SQL ASSIGNMENT 1

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`

Solution :

```
SELECT p.PARTY_ID,

p.FIRST_NAME,

p.LAST_NAME,

pr.ROLE_TYPE_ID AS Role_Type,

cm.INFO_STRING AS EmailAddress,

t.CONTACT_NUMBER AS Phone,

p.CREATED_STAMP
FROM PERSON p
JOIN PARTY_ROLE pr ON p.PARTY_ID = pr.PARTY_ID
LEFT JOIN PARTY_CONTACT_MECH pcm ON p.PARTY_ID = pcm.PARTY_ID
LEFT JOIN CONTACT_MECH cm ON pcm.CONTACT_MECH_ID = cm.CONTACT_MECH_ID
LEFT JOIN TELECOM_NUMBER t ON pcm.CONTACT_MECH_ID = t.CONTACT_MECH_ID
WHERE pr.ROLE_TYPE_ID = 'CUSTOMER'
AND p.CREATED_STAMP 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

Solution :

```
SELECT p.PRODUCT_ID,
        pt.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)

Solution:

```
SELECT

  p.product_id,

  p.internal_name,

  p.product_type_id,

  gi.id_value as NetSuitId

FROM product p left join good_identification gi on
p.product_id=gi.product_id

where gi.id_value is null OR gi.id_value='' and
gi.good_identification_type_id='ERP_ID';
```

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)

Solution:

```
SELECT

   p.product_id,

  gi.id_value as NetSuitId,

  sp.shopify_product_id as shopify_id
```

```
FROM product p left join good_identification gi on p.product_id=gi.product_id

join shopify_product sp on sp.product_id=p.product_id

where gi.id_value is not null and 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

Solution:

```
SELECT

    distinct oi.product_id,

    p.product_type_id,

    p.internal_name,

    pf.facility_id,

    f.facility_type_id,

    oh.order_history_id,
    oh.order_item_seq_id,
    oh.ship_group_seq_id,
    orh.status_id,
    oi.order_id,
    oi.quantity,
    orh.external_id,
    orh.product_store_id
FROM order_item oi
JOIN product p ON oi.product_id = p.product_id
JOIN order_item_ship_group oisg ON oisg.order_id=oi.order_id
JOIN order_header orh ON oi.order_id = orh.order_id
JOIN product_facility pf ON pf.product_id=oi.product_id AND
oisg.facility_id=pf.facility_id
JOIN facility f ON f.facility_id=oisg.facility_id
JOIN order_history oh ON oi.order_id = oh.order_id
WHERE orh.STATUS_ID='ORDER_COMPLETED' AND oh.CREATED_DATE
>='2023-08-01' AND oh.CREATED_DATE<='2023-09-01';
```

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)

Solution:

```
SELECT
        o.ORDER_ID,
        o.grand_total AS TOTAL_AMOUNT,
        opp.PAYMENT_METHOD_TYPE_ID AS PAYMENT_METHOD,
        oi.EXTERNAL_ID AS SHOPIFY_ID
FROM
        ORDER_HEADER o
JOIN ORDER_PAYMENT_PREFERENCE opp
        ON o.ORDER_ID = opp.ORDER_ID
JOIN ORDER_ITEM o
        ON o.ORDER_ID = oi.ORDER_ID
WHERE
        o.status_id = 'ORDER_CREATED'
        AND o.order_type_id = 'SALES_ORDER'
ORDER BY
        o.ORDER_DATE DESC limit 10;
```

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

Solution:

```
SELECT

        o.ORDER_ID,
        o.STATUS_ID AS ORDER_STATUS,
        opp.STATUS_ID AS PAYMENT_STATUS ,
        s.STATUS_ID  AS SHIPMENT_STATUS
FROM
        ORDER_HEADER o
JOIN ORDER_PAYMENT_PREFERENCE opp
        ON o.ORDER_ID = opp.ORDER_ID
JOIN SHIPMENT s
        ON o.ORDER_ID = s.PRIMARY_ORDER_ID
WHERE
        opp.STATUS_ID = 'PAYMENT_SETTLED'
```

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

Solution:

```
 SELECT
        HOUR(o.STATUS_DATETIME) AS HOUR,
        COUNT(o.ORDER_ID) AS TOTAL_ORDERS
FROM
        ORDER_STATUS o
WHERE
        o.STATUS_ID = 'ORDER_COMPLETED'
GROUP BY
        HOUR(o.STATUS_DATETIME)
ORDER BY
        HOUR(o.STATUS_DATETIME);
```

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

Solution:

```
SELECT count(o.order_id) as TOTAL_ORDER,
sum(o.grand_total)as TOTAL_REVENUE
FROM order_item_ship_group og JOIN order_header o
on
og.order_Id=o.order_Id where og.shipment_method_type_id='STOREPICKUP'
and o.order_date >= curdate()- INTERVAL 1 YEAR;
```

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

Solution:

```
SELECT
        COUNT(os.order_id) AS total_orders,
  os.change_reason AS cancellation_reason

FROM
```

```
order_status os  WHERE
os.STATUS_ID='ORDER_CANCELLED'AND os.status_datetime>=curdate()- INTERVAL 1
MONTH
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

Solution:

```
select

   distinct p.product_id,

  pf.minimum_stock as Threshold

from product p

join product_facility pf on p.product_id=pf.product_id;
```