

Лабораторна робота №5

Тема: Використання умовних виразів, сортування та агрегатних функцій у SQLite

Мета роботи:

- Ознайомитися з умовними виразами (WHERE, CASE), сортуванням (ORDER BY) та агрегатними функціями (COUNT, SUM, AVG, MAX, MIN).
- Навчитися виконувати складні SQL-запити для аналізу даних.
- Використати групування (GROUP BY) та фільтрацію (HAVING).

Хід роботи

1. Підготовка до роботи

1. Запустіть SQLite та відкрийте базу test.db
`sqlite3 test.db`
2. Переконайтеся, що таблиця users існує.
`.tables`

Якщо її немає, створіть її:

```
CREATE TABLE users (  
    id INTEGER PRIMARY KEY AUTOINCREMENT,  
    name TEXT NOT NULL,  
    age INTEGER,  
    email TEXT UNIQUE,  
    phone TEXT,  
    salary REAL  
);
```

3. Додайте тестові дані:

```
INSERT INTO users (name, age, email, phone, salary) VALUES  
('Олег', 27, 'oleg@example.com', '+380971234567', 1200.50),
```

('Анна', 22, 'anna@example.com', NULL, 950.75),
('Максим', 35, 'max@example.com', '+380981234567', 1800.00),
('Катерина', 29, 'katya@example.com', '+380931234567', 1500.30),
('Іван', 41, 'ivan@example.com', NULL, 2100.00);

2. Використання умовних виразів (WHERE)

1. Отримати користувачів віком від 25 років:

```
SELECT * FROM users WHERE age >= 25;
```

2. Отримати користувачів, у яких немає номера телефону (NULL):

```
SELECT * FROM users WHERE phone IS NULL;
```

3. Отримати користувачів, які мають зарплату більше 1000 і менше 2000:

```
SELECT * FROM users WHERE salary BETWEEN 1000 AND  
2000;
```

3. Сортування (ORDER BY)

1. Відсортувати користувачів за віком у зростаючому порядку:

```
SELECT * FROM users ORDER BY age ASC;
```

2. Відсортувати користувачів за зарплатою у спадаючому порядку:

```
SELECT * FROM users ORDER BY salary DESC;
```

4. Агрегатні функції (COUNT, SUM, AVG, MAX, MIN)

1. Порахувати кількість користувачів у базі:

```
SELECT COUNT(*) FROM users;
```

2. Знайти середню зарплату всіх користувачів:

```
SELECT AVG(salary) FROM users;
```

3. Отримати максимальну та мінімальну зарплату:

```
SELECT MAX(salary), MIN(salary) FROM users;
```

4. Отримати загальну суму зарплат усіх користувачів:

```
SELECT SUM(salary) FROM users;
```

5. Використання GROUP BY та HAVING

1. Порахувати кількість користувачів у кожній віковій категорії:

```
SELECT age, COUNT(*) FROM users GROUP BY age;
```

2. Отримати середню зарплату для користувачів, розбитих за віком:

```
SELECT age, AVG(salary) FROM users GROUP BY age;
```

3. Отримати вікові групи, де середня зарплата перевищує 1500:

```
SELECT age, AVG(salary) FROM users GROUP BY age HAVING AVG(salary) > 1500;
```

6. Використання умовних операторів (CASE)

1. Створити категорії зарплат (низька, середня, висока):

```
SELECT name, salary,  
CASE  
    WHEN salary < 1000 THEN 'Низька'  
    WHEN salary BETWEEN 1000 AND 2000 THEN 'Середня'  
    ELSE 'Висока'  
END AS salary_category  
FROM users;
```

Контрольні запитання

1. Як отримати лише користувачів, чий вік перевищує 30 років?
2. Якою командою можна відсортувати дані за зарплатою у порядку спадання?
3. Які агрегатні функції використовуються в SQL?
4. Як працює оператор GROUP BY?
5. У чому різниця між WHERE та HAVING?
6. Як за допомогою CASE створити категоризацію даних?