### **1 Completed Sales Orders (Physical Items)**

Business Problem:  
Merchants need to track only physical items (requiring shipping and fulfillment) for logistics and shipping-cost analysis.

Fields to Retrieve:

* ORDER\_ID
* ORDER\_ITEM\_SEQ\_ID
* PRODUCT\_ID
* PRODUCT\_TYPE\_ID
* SALES\_CHANNEL\_ENUM\_ID
* ORDER\_DATE
* ENTRY\_DATE
* STATUS\_ID
* STATUS\_DATETIME
* ORDER\_TYPE\_ID
* PRODUCT\_STORE\_ID

select distinct oh.ORDER\_ID, oi.ORDER\_ITEM\_SEQ\_ID, oi.PRODUCT\_ID, p.PRODUCT\_TYPE\_ID,

oh.SALES\_CHANNEL\_ENUM\_ID, oh.ORDER\_DATE, oh.ENTRY\_DATE,

oh.STATUS\_ID, os.STATUS\_DATETIME, oh.ORDER\_TYPE\_ID, oh.PRODUCT\_STORE\_ID

from order\_header oh

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

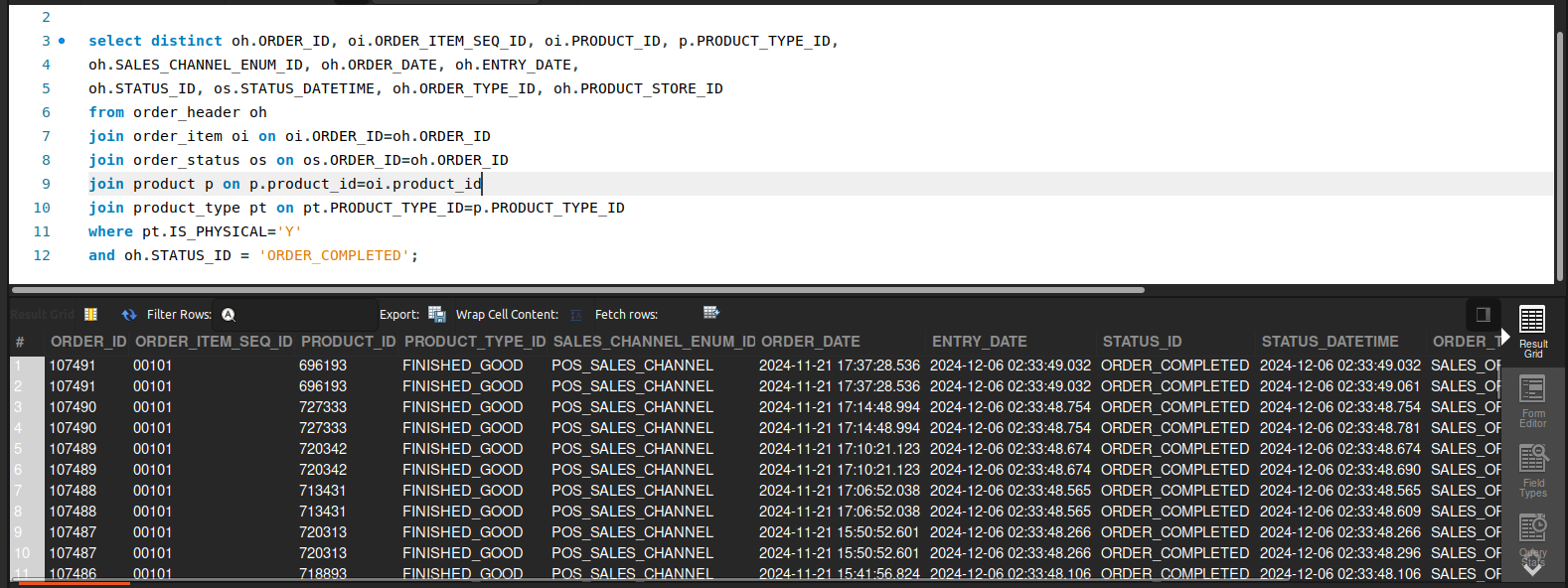
join order\_status os on os.ORDER\_ID=oh.ORDER\_ID

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

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

where pt.IS\_PHYSICAL='Y'

and oh.STATUS\_ID = 'ORDER\_COMPLETED';



### **2 Completed Return Items**

Business Problem:  
Customer service and finance often need insights into returned items to manage refunds, replacements, and inventory restocking.

Fields to Retrieve:

* RETURN\_ID
* ORDER\_ID
* PRODUCT\_STORE\_ID
* STATUS\_DATETIME
* ORDER\_NAME
* FROM\_PARTY\_ID
* RETURN\_DATE
* ENTRY\_DATE
* RETURN\_CHANNEL\_ENUM\_ID

select ri.RETURN\_ID, ri.order\_id, oh.PRODUCT\_STORE\_ID, rs.STATUS\_DATETIME,

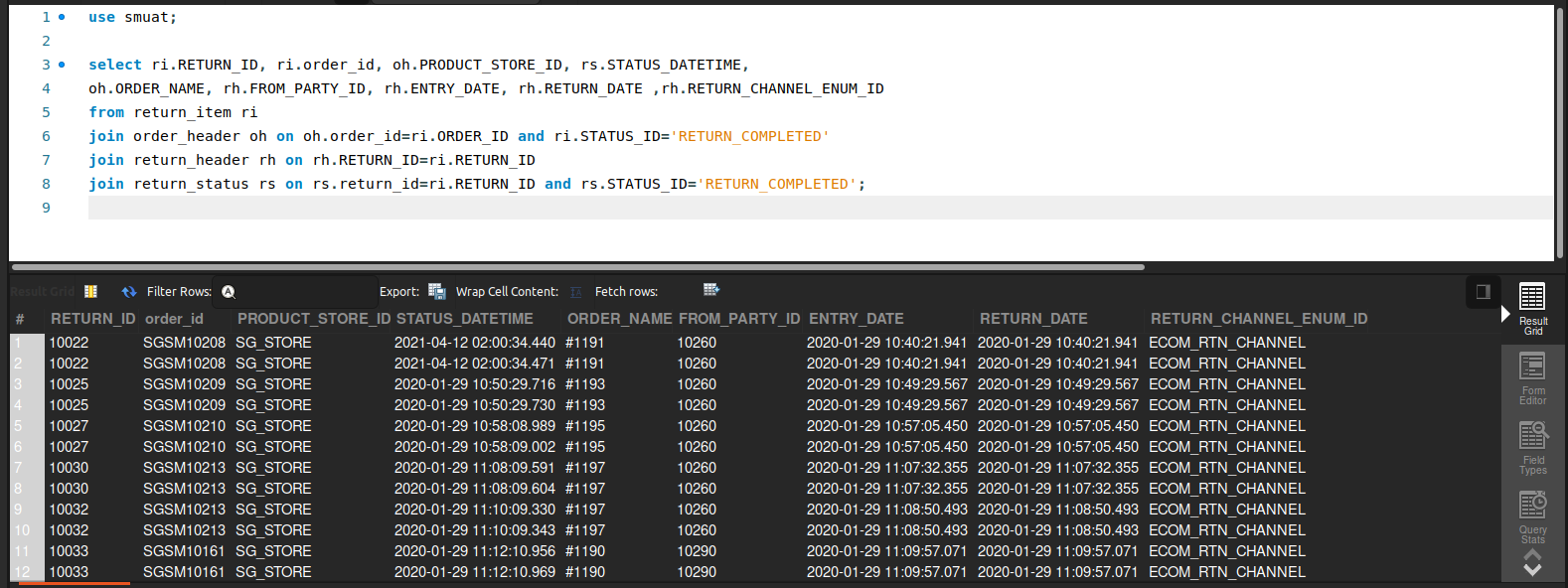
oh.ORDER\_NAME, rh.FROM\_PARTY\_ID, rh.ENTRY\_DATE, rh.RETURN\_DATE ,rh.RETURN\_CHANNEL\_ENUM\_ID

from return\_item ri

join order\_header oh on oh.order\_id=ri.ORDER\_ID and ri.STATUS\_ID='RETURN\_COMPLETED'

join return\_header rh on rh.RETURN\_ID=ri.RETURN\_ID

join return\_status rs on rs.return\_id=ri.RETURN\_ID and rs.STATUS\_ID='RETURN\_COMPLETED';￼



3)

### **3 Single-Return Orders (Last Month)**

Business Problem:  
The mechandising team needs a list of orders that only have one return.

Fields to Retrieve:

* PARTY\_ID
* FIRST\_NAME

select rh.FROM\_PARTY\_ID as PARTY\_ID, p.FIRST\_NAME

from return\_header rh

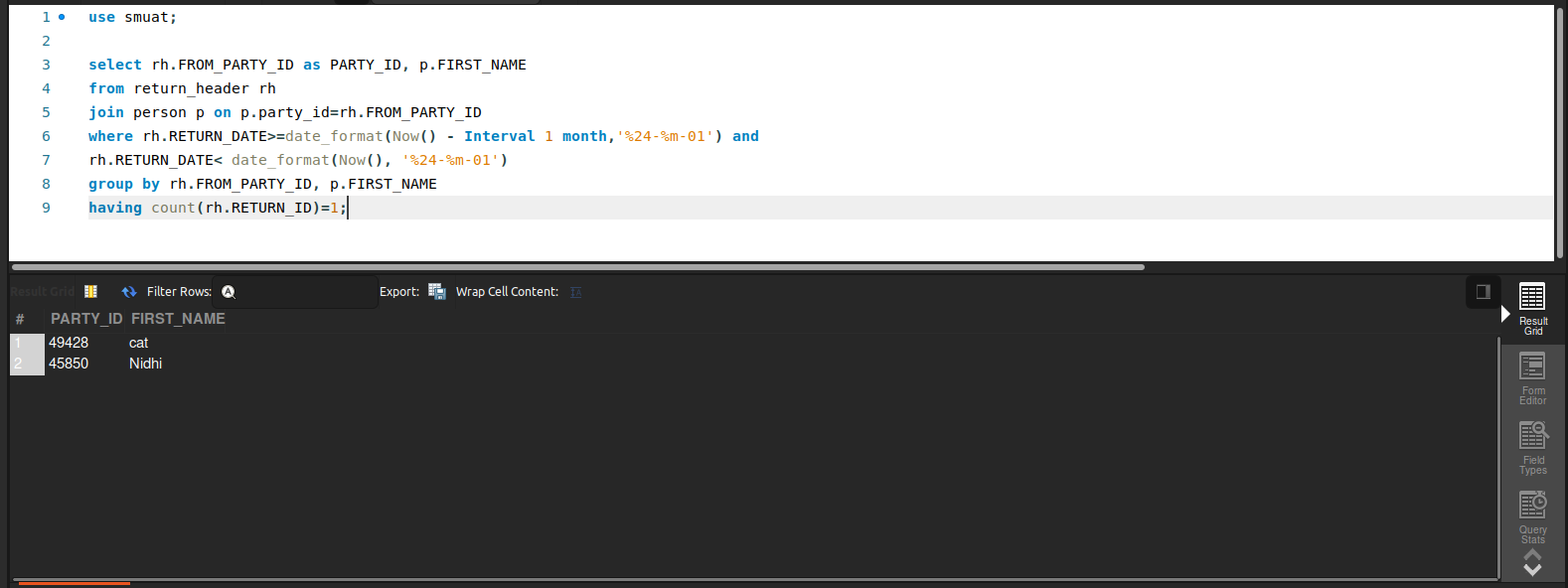
join person p on p.party\_id=rh.FROM\_PARTY\_ID

where rh.RETURN\_DATE>=date\_format(Now() - Interval 1 month,'%24-%m-01') and

rh.RETURN\_DATE< date\_format(Now(), '%24-%m-01')

group by rh.FROM\_PARTY\_ID, p.FIRST\_NAME

having count(rh.RETURN\_ID)=1;



4)

### **4 Returns and Appeasements**

Business Problem:  
The retailer needs the total amount of items, were returned as well as how many appeasements were issued.

Fields to Retrieve:

* TOTAL RETURNS
* RETURN $ TOTAL
* TOTAL APPEASEMENTS
* APPEASEMENTS $ TOTAL

select count(rh.RETURN\_ID) as TOTAL\_RETURNS, sum(ri.RETURN\_PRICE \* ri.RETURN\_QUANTITY) as RETURN\_$\_TOTAL,

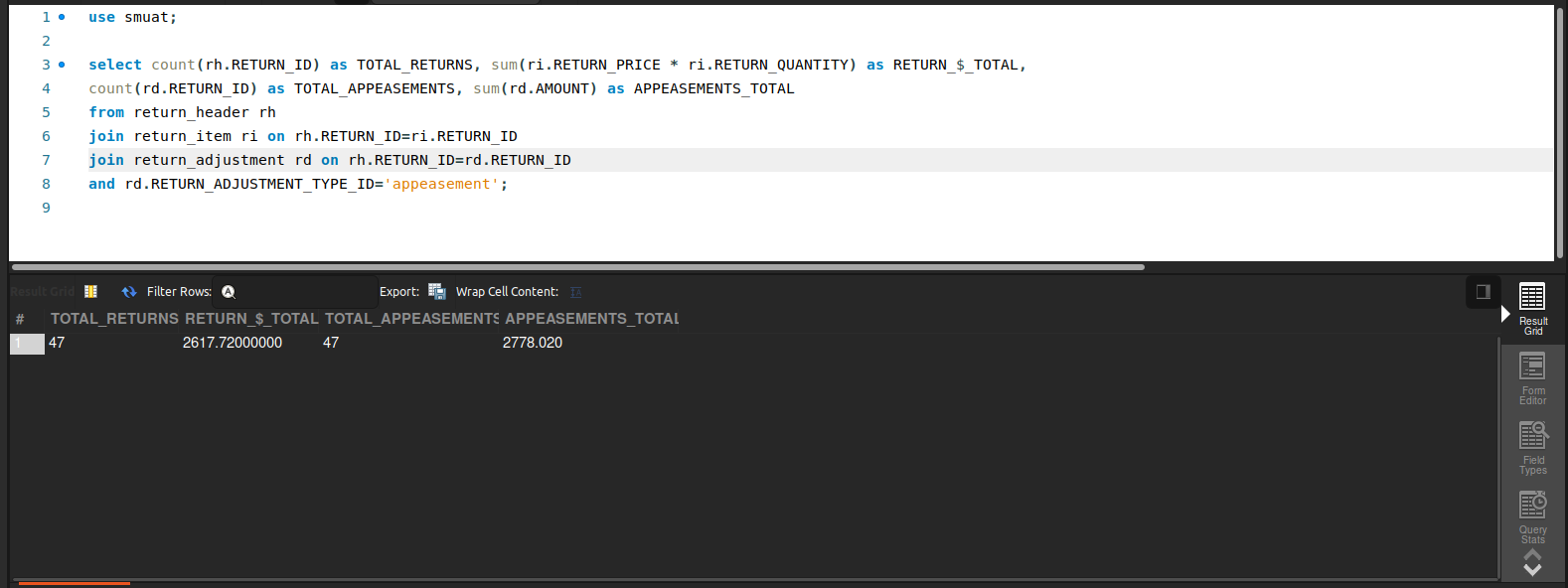
count(rd.RETURN\_ID) as TOTAL\_APPEASEMENTS, sum(rd.AMOUNT) as APPEASEMENTS\_TOTAL

from return\_header rh

join return\_item ri on rh.RETURN\_ID=ri.RETURN\_ID

join return\_adjustment rd on rh.RETURN\_ID=rd.RETURN\_ID

and rd.RETURN\_ADJUSTMENT\_TYPE\_ID='appeasement';



### **5 Detailed Return Information**

Business Problem:  
Certain teams need granular return data (reason, date, refund amount) for analyzing return rates, identifying recurring issues, or updating policies.

Fields to Retrieve:

* RETURN\_ID
* ENTRY\_DATE
* RETURN\_ADJUSTMENT\_TYPE\_ID (refund type, store credit, etc.)
* AMOUNT
* COMMENTS
* ORDER\_ID
* ORDER\_DATE
* RETURN\_DATE
* PRODUCT\_STORE\_ID

select rh.return\_id, rh.ENTRY\_DATE, ra.return\_adjustment\_type\_id,

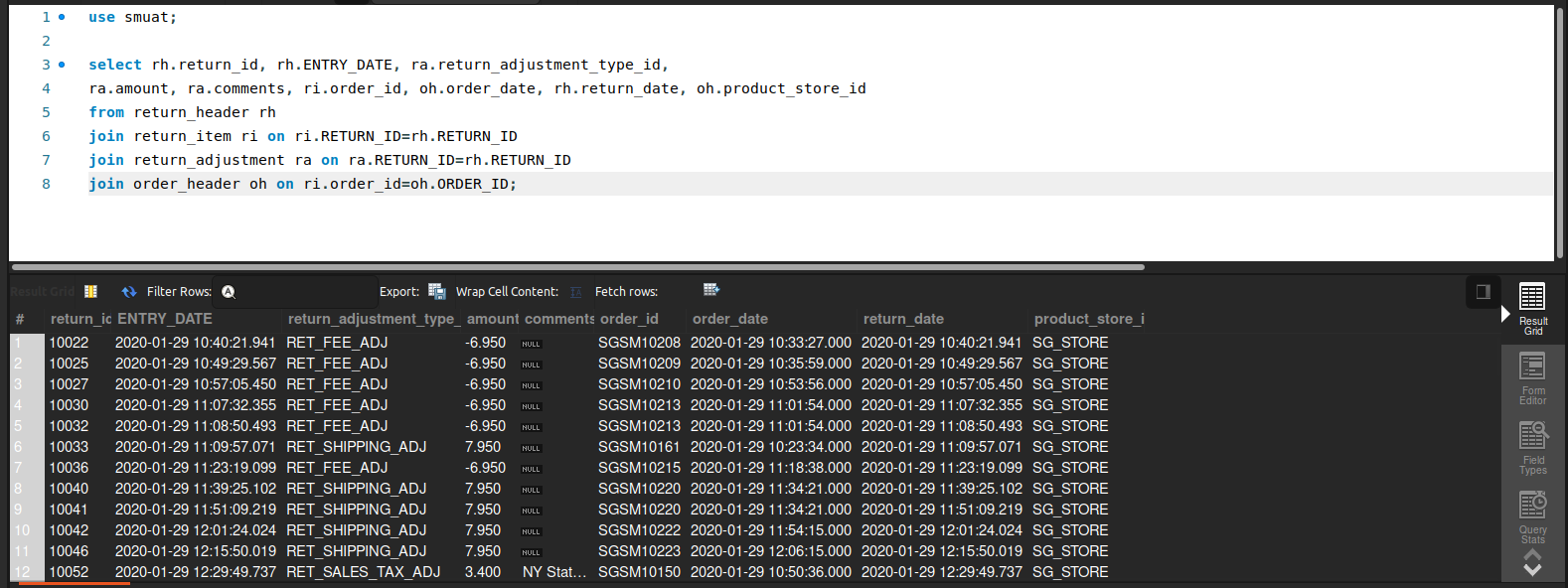
ra.amount, ra.comments, ri.order\_id, oh.order\_date, rh.return\_date, oh.product\_store\_id

from return\_header rh

join return\_item ri on ri.RETURN\_ID=rh.RETURN\_ID

join return\_adjustment ra on ra.RETURN\_ID=rh.RETURN\_ID

join order\_header oh on ri.order\_id=oh.ORDER\_ID;



### **6 Orders with Multiple Returns**

Business Problem:  
Analyzing orders with multiple returns can identify potential fraud, chronic issues with certain items, or inconsistent shipping processes.

Fields to Retrieve:

* ORDER\_ID
* RETURN\_ID
* RETURN\_DATE
* RETURN\_REASON
* RETURN\_QUANTITY

select ri.ORDER\_ID, ri.RETURN\_ID, rh.RETURN\_DATE, rs.DESCRIPTION as RETURN\_REASON,

ri.RETURN\_QUANTITY

from return\_header rh

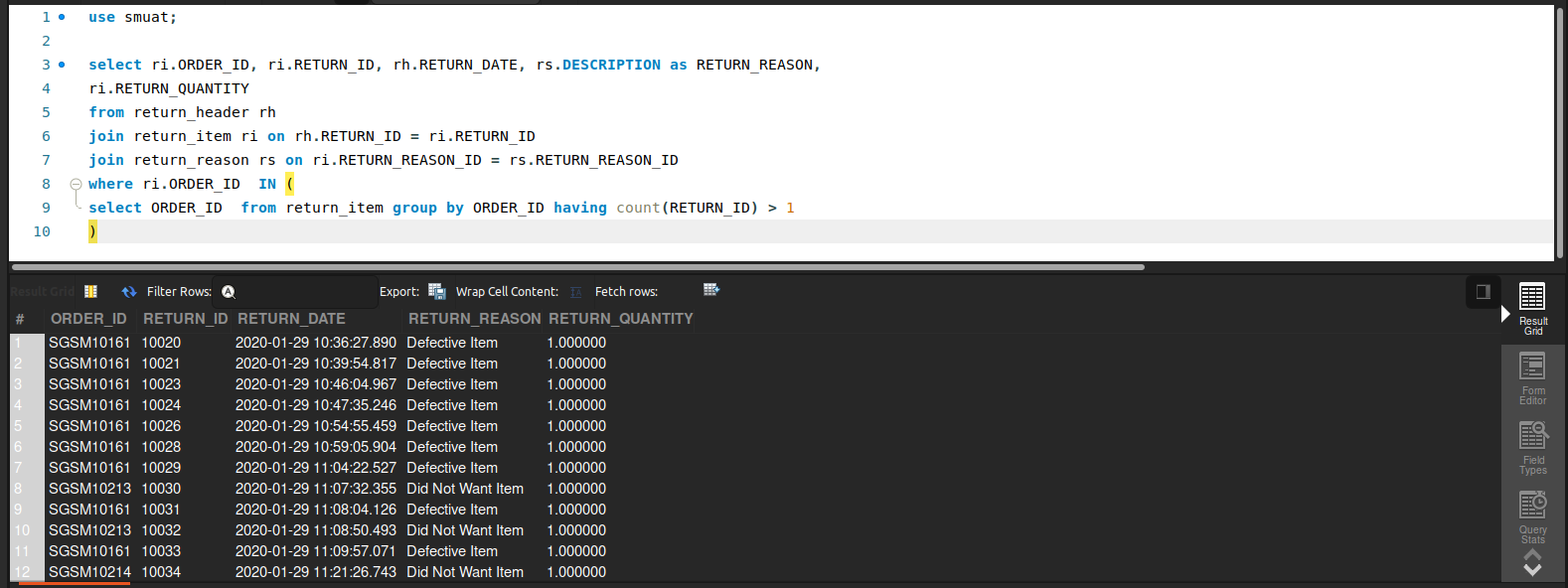
join return\_item ri on rh.RETURN\_ID = ri.RETURN\_ID

join return\_reason rs on ri.RETURN\_REASON\_ID = rs.RETURN\_REASON\_ID

where ri.ORDER\_ID IN (

select ORDER\_ID from return\_item group by ORDER\_ID having count(RETURN\_ID) > 1

)



### **7 Store with Most One-Day Shipped Orders (Last Month)**

Business Problem:  
Identify which facility (store) handled the highest volume of “one-day shipping” orders in the previous month, useful for operational benchmarking.

Fields to Retrieve:

* FACILITY\_ID
* FACILITY\_NAME
* TOTAL\_ONE\_DAY\_SHIP\_ORDERS
* REPORTING\_PERIOD

select f.facility\_id, f.facility\_name,

count(oi.order\_id) as TOTAL\_ONE\_DAY\_SHIP\_ORDERS

from facility f

join order\_item\_ship\_group oi on f.FACILITY\_ID = oi.FACILITY\_ID AND oi.SHIPMENT\_METHOD\_TYPE\_ID = 'NEXT\_DAY'

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

where oh.ORDER\_DATE >= DATE\_FORMAT(NOW() - INTERVAL 1 MONTH, '23-%m-01')

AND oh.ORDER\_DATE < DATE\_FORMAT(NOW(), '23-%m-01')

group by f.FACILITY\_ID , f.FACILITY\_NAME

order by TOTAL\_ONE\_DAY\_SHIP\_ORDERS desc

limit 1;



### **8 List of Warehouse Pickers**

Business Problem:  
Warehouse managers need a list of employees responsible for picking and packing orders to manage shifts, productivity, and training needs.

Fields to Retrieve:

* PARTY\_ID (or Employee ID)
* NAME (First/Last)
* ROLE\_TYPE\_ID (e.g., “WAREHOUSE\_PICKER”)
* FACILITY\_ID (assigned warehouse)
* STATUS (active or inactive employee)

select pr.PARTY\_ID, per.FIRST\_NAME, pr.ROLE\_TYPE\_ID, p.FACILITY\_ID, prt.STATUS\_ID

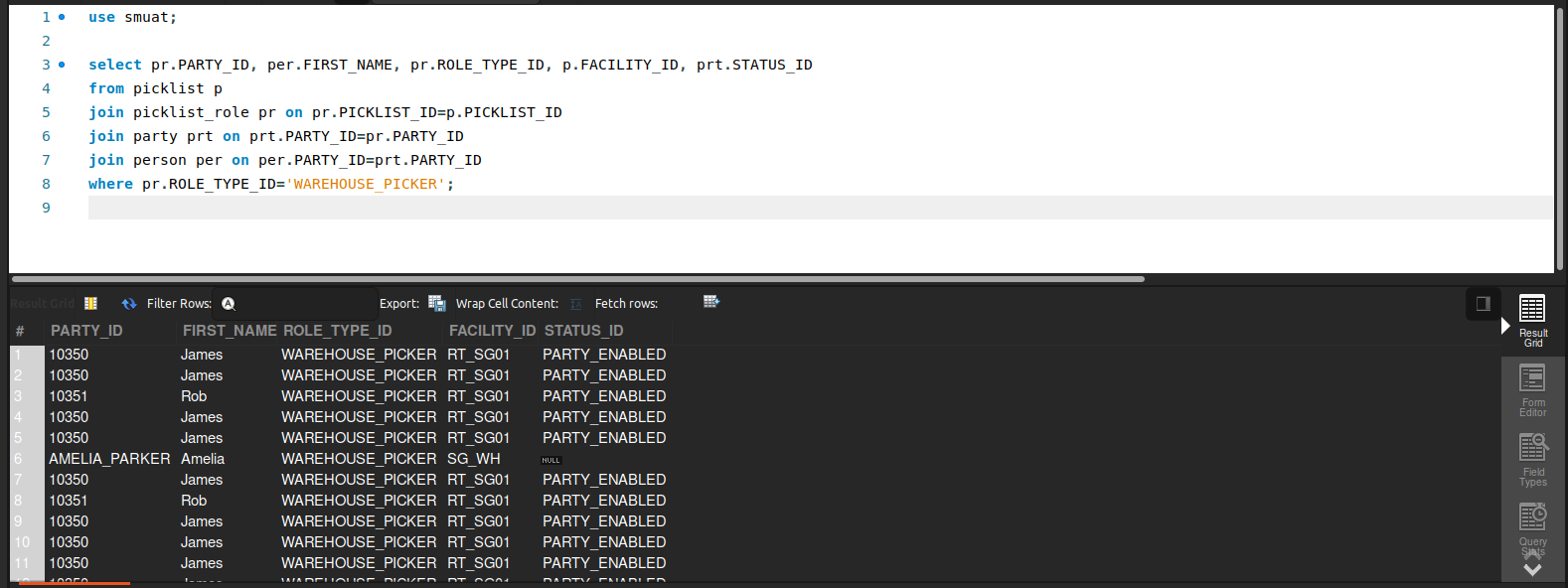
from picklist p

join picklist\_role pr on pr.PICKLIST\_ID=p.PICKLIST\_ID

join party prt on prt.PARTY\_ID=pr.PARTY\_ID

join person per on per.PARTY\_ID=prt.PARTY\_ID

where pr.ROLE\_TYPE\_ID='WAREHOUSE\_PICKER';



### **9 Total Facilities That Sell the Product**

Business Problem:  
Retailers want to see how many (and which) facilities (stores, warehouses, virtual sites) currently offer a product for sale.

Fields to Retrieve:

* PRODUCT\_ID
* PRODUCT\_NAME (or INTERNAL\_NAME)
* FACILITY\_COUNT (number of facilities selling the product)

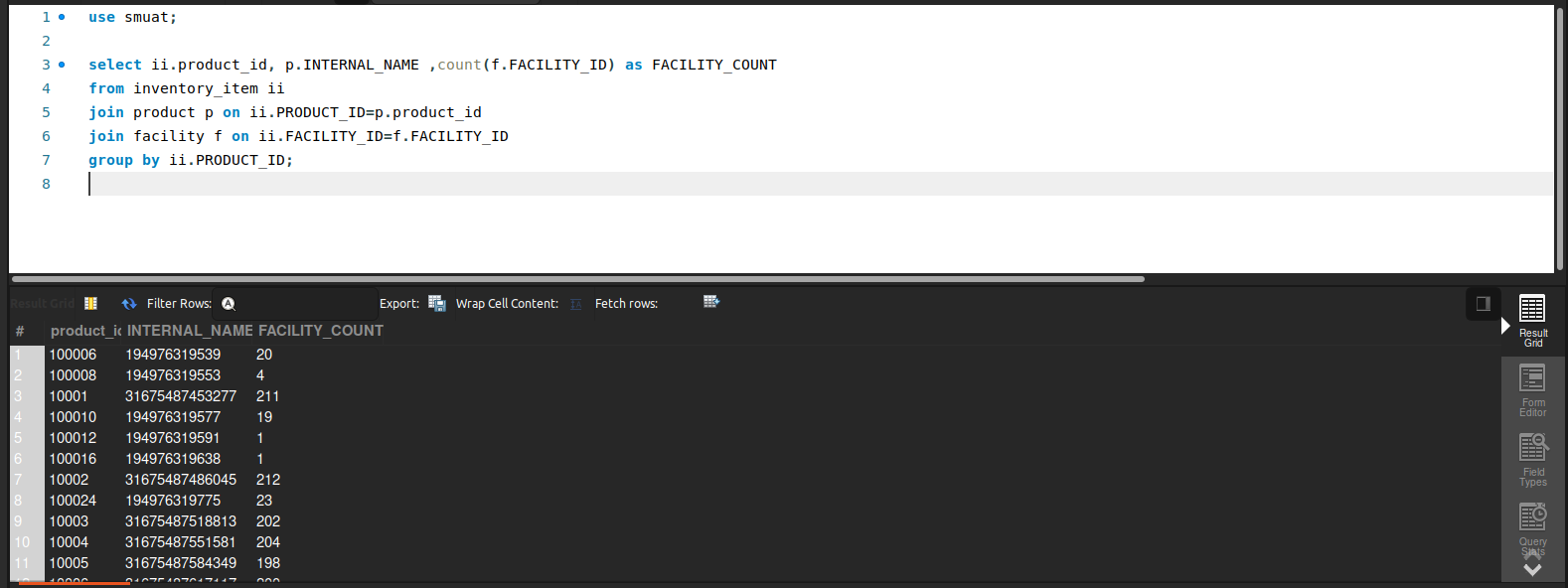
select ii.product\_id, p.INTERNAL\_NAME ,count(f.FACILITY\_ID) as FACILITY\_COUNT

from inventory\_item ii

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

join facility f on ii.FACILITY\_ID=f.FACILITY\_ID

group by ii.PRODUCT\_ID;



### **10 Total Items in Various Virtual Facilities**

Business Problem:  
Retailers need to study the relation of inventory levels of products to the type of facility it's stored at. Retrieve all inventory levels for products at locations and include the facility type Id. Do not retrieve facilities that are of type Virtual.

Fields to Retrieve:

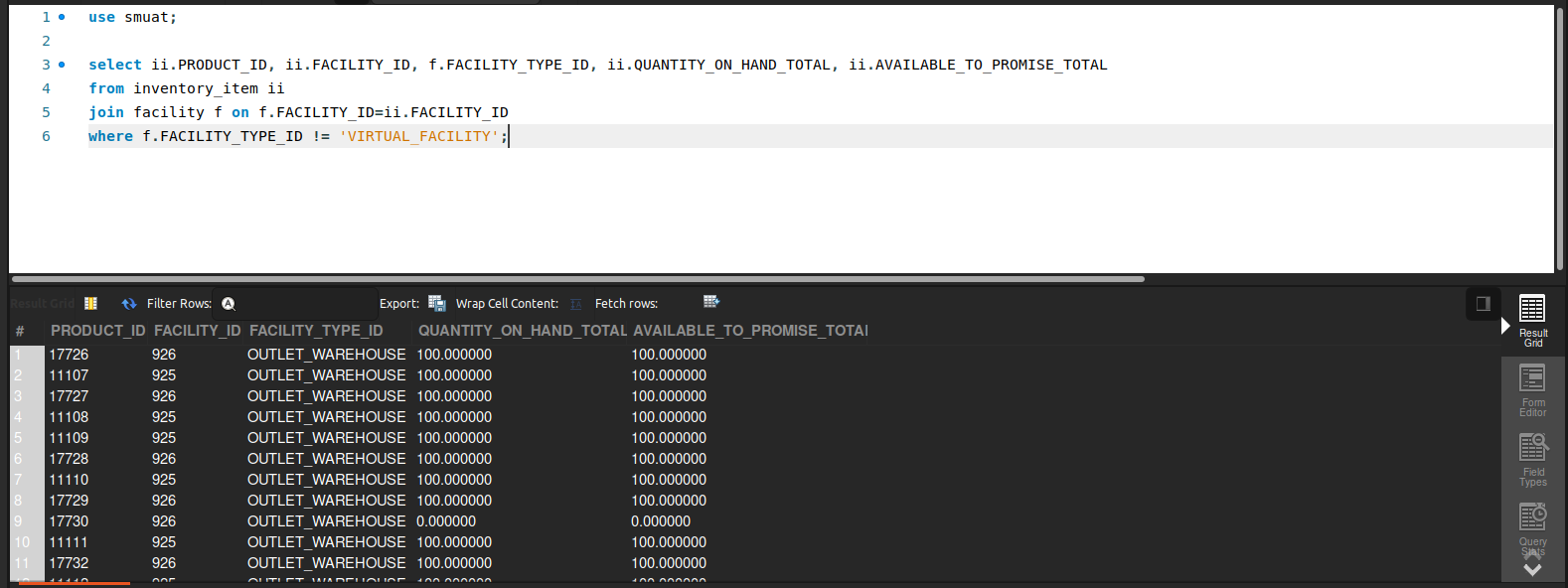
* PRODUCT\_ID
* FACILITY\_ID
* FACILITY\_TYPE\_ID
* QOH (Quantity on Hand)
* ATP (Available to Promise)

select ii.PRODUCT\_ID, ii.FACILITY\_ID, f.FACILITY\_TYPE\_ID, ii.QUANTITY\_ON\_HAND\_TOTAL, ii.AVAILABLE\_TO\_PROMISE\_TOTAL

from inventory\_item ii

join facility f on f.FACILITY\_ID=ii.FACILITY\_ID

where f.FACILITY\_TYPE\_ID != 'VIRTUAL\_FACILITY';



### **11 Transfer Orders Without Inventory Reservation**

Business Problem:  
When transferring stock between facilities, the system should reserve inventory. If it isn’t reserved, the transfer may fail or oversell.

Fields to Retrieve:

* TRANSFER\_ORDER\_ID
* FROM\_FACILITY\_ID
* TO\_FACILITY\_ID
* PRODUCT\_ID
* REQUESTED\_QUANTITY
* RESERVED\_QUANTITY
* TRANSFER\_DATE
* STATUS

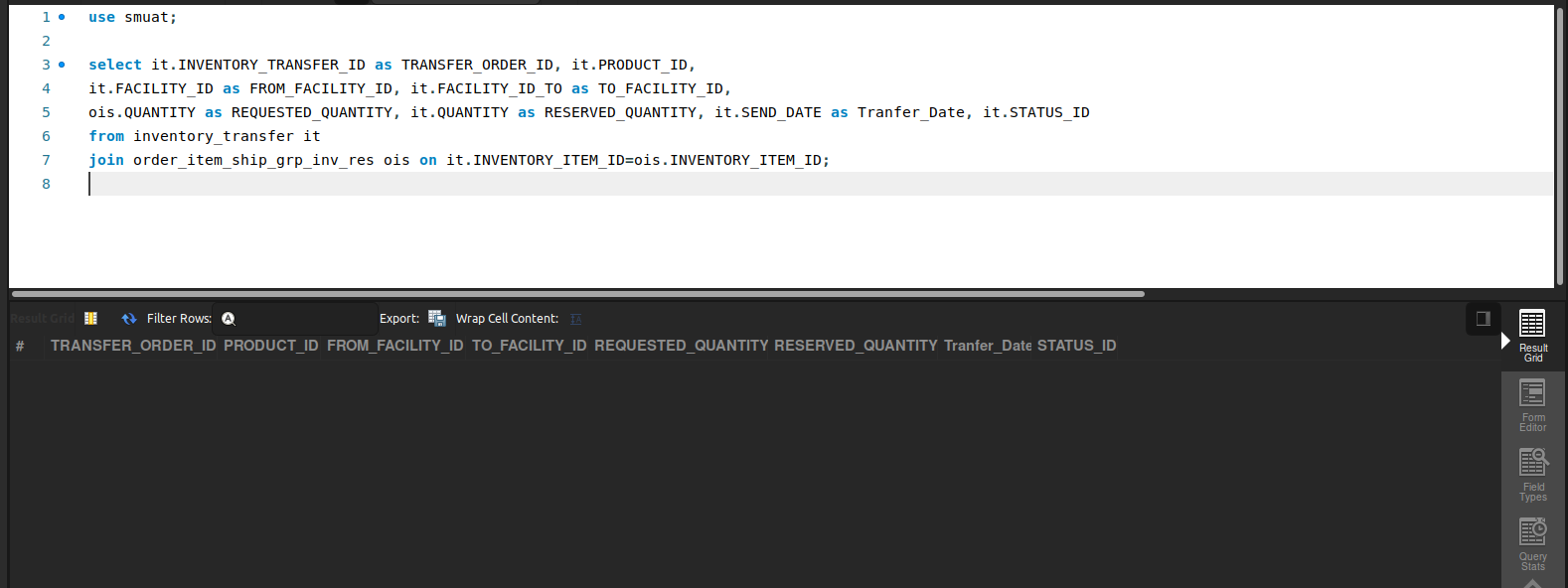
select it.INVENTORY\_TRANSFER\_ID as TRANSFER\_ORDER\_ID, it.PRODUCT\_ID,

it.FACILITY\_ID as FROM\_FACILITY\_ID, it.FACILITY\_ID\_TO as TO\_FACILITY\_ID,

ois.QUANTITY as REQUESTED\_QUANTITY, it.QUANTITY as RESERVED\_QUANTITY, it.SEND\_DATE as Tranfer\_Date, it.STATUS\_ID

from inventory\_transfer it

join order\_item\_ship\_grp\_inv\_res ois on it.INVENTORY\_ITEM\_ID=ois.INVENTORY\_ITEM\_ID;



### **12 Orders Without Picklist**

Business Problem:  
A picklist is necessary for warehouse staff to gather items. Orders missing a picklist might be delayed and need attention.

Fields to Retrieve:

* ORDER\_ID
* ORDER\_DATE
* ORDER\_STATUS
* FACILITY\_ID
* DURATION (How long has the order been assigned at the facility)

select distinct oh.order\_id, oh.order\_date, oh.status\_id,

ois.facility\_id,

datediff(date(os.status\_datetime), date(oh.entry\_date)) as duration

from order\_header oh

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

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

join picklist pl on pl.FACILITY\_ID = ois.FACILITY\_ID

where pl.STATUS\_ID is null;

