In Class 6 completed aggregate completed

**GROUP BY AND HAVING**

ERROR

QUERY

SELECT sum(rating),length , COUNT(\*) FROM film GROUP BY rental\_rate;

Error Code: 1055. Expression #2 of SELECT list is not in GROUP BY clause and contains nonaggregated column 'mavenmovies.film.length' which is not functionally dependent on columns in GROUP BY clause; this is incompatible with sql\_mode=only\_full\_group\_by

The error you are encountering occurs because the ONLY\_FULL\_GROUP\_BY SQL mode requires that all non-aggregated columns in the SELECT clause must be included in the GROUP BY clause. In your query, length is neither aggregated nor included in the GROUP BY clause, which causes the error.

keyboard\_arrow\_down

### **Solutions**

1. **Include length in the GROUP BY clause**: If you want to group by both rental\_rate and length, you should include both columns in the GROUP BY clause.

SELECT SUM(rating), length, COUNT(\*)  
FROM film  
GROUP BY rental\_rate, length;

1. **Aggregate the length column**: If you need to aggregate the length column in some way (e.g., taking the average or maximum length), you can use an aggregate function on it.

SELECT SUM(rating), AVG(length) AS avg\_length, COUNT(\*)  
FROM film  
GROUP BY rental\_rate;

### **Explanation**

* **Solution 1**: Grouping by both rental\_rate and length allows you to include both columns in the SELECT clause without causing an error.
* **Solution 2**: Aggregating length allows you to keep it in the SELECT clause without including it in the GROUP BY clause.

### **Examples**

#### **Example 1: Grouping by Both Columns**

This query groups by both rental\_rate and length, and calculates the sum of rating and the count of rows for each combination of rental\_rate and length.

SELECT SUM(rating), length, COUNT(\*)  
FROM film  
GROUP BY rental\_rate, length;

#### **Example 2: Aggregating the length Column**

This query groups by rental\_rate, calculates the sum of rating, the average length, and the count of rows for each rental\_rate.

SELECT SUM(rating), AVG(length) AS avg\_length, COUNT(\*)  
FROM film  
GROUP BY rental\_rate;

### **Choosing the Appropriate Solution**

* **Solution 1**: Use this if you need to maintain the distinct length values within each rental\_rate group.
* **Solution 2**: Use this if you are looking for aggregated measures (e.g., average length) within each rental\_rate group.

### **Additional Consideration**

If you need to include more columns in your query while adhering to the ONLY\_FULL\_GROUP\_BY mode, you may need to reconsider your data aggregation logic. Ensure that every column in the SELECT list that is not an aggregate function is also included in the GROUP BY clause.

### **Final Note**

If you decide to disable ONLY\_FULL\_GROUP\_BY mode (not recommended for production environments as it ensures more consistent and predictable query results), you can do so by modifying the SQL mode. However, it's better to adapt your queries to comply with ONLY\_FULL\_GROUP\_BY:

SET sql\_mode=(SELECT REPLACE(@@sql\_mode,'ONLY\_FULL\_GROUP\_BY',''));

Remember, adhering to the ONLY\_FULL\_GROUP\_BY rule helps maintain clear and correct SQL queries, especially in complex databases.