Aliases and GROUP BY

Aliases are the best way to use a column with various clauses, but this is not the case with the GROUP BY clause. Column aliases cannot be used with the GROUP BY clause. This is because the GROUP BY clause is processed before the SELECT statement. Let's see the execution flow of the GROUP BY statements.

1. FROM clause
2. WHERE clause
3. GROUP BY clause
4. HAVING clause
5. SELECT clause
6. ORDER BY clause

We can better understand this flow of order with the help of an example.

```
SELECT job_id AS "Job ID",
       ROUND(AVG(salary)) AS "Average Salary"
FROM employees
WHERE manager_id = 101
GROUP BY job_id
HAVING AVG(salary) > 10000
ORDER BY "Average Salary";
```

In this query, firstly the "employees" table is selected using the FROM clause. Then the WHERE clause is used to filter the data based on the manager_id. After this, the filtered data is grouped by job_id which is then filtered using the HAVING clause based on average salary. After all this, the columns are selected using the SELECT statement and then ordered based on the ORDER BY clause.

As it can be seen that GROUP BY is used before the SELECT statement in which aliases are defined, group by does not know about the aliases by then and it will throw an error. But this is not the case with the ORDER BY as it is used after the SELECT statement and knows about the aliases defined in the SELECT statement.

Important Points

- Column aliases cannot be used with the GROUP BY clause.
- The ORDER BY clause cannot have any other individual columns that the GROUP BY clause has.
- We can use the WHERE clause to restrict the resulting data.