## 1. \*\*Monthly Customer Rank by Spend\*\*    - For each month (based on `order\_date`), rank customers by \*\*total order value\*\* in that month using `RANK()`.    - Output: month (YYYY-MM), customer\_id, total\_monthly\_spend, rank\_in\_month.

SELECT

TO\_CHAR(*o*.order\_date, 'YYYY-MM') AS *month*,

*c*.customer\_id,

SUM(*oi*.quantity \* *oi*.unit\_price) AS *total\_monthly\_spend*,

RANK() OVER (PARTITION BY TO\_CHAR(o.order\_date, 'YYYY-MM') ORDER BY SUM(oi.quantity \* oi.unit\_price) DESC) AS *rank\_in\_month*

FROM training\_ecom.customers *c*

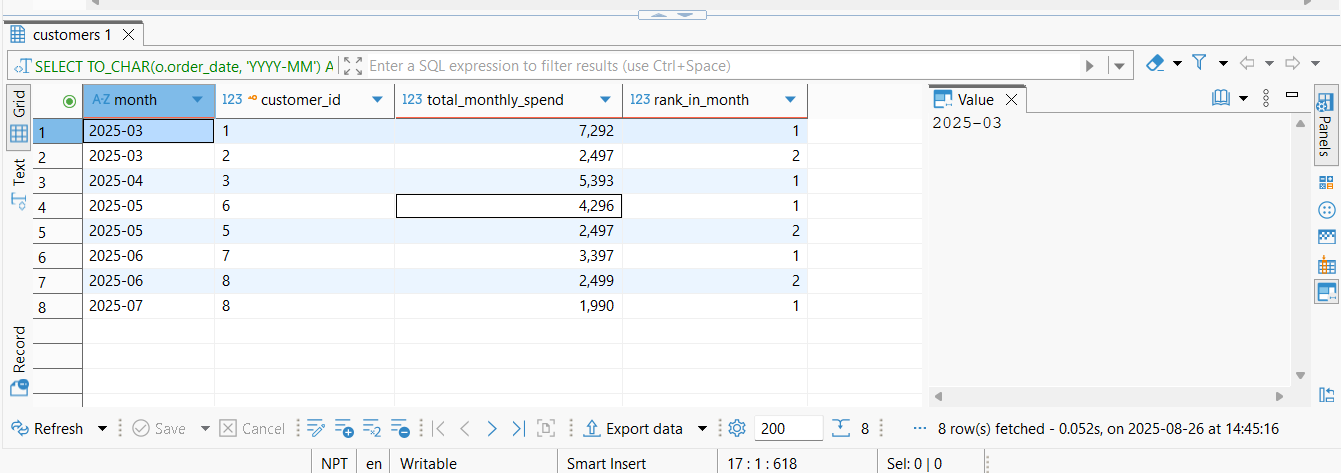
JOIN training\_ecom.orders *o* ON *c*.customer\_id = *o*.customer\_id

JOIN training\_ecom.order\_items *oi* ON *o*.order\_id = *oi*.order\_id

WHERE *o*.status != 'cancelled'

GROUP BY TO\_CHAR(*o*.order\_date, 'YYYY-MM'), *c*.customer\_id

ORDER BY *month*, *rank\_in\_month*;



## 2. \*\*Share of Basket per Item\*\*    - For each order, compute each item's \*\*revenue share\*\* in that order:      `item\_revenue / order\_total` using `SUM() OVER (PARTITION BY order\_id)`.

**SELECT**

*oi*.order\_id,

*oi*.product\_id,

*p*.product\_name,

(*oi*.quantity \* *oi*.unit\_price) **AS** *item\_revenue*,

(*oi*.quantity \* *oi*.unit\_price) / **SUM**(*oi*.quantity \* *oi*.unit\_price) **OVER** (**PARTITION** **BY** oi.order\_id) **AS** *revenue\_share*

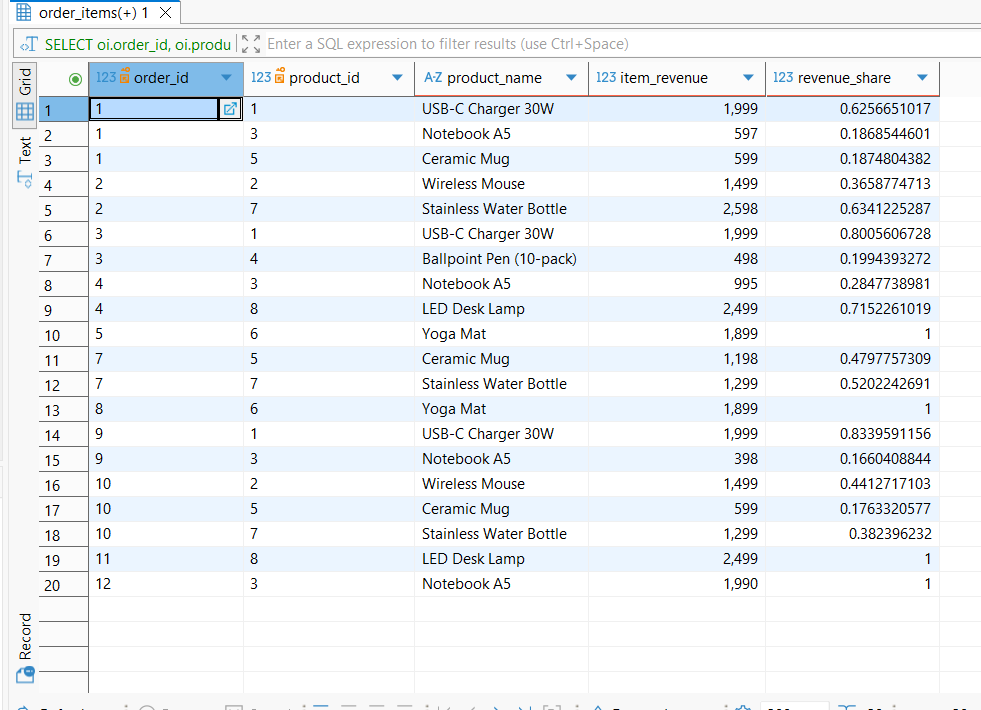
**FROM** training\_ecom.order\_items *oi*

**JOIN** training\_ecom.products *p* **ON** *oi*.product\_id = *p*.product\_id

**JOIN** training\_ecom.orders *o* **ON** *oi*.order\_id = *o*.order\_id

**WHERE** *o*.status != 'cancelled'

**ORDER** **BY** *oi*.order\_id, *oi*.product\_id;



**3. \*\*Time Between Orders (per Customer)\*\*  
   - Show days since the \*\*previous order\*\* for each customer using `LAG(order\_date)` and `AGE()`**

**SELECT**

*c*.customer\_id,

*c*.full\_name,

*o*.order\_id,

*o*.order\_date,

**LAG**(*o*.order\_date) **OVER** (**PARTITION** **BY** c.customer\_id **ORDER** **BY** o.order\_date) **AS** *previous\_order\_date*,

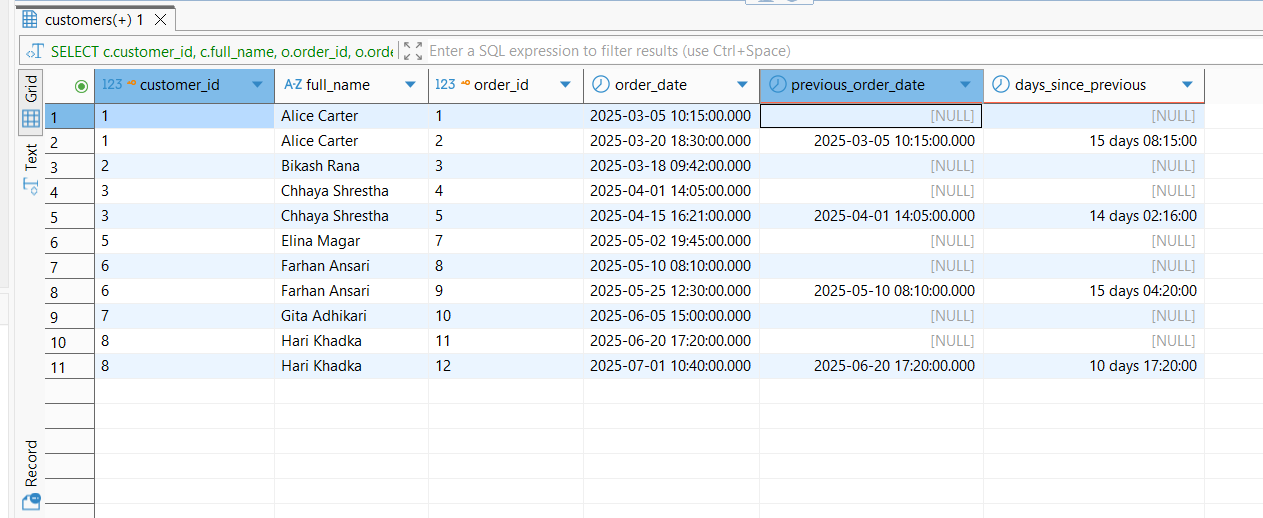
**AGE**(*o*.order\_date, **LAG**(*o*.order\_date) **OVER** (**PARTITION** **BY** c.customer\_id **ORDER** **BY** o.order\_date)) **AS** *days\_since\_previous*

**FROM** training\_ecom.customers *c*

**JOIN** training\_ecom.orders *o* **ON** *c*.customer\_id = *o*.customer\_id

**WHERE** *o*.status != 'cancelled'

**ORDER** **BY** *c*.customer\_id, *o*.order\_date;



**4. \*\*Product Revenue Quartiles\*\*  
   - Compute total revenue per product and assign \*\*quartiles\*\* using `NTILE(4)` over total revenue.**

**SELECT**

*p*.product\_id,

*p*.product\_name,

**SUM**(*oi*.quantity \* *oi*.unit\_price) **AS** *total\_revenue*,

**NTILE**(4) **OVER** (**ORDER** **BY** **SUM**(oi.quantity \* oi.unit\_price)) **AS** *revenue\_quartile*

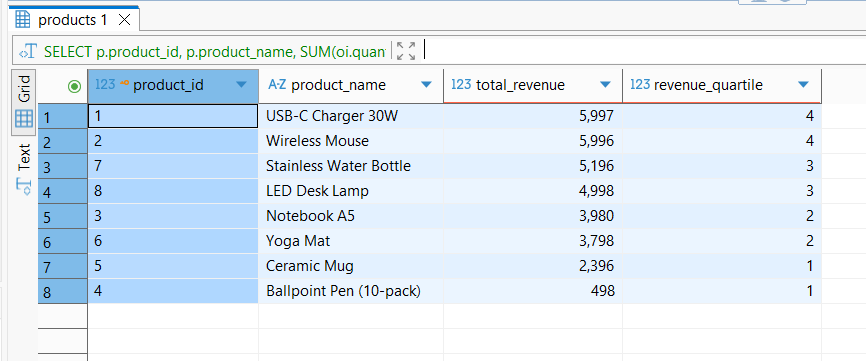
**FROM** training\_ecom.products *p*

**LEFT** **JOIN** training\_ecom.order\_items *oi* **ON** *p*.product\_id = *oi*.product\_id

**LEFT** **JOIN** training\_ecom.orders *o* **ON** *oi*.order\_id = *o*.order\_id **AND** *o*.status != 'cancelled'

**GROUP** **BY** *p*.product\_id, *p*.product\_name

**ORDER** **BY** *total\_revenue* **DESC**;



## 5. \*\*First and Last Purchase Category per Customer\*\*    - For each customer, show the \*\*first\*\* and \*\*most recent\*\* product category they've bought using `FIRST\_VALUE` and `LAST\_VALUE` over `order\_date`.

**SELECT** **DISTINCT**

*c*.customer\_id,

*c*.full\_name,

**FIRST\_VALUE**(*p*.category) **OVER** (**PARTITION** **BY** c.customer\_id **ORDER** **BY** o.order\_date **ROWS** **BETWEEN** **UNBOUNDED** **PRECEDING** **AND** **UNBOUNDED** **FOLLOWING**) **AS** *first\_category*,

**LAST\_VALUE**(*p*.category) **OVER** (**PARTITION** **BY** c.customer\_id **ORDER** **BY** o.order\_date **ROWS** **BETWEEN** **UNBOUNDED** **PRECEDING** **AND** **UNBOUNDED** **FOLLOWING**) **AS** *last\_category*

**FROM** training\_ecom.customers *c*

**JOIN** training\_ecom.orders *o* **ON** *c*.customer\_id = *o*.customer\_id

**JOIN** training\_ecom.order\_items *oi* **ON** *o*.order\_id = *oi*.order\_id

**JOIN** training\_ecom.products *p* **ON** *oi*.product\_id = *p*.product\_id

**WHERE** *o*.status != 'cancelled'

**ORDER** **BY** *c*.customer\_id;

