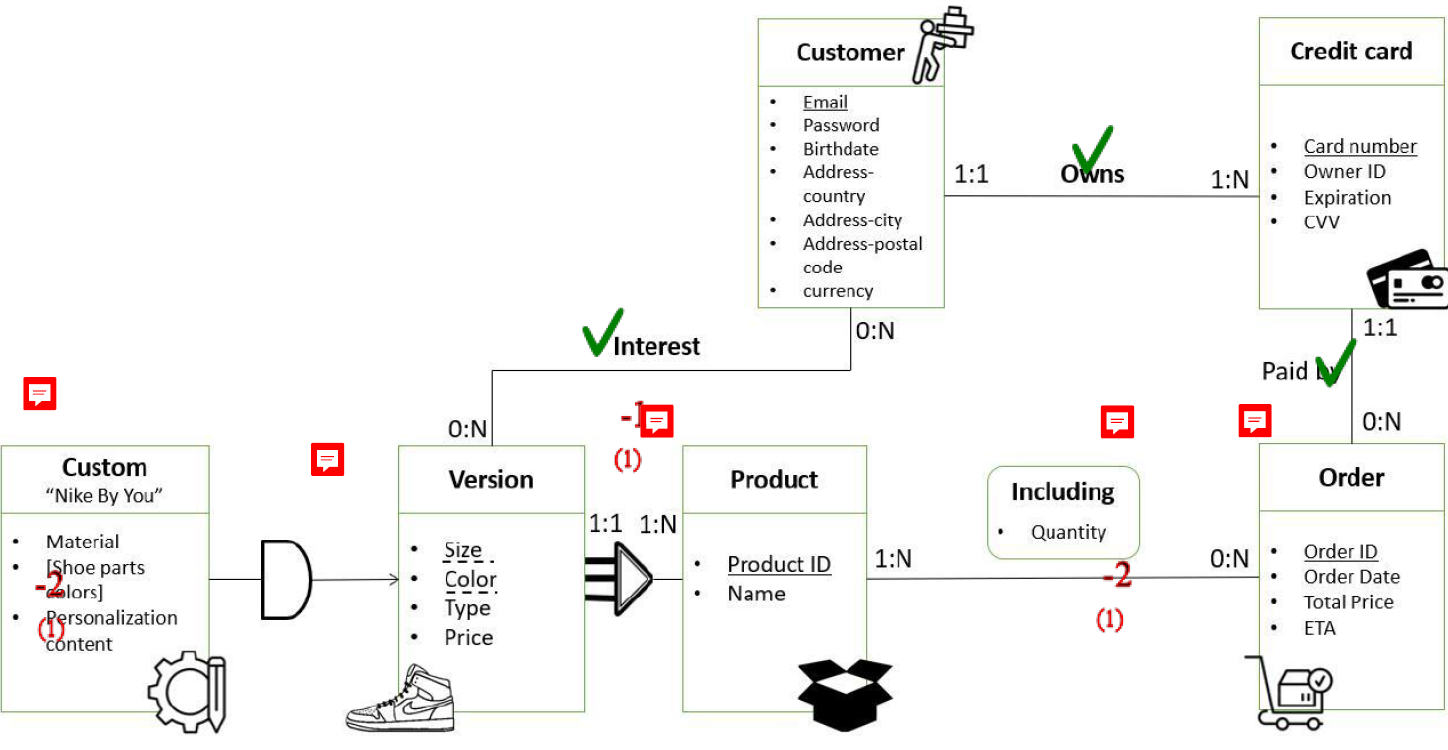
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מס' קבוצה | אתר | | | תאריך הגשה |
| 16 | [www.nike.com](http://www.nike.com) | | | 12/01/2022 |
| חברי הצוות - מספרי ת.ז | | | | |
| 316512987 | | 318404191 | 207080581 | |

## פרויקט בסיסי נתונים – חלק ג'

## הערות חלק ב:

תרשים ERD מקורי:

ההערות מימין לשמאל:

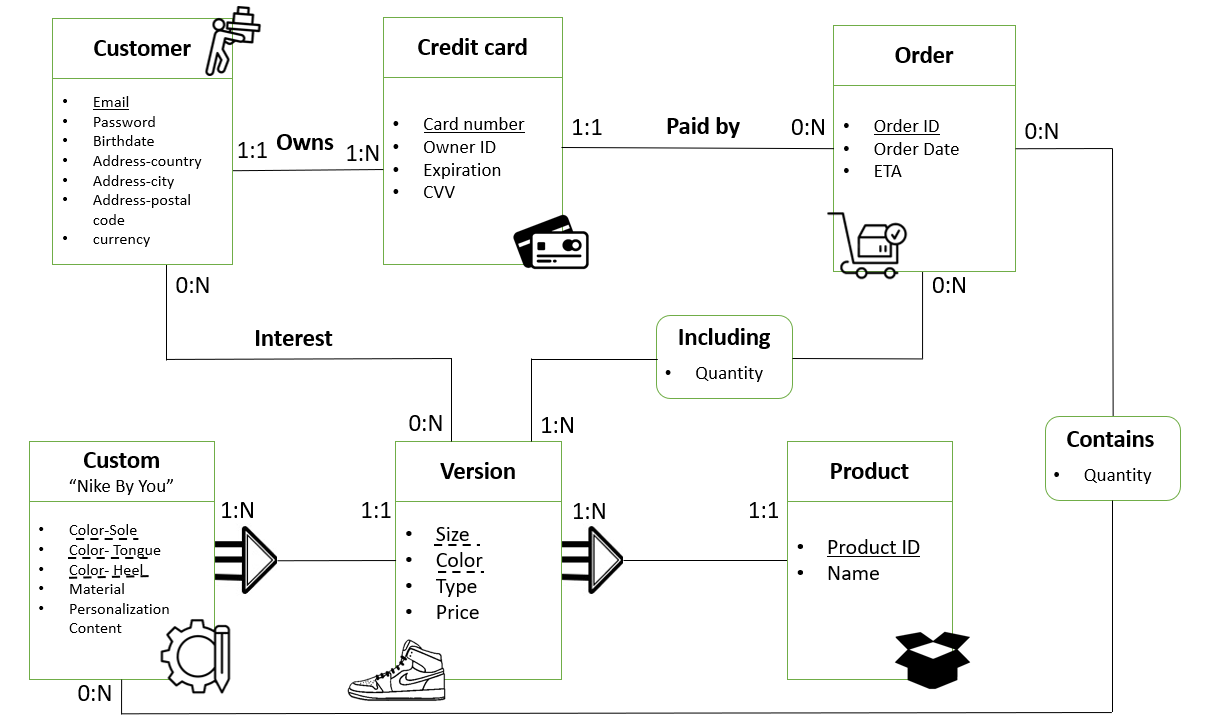
לא ירד לכם על כך ציון אך שימו לב לתקן לקראת חלק ג` - תוכלו לשים את CUSTOM גישות חלשה של גרסה וגם של לקוח וכך להשתמש רק במזהה קטן עבורה.

שימו לב גם לקשר את הגרסה המותאמת להזמנה.

מודל טבלאי:

1. **CUSTOMERS** (Email, Password, Birthdate, Address-Country, Address-City, Address-PostalCode, Currency)
2. **PRODUCTS** (Product ID, Name)
3. **ORDERSCONTENT** (Order ID(ORDERS) ,Product ID(Products),Quantity)
4. **VERSIONS** (Product ID (Products), Size, Color, Type , Price)
5. **INTEREST** (Email(Customers), {Product ID, Size, Color} (Versions))
6. **CREDITCARDS** (Card Number, Owner ID, Expiration, CVV, Customer ID(Customer))
7. **ORDERS** (Order ID, Order Date, ETA, Total Price, Card Number(CreditCards))
8. **CUSTOM** ({Product ID, Size, Color}(Versions), Material, Personalization Content)
9. **SHOEPARTS** ({Product ID, Size, Color}(Versions), Shoe Part Color)

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

**תרשים ERD מתוקן:**

**שינויים:**

* ישות חלשה, Custom תחת Version. בהורשה לא ניתן להזמין פעמיים את אותה הנעל באותו הצבע.
* ניתוק הקשר בין Product ל- Order וחיבור בין Custom ל- Order ובין Version ל- Order.
* שינוי השדות תחת Custom. פיצול השדה מרובה הערכים למזהים חלקיים.
* מחיקת השדה Total Price תחת Order. השדה יחושב כחלק מתהליך ההזמנה.

**מודל טבלאי מעודכן:**

* **CUSTOMERS** (Email, Password, Birthdate, Address-Country, Address-City, Address-PostalCode, Currency)
* **PRODUCTS** (Product ID, Name, Category)
* **ORDERSCONTENT** (Order ID(Orders) ,Product ID(Versions),Quantity)
* **VERSIONS** (Product ID (Products), Size, Color, Price)
* **INTEREST** (Email(Customers), {Product ID, Size, Color} (Versions))
* **CREDITCARDS** (Card Number, Owner ID, Expiration, CVV, Customer ID(Customers))
* **ORDERS** (Order ID, Order Date, ETA, Total Price, Card Number(CreditCards))
* **CUSTOM** ({Product ID, Size, Color}(Versions), Color- Sole, Color- Tongue, Color- Heel, Material, Personalization Content)
* **CUSTOMINORDER** (Order ID(Orders) , {Product ID, Size, Color} (Custom), Quantity)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Table** | **Field** | **Data Type** | | **Size (Bytes)** |
| **Customers** | Email | Varchar(50) | | 50 |
| Password | Varchar(20) | | 20 |
| Birthdate | Date | | 3 |
| Address-Country | Varchar(20) | | 20 |
| Address-City | Varchar(50) | | 50 |
| Address-PostalCode | Varchar(20) | | 20 |
| Currency | Varchar(20) | | 20 |
| Cupon | Demical(10,2) | | 9 |
| **Total Record Size (Bytes)** | | | **192** |
| **OrdersContent** | OrderID(Orders) | Int | | 4 |
| PID (Products) | Int | | 4 |
| Quantity | Int | | 4 |
| **Total Record Size (Bytes)** | | | **12** |
| **Versions** | PID (Products) | Int | | 4 |
| Size | Decimal(10,2) | | 9 |
| Color | Varchar(20) | | 20 |
| Price | Decimal(10,2) | | 9 |
| **Total Record Size (Bytes)** | | | **42** |
| **Products** | PID | Int | | 4 |
| Name | Varchar(30) | | 30 |
| Category | Varchar(30) | | 30 |
| **Total Record Size (Bytes)** | | | **64** |
| **Interest** | Email (Customers) | Varchar(50) | | 50 |
| PID (Custom) | Int | | 4 |
| Size (Version) | Decimal(10,2) | | 9 |
| Color (Version) | Varchar(20) | | 20 |
| **Total Record Size (Bytes)** | | | **83** |
| **CreditCards** | CardNumber | Varchar(20) | | 20 |
| OwnerID | Int | | 4 |
| Expiration | Date | | 3 |
| CVV | Int | | 4 |
| CustomerID (Customer) | Varchar(50) | | 50 |
| **Total Record Size (Bytes)** | | | **81** |
| **Orders** | OrderID | Int | | 4 |
| OrderDate | Datetime | | 8 |
| ETA | Date | | 3 |
| TotalPrice | Money | | 8 |
| CardNumber (CreditCards) | Varchar(20) | | 20 |
| UseCupon | Decimal(10,2) | | 9 |
| **Total Record Size (Bytes)** | | | **52** |
| **Custom** | PID (Products) | Int | | 4 |
| Size (Versions) | Decimal(10,2) | | 9 |
| Color (Versions) | Varchar(20) | | 20 |
| Color- Sole | Varchar(20) | | 20 |
| Color- Tongue | Varchar(20) | | 20 |
| Color- Heel | Varchar(20) | | 20 |
| Material | Varchar(30) | | 30 |
| PersonalizationContent | Varchar(50) | | 50 |
| **Total Record Size (Bytes)** | | | **173** |
| **Countries** | Country | Varchar(20) | | 20 |
| **Total Record Size (Bytes)** | | | **20** |
| **Currencies** | Currency | Varchar(20) | | 20 |
| **Total Record Size (Bytes)** | | | **20** |
| **Categories** | Category | | Varchar(30) | 30 |
|  | | | **30** |
| **Sizes** | Size | | Decimal(10,2) | 9 |
|  | | | **9** |
| **Total Record Size (Bytes)** | | | | **778** |

## מטלה 1 – שאילתות

# שתי שאילתות SELECT ללא קינון

1. בדיקת המידה הנפוצה ביותר עבור כל נעל. זאת, כדי לאפיין את קהל היעד העיקרי של המוצרים, ובאמצעות זאת לשווק אליו בהתאמה. בנוסף, באמצעות הנתון יהיה ניתן לצפות את מידות הנעליים שכדאי לייצר.

SELECT DISTINCT PR.Name, v.Size, [Num Of Orders]=Sum(OC.Quantity)

FROM dbo.PRODUCTS AS PR JOIN dbo.VERSIONS AS V ON V.ProductID = PR.ProductID

JOIN dbo.ORDERSCONTENT AS OC ON OC.ProductID = PR.ProductID

WHERE PR.CATEGORY = ('Shoe')

GROUP BY PR.Name, V.Size

ORDER BY [Num Of Orders] DESC

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

2. בדיקת המדינות בהן נמכרו פחות מ-200 יחידות מסוג Accessories ב-5 השנים האחרונות. זאת, כדי לבדוק באילו מדינות אין ביקוש מספק לקטגוריה זו, ובהתאם אפשר לשקול להפחית מכמות הייצור הייעודי אליהן.

SELECT Country= c.[Address-Country], [Num Of Units Ordered] = SUM(OC.Quantity)

FROM dbo.CUSTOMERS AS C JOIN dbo.CREDITCARDS AS CC ON CC.CustomerID = C.Email

JOIN dbo.ORDERS AS O ON O.CardNumber = CC.CardNumber

JOIN dbo.ORDERSCONTENT AS OC ON OC.OrderID = O.OrderID

JOIN dbo.PRODUCTS AS P ON P.ProductID = OC.ProductID

WHERE p.CATEGORY = ('Accessories') AND YEAR(o.OrderDate) BETWEEN '2016' AND '2021'

GROUP BY c.[Address-Country]

HAVING SUM(OC.Quantity) < 200

ORDER BY [Num Of Units Ordered]

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

# שתי שאילתות SELECT מקוננות 10%

1. הצגת המוצרים שלא נמכרו בשנתיים האחרונות. בעקבות נתונים אלה, נוכל לבחון את כדאיות מכירות המוצרים. בנוסף, ניתן לעדכן את השיווק בהתאם.

SELECT ProductID

FROM PRODUCTS

WHERE ProductID NOT IN( SELECT ProductID

FROM dbo.ORDERSCONTENT AS oc JOIN dbo.ORDERS AS o ON o.OrderID = oc.OrderID

WHERE YEAR(o.OrderDate) BETWEEN 2019 AND 2021)

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

1. בדיקת היחס בין סך כל ההכנסות של החברה לבין כל קטגוריה. כך נוכל לדעת באיזה אופן עלינו להתייחס לכל קטגוריה. קטגוריות חלשות יש לשדרג וככל הנראה לשווק בצורה חזקה יותר, וקטגוריות חזקות להשאיר כפי שהן, היות שהן מתפקדות כראוי.

SELECT P.CATEGORY, [Total Revenue] = SUM(V.Price\*OC.Quantity), RATIO =

(SUM(V.Price\*OC.Quantity)/ (SELECT

SUM(V.Price\*oc.Quantity)

FROM VERSIONS AS V JOIN ORDERSCONTENT AS OC ON V.ProductID=OC.ProductID

JOIN ORDERS AS O ON OC.OrderID=O.OrderID

JOIN PRODUCTS AS P ON P.ProductID=V.ProductID))

FROM VERSIONS AS V JOIN ORDERSCONTENT AS OC ON V.ProductID=OC.ProductID

JOIN ORDERS AS O ON OC.OrderID=O.OrderID

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

GROUP BY P.CATEGORY

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

# שתי שאילתות מקוננות תוך שימוש במרכיבים נוספים 10%

# שאילתת Update מקוננת- השאילתה מעדכנת את המחיר הסופי של כל הזמנה בהתאם לתוכן שלה.

ALTER TABLE dbo.ORDERS

ADD TotalPrice MONEY ;

UPDATE dbo.ORDERS SET TotalPrice= ISNULL(TotalPrice,0);

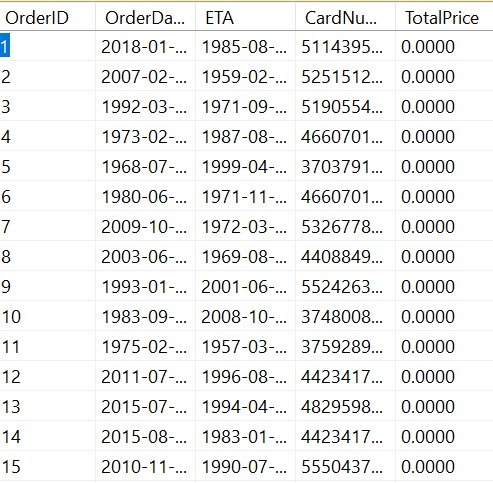
UPDATE ORDERS SET TotalPrice = (SELECT SUM(V.Price\*oc.Quantity)

FROM VERSIONS AS V JOIN ORDERSCONTENT AS OC ON V.ProductID=OC.ProductID

WHERE OC.OrderID=ORDERS.OrderID

)

לפני העדכון:



אחרי העדכון:



**שאילתת חיסור** המציגה את כל הלקוחות שביצעו הזמנות אך לא רכשו מוצרים מותאמים אישית.   
המטרה: הרחבת השיווק של מוצרים מותאמים אישית בפני הלקוחות שלא רכשו אותם. זאת, מפני שאנו לא יודעים אם הם לא רכשו מפני שלא ידעו או כי הם לא היו מעוניינים בכך.

SELECT DISTINCT CustomerID

FROM dbo.ORDERS JOIN dbo.CREDITCARDS ON CREDITCARDS.CardNumber = ORDERS.CardNumber

EXCEPT

SELECT DISTINCT CustomerID

FROM dbo.CUSTOMINORDER JOIN dbo.ORDERS ON ORDERS.OrderID = CUSTOMINORDER.OrderID

JOIN dbo.CREDITCARDS ON CREDITCARDS.CardNumber = ORDERS.CardNumber

**תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**

## מטלה 2 – יישומי כלים מתקדמים

# View

יצרנו View שמציג את כמות ההזמנות לכל מדינה לפי שנה. באמצעותו, הדרג הניהולי יידע לבנות תמונת מצב נוכחית, מגמות עלייה וירידה ובעקבותיהן ליצור תוכניות עסקיות עתידיות.

CREATE VIEW View\_CountriesOrders AS

SELECT DISTINCT Country= C.[Address-Country] ,ProductsOrdered= SUM(oc.Quantity), [Year Of Orders] = YEAR(O.OrderDate)

FROM dbo.CUSTOMERS AS C JOIN dbo.CREDITCARDS AS CC ON CC.CustomerID = C.Email

JOIN dbo.ORDERS AS O ON O.CardNumber = CC.CardNumber JOIN

dbo.ORDERSCONTENT AS OC ON OC.OrderID = O.OrderID

GROUP BY C.[Address-Country], YEAR(O.OrderDate)

בהדגמה ניתן לראות את כמות המוצרים שנמכרה לכל מדינה במהלך שנת 2020.

SELECT \*

FROM View\_CountriesOrders

WHERE ProductsOrdered<200 AND [Year Of Orders]='2020'

ORDER BY ProductsOrdered DESC

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

# פונקציות (Functions)

**פונקציה המחזירה ערך יחיד**. הפונקציה מקבלת את מספר ההזמנה ולבסוף מחזירה את מספר המוצרים השונים שיש בה. כמות המוצרים מעידה על הפקת המיטב מכל מכירה. אילו מספר המוצרים נמוך, יש לפעול לטובת מיקסום המכירה והגדלתה.

-- DROP FUNCTION NumOfProductsInOrders

CREATE FUNCTION NumOfProductsInOrders (@OID INT)

RETURNS INT

AS BEGIN

DECLARE @Sum INT

SELECT @Sum=COUNT(DISTINCT ProductID)

FROM ORDERSCONTENT

WHERE OrderID=@OID

RETURN @Sum

END

SELECT ORDERD=(dbo.NumOfProductsInOrders(1))

FROM ORDERSCONTENT



**פונקציה המחזירה טבלה**. הפונקציה מקבלת את מזהה הלקוח ומחזירה את מזהי ההזמנות שביצע ועלותן. כך, יש ביכולתנו לנתח את הרגלי הצריכה שלו.

--DROP FUNCTION CustomerOrders

CREATE FUNCTION CustomerOrders (@CID VARCHAR (50))

RETURNS @cOrders

TABLE ([orderID] VARCHAR(50), [Total Price] MONEY)

AS

BEGIN

INSERT @cOrders ([orderID], [Total Price])

SELECT O.OrderID, SUM(OC.Quantity\*O.TotalPrice)

FROM ORDERSCONTENT AS OC JOIN ORDERS AS O ON O.OrderID = OC.OrderID JOIN dbo.CREDITCARDS AS CC ON CC.CardNumber = O.CardNumber

JOIN dbo.CUSTOMERS AS CT ON CT.Email = CC.CustomerID

WHERE CC.CustomerID= @CID

GROUP BY O.OrderID

RETURN

END

SELECT \*

FROM dbo.CustomerOrders('MHVJJD@K3RSSC6MSFK.org.il')

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

# Trigger פשוט

ה- Trigger מבצע את עדכון השדה Total Price בטבלת Orders לאחר הוספה/ מחיקה/ עדכון של מוצרים להזמנה.

--עדכון שדה TOTAL PRICE:

CREATE TRIGGER OrderTotalCost

ON orderscontent

FOR INSERT,UPDATE,DELETE AS

UPDATE ORDERS

SET dbo.Orders.TOTALPRICE= (SELECT SUM(OC.Quantity\*V.Price)

FROM dbo.ORDERSCONTENT AS OC JOIN dbo.VERSIONS AS V ON V.ProductID = OC.ProductID

)

WHERE Orders.OrderID IN (

SELECT DISTINCT OrderID FROM INSERTED

UNION

SELECT DISTINCT OrderID FROM deleted

)

--עלות לפני הוספת מוצר

SELECT TotalPrice

FROM dbo.ORDERS

WHERE OrderID=2



--הוספת 2 יחידות להזמנה

UPDATE dbo.ORDERSCONTENT

SET Quantity=( SELECT Quantity

FROM dbo.ORDERSCONTENT

WHERE OrderID=2)+2

WHERE ProductID=1 AND OrderID=2

--בדיקת עלות כוללת

SELECT TotalPrice

FROM dbo.ORDERS

WHERE OrderID=2



# פרוצדורה שמורה ((Stored Procedure פשוטה

הפרוצדורה מקבלת מוצר ואחוז הנחה ומעדכנת את מחירו בהתאם ,השימוש בה ייעשה בימים מיוחדים בהם האתר נותן הנחה על מגוון מוצרים -כגון "Black Friday" ,"Cyber Monday" וכו'.

מטרת הפעולה היא עדכון נתונים ללא מתן גישה לטבלאות-העובד אשר יבצע את הפרוצדורה לא נחשף למרבית הטבלאות ולכן נעדיף להשתמש בפרוצדורה ולא בפונקציה/כתיבת קוד SQL ישיר.

CREATE PROCEDURE Set\_Discount @Discount money , @Pid Int

AS

UPDATE VERSIONS SET Price=Price-(Price\*@Discount)

SELECT \*

FROM dbo.VERSIONS

EXECUTE set\_Discount 0.2,2

**לפני הנחה:**



**אחרי ההנחה:**

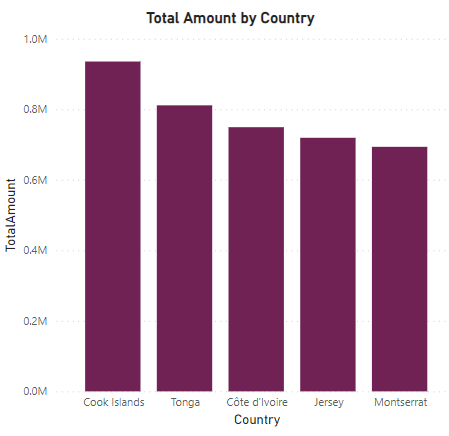


## מטלה 3 (20%) – כלים להצגת נתונים

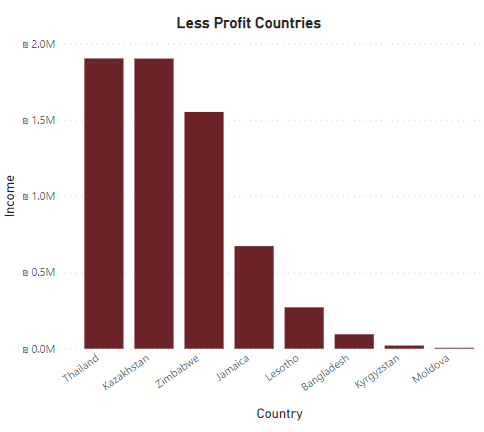
הדו"ח העסקי מיועד **למחלקת הכלכלה** של חברת Nike. מטרתו העסקית היא סיפוק מידע עדכני הרלוונטי לניתוח כלכלי-עסקי של החברה. בנוסף, דו"ח זה יוכל בעתיד לסייע בבניית תוכניות עבודה להמשך.

1. גרפים המייצגים את **מכירות החברה המפולג לפי מדינות**. ככל שהמכירות גדולות יותר, ניתן להסיק היכן יש ביקוש למוצרים של החברה, ובהמשך לכך באילו מקומות יהיה רווחי יותר להשקיע בשיווק רחב יותר ואפילו הקמת שלוחות נוספות של החברה במדינות ייעודיות. לחילופין, ייתכן שיהיה צורך להפסיק שיתופי פעולה עם חברות שליחויות למדינות מסוימות. כחלק מכך, גרף שמייצג את **המדינות בעלות שיעור ההכנסה הנמוך ביותר** בחמש השנים האחרונות. מדינות אלו ימשיכו להיות תחת פיקוח על מנת לבחון את כדאיות השארת המשאבים בהן.

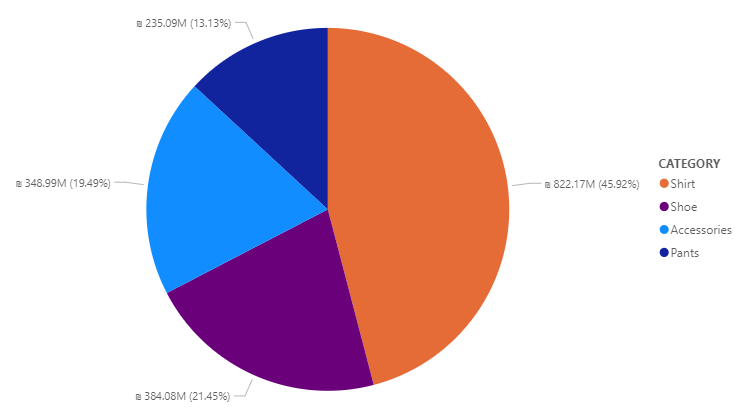
5 המדינות הרווחיות ביותר:



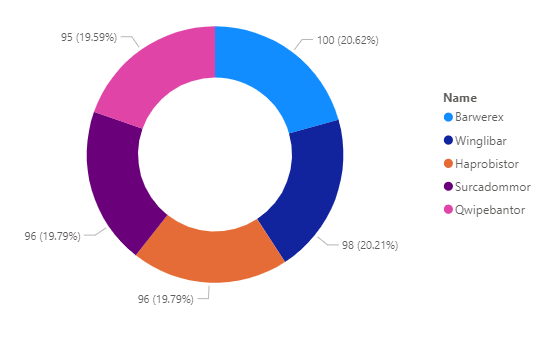
המדינות עם שיעור הרווח הנמוך ביותר:



1. גרף המראה את **התפלגות מכירות החברה לפי קטגוריות**. בהתאם לכך, ניתן להעריך אילו מוצרים כדאי לייצר בכמויות גדולות יותר, לפתח מגוון רחב יותר של קולקציות ולהתאים לכך את השיווק. בנוסף, ייתכן שישתלם לפתוח חנויות ייעודיות לקטגוריות הנ"ל.



1. גרף המייצג את **חמשת הדגמים הנפוצים ביותר שעליהם התבסס העיצוב האישי** של הנעל. בהתאם לנתונים אלה, יהיה ניתן לצפות אילו דגמים כדאי לייצר יותר. בנוסף, אפשר לשווק דגמים אלה בצורה רחבה יותר כדי להגיע ללקוחות נוספים.



## מטלה 4 (צבירה - עד 35%) – יישומים מתקדמים

## כלי מורכב (SP או Trigger) העושה שימוש בסמן (Cursor) 10%

--פרוצדורה עם לולאה שנותנת ללקוח קופון במידה וקנה החל מסכום מסויים שהצטבר במהלך כל ההזמנות שביצע.  
הפרוצדורה נועדה עבור שימור לקוחות קיימים.

--ALTER TABLE dbo.ORDERS DROP COLUMN Cupon

ALTER TABLE dbo.ORDERS ADD useCupon DECIMAL(10,2)

--ALTER TABLE dbo.CUSTOMERS DROP COLUMN Cupon

ALTER TABLE dbo.CUSTOMERS ADD Cupon DECIMAL(10,2)

--DROP PROCEDURE SP\_Cupon

CREATE PROCEDURE SP\_Cupon (@cupon5 int ,@cupon10 int ,@cupon20 int)

AS

DECLARE @cEmail varchar(50);

DECLARE @Total\_Price int;

BEGIN

DECLARE c1 CURSOR LOCAL FOR

SELECT CC.CustomerID,TotalOrders=SUM(o.TotalPrice)

FROM dbo.ORDERS AS o JOIN dbo.CREDITCARDS AS CC ON CC.CardNumber = o.CardNumber

GROUP BY CC.CustomerID

OPEN c1

FETCH NEXT FROM c1 INTO @cEmail,@Total\_Price

WHILE (@@FETCH\_STATUS=0)

BEGIN

UPDATE dbo.CUSTOMERS SET Cupon=(

CASE WHEN @Total\_Price BETWEEN @cupon5 AND @cupon10 THEN 0.05

WHEN @Total\_Price BETWEEN @cupon10 AND @cupon20 THEN 0.1

WHEN @Total\_Price>@cupon20 THEN 0.2 END)

WHERE @cEmail=Email

FETCH NEXT FROM c1 INTO @cEmail,@Total\_Price

END

CLOSE c1

DEALLOCATE c1

END

go

EXECUTE SP\_Cupon 20000,50000,100000

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

## שילוב מערכתי של מספר כלים 10%

בהמשך לסעיף הקודם, יצרנו פעולה למימוש הקופון.

לשם כך, בנינו טריגר, שתי פרוצדרות ופונקציה שמטרתן לעדכן את הקופון בהזמנה, לאכוף את הקופון בהזמנה ולאחר מכן למחוק את הזכאות לקופון אצל הלקוח.

* **הטריגר**: כאשר נכנסת הזמנה חדשה, ישנו שילוב מערכתי של מספר כלים- בדיקת נכונות הקופון, עדכון הקופון בהזמנה ואצל הלקוח, ולבסוף עדכון המחיר הסופי.
* **הפרוצדורה הראשונה**: מקבלת את מספר ההזמנה שנכנסה ואת הקופון שהוזן בה. הפרוצדורה שולחת לפונקציה את הנתונים. במידה והקופון אכן קיים אצל הלקוח, הוא מתעדכן בהזמנה ונמחק אצל הלקוח. אם הוא לא תואם, הקופון לא נכנס להזמנה.
* **הפרוצדורה השנייה**: מקבלת את מספר ההזמנה שנכנסה ואת הקופון שהוזן בה.  
  באמצעותה מתעדכן המחיר הכולל של ההזמנה בהתאם לקופון.
* **הפונקציה**: מקבלת את מספר ההזמנה שנכנסה.  
  בודקת האם הקופון שהוזן בהזמנה קיים אצל הלקוח.

הקופון של הלקוח לפני הכנסת הקופון להזמנה:



--DROP TRIGGER UPDATE\_Cupon

CREATE TRIGGER Update\_Cupon

ON ORDERS

FOR INSERT AS

DECLARE @CUPON DECIMAL(10,2)

DECLARE @ORDER\_ID INT

SET @ORDER\_ID=(SELECT inserted.orderID FROM INSERTED)

SET @cupon=(SELECT inserted.useCupon FROM inserted)

EXECUTE SP\_CHECK\_CUPON @ORDER\_ID,@CUPON

EXECUTE sp\_update\_totalPrice @ORDER\_ID,@CUPON

--DROP PROCEDURE SP\_CHECK\_CUPON

CREATE PROCEDURE SP\_CHECK\_CUPON @ORDER\_ID int ,@CUPON decimal(10,2)

AS

IF (dbo.Check\_IF\_HAVECUPON(@ORDER\_ID)=@CUPON)

UPDATE dbo.CUSTOMERS SET CUPON=0

WHERE Email =(SELECT cc.customerID FROM dbo.ORDERS AS o JOIN dbo.CREDITCARDS

AS cc ON cc.CardNumber = o.CardNumber WHERE o.OrderID=@ORDER\_ID)

ELSE

UPDATE dbo.ORDERS SET useCupon=0

WHERE OrderID=@ORDER\_ID

--drop PROCEDURE sp\_update\_totalPrice

CREATE PROCEDURE sp\_update\_totalPrice @ORDER\_ID int ,@cupon decimal(10,2)

AS

UPDATE dbo.ORDERS SET TotalPrice=TotalPrice\*(1-@CUPON)

WHERE OrderID=@ORDER\_ID

CREATE FUNCTION Check\_IF\_HAVECUPON(@OID INT)

RETURNS decimal(10,2)

AS BEGIN

DECLARE @output DECIMAL(10,2)

(SELECT @output=c.cupon

FROM dbo.ORDERS AS o JOIN dbo.CREDITCARDS AS cc ON cc.CardNumber = o.CardNumber JOIN dbo.CUSTOMERS

AS c ON c.Email = cc.CustomerID

WHERE o.OrderID=@OID)

RETURN @output

END

INSERT INTO dbo.ORDERS

(

OrderID,

OrderDate,

ETA,

TotalPrice,

CardNumber,

usecupon

)

VALUES

( 501, -- OrderID - int

GETDATE(), -- OrderDate - date

GETDATE(), -- ETA - date

200000, -- TotalPrice - money

'5190554204475847', -- CardNumber - varchar(20)

0.2

)

--DELETE FROM dbo.ORDERS WHERE OrderID=501

SELECT \*

FROM dbo.ORDERS

WHERE OrderID=501

הצגה בהזמנה:

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

SELECT \*

FROM dbo.CUSTOMERS

WHERE Email='25MNA8W@QB70YL.com'

מחיקת הקופון לאחר מימוש בהזמנה:



## דו"ח המושתת על שאילתה מקוננת מורכבת 10%

בדו"ח זה, בחרנו להתמקד בנתונים בעלי משמעות עסקית על בסיס פעולות הלקוח.

הדו"ח מציג עבור כל לקוח:

* תאריך הזמנה אחרון.
* קטגוריה מועדפת על בסיס הפריטים שרכש.
* מחיר ממוצע בהתאם למשלוחים שביצע.
* מספר חיפושים שביצע טרם הרכישות.
* מספר ההזמנות שביצע בעיצוב אישי.
* מידה נפוצה.

לצורך כך ולטובת מודולריות, נעזרנו בטבלת View ובפונקציית אגריגציה.

--DROP VIEW VIEW\_ORDER\_TABLES

CREATE VIEW View\_Order\_Tables AS

SELECT O.OrderID,CC.CustomerID,O.OrderDate,OC.ProductID,OC.Quantity,O.TotalPrice,CC.CardNumber

FROM dbo.ORDERS AS O JOIN dbo.ORDERSCONTENT AS OC ON OC.OrderID = O.OrderID

JOIN dbo.CREDITCARDS AS CC ON CC.CardNumber = O.CardNumber

SELECT Email,

[Last Time Ordered]=(SELECT MAX(OrderDate)

FROM VIEW\_ORDER\_TABLES

WHERE CustomerID=customers.Email

GROUP BY CustomerID),

[Favorite Category]=(SELECT T.Category

FROM (

SELECT TOP 1 CustomerID,Category, QuantityOrdered=SUM(Quantity)

FROM dbo.PRODUCTS AS p JOIN VIEW\_ORDER\_TABLES AS V ON P.ProductID=V.productID

WHERE CustomerID=dbo.CUSTOMERS.Email

GROUP BY CustomerID,p.CATEGORY

ORDER BY QuantityOrdered desc

) AS T

)

,

[Average Price of Orders]=CAST(ROUND((SELECT tot=(SUM(Quantity\*v.Price)/COUNT(DISTINCT OrderID))

FROM VIEW\_ORDER\_TABLES AS VI JOIN dbo.VERSIONS AS v ON v.ProductID = vi.ProductID

WHERE CustomerID=dbo.CUSTOMERS.Email

GROUP BY CustomerID

),2)AS decimal(10,2)) ,

[Num Of Searches]=(SELECT numOfSearches=COUNT(\*)

FROM dbo.INTEREST

WHERE Email=dbo.CUSTOMERS.Email

GROUP BY Email

),

[num of Custom Orders]=ISNULL((

SELECT COUNT(\*)

FROM dbo.CUSTOMINORDER AS CIO JOIN View\_ORDER\_Tables AS V ON cio.OrderID=v.orderID

WHERE CustomerID=dbo.CUSTOMERS.Email

GROUP BY CustomerID),0),

[Common Size]=(SELECT CIO.Size

FROM(SELECT TOP 1 CustomerID ,V.Size,[count sizes]=COUNT(V.Size)

FROM VIEW\_ORDER\_TABLES AS VI

JOIN dbo.VERSIONS AS V ON V.ProductID = VI.ProductID

WHERE CustomerID=dbo.CUSTOMERS.Email

GROUP BY CustomerID,V.Size

ORDER BY [count sizes] desc

) AS CIO

)

FROM dbo.CUSTOMERS

WHERE (SELECT MAX(o.OrderDate)

FROM dbo.ORDERS AS o JOIN dbo.CREDITCARDS AS cc ON cc.CardNumber = o.CardNumber

WHERE cc.CustomerID=customers.Email) IS NOT NULL

ORDER BY [Average Price of Orders] DESC

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי