### **1 New Customers Acquired in June 2023**

**Business Problem:**  
The marketing team ran a campaign in June 2023 and wants to see how many new customers signed up during that period.

**Fields to Retrieve:**

* PARTY\_ID
* FIRST\_NAME
* LAST\_NAME
* EMAIL
* PHONE
* ENTRY\_DATE

select p.PARTY\_ID, p.FIRST\_NAME, p.LAST\_NAME, cm.INFO\_STRING as Email, tn.CONTACT\_NUMBER as Phone, pt.CREATED\_DATE as Entry\_Date

from person p

join party pt on p.PARTY\_ID = pt.PARTY\_ID

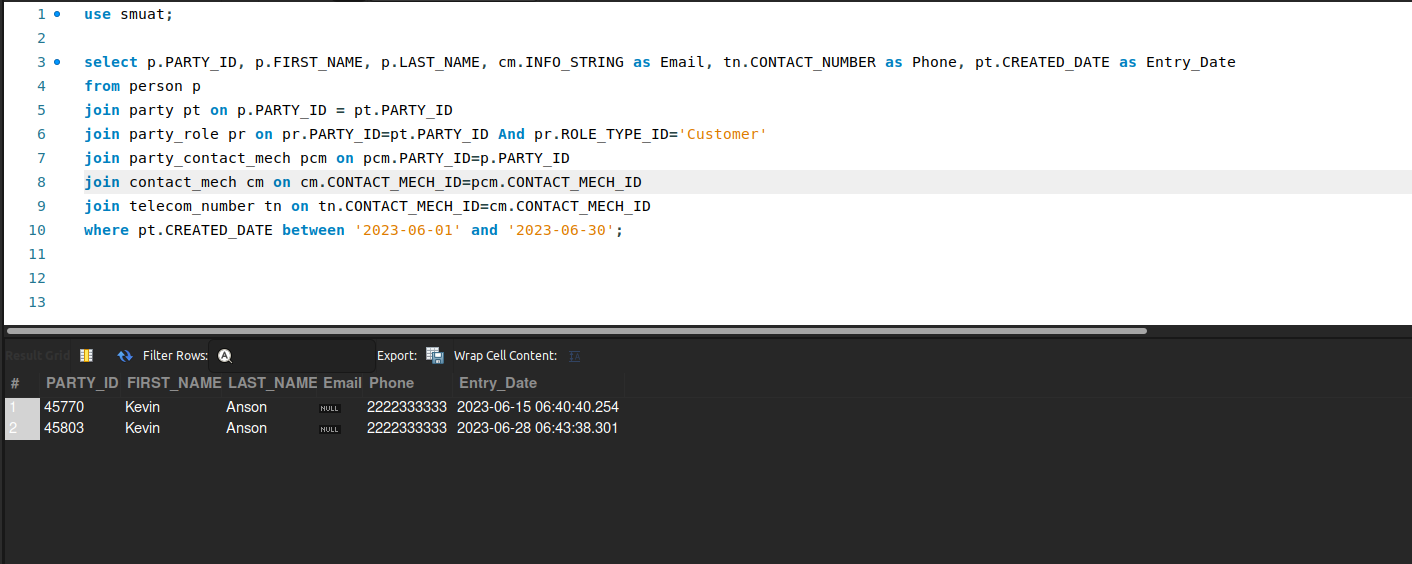
join party\_role pr on pr.PARTY\_ID=pt.PARTY\_ID And pr.ROLE\_TYPE\_ID='Customer'

join party\_contact\_mech pcm on pcm.PARTY\_ID=p.PARTY\_ID

join contact\_mech cm on cm.CONTACT\_MECH\_ID=pcm.CONTACT\_MECH\_ID

join telecom\_number tn on tn.CONTACT\_MECH\_ID=cm.CONTACT\_MECH\_ID

where pt.CREATED\_DATE between '2023-06-01' and '2023-06-30';



### **2 List All Active Physical Products**

**Business Problem:**  
Merchandising teams often need a list of all physical products to manage logistics, warehousing, and shipping.

**Fields to Retrieve:**

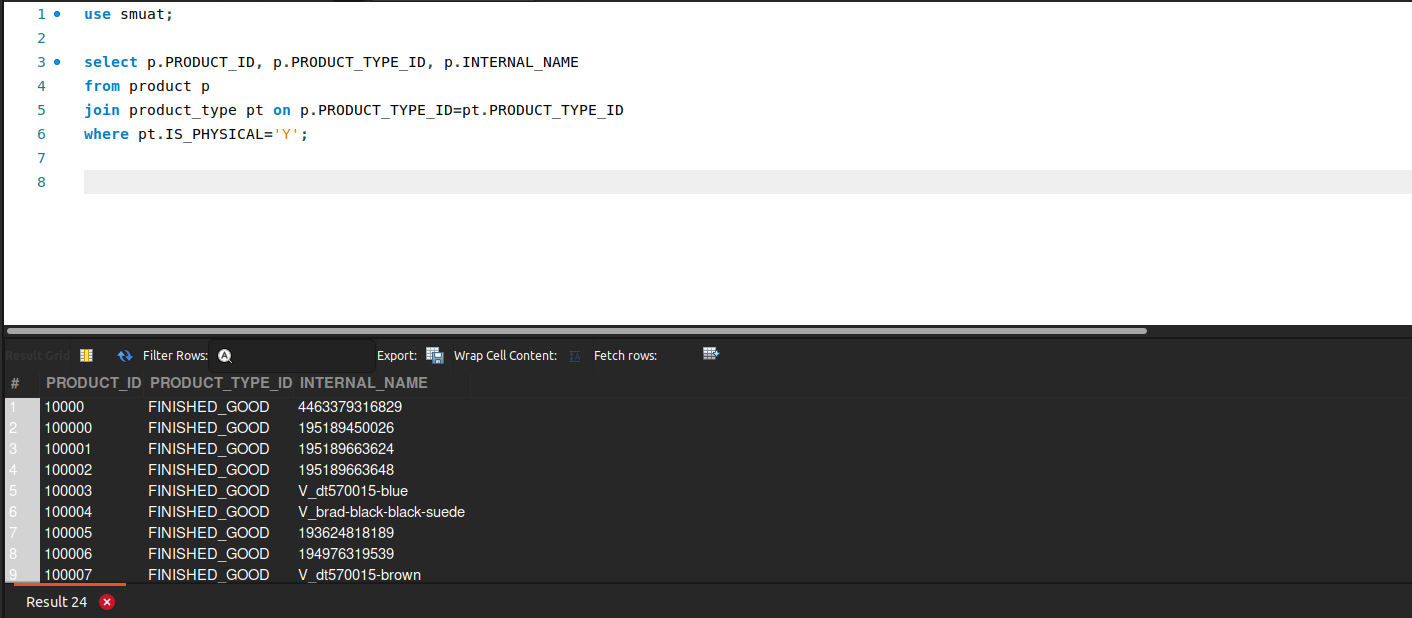
* PRODUCT\_ID
* PRODUCT\_TYPE\_ID
* INTERNAL\_NAME

select p.PRODUCT\_ID, p.PRODUCT\_TYPE\_ID, p.INTERNAL\_NAME

from product p

join product\_type pt on p.PRODUCT\_TYPE\_ID=pt.PRODUCT\_TYPE\_ID

where pt.IS\_PHYSICAL='Y';



### **3 Products Missing NetSuite ID**

**Business Problem:**  
A product cannot sync to NetSuite unless it has a valid NetSuite ID. The OMS needs a list of all products that still need to be created or updated in NetSuite.

**Fields to Retrieve:**

* PRODUCT\_ID
* INTERNAL\_NAME
* PRODUCT\_TYPE\_ID
* NETSUITE\_ID (or similar field indicating the NetSuite ID; may be NULL or empty if missing)

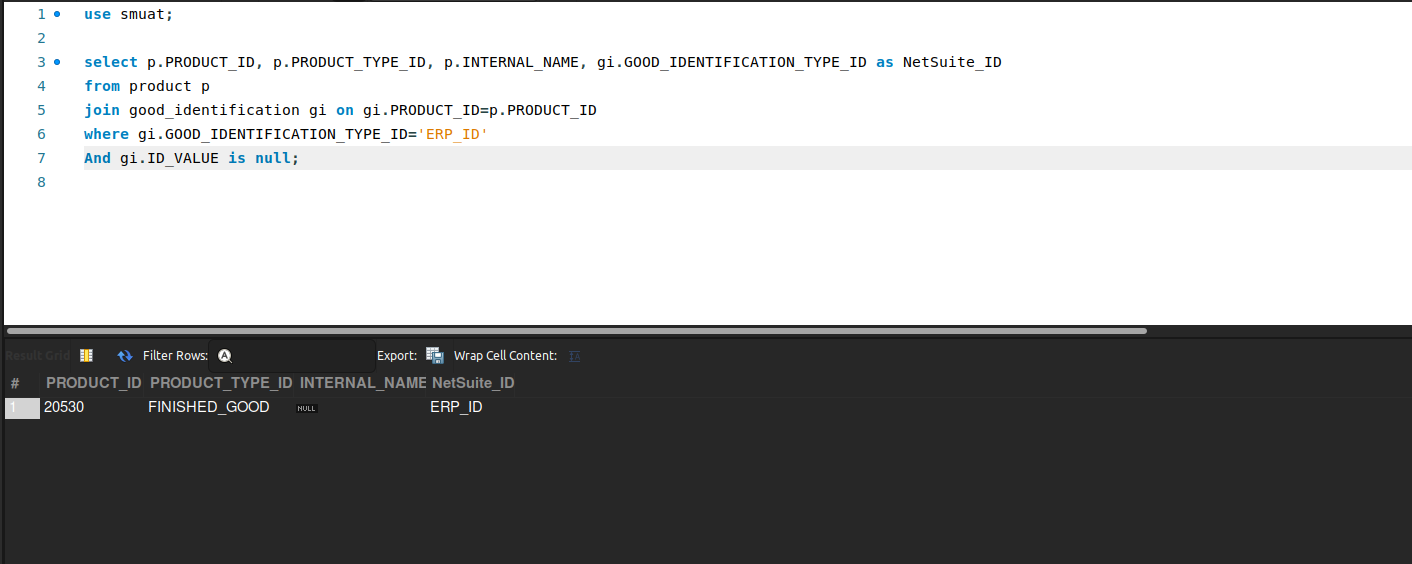
select p.PRODUCT\_ID, p.PRODUCT\_TYPE\_ID, p.INTERNAL\_NAME, gi.GOOD\_IDENTIFICATION\_TYPE\_ID as NetSuite\_ID

from product p

join good\_identification gi on gi.PRODUCT\_ID=p.PRODUCT\_ID

where gi.GOOD\_IDENTIFICATION\_TYPE\_ID='ERP\_ID'

And gi.ID\_VALUE is null;



### **4 Product IDs Across Systems**

**Business Problem:**  
To sync an order or product across multiple systems (e.g., Shopify, HotWax, ERP/NetSuite), the OMS needs to know each system’s unique identifier for that product. This query retrieves the Shopify ID, HotWax ID, and ERP ID (NetSuite ID) for all products.

**Fields to Retrieve:**

* PRODUCT\_ID (internal OMS ID)
* SHOPIFY\_ID
* HOTWAX\_ID
* ERP\_ID or NETSUITE\_ID (depending on naming)

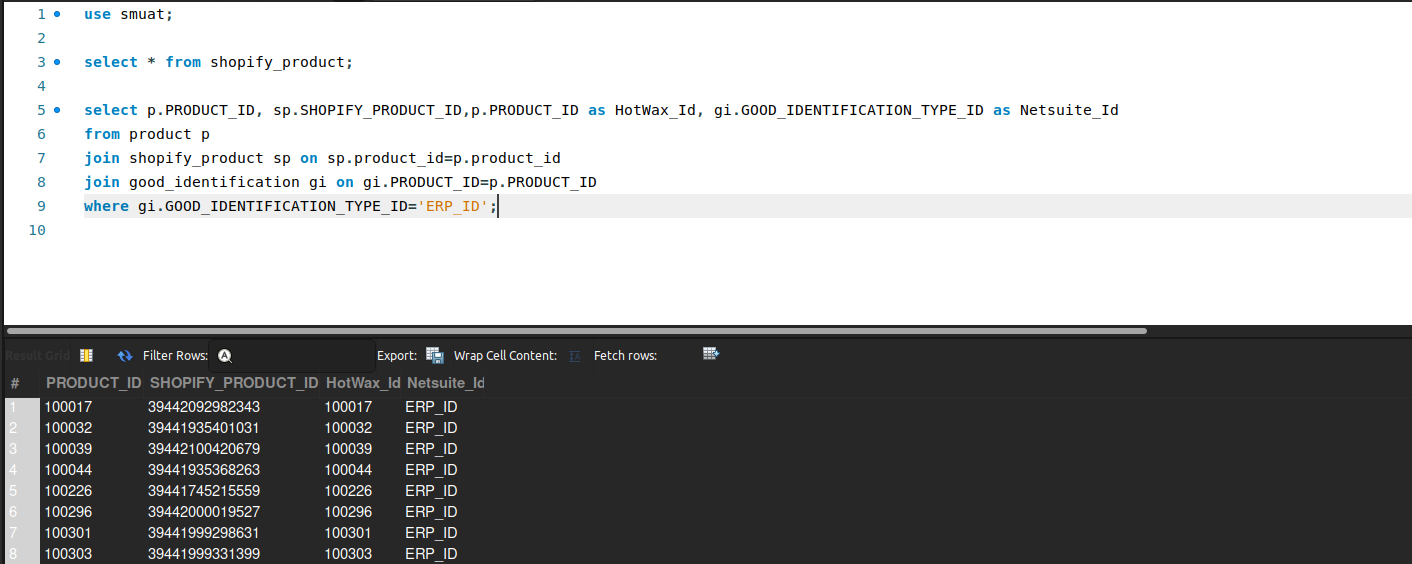
select p.PRODUCT\_ID, sp.SHOPIFY\_PRODUCT\_ID,p.PRODUCT\_ID as HotWax\_Id, gi.GOOD\_IDENTIFICATION\_TYPE\_ID as Netsuite\_Id

from product p

join shopify\_product sp on sp.product\_id=p.product\_id

join good\_identification gi on gi.PRODUCT\_ID=p.PRODUCT\_ID

where gi.GOOD\_IDENTIFICATION\_TYPE\_ID='ERP\_ID';



### **5 Completed Orders in August 2023**

**Business Problem:**  
After running similar reports for a previous month, you now need all completed orders in August 2023 for analysis.

**Fields to Retrieve:**

* PRODUCT\_ID
* PRODUCT\_TYPE\_ID
* PRODUCT\_STORE\_ID
* TOTAL\_QUANTITY
* INTERNAL\_NAME
* FACILITY\_ID
* EXTERNAL\_ID
* FACILITY\_TYPE\_ID
* ORDER\_HISTORY\_ID
* ORDER\_ID
* ORDER\_ITEM\_SEQ\_ID
* SHIP\_GROUP\_SEQ\_ID

select oh.ORDER\_ID, p.PRODUCT\_ID, p.PRODUCT\_TYPE\_ID,

oh.PRODUCT\_STORE\_ID,sum(oi.QUANTITY) as Total\_Quantity,p.INTERNAL\_NAME,

oi.SHIP\_GROUP\_SEQ\_ID,oi.ORDER\_ITEM\_SEQ\_ID,f.FACILITY\_TYPE\_ID,oh.ORIGIN\_FACILITY\_ID, odh.ORDER\_HISTORY\_ID

from order\_header oh

join order\_item oi on oi.ORDER\_ID=oh.ORDER\_ID

join product p on p.product\_id=oi.PRODUCT\_ID

join facility f on f.facility\_id=oh.ORIGIN\_FACILITY\_ID

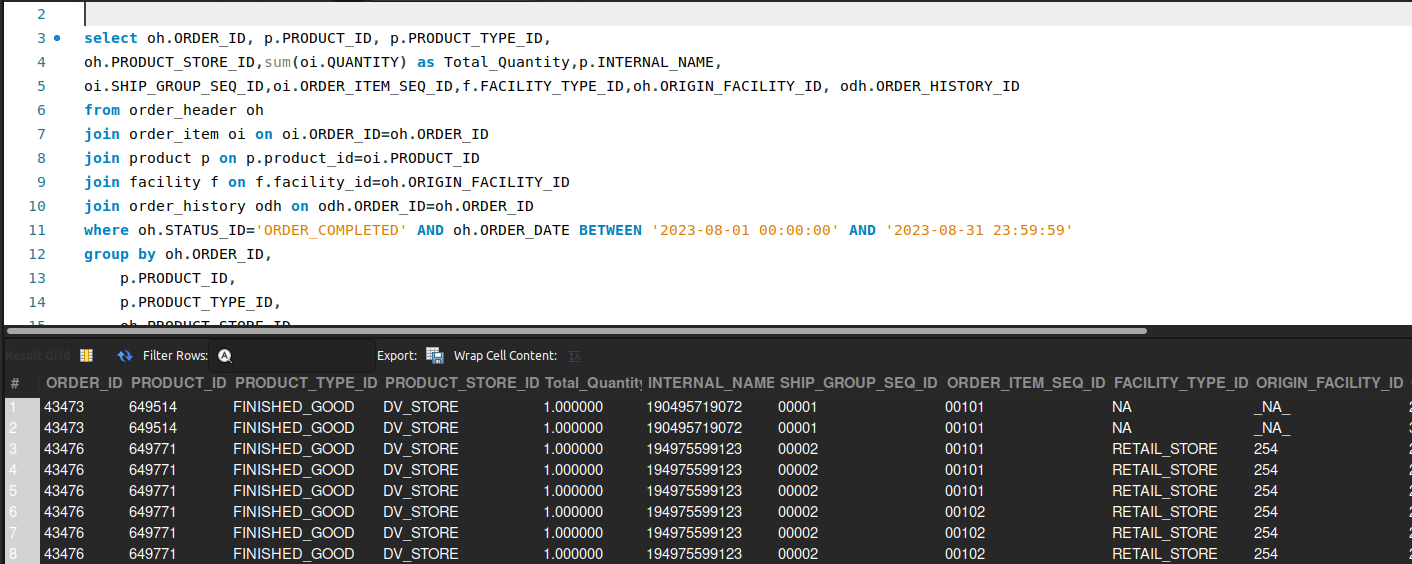
join order\_history odh on odh.ORDER\_ID=oh.ORDER\_ID

where oh.STATUS\_ID='ORDER\_COMPLETED' AND oh.ORDER\_DATE BETWEEN '2023-08-01' AND '2023-08-31'

group by oh.ORDER\_ID, p.PRODUCT\_ID, p.PRODUCT\_TYPE\_ID, oh.PRODUCT\_STORE\_ID,

p.INTERNAL\_NAME, oi.SHIP\_GROUP\_SEQ\_ID, oi.ORDER\_ITEM\_SEQ\_ID,

f.FACILITY\_TYPE\_ID, oh.ORIGIN\_FACILITY\_ID, odh.ORDER\_HISTORY\_ID;



### **7 Newly Created Sales Orders and Payment Methods**

**Business Problem:**  
Finance teams need to see new orders and their payment methods for reconciliation and fraud checks.

**Fields to Retrieve:**

* ORDER\_ID
* TOTAL\_AMOUNT
* PAYMENT\_METHOD
* Shopify Order ID (if applicable)

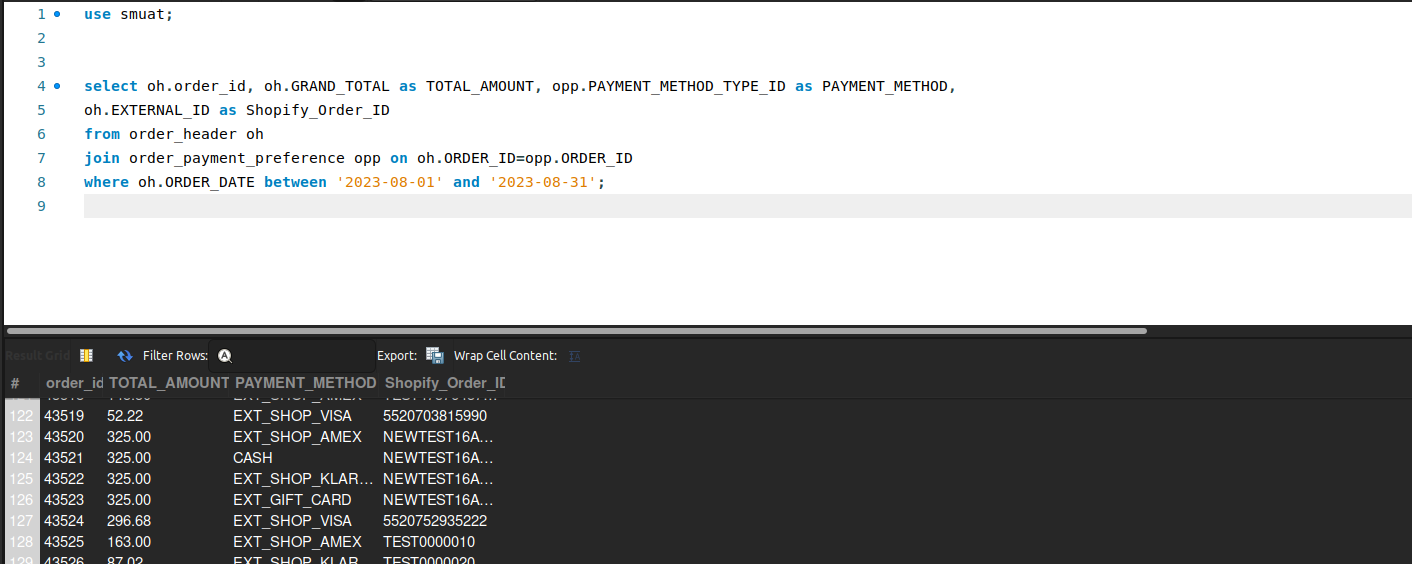
select oh.order\_id, oh.GRAND\_TOTAL as TOTAL\_AMOUNT, opp.PAYMENT\_METHOD\_TYPE\_ID as PAYMENT\_METHOD,

oh.EXTERNAL\_ID as Shopify\_Order\_ID

from order\_header oh

join order\_payment\_preference opp on oh.ORDER\_ID=opp.ORDER\_ID

where oh.ORDER\_DATE between '2023-08-01' and '2023-08-31';



### **8 Payment Captured but Not Shipped**

**Business Problem:**  
Finance teams want to ensure revenue is recognized properly. If payment is captured but no shipment has occurred, it warrants further review.

**Fields to Retrieve:**

* ORDER\_ID
* ORDER\_STATUS
* PAYMENT\_STATUS
* SHIPMENT\_STATUS

select oh.ORDER\_ID, oh.STATUS\_ID as Order\_Status, opp.STATUS\_ID as Payment\_Status, sh.STATUS\_ID as Shipment\_Status

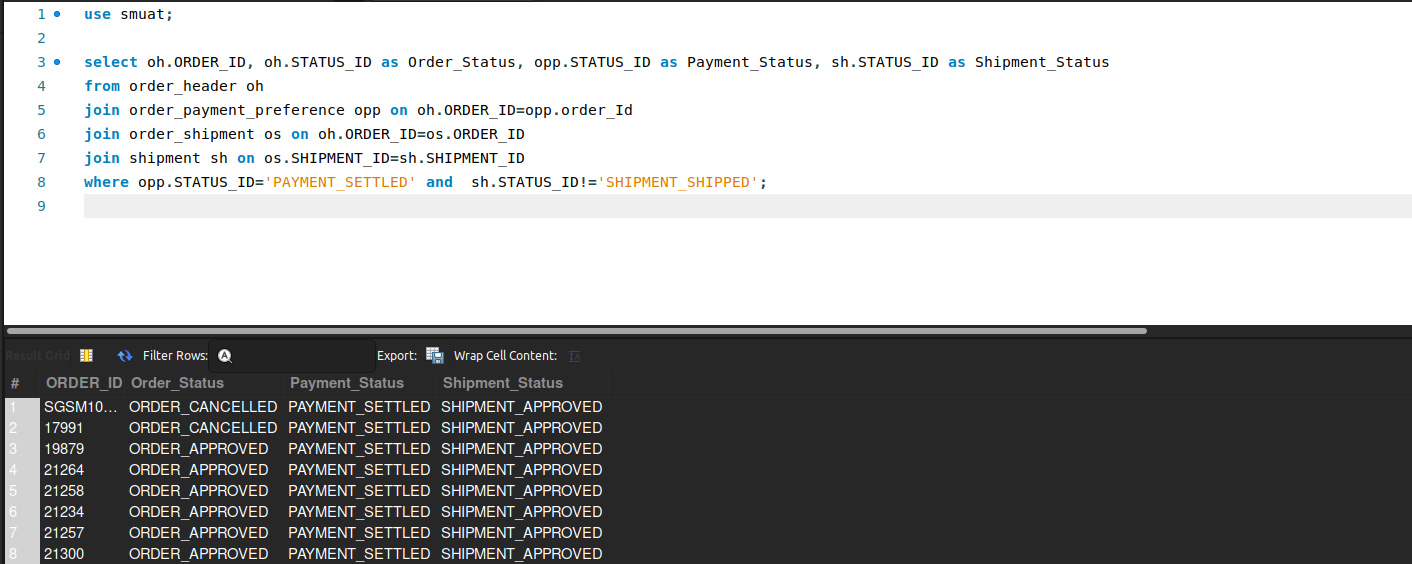
from order\_header oh

join order\_payment\_preference opp on oh.ORDER\_ID=opp.order\_Id

join order\_shipment os on oh.ORDER\_ID=os.ORDER\_ID

join shipment sh on os.SHIPMENT\_ID=sh.SHIPMENT\_ID

where opp.STATUS\_ID='PAYMENT\_SETTLED' and sh.STATUS\_ID!='SHIPMENT\_SHIPPED';



### **9 Orders Completed Hourly**

**Business Problem:**  
Operations teams may want to see how orders complete across the day to schedule staffing.

**Fields to Retrieve:**

* TOTAL ORDERS
* HOUR

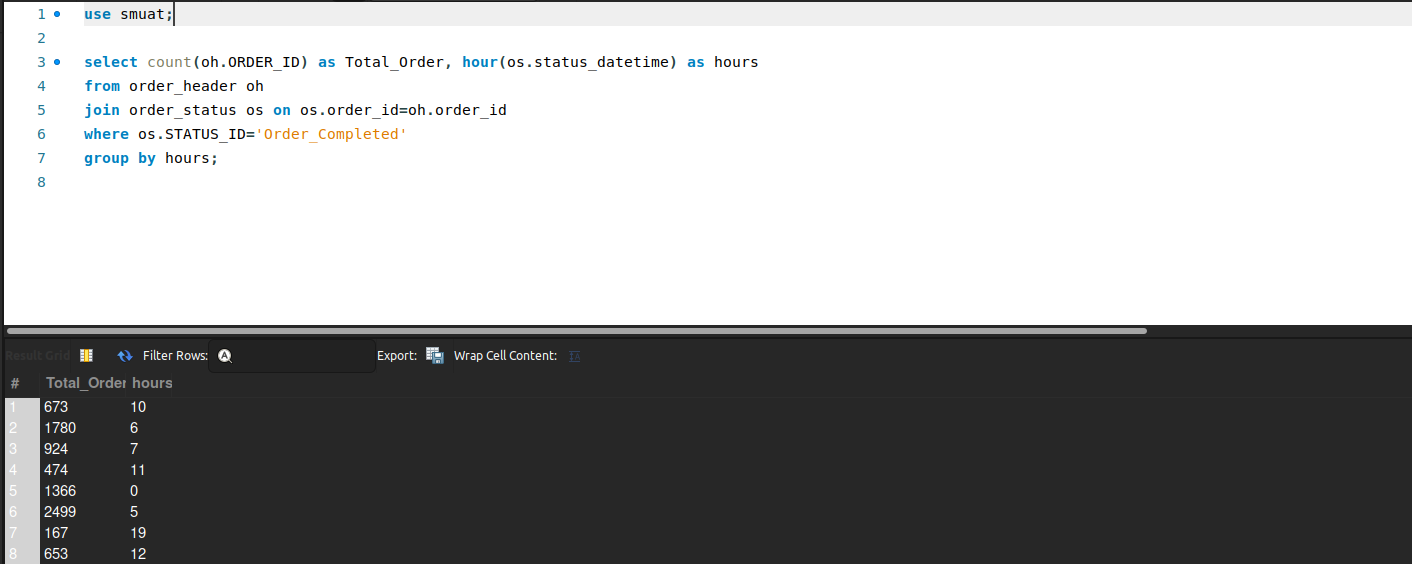
select count(oh.ORDER\_ID) as Total\_Order, hour(os.status\_datetime) as hours

from order\_header oh

join order\_status os on os.order\_id=oh.order\_id

where os.STATUS\_ID='Order\_Completed'

group by hours;



### **10 BOPIS Orders Revenue (Last Year)**

**Business Problem:**  
**BOPIS** (Buy Online, Pickup In Store) is a key retail strategy. Finance wants to know the revenue from BOPIS orders for the previous year.

**Fields to Retrieve:**

* TOTAL ORDERS
* TOTAL REVENUE

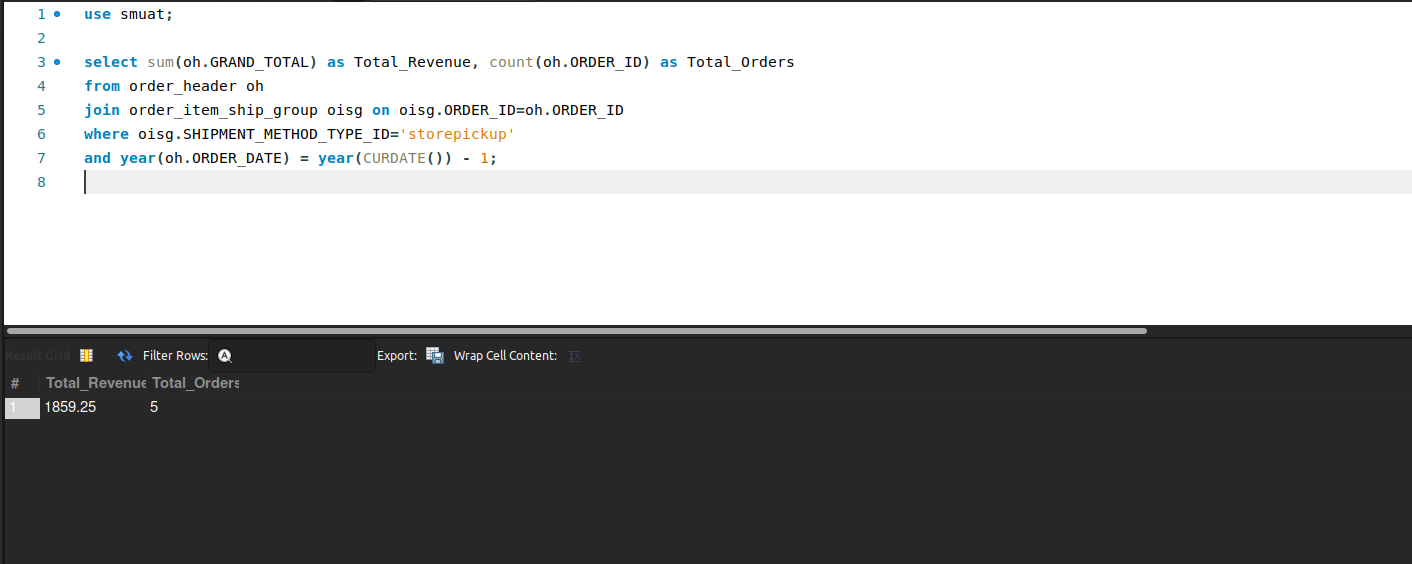
select sum(oh.GRAND\_TOTAL) as Total\_Revenue, count(oh.ORDER\_ID) as Total\_Orders

from order\_header oh

join order\_item\_ship\_group oisg on oisg.ORDER\_ID=oh.ORDER\_ID

where oisg.SHIPMENT\_METHOD\_TYPE\_ID='storepickup'

and year(oh.ORDER\_DATE) = year(CURDATE()) - 1;



### **11 Canceled Orders (Last Month)**

**Business Problem:**  
The merchandising team needs to know how many orders were canceled in the previous month and their reasons.

**Fields to Retrieve:**

* TOTAL ORDERS
* CANCELATION REASON

select count(distinct oh.order\_id) as Total\_Orders, os.CHANGE\_REASON as Cancelation\_Reason

from order\_header oh

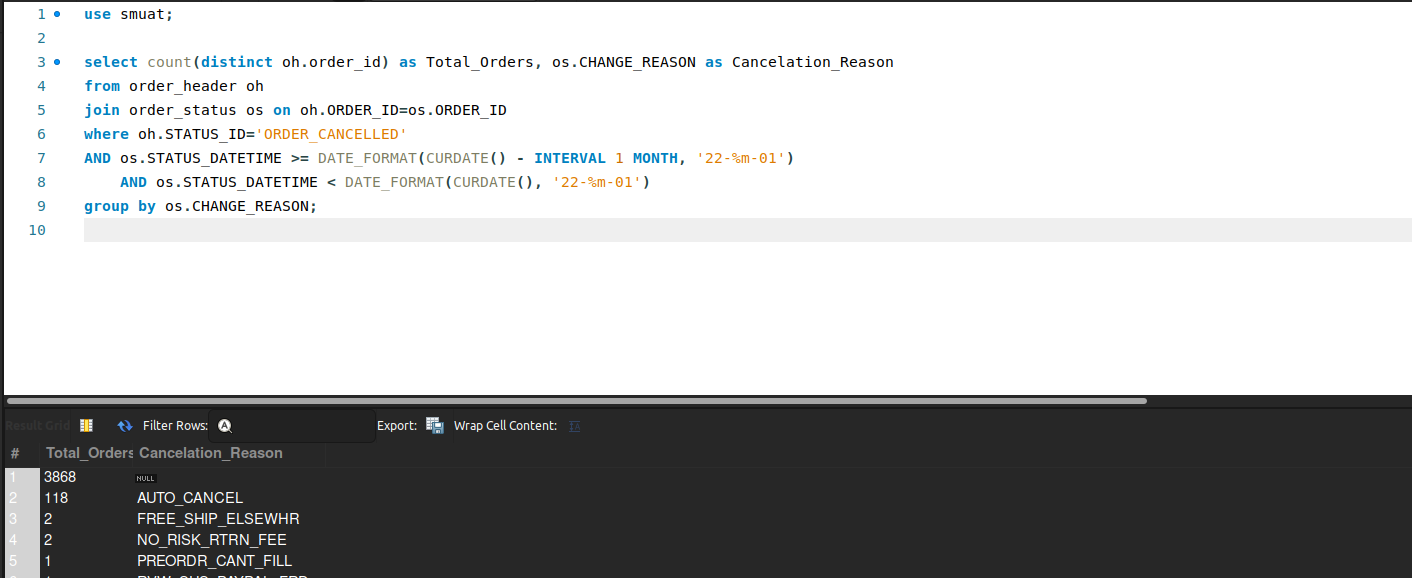
join order\_status os on oh.ORDER\_ID=os.ORDER\_ID

where oh.STATUS\_ID='ORDER\_CANCELLED'

AND os.STATUS\_DATETIME >= DATE\_FORMAT(CURDATE() - INTERVAL 1 MONTH, '22-%m-01')

AND os.STATUS\_DATETIME < DATE\_FORMAT(CURDATE(), '22-%m-01')

group by os.CHANGE\_REASON;



### **12 Product Threshold Value**

**Business Problem** The retailer has set a threshild value for products that are sold online, in order to avoid over selling.

**Fields to Retrieve:**

* PRODUCT ID
* THRESHOLD

select PRODUCT\_STORE\_ID, REPLENISH\_THRESHOLD from product\_store\_fin\_act\_setting;

