1. **LEAD and LAG Function:**

LAG helps track wheather sales are increasing or decreasing compared to the previous day.

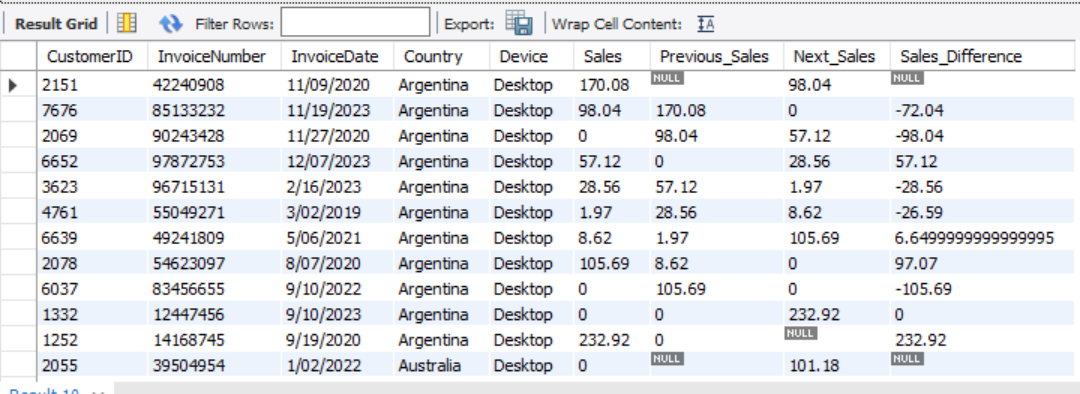
LEAD helps anticipate sales trends for future comparisons.

select CustomerID, InvoiceNumber, InvoiceDate,Country,Device,Sales,

LAG(Sales) over(partition by Device,Country order by InvoiceDate) as Previous\_Sales,

LEAD(Sales) over(partition by Device, Country order by InvoiceDate) as Next\_Sales,

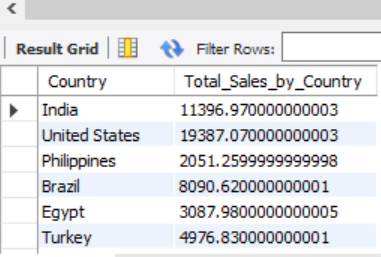
Sales - LAG(Sales) over(partition by Device, Country order by InvoiceDate) as Sales\_Difference from test.`e-commerce`;



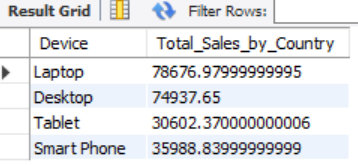
1. **GREATEST AND LEAST**

Scenario : Identify the best and worst performing products and regions.

select Country ,sum(Sales) as Total\_Sales\_by\_Country FROM test.`e-commerce` group by Country;



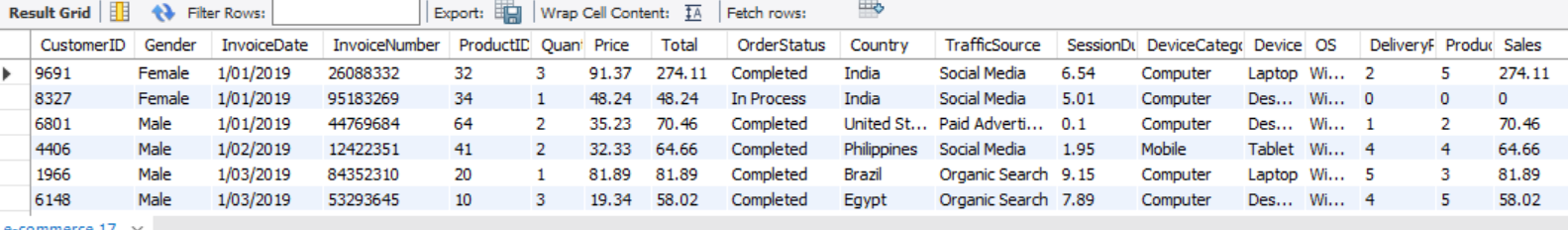
select Device ,sum(Sales) as Total\_Sales\_by\_Country FROM test.`e-commerce` group by Device;



The **PIVOT** operation in SQL is used to transform rows into columns, allowing for easier aggregation and comparison of data. This is especially useful for creating summary tables or reports.

Scenario:

Below is the data:



**Goal**: Create a pivot table where Country (India, US etc.) become columns, and the Sales is displayed for each Country.

SELECT Device,

sum(case when Country = 'India' then Sales else 0 end) as Sales\_India,

sum(case when Country = 'United States' then Sales else 0 end) as Sales\_US,

sum(case when Country = 'Philippines' then Sales else 0 end) as Sales\_Phillipines,

sum(case when Country = 'Brazil' then Sales else 0 end) as Sales\_Brazil,

sum(case when Country = 'Egypt' then Sales else 0 end) as Sales\_Egypt

FROM test.`e-commerce`

|  |
| --- |
| group by Device; |
|  |
|  |
|  |
| To apply the **GREATEST** and **LEAST** functions on a pivot table generated by a SQL query, the best practice in the industry is to use **Common Table Expressions (CTEs)** or a **subquery**. This method improves code readability and allows you to reuse the pivot query without duplicating it.   1. **SUB Query Method:**   select Device,Sales\_US,Sales\_Phillipines,Sales\_Brazil,Sales\_Egypt,  greatest(Sales\_India,Sales\_US,Sales\_Phillipines,Sales\_Brazil,Sales\_Egypt) as Highest\_Sales,  least(Sales\_India,Sales\_US,Sales\_Phillipines,Sales\_Brazil,Sales\_Egypt) as Lowest\_Sales  from  (  SELECT Device,  sum(case when Country = 'India' then Sales else 0 end) as Sales\_India,  sum(case when Country = 'United States' then Sales else 0 end) as Sales\_US,  sum(case when Country = 'Philippines' then Sales else 0 end) as Sales\_Phillipines,  sum(case when Country = 'Brazil' then Sales else 0 end) as Sales\_Brazil,  sum(case when Country = 'Egypt' then Sales else 0 end) as Sales\_Egypt  FROM test.`e-commerce`  group by Device  ) as Pivot\_Table; |

**CTE Method:**

With pivottable as (

select

device,

sum(case when Country = 'India' then Sales else 0 end) as Sales\_India,

sum(case when Country = 'United States' then Sales else 0 end) as Sales\_US,

sum(case when Country = 'Philippines' then Sales else 0 end) as Sales\_Phillipines,

sum(case when Country = 'Brazil' then Sales else 0 end) as Sales\_Brazil,

sum(case when Country = 'Egypt' then Sales else 0 end) as Sales\_Egypt

FROM test.`e-commerce`

group by Device)

select Device,Sales\_US,Sales\_Phillipines,Sales\_Brazil,Sales\_Egypt,

greatest(Sales\_India,Sales\_US,Sales\_Phillipines,Sales\_Brazil,Sales\_Egypt) as Highest\_Sales,

least(Sales\_India,Sales\_US,Sales\_Phillipines,Sales\_Brazil,Sales\_Egypt) as Lowest\_Sales

from pivottable;

