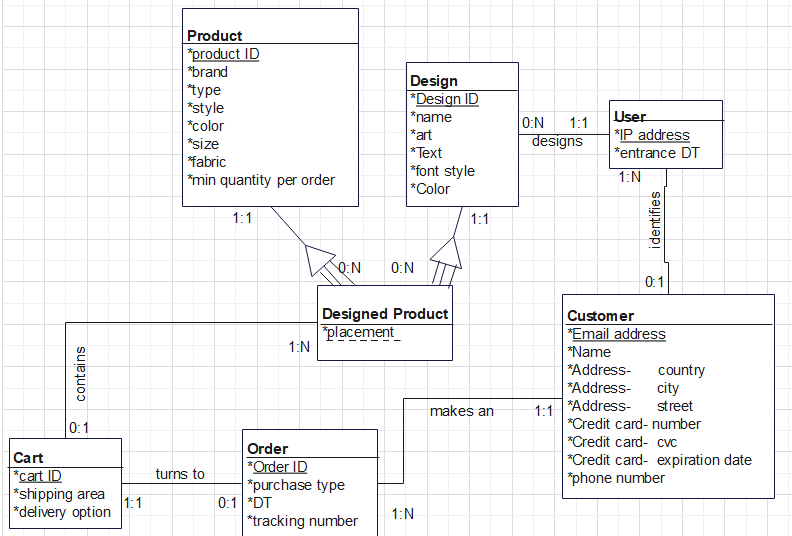
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מס' קבוצה | אתר | | | תאריך הגשה |
| 12 | https://www.customink.com/ | | | 25/06/2022 |
| חברי הצוות - מספרי ת.ז | | | | |
| 207094483 | | 316061043 | 206399123 | |

## פרויקט בסיסי נתונים – חלק ג'

## מטלת חובה מקדימה – תיקון ה-ERD והעשרת בסיס נתונים

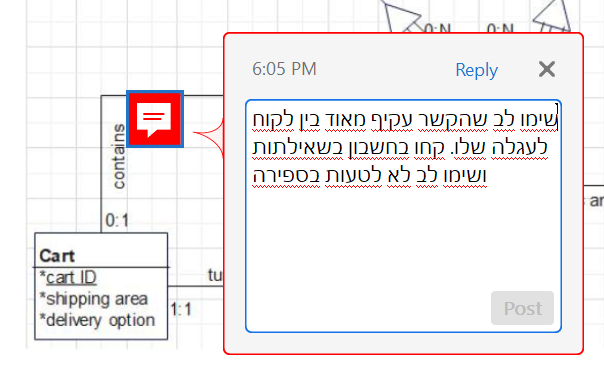
מודל הERD שהוגש בפעם הקודמת:



ההערות שניתנו עבור המודל הקודם:

תמונה שמכילה שולחן

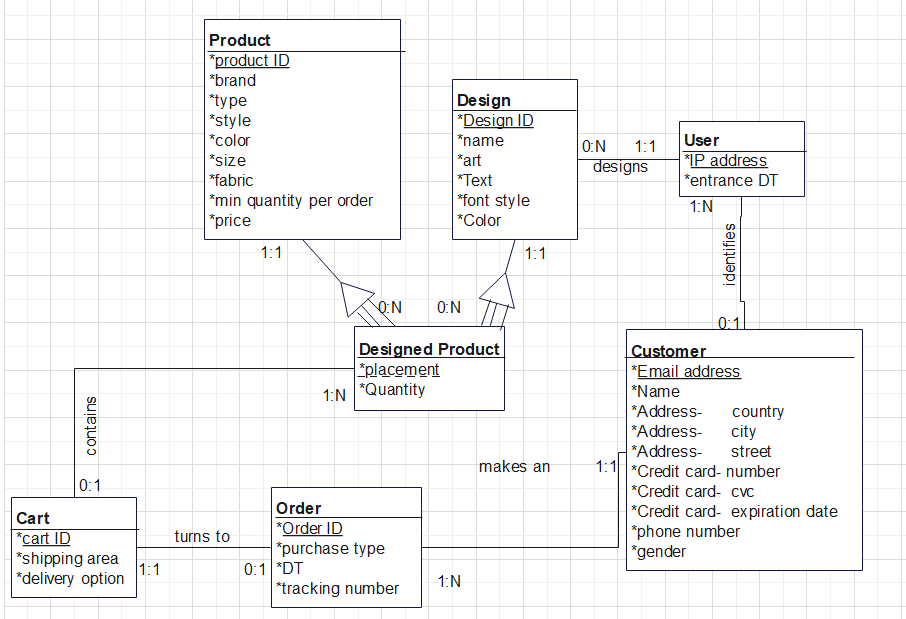
התיאור נוצר באופן אוטומטי



מצורפות ההנחות שביצענו בחלק הקודם:

|  |  |
| --- | --- |
| **מרכיב** | **הנחות** |
| התהליך העסקי | * התייחסנו רק לעיצוב על פריטי הלבוש ולא על פריטים אחרים. * ניתן לשמור את העיצוב ולשלוח אותו למייל- מכיוון ומדובר בהפסקת תהליך ההזמנה ושמירת המוצר, לא הגדרנו כחלק מהתהליך העסקי. |
| Design (עיצוב) | * משתמש מחויב לבחור עיצוב עבור פריט הלבוש אותו ירכוש. * יצרנו ישות עצמאית משום שישנה אפשרות להשתמש באותו העיצוב עבור מוצרים שונים, במיקום שונה. בנוסף, לעיצוב ישנם הרבה שדות רלוונטיים (תמונה, טקסט וכדומה) אשר קשורים באופן מהותי לתהליך העסקי. |
| Customer (לקוח) | * לקוח יכול להתחבר ממספר כתובות IP שונות. * ללקוח יש כרטיס אשראי אחד בלבד המזוהה עם כלל רכישותיו. |
| User (משתמש) | * משתמש (User) יוכל לעצב פריטי לבוש ולהכניסם לעגלה, ויזוהה עפ"י כתובת הIP ממנה התחבר. |
| הקשר בין "Customer' ל"User" | * לקוח (Customer) מזוהה כמשתמש שביצע הזמנה והכניס למערכת פרטי זיהוי נרחבים. * משום שלקוח יכול להתחבר ממספר כתובות IP, יכול להיות מזוהה כמספר משתמשים. |
| Designed Product | * מיקום העיצוב, על גבי פריט הלבוש הינו ייחודי ועל כן מהווה מזהה חלקי. |
| הקשר בין "designed product" ל"Cart" | * ניתן לעצב פריט ולשלוח אותו במייל, ולא להעבירו ישירות לעגלה, על כן הקרדינליות 0:1. |
| Cart | * "Cart ID" המזהה, משום שהעגלה מורכבת מפריטים אותם מכניס משתמש (USER), לעומת הזמנה אשר מבצע לקוח (Customers). למשתמש יכולות להיות מספר עגלות, ומכיוון שלא כל עגלה הופכת להזמנה, נדרש ליצור מזהה עבור עגלה ולשמור את הפרטים שלה במסד נתונים. |

**הERD העדכני שלנו הינו:**

 **השינויים שביצענו בERD**:

* הוספת שדה Quantity לישות Designed Product שתייצג את הכמות בה המוצר נרכש. השדה הינו בעצם תכונה על הקשר בין עגלה למוצר מעוצב ונמצא במיקום זה בגלל הקשר של יחיד לרבים.
* הוספת שדה price לישות Product המייצג מהו המחיר של המוצר עצמו.
* הוספת שדה gender בישות Customer המייצג את מין הלקוח.

כלל השדות הללו נוספו עבור ניתוחים מעניינים יותר בpower bi.

**התייחסותנו להערות שניתנו בחלק ב':**

* עבור הקשר בין לקוח לעגלה שלו, נדאג לאכוף את החיבורים הנדרשים בFROM והWHERE כשנבצע JOIN ובמסד הנתונים.
* בנוגע להערה על מזהה DESIGNED PRODUCT- לא שינינו מכיוון שכל משתמש יוצר עיצוב בהכרח עבור הפריט אותו יבחר (או מספר פריטים שונים), וימקם אותם לפי רצונו- ז"א שיכול להיות מספר רב מאוד של קומבינציות עיצוב, התלויות בפריט שונה, עיצוב שונה ומיקום, ועל כן נדרש מזהה התלוי בכולם.

הטבלאות העדכניות שלנו:

* **PRODUCTS** (Product ID, Brand, Type, Style, Color, Size, Fabric, Min quantity, Price)
* **DESIGNS** (Design ID, Name, Art, Text, Font style, Color, IP Address (USERS))
* **CUSTOMERS** (Email address, Name, country, city, street, CC-number, CC-cvc, CC- expiration date, phone number, Gender)
* **USERS** (IP Address, entrance DT, Email Address (CUSTOMERS))
* **ORDERS** (Order ID, Purchase type, DT, tracking number, Cart ID(CARTS), Email Address (CUSTOMERS))
* **CARTS** (Cart ID, Shipping area, Delivery option)
* **DESIGNED PRODUCTS** (Product ID (PRODUCTS), Design ID (DESIGNS), Placement, Quantity, Cart ID (CARTS))

## מטלה 1 שאילתות

# שאילתות SELECT ללא קינון

* שאילתא המציגה מה הם סוג המוצרים הנמכר ביותר בחברה. שאילתא זו תוכל להכווין אותנו לניהול מלאי טוב יותר- החזקת מגוון רחב יותר של מוצרים מסוג זה ואפשרות לתמחור תחרותי יותר.

select P.Type, Popularity = count (P.Type)

from PRODUCTS as P JOIN DESIGNED\_PRODUCTS as DP

on P.ProductID= DP.ProductID JOIN ORDERS as O

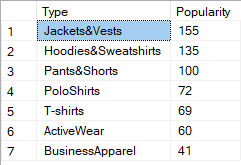
on DP.CartID = O.CartID

where P.Type is not null

group by P.Type

order by Popularity desc

הפלט שקיבלנו עבור השאילתא:



ניתן לראות כי המוצר שהופיע בהכי הרבה הזמנות הינו ג'קטים ווסטים.

* שאילתא המציגה מה הם הג'קטים והוסטים הנמכרים ביותר עבור כל חברה, אשר נמכרו מעל 4 פעמים לפחות בהזמנות שונות.

ראינו בשאילתא הקודמת כי ג'קטים ווסטים הם המוצר הנמכר ביותר בחברה.  
שאילתא זו עונה על הצורך העסקי לדעת עבור כל חברה מה הם הנמכרים ביותר, כך נוכל להחזיק אותם במלאי או לבצע הסכם מראש עם החברה על הנחה להזמנה בכמות גדולה.

select P.Brand, P.ProductID, P.Type , NumOfOrders = count (P.ProductID)

from PRODUCTS as P JOIN DESIGNED\_PRODUCTS as DP

on P.ProductID= DP.ProductID JOIN ORDERS as O

on DP.CartID = O.CartID

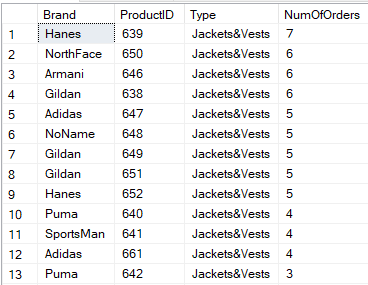
where P.Type = 'Jackets&Vests'

group by P.Type, P.Brand, P.ProductID

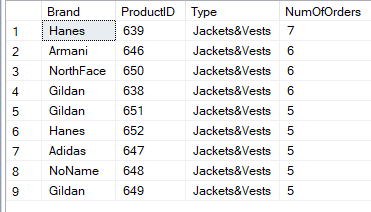
having count (P.ProductID) > 4

order by NumOfOrders desc

בעבור הרצת השאילתא ללא הגדרת מינימום פעמים בהם הוזמנה (having), קיבלנו 51 רשומות-



בעבור הרצת השאילתא כפי שמתוארת מעלה, כאשר הגדרנו מינימום הזמנות ל5 לפחות, קיבלנו 9 רשומות-



# שתי שאילתות SELECT מקוננות

* שאילתא המציגה את רשימת המדינות שלא בוצעו מהן הזמנות בשנת 2021 בחודשיים הראשונים. נתונים אלו יאפשרו לנו לנהל נכון יותר את עלויות המשלוח למדינות אלו, ולנהל פרסום ישיר במידת הצורך אם זהו שוק שהיינו מעוניינים בו.

SELECT Country

FROM CUSTOMERS

WHERE Country NOT IN (SELECT C.Country

FROM CUSTOMERS as C JOIN ORDERS as O ON

C.EmailAddress = O.EmailAddress

WHERE Year(O.DT) = 2021 AND MONTH (O.DT) IN (1,2)

GROUP BY C.Country)

GROUP BY Country

הפלט שקיבלנו הינו:



* שאילתא המציגה אילו לקוחות הם לקוחות חוזרים- זאת אומרת, אילו לקוחות ביצעו יותר מרכישה אחת באתר.

select C.EmailAddress, C.Name,

NumOfOrders = (select count(\*)

from ORDERS as OO join CUSTOMERS as CC

on CC.EmailAddress = OO.EmailAddress

where CC.EmailAddress = C.EmailAddress)

from ORDERS as O join CUSTOMERS as C

on C.EmailAddress = O.EmailAddress

where (select count(\*)

from ORDERS as OO join CUSTOMERS as CC

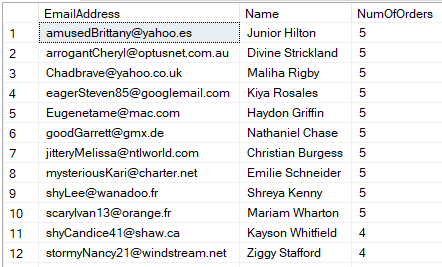
on CC.EmailAddress = OO.EmailAddress

where CC.EmailAddress = C.EmailAddress) > 1

group by C.EmailAddress, C.Name

order by NumOfOrders desc

הפלט שקיבלנו הינו (189 רשומות):



# שתי שאילתות מקוננות תוך שימוש במרכיבים נוספים

* **שאילתא המשתמשת בDELETE:**

שאילתא זו תאפשר לנו למחוק מהDB את המשתמשים אשר נכנסו לאתר, אך מעולם לא הזינו את האימייל שלהם בשום שלב בתהליך (שמירת העיצוב או ביצוע רכישה). שאילתא זו תועיל עבורנו משום שאין ערך במאגר הנתונים לכתובות IP אשר אינן מקושרות לאימייל- הן יבזבזו עבורנו מקום במאגר הנתונים, ולא נוכל לשלוח למשתמשים מבצעים, המלצות או מיילים אישיים לקידום מכירות.

DELETE FROM USERS

WHERE [IP Address] IN (

SELECT [IP Address]

FROM USERS

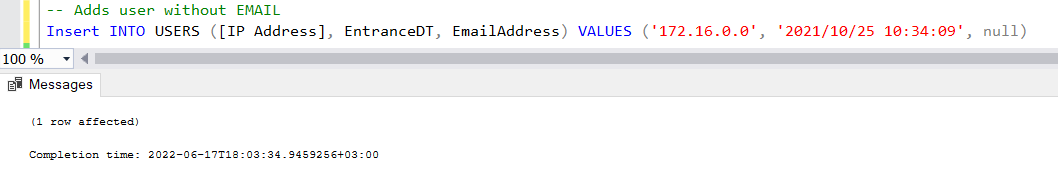
WHERE EmailAddress IS NULL)

בדיקה, הכנסת משתמש ללא כתובת אימייל לDB:

-- Adds user without EMAIL

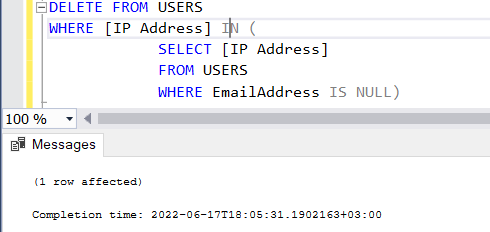
Insert INTO USERS ([IP Address], EntranceDT, EmailAddress) VALUES ('172.16.0.0', '2021/10/25 10:34:09', null)

המשתמש נכנס למערכת:





מחיקת הרשומה:



* **שאילתא המשתמשת בחיתוך:**

שאילתא זו תאפשר לנו לראות מה הם הם המוצרים שלא נמכרו מעולם, אבל כן היו בעגלות שננטשו.  
מצב זה יאפשר לנו לראות אילו מוצרים שלא נמכרו הם בעלי פוטנציאל למכירה, וכך נוכל להבחין בינם לבין מוצרים שלא נמכרו כלל וגם לא היה ביקוש עבורם (למשל יצירת הפרש בין כלל המוצרים שלא נמכרו לבין השאילתא הזאת).

השאילתא שיצרנו:

הטבלה הראשונה מייצגת את כלל המוצרים שלא נמכרו, הטבלה השנייה מייצגת את המוצרים בעגלות שננטשו.

select P.productID

from PRODUCTS as P

WHERE P.productID not in (select DP.ProductID

from DESIGNED\_PRODUCTS as DP join ORDERS AS O

ON O.CartID = DP.CartID join CARTS as C

ON C.CartID = O.CartID)

INTERSECT

select DP.ProductID

from CARTS as C join DESIGNED\_PRODUCTS as DP

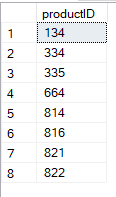
on DP.CartID =C.CartID

where C.CartID not in (select O.cartID

from DESIGNED\_PRODUCTS as DP join ORDERS as O

on DP.CartID = O.CartID)

הפלט שקיבלנו (8 רשומות):



## מטלה 2 – יישומי כלים מתקדמים

# View

בניית VIEW נפרד עבור שני צוותים שונים באתר, המבוסס על טבלת CUSTOMER.  
הVIEW רלוונטי משום ששני הצוותים צריכים לראות נתונים שונים הנוגעים למשתמש, ולאו דווקא את כל פרטיו:

1. VIEW עבור צוות Data Analysis בו יוכלו לראות את פרטי המשתמש מבחינת אימייל, מדינה וכו'

CREATE VIEW DATA\_ANALYZE\_TEAM\_VIEW AS

SELECT EmailAddress, Name, Country, City, Street, PhoneNumber

FROM CUSTOMERS

הטבלה שקיבלנו:



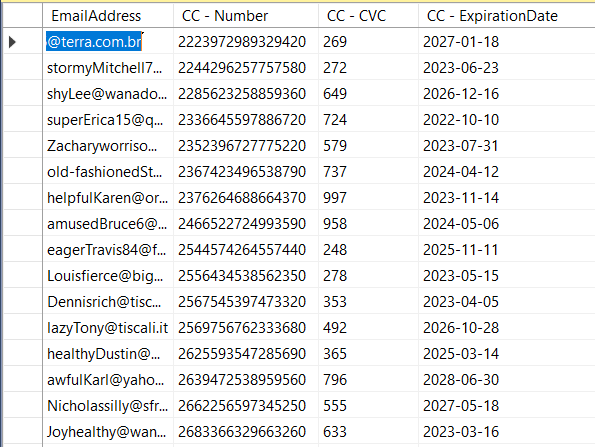
1. VIEW עבור צוות פיננסים אשר מציג גם את פרטי התשלום האישיים של כל לקוח:

CREATE VIEW FINANCE\_TEAM\_VIEW AS

SELECT EmailAddress, [CC - Number], [CC - CVC], [CC - ExpirationDate]

FROM CUSTOMERS

1. הטבלה שקיבלנו:



# פונקציות (Functions)

1. פונקציה סקלרית, אשר מחזירה עבור כל מותג את כמות ההזמנות מהמוצרים שלו.  
   הפונקציה רלוונטית עבורנו כאשר נרצה לנתח המשך עבודה עם מותג ספציפי ואפשרות לשיתוף פעולה.

CREATE FUNCTION OrdersPerBrand (@Brand Varchar(15))  
RETURNS int  
AS BEGIN  
DECLARE @OrderAmount int  
SELECT @OrderAmount = COUNT(\*)  
FROM PRODUCTS as P JOIN DESIGNED\_PRODUCTS as D

ON P.ProductID = D.ProductID  
JOIN ORDERS as O ON D.CartID = O.CartID

WHERE P.Brand = @Brand  
RETURN @OrderAmount END

שאילתת SELECT להפעלה, לדוגמא עבור החברה אדידס:

SELECT Brand, Orders = dbo.OrdersPerBrand('Adidas')  
FROM Products  
WHERE Brand = 'Adidas'  
GROUP BY Brand

הפלט שמתקבל:

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

הפלט שמתקבל לדוגמא עבור החברה פומה:

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

1. פונקציה המציגה את המכירות עבור כל חג, מהו סוג המוצר שנמכר, מהו הפריט שנמכר, וכמה נמכרו ממנו. יאפשר לנו לנהל מבצעים נכונים סביב החגים לפי המוצרים הנמכרים ביותר, ולהחזיק מראש במלאי הנדרש על סמך נתוני החגים.

מימשנו CASE עבור שם חג וחודשים, ברירת המחדל עבור חג שאינו ברשימה הינה ינואר.

-- Drop table before check

drop function SeasonalProducts

השאילתה שנבנתה:

CREATE FUNCTION SeasonalProducts (@Holiday Varchar(15)

RETURNS Table

AS RETURN

SELECT P.Type, P.ProductID, month = month(O.dt),

NumOfOrders = count (\*)

FROM PRODUCTS as P JOIN DESIGNED\_PRODUCTS as D

ON P.ProductID = D.ProductID JOIN ORDERS as O

ON D.CartID = O.CartID

WHERE  month (O.dt) = CASE WHEN @Holiday = 'Christmas' THEN 12

WHEN @Holiday = 'Ramadan' THEN 03  
WHEN @Holiday = 'Passover' THEN 04  
ELSE 01 END

Group BY P.Type, P.ProductID, month(O.dt)

שאילתה למימוש עבור חג פסח (מתקיים בחודש אפריל):

SELECT \*  
FROM SeasonalProducts('Passover')  
Order By NumOfOrders desc

הפלט שקיבלנו עבור פסח- 48 רשומות:

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

# Trigger פשוט

ניצור טבלת משפיעים. בכל הוספה של הזמנה מסוג fundraiser, נוסיף לטבלת משפיעים לקוח, במידה והוא ביצע 3 ומעלה הזמנות - ככה נדע מי המשפיעים שנרצה לשתף פעולה איתם, כחלק מהאג'נדה להובלה ותרומה חברתית שהחברה מקדמת.

יצירת טבלת משפיעים:

CREATE TABLE INFLUENCERS (

EmailAddress Varchar (40) not null,

PhoneNumber Varchar (20) not null,

Name Varchar (20) null,

Country Varchar (20) null,

CONSTRAINT PK\_INFLUENCERS PRIMARY KEY (EmailAddress),

CONSTRAINT CK\_Email\_INFLUENCERS CHECK (EmailAddress LIKE '%@%.%'),

CONSTRAINT AK\_PHONE\_INFLUENCERS UNIQUE (PhoneNumber))

יצירת הטריגר:

CREATE TRIGGER FundraisingOrder\_Trigger

ON ORDERS

FOR INSERT, UPDATE

AS

INSERT INTO INFLUENCERS

SELECT O.EmailAddress, C.PhoneNumber, C.Name, C.Country

FROM CUSTOMERS as C JOIN ORDERS as O ON C.EmailAddress = O.EmailAddress

WHERE O.PurchaseType = 'Fundraising'

GROUP BY O.EmailAddress, C.PhoneNumber, C.Name, C.Country

HAVING COUNT(\*) > 2

INTERSECT

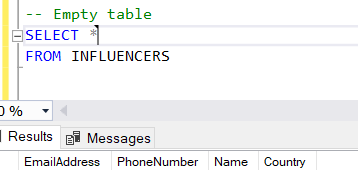
SELECT I.EmailAddress, C.PhoneNumber, C.Name, C.Country

FROM CUSTOMERS as C JOIN INSERTED as I ON C.EmailAddress = I.EmailAddress

WHERE I.PurchaseType = 'Fundraising'

GROUP BY I.EmailAddress, C.PhoneNumber, C.Name, C.Country

המחשת הטריגר טרם הפעלתו- קיבלנו טבלה ריקה:



על מנת לבצע את הבדיקה של התקינות מחקנו הזמנה והוספנו אותה מחדש, להלן הטבלה לאחר הפעלת הטריגר:

INSERT INTO ORDERS VALUES (333,'2022/04/14 8:32','Fundraising', 111, 1185, 'Erinbrave@gmx.net')

-- A customer added into the table

SELECT \*

FROM INFLUENCERS

-- DELETE FROM ORDERS WHERE OrderID = 333

-- DROP TABLE INFLUENCERS



# פרוצדורה שמורה ((Stored Procedure פשוטה

פרוצדורה המאפשרת לעדכן כאשר מוצר כבר אינו במלאי.

על מנת לא לפגוע באמינות המידע בשאר ההזמנות במערכת, החלטנו להגדיר שדה ביט (בדומה לבוליאני) במוצר הממחיש האם מוצר כבר יצא מהמלאי, ובנוסף תאריך הוצאתו מהמלאי (יתעדכן ליום הפעלת הפרוצדורה).

הפרצודורה תאפשר לצוות המתחזק את האתר או הצוות הלוגיסטי את הפעלתה ללא גישה למאגר הנתונים עצמו.

לשם כך הוספנו את השדות הבאים בטבלת PRODUCTS:

ALTER TABLE PRODUCTS ADD InStock bit

ALTER TABLE PRODUCTS ADD NotInStockDate Datetime

-- defult 0, 1 = discontinued

הפרצודורה:

DROP PROCEDURE sp\_NotInStockProduct

CREATE PROCEDURE sp\_NotInStockProduct @ProductID int

AS BEGIN

IF (SELECT Object\_ID('NotInStockProduct')) IS NOT NULL DROP Table NotInStockProduct

UPDATE PRODUCTS

SET

InStock = 1,

NotInStockDate = Getdate()

WHERE ProductID = @ProductID

END

מימוש הפרוצדורה:

EXECUTE sp\_NotInStockProduct 664

המוצר טרם הפעלת הפרצדורה:



המוצר לאחר הפעלת הפרוצדורה:



## מטלה 3 – כלים להצגת נתונים

תחילה יצרנו VIEW עבור ייבוא הנתונים לpower bi:

CREATE VIEW VIEW\_CustomInk AS

SELECT P.ProductID, P.Brand, P.Type, P.Color, P.Size, P.Price,

CUS.EmailAddress, CUS.City, CUS.Country, CUS.Gender,

C.CartID, C.DeliveryOption, O.OrderID, O.PurchaseType, [Date Of Purchase] = O.DT,

DP.Quantity

FROM CARTS AS C full JOIN DESIGNED\_PRODUCTS AS DP ON C.CartID = DP.CartID

full JOIN PRODUCTS AS P ON DP.ProductID = P.ProductID

JOIN DESIGNS AS D ON D.DesignID = DP.DesignID

full JOIN ORDERS AS O ON O.CartID = C.CartID

JOIN USERS AS U ON U.[IP Address] = D.[IP Address]

JOIN CUSTOMERS AS CUS ON CUS.EmailAddress = U.EmailAddress

**דו"ח עסקי-**

את הדו"ח העסקי שלנו בחרנו להציג לצוות ניהול שותפים עסקיים.

צוות זה יהיה אמון גם על הקשרים עם המותגים הנמכרים באתר, ולכן הדוח יעזור להם לקבל מידע אודות נתוני המכירות לכל מותג, המוצרים הנמכרים מהמותגים השונים והעדפותיהם של הלקוחות, זאת על מנת לנהל קשרים עסקיים רווחיים כלכלית עבור החברה ולהגדיל את המכירות.

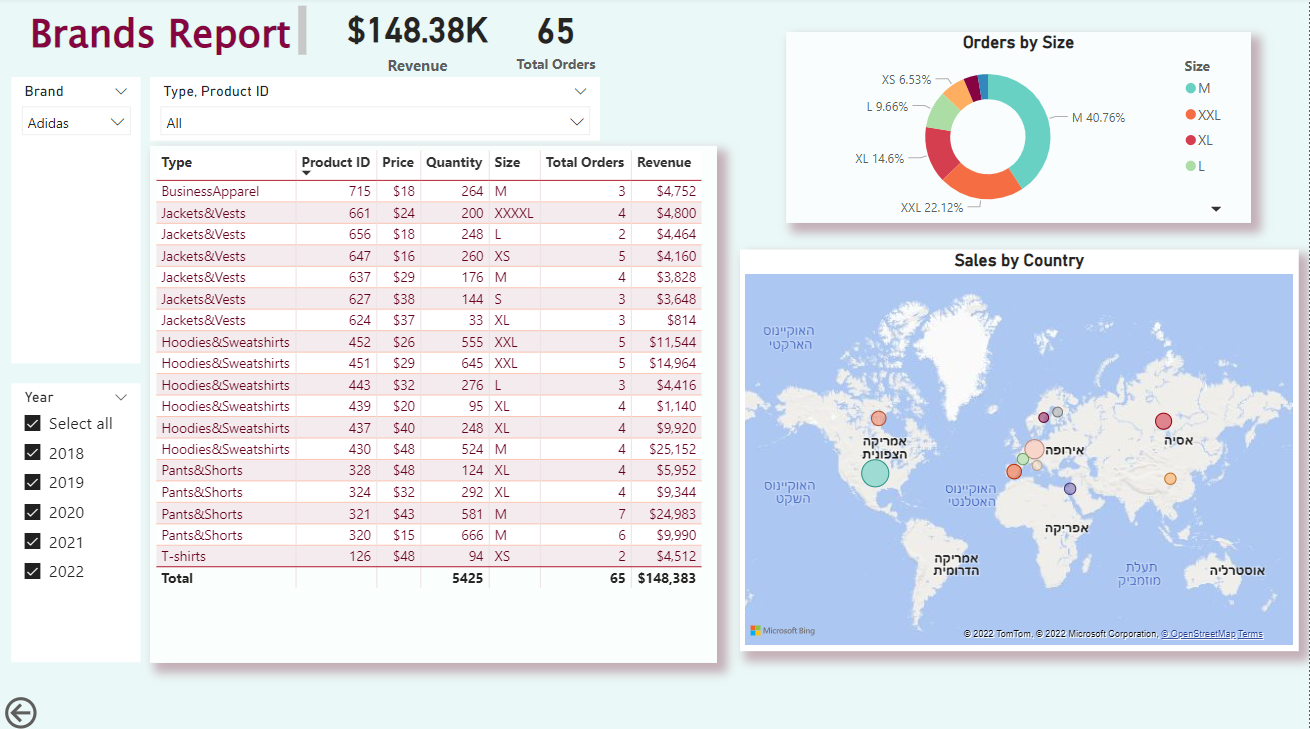
הדוח עונה על השאלות העסקיות הבאות:

1. מאילו מדינות מגיעים הלקוחות, ומה הם נתוני המכירות במדינות אלו? (כמות הזמנות ורווח).
2. מהו יחס ההזמנות על פי מידות עבור המוצרים?
3. מה הם מוצרי החברה שנמכרו באתר, באילו כמויות ומה היקף ההכנסה?

על מנת לענות על שאלות אלו בחרנו במגוון כלים גרפים להמחשת הנתונים הללו, עם זאת רצינו לוודא כי המידע אכן מדויק ומאפשר לצוות לקבל נתונים ממוקדים עבור כל מותג ופריטי המותג הרלוונטיים.

הוספנו לדוח פילטרים לסינון הנתונים:

1. סינון על פי שם המותג, חובה לבחור רק אחד.
2. סינון על פי סוג המוצר, ניתן לבצע סינון פנימי עבור קוד מוצר (בוצע באמצעות היררכיה), מאפשר לבחור מספר קטגוריות או מוצרים.
3. סינון על פי שנים, ניתן לבחור מספר שנים או אחת.

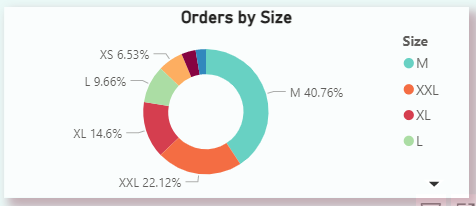


בסינון זה הפעלנו את הרכישות עבור ADIDAS לכלל מוצרי החברה, עבור כל השנים.

הצגת כמות הזמנות (המתייחס לכמות פריטים בהזמנה) עבור כל מידה:

יאפשר לחברה לנהל קשר עם המותג ולהציע מבחר מידות רלוונטיים יותר לקהל הלקוחות, ביחס לקהל הלקוחות שרוכש את המותג.

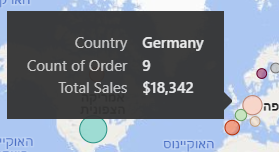
לדוגמא- בסינון שלנו ניתן לראות כי המידות המועדפות ברכישות היו M וXXL, לכן נבחר לשתף פעולה עם המותג ולוודא כי המוצרים שנזמין מהם עבור האתר שלנו יהיו בטווח המידות הנ"ל, וכי יהיה מלאי נרחב ממידות אלו.



מכירות עבור כל מדינה למוצרים שנבחרו בסינון:

המחשה ויזואלית זו תאפשר לנו לראות את נתח המכירות העולמי של המותג (ביחס למוצרים שנבחרו), כל פירוט על מדינה יציג גם את הכמות שנמכרה והרווח מאותה מדינה.

תמונה שמכילה מפה

התיאור נוצר באופן אוטומטי  


טבלה המציגה את פרטי ההכנסות וההזמנות עבור מוצרי החברה על פי הסינון שנבחר:

בחרנו להציג את סוג המוצר, מזהה המוצר, מחירו לצרכן, הכמות שהוזמנה מנו (סה"כ, בכל ההזמנות בתקופת הזמן), המידה, מספר ההזמנות שבוצעו עבור אותו מוצר והרווח הכללי ממנו.

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

נתונים אגרגטיביים עבור כל מותג:

הרווח הסופי מהסינון שנבחר; מספר ההזמנות שהיו עבור אותו סינון.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

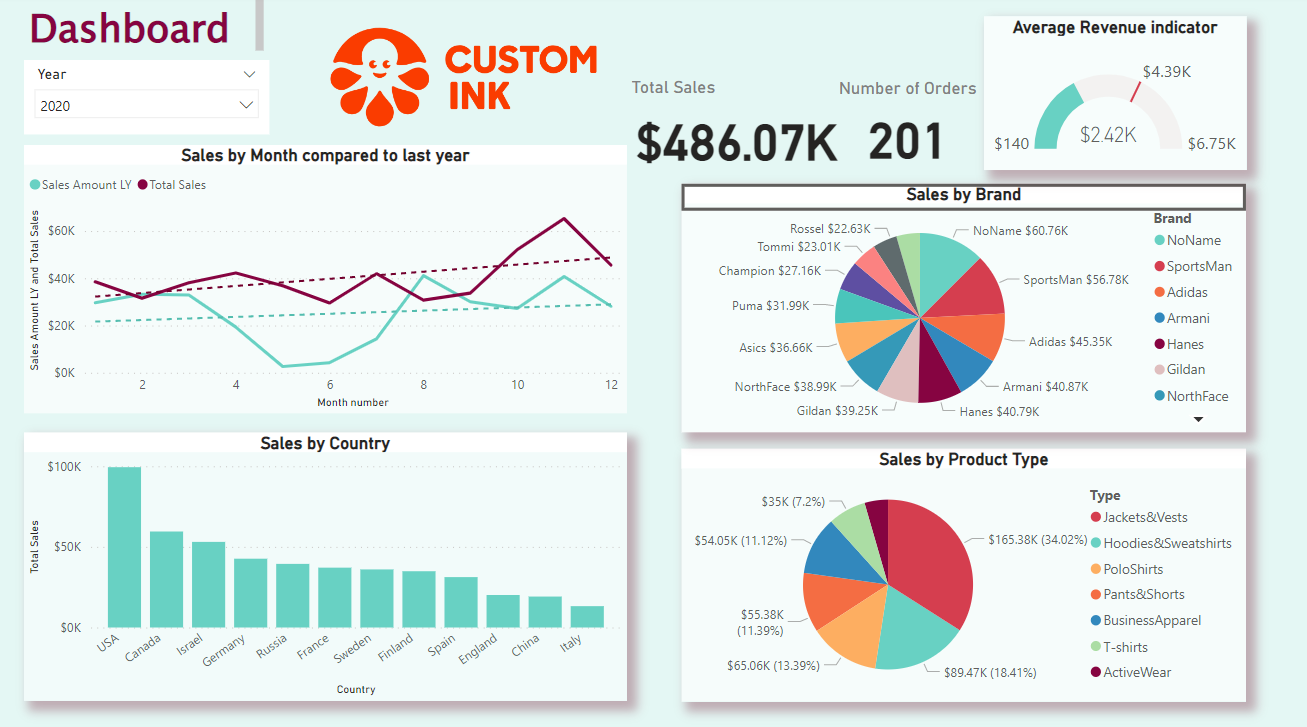
**לוח מחוונים-**

את הDashboard שלנו, כחלק מכך שמוצג לדרג הבכיר בחברה, רצינו למקד עבור הכנסות החברה בניתוחים שונים. נתונים אלו יעזרו לדרג הניהולי לראות את המידע בראיה כוללת, ובנוסף לסווג עבור מידע ספציפי בו יבחרו להתמקד, לשם קבלת החלטות עתידיות.

השאלות העסקיות עליהן לוח המחוונים שלנו עונה:

1. מה הן המכירות של החברה לפי החודשים ומהי המגמה?
2. מה הן מכירות החברה השנה ביחס לשנה שעברה?
3. מה הוא פילוח ההכנסות על פי המדינות להן אנו מוכרים?
4. מה הן המכירות על פי מותג?
5. מה הן המכירות על פי סוג המוצר?
6. מהו הרווח של החברה מהזמנה ממוצעת?

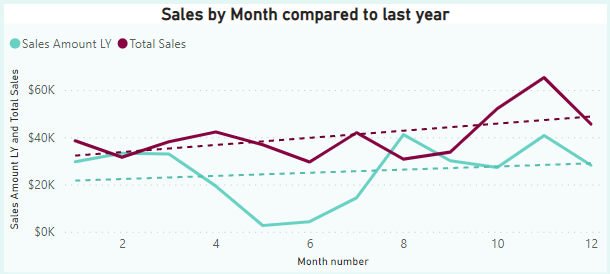
הפילטר בו השתמשנו לסינון הדשבורד הינו עבור שנה.



המכירות השנה עבור כל חודש, בהשוואה לשנה שעברה:

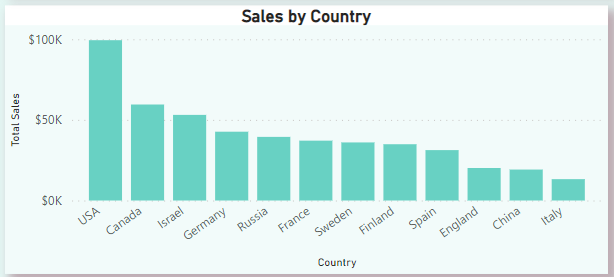
גרף המציג את המכירות הכוללות שהיו השנה (בבורדו), את המכירות שהיו בשנה שעברה (בטורקיז) ואת קווי המגמה עבור כל שנה.

תרשים זה יאפשר לנו לראות מה הן מגמות המכירות בחברה, היכן אנו עומדים ביחס לשנה שעברה, והאם אנו ממשיכים במגמה של עליית המכירות או ירידה במכירות, בהתאם לקו המגמה של שנה שעברה.



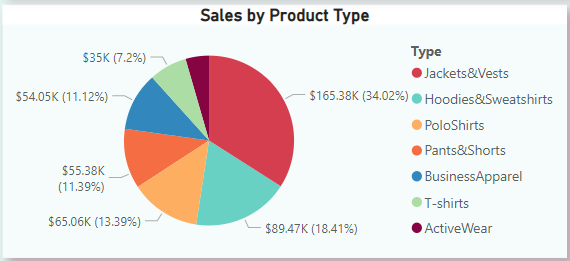
המכירות עבור כל מדינה:

תרשים זה יאפשר לנו לראות אילו מדינות מובילות במכירות, ויאפשר לדרג הניהולי לבחון האם עמד ביעדים שלו מבחינת המכירות במדינות אלו, ולבחור באילו מדינות ירצה לקדם את השיווק ולפנות לקהלי יעד נוספים.



מכירות עבור סוג מוצר:

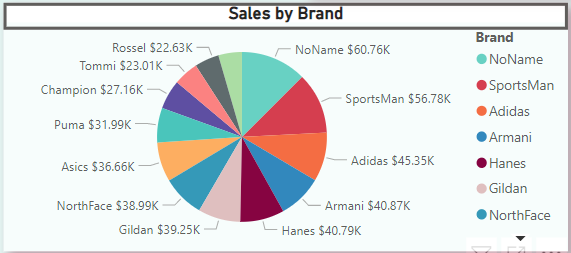
פילוח נתונים זה יאפשר לדרג הניהולי לראות אילו סוגי מוצרים הכי רווחיים עבור החברה, להגדיל את ההיצע עבור סוג מוצר זה, או בנוסף להציע מבצעים עבור סוגי מוצרים פחות נמכרים ולהגדיל את המכירות עבורם.



מכירות עבור מותג:

פילוח זה מציג לנו את נתח ההכנסה מכל מותג הנמכר באתר, ויאפשר לנו לבצע שיתופי פעולה רלוונטיים יותר עם החברות, ולהציע מגוון רחב יותר של המוצרים.

בנוסף, בכותרת של התרשים ישנו קישור לדוח העסקי עבור שותפים עסקיים (Brand Report).

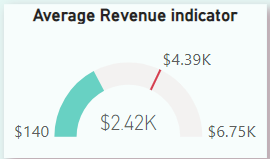


שעון עבור הכנסה ממוצעת מהזמנה:

כחלק מיעדי החברה רצינו להציג מדד הממחיש רווח ממוצע מהזמנה.

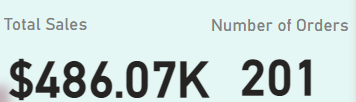
המינימום עבור שעון זה הינו ההזמנה המינימלית באותה שנה; המקסימום הינו ההזמנה המקסימלית (הרווחית ביותר) עבור אותה שנה.

היעד אותו הצמדנו למדד זה הינו 65% מההזמנה המקסימלית, זאת אומרת- נשאף שההזמנה הממוצעת תעמוד על 65% מההזמנה המקסימלית לאותה שנה.



נתונים אגרגטיביים:

לבסוף, רצינו להציג את הנתונים הכמותיים עבור אותה שנה- הכנסות מהמכירות ומספר ההזמנות עבור אותה שנה.



## מטלה 4 – יישומים מתקדמים

## שאילתות עסקיות המשלבות Window Functions

1. השאילתה מציגה את הכנסות כלל ההזמנות מכל המדינות, ומציגה מיהי המדינה הכי רווחית, ומיהי המדינה הכי פחות רווחית- ביצענו שימוש בפונקציות "חלון" – LAST\_VALUEו FIRST\_VALUE.

הרעיון העסקי – נרצה לדעת לאיזה מדינות יש את הפוטנציאל הכנסות הגדול או הקטן ביותר וכך נוכל לנהל נכון יותר את עלויות המשלוח למדינות אלו, ולנהל פרסום ישיר במידת הצורך.

יצירת VIEW עבור ההזמנה:

CREATE VIEW VIEW\_ORDERS as

SELECT C.EmailAddress, C.Country, O.PurchaseType, DP.Quantity, P.Price

FROM CUSTOMERS AS C JOIN ORDERS AS O

ON C.EmailAddress = O.EmailAddress

JOIN DESIGNED\_PRODUCTS AS DP ON DP.CartID = O.CartID

JOIN PRODUCTS AS P ON P.ProductID = DP.ProductID

שאילתת הSELECT :

SELECT C.Country, [Total amount] = SUM(C.[Total amount]),

FIRST\_VALUE (C.Country) OVER (ORDER BY SUM(C.[Total amount])) [Most unprofitable Country],

LAST\_VALUE (C.Country) OVER (ORDER BY SUM(C.[Total amount]) RANGE BETWEEN

UNBOUNDED PRECEDING AND UNBOUNDED FOLLOWING) [Most profitable Country]

FROM (SELECT Country, [Total amount] = SUM(Price\*Quantity)

FROM VIEW\_ORDERS

GROUP BY Country) AS C

GROUP BY Country, C.[Total amount]

ORDER BY [Total amount] DESC

הפלט שקיבלנו:

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

1. השאילתה מציגה לכל סוג משלוח (Fundraising, Group Order, Individual Order) את כמות הלקוחות שביצעו הזמנה ובאילו מדינות. בנוסף, השאילתא מדרגת את המדינות מ1 עד 12 לפי סוג משלוח זה.  
   ביצענו שימוש בפונקציות "חלון" – RANK וNTILE

הרעיון עסקי - נרצה לעודד הזמנות ולכן נרצה לדעת האם במדינה מסוימת יש למשל ריבוי של הזמנות מסוג Fundraising וכך נוכל להתאים את הפרסומות, תנאי ההזמנה ומחירי המשלוח לפי "סוג הלקוחות".

SELECT X.Country, COUNT (DISTINCT X.EmailAddress) [Number Of Customers],

RANK () OVER (PARTITION BY X.PurchaseType ORDER BY COUNT (DISTINCT X.EmailAddress) DESC) [RANK By Number Of orders],

NTILE (3) OVER (ORDER BY X.PurchaseType) [Group Of Purchase Type], X.PurchaseType

FROM (SELECT Country, EmailAddress, PurchaseType

FROM VIEW\_ORDERS) AS X

GROUP BY Country, PurchaseType

ORDER BY Country, [RANK By Number Of orders]

הפלט שהתקבל כולל 36 רשומות :

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

1. הפונקציה מציגה הכנסות של מותג בשנה מסוימת בהשוואה לשנה שעברה, בנוסף מציגה את השינוי ברווח באחוזים בין השנה הנוכחית לשנה הקודמת לה ומדרגת את הכנסות המותג ביחס לעצמו לאורך השנים.  
   השתמשנו בפונקציות "חלון" – RANK וLAG

הרעיון העסקי – נרצה לדעת כיצד הכנסות המותג מתנהלות לאורך השנים, וזאת על מנת להפיק לקחים האם לקוחות מרוצים מהמותג (לדוגמא המידה אמיתית, הבד נעים), האם המחיר שלו משתלם, האם צריך להשקיע יותר בקידום שלו באתר.

יצרנו VIEW עבור השאילתא:

CREATE VIEW VIEW\_Sales as

SELECT YEAR = YEAR (O.DT), P.Brand,

[Quantity Per Year] = SUM(DP.Quantity),

[This year profit] = SUM (P.Price\*DP.Quantity)

FROM ORDERS AS O JOIN DESIGNED\_PRODUCTS AS DP

ON O.CartID = DP.CartID

JOIN PRODUCTS AS P

ON P.ProductID = DP.ProductID

GROUP BY YEAR (O.DT), P.Brand

יצרנו SELECT לשאילתא:

SELECT Sales.YEAR , Sales.Brand, Sales.[This year profit],

LAG ( Sales.[This year profit],13) OVER (ORDER BY YEAR) AS [Previous year profit],

[Change in % from previous year] = ROUND (( Sales.[This year profit] / Sales.[Previous year profit])\*100,2) ,

RANK() OVER (PARTITION BY Brand ORDER BY Sales.[This year profit] ASC) Brand\_RANK\_By\_Years

FROM (SELECT YEAR, Brand, [This year profit], LAG ([This year profit],13) OVER (ORDER BY YEAR) AS [Previous year profit]

FROM VIEW\_Sales) AS Sales

ORDER BY Brand, Year

הפלט שקיבלנו כולל 64 רשומות :

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

## שילוב מערכתי של מספר כלים

ניצור פרוצדורה שמטרתה למחוק מוצר, ולוודא כי גם המוצרים המעוצבים של אותו מוצר נמחקים. לפני המחיקה נבצע שאילת SELECT שמחזירה את מס' המוצרים כרגע, מס' המוצרים המעוצבים ומס' ההזמנות שכללו את המוצר שאותו נרצה למחוק. לאחר מכן תתבצע מחיקה של המוצר, מה שיפעיל טריגר שיוודא כי קודם כל ימחקו המוצרים המעוצבים של אותו מוצר. בסיום הפרוצדורה תתבצע שאילתת SELECT נוספת כדי לוודא כי אכן המחיקות בוצעו כהלכה.

-- Creating function

CREATE FUNCTION OrdersByProducts (@ProductID INT)

RETURNS INT

AS BEGIN

DECLARE @Orders INT

SELECT @Orders = COUNT(DISTINCT O.OrderID)

FROM PRODUCTS as P JOIN DESIGNED\_PRODUCTS as D ON P.ProductID = D.ProductID

JOIN ORDERS as O ON D.CartID = O.CartID

WHERE P.ProductID = @ProductID

RETURN @Orders

END

-- Creating Trigger

CREATE TRIGGER Delete\_Designed\_Products

ON PRODUCTS

INSTEAD OF DELETE

AS

DELETE FROM DESIGNED\_PRODUCTS

WHERE DESIGNED\_PRODUCTS.ProductID = (SELECT ProductID FROM deleted)

DELETE FROM PRODUCTS

WHERE PRODUCTS.ProductID = (SELECT ProductID FROM deleted)

-- Creating Procedure

CREATE PROCEDURE SP\_DeleteProduct @ProductID INT

AS

SELECT ProductsBefore = COUNT(DISTINCT ProductID), DesignedProductsBefore = (SELECT COUNT(\*) FROM DESIGNED\_PRODUCTS), OrdersNum = dbo.OrdersByProducts(@ProductID)

FROM PRODUCTS

DELETE FROM PRODUCTS

WHERE PRODUCTS.ProductID = @ProductID

SELECT ProductsAfter = COUNT(DISTINCT ProductID), DesignedProductsAfter = (SELECT COUNT(\*) FROM DESIGNED\_PRODUCTS)

FROM PRODUCTS

-- Insert Product & Designed Products & Orders -------------------------------

INSERT INTO PRODUCTS (ProductID, Brand, Type, Style, Color, Size, Fabric, MinQuantity, Price) VALUES (8582, 'Puma', 'Pants&Shorts', 'MeshShorts', 'Black', 'S', 'Mesh', 6, 45)

INSERT INTO CARTS (CartID, ShippingArea, DeliveryOption) VALUES

(10000, 'INTERNATIONAL', 'Rush'),

(10001, 'INTERNATIONAL', 'Rush'),

(10002, 'INTERNATIONAL', 'Rush')

INSERT INTO DESIGNED\_PRODUCTS (ProductID, DesignID, Placement, CartID, Quantity) VALUES

(8582, 1, 'Front Adult', 10000, 350),

(8582, 2, 'Front Adult', 10001, 220),

(8582, 3, 'Front Adult', 10002, 150)

INSERT INTO ORDERS (OrderID, DT, PurchaseType, TrackingNumber, CartID, EmailAddress) VALUES

(700, '2021/10/25', 'Fundraising', 565, 10000, 'adventurousBrian44@uol.com.br'),

(701, '2021/03/29', 'Group Order', 852, 10002, 'adventurousBrian44@uol.com.br')

EXECUTE SP\_DeleteProduct 8582

------------------------------------------------------------------------------

-- DELETE ORDERS & CARTS

DELETE FROM ORDERS WHERE OrderID = 700

DELETE FROM ORDERS WHERE OrderID = 701

DELETE FROM CARTS WHERE CartID = 10000

DELETE FROM CARTS WHERE CartID = 10001

DELETE FROM CARTS WHERE CartID = 10002

**פלט שאילתות ה SELECT:**

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

## דו"ח המושתת על שימוש בפסקת WITH מורכבת

מטרת הדו"ח היא לנתח את אופי הרכישות שמבצעים הלקוחות המובילים באתר (נבחרו על פי יחס בין הוצאתם הכוללת לפי כמות הרכישות- כאשר גדול מ4000$ נכנסו לרשימה).

נוכל לראות מהי ההוצאה הממוצעת ללקוח בכל רכישה, מה הן מספר ההזמנות הממוצעת לכל שנה, מספר העיצובים אותם יצר ביחס למספר ההזמנות שביצע, ופרטים אודות הלקוח עצמו.

-- Creating Function

CREATE FUNCTION YearRange (@Email Varchar(100))

RETURNS INT

AS BEGIN

DECLARE @Range INT

SELECT @Range = MAX(Year(O.DT)) - MIN(Year(O.DT))

FROM CUSTOMERS as C JOIN ORDERS as O ON C.EmailAddress = O.EmailAddress

WHERE C.EmailAddress = @Email

RETURN @Range

END

-- Function check

SELECT EmailAddress, Range = dbo.YearRange(CUSTOMERS.EmailAddress)

FROM CUSTOMERS

-- WITH

WITH

CUSTOMER\_ORDERS AS (

SELECT C.EmailAddress, Orders = COUNT(DISTINCT OrderID)

FROM CUSTOMERS as C JOIN ORDERS as O ON C.EmailAddress = O.EmailAddress

GROUP BY C.EmailAddress

),

ORDER\_DETAILS AS (

SELECT CO.EmailAddress, O.OrderID, O.DT, O.CartID

FROM CUSTOMER\_ORDERS as CO JOIN ORDERS as O ON CO.EmailAddress = O.EmailAddress

),

TRANSACTIONS\_DETAILS AS (

SELECT OD.EmailAddress, TotalAmount = SUM(P.Price \* DP.Quantity)

FROM ORDER\_DETAILS as OD JOIN DESIGNED\_PRODUCTS as DP ON OD.CartID = DP.CartID

JOIN PRODUCTS as P ON DP.ProductID = P.ProductID

GROUP BY OD.EmailAddress

),

CUSTOMER\_DESIGNS AS (

SELECT C.EmailAddress, Designs = COUNT(\*)

FROM CUSTOMERS as C JOIN USERS as U ON C.EmailAddress = U.EmailAddress

JOIN DESIGNS as D ON U.[IP Address] = D.[IP Address]

GROUP BY C.EmailAddress

)

-- Main Query

SELECT C.EmailAddress, C.Gender, C.Country,

Orders = CO.Orders,

[Average Orders Per Year] = CASE

WHEN dbo.YearRange(C.EmailAddress) > 0 THEN (CO.Orders / CAST(dbo.YearRange(C.EmailAddress) AS DECIMAL(3,1)))

WHEN dbo.YearRange(C.EmailAddress) = 0 THEN CO.Orders

END,

TotalAmount = TD.TotalAmount,

[Average Expense Per Order] = TD.TotalAmount / CO.Orders,

[Designs-Orders Ratio] = CD.Designs / CAST(CO.Orders AS DECIMAL(4,1))

FROM CUSTOMERS as C JOIN TRANSACTIONS\_DETAILS as TD ON C.EmailAddress = TD.EmailAddress

JOIN CUSTOMER\_ORDERS as CO ON CO.EmailAddress = C.EmailAddress

JOIN CUSTOMER\_DESIGNS as CD ON CD.EmailAddress = C.EmailAddress

WHERE C.EmailAddress IN (SELECT EmailAddress FROM TRANSACTIONS\_DETAILS WHERE TotalAmount / CO.Orders > 4000)

## ORDER BY [Average Expense Per Order] DESC

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטיהפלט המתקבל – 22 רשומות: