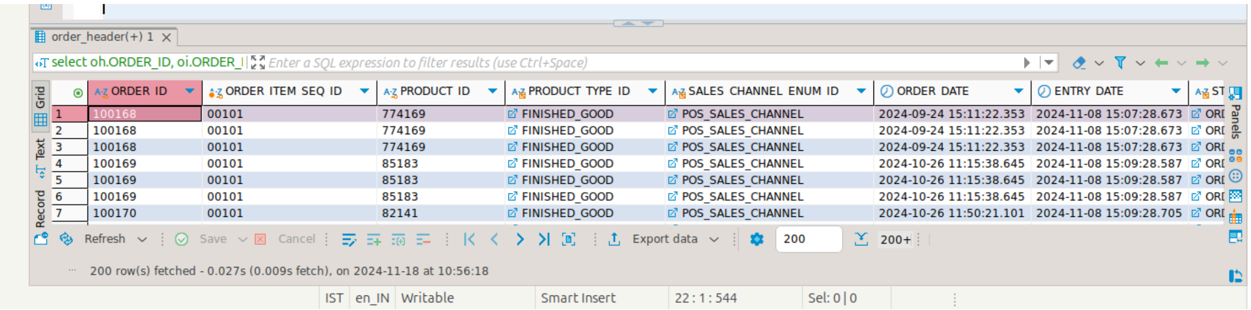
SQL ASSIGNMENT 2

1. **Fetch the following columns for completed order items for sales orders of SM_STORE product store and that are physical items.**

Updated Query:-

```
select oh.ORDER_ID,
    oi.ORDER_ITEM_SEQ_ID,
    p.PRODUCT_ID,
    p.PRODUCT_TYPE_ID,
    oh.SALES_CHANNEL_ENUM_ID,
    oh.ORDER_DATE,
    oh.ENTRY_DATE,
    oh.STATUS_ID,
    oh.ORDER_TYPE_ID,
    os.STATUS_DATETIME ,
    oh.PRODUCT_STORE_ID
FROM
order_header oh
JOIN
order_item oi
on oh.ORDER_ID = oi.ORDER_ID
JOIN
product p on p.PRODUCT_ID = oi.PRODUCT_ID
join order_status os on oh.ORDER_ID = os.ORDER_ID
where oh.PRODUCT_STORE_ID = 'SM_STORE' and oh.STATUS_ID =
'ORDER_COMPLETED' AND
oh.ORDER_TYPE_ID = 'SALES_ORDER';
```

Result :-



order_header(+) 1

select oh.ORDER_ID, oi.ORDER_I | Enter a SQL expression to filter results (use Ctrl+Space)

	ORDER ID	ORDER ITEM SEQ ID	PRODUCT ID	PRODUCT TYPE ID	SALES CHANNEL ENUM ID	ORDER DATE	ENTRY DATE	ST
1	100168	00101	774169	FINISHED_GOOD	POS_SALES_CHANNEL	2024-09-24 15:11:22.353	2024-11-08 15:07:28.673	ORI
2	100168	00101	774169	FINISHED_GOOD	POS_SALES_CHANNEL	2024-09-24 15:11:22.353	2024-11-08 15:07:28.673	ORI
3	100168	00101	774169	FINISHED_GOOD	POS_SALES_CHANNEL	2024-09-24 15:11:22.353	2024-11-08 15:07:28.673	ORI
4	100169	00101	85183	FINISHED_GOOD	POS_SALES_CHANNEL	2024-10-26 11:15:38.645	2024-11-08 15:09:28.587	ORI
5	100169	00101	85183	FINISHED_GOOD	POS_SALES_CHANNEL	2024-10-26 11:15:38.645	2024-11-08 15:09:28.587	ORI
6	100169	00101	85183	FINISHED_GOOD	POS_SALES_CHANNEL	2024-10-26 11:15:38.645	2024-11-08 15:09:28.587	ORI
7	100170	00101	82141	FINISHED_GOOD	POS_SALES_CHANNEL	2024-10-26 11:50:21.101	2024-11-08 15:09:28.705	ORI

200 row(s) fetched - 0.027s (0.009s fetch), on 2024-11-18 at 10:56:18

Query :-

```
select oh.ORDER_ID,
      oi.ORDER_ITEM_SEQ_ID,
      p.PRODUCT_ID,
      p.PRODUCT_TYPE_ID,
      oh.SALES_CHANNEL_ENUM_ID,
      oh.ORDER_DATE,
      oh.ENTRY_DATE,
      os.STATUS_ID,
      os.STATUS_DATETIME,
      oh.ORDER_TYPE_ID,
      oh.PRODUCT_STORE_ID,
case when p.PRODUCT_TYPE_ID = 'digital_good' then 'Y' ELSE
'N' end AS
IS_DIGITAL,
case when p.PRODUCT_TYPE_ID = 'finished_good' then 'Y' else 'N' end  AS
IS_PHYSICAL
FROM
order_header oh
JOIN
order_item oi
on oh.ORDER_ID = oi.ORDER_ID
JOIN
product p on p.PRODUCT_ID = oi.PRODUCT_ID
JOIN
order_status os
on os.ORDER_ID = oi.ORDER_ID
and os.ORDER_ITEM_SEQ_ID = oi.ORDER_ITEM_SEQ_ID
```

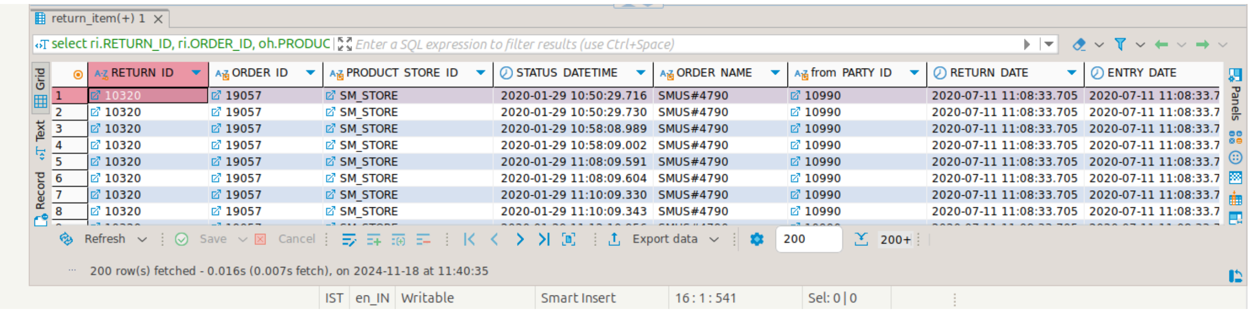
```
where oh.PRODUCT_STORE_ID = 'SM_STORE' and os.STATUS_ID =
'ORDER_COMPLETED' AND
oh.ORDER_TYPE_ID= 'SALES_ORDER';
```

2. Fetch the following columns for completed return items of SM_STORE for ecom return channel.

Updated Query: -

```
select ri.RETURN_ID,
   ri.ORDER_ID,
   oh.PRODUCT_STORE_ID,
   rs.STATUS_DATETIME,
   oh.ORDER_NAME,
   rh.from_PARTY_ID,
   rh.RETURN_DATE,
   rh.ENTRY_DATE,
   rh.RETURN_CHANNEL_ENUM_ID
from return_header rh join return_item ri
on rh.RETURN_ID = ri.RETURN_ID
join order_header oh on ri.ORDER_ID = oh.ORDER_ID
join return_status rs on rh.STATUS_ID = rs.STATUS_ID
and ri.STATUS_ID = rs.STATUS_ID
where ri.STATUS_ID = 'return_completed' and oh.PRODUCT_STORE_ID = 'sm_store'
and rh.RETURN_CHANNEL_ENUM_ID = 'ecom_rtn_channel'
```

Result:-



	RETURN ID	ORDER ID	PRODUCT STORE ID	STATUS DATETIME	ORDER NAME	from PARTY ID	RETURN DATE	ENTRY DATE
1	10320	19057	SM_STORE	2020-01-29 10:50:29.716	SMUS#4790	10990	2020-07-11 11:08:33.705	2020-07-11 11:08:33.7
2	10320	19057	SM_STORE	2020-01-29 10:50:29.730	SMUS#4790	10990	2020-07-11 11:08:33.705	2020-07-11 11:08:33.7
3	10320	19057	SM_STORE	2020-01-29 10:58:08.989	SMUS#4790	10990	2020-07-11 11:08:33.705	2020-07-11 11:08:33.7
4	10320	19057	SM_STORE	2020-01-29 10:58:09.002	SMUS#4790	10990	2020-07-11 11:08:33.705	2020-07-11 11:08:33.7
5	10320	19057	SM_STORE	2020-01-29 11:08:09.591	SMUS#4790	10990	2020-07-11 11:08:33.705	2020-07-11 11:08:33.7
6	10320	19057	SM_STORE	2020-01-29 11:08:09.604	SMUS#4790	10990	2020-07-11 11:08:33.705	2020-07-11 11:08:33.7
7	10320	19057	SM_STORE	2020-01-29 11:10:09.330	SMUS#4790	10990	2020-07-11 11:08:33.705	2020-07-11 11:08:33.7
8	10320	19057	SM_STORE	2020-01-29 11:10:09.343	SMUS#4790	10990	2020-07-11 11:08:33.705	2020-07-11 11:08:33.7

200 row(s) fetched - 0.016s (0.007s fetch), on 2024-11-18 at 11:40:35

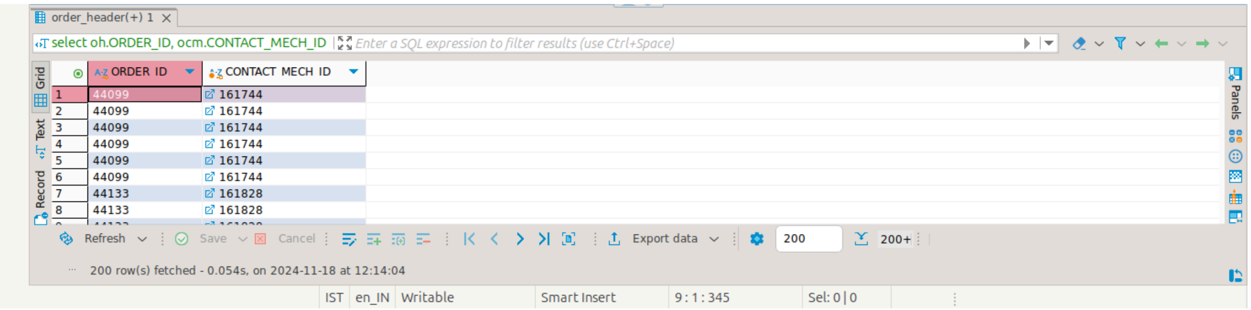
Query:-

```
select ri.RETURN_ID,
    ri.ORDER_ID,
    ps.PRODUCT_STORE_ID,
    rs.STATUS_DATETIME,
    oh.ORDER_NAME,
    rh.from_PARTY_ID,
    rh.RETURN_DATE,
    rh.ENTRY_DATE,
    rh.RETURN_CHANNEL_ENUM_ID
from return_header rh join return_item ri
on rh.RETURN_ID = ri.RETURN_ID
join order_header oh on ri.ORDER_ID = oh.ORDER_ID
join product p on ri.PRODUCT_ID = p.PRODUCT_ID
join product_category pc on p.PRIMARY_PRODUCT_CATEGORY_ID =
pc.PRODUCT_CATEGORY_ID
join prod_catalog_category pcc on pc.PRODUCT_CATEGORY_ID =
pcc.PRODUCT_CATEGORY_ID
join prod_catalog pc2 on pc2.PROD_CATALOG_ID  = pcc.PROD_CATALOG_ID
join product_store_catalog psc on psc.PROD_CATALOG_ID =
pc2.PROD_CATALOG_ID
join product_store ps on ps.PRODUCT_STORE_ID = psc.PRODUCT_STORE_ID
join return_status rs on rh.STATUS_ID = rs.STATUS_ID
and ri.STATUS_ID = rs.STATUS_ID
where ri.STATUS_ID = 'return_completed' and ps.PRODUCT_STORE_ID = 'sm_store'
and rh.RETURN_CHANNEL_ENUM_ID = 'ecom_rtn_channel'
```

3. Fetch the order id and contact mech id for the shipping address of the orders completed in October of 2023.

Updated Query:-

Result:-



select oh.ORDER_ID, ocm.CONTACT_MECH_ID

Enter a SQL expression to filter results (use Ctrl+Space)

	ORDER ID	CONTACT MECH ID
1	44099	161744
2	44099	161744
3	44099	161744
4	44099	161744
5	44099	161744
6	44099	161744
7	44133	161828
8	44133	161828

200 row(s) fetched - 0.054s, on 2024-11-18 at 12:14:04

IST en_IN Writable Smart Insert 9 : 1 : 345 Sel: 0 | 0

Query:-

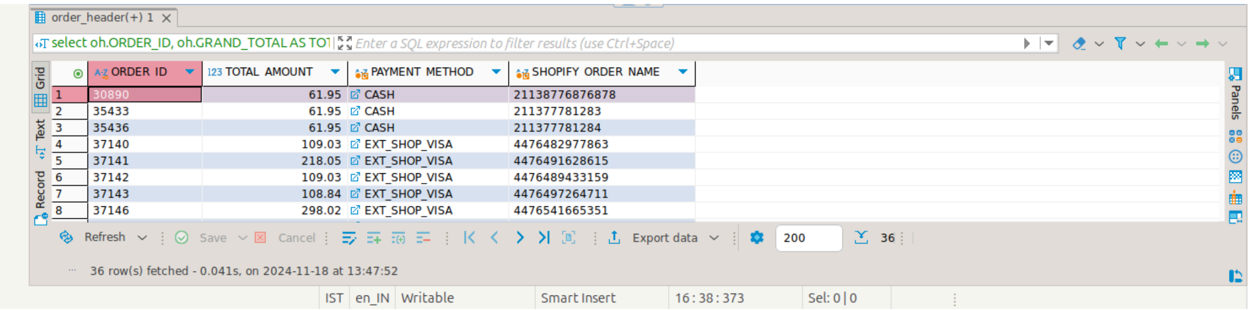
```
select oh.ORDER_ID,
     ocm.CONTACT_MECH_ID
from order_header oh
join order_contact_mech ocm on oh.ORDER_ID = ocm.ORDER_ID
join order_status os on  os.ORDER_ID = oh.ORDER_ID
where  ocm.CONTACT_MECH_PURPOSE_TYPE_ID = 'SHIPPING_LOCATION'
 and  oh.STATUS_ID = 'ORDER_COMPLETED'
 and  os.STATUS_DATETIME between '2023-10-01' and '2023-10-31';
```

4. Fetch the following columns for created orders. These should be sales orders.

Update Query:-

```
select
  oh.ORDER_ID,
  oh.GRAND_TOTAL AS TOTAL_AMOUNT,
  opp.PAYMENT_METHOD_TYPE_ID AS PAYMENT_METHOD,
  oi.ID_VALUE AS SHOPIFY_ORDER_NAME
from
  order_header oh
join
  order_identification oi
on oh.ORDER_ID = oi.ORDER_ID
join
  order_payment_preference opp
on opp.ORDER_ID = oh.ORDER_ID
where
  oh.STATUS_ID = 'order_created'
and  oh.ORDER_TYPE_ID = 'sales_order'
 group by ORDER_ID
```

Result : -



select oh.ORDER_ID, oh.GRAND_TOTAL AS TOT

Enter a SQL expression to filter results (use Ctrl+Space)

	ORDER ID	TOTAL AMOUNT	PAYMENT METHOD	SHOPIFY ORDER NAME
1	30890	61.95	CASH	21138776876878
2	35433	61.95	CASH	211377781283
3	35436	61.95	CASH	211377781284
4	37140	109.03	EXT_SHOP_VISA	4476482977863
5	37141	218.05	EXT_SHOP_VISA	4476491628615
6	37142	109.03	EXT_SHOP_VISA	4476489433159
7	37143	108.84	EXT_SHOP_VISA	4476497264711
8	37146	298.02	EXT_SHOP_VISA	4476541665351

36 row(s) fetched - 0.041s, on 2024-11-18 at 13:47:52

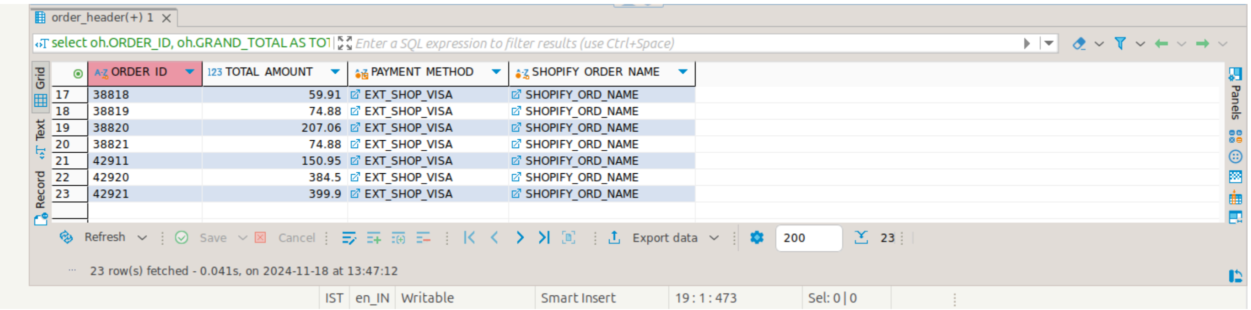
IST | en_IN | Writable | Smart Insert | 16 : 38 : 373 | Sel: 0 | 0

Query :-

```
select
   oh.ORDER_ID,
   oh.GRAND_TOTAL AS TOTAL_AMOUNT,
   opp.PAYMENT_METHOD_TYPE_ID AS PAYMENT_METHOD,
   oi.ORDER_IDENTIFICATION_TYPE_ID AS SHOPIFY_ORDER_NAME
from
   order_header oh
join
   order_identification oi
on oh.ORDER_ID = oi.ORDER_ID
join
   order_payment_preference opp
on opp.ORDER_ID = oh.ORDER_ID
where
   oh.STATUS_ID = 'order_created'
```

```
and  oh.ORDER_TYPE_ID = 'sales_order'
and  oi.ORDER_IDENTIFICATION_TYPE_ID = 'shopify_ord_name'
  group by ORDER_ID;
```

Result :-



order_header(+) 1
select oh.ORDER_ID, oh.GRAND_TOTAL AS TOT
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
Record
Panels

	ORDER ID	TOTAL AMOUNT	PAYMENT METHOD	SHOPIFY ORDER NAME
17	38818	59.91	EXT_SHOP_VISA	SHOPIFY_ORD_NAME
18	38819	74.88	EXT_SHOP_VISA	SHOPIFY_ORD_NAME
19	38820	207.06	EXT_SHOP_VISA	SHOPIFY_ORD_NAME
20	38821	74.88	EXT_SHOP_VISA	SHOPIFY_ORD_NAME
21	42911	150.95	EXT_SHOP_VISA	SHOPIFY_ORD_NAME
22	42920	384.5	EXT_SHOP_VISA	SHOPIFY_ORD_NAME
23	42921	399.9	EXT_SHOP_VISA	SHOPIFY_ORD_NAME

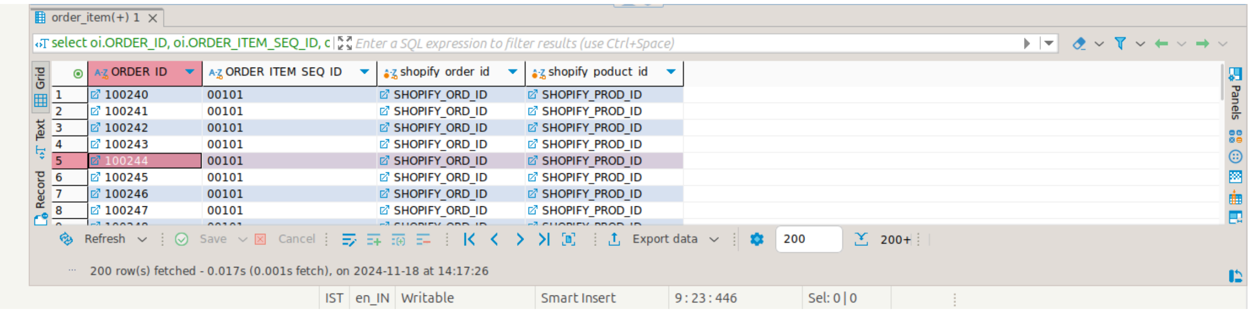
23 row(s) fetched - 0.041s, on 2024-11-18 at 13:47:12

5. Fetch the following data for completed order items in July of 2023

Query :-

```
select oi.ORDER_ID,
     oi.ORDER_ITEM_SEQ_ID,
     oi2.ORDER_IDENTIFICATION_TYPE_ID as shopify_order_id,
     gi.GOOD_IDENTIFICATION_TYPE_ID as shopify_product_id
from order_item oi join order_identification oi2
on oi.ORDER_ID = oi2.ORDER_ID
join good_identification gi on gi.PRODUCT_ID = oi.PRODUCT_ID
where oi2.ORDER_IDENTIFICATION_TYPE_ID = 'shopify_ord_id' and
gi.GOOD_IDENTIFICATION_TYPE_ID = 'shopify_prod_id'
group by oi.ORDER_ID;
```

Result :-



select oi.ORDER_ID, oi.ORDER_ITEM_SEQ_ID, c

	ORDER ID	ORDER ITEM SEQ ID	shopify order id	shopify poduct id
1	100240	00101	SHOPIFY_ORD_ID	SHOPIFY_PROD_ID
2	100241	00101	SHOPIFY_ORD_ID	SHOPIFY_PROD_ID
3	100242	00101	SHOPIFY_ORD_ID	SHOPIFY_PROD_ID
4	100243	00101	SHOPIFY_ORD_ID	SHOPIFY_PROD_ID
5	100244	00101	SHOPIFY_ORD_ID	SHOPIFY_PROD_ID
6	100245	00101	SHOPIFY_ORD_ID	SHOPIFY_PROD_ID
7	100246	00101	SHOPIFY_ORD_ID	SHOPIFY_PROD_ID
8	100247	00101	SHOPIFY_ORD_ID	SHOPIFY_PROD_ID

200 row(s) fetched - 0.017s (0.001s fetch), on 2024-11-18 at 14:17:26

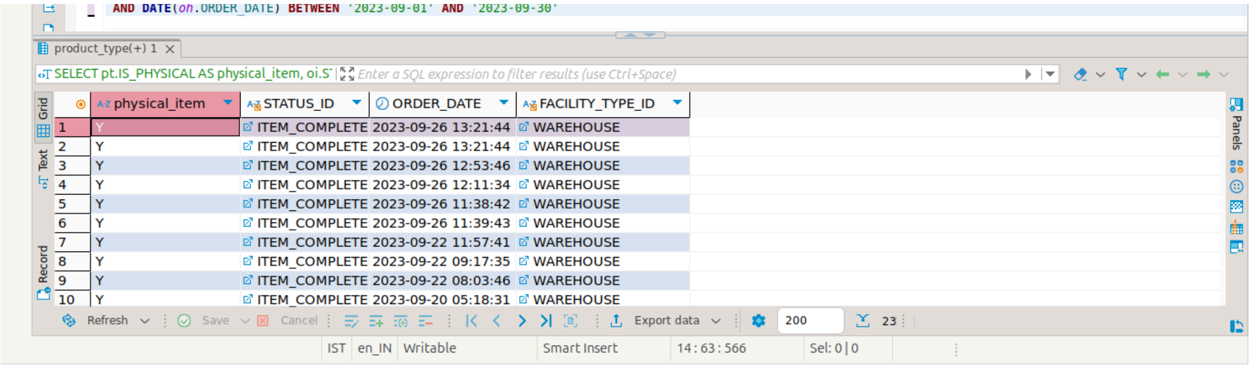
IST en_IN Writable Smart Insert 9 : 23 : 446 Sel: 0 | 0

6. Fetch all the physical items completed from Warehouse in September of 2023.

Update Query:-

```
SELECT pt.IS_PHYSICAL AS physical_item,
    oi.STATUS_ID,
    oh.ORDER_DATE,
    f.FACILITY_TYPE_ID
FROM product_type pt
JOIN product p ON pt.PRODUCT_TYPE_ID = p.PRODUCT_TYPE_ID
JOIN order_item oi ON oi.PRODUCT_ID = p.PRODUCT_ID
JOIN order_header oh ON oh.ORDER_ID = oi.ORDER_ID
join order_item_ship_group oisg on oisg.ORDER_ID = oh.ORDER_ID
JOIN facility f ON f.FACILITY_ID = oisg.FACILITY_ID
WHERE pt.IS_PHYSICAL = 'Y'
AND oi.STATUS_ID = 'item_completed'
AND f.FACILITY_TYPE_ID = 'warehouse'
AND DATE(oh.ORDER_DATE) BETWEEN '2023-09-01' AND '2023-09-30'
```

Result:-



```
AND DATE(oh.ORDER_DATE) BETWEEN '2023-09-01' AND '2023-09-30'
```

product_type(+) 1

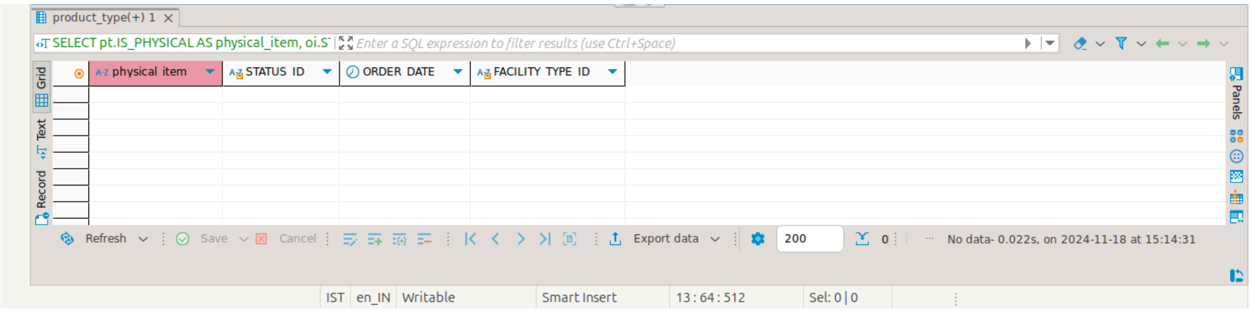
SELECT pt.IS_PHYSICAL AS physical_item, oi.S | Enter a SQL expression to filter results (use Ctrl+Space)

	physical_item	STATUS_ID	ORDER_DATE	FACILITY_TYPE_ID
1	Y	ITEM_COMPLETE	2023-09-26 13:21:44	WAREHOUSE
2	Y	ITEM_COMPLETE	2023-09-26 13:21:44	WAREHOUSE
3	Y	ITEM_COMPLETE	2023-09-26 12:53:46	WAREHOUSE
4	Y	ITEM_COMPLETE	2023-09-26 12:11:34	WAREHOUSE
5	Y	ITEM_COMPLETE	2023-09-26 11:38:42	WAREHOUSE
6	Y	ITEM_COMPLETE	2023-09-26 11:39:43	WAREHOUSE
7	Y	ITEM_COMPLETE	2023-09-22 11:57:41	WAREHOUSE
8	Y	ITEM_COMPLETE	2023-09-22 09:17:35	WAREHOUSE
9	Y	ITEM_COMPLETE	2023-09-22 08:03:46	WAREHOUSE
10	Y	ITEM_COMPLETE	2023-09-20 05:18:31	WAREHOUSE

Query: -

```
SELECT pt.IS_PHYSICAL AS physical_item,
      oi.STATUS_ID,
      oh.ORDER_DATE,
      f.FACILITY_TYPE_ID
FROM product_type pt
JOIN product p ON pt.PRODUCT_TYPE_ID = p.PRODUCT_TYPE_ID
JOIN order_item oi ON oi.PRODUCT_ID = p.PRODUCT_ID
JOIN order_header oh ON oh.ORDER_ID = oi.ORDER_ID
JOIN facility f ON f.FACILITY_ID = oh.ORIGIN_FACILITY_ID
WHERE pt.IS_PHYSICAL = 'Y'
 AND oi.STATUS_ID = 'item_completed'
 AND f.FACILITY_TYPE_ID = 'warehouse'
 AND DATE(oh.ORDER_DATE) BETWEEN '2023-09-01' AND '2023-09-30'
```

Result:-



product_type(+) 1

SELECT pt.IS_PHYSICAL AS physical_item, oi.S | Enter a SQL expression to filter results (use Ctrl+Space)

	physical item	STATUS ID	ORDER DATE	FACILITY TYPE ID

No data- 0.022s, on 2024-11-18 at 15:14:31

Note:- There is no record in September 2023 based on condition

7. Fetch all the physical items ordered in the month of September 2023.

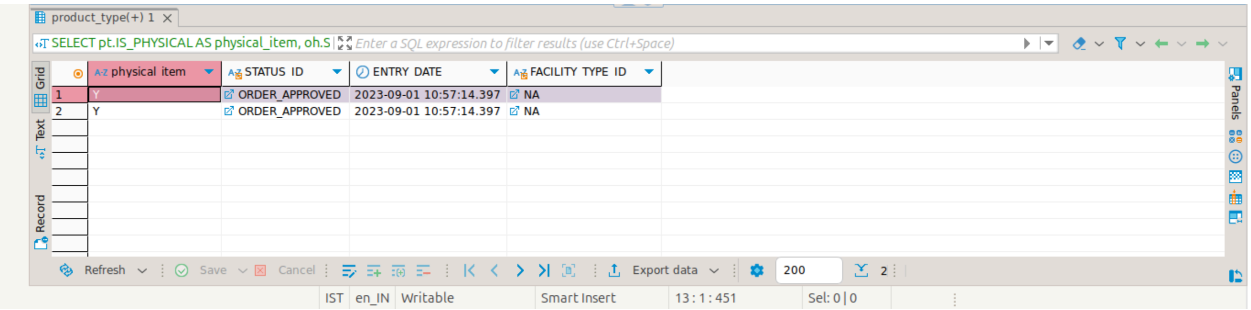
Update Query:-

```
SELECT pt.IS_PHYSICAL AS physical_item,
    oh.STATUS_ID,
    oh.ENTRY_DATE,
    f.FACILITY_TYPE_ID
FROM product_type pt
JOIN product p ON pt.PRODUCT_TYPE_ID = p.PRODUCT_TYPE_ID
JOIN order_item oi ON oi.PRODUCT_ID = p.PRODUCT_ID
JOIN order_header oh ON oh.ORDER_ID = oi.ORDER_ID
join order_item_ship_group oisg on oisg.ORDER_ID = oh.ORDER_ID
JOIN facility f ON f.FACILITY_ID = oisg.FACILITY_ID
WHERE pt.IS_PHYSICAL = 'Y'
AND oh.STATUS_ID = 'order_approved'
AND DATE(oh.ENTRY_DATE) = '2023-09-01'
```

Query:-

```
SELECT pt.IS_PHYSICAL AS physical_item,
     oh.STATUS_ID,
     oh.ENTRY_DATE,
     f.FACILITY_TYPE_ID
FROM product_type pt
JOIN product p ON pt.PRODUCT_TYPE_ID = p.PRODUCT_TYPE_ID
JOIN order_item oi ON oi.PRODUCT_ID = p.PRODUCT_ID
JOIN order_header oh ON oh.ORDER_ID = oi.ORDER_ID
JOIN facility f ON f.FACILITY_ID = oh.ORIGIN_FACILITY_ID
WHERE pt.IS_PHYSICAL = 'Y'
 AND oh.STATUS_ID = 'order_approved'
 AND DATE(oh.ENTRY_DATE) = '2023-09-01'
```

Result :-



```
SELECT pt.IS_PHYSICAL AS physical_item, oh.S
```

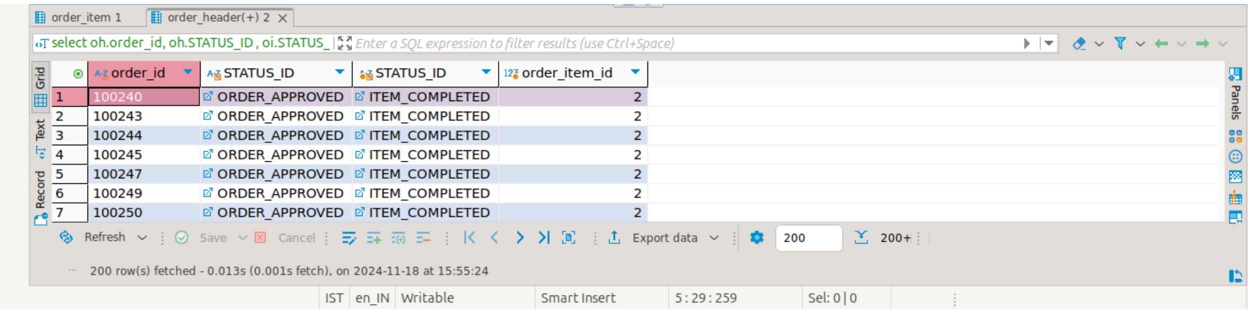
	physical item	STATUS ID	ENTRY DATE	FACILITY TYPE ID
1	Y	ORDER_APPROVED	2023-09-01 10:57:14.397	NA
2	Y	ORDER_APPROVED	2023-09-01 10:57:14.397	NA

8. Find all the orders whose two or more items are completed but the orders are still in the approved status.

Query :-

```
select oh.order_id,
    count(oi.order_item_seq_id) as order_item_id
    from order_header oh join order_item oi on oh.ORDER_ID  = oi.ORDER_ID
    where oi.STATUS_ID  = 'item_completed' and oh.STATUS_ID = 'order_approved'
    group by oh.ORDER_ID
    having order_item_id  >= 2 ;
```

Result :-



```
select oh.order_id, oh.STATUS_ID , oi.STATUS_
```

	order_id	STATUS_ID	STATUS_ID	order_item_id
1	100240	ORDER_APPROVED	ITEM_COMPLETED	2
2	100243	ORDER_APPROVED	ITEM_COMPLETED	2
3	100244	ORDER_APPROVED	ITEM_COMPLETED	2
4	100245	ORDER_APPROVED	ITEM_COMPLETED	2
5	100247	ORDER_APPROVED	ITEM_COMPLETED	2
6	100249	ORDER_APPROVED	ITEM_COMPLETED	2
7	100250	ORDER_APPROVED	ITEM_COMPLETED	2

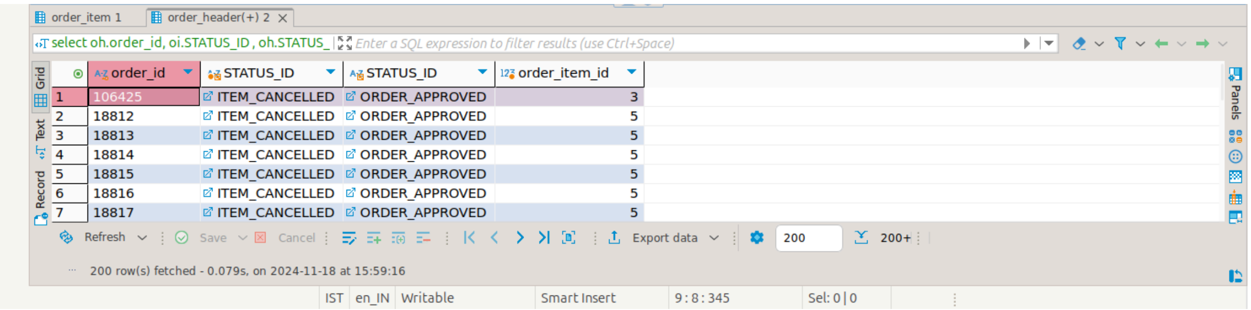
```
200 row(s) fetched - 0.013s (0.001s fetch), on 2024-11-18 at 15:55:24
```

9. Find all the orders whose two or more items are canceled but the orders are still in the approved status.

Query :-

```
select oh.order_id,
    oi.STATUS_ID ,
    oh.STATUS_ID ,
    count(oi.order_item_seq_id) as order_item_id
    from order_header oh join order_item oi on oh.ORDER_ID  = oi.ORDER_ID
    where oi.STATUS_ID  = 'item_cancelled' and oh.STATUS_ID = 'order_approved'
    group by oh.ORDER_ID
    having order_item_id  >= 2
```

Result :-



order_item 1
order_header(+) 2
select oh.order_id, oi.STATUS_ID , oh.STATUS_
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
Record
Panels
Refresh
Save
Cancel
Export data
200
200+

	order_id	STATUS_ID	STATUS_ID	order_item_id
1	106425	ITEM_CANCELLED	ORDER_APPROVED	3
2	18812	ITEM_CANCELLED	ORDER_APPROVED	5
3	18813	ITEM_CANCELLED	ORDER_APPROVED	5
4	18814	ITEM_CANCELLED	ORDER_APPROVED	5
5	18815	ITEM_CANCELLED	ORDER_APPROVED	5
6	18816	ITEM_CANCELLED	ORDER_APPROVED	5
7	18817	ITEM_CANCELLED	ORDER_APPROVED	5

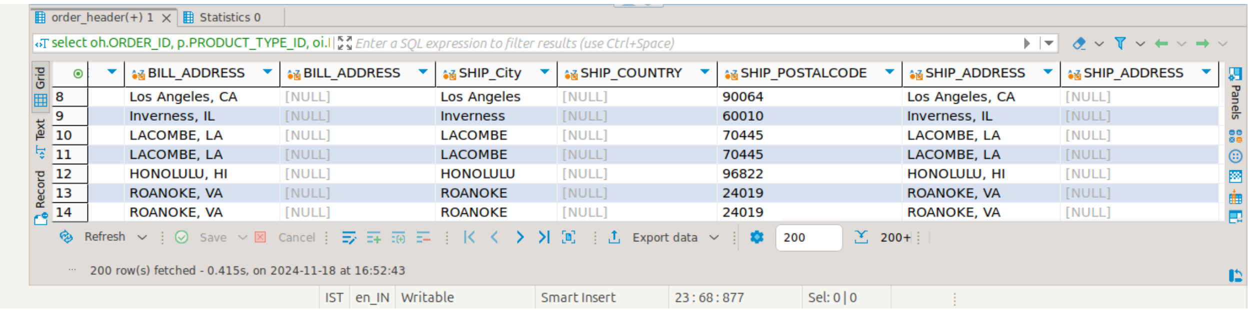
```
200 row(s) fetched - 0.079s, on 2024-11-18 at 15:59:16
IST  en_IN  Writable  Smart Insert  9 : 8 : 345  Sel: 0 | 0
```

10. Fetch all the order items that are in the created status and the order type should be a sales order

Query :-

```
select oh.ORDER_ID,
    p.PRODUCT_TYPE_ID,
    oi.EXTERNAL_ID as ORDER_LINE_ID,
    oh.EXTERNAL_ID,
    oh.SALES_CHANNEL_ENUM_ID as SALES_CHANNEL,
    oi.QUANTITY,
    oi.STATUS_ID,
    p.PRODUCT_ID,
    ocm.CONTACT_MECH_PURPOSE_TYPE_ID,
    pa.city as Bill_City,
    tn.COUNTRY_CODE as BILL_COUNTRY,
    pa.POSTAL_CODE as BILL_POSTALCODE,
    pa.ADDRESS1 as BILL_ADDRESS,
      pa.ADDRESS2 as BILL_ADDRESS,
      pa.city as SHIP_City,
      tn.COUNTRY_CODE as SHIP_COUNTRY,
      pa.POSTAL_CODE as SHIP_POSTALCODE,
      pa.ADDRESS1 as SHIP_ADDRESS,
      pa.ADDRESS2 as SHIP_ADDRESS
from order_header oh join order_item oi on oh.ORDER_ID = oi.ORDER_ID
join product p on p.PRODUCT_ID = oi.PRODUCT_ID
join order_contact_mech ocm on ocm.CONTACT_MECH_ID = oh.ORDER_ID
join postal_address pa on ocm.CONTACT_MECH_ID = pa.CONTACT_MECH_ID
join telecom_number tn  on ocm.CONTACT_MECH_ID = pa.CONTACT_MECH_ID
where oi.STATUS_ID = 'item_created' and oh.ORDER_TYPE_ID = 'sales_order';
```

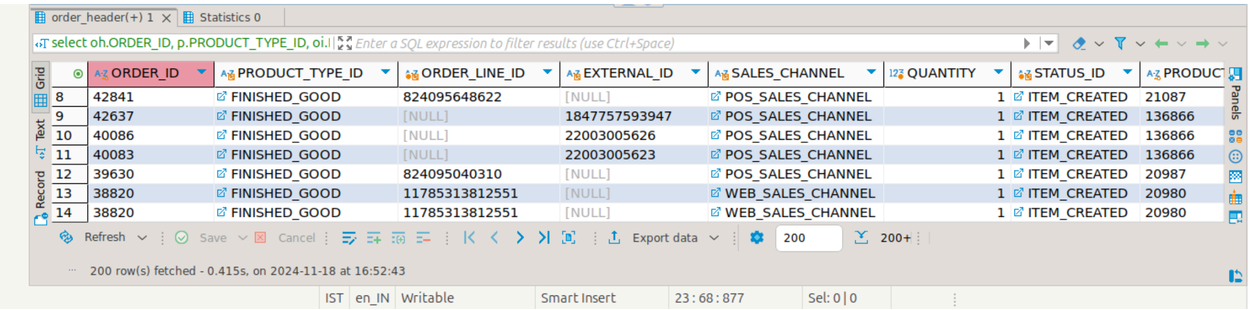
Result :-



select oh.ORDER_ID, p.PRODUCT_TYPE_ID, oi.

		BILL_ADDRESS	BILL_ADDRESS	SHIP_City	SHIP_COUNTRY	SHIP_POSTALCODE	SHIP_ADDRESS	SHIP_ADDRESS
8		Los Angeles, CA	[NULL]	Los Angeles	[NULL]	90064	Los Angeles, CA	[NULL]
9		Inverness, IL	[NULL]	Inverness	[NULL]	60010	Inverness, IL	[NULL]
10		LACOMBE, LA	[NULL]	LACOMBE	[NULL]	70445	LACOMBE, LA	[NULL]
11		LACOMBE, LA	[NULL]	LACOMBE	[NULL]	70445	LACOMBE, LA	[NULL]
12		HONOLULU, HI	[NULL]	HONOLULU	[NULL]	96822	HONOLULU, HI	[NULL]
13		ROANOKE, VA	[NULL]	ROANOKE	[NULL]	24019	ROANOKE, VA	[NULL]
14		ROANOKE, VA	[NULL]	ROANOKE	[NULL]	24019	ROANOKE, VA	[NULL]

200 row(s) fetched - 0.415s, on 2024-11-18 at 16:52:43



select oh.ORDER_ID, p.PRODUCT_TYPE_ID, oi.

	ORDER_ID	PRODUCT_TYPE_ID	ORDER_LINE_ID	EXTERNAL_ID	SALES_CHANNEL	QUANTITY	STATUS_ID	PRODUC
8	42841	FINISHED_GOOD	824095648622	[NULL]	POS_SALES_CHANNEL	1	ITEM_CREATED	21087
9	42637	FINISHED_GOOD	[NULL]	1847757593947	POS_SALES_CHANNEL	1	ITEM_CREATED	136866
10	40086	FINISHED_GOOD	[NULL]	22003005626	POS_SALES_CHANNEL	1	ITEM_CREATED	136866
11	40083	FINISHED_GOOD	[NULL]	22003005623	POS_SALES_CHANNEL	1	ITEM_CREATED	136866
12	39630	FINISHED_GOOD	824095040310	[NULL]	POS_SALES_CHANNEL	1	ITEM_CREATED	20987
13	38820	FINISHED_GOOD	11785313812551	[NULL]	WEB_SALES_CHANNEL	1	ITEM_CREATED	20980
14	38820	FINISHED_GOOD	11785313812551	[NULL]	WEB_SALES_CHANNEL	1	ITEM_CREATED	20980

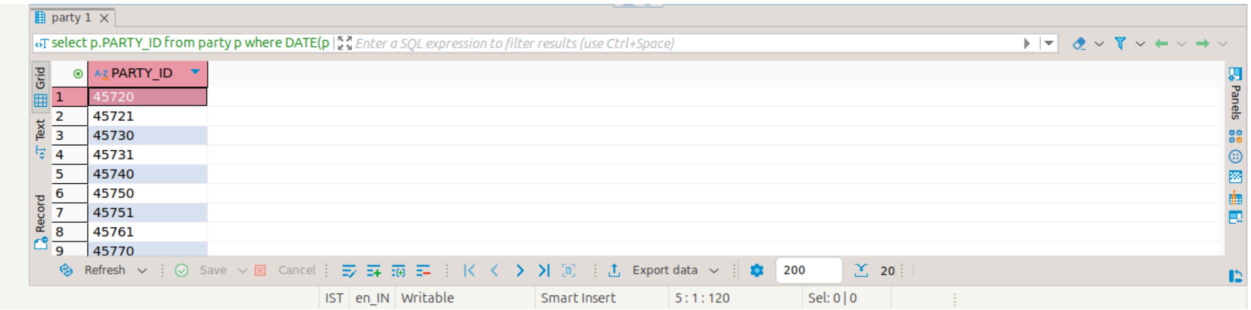
200 row(s) fetched - 0.415s, on 2024-11-18 at 16:52:43

11. Fetch all the customers created in June 2023.

Query:-

```
select p.PARTY_ID
from party p  where DATE(p.CREATED_DATE) BETWEEN '2023-06-01' AND
'2023-06-30'
```

Result :-



```
select p.PARTY_ID from party p where DATE(p
```

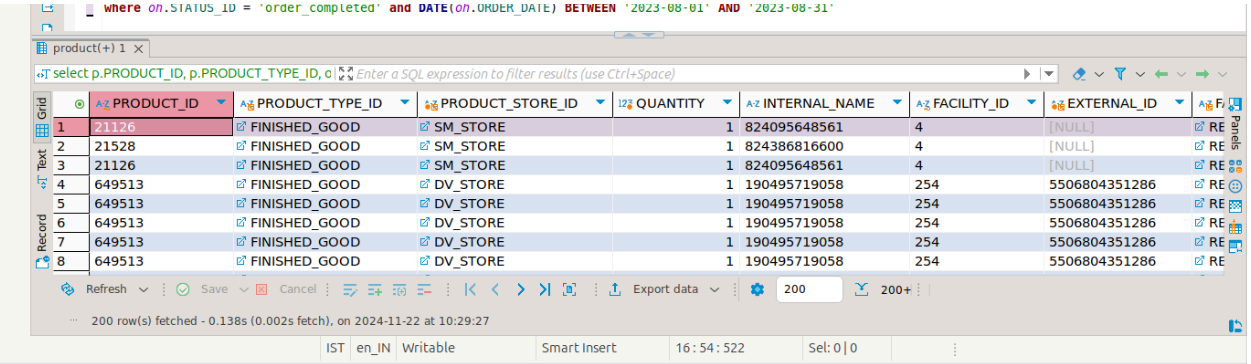
	PARTY_ID
1	45720
2	45721
3	45730
4	45731
5	45740
6	45750
7	45751
8	45761
9	45770

12. Fetch the following details for orders completed in August of 2023.

Update query:-

```
select p.PRODUCT_ID,
    p.PRODUCT_TYPE_ID,
    oh.PRODUCT_STORE_ID,
    oi.QUANTITY,
    p.INTERNAL_NAME,
    f.FACILITY_ID,
    oh.EXTERNAL_ID,
    f.FACILITY_TYPE_ID,
    oh2.ORDER_HISTORY_ID,
    oh2.ORDER_ID,
    oh2.ORDER_ITEM_SEQ_ID,
    oh2.SHIP_GROUP_SEQ_ID
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_item_ship_group oisg on oisg.ORDER_ID = oh.ORDER_ID
join facility f on f.FACILITY_ID  = oisg.FACILITY_ID
join order_history oh2 on oh2.ORDER_ID = oh.ORDER_ID
where oh.STATUS_ID = 'order_completed' and DATE(oh.ORDER_DATE) BETWEEN
'2023-08-01' AND '2023-08-31'
```

Result:-



```
where oh.STATUS_ID = 'order_completed' and DATE(oh.ORDER_DATE) BETWEEN '2023-08-01' AND '2023-08-31'
```

product(+) 1
select p.PRODUCT_ID, p.PRODUCT_TYPE_ID, o
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
Record
Panels

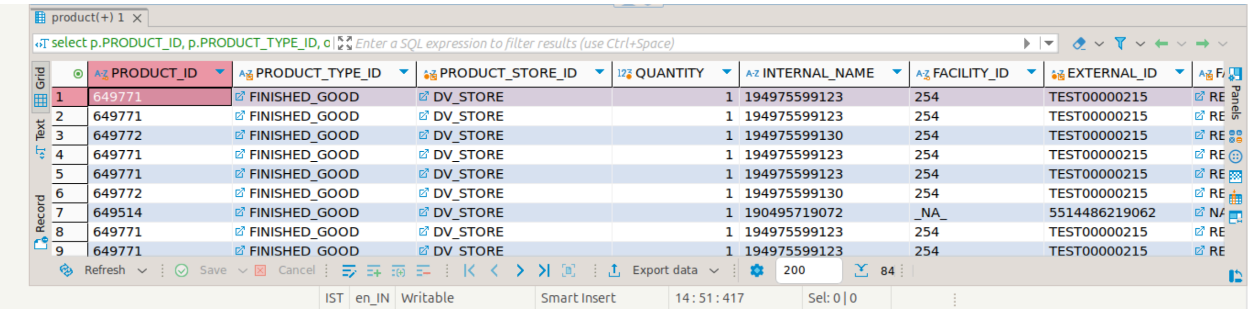
	PRODUCT_ID	PRODUCT_TYPE_ID	PRODUCT_STORE_ID	QUANTITY	INTERNAL_NAME	FACILITY_ID	EXTERNAL_ID	F/
1	21126	FINISHED_GOOD	SM_STORE	1	824095648561	4	[NULL]	RE
2	21528	FINISHED_GOOD	SM_STORE	1	824386816600	4	[NULL]	RE
3	21126	FINISHED_GOOD	SM_STORE	1	824095648561	4	[NULL]	RE
4	649513	FINISHED_GOOD	DV_STORE	1	190495719058	254	5506804351286	RE
5	649513	FINISHED_GOOD	DV_STORE	1	190495719058	254	5506804351286	RE
6	649513	FINISHED_GOOD	DV_STORE	1	190495719058	254	5506804351286	RE
7	649513	FINISHED_GOOD	DV_STORE	1	190495719058	254	5506804351286	RE
8	649513	FINISHED_GOOD	DV_STORE	1	190495719058	254	5506804351286	RE

```
200 row(s) fetched - 0.138s (0.002s fetch), on 2024-11-22 at 10:29:27
```

Query:-

```
select p.PRODUCT_ID,
    p.PRODUCT_TYPE_ID,
    oh.PRODUCT_STORE_ID,
    oi.QUANTITY,
    p.INTERNAL_NAME,
    f.FACILITY_ID,
    oh.EXTERNAL_ID,
    f.FACILITY_TYPE_ID,
    oh2.ORDER_HISTORY_ID,
    oh2.ORDER_ID,
    oh2.ORDER_ITEM_SEQ_ID,
    oh2.SHIP_GROUP_SEQ_ID
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 facility f on f.FACILITY_ID  = oh.ORIGIN_FACILITY_ID
join order_history oh2 on oh2.ORDER_ID = oh.ORDER_ID
where oh.STATUS_ID = 'order_completed' and DATE(oh.ORDER_DATE) BETWEEN
'2023-08-01' AND '2023-08-31'
```

Result :-



product(+) 1
select p.PRODUCT_ID, p.PRODUCT_TYPE_ID, o
Enter a SQL expression to filter results (use Ctrl+Space)
Grid
Text
Record
Panels

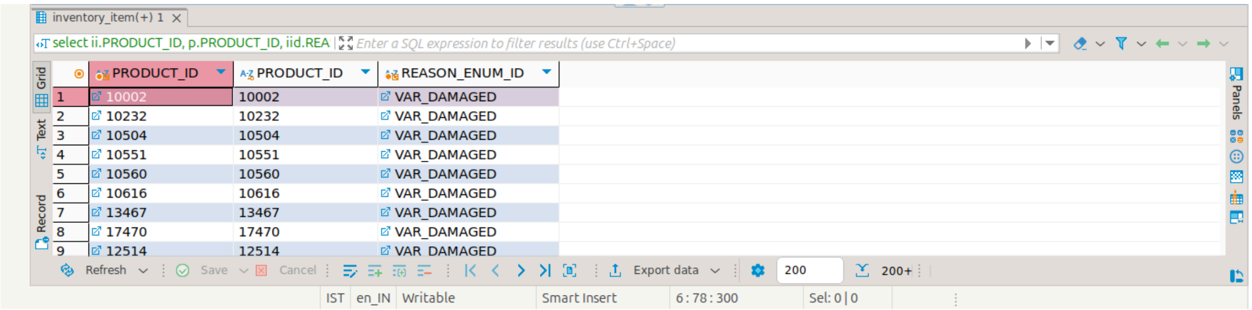
	PRODUCT_ID	PRODUCT_TYPE_ID	PRODUCT_STORE_ID	QUANTITY	INTERNAL_NAME	FACILITY_ID	EXTERNAL_ID	F/
1	649771	FINISHED_GOOD	DV_STORE	1	194975599123	254	TEST00000215	RE
2	649771	FINISHED_GOOD	DV_STORE	1	194975599123	254	TEST00000215	RE
3	649772	FINISHED_GOOD	DV_STORE	1	194975599130	254	TEST00000215	RE
4	649771	FINISHED_GOOD	DV_STORE	1	194975599123	254	TEST00000215	RE
5	649771	FINISHED_GOOD	DV_STORE	1	194975599123	254	TEST00000215	RE
6	649772	FINISHED_GOOD	DV_STORE	1	194975599130	254	TEST00000215	RE
7	649514	FINISHED_GOOD	DV_STORE	1	190495719072	_NA_	5514486219062	NA
8	649771	FINISHED_GOOD	DV_STORE	1	194975599123	254	TEST00000215	RE
9	649771	FINISHED_GOOD	DV_STORE	1	194975599123	254	TEST00000215	RE

13. Fetch the inventory variances of the products where the reason is ‘VAR_LOST’ or VAR_DAMAGED.

Query:-

```
select ii.PRODUCT_ID,
     p.PRODUCT_ID,
     iid.REASON_ENUM_ID
from inventory_item ii join product p on ii.PRODUCT_ID = p.PRODUCT_ID
join inventory_item_detail iid on iid.INVENTORY_ITEM_ID = ii.INVENTORY_ITEM_ID
where iid.REASON_ENUM_ID = 'VAR_LOST' or iid.REASON_ENUM_ID =
'VAR_DAMAGED';
```

Result :-



```
select ii.PRODUCT_ID, p.PRODUCT_ID, iid.REA
```

Enter a SQL expression to filter results (use Ctrl+Space)

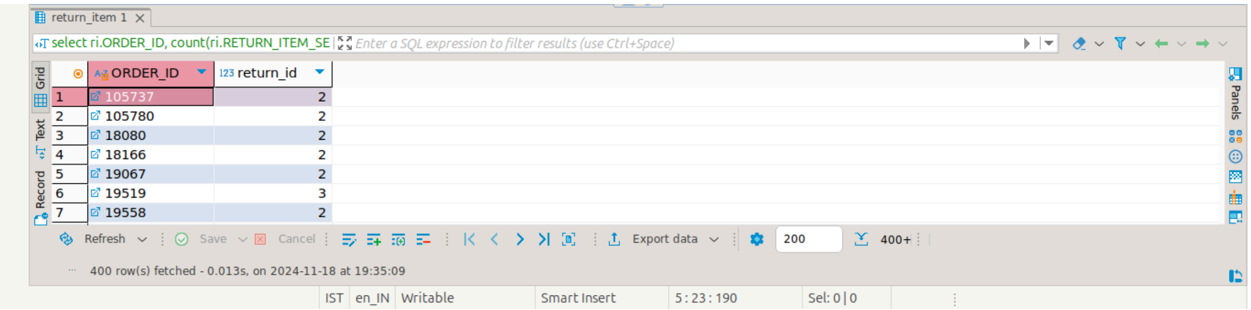
	PRODUCT_ID	PRODUCT_ID	REASON_ENUM_ID
1	10002	10002	VAR_DAMAGED
2	10232	10232	VAR_DAMAGED
3	10504	10504	VAR_DAMAGED
4	10551	10551	VAR_DAMAGED
5	10560	10560	VAR_DAMAGED
6	10616	10616	VAR_DAMAGED
7	13467	13467	VAR_DAMAGED
8	17470	17470	VAR_DAMAGED
9	12514	12514	VAR_DAMAGED

14. Find all the orders that have more than one return.

Query:-

```
select ri.ORDER_ID,
     count(ri.RETURN_ITEM_SEQ_ID) as return_id
  from return_item ri join order_header oh on oh.ORDER_ID = ri.ORDER_ID
 group by ri.ORDER_ID
having return_id >1
```

Result:-



return_item 1

```
select ri.ORDER_ID, count(ri.RETURN_ITEM_SE
```

	ORDER_ID	return_id
1	105737	2
2	105780	2
3	18080	2
4	18166	2
5	19067	2
6	19519	3
7	19558	2

400 row(s) fetched - 0.013s, on 2024-11-18 at 19:35:09

IST | en_IN | Writable | Smart Insert | 5 : 23 : 190 | Sel: 0 | 0

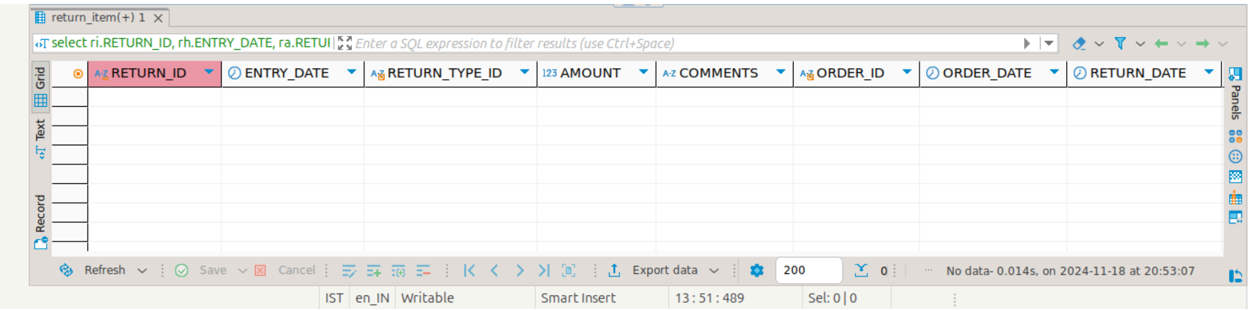
15. Get all the appeasements in July month.

- **How do we differentiate between returns and appeasements?**
  1. **Returns:**

     Typically involves goods being returned by the customer. Returns focus on reversing the transaction for a specific product or order.
  2. **Appeasement :**

     Appeasements are compensations or discounts given to customers to make up for a problem or inconvenience they faced, without requiring them to return the product.
     1. **For example:**
        - If a customer receives a damaged item and decides to keep it, the company might offer a partial refund or discount as an appeasement.
        - If there's a delay in delivery, the company might offer a voucher or credit to apologize.
- **Get all the below fields.**

Result:-



return_item(+) 1

```
select ri.RETURN_ID, rh.ENTRY_DATE, ra.RETUI
```

RETURN_ID	ENTRY_DATE	RETURN_TYPE_ID	AMOUNT	COMMENTS	ORDER_ID	ORDER_DATE	RETURN_DATE

No data- 0.014s, on 2024-11-18 at 20:53:07

IST | en_IN | Writable | Smart Insert | 13 : 51 : 489 | Sel: 0 | 0

Query:-

```
select ri.RETURN_ID,
    rh.ENTRY_DATE,
    ra.RETURN_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 rh.RETURN_ID = ri.RETURN_ID
join return_adjustment ra on ra.RETURN_ITEM_SEQ_ID = ri.RETURN_ITEM_SEQ_ID
join return_adjustment_type rat on rat.RETURN_ADJUSTMENT_TYPE_ID =
ra.RETURN_ADJUSTMENT_TYPE_ID
join order_header oh on oh.ORDER_ID = ri.ORDER_ID
where DATE (rh.RETURN_DATE) between '2024-07-01' AND '2024-07-31'
and rat.RETURN_ADJUSTMENT_TYPE_ID = 'appeasement'
```