SQL Group Functions

Module 2: Quick Review

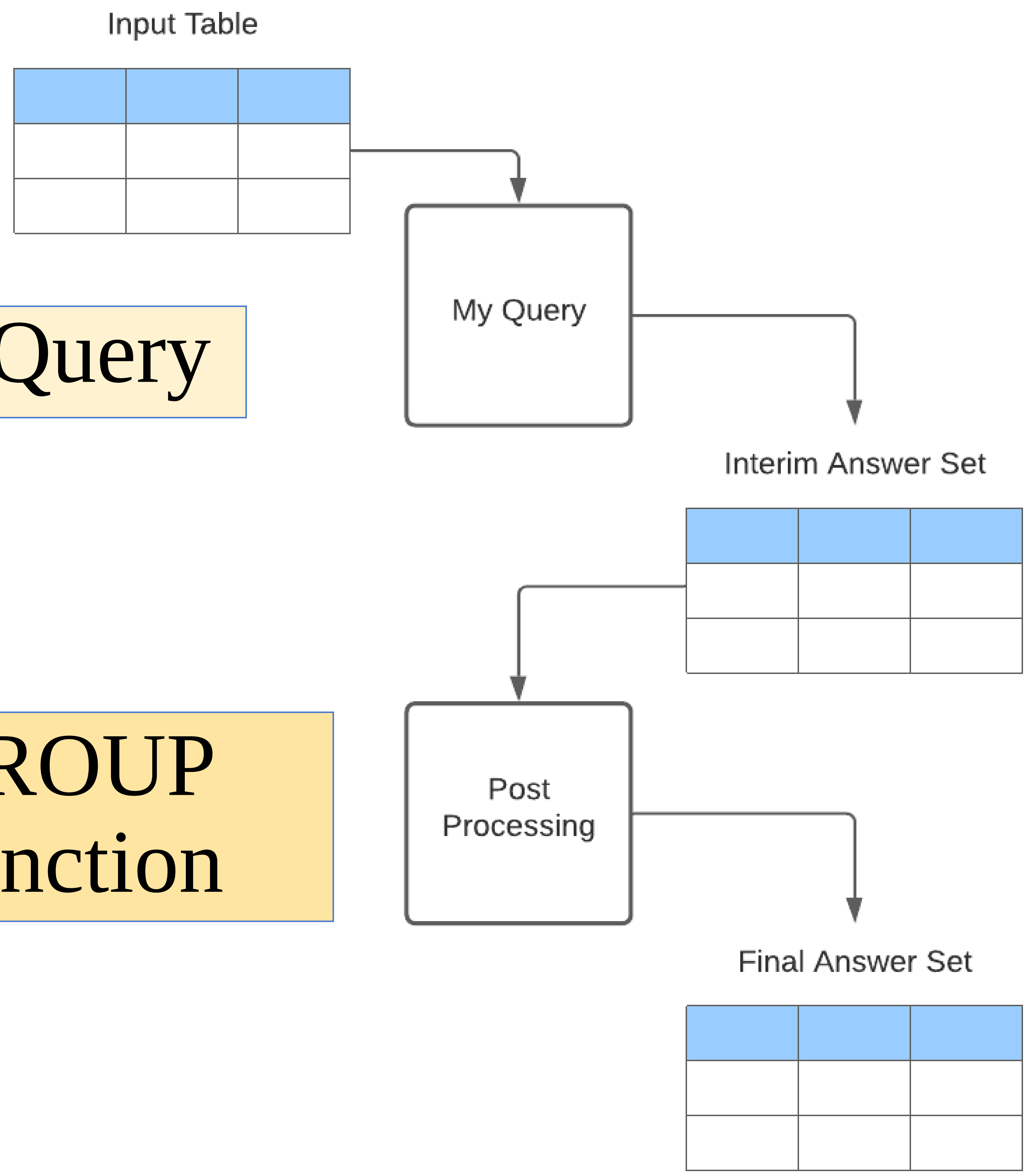
- Select statement
- Where clause
- Distinct
- Order By, Limit
- Dates, Date Functions
- Nulls

SQL Group Functions

Module 3:

- Group Functions
- Group By + Having
- Sub Queries

SQL Group Functions



Query

GROUP Function

SQL Group Functions

The Group Function

- SQL creates an interim answer set
- SQL processes the group function against the interim answer set
- Final answer set contains only the final result for the function

SQL Group Functions

Note on GROUP FUNCTION with a WHERE clause:

1. The WHERE clause determines the number of rows in the INTERIM answer set
2. The GROUP function does its calculation against the rows in the INTERIM answer set AFTER the WHERE is done

SQL Group Functions

SUM	Sums the non-null values in a column
AVG	Calculates the average of the non-null values in a column
COUNT	Counts how many rows have a non-null value in a column
MIN	Returns the lowest non-null value in a column
MAX	Returns the highest non-null value in a column

SQL Group Functions

NOTE:

- SUM, AVG can only be used with NUMERIC columns
- MIN, MAX can be used with any data type column
- COUNT can be used with any data type column, or with a (*) to simply count rows

SQL Group Functions

```
select COUNT(*) as "Staff Size"
   from "alanparadise/nw"."employees";
select COUNT(Distinct Country) as "Countries"
   from "alanparadise/nw"."customers";
select SUM(unitsinstock) as "Inventory"
   from "alanparadise/nw"."products";
select MAX(unitprice) as "High Price"
   from "alanparadise/nw"."products";
select MIN(unitPrice) as "Low Price"
   from "alanparadise/nw"."products";
```

SQL Group Functions

```
select AVG(unitprice) as "Average Price"
   from "alanparadise/nw"."products";


select to_char(avg(UnitPrice),'999,999.99') as "Average Price"
   from "alanparadise/nw"."products";
```

NOTE: It rounded the result...

SQL Group Functions

Count Example –

```
select COUNT(*) as "Staff Size"
  from "alanparadise/nw"."employees";

select COUNT(employeeid) as "Staff Size"
  from "alanparadise/nw"."employees";
```

Are these the same? How are they different?

SQL Group Functions

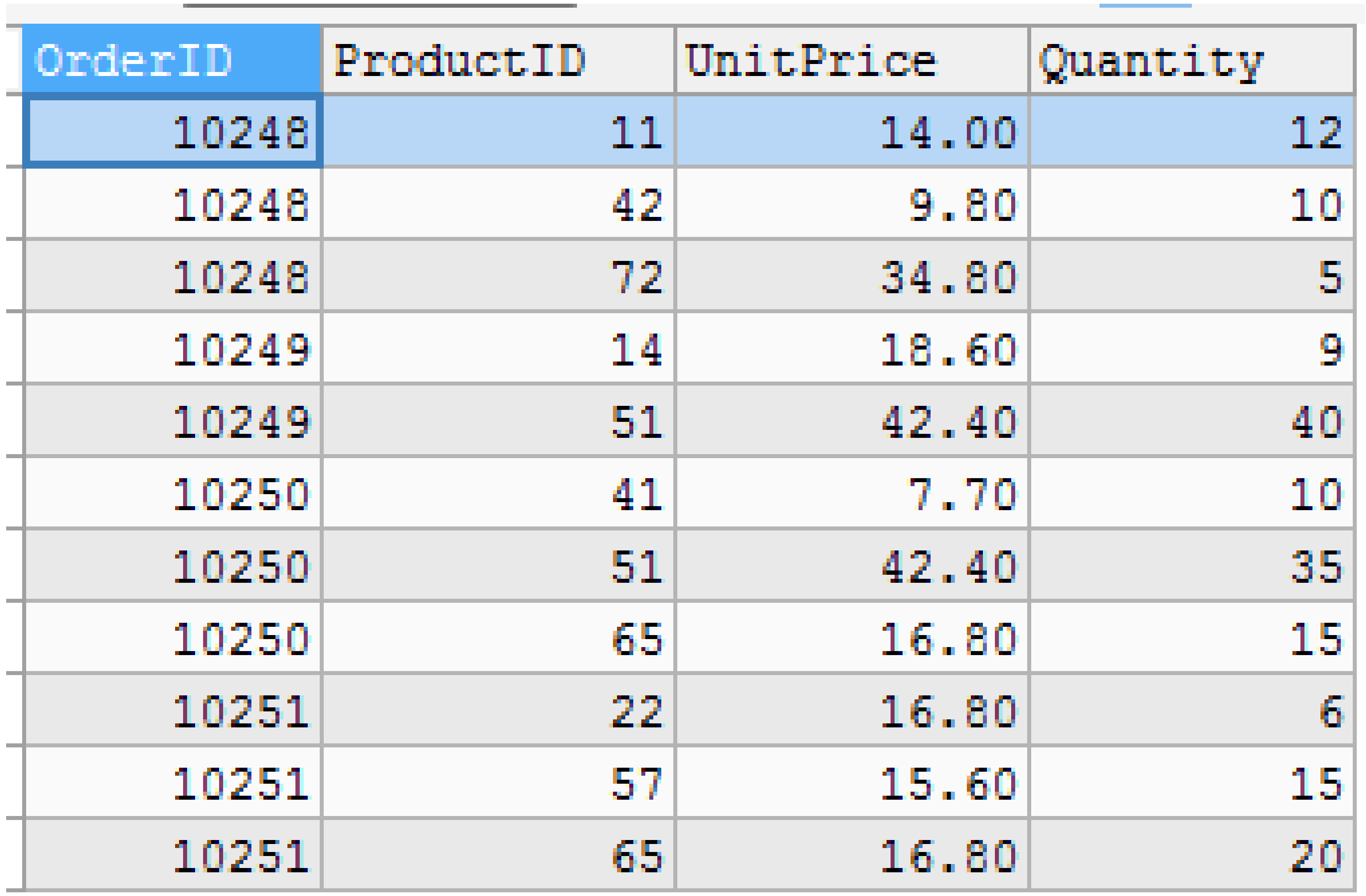
RULE:

The level of the group function must match the level of detail in your select statement.

Look at the example on the next slide

SQL Group Functions

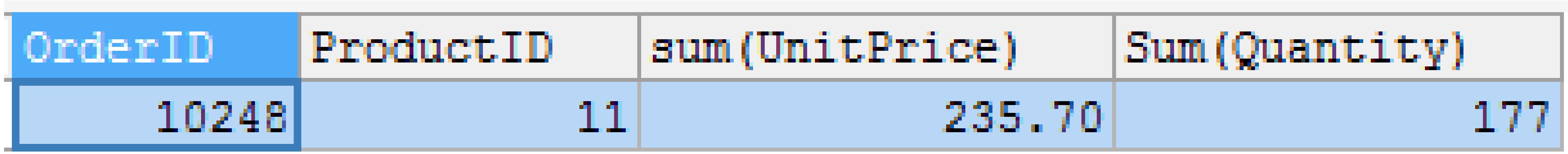
```
SELECT OrderID, ProductID, UnitPrice, Quantity
    FROM "alanparadise/nw"."orderdetails"
    WHERE OrderID in (10248, 10249, 10250, 10251);
```



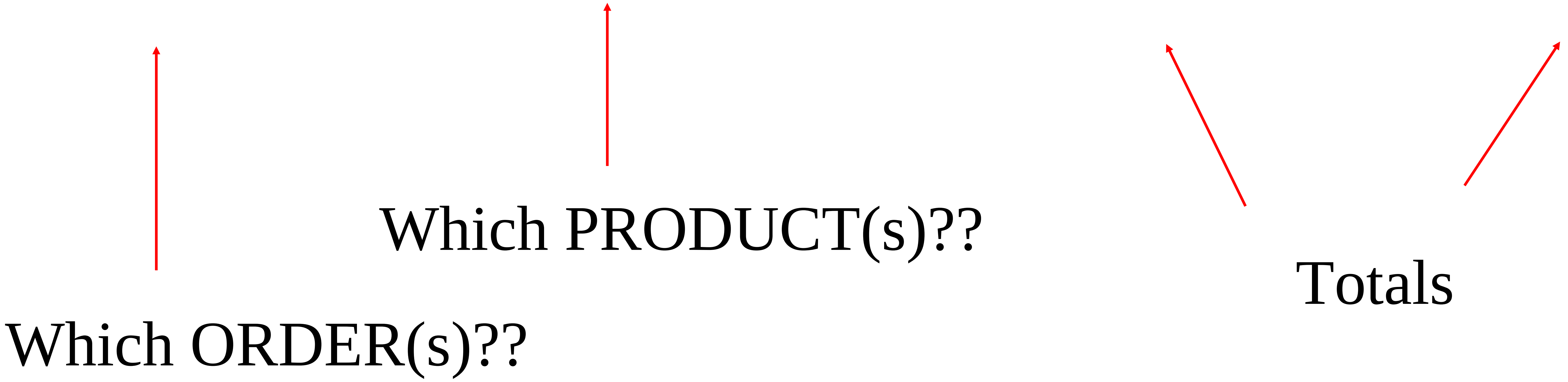
OrderID	ProductID	UnitPrice	Quantity
10248	11	14.00	12
10248	42	9.80	10
10248	72	34.80	5
10249	14	18.60	9
10249	51	42.40	40
10250	41	7.70	10
10250	51	42.40	35
10250	65	16.80	15
10251	22	16.80	6
10251	57	15.60	15
10251	65	16.80	20

SQL Group Functions

```
SELECT OrderID, ProductID, SUM(UnitPrice), SUM(Quantity)
  FROM "alanparadise/nw"."orderdetails"
  WHERE OrderID in (10248, 10249, 10250, 10251);
```



OrderID	ProductID	sum(UnitPrice)	Sum(Quantity)
10248	11	235.70	177



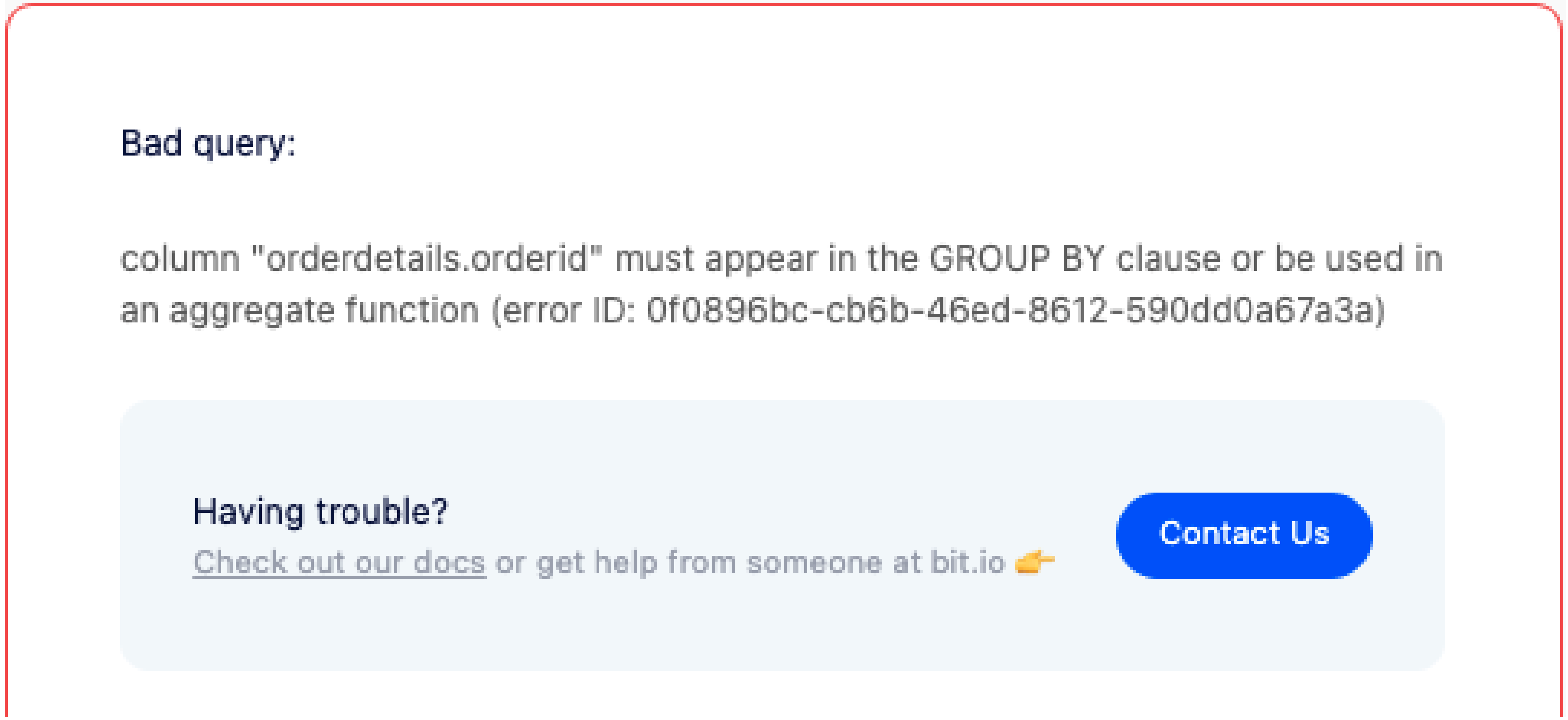
Which PRODUCT(s)??

Totals

Which ORDER(s)??

SQL Group Functions

This level mis-match generates an error



Bad query:

column "orderdetails.orderid" must appear in the GROUP BY clause or be used in an aggregate function (error ID: 0f0896bc-cb6b-46ed-8612-590dd0a67a3a)

Having trouble?

Check out our docs or get help from someone at bit.io 👉

SQL Group Functions

To comply with this rule, and to avoid this error, we must consider the next topic: GROUP BY