MySkill SQL Showcase

By : Immanuel Mayerd

The *introduction*

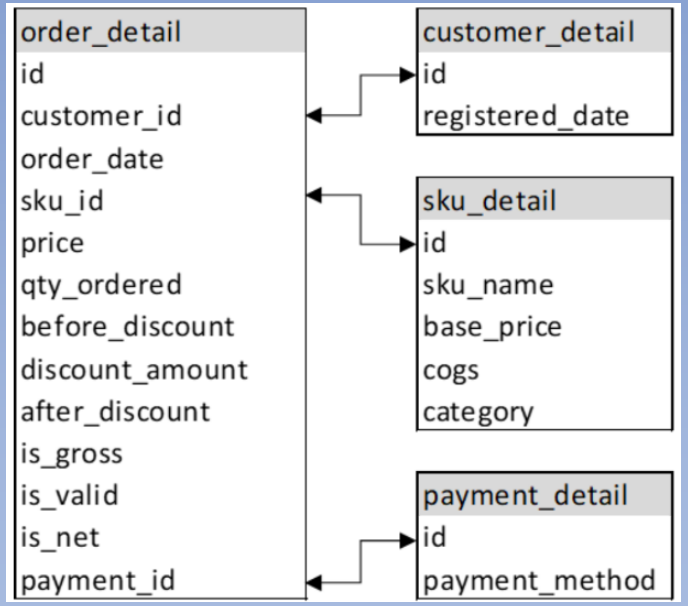
As a data analyst in an e-commerce company, I have been tasked with conducting an analysis to address five critical business questions for the company.

Dataset

The dataset consists of four tables, namely:

- order_detail
- sku_detail
- customer_detail
- payment_detail

The data provided is dummy data used for training purposes in data analysis.



Dataset

Order_detail

1. id: Unique number for the order/id_order.
2. customer_id: Unique number for the customer.
3. order_date: Date when the transaction took place.
4. sku_id: Unique number for the product (SKU - Stock Keeping Unit).
5. price: Price listed on the price tag.
6. qty_ordered: Quantity of items purchased by the customer.
7. before_discount: Total value of the product (price * qty_ordered) before any discount.
8. discount_amount: Total discount amount for the product.
9. after_discount: Total value of the product after deducting the discount.
10. is_gross: Indicates whether the customer has not paid for the order.
11. is_valid: Indicates whether the customer has made the payment.
12. is_net: Indicates whether the transaction is completed.
13. payment_id: Unique number for the payment method.

Dataset

sku_detail:

1. id: Unique number for the product (can be used as a key when joining).
2. sku_name: Name of the product.
3. base_price: Price of the item listed on the price tag.
4. cogs: Cost of goods sold, total cost to sell one product.
5. category: Product category.

customer_detail:

1. id: Unique number for the customer.
2. registered_date: Date when the customer started registering as a member.

payment_detail:

1. id: Unique number for the payment method.
2. payment_method: Payment method used.

Question 1

During the transactions that occurred throughout 2021, in which month did the total transaction value (after_discount) reach its highest? Use is_valid = 1 to filter transaction data.

Query 1

```sql
-- Pertanyaan 1
-- Q: Selama transaksi yang terjadi selama 2021, pada bulan apa total nilai transaksi
-- (after_discount) paling besar? Gunakan is_valid = 1 untuk memfilter data transaksi.

select
  extract(month from order_date) as month_2021,
  sum(after_discount) as total_sales
from
  `dataset_fp.order_detail`
where
  extract (year from order_date)=2021 and is_valid = 1
group by 1
order by 2 desc limit 1
```

This query retrieves the total sales for each month in 2021 from the 'dataset_fp.order_detail' table, considering only valid orders. It extracts the month from the 'order_date', sums the 'after_discount' values, and then groups the results by month. Finally, it orders the results by total sales in descending order and limits the output to the top result.

Result 1

Query results

JOB INFORMATION | RESULTS | CHA

Row	month_2021	total_sales
1	8	227862744.0

Based on the previous query results, the highest total transaction value (after_discount) in the year 2021 occurred in August (month 8) with a transaction value of 227,862,744.

Question 2

During transactions in the year 2022, which category generated the highest transaction value? Please use is_valid = 1 to filter transaction data.

Query 2

```sql
-- Pertanyaan 2
-- Q: Selama transaksi pada tahun 2022, kategori apa yang menghasilkan nilai transaksi paling
-- besar? Gunakan is_valid = 1 untuk memfilter data transaksi.

select
  category,
  sum(after_discount) total_transaction
from
  `dataset_fp.order_detail` od
left join
  `dataset_fp.sku_detail` sd
on
  od.sku_id = sd.id
where
  extract(year from order_date) = 2022
and
  is_valid = 1
group by 1
order by 2 desc
limit 1;
```

This query selects the category and calculates the total transaction value by summing 'after_discount' from the 'dataset_fp.order_detail' table. It includes a left join with 'dataset_fp.sku_detail' using the 'sku_id' and 'id' columns. The data is filtered for the year 2022 and valid transactions (is_valid = 1). The results are then grouped by category, ordered in descending order by total transaction value, and limited to the top result.

Result 2

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	category	total_transaction
1	Mobiles & Tablets	918451576.0

Based on the previous query results, the highest transaction value (after_discount) during the year 2022 is in the 'Mobiles & Tablets' category, with a value of 918,451,576.

Question 3

Compare the transaction values for each category in 2021 and 2022. Identify the categories that experienced an increase and those that experienced a decrease in transaction values from 2021 to 2022. Use is_valid = 1 to filter transaction data.

Query 3

```sql
-- Pertanyaan 3
-- Q: Bandingkan nilai transaksi dari masing-masing kategori pada tahun 2021 dengan 2022.
-- Sebutkan kategori apa saja yang mengalami peningkatan dan kategori apa yang mengalami
-- penurunan nilai transaksi dari tahun 2021 ke 2022. Gunakan is_valid = 1 untuk memfilter data
-- transaksi.

with yearly_trx AS (
  select
    category,
    sum(case when extract(year from order_date) = 2021 then after_discount end) as total_transaction_2021,
    sum(case when extract(year from order_date) = 2022 then after_discount end) as total_transaction_2022
  from `dataset_fp.order_detail` as od
  left join `dataset_fp.sku_detail` as sd
  on od.sku_id = sd.id
  where is_valid = 1
  group by category
)

select
  category,
  total_transaction_2021,
  total_transaction_2022,
  (total_transaction_2022 - total_transaction_2021) change,
  case
    when total_transaction_2022 > total_transaction_2021 then 'Inc'
    else 'Dec'
  end as performance
from yearly_trx
order by 4 desc;
```

This query uses a common table expression (CTE) named ‘yearly_trx' to calculate the total transaction values for each category in both 2021 and 2022. It employs a left join between 'dataset_fp.order_detail' and 'dataset_fp.sku_detail' tables based on 'sku_id' and 'id'. The results are then grouped by category.

The main query selects the category, total transaction values for 2021 and 2022, the change in values, and a performance indicator ('Inc' for increase, 'Dec' for decrease) based on the comparison. The results are ordered in descending order of change in transaction values. The query provides insights into which categories had increased or decreased transaction values from 2021 to 2022, considering only valid transactions (is_valid = 1).

Result 3

Query results

JOB INFORMATION | RESULTS | CHART PREVIEW | JSON | EXECUTION DETAILS | EXECUTION GRA

Row	category	total_transaction_2021	total_transaction_2022	change	performance
1	Mobiles & Tablets	370606718.0	918451576.0	547844858.0	Inc
2	Entertainment	162326357.4	365344148.9	203017791.4999...	Inc
3	Appliances	218550177.0	316358100.0	97807923.0	Inc
4	Men Fashion	58628198.0	135588253.0	76960055.0	Inc
5	Computing	172878860.0	214028543.4	41149683.40000...	Inc
6	Home & Living	45797873.0	79483716.2	33685843.2	Inc
7	Health & Sports	33837965.6	54235579.5999999...	20397613.99999...	Inc
8	Women Fashion	84045961.4	93014970.62	8969009.219999...	Inc
9	School & Education	11558982.4	17362465.2999999...	5803482.899999...	Inc
10	Superstore	28828088.0	32643266.5200000...	3815178.520000...	Inc
11	Soghaat	15056202.6000000...	17658332.0	2602129.399999...	Inc
12	Kids & Baby	23971057.7999999...	25931276.84	1960219.040000...	Inc
13	Beauty & Grooming	46047360.0	46211019.18	163659.1799999...	Inc
14	Books	10124596.0	6792519.2	-3332076.8	Dec
15	Others	40468515.7399999...	21744646.02	-18723869.7199...	Dec

Based on the previous query results, it was found that only the 'Books' and 'Others' categories experienced a decrease in transaction values from 2021 to 2022.

Question 4

Display the top 5 most popular payment methods used during 2022 based on the total number of unique orders. Use is_valid = 1 to filter transaction data.

Query 4

```sql
-- Pertanyaan 4
-- Q: Tampilkan top 5 metode pembayaran yang paling populer digunakan selama 2022
-- (berdasarkan total unique order). Gunakan is_valid = 1 untuk memfilter data transaksi.

select
  pd.payment_method,
  count(distinct od.id) total_order
from `dataset_fp.order_detail` od
left join `dataset_fp.payment_detail` pd
on od.payment_id = pd.id
where
  extract(year from od.order_date) = 2022 and is_valid = 1
group by 1
order by 2 desc
limit 5;
```

This query selects the payment method and counts the total number of distinct orders for each method from the 'dataset_fp.order_detail' table. It includes a left join with 'dataset_fp.payment_detail' using the 'payment_id' and 'id' columns. The data is filtered for the year 2022 and valid transactions (is_valid = 1). The results are then grouped by payment method, ordered in descending order by the total number of orders, and limited to the top 5 most popular payment methods.

Result 4

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	payment_method	total_order
1	cod	1809
2	Payaxis	181
3	customercredit	75
4	Easypay	69
5	jazzwallet	26

Based on the previous query, the top 5 most popular payment methods used during 2022 based on the total number of unique orders are COD, Payaxis, customercredit, Easypay, and Jazzwallet.

Question 5

Sort these 5 products based on their transaction values.

- Samsung
- Apple
- Sony
- Huawei
- Lenovo

Use is_valid = 1 to filter transaction data.

Query 5

```sql
-- Pertanyaan 5
-- Q: Urutkan dari ke-5 produk ini berdasarkan nilai transaksinya.
-- Samsung, Apple, Sony, Huawei, Lenovo
-- Gunakan is_valid = 1 untuk memfilter data transaksi.

with regex as (
  select
    case
      when lower(sd.sku_name) like '%samsung%' then 'Samsung'
      when lower(sd.sku_name) like '%sony%' then 'Sony'
      when lower(sd.sku_name) like '%huawei%' then 'Huawei'
      when lower(sd.sku_name) like '%lenovo%' then 'Lenovo'
      when regexp_contains(lower(sd.sku_name), r'(apple|imac|macbook|iphone)') then 'Apple'
    end as brand_name,
    sum(od.after_discount) as transaction_value
  from `dataset_fp.order_detail` od
  left join `dataset_fp.sku_detail` sd
  on od.sku_id = sd.id
  where is_valid = 1
  group by 1
)

select
  brand_name,
  transaction_value
from
  regex
where
  brand_name is not null
order by 2 desc
```

This query utilizes a common table expression (CTE) named 'regex' to categorize products into brands (Samsung, Sony, Huawei, Lenovo, and Apple) based on specific keywords or patterns in the product names. It also calculates the total transaction values for each brand from the 'dataset_fp.order_detail' table, with a left join to 'dataset_fp.sku_detail' using 'sku_id' and 'id'.

The main query then selects the brand name and transaction value from the 'regex' CTE, filtering out null brand names. The results are ordered in descending order by transaction value, providing a sorted list of the 5 products based on their respective transaction values, considering only valid transactions (is_valid = 1).

Result 5

Query results

JOB INFORMATION | RESULTS | CHART | PREVIEW

Row	brand_name	transaction_value
1	Samsung	588764148.0
2	Apple	444855360.0
3	Sony	63960718.0
4	Huawei	63160260.0
5	Lenovo	62379800.4

Based on the results of the previous query, the brand order based on transactions from highest to lowest is Samsung, Apple, Sony, Huawei, and Lenovo.

Thankyou