Assignment 2

1)

### **Shipping Addresses for October 2023 Orders**

**Business Problem:**  
Customer Service might need to verify addresses for orders placed or completed in October 2023. This helps ensure shipments are delivered correctly and prevents address-related issues.

**Fields to Retrieve:**

* ORDER\_ID
* PARTY\_ID (Customer ID)
* CUSTOMER\_NAME (or FIRST\_NAME / LAST\_NAME)
* STREET\_ADDRESS
* CITY
* STATE\_PROVINCE
* POSTAL\_CODE
* COUNTRY\_CODE
* ORDER\_STATUS
* ORDER\_DATE

SELECT distinct oh.ORDER\_ID, ol.PARTY\_ID, p.FIRST\_NAME, p.LAST\_NAME, pa.ADDRESS1 as STREET\_ADDRESS, pa.CITY,

pa.STATE\_PROVINCE\_GEO\_ID as STATE\_PROVINCE, pa.POSTAL\_CODE, tn.COUNTRY\_CODE,

oh.STATUS\_ID as ORDER\_STATUS, oh.ORDER\_DATE

FROM order\_header oh

join order\_role ol on oh.ORDER\_ID = ol.ORDER\_ID

join person p on ol.PARTY\_ID = p.PARTY\_ID

join order\_contact\_mech ocm on oh.ORDER\_ID = ocm.ORDER\_ID

AND ocm.CONTACT\_MECH\_PURPOSE\_TYPE\_ID='SHIPPING\_LOCATION'

join order\_contact\_mech as oc on oh.ORDER\_ID = oc.ORDER\_ID AND oc.CONTACT\_MECH\_PURPOSE\_TYPE\_ID = "PHONE\_SHIPPING"

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

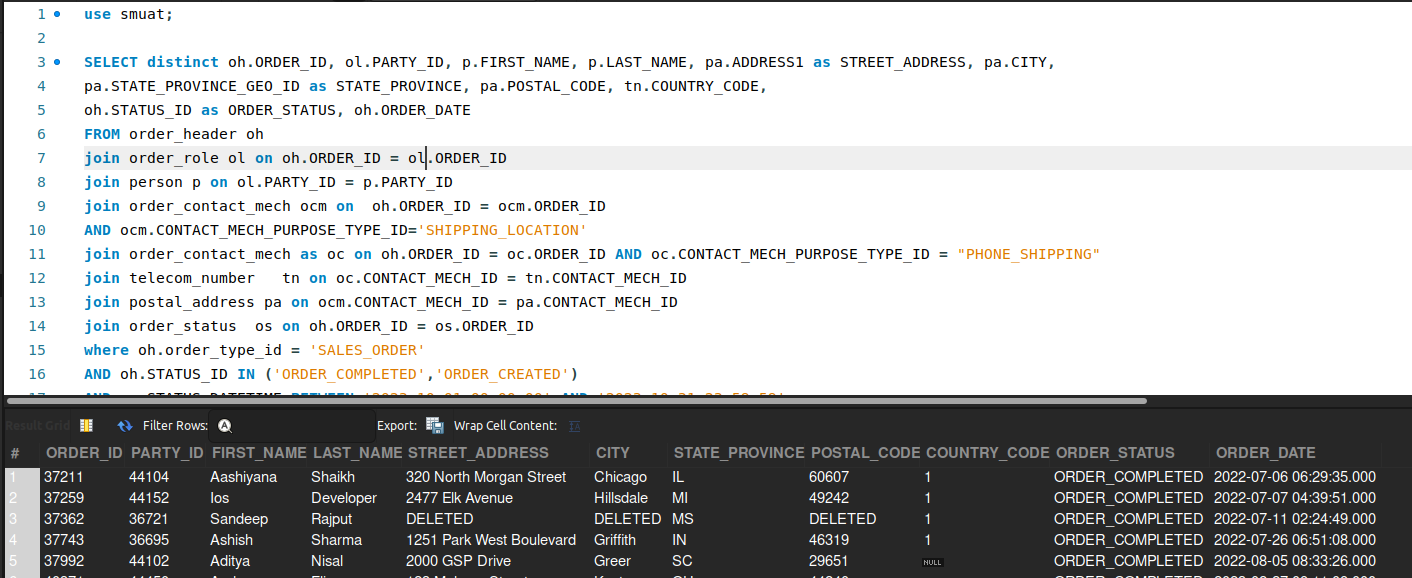
join postal\_address pa on ocm.CONTACT\_MECH\_ID = pa.CONTACT\_MECH\_ID

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

where oh.order\_type\_id = 'SALES\_ORDER'

AND oh.STATUS\_ID IN ('ORDER\_COMPLETED','ORDER\_CREATED')

AND os.STATUS\_DATETIME BETWEEN '2023-10-01 00:00:00' AND '2023-10-31 23:59:59';



2)

### **Orders from New York**

**Business Problem:**  
Companies often want region-specific analysis to plan local marketing, staffing, or promotions in certain areas—here, specifically, New York.

**Fields to Retrieve:**

* ORDER\_ID
* CUSTOMER\_NAME
* STREET\_ADDRESS (or shipping address detail)
* CITY
* STATE\_PROVINCE
* POSTAL\_CODE
* TOTAL\_AMOUNT
* ORDER\_DATE
* ORDER\_STATUS

select distinct oh.ORDER\_ID, p.FIRST\_NAME, p.LAST\_NAME,

pa.ADDRESS1 as STREET\_ADDRESS, pa.CITY, pa.STATE\_PROVINCE\_GEO\_ID as STATE\_PROVINCE,

pa.POSTAL\_CODE, oh.GRAND\_TOTAL as TOTAL\_AMOUNT, oh.ORDER\_DATE, oh.STATUS\_ID as ORDER\_STATUS

from order\_header oh

join order\_role ol on ol.order\_id=oh.order\_id

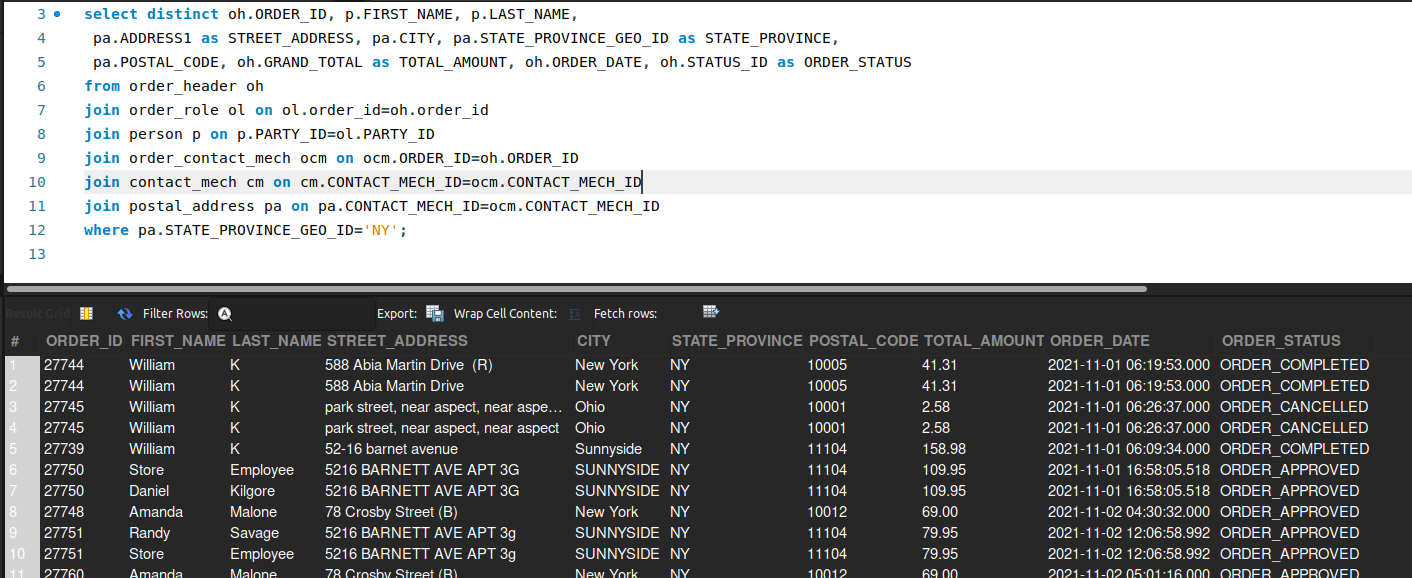
join person p on p.PARTY\_ID=ol.PARTY\_ID

join order\_contact\_mech ocm on ocm.ORDER\_ID=oh.ORDER\_ID

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

join postal\_address pa on pa.CONTACT\_MECH\_ID=ocm.CONTACT\_MECH\_ID

where pa.STATE\_PROVINCE\_GEO\_ID='NY';



3)

### **Top-Selling Product in New York**

**Business Problem:**  
Merchandising teams need to identify the best-selling product(s) in a specific region (New York) for targeted restocking or promotions.

**Fields to Retrieve:**

* PRODUCT\_ID
* INTERNAL\_NAME
* TOTAL\_QUANTITY\_SOLD
* CITY / STATE (within New York region)
* REVENUE (optionally, total sales amount)

select p.product\_id, p.internal\_name, sum(oi.quantity) as TOTAL\_QUANTITY\_SOLD, pa.CITY, sum(oi.quantity \* oi.unit\_price) as REVENUE

from product p

join order\_item oi on p.PRODUCT\_ID = oi.PRODUCT\_ID

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

join order\_contact\_mech ocm on oi.ORDER\_ID = ocm.ORDER\_ID

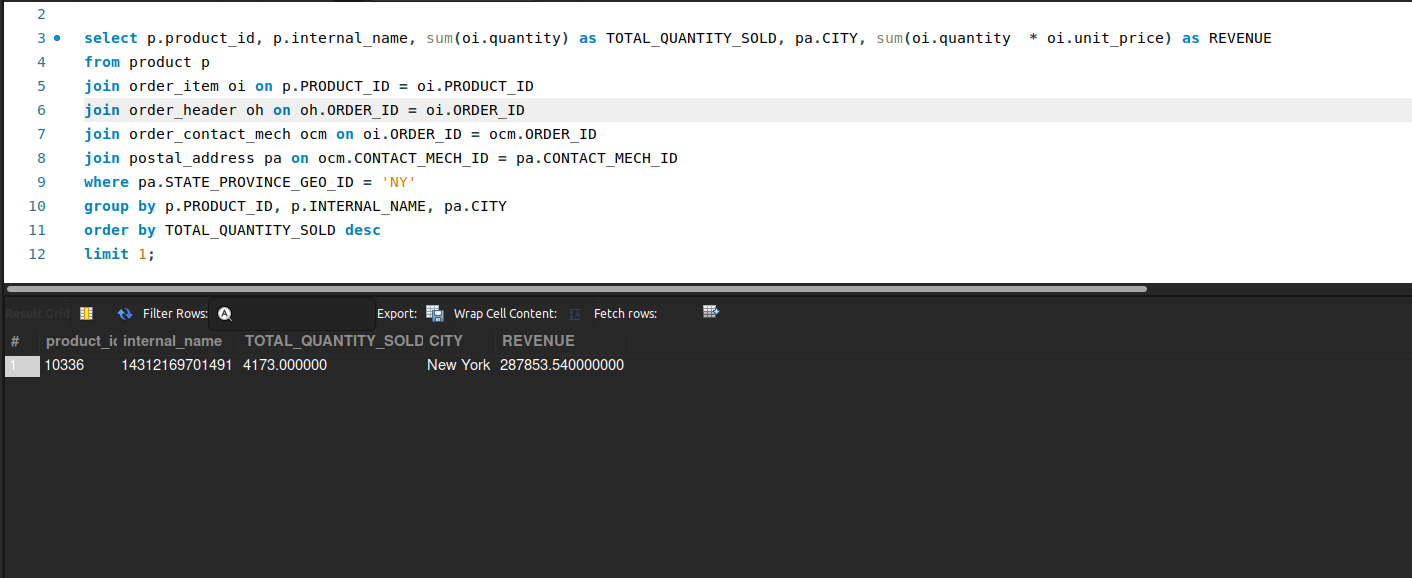
join postal\_address pa on ocm.CONTACT\_MECH\_ID = pa.CONTACT\_MECH\_ID

where pa.STATE\_PROVINCE\_GEO\_ID = 'NY'

group by p.PRODUCT\_ID, p.INTERNAL\_NAME, pa.CITY

order by TOTAL\_QUANTITY\_SOLD desc

limit 1;



4)

### **Store-Specific (Facility-Wise) Revenue**

**Business Problem:**  
Different physical or online stores (facilities) may have varying levels of performance. The business wants to compare revenue across facilities for sales planning and budgeting.

**Fields to Retrieve:**

* FACILITY\_ID
* FACILITY\_NAME
* TOTAL\_ORDERS
* TOTAL\_REVENUE
* DATE\_RANGE

select fa.FACILITY\_ID, fa.FACILITY\_NAME, count(oh.ORDER\_ID) as TOTAL\_ORDERS,sum(oh.GRAND\_TOTAL) as TOTAL\_REVENUE,

CONCAT(DATE(MIN(oh.ORDER\_DATE)), ' to ', DATE(MAX(oh.ORDER\_DATE))) AS DATE\_RANGE

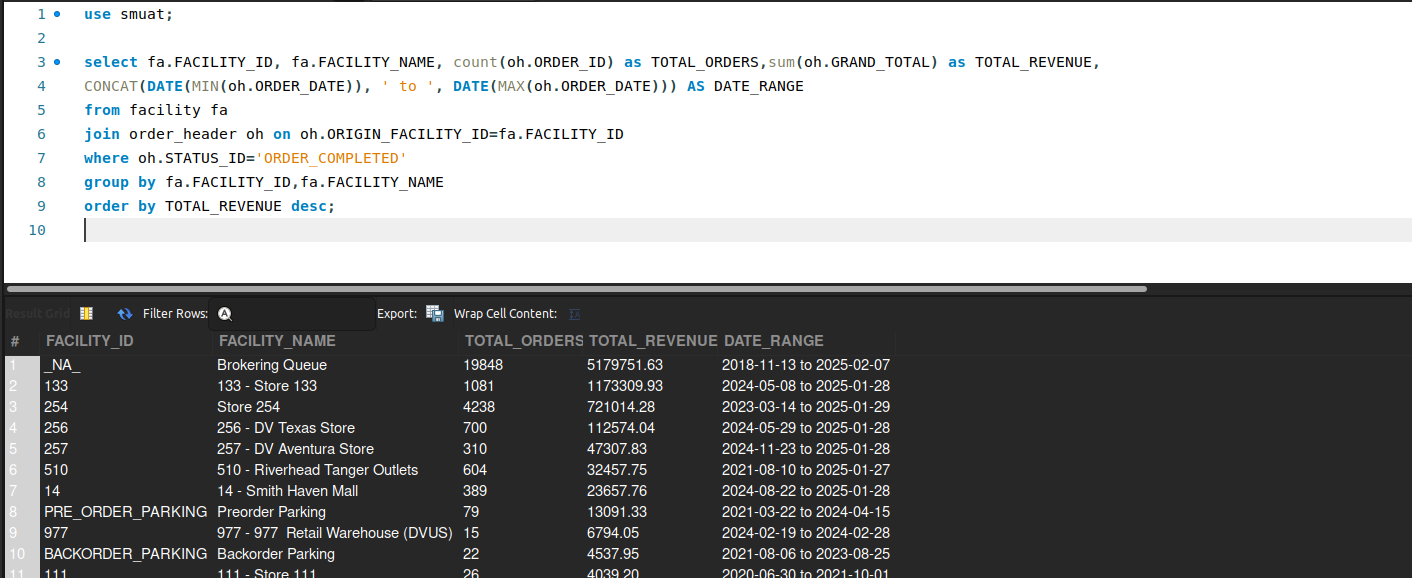
from facility fa

join order\_header oh on oh.ORIGIN\_FACILITY\_ID=fa.FACILITY\_ID

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

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

order by TOTAL\_REVENUE desc;



5)

### **Lost and Damaged Inventory**

**Business Problem:**  
Warehouse managers need to track “shrinkage” such as lost or damaged inventory to reconcile physical vs. system counts.

**Fields to Retrieve:**

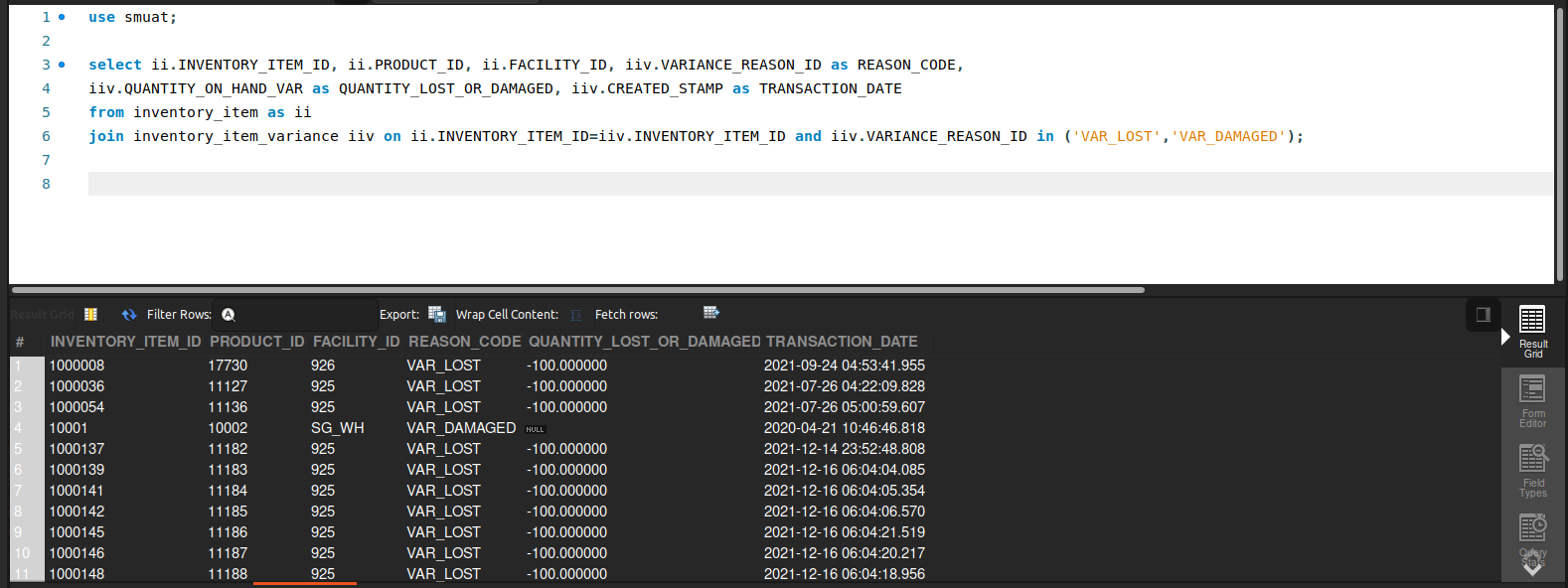
* INVENTORY\_ITEM\_ID
* PRODUCT\_ID
* FACILITY\_ID
* QUANTITY\_LOST\_OR\_DAMAGED
* REASON\_CODE (Lost, Damaged, Expired, etc.)
* TRANSACTION\_DATE

select ii.INVENTORY\_ITEM\_ID, ii.PRODUCT\_ID, ii.FACILITY\_ID, iiv.VARIANCE\_REASON\_ID as REASON\_CODE,

iiv.QUANTITY\_ON\_HAND\_VAR as QUANTITY\_LOST\_OR\_DAMAGED, iiv.CREATED\_STAMP as TRANSACTION\_DATE

from inventory\_item as ii

join inventory\_item\_variance iiv on ii.INVENTORY\_ITEM\_ID=iiv.INVENTORY\_ITEM\_ID and iiv.VARIANCE\_REASON\_ID in ('VAR\_LOST','VAR\_DAMAGED');



6)

### **Low Stock or Out of Stock Items Report**

**Business Problem:**  
Avoiding out-of-stock situations is critical. This report flags items that have fallen below a certain reorder threshold or have zero available stock.

**Fields to Retrieve:**

* PRODUCT\_ID
* PRODUCT\_NAME
* FACILITY\_ID
* QOH (Quantity on Hand)
* ATP (Available to Promise)
* REORDER\_THRESHOLD
* DATE\_CHECKED

select p.PRODUCT\_ID, p.PRODUCT\_NAME, pf.FACILITY\_ID, ii.QUANTITY\_ON\_HAND\_TOTAL, ii.AVAILABLE\_TO\_PROMISE\_TOTAL,

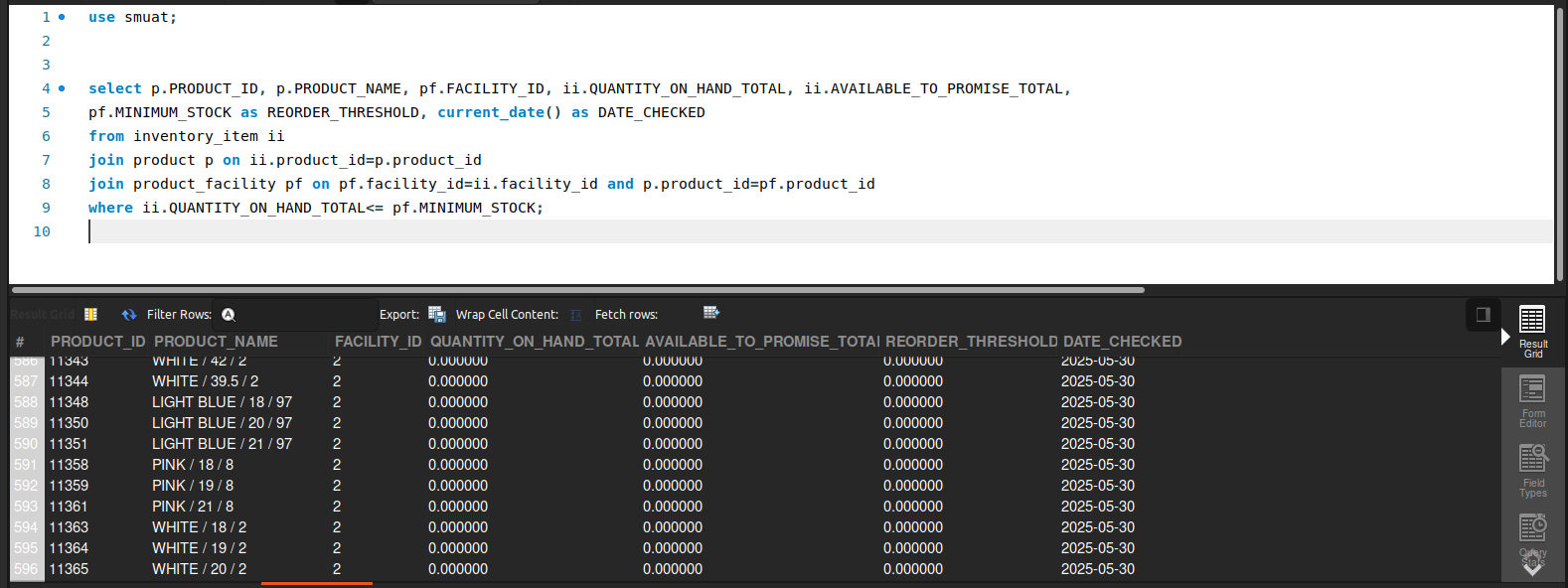
pf.MINIMUM\_STOCK as REORDER\_THRESHOLD, current\_date() as DATE\_CHECKED

from inventory\_item ii

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

join product\_facility pf on pf.facility\_id=ii.facility\_id and p.product\_id=pf.product\_id

where ii.QUANTITY\_ON\_HAND\_TOTAL<= pf.MINIMUM\_STOCK;



7)

### **Retrieve the Current Facility (Physical or Virtual) of Open Orders**

**Business Problem:**  
The business wants to know where open orders are currently assigned, whether in a physical store or a virtual facility (e.g., a distribution center or online fulfillment location).

**Fields to Retrieve:**

* ORDER\_ID
* ORDER\_STATUS
* FACILITY\_ID
* FACILITY\_NAME
* FACILITY\_TYPE\_ID

SELECT oh.ORDER\_ID, oh.STATUS\_ID as ORDER\_STATUS, fa.FACILITY\_ID,

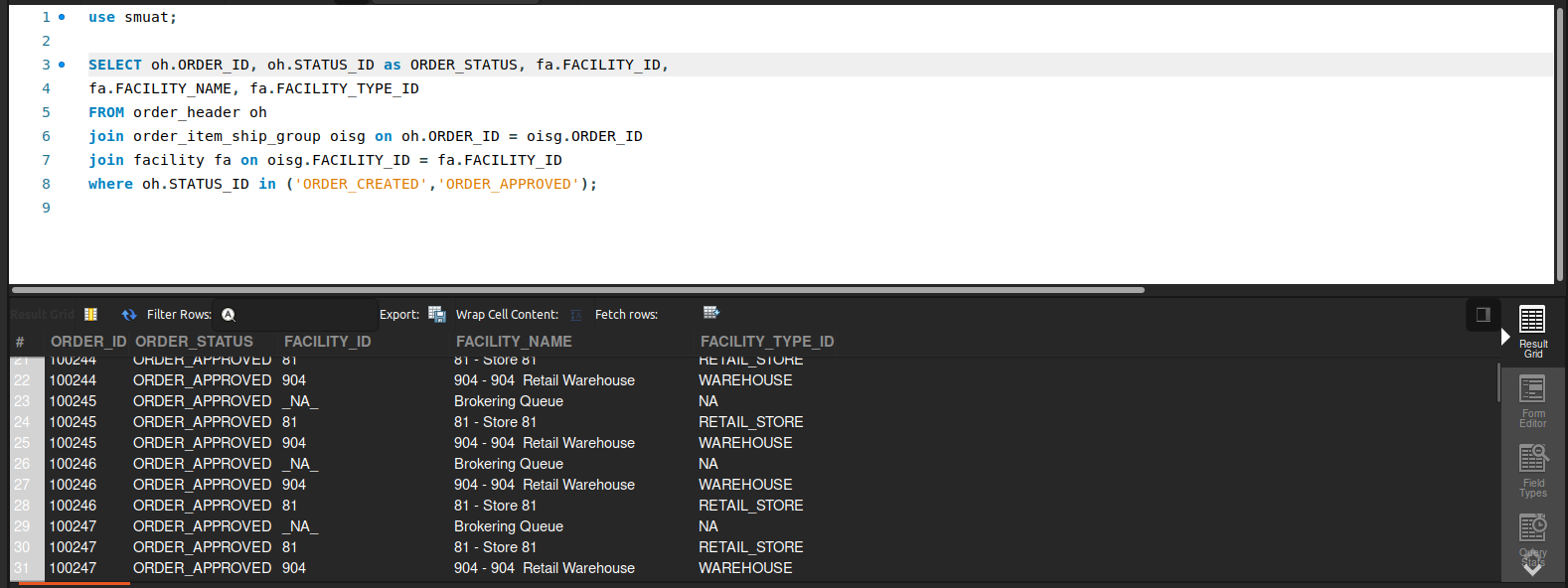
fa.FACILITY\_NAME, fa.FACILITY\_TYPE\_ID

FROM order\_header oh

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

join facility fa on oisg.FACILITY\_ID = fa.FACILITY\_ID

where oh.STATUS\_ID in ('ORDER\_CREATED','ORDER\_APPROVED');



8)

### **Items Where QOH and ATP Differ**

**Business Problem:**  
Sometimes the **Quantity on Hand (QOH)** doesn’t match the **Available to Promise (ATP)** due to pending orders, reservations, or data discrepancies. This needs review for accurate fulfillment planning.

**Fields to Retrieve:**

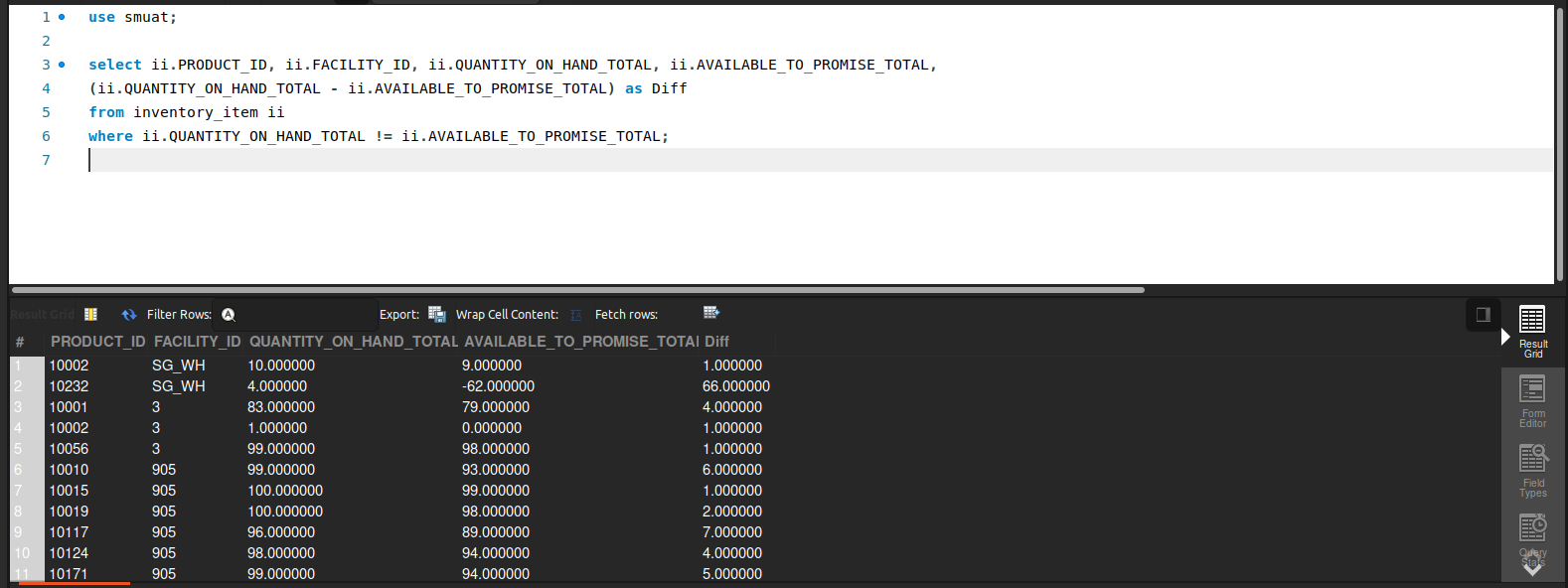
* PRODUCT\_ID
* FACILITY\_ID
* QOH (Quantity on Hand)
* ATP (Available to Promise)
* DIFFERENCE (QOH - ATP)

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

(ii.QUANTITY\_ON\_HAND\_TOTAL - ii.AVAILABLE\_TO\_PROMISE\_TOTAL) as Diff

from inventory\_item ii

where ii.QUANTITY\_ON\_HAND\_TOTAL != ii.AVAILABLE\_TO\_PROMISE\_TOTAL;



9)

### **Order Item Current Status Changed Date-Time**

**Business Problem:**  
Operations teams need to audit when an order item’s status (e.g., from “Pending” to “Shipped”) was last changed, for shipment tracking or dispute resolution.

**Fields to Retrieve:**

* ORDER\_ID
* ORDER\_ITEM\_SEQ\_ID
* CURRENT\_STATUS\_ID
* STATUS\_CHANGE\_DATETIME
* CHANGED\_BY

select os.ORDER\_ID, os.ORDER\_ITEM\_SEQ\_ID,

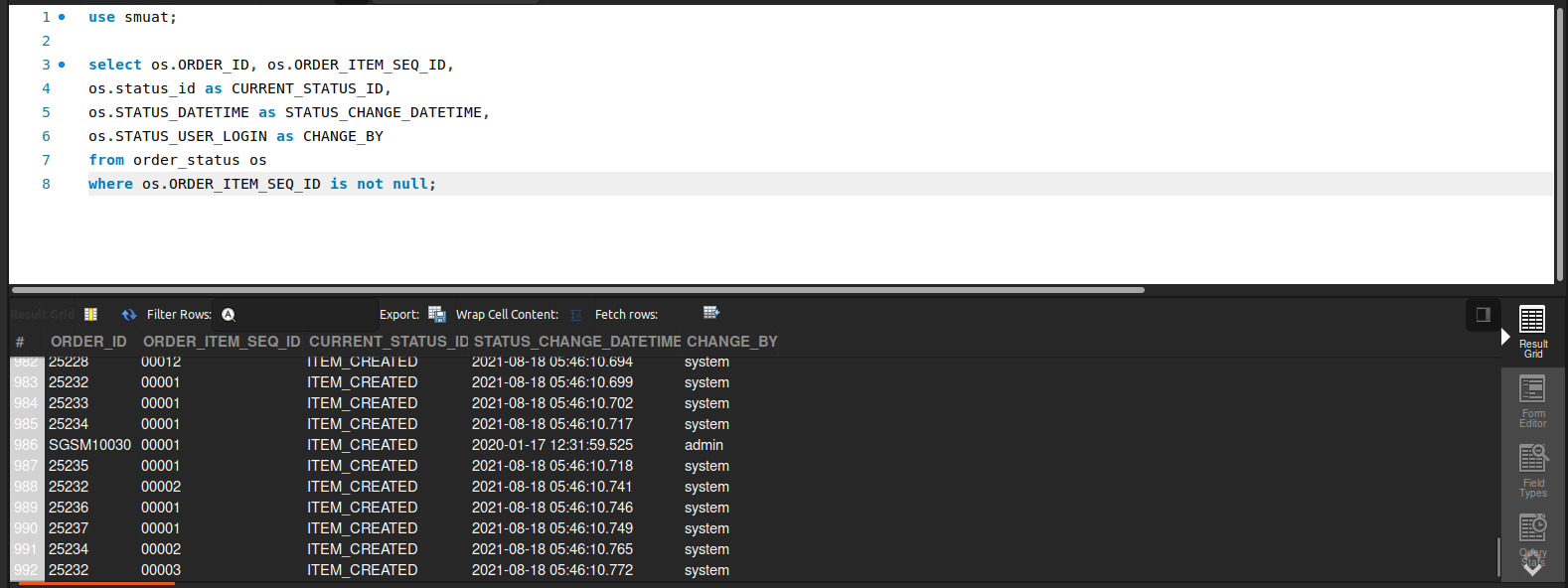
os.status\_id as CURRENT\_STATUS\_ID,

os.STATUS\_DATETIME as STATUS\_CHANGE\_DATETIME,

os.STATUS\_USER\_LOGIN as CHANGE\_BY

from order\_status os

where os.ORDER\_ITEM\_SEQ\_ID is not null;



10)

### **Total Orders by Sales Channel**

**Business Problem:**  
Marketing and sales teams want to see how many orders come from each channel (e.g., web, mobile app, in-store POS, marketplace) to allocate resources effectively.

**Fields to Retrieve:**

* SALES\_CHANNEL
* TOTAL\_ORDERS
* TOTAL\_REVENUE
* REPORTING\_PERIOD

select oh.sales\_channel\_enum\_id as SALES\_CHANNEL,

count(oh.order\_id) as TOTAL\_ORDERS,

sum(oh.grand\_total) as TOTAL\_REVENUE,

DATE\_FORMAT(oh.ORDER\_DATE, '%Y-%m') AS REPORTING\_PERIOD

from order\_header oh

group by SALES\_CHANNEL, REPORTING\_PERIOD;

