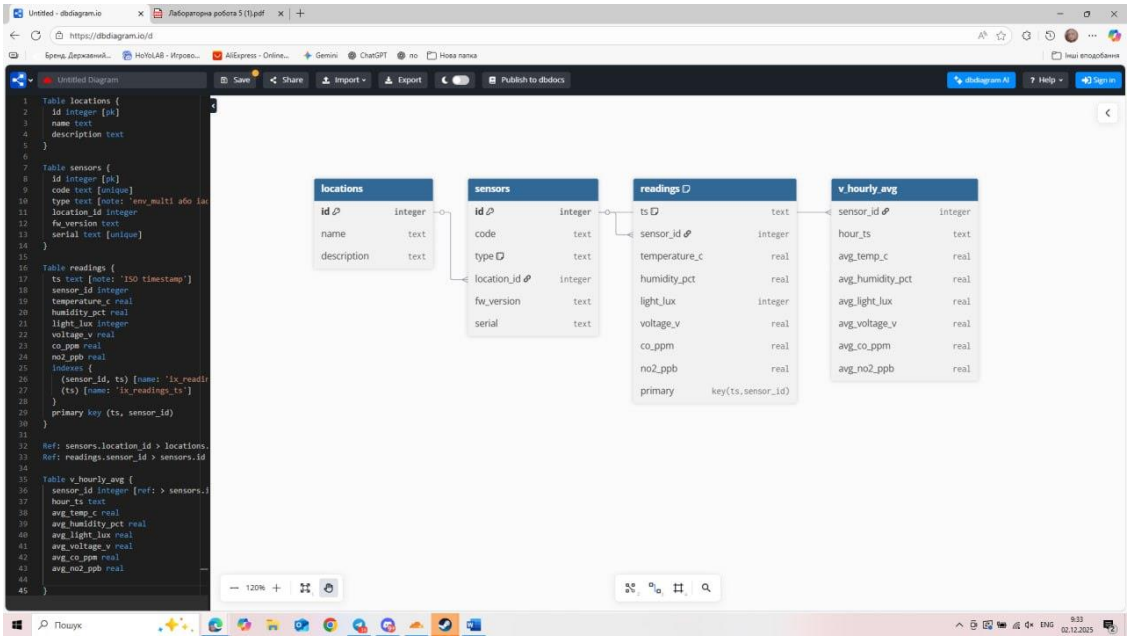



Рис 5.2

4. Вивести скільки всього було виміряно значень з усіх датчиків

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| | | | | | МІВТ.420.002.002-ЗЛ5 | | |
| Змн. | Арк. | № докум. | Підпис | Дата | | | |
| Розроб. | | Величко В.С. | | | ПЗІВС | Літ. | Арк. 1 / Акрушів 6 |
| Перевір. | | Рижук А.В. | | | Звіт з лабораторних робіт | | |
| Н. Контр. | | | | | | ДУ "Житомирська політехніка", ІВТК-5 | |
| Затверд. | | | | | | | |

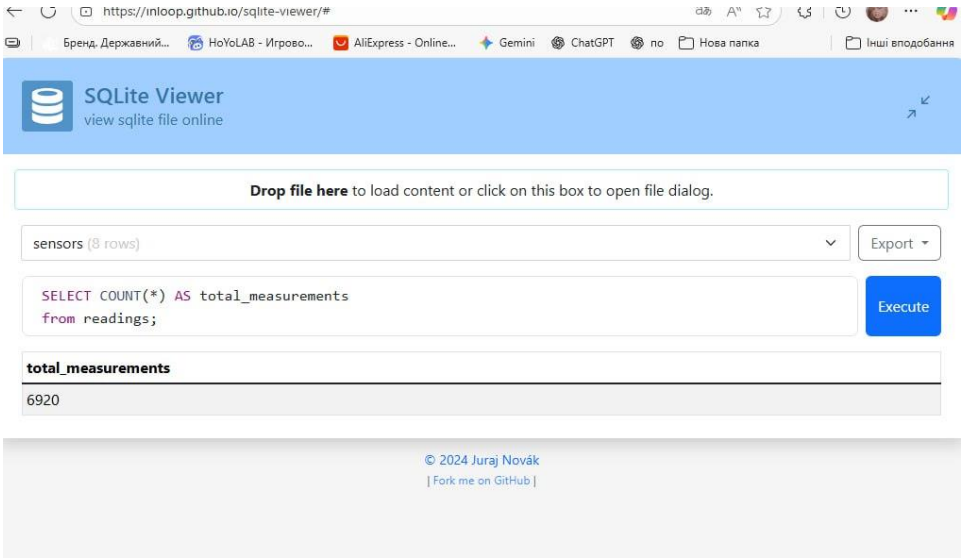
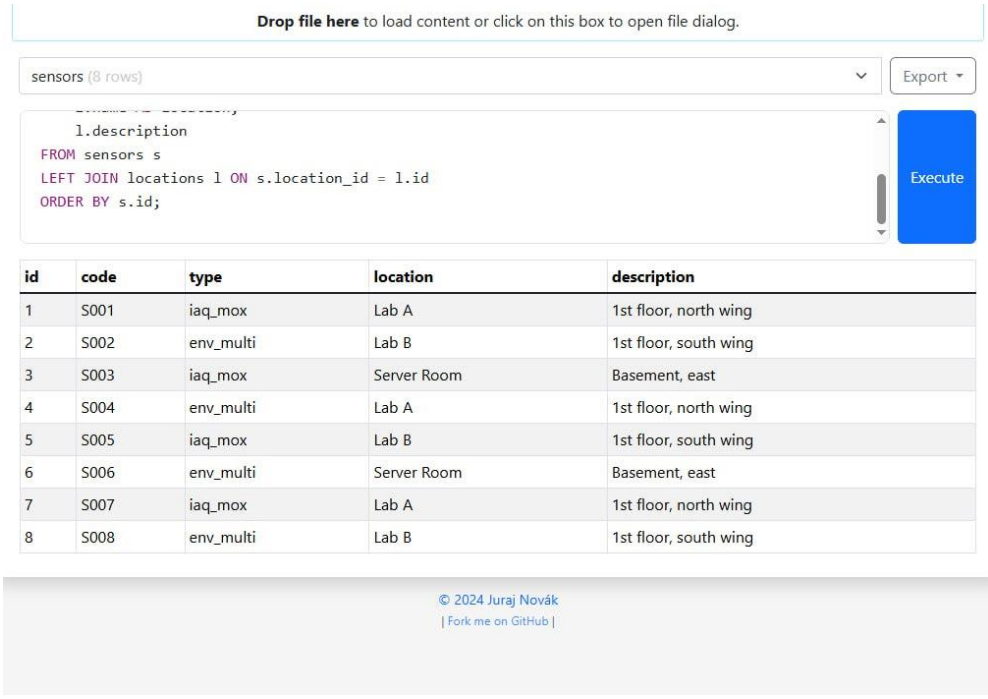



Рис 5.3

5. Вивести список сенсорів та їх місце положення

```
SELECT
   s.id,
   s.code,
   s.type,
   l.name AS location,
   l.description
FROM sensors s
LEFT JOIN locations l ON s.location_id = l.id
ORDER BY s.id;
```




| id | code | type | location | description |
|---|---|---|---|---|
| 1 | S001 | iaq_mox | Lab A | 1st floor, north wing |
| 2 | S002 | env_multi | Lab B | 1st floor, south wing |
| 3 | S003 | iaq_mox | Server Room | Basement, east |
| 4 | S004 | env_multi | Lab A | 1st floor, north wing |
| 5 | S005 | iaq_mox | Lab B | 1st floor, south wing |
| 6 | S006 | env_multi | Server Room | Basement, east |
| 7 | S007 | iaq_mox | Lab A | 1st floor, north wing |
| 8 | S008 | env_multi | Lab B | 1st floor, south wing |

Рис 5.4

6. Вивести останні 10 вимірювань певного сенсора;

```
SELECT
    s.code AS sensor_code,
    s.type AS sensor_type,
    r.temperature_c,
    r.humidity_pct,
    r.light_lux,
    r.voltage_v,
    r.co_ppm,
    r.no2_ppb
FROM readings r
JOIN sensors s ON r.sensor_id = s.id
WHERE r.sensor_id = 1       -- тут ставиш потрібний сенсор
ORDER BY r.ts DESC
LIMIT 10;
```

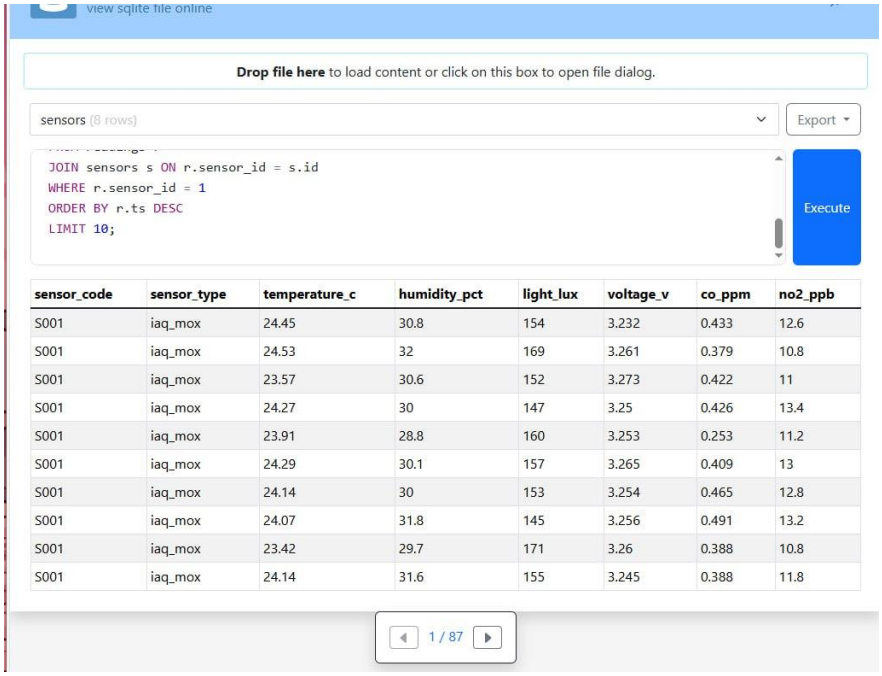



Рис 5.5

7. Вивести кількість вимірювань на день

```
SELECT
    date(ts) as measurement_date,
    count(*) as measurement_count
from readings
group by date(ts)
order by measurement_date desc;
```




Рис 5.6

8. Обчислити середню температуру та вологість за день для кожного сенсора

```
SELECT
    s.code as sensor_code,
    s.type as sensor_type,
    date(r.ts) as measurement_date,
    round(avg(r.temperature_c),2) as avg_temperature_c,
    count(*) as measurement_count
from readings r
join sensors s on r.sensor_id = s.id
group by s.code, s.type, date(r.ts)
order by measurement_date desc, sensor_code;
```

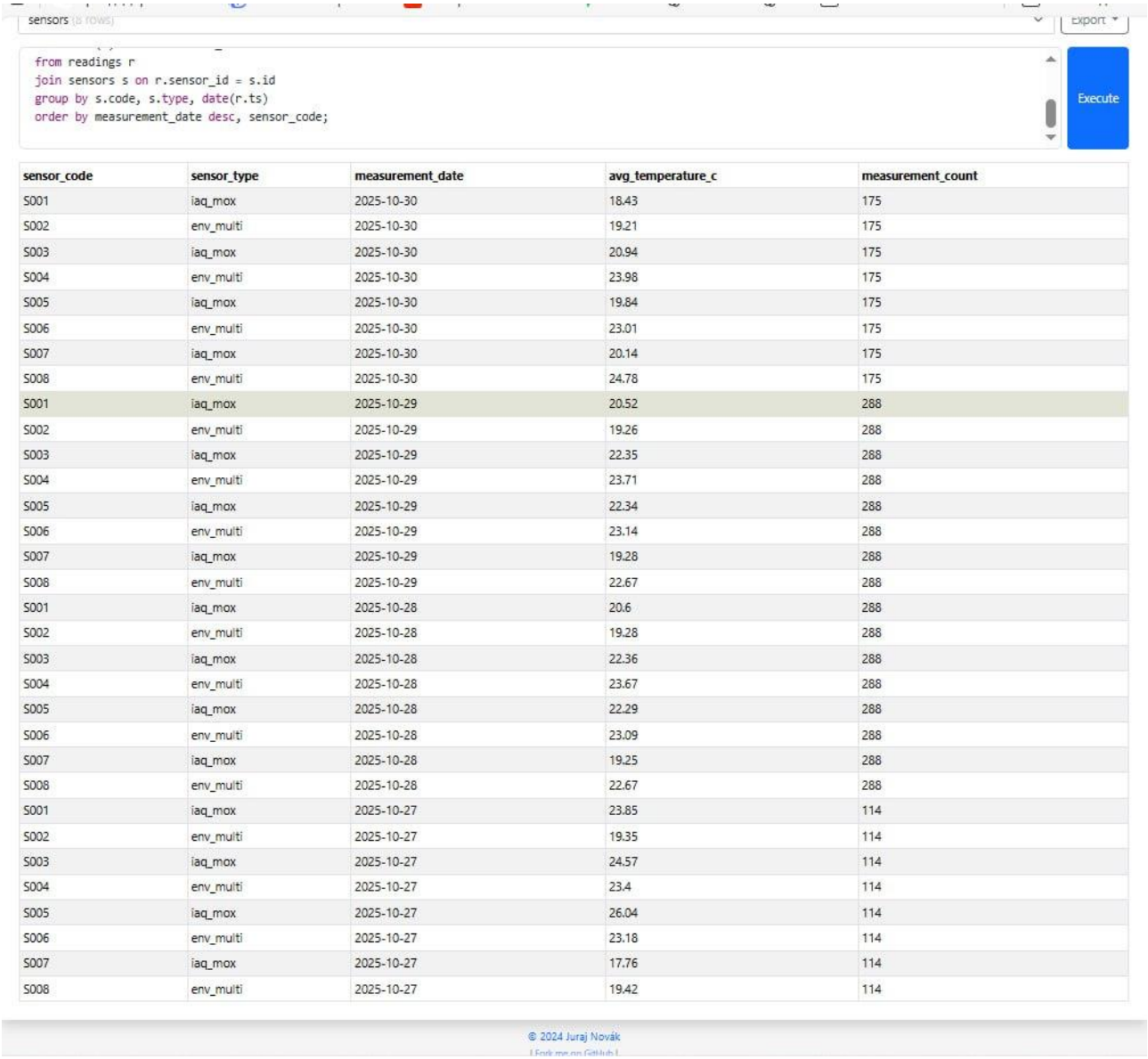

sensors (8 rows)
Export
from readings r
join sensors s on r.sensor_id = s.id
group by s.code, s.type, date(r.ts)
order by measurement_date desc, sensor_code;
Execute
sensor_code | sensor_type | measurement_date | avg_temperature_c | measurement_count
S001 | iaq_mox | 2025-10-30 | 18.43 | 175
S002 | env_multi | 2025-10-30 | 19.21 | 175
S003 | iaq_mox | 2025-10-30 | 20.94 | 175
S004 | env_multi | 2025-10-30 | 23.98 | 175
S005 | iaq_mox | 2025-10-30 | 19.84 | 175
S006 | env_multi | 2025-10-30 | 23.01 | 175
S007 | iaq_mox | 2025-10-30 | 20.14 | 175
S008 | env_multi | 2025-10-30 | 24.78 | 175
S001 | iaq_mox | 2025-10-29 | 20.52 | 288
S002 | env_multi | 2025-10-29 | 19.26 | 288
S003 | iaq_mox | 2025-10-29 | 22.35 | 288
S004 | env_multi | 2025-10-29 | 23.71 | 288
S005 | iaq_mox | 2025-10-29 | 22.34 | 288
S006 | env_multi | 2025-10-29 | 23.14 | 288
S007 | iaq_mox | 2025-10-29 | 19.28 | 288
S008 | env_multi | 2025-10-29 | 22.67 | 288
S001 | iaq_mox | 2025-10-28 | 20.6 | 288
S002 | env_multi | 2025-10-28 | 19.28 | 288
S003 | iaq_mox | 2025-10-28 | 22.36 | 288
S004 | env_multi | 2025-10-28 | 23.67 | 288
S005 | iaq_mox | 2025-10-28 | 22.29 | 288
S006 | env_multi | 2025-10-28 | 23.09 | 288
S007 | iaq_mox | 2025-10-28 | 19.25 | 288
S008 | env_multi | 2025-10-28 | 22.67 | 288
S001 | iaq_mox | 2025-10-27 | 23.85 | 114
S002 | env_multi | 2025-10-27 | 19.35 | 114
S003 | iaq_mox | 2025-10-27 | 24.57 | 114
S004 | env_multi | 2025-10-27 | 23.4 | 114
S005 | iaq_mox | 2025-10-27 | 26.04 | 114
S006 | env_multi | 2025-10-27 | 23.18 | 114
S007 | iaq_mox | 2025-10-27 | 17.76 | 114
S008 | env_multi | 2025-10-27 | 19.42 | 114
© 2024 Juraj Novák

Рис 5.7

9. Визначити мінімальну напругу живлення, вивести список батарей в яких наруга менша за 3.22V

```
SELECT
    s.code AS sensor_code,
    s.type AS sensor_type,
    l.name AS location_name,
    r.ts AS last_measurement_time,
    r.voltage_v,
    s.fw_version
FROM readings r
JOIN sensors s ON r.sensor_id = s.id
LEFT JOIN locations l ON s.location_id = l.id
WHERE r.voltage_v < 3.22
ORDER BY r.voltage_v ASC;
```

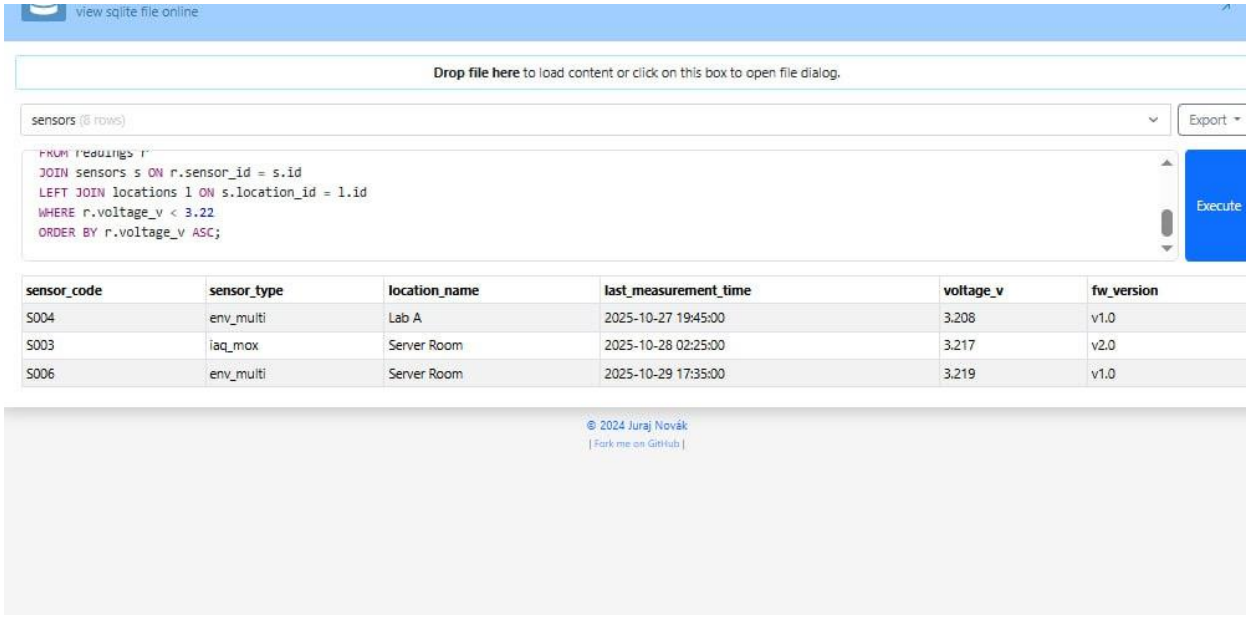



Рис 5.8

**Висновок:** У ході роботи було успішно освоєно принципи роботи з SQLite для зберігання вимірювальних даних, включаючи агрегацію (COUNT, AVG), фільтрацію (WHERE), часові запити та об'єднання таблиць (JOIN). Структура бази даних з таблицями sensors, readings та locations забезпечила ефективне управління даними датчиків, а використання індексів оптимізувало вибірку. Отримані навички підтвердили придатність SQLite для аналізу IoT-даних та часових рядів.

**Контрольні запитання:**

1. Що таке SQL і для чого він використовується?
   **SQL** - мова для роботи з базами даних. Використовується для створення, зміни, пошуку та видалення даних.
2. Як вибрати лише унікальні значення з поля?

```sql
SELECT DISTINCT поле FROM таблиця;
```

3. Як працює BETWEEN і чи включає він межі діапазону?
   **BETWEEN** - включає обидві межі

```sql
WHERE значення BETWEEN min AND max
```

4. Як знайти максимальне/мінімальне значення?

```sql
SELECT MAX(поле), MIN(поле) FROM таблиця;
```

5. Що таке JOIN і для чого він використовується?

   **JOIN** - об'єднує дані з різних таблиць по спільному полю:

   **INNER JOIN** - тільки спільні записи

   **LEFT JOIN** - всі з лівої таблиці + спільні

   **RIGHT JOIN** - всі з правої + спільні