

Упражнение: Създаване на заявки, използвайки HAVING

1. Създайте База данни Training, съдържаща следната таблица с данни:

```
CREATE DATABASE Training
CREATE TABLE exercise_logs
(id INTEGER IDENTITY(1,1) PRIMARY KEY, (SQLite-id INTEGER PRIMARY KEY AUTOINCREMENT)
type nvarchar(50),
minutes INTEGER,
calories INTEGER,
heart_rate INTEGER);
INSERT INTO exercise_logs(type, minutes, calories, heart_rate) VALUES
('biking', 30, 115, 110),
('biking', 10, 45, 105),
('dancing', 15, 200, 120),
('dancing', 15, 165, 120),
('tree climbing', 30, 70, 90),
('tree climbing', 25, 72, 80),
('rowing', 30, 70, 90),
('hiking', 60, 80, 85);
SELECT * FROM exercise_logs;
```

2. Изпълнете следните заявки и направете анализ на резултатите

```
-- Извежда всички записи от таблицата exercise_logs
SELECT * FROM exercise_logs;
-- Групира записите по тип упражнение и изчислява общо изгорените калории за всеки тип
SELECT type, SUM(calories) AS total_calories
FROM exercise_logs
GROUP BY type;
-- Сумира калориите по тип, но само за записи, в които калориите са над 150
SELECT type, SUM(calories) AS total_calories
FROM exercise_logs
WHERE calories > 150
GROUP BY type;
-- Групира по тип и показва само тези групи, при които общите калории са над 150
-- (HAVING филтрира групи, докато WHERE филтрира редове)
SELECT type, SUM(calories) AS total_calories
FROM exercise_logs
GROUP BY type
HAVING SUM(calories) > 150;
-- Изчислява средните калории по тип и показва само типовете със средно над 70 калории
SELECT type, AVG(calories) AS avg_calories
FROM exercise_logs
GROUP BY type
HAVING AVG(calories) > 70;
-- Показва само типовете упражнения, които се срещат поне 2 пъти в таблицата
SELECT type
FROM exercise_logs
GROUP BY type
HAVING COUNT(*) >= 2;
```

WHERE	HAVING
<ul style="list-style-type: none">Филтрира редове преди групиранеРаботи върху оригиналните данниНе може да използва агрегатни функции (SUM, AVG, COUNT...) <p>Филтрира редовете, преди да се прави каквото и да е групиране.</p> <pre>SELECT * FROM exercise_logs WHERE calories > 150;</pre>	<ul style="list-style-type: none">Филтрира групи след GROUP BYРаботи върху агрегирани стойностиМоже да използва SUM, AVG, COUNT и др. <p>Филтрира групите след групиране.</p> <pre>SELECT type, SUM(calories) FROM exercise_logs GROUP BY type HAVING SUM(calories) > 150;</pre>

Упражнение: Създаване на заявки, използвайки HAVING

Създайте База данни Book, съдържаща следната таблица с данни:

```
CREATE DATABASE Book
```

```
CREATE TABLE books (  
    id INTEGER identity(1,1) PRIMARY KEY ,  
    author TEXT,  
    title TEXT,  
    words INTEGER);
```

```
INSERT INTO books (author, title, words) VALUES
```

```
('J.K. Rowling', 'Harry Potter and the Philosopher`s Stone', 79944),  
( 'J.K. Rowling', 'Harry Potter and the Chamber of Secrets', 85141),  
( 'J.K. Rowling', 'Harry Potter and the Prisoner of Azkaban', 107253),  
( 'J.K. Rowling', 'Harry Potter and the Goblet of Fire', 190637),  
( 'J.K. Rowling', 'Harry Potter and the Order of the Phoenix', 257045),  
( 'J.K. Rowling', 'Harry Potter and the Half-Blood Prince', 168923),  
( 'J.K. Rowling', 'Harry Potter and the Deathly Hallows', 197651),  
( 'Stephenie Meyer', 'Twilight', 118501),  
( 'Stephenie Meyer', 'New Moon', 132807),  
( 'Stephenie Meyer', 'Eclipse', 147930),  
( 'Stephenie Meyer', 'Breaking Dawn', 192196),  
( 'J.R.R. Tolkien', 'The Hobbit', 95022),  
( 'J.R.R. Tolkien', 'Fellowship of the Ring', 177227),  
( 'J.R.R. Tolkien', 'Two Towers', 143436),  
( 'J.R.R. Tolkien', 'Return of the King', 134462);
```

```
select * from books
```

```
SELECT author, SUM (words) AS total_words FROM books GROUP BY author;
```

```
SELECT author, SUM (words) AS total_words FROM books GROUP BY author order by author  
desc;
```

```
SELECT author, SUM (words) AS total_words FROM books GROUP BY author order by author asc;
```

Задача 1

Изберете всички автори, които са написали повече от 1 милион думи, като използвате GROUP BY и HAVING. Получената таблица трябва да включва автора 'author' и общия брой думи в колоната 'total_words'.

Решение

```
SELECT author, SUM(words) AS total_words FROM books GROUP BY author  
HAVING SUM(words) > 1000000;
```

Задача 2

Изберете всички автори, които пишат средно по повече от 150 000 думи на книга. Получената таблица трябва да включва автора 'author' и средния брой думи, поместени в колона 'avg_words'.

Решение

```
SELECT author, AVG(words) AS avg_words FROM books GROUP BY author  
HAVING AVG(words) > 150000;
```