

The error you're encountering happens because of SQL's `ONLY_FULL_GROUP_BY` mode, which requires all non-aggregated columns in the `SELECT` list to be included in the `GROUP BY` clause or to be functionally dependent on the columns being grouped.

In your first query:

sql

Copy code

```
select Region , Category , sum(Sales) as total_sales
from superstore
group by Region;
```

You are trying to select `Category` along with `Region` and `sum(Sales)`. However, `Category` is not aggregated (with an aggregate function like `SUM`, `COUNT`, etc.), nor is it included in the `GROUP BY` clause, which leads to the error.

Solution:

There are two common ways to resolve this:

1. Add `Category` to the `GROUP BY` clause: You can group by both `Region` and `Category` to get total sales by both dimensions:

sql

Copy code

```
select Region, Category, sum(Sales) as total_sales
from superstore
group by Region, Category;
```

2. Use an aggregate function for `Category`: If you want to keep `Region` in the `GROUP BY` clause

Message ChatGPT



ChatGPT can make mistakes. Check important info.

Search



Solution:

There are two common ways to resolve this:

1. Add **Category** to the **GROUP BY** clause: You can group by both **Region** and **Category** to get total sales by both dimensions:

sql

Copy code

```
select Region, Category, sum(Sales) as total_sales
from superstore
group by Region, Category;
```

2. Use an aggregate function for **Category**: If you want to keep **Region** in the **GROUP BY** clause and aggregate **Category** in some way (e.g., **GROUP_CONCAT** for multiple categories), you can do this:

sql

Copy code

```
select Region, GROUP_CONCAT(DISTINCT Category) as categories, sum(Sales) as total_sale
from superstore
group by Region;
```

This will return a list of distinct categories for each **Region** and their corresponding total sales.

Choose the approach based on how you want to group and display the data.

Why the second query works:



In your second query:

Message ChatGPT



ChatGPT can make mistakes. Check important info.

Why the second query works:

In your second query:

sql

Copy code

```
select Region, sum(Sales) as total_sales  
from superstore  
group by Region;
```

You only group by `Region`, and since there is no other non-aggregated column like `Category` in the `SELECT` list, it works as expected.



what is non-aggregated column

A non-aggregated column is a column in your SQL query's `SELECT` statement that is not wrapped in an aggregate function (such as `SUM()`, `COUNT()`, `AVG()`, `MIN()`, `MAX()`, etc.) and is also not part of the `GROUP BY` clause.

In simpler terms:

- Non-aggregated columns are just the regular columns you're selecting without any form of calculation applied to them.
- These columns can be any field that is directly included in your `SELECT` statement but is not being used in a group operation.



Message ChatGPT



ChatGPT can make mistakes. Check important info.

Search





A **non-aggregated column** is a column in your SQL query's **SELECT** statement that is not wrapped in an aggregate function (such as **SUM()**, **COUNT()**, **AVG()**, **MIN()**, **MAX()**, etc.) and is also not part of the **GROUP BY** clause.

In simpler terms:

- **Non-aggregated columns** are just the regular columns you're selecting without any form of calculation applied to them.
- These columns can be any field that is directly included in your **SELECT** statement but is **not** being used in a group operation.

Example:

Consider the following query:

sql

Copy code

```
SELECT Region, Category, SUM(Sales) AS total_sales
FROM superstore
GROUP BY Region;
```

Here:

- **Region** is in the **GROUP BY** clause, so it's grouped.
- **Category** is a **non-aggregated column** because it's neither aggregated (like **SUM()**, **COUNT()**) nor part of the **GROUP BY** clause.
- **SUM(Sales)** is an **aggregated column** because it uses the **SUM()** function to calculate the total sales.

Message ChatGPT



ChatGPT can make mistakes. Check important info.

Search



Why does this matter?

When you're grouping data in SQL using `GROUP BY`, SQL requires that any column that is not part of the `GROUP BY` clause must have an aggregate function applied to it. This ensures that the result is logically consistent, as SQL needs to know how to combine values from non-grouped columns.

In this example:

- `Region` is okay because it's in the `GROUP BY` clause.
- `Category` is not in the `GROUP BY` clause, and there is no aggregate function applied to it, which causes an error in `ONLY_FULL_GROUP_BY` mode.

How to fix it:

You can either:

1. Add `Category` to the `GROUP BY` clause:

```
sql
SELECT Region, Category, SUM(Sales) AS total_sales
FROM superstore
GROUP BY Region, Category;
```

Copy code

This will group the data by both `Region` and `Category`.

2. Apply an aggregate function to `Category`: If you want to keep grouping only by `Region`, you could use an aggregate function like `GROUP_CONCAT()` to combine the `Category` values:

```
sql
```



Copy code

```
SELECT Region, GROUP_CONCAT(DISTINCT Category) AS categories, SUM(Sales) AS total_sales
```

Message ChatGPT



ChatGPT can make mistakes. Check important info.

Copy code

This will group the data by both **Region** and **Category**.

2. **Apply an aggregate function to **Category****: If you want to keep grouping only by **Region**, you could use an aggregate function like **GROUP_CONCAT()** to combine the **Category** values:

sql

Copy code

```
SELECT Region, GROUP_CONCAT(DISTINCT Category) AS categories, SUM(Sales) AS total_sale
FROM superstore
GROUP BY Region;
```

This would list all categories for each **Region** and also show the sum of **Sales**.

Summary:

- **Aggregated columns**: Columns that are processed by aggregate functions like **SUM()**, **COUNT()**, etc.
- **Non-aggregated columns**: Columns that are not processed by any aggregate function and are not included in the **GROUP BY** clause.

If you want to select non-aggregated columns while grouping by other columns, those non-aggregated columns must either be included in the **GROUP BY** clause or wrapped in an aggregate function.



Message ChatGPT



ChatGPT can make mistakes. Check important info.

Search

