

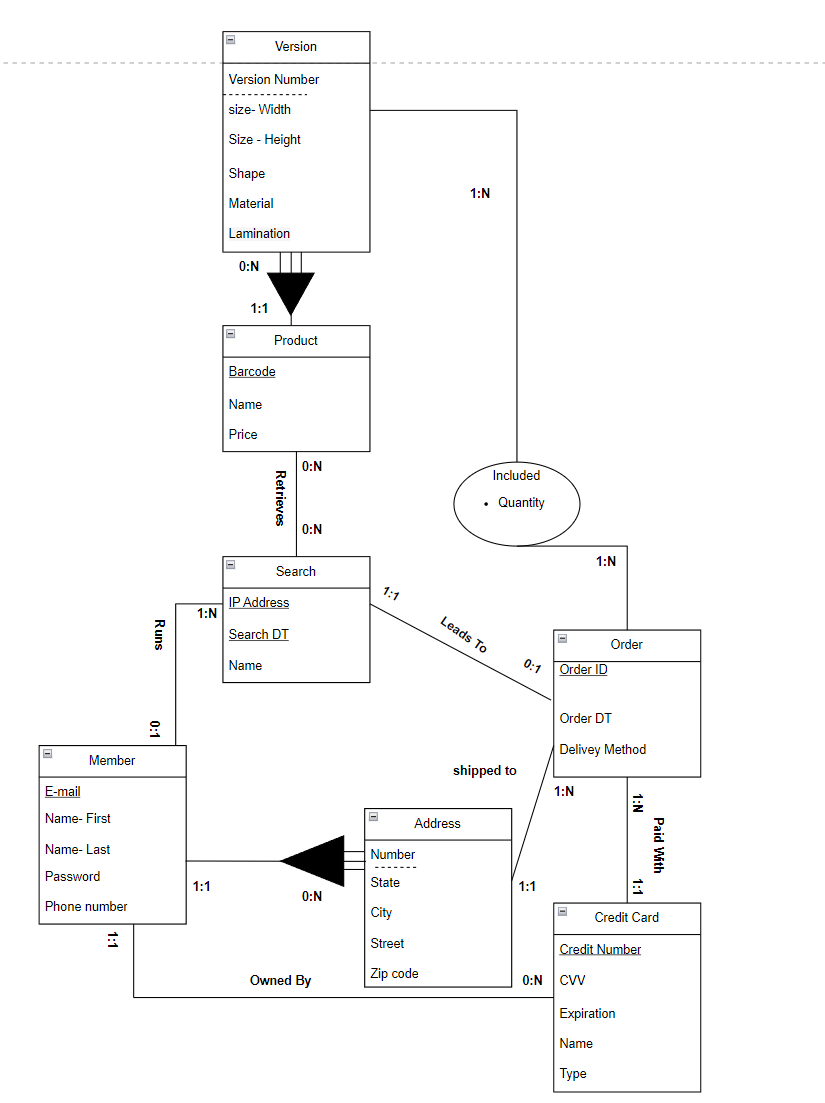
תוסיפו תוכן חיפוש- שדות המתייחסים לחפש

בלי קשר בין הזמנה לכתובת, לא נדע איזה מכתובות הלקוח נבחרה להזמנה זו

לא חובה שלכל מוצר תהיה גרסה, 0:N

## מטלה 2 (30%) – עיצוב לוגי (מודל נתונים טבלאי)

* **PRODUCTS** (Barcode, Name, Price)
* **VERSIONS** (Barcode(PRODUCTS), Version Number, Size – Width, Size – Height, , Shape, Material, Lamination)
* **MEMBERS** (Mail, Name – First, Name – Last, Password, Phone Number)
* **SEARCHES** (IP Address, Search DT, Mail (MEMBERS))
* **CREDIT CARDS** (Credit Number, CVV, Expiration Date, Name,CC Type, Mail (MEMBERS))
* **ORDERS** (Order ID, Order DT, Delivery method, {IP Address, Search DT} (SEARCHES), Credit Number (CREDIT CARDS))
* **ADDRESSES** (Mail (MEMBERS), Number, State, City, Street ,ZIP Code)
* **INCULDES** ({Barcode, Version Number} (VERSIONS), Order ID (ORDERS), Quantity)
* **RETRIEVES** (Barcode (PRODUCTS), {IP Address, Search DT} (SEARCHES))



* **PRODUCTS** (Barcode, Name, Price)
* **VERSIONS** (Barcode(PRODUCTS), Version Number, Size – Width, Size – Height, , Shape, Material, Lamination)
* **MEMBERS** (Mail, Name – First, Name – Last, Password, Phone Number)
* **SEARCHES** (IP Address, Search DT, Name , Mail (MEMBERS))
* **CREDIT CARDS** (Credit Number, CVV, Expiration Date, Name,CC Type, Mail (MEMBERS))
* **ORDERS** (Order ID, Order DT, Delivery method, {IP Address, Search DT} (SEARCHES), Credit Number (CREDIT CARDS), {E-mail, Number} (ADDRESSES))
* **ADDRESSES** (Mail (MEMBERS), Number, State, City, Street,ZIP Code)
* **INCULDES** ({Barcode, Version Number} (VERSIONS), Order ID (ORDERS), Quantity)
* **RETRIEVES** (Barcode (PRODUCTS), {IP Address, Search DT} (SEARCHES))

## מטלה 1 (25%) – שאילתות

# שתי שאילתות SELECT ללא קינון (5%)

**שאילתה ראשונה-**

לכל לקוח ולכל מדינה- מי הם 10 הלקוחות שהרווח מהם הכי גדול שהזמינו משלוח? ומהי מדינתם?

הגיון עסקי: החברה תרצה לדעת זאת לצורך שימור לקוחות. כך היא תדע לשלוח להם מידי פעם התראות על מבצעים, מוצרים חדשים שנוספו ואפילו שוברי קניה לאתר (לכתובת המייל שלהם ולביתם), הואיל והם לקוחות המעוניינים במוצרי החברה.

SELECT TOP 10 [Full Name] = M.[Name\_First] +' '+ M.[Name\_Last],M.[Mail],Country= A.[State],A.[City],A.[Street], Revenue = SUM(P.Price \* I.Quantity)

FROM ORDERS AS O JOIN INCLUDES AS I ON O.[Order\_ID] = I.[Order]

JOIN MEMBERS AS M ON M.[Mail]= O.[Member]

JOIN ADDRESSES AS A ON M.[Mail]= A.[Member]

JOIN PRODUCTS AS P ON I.[Product] = P.Barcode

WHERE O.[Delivery\_Mehtod] = 'shippment'

GROUP BY M.[Name\_First] ,M.[Name\_Last],M.[Mail], A.[State],A.[City],A.[Street]

ORDER BY Revenue DESC



1. מהם המוצרים שהוזמנו במספר גרסאות שונות, הגבוה ביותר בחצי שנה האחרונה?

הגיון עסקי:

SELECT p.Barcode, p.Name, amount = COUNT(distinct version)

FROM ORDERS as o

Join INCLUDES as i ON o.Order\_ID = i.[Order]

join PRODUCTS as p ON i.Product = p.Barcode

Join [ADDRESSES ] as a ON o.address = a.[Number] AND o.[Member] = a.[Member]

WHERE DATEDIFF(mm, o.Order\_DT, getdate()) <= 6

GROUP BY p.Barcode, p.Name

ORDER BY amount DESC



# שתי שאילתות SELECT מקוננות (10%)

1. מי הם המוצרים, שהרווח עליהם בשנת 2022 גדול מהרווח הכולל עליהם בשנת 2021, ומהו הרווח לכל שנה?

הגיון עסקי: החברה תרצה לבחון, עבור כל מוצר, מי הם המוצרים שהרווח עבורם היה במגמת עליה משנת 2021 לשנת 2022. בעזרת נתון זה החברה תוכל לדעת מה היו הכלים שגרמו לעליה זו, ושמר את אותם דברים.

SELECT P.[Barcode],P.[Name],AVG\_IN\_22= AVG (I.[Quantity] \* P.[Price]),

AVG\_IN\_21 =(SELECT AVG(I.[Quantity] \* P.[Price])

FROM PRODUCTS as P JOIN INCLUDES as I ON P.[Barcode] = I.[Product]

JOIN ORDERS as O ON I.[Order] = O.[Order\_ID]

JOIN [VERSIONS] as V ON V.[Product] = P.[Barcode]

WHERE YEAR( O.[Order\_DT]) = 2021 AND P.[Barcode] = V.[Product])

FROM PRODUCTS as P JOIN INCLUDES as I ON P.[Barcode] = I.[Product]

JOIN ORDERS as O ON I.[Order] = O.[Order\_ID]

JOIN [VERSIONS] as V ON V.[Product] = P.[Barcode]

WHERE YEAR( O.[Order\_DT]) = 2022

GROUP BY P.[Barcode],P.[Name]

HAVING AVG(I.[Quantity] \* P.[Price]) > (

SELECT AVG(I.[Quantity] \* P.[Price])

FROM PRODUCTS as P JOIN INCLUDES as I ON P.[Barcode] = I.[Product]

JOIN ORDERS as O ON I.[Order] = O.[Order\_ID]

JOIN [VERSIONS] as V ON V.[Product] = P.[Barcode]

WHERE YEAR( O.[Order\_DT]) = 2021 AND P.[Barcode] = V.[Product]

)

Graphical user interface, text, application

Description automatically generated

1. לאיזה מדינות הוזמנו מוצרים על ידי משלוח פי 2 מעל ממוצע המשלוחים בכל שאר המדינות?

הגיון עסקי: החברה תרצה לבחון אפשרויות משלוח מול ספקים שונים אשר מספקים משלוחים למדינות שונות, למדינות בהם יש ביקוש גבוה של מוצרים מאשר מדינות אחרות.

SELECT a.state, amount = COUNT(\*)

FROM ORDERS as o

Join INCLUDES as I ON i.[Order] = o.order\_ID

Join ADDRESSES as a ON o.address = a.number AND o.member = a.member

WHERE o.Delivery\_Mehtod = 'shippment'

GROUP BY a.state

HAVING (

SELECT CAST(SUM(temp.amount) as real)/COUNT(\*)

FROM (

SELECT amount = Count(\*)

FROM ORDERS as o

JOIN ADDRESSES as a ON o.address=a.number AND o.member=a.member

WHERE o.Delivery\_Mehtod = 'shippment'

GROUP BY a.state

)

as temp

) < 0.5 \* COUNT(\*)

ORDER BY amount DESC



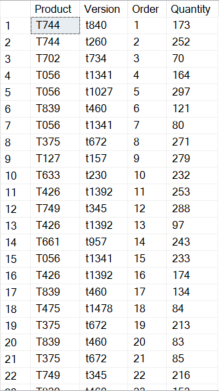
# שתי שאילתות מקוננות תוך שימוש במרכיבים נוספים (10%)

**חיסור בין שאילתות:**

**לפני העידכון:**

SELECT\*

FROM INCLUDES

****

**אחרי העידכון:**

ALTER TABLE INCLUDES DROP COLUMN DISCOUNT

ALTER TABLE INCLUDES ADD discount real

UPDATE INCLUDES

SET

discount = (select t1.t

from (SELECT temp.Order\_ID,t=temp.[total price]\*0.025

FROM (

SELECT o.Order\_ID, [total price] = SUM(p.Price\*i.Quantity)

FROM ORDERS as o

JOIN INCLUDES as i on o.Order\_ID = i.[Order]

JOIN PRODUCTS as p ON p.Barcode = i.Product

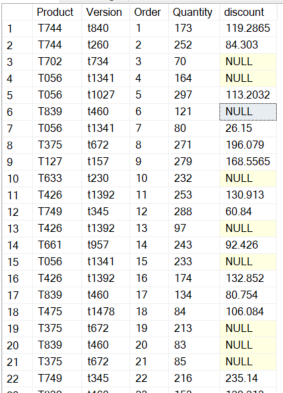
GROUP BY o.Order\_ID

having sum(i.Quantity)>=250

) as temp

) as t1

where INCLUDES.[Order] = t1.Order\_ID)

****

1. עבור אילו מוצרים שנקנו ב2022 ההפרש בין הרווח הכולל (מספר היחידות\* כמות) שנמכרו ברבעון הראשון הרווח הכולל (מספר היחידות\* כמות) שנמכרו ברבעון הרביעי קטן מ-0?

הגיון עסקי: לרוב בחברות, ברבעון הרביעי, עושים הזמנת מוצרים גדולה לחדש את המלאי. על מנת לדעת מי הם המוצרים שהביקוש שלהם גדול מההיצע שלהם (כך החברה תדע להערך לכך בשנת 2023). בנוסף לכך, חשוב לדעת כמה הרווחנו עליהם.

SELECT first\_Summary.[Barcode], first\_Summary.[Name],

[First\_Quarter] = first\_Summary.[First\_Quarter],

[Fourth\_Quarter] = ISNULL (Fourth\_Summary.[Fourth\_Quarter], 0),

Gap = [First\_Quarter] - ISNULL ([Fourth\_Quarter], 0)

FROM (

SELECT P.[Barcode], P.[Name],

[First\_Quarter] = SUM (I.Quantity\*P.[Price])

FROM [PRODUCTS] as P JOIN INCLUDES as I ON P.[Barcode] = I.[Product]

JOIN ORDERS as O ON I.[Order] = O.[Order\_ID]

WHERE MONTH (O.[Order\_DT]) IN (1,2,3) and YEAR (O.[Order\_DT]) =2021

GROUP BY P.[Barcode], P.[Name]

) AS first\_Summary LEFT JOIN

(

SELECT P.[Barcode], [Fourth\_Quarter] = SUM (I.Quantity\*P.[Price])

FROM PRODUCTS as P JOIN INCLUDES as I ON P.[Barcode] = I.[Product]

JOIN ORDERS as O ON I.[Order] = O.[Order\_ID]

WHERE MONTH (O.[Order\_DT]) IN (10,11,12) and YEAR (O.[Order\_DT]) =2022

GROUP BY P.[Barcode]

) AS Fourth\_Summary ON first\_Summary.[Barcode] = Fourth\_Summary.[Barcode]

WHERE [First\_Quarter] - ISNULL ([Fourth\_Quarter], 0)<0

ORDER BY Gap DESC

Graphical user interface, application, table

Description automatically generated

# View (5%)

בטבלת הview הזאת ניתן לראות מי הם האנשים אשר קנו מוצר מסויים, מה מספר כרטיס האשראי שלהם, ומתי הוא נרכש, ואיזה מוצר נרכש.

כך ההנהלה תוכל לפקח עבור מכירותיה ותוכל לתת מענה ללקוחות שיש להם בעיה כגון: המוצר אינו הגיע ליד או שהם מעוניינים בהחלפה וכו...

בעזרת נתונים אלו החברה תוכל לראות פרטים עבור הרכישה של המוצר.

drop VIEW VIEW\_ORDERED\_PRODUCTS;

CREATE VIEW VIEW\_ORDERED\_PRODUCTS AS

SELECT [Full Name] = M.[Name\_First] +' '+ M.[Name\_Last] ,M.Mail,C.Credit\_Number,

O.[Order\_DT], I.[Product]

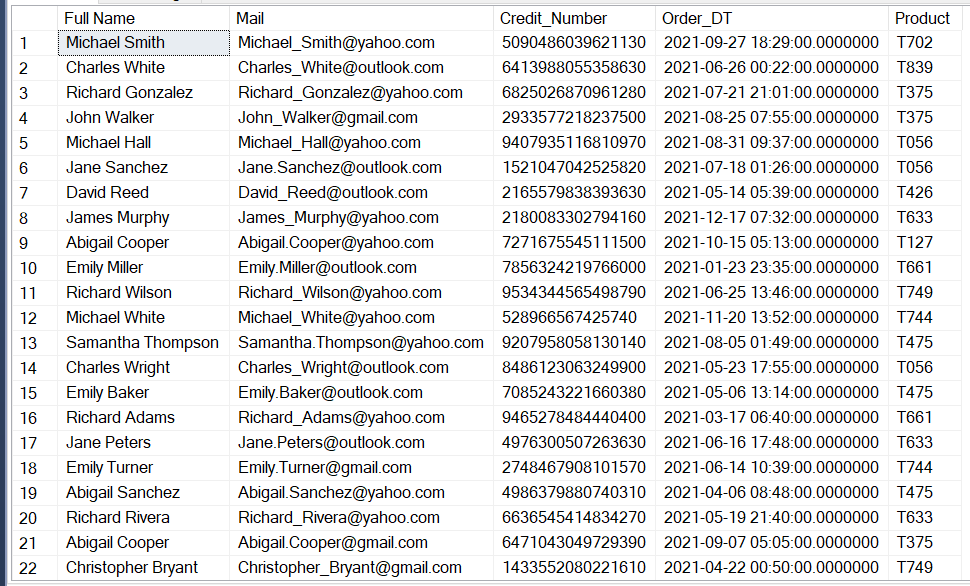
FROM INCLUDES as I JOIN ORDERS AS O ON I.[Order]= O.[Order\_ID]

JOIN [CREDIT CARDS] AS C ON C.[Credit\_Number] = O.[Credit\_Card]

JOIN MEMBERS AS M ON M.[Mail] = C.[Member]

SELECT \*

FROM VIEW\_ORDERED\_PRODUCTS

WHERE YEAR([Order\_DT])=2021

פונקציות (Functions) (10%)

**פונקציה המחזירה ערך יחיד (Scalar):**

* + - * בעזרת הפונקציה ניתן לאמוד את המחיר של מוצר מסויים. האם המחיר נמצא בטווח גבוה, נמוך או רגיל. כך ניתן לסווג את המוצרים לקבוצות אלו ולבחון לדוגמא- כמה מוצרים יקרים נמכרו בשנת 2021?

CREATE FUNCTION PriceRange (@Price Money)

RETURNS Varchar(50)

AS BEGIN

DECLARE @PriceRange Varchar(50)

SELECT @PriceRange = CASE

WHEN @Price BETWEEN 10 AND 15 THEN 'Hight'

WHEN @Price BETWEEN 5 AND 9 THEN 'Regular'

WHEN @Price BETWEEN 1 AND 4 THEN 'Low'

END

RETURN @PriceRange

END

SELECT Barcode, PRICERANGE = dbo.PriceRange (PRICE)

FROM PRODUCTS

Table

Description automatically generated

**פונקציה המחזירה טבלה:**

* + - * בעזרת הפונקציה ניתן לראות עבור כל מוצר מה סך הכמות שהוא נמכר, מה סך הקניות של כל המוצרים, ואחוז- מה רמת הפופולריות שלו? (מחושב ע"י- כמה הוא נקנה לחלק לסך הקניות של כל המוצרים)

DROP FUNCTION dbo.PopularityOfProduct;

CREATE FUNCTION PopularityOfProduct(@Barcode NVARCHAR(50))

RETURNS TABLE

AS RETURN

SELECT

BARCODE = @Barcode,

[sales\_count] =( select ISNULL(COUNT([Product]),0)

from INCLUDES

where @Barcode = [Product])

,[total\_sales]= ISNULL(COUNT([Product]),0),

[popularity\_measure]=CAST(((select ISNULL(COUNT([Product]),0)

from INCLUDES

where @Barcode = [Product])) AS REAL)/(ISNULL(COUNT([Product]),0))\*100

FROM INCLUDES

**לצורך בדיקת הפונקציה:**

SELECT \*

FROM dbo.PopularityOfProduct ('T702');



# פרוצדורה שמורה ((Stored Procedure פשוטה (5%)

הפרוצדורה השמורה הזאת מאפשרת לנו לשנות את מחירו של מוצר מסויים לפי הנחה מסויימת שנבחר.

DROP procedure dbo.ProductDiscount;

create procedure ProductDiscount

@ProductID varchar(50), @Discount real

as

update PRODUCTS

set Price = Price \* (1-@Discount)

where PRODUCTS.[Barcode] = @ProductID

select \*

from PRODUCTS

where PRODUCTS.[Barcode] = @ProductID

**לצורך בדיקת הפרוצדורה:**

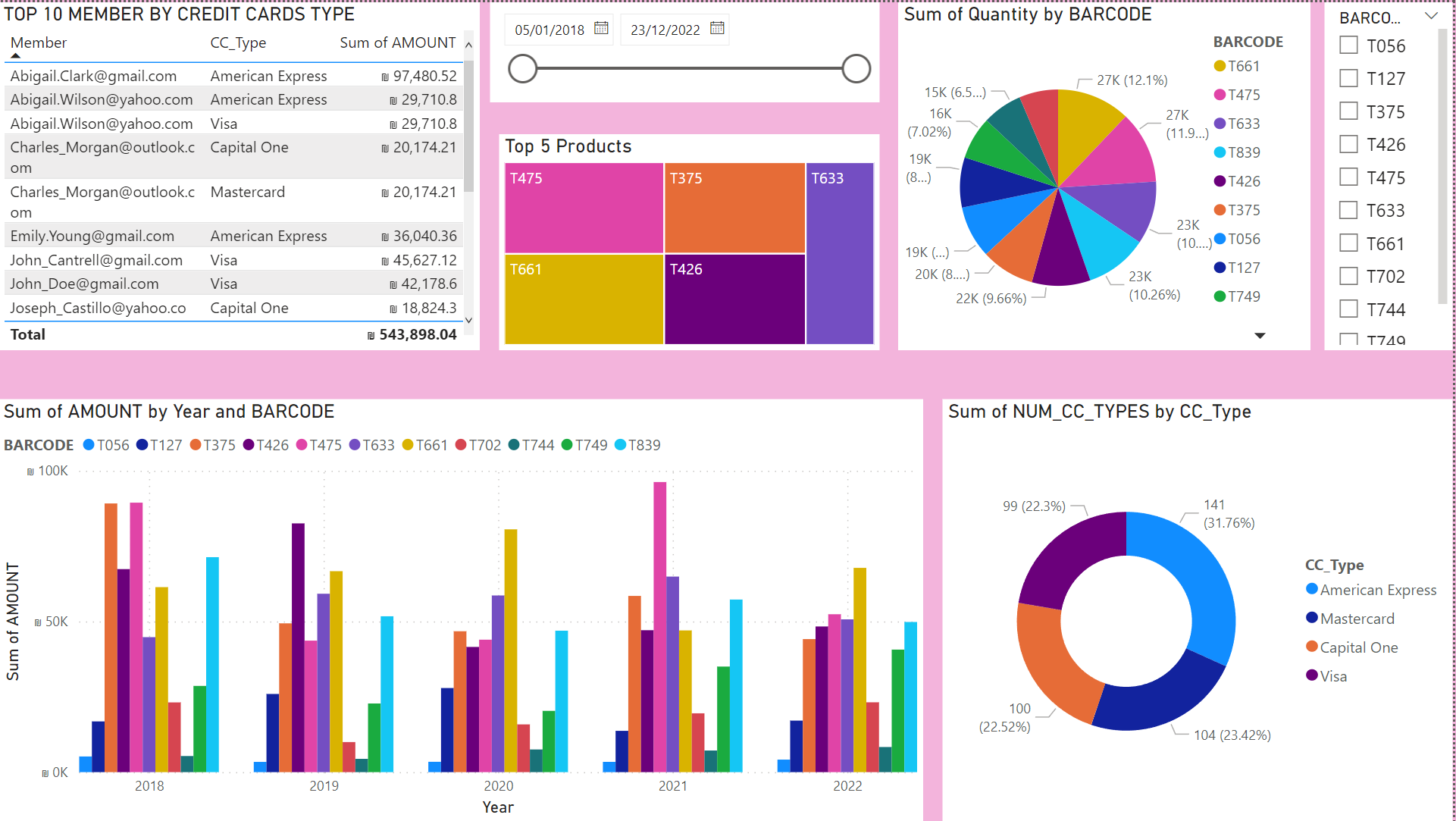
execute ProductDiscount 'T702', 0.5

## 

## מטלה 3 (20%) – כלים להצגת נתונים

**דוח עסקי**

הדוח שלנו מופק עבור מחלקת שיווק ותפעול. בעזרת הנתונים מחלקות אלו יוכלו לבחון שיתופי פעולה עם חברות אשראי שונות והנחות שונות או מבצעים, לבחון אילו מוצרים הם אטרקטיביים יותר באילו מקומות ומה ניתן לעשות על מנת לשפר ולשמר.

בדוח ישנם גרפים שונים ופילטרים שונים שבאמצעותם ניתן להפיק מידע אודות החברה, מוצריה ולקוחותיה:

**קוד עזר ליצירת הדוח:**

DROP VIEW VIEW\_ORDERS\_CC;

CREATE VIEW VIEW\_ORDERS\_CC AS

SELECT C.CC\_Type,NUM\_CC\_TYPES= COUNT(Credit\_Number)

FROM

ORDERS AS O JOIN [CREDIT CARDS] AS C ON O.Credit\_Card = C.Credit\_Number

GROUP BY C.CC\_Type

DROP VIEW VIEW\_MEMBERS\_REVENUE;

CREATE VIEW VIEW\_MEMBERS\_REVENUE AS

SELECT O.Member ,O.[Order\_DT],A.[State],C.CC\_Type,P.BARCODE, I.[Quantity],P.Price,AMOUNT= SUM( I.[Quantity]\*P.Price)

FROM INCLUDES as I JOIN PRODUCTS AS P ON I.[Product]= P.BARCODE

JOIN ORDERS AS O ON O.[Order\_ID]=I.[Order]

JOIN MEMBERS AS M ON M.[Mail]=O.Member

JOIN [CREDIT CARDS] AS C ON M.[Mail]=C.Member

JOIN [ADDRESSES ] AS A ON A.[Member]=M.[Mail]

GROUP BY O.Member,O.[Order\_DT],A.[State],C.CC\_Type,P.BARCODE, I.[Quantity],P.Price

**לוח מחוונים**

**Graphical user interface

Description automatically generated**לוח המחוונים שלנו מופק עבור מנכ"ל החברה והדריקטריון. הנהלת החברה הם הגורמים שצריכים לראות נתונים בעלי גרעיניות מינימלית, ועל כן אופן פירוט הנתונים שלנו הינו פשטני ביותר. הלוח בעיקרו מתמקד בהכנסות החברה:

**קוד עזר ליצירת הדוח:**

CREATE VIEW VIEW\_Dashboard1 AS

SELECT O.[Order\_DT],O.Order\_ID, O.Member,P.[Name],V.Version\_Number, P.Barcode,I.Quantity ,P.Price, A.[State], Revenue = I.Quantity \* P.Price,discount=ISNULL(I.[discount],0)

FROM INCLUDES AS I JOIN ORDERS AS O ON I.[Order]=O.Order\_ID

JOIN PRODUCTS AS P ON I.Product=P.Barcode

JOIN MEMBERS AS M ON O.Member = M.[Mail]

JOIN VERSIONS AS V ON V.Product = P.Barcode

JOIN [ADDRESSES ] AS A ON O.[Address]= A.[Number]

GROUP BY O.[Order\_DT],O.Order\_ID, O.Member, P.Barcode, A.[State],V.Version\_Number , I.Quantity , P.Price,P.[Name],I.[discount]

## מטלה 4– יישומים מתקדמים

## שאילתות עסקיות המשלבות Window Functions

**פונקציית עזר:**

פונקציית עזר שלכל לקוח שאנחנו מכניסים הפלט יהיה- מה המוצרים שהלקוח קנה.

CREATE FUNCTION get\_customer\_products(@customer\_email varchar(255))

RETURNS VARCHAR(MAX)

AS

BEGIN

DECLARE @products VARCHAR(MAX)

SELECT @products = STRING\_AGG(P.Barcode, ', ')

FROM INCLUDES AS I

INNER JOIN ORDERS AS O ON I.[Order]=O.Order\_ID

INNER JOIN PRODUCTS AS P ON I.Product=P.Barcode

INNER JOIN MEMBERS AS M ON O.Member = M.Mail

WHERE M.Mail = @customer\_email

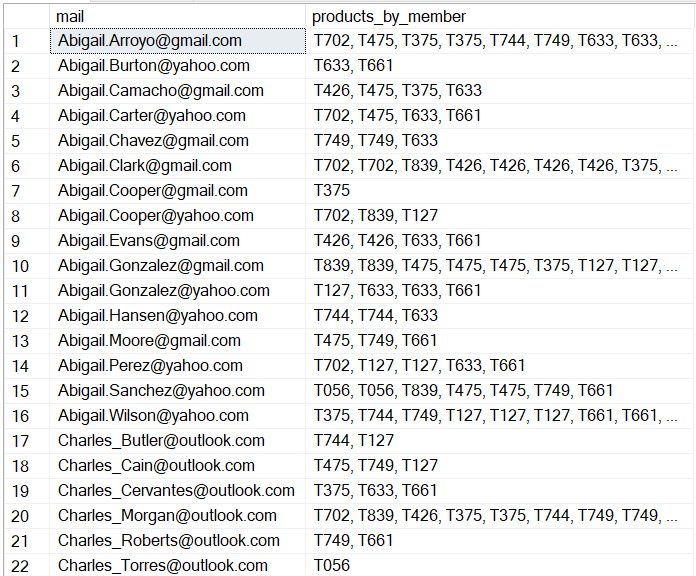
RETURN @products

END

SELECT mail , products\_by\_member= dbo.get\_customer\_products(mail)

from MEMBERS

group by mail



* + - * השאילתה מציגה לכל לקוח, אילו מוצרים הוא קנה, מה סך הכסף שהוא שילם, מה הדירוג של אותו לקום לפי הקניה הכי גדולה שלו (כמות\*מחיר) ומתי (באיזה שנה) התבצעה הקניה הכי גדולה שלו?

SELECT DISTINCT

O.Member,

products\_by\_member= dbo.get\_customer\_products( O.Member),

Revenue =I.Quantity \* P.Price ,

[Rank] = RANK () OVER (ORDER BY I.Quantity \* P.Price desc),

Highest = MAX (YEAR(o.Order\_DT)) OVER ( PARTITION BY O.Member ORDER BY O.Member)

FROM INCLUDES AS I

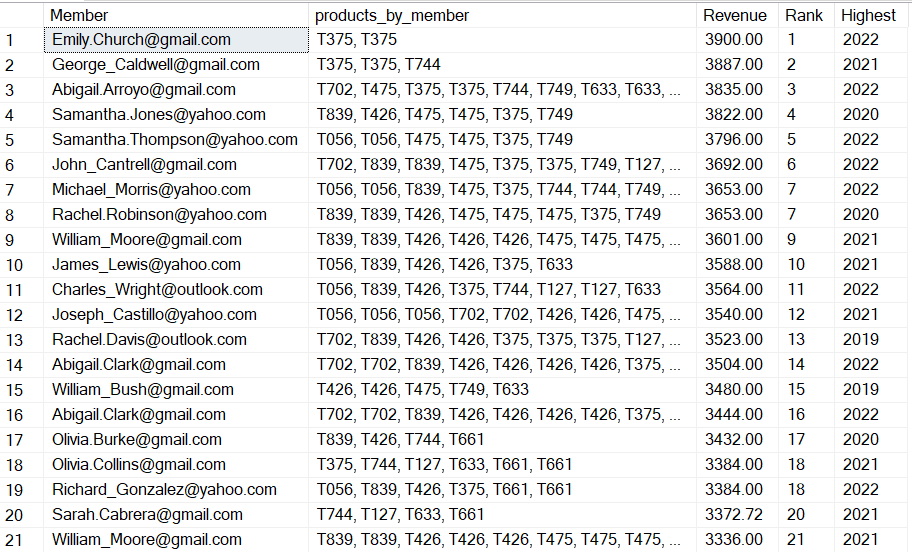
JOIN ORDERS AS O ON I.[Order]=O.Order\_ID

JOIN PRODUCTS AS P ON I.Product=P.Barcode

JOIN MEMBERS AS M ON O.Member = M.[Mail]

GROUP BY P.Barcode,o.Order\_DT,I.Quantity \* P.Price ,O.Order\_ID,I.Quantity ,O.Member

ORDER BY [Rank]



* + - * השאילתה מציגה- לכל מדינה, מה סך הכסף של מוצרים שנקנו/ נשלחו אליה, מה הגידול משנת 2021 לשנת 2022 בסך הכסף שנכנס לאותה מדינה, ועוד עמודה שמציינת האם יש גידול/ הפסד או מצב נטרלי משנת 2021.

השאילתה בוחנת את המגמתיות בין השנים 2021 ל2022 לפי מדינה.

SELECT

A.[State],

SUM(I.Quantity \* P.Price) as total\_revenue,

revenue\_growth = isnull(SUM(quantity \* price) - LAG(SUM(quantity \* price)) OVER (ORDER BY Year(O.Order\_DT)),0),

CASE

WHEN SUM(quantity \* price) - LAG(SUM(quantity \* price)) OVER (ORDER BY Year(O.Order\_DT)) > 0 THEN 'Profit'

WHEN SUM(quantity \* price) - LAG(SUM(quantity \* price)) OVER (ORDER BY Year(O.Order\_DT)) < 0 THEN 'Loss'

ELSE 'Break-even'

END as profit\_or\_loss

FROM INCLUDES I

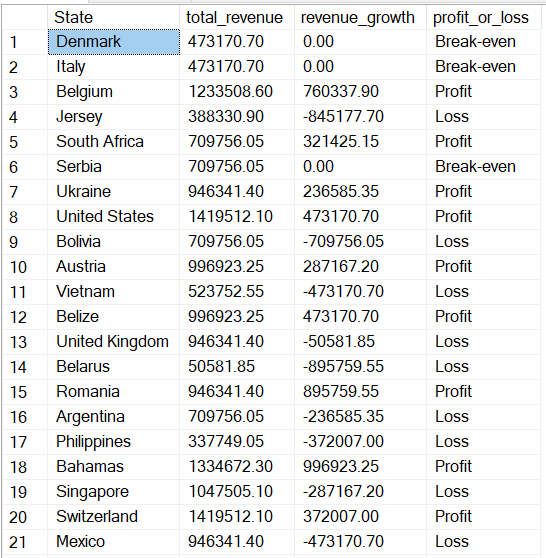
JOIN ORDERS O ON I.[Order]=O.Order\_ID

JOIN PRODUCTS P ON I.Product=P.Barcode

JOIN [ADDRESSES ] A ON O.Address= A.Number

WHERE year(O.Order\_DT)>= year(dateadd(yy, -2, getdate()))

GROUP BY A.[State], Year(O.Order\_DT)



## שילוב מערכתי של מספר כלים (10%)

SELECT \* FROM MEMBERS

ALTER TABLE MEMBERS ADD Priority VARCHAR(10) ;

DROP FUNCTION dbo.SetType;

CREATE FUNCTION SetType (@Mail VARCHAR(50), @FROM DATE, @TO DATE)

RETURNS VARCHAR(10)

AS BEGIN

DECLARE @MemberType VARCHAR(10)

DECLARE @OrdersSum INT

DECLARE @SearchesSum INT

SELECT @OrdersSum = COUNT(O.Member)

FROM ORDERS AS O

JOIN MEMBERS AS M ON M.Mail = O.Member

WHERE O.Order\_DT BETWEEN @FROM AND @TO

AND M.Mail = @Mail

SELECT @SearchesSum = COUNT(S.Member)

FROM [SEARCHES ] AS S

JOIN MEMBERS AS M ON M.Mail = S.Member

WHERE S.Search\_DT BETWEEN @FROM AND @TO

AND M.Mail = @Mail

SET @MemberType =

CASE

WHEN (@OrdersSum - @SearchesSum) > 0 THEN 'MAX'

ELSE 'MIN'

END

RETURN @MemberType

END

DROP VIEW dbo.VIEW\_VIP\_MEMBERS;

CREATE VIEW VIEW\_VIP\_MEMBERS AS

SELECT [FULL NAME]= M.[Name\_First] +' '+ M.[Name\_Last] ,M.Mail

FROM MEMBERS AS M

WHERE M.Priority='MAX'

DROP PROCEDURE dbo.MemberType;

CREATE PROCEDURE MemberType

AS

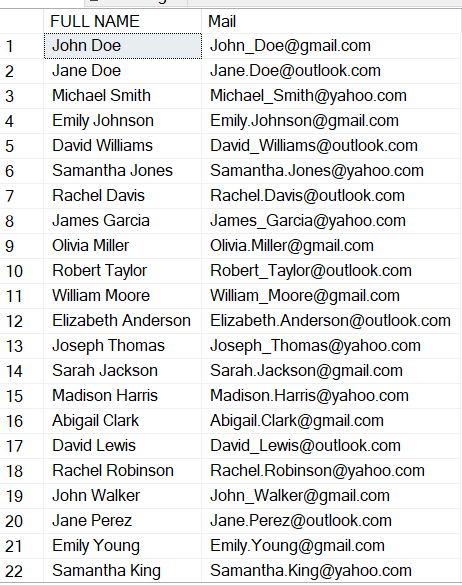
UPDATE MEMBERS

SET Priority=dbo.SetType (Mail, '01-01-2015','01-01-2020')

SELECT \*

FROM VIEW\_VIP\_MEMBERS

EXECUTE MemberType



## דו"ח המושתת על שימוש בפסקת WITH מורכבת (10%)

WITH

productsList AS (

SELECT p.Barcode as ProductCategory, SUM(i.Quantity \* p.Price) AS TotalSales, AVG(p.Price) AS AveragePrice

FROM PRODUCTS as p JOIN INCLUDES as i ON p.Barcode = i.Product

GROUP BY p.Barcode

),

ordersList AS (

SELECT p.Barcode as ProductCategory, COUNT(DISTINCT i.[Order]) AS TotalOrders

FROM PRODUCTS as p JOIN INCLUDES as i ON p.Barcode = i.Product

GROUP BY p.Barcode

),

deliveries AS (

SELECT i.Product, o.Delivery\_Mehtod, total = COUNT(DISTINCT i.[Order])

FROM ORDERS as o JOIN INCLUDES as i ON o.Order\_ID = i.[Order]

GROUP BY o.Delivery\_Mehtod, i.Product

),

searchesList AS (

SELECT r.Product, COUNT(\*) AS TotalSearches

FROM RETRIEVES as r

GROUP BY r.Product

),

searchOrderRatio AS (

SELECT ol.ProductCategory, CAST(TotalSearches as FLOAT)/TotalOrders AS SearchOrderRatio

FROM ordersList as ol

JOIN searchesList as sl ON ol.ProductCategory = sl.Product

)

SELECT TOP 10 pl.ProductCategory, ol.TotalOrders, pl.TotalSales, pl.AveragePrice, d.Delivery\_Mehtod, sor.SearchOrderRatio

FROM productsList as pl

JOIN ordersList as ol ON pl.ProductCategory = ol.ProductCategory

JOIN deliveries as d ON d.Product = pl.ProductCategory

JOIN searchOrderRatio as sor ON sor.ProductCategory = pl.ProductCategory

ORDER BY ol.TotalOrders DESC

