\*\* Viewing all tables one by one -

select \* from Customer;

select \* from Item;

select \* from CustomerTransactionData;

select \* from CouponMapping;

select \* from City;

select \* from Campaign;

**Section 1: Getting the overview of the data**

**Q1. Checking the cardinality of following columns:**

a) Different colour segments (categories) provided by the company.

SELECT COUNT(DISTINCT(Item\_Category))

FROM Item;

b) Different Coupon Types that are offered.

SELECT COUNT(DISTINCT(couponType))

FROM CouponMapping;

c) States where the company is currently delivering its products and services.

SELECT COUNT(DISTINCT (State))

FROM CityData;

d) Different Order Types.

SELECT COUNT(DISTINCT (OrderType))

FROM CustomerTransactionData;

**Q2. Identifying total number of sales (transactions) happened by**

**-- Yearly basis**

SELECT YEAR(PurchaseDate), COUNT(Trans\_Id) AS Total\_Transactions

FROM CustomerTransactionData

GROUP BY YEAR(PurchaseDate)

ORDER BY YEAR(PurchaseDate);

**-- Quarterly basis**

SELECT EXTRACT(YEAR FROM PurchaseDate), EXTRACT(MONTH FROM PurchaseDate), COUNT(Trans\_Id) AS Total\_Trans

FROM CustomerTransactionData

GROUP BY EXTRACT(YEAR FROM PurchaseDate),EXTRACT(MONTH FROM PurchaseDate);

**-- Yearly and Monthly basis**

SELECT EXTRACT(YEAR FROM PurchaseDate), EXTRACT(MONTH FROM PurchaseDate),

COUNT(Trans\_Id) AS Total\_Trans

FROM CustomerTransactionData

GROUP BY EXTRACT(YEAR FROM PurchaseDate),EXTRACT(MONTH FROM PurchaseDate);

**Q3. Identifying the total purchase order by**

**-- Product category**

SELECT Item\_Category, Round(SUM(PurchasingAmt)) AS Total\_Puchase

FROM CustomerTransactionData CTD

JOIN Item I ON CTD.item\_id = I.Item\_Id

GROUP BY Item\_Category

ORDER BY Total\_Puchase;

**-- Yearly and Quarterly basis**

SELECT EXTRACT(YEAR FROM PurchaseDate),

EXTRACT(QUARTER FROM PurchaseDate),

SUM(PurchasingAmt) AS Total\_Sales

FROM CustomerTransactionData

GROUP BY EXTRACT(YEAR FROM PurchaseDate), EXTRACT(QUARTER FROM PurchaseDate);

**-- Order Type**

SELECT OrderType,

SUM(PurchasingAmt) AS Total\_Sales

FROM CustomerTransactionData

GROUP BY OrderType;

**-- City Tier**

SELECT CityTier, SUM(PurchasingAmt) AS Total\_Sales

FROM CustomerTransactionData AS CTD

JOIN Customer AS C ON C.Customer\_Id = CTD.Cust\_Id

JOIN CityData ON CityData.City\_Id = C.City\_Id

GROUP BY CityTier;

**Section 2: Understanding lead conversion**

-- Here in this section I have tried to understand the customer path to conversion as a potential purchaser based on our campaigns.

-- 1. Identifying the total number of transactions with campaign coupon vs total number of transactions without campaign coupon.

SELECT 'Without Coupons' AS CampaignCoupons,

COUNT(\*) AS TotalTransactions FROM CustomerTransactionData

WHERE campaign\_id IS NULL

UNION ALL

SELECT 'With Coupons' AS CampaignCoupons,

COUNT(\*) AS TotalTransactions FROM CustomerTransactionData

WHERE campaign\_id IS NOT NULL;

-- 2. Identifying the number of customers with first purchase done with or without campaign coupons.

SELECT COUNT(coupon\_id) AS TotalCustomersFirstPurchaseWcoupon, COUNT(\*) - COUNT(coupon\_id) AS TotalCustomersFirstPurchaseWithoutcoupon

FROM CustomerTransactionData

WHERE Trans\_Id IN (

SELECT

FIRST\_VALUE(Trans\_Id) OVER(PARTITION BY Cust\_Id ORDER BY PurchaseDate RANGE BETWEEN

UNBOUNDED PRECEDING AND

UNBOUNDED FOLLOWING) AS trans\_id

FROM CustomerTransactionData);

-- 3. Identifying the impact of campaigns on users

-- Checking the total number of unique users making purchases with or without campaign coupons.

SELECT segment, COUNT(\*) AS No\_Of\_Unique\_Customers

FROM

(SELECT DISTINCT Cust\_Id,

CASE

WHEN campaign\_id IS NULL THEN 'Without\_Campaign\_Coupon'

ELSE 'With\_Campaign\_Coupon'

END AS segment

FROM CustomerTransactionData) CTD

GROUP BY segment;

-- Checking the purchase amount with campaign coupons vs normal coupons vs no coupons.

SELECT segment, ROUND(SUM(PurchasingAmt)) AS Total\_Purchase

FROM (SELECT \*,

CASE

WHEN campaign\_id IS NOT NULL AND coupon\_id IS NOT NULL THEN 'Campaign\_Coupan'

WHEN campaign\_id IS NULL AND coupon\_id IS NOT NULL THEN 'Normal\_Coupan'

WHEN campaign\_id IS NULL AND coupon\_id IS NULL THEN 'No\_Coupan'

END AS segment

FROM

CustomerTransactionData) CTD

GROUP BY segment

ORDER BY Total\_Purchase;

“Based on the above analysis, I have observed that Campaigns were very effective in adding new customers.

They also accounted for maximum share in total revenue of the company.”

**Section 3: Understanding company growth and decline**

In this section I have tried to help the marketing team in understanding the growth and decline pattern of the company in terms of new leads or sales amount by the customers.

1. Identifying the total growth on year by year basis

a) Based on quantity of paint that is sold.

**SELECT** \*

**FROM** (

**SELECT** year\_purchase, Total\_Quantity **AS** Total\_Quantity\_2022,

LAG(Total\_Quantity) OVER(**ORDER** **BY** year\_purchase) **AS** pastoffset\_1,

LAG(Total\_Quantity,**2**) OVER(**ORDER** **BY** year\_purchase) **AS** pastoffset\_2,

LAG(Total\_Quantity,**3**) OVER(**ORDER** **BY** year\_purchase) **AS** pastoffset\_3

**FROM**(

**SELECT** **EXTRACT**(**YEAR** **FROM** PurchaseDate) **AS** year\_purchase,

**SUM**(Quantity) **AS** Total\_Quantity

**FROM** CustomerTransactionData

**WHERE** **EXTRACT**(**YEAR** **FROM** PurchaseDate) < **2023**

**GROUP** **BY** **EXTRACT**(**YEAR** **FROM** PurchaseDate)) **AS** T) **AS** T

**WHERE** year\_purchase = **2022**;

Company made a steady growth in the past 2 years and saw a declining pattern in the current year.

b) Based on amount of paint that is sold.

**SELECT** \*

**FROM** (

**SELECT** year\_purchase, Total\_Purchase **AS** Total\_Purchase\_2022,

LAG(Total\_Purchase) OVER(**ORDER** **BY** year\_purchase) **AS** pastoffset\_1,

LAG(Total\_Purchase,**2**) OVER(**ORDER** **BY** year\_purchase) **AS** pastoffset\_2,

LAG(Total\_Purchase,**3**) OVER(**ORDER** **BY** year\_purchase) **AS** pastoffset\_3

**FROM**(

**SELECT** **EXTRACT**(**YEAR** **FROM** PurchaseDate) **AS** year\_purchase,

**SUM**(PurchasingAmt) **AS** Total\_Purchase

**FROM** CustomerTransactionData

**WHERE** **EXTRACT**(**YEAR** **FROM** PurchaseDate) < **2023**

**GROUP** **BY** **EXTRACT**(**YEAR** **FROM** PurchaseDate)) **AS** T) **AS** T

**WHERE** year\_purchase = **2022**;

Company saw growth in the first 2 years and then a declining pattern since then.

c) Based on new customers that are acquired.

**SELECT** \*

**FROM** (

**SELECT** year\_purchase, NewUsers **AS** NewUsers\_2022,

LAG(NewUsers) OVER(**ORDER** **BY** year\_purchase) **AS** pastoffset\_1,

LAG(NewUsers,**2**) OVER(**ORDER** **BY** year\_purchase) **AS** pastoffset\_2,

LAG(NewUsers,**3**) OVER(**ORDER** **BY** year\_purchase) **AS** pastoffset\_3

**FROM**(

**SELECT** **EXTRACT**(**YEAR** **FROM** PurchaseDate) **AS** year\_purchase,

**COUNT**(**DISTINCT** Cust\_Id) **AS** NewUsers

**FROM** CustomerTransactionData

**WHERE** **EXTRACT**(**YEAR** **FROM** PurchaseDate) < **2023**

**GROUP** **BY** **EXTRACT**(**YEAR** **FROM** PurchaseDate)) **AS** T) **AS** T

**WHERE** year\_purchase = **2022**;

d) Segregating new customers by OrderType

**SELECT** \*

**FROM** (

**SELECT** year\_purchase, OrderType,

NewUsers **AS** NewUsers\_2022,

LAG(NewUsers) OVER(PARTITION **BY** OrderType **ORDER** **BY** year\_purchase) **AS** pastoffset\_1,

LAG(NewUsers,**2**) OVER(PARTITION **BY** OrderType **ORDER** **BY** year\_purchase) **AS** pastoffset\_2,

LAG(NewUsers,**3**) OVER(PARTITION **BY** OrderType **ORDER** **BY** year\_purchase) **AS** pastoffset\_3

**FROM**(

**SELECT** **EXTRACT**(**YEAR** **FROM** PurchaseDate) **AS** year\_purchase,

OrderType,

**COUNT**(**DISTINCT** Cust\_Id) **AS** NewUsers

**FROM** CustomerTransactionData

**WHERE** **EXTRACT**(**YEAR** **FROM** PurchaseDate) < **2023**

**GROUP** **BY** **EXTRACT**(**YEAR** **FROM** PurchaseDate), OrderType) **AS** T) **AS** T

**WHERE** year\_purchase = **2022**;

Household order type has done exceptionally well in acquiring new and retaining existing customers compared to rest of the 2 order type

**Based on the above results, we can see that the overall growth has declined in the past 1 year compared to recent years. It's very apparent from total quantities sold, total sales as well as customers acquired.**

“Making an analysis of what is happening with our customer acquisition and sales growth over the years.

It can be observed that the total quantity sold per year, first increased in the year 2020 by more than 100%.

However, it showed a negative trend in the coming years, which is 2021 and 2022.

Similar pattern can be observed in Total revenue collection and new customer acquisition.”

2. Identifying the total decline, if any, within the total sales amount on year by year basis excluding the current year.

**SELECT** \*

**FROM** (

**SELECT** year\_purchase, Total\_Purchase, Total\_Purchase - LAG(Total\_Purchase) OVER(**ORDER** **BY** year\_purchase) **AS** Profit\_Loss\_W\_PreviousYear

**FROM**(

**SELECT** **EXTRACT**(**YEAR** **FROM** PurchaseDate) **AS** year\_purchase,

**SUM**(PurchasingAmt) **AS** Total\_Purchase

**FROM** CustomerTransactionData

**WHERE** **EXTRACT**(**YEAR** **FROM** PurchaseDate) < **2023**

**GROUP** **BY** **EXTRACT**(**YEAR** **FROM** PurchaseDate)) **AS** T) **AS** T;

3. Commenting on whether we need to launch a campaign for the consumers based on the recent pattern.

Brand Awareness campaign can be launched

Seasonal campaign can be launched

“There is a significant decline in Total Sales of the company. Hence, there is definitely a need to launch more and more campaigns."

-- Finding best campaigns

SELECT

campaignType,

COUNT(\*) AS Customers\_Acquired\_From\_Campaign\_Type

FROM

CustomerTransactionData CTD

JOIN

Campaign C ON CTD.campaign\_id = C.campaign\_id

GROUP BY campaignType

ORDER BY Customers\_Acquired\_From\_Campaign\_Type DESC;

"Section 2 we have seen that campaign coupons proved beneficial in acquiring new customers and increasing total sales.

Looking at above data it is clear that Brand Awareness and Seasonal Push Campaign will prove most beneficial.”

**Section 4: Market basket analysis**

- A market basket analysis is defined as a customer’s overall buying pattern of different sets of products.

- The marketing team wants to understand customer purchasing pattern. Their proposal is if they promote the products in their next campaign,

which are bought couple of times together, then this will increase the revenue for company.

In order to answer their question, I have tried to answer these questions given below.

1. Identifying the dates when the same customer has purchased some product (same order types and different products) from the company outlets.

**SELECT** C1.Cust\_Id, C1.PurchaseDate **AS** PurchaseDate1, C2.PurchaseDate **AS** PurchaseDate **FROM** CustomerTransactionData **AS** C1 **INNER** **JOIN** CustomerTransactionData **AS** C2 **ON** C1.Cust\_Id = C2.Cust\_Id **WHERE** C1.Trans\_Id != C2.Trans\_Id **AND** C1.OrderType = C2.OrderType **AND** C1.item\_id != C2.item\_id;

1. Out of the above, identifying the same combination of products coming at least thrice sorted in descending order of their appearance.

**SELECT** CONCAT\_WS(",", C1.item\_id, C2.item\_id) **AS** Item\_Combination,

**COUNT**(\*) **AS** TotalTransaction

**FROM** CustomerTransactionData **AS** C1

**INNER** **JOIN** CustomerTransactionData **AS** C2

**ON** C1.Cust\_Id = C2.Cust\_Id

**WHERE** C1.Trans\_Id != C2.Trans\_Id

**AND** C1.OrderType = C2.OrderType

**AND** C1.item\_id != C2.item\_id

**GROUP** **BY** CONCAT\_WS(",", C1.item\_id, C2.item\_id)

**HAVING** **COUNT**(\*) >= **3**

**ORDER** **BY** **COUNT**(\*) **DESC**;

Maximum transactions happened for combination of item 13 and 16 with total transactions = 8

1. Out of the above combinations (coming thrice), checking which of these combinations are popular in different sectors (household, industrial and government).

**SELECT** C1.OrderType, CONCAT\_WS(",", C1.item\_id, C2.item\_id) **AS** Item\_Combination,

**COUNT**(\*) **AS** TotalTransaction

**FROM** CustomerTransactionData **AS** C1

**INNER** **JOIN** CustomerTransactionData **AS** C2

**ON** C1.Cust\_Id = C2.Cust\_Id

**WHERE** C1.Trans\_Id != C2.Trans\_Id

**AND** C1.OrderType = C2.OrderType

**AND** C1.item\_id != C2.item\_id

**GROUP** **BY** C1.OrderType, CONCAT\_WS(",", C1.item\_id, C2.item\_id)

**HAVING** **COUNT**(\*) >= **3**

**ORDER** **BY** **COUNT**(\*) **DESC**;

Item 16 and Item 13, Item 9 and Item 45, Item 14 and Item 35 are being bought frequently together amongst different sectors. Especially in households, we have seen quite a good demand for these products. We can try to promote all the 3 in the household sector and the first combination in all the sectors.

On the basis of above three, highlighting the combinations of products that are bought couple of times in data on the same day.

Also, identifying which combinations should be promoted/advertised together to these different sectors for maximum growth.

Overall:

1.White emulsion paint + Navy blue emulsion paint (7 times)

2. Soft red synthetic paint + Cream Emulsion paint (3 times)

3. Green sage enamel paint + Soft Green Oil Paint (3 times)

Category Wise:

1. Household - White emulsion paint + Navy blue emulsion paint

2. Household - Soft red synthetic paint + Cream Emulsion paint

These combinations should be promoted together to given sectors.

**Section 5:** **Automating tasks**

Company is thinking of launching a new campaign in upcoming months. Here I have automated the following tasks.

1. Created Functions for the following:

-- Getting the total discount, if any.

**DELIMITER** $$

**CREATE** **FUNCTION** Discount

(Quantity int, Price float, PurchasingAmt float)

**RETURNS** INT

**DETERMINISTIC**

**BEGIN**

**DECLARE** discount INT;

**SET** discount = Quantity \* Price - PurchasingAmt;

**RETURN** discount;

**END**$$

**DELIMITER** ;

-- function use

SELECT

DISCOUNT(PurchasingAmt, quantity, Price) AS Total\_Discount

FROM

CustomerTransactionData CTD

JOIN

Item I ON CTD.item\_id = I.Item\_Id;

-- Getting the days/month/year elapsed since the last purchase of a customer depending on input from user.

**DELIMITER** $$

**CREATE** **FUNCTION** Time\_Elapsed

(val varchar(**4**), date\_last\_purchase date)

**RETURNS** INT

**DETERMINISTIC**

**BEGIN**

**DECLARE** time\_elapsed INT;

**SET** time\_elapsed = IF(val='day', DATEDIFF(NOW(), date\_last\_purchase), **YEAR**(NOW()) - **YEAR**(date\_last\_purchase));

**RETURN** time\_elapsed;

**END**$$

**DELIMITER** ;

-- function use

SELECT

ELAPSED(PurchaseDate) AS Time\_Elapsed

FROM

CustomerTransactionData;

2. Creating Stored Procedures for following data validation tasks:

a. Identifying whether a particular transaction amount (purchase amount) is ‘correct’ or ‘not correct’.

(It is correct if price and quantity are used to calculate without a coupon.

In case of a coupon, the coupon amount should be deducted from the original amount given the original amount is greater

than equal to min purchase for a coupon; else you can simply calculate original amount based on quantity. )[Input is transaction id]

**DELIMITER** $$

**CREATE** **PROCEDURE** PurchaseAmountValidation (**IN** p1 varchar(**32**), **OUT** p2 varchar(**128**))

**BEGIN**

**SELECT**

IF(PurchasingAmt != totalamt, 'not correct', 'correct') **AS** message

**INTO** p2

**FROM** (

**SELECT** CT.PurchasingAmt,

IF( CT.coupon\_id **IS** **NOT** **NULL** **AND** Quantity \* Price >= Min\_Purchase, Quantity \* Price - IF(couponType != 'Flat', Quantity \* Price \* Value \* **0**.**01**, Value), Quantity \* Price) **AS** totalamt

**FROM**

Item **AS** I

**JOIN**

CustomerTransactionData **AS** CT

**ON** I.Item\_Id = CT.item\_id

**LEFT** **JOIN** CouponMapping **AS** CM

**ON** CT.coupon\_id = CM.coupon\_id

**WHERE** CT.Trans\_Id = p1) **AS** T;

**END** $$

**DELIMITER** ;

**CALL** PurchaseAmountValidation('TID00240', @p2);

**SELECT** @p2;

b. Checking if there is any customer with age < 12. Printing out the total such customers on-screen.

**ELIMITER** $$

**CREATE** **PROCEDURE** CustomerAgeValidation (**OUT** p2 varchar(**128**))

**BEGIN**

**SELECT** **COUNT**(\*)

**INTO** p2

**FROM** Customer

**WHERE** **YEAR**(NOW()) - **YEAR**(Birthdate) <= **12**;

**END** $$

**DELIMITER** ;

**CALL** CustomerAgeValidation( @p1);

**SELECT** @p1