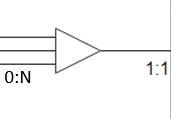
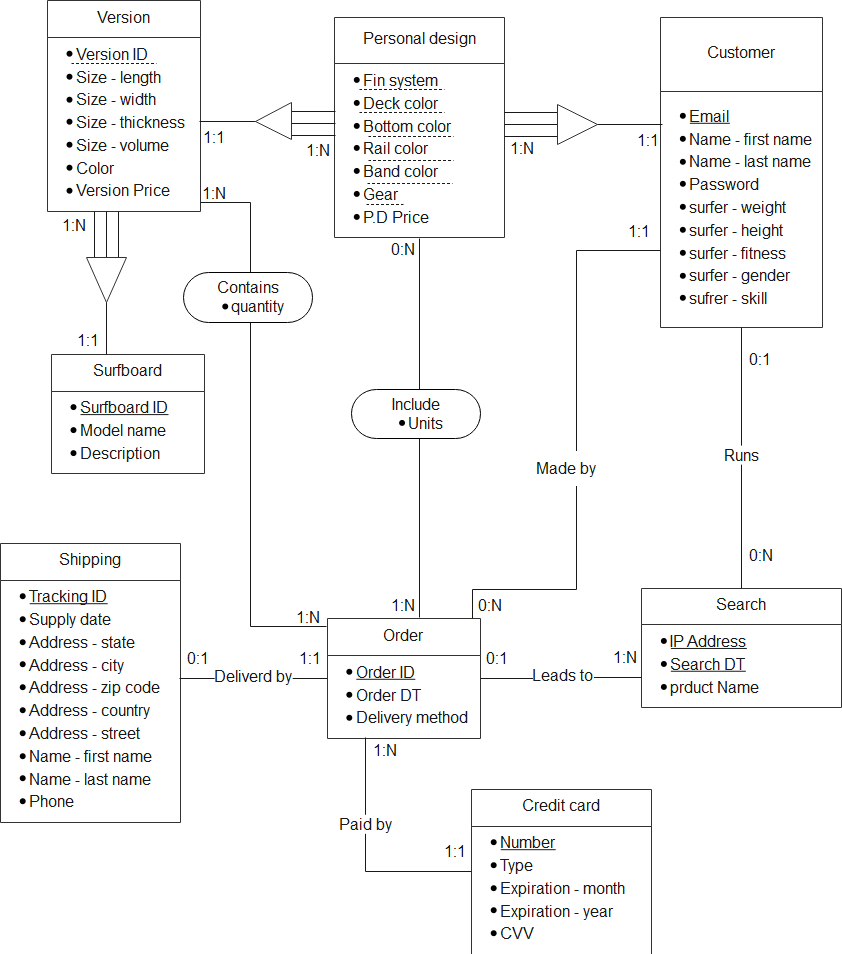
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מס' קבוצה | אתר | | | תאריך הגשה |
| 17 | <https://rustysurfboards.com/pages/custom> | | | 10/01/2022 |
| חברי הצוות - מספרי ת.ז | | | | |
|  | |  |  | |

## פרויקט בסיסי נתונים – חלק ג'

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי**מבנה הטבלאות כפי שהוגשו בחלק ב' :**

**תרשים ERD מתוקן :**



0:N

0:N

1:1

0:1

Belongs to

0:N



**טבלאות מעודכנות :**

**COSTUMERS** (Email, Name - first name, Name - last name, Password, surfer – weight, surfer– height, surfer – fitness, surfer – gender, surfer – skill)

**CREDITCARDS** (Number, Type, Expiration – month, Expiration – year, CVV , Email (COSTUMERS))

**ORDERS** (Order ID, Order DT, Delivery method , Credit Card (CREDITCARDS))

**SHIPPINGS** (Order ID (ORDERS), Tracking ID, Supply date, Address – state, Address – city, Address - zip code, Address – country, Address – street, Name - first name, Name - last name, Phone)

**SEARCHES** (IP Address, Search DT, product name, Email (CUSTOMERS), Order ID (ORDERS))

**SURFBOARDS** (Surfboard ID, Model name, Description)

**VERSIONS** (Surfboard ID (SURFBOARDS) , Version ID , Size – length, Size – width, Size – thickness, Size - volume, Color, Version Price)

**PERSONAL DESIGNS** ({Surfboard ID, Version ID} (VERSIONS), Email (CUSTOMERS) , Fin system , Deck color , Bottom color , Rail color , Band color , Gear , P.D price )

**CONTAINS** ({Surfboard ID , Version ID} (VERSIONS) , Order ID (ORDERS), quantity)

**INCLUDE** ( {Surfboard ID, Version ID, Email , Fin system , Deck color , Bottom color , Rail color , Band color , Gear } (PERSONAL DESIGNS) , Order ID (ORDERS) , Units)

**התייחסות לשינויים :**

* שינינו את הקרדינליות במקומות הנדרשים על פי ההערות.
* בחרנו לבטל את הקשר בין הזמנה ללקוח ובמקומו הוספנו קשר בין האשראי ללקוח. אנחנו מניחים שכדי לבצע הזמנה הלקוח מוכרח להזין פרטי אשראי וניתן לשלם עבור כל הזמנה בכרטיס אשראי אחד ויחיד המקושר ללקוח אחד ויחיד.
* בחרנו להפוך את הישות SHIPPING לישות חלשה של ORDER כיוון שנרצה שמשלוח יזוהה על פי ההזמנה אליה מקושר לטובת זיהוי ההזמנה על ידי הלקוח ועל ידי האתר ועל פי מספר משלוח שישרת את חברת המשלוחים ויאפשר מעקב אחר מיקום המשלוח. נזכיר כי לא כל הזמנה כרוכה במשלוח לכן הקרדינליות היא 0:N בצד של המשלוח.

## מטלה 1 (25%) – שאילתות

# שתי שאילתות SELECT ללא קינון 5%

## שאילתה 1 :

השאילתה מציגה את הלקוחות שביצעו יותר מהזמנה אחת במהלך השנה האחרונה.

עבור כל לקוח נציג את פרטי הלקוח (מזהה , ושם מלא) ואת מספר ההזמנות שביצע.

הפלט יסודר בסדר יורד של כמות ההזמנות.

השאילתה מיועדת לאיתור וסקירה של לקוחות חוזרים באתר.

המידע יכול לשמש את החברה למתן קופונים/שליחת פרסומות ועידוד רכישות עתידיות.

**נוסח השאילתה ב – SQL :**

select c.Email ,

full\_name= c.[name-first name] + ' ' + c.[name-last name],

NumOfOrders = count(\*)

from ORDERS as o join CREDITCARDS as cc

on o.[Credit Card] = cc.Number

join CUSTOMERS as c

on c.Email = cc.EMAIL

WHERE DATEDIFF(YY , O.[Order DT] , GETDATE()) <=1

group by c.Email,

c.[name-first name],

c.[name-last name]

HAVING COUNT(\*) > 1

order by NumOfOrders desc

**פלט השאילתה כולל 38 רשומות :**

(נציג את 10 הרשומות הראשונות)

תמונה שמכילה טקסט, שולחן

התיאור נוצר באופן אוטומטי

**שאילתה 2 :**

השאילתה מציגה עבור כל הזמנה שכרוכה בביצוע משלוח ומועד האספקה שלה הוא תאריך בחודש האחרון את מזהה הלקוח המזמין, מספר ההזמנה, שיטת המשלוח ותאריך האספקה.

הפלט ימוין בסדר עולה של מספרי ההזמנות.

המטרה העסקית של השאילתה היא בקרה על ביצוע המשלוחים על ידי חברת השליחויות ווידוא שביעות רצון אל מול לקוחות שקיבלו לאחרונה את הזמנתם.

**נוסח השאילתה ב - SQL :**

SELECT O.[Order ID],

cc.Email,

O.[Delivery Method],

S.[Supply date]

FROM ORDERS as o join CREDITCARDS as cc

on o.[Credit Card] = cc.Number

JOIN SHIPPING AS S

ON O.[Order ID] = S.[Order ID]

WHERE O.[Delivery Method] IN ('regular delivery' , 'express dalivery') AND

DATEDIFF (mm , S.[Supply date] , GETDATE() ) <= 1

ORDER BY O.[Order ID]

**פלט השאילתה כולל 11 רשומות :**

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

# שתי שאילתות SELECT מקוננות10%

**שאילתה 1 :**

השאילתה מציגה את החלק היחסי של הזמנות שנאספו באיסוף עצמי מתוך כלל ההזמנות בשנה האחרונה.

השאלה העסקית שהשאילתה בוחנת היא נחיצות חברת המשלוחים בשנה שחלפה.

**נוסח השאילתה ב - SQL :**

SELECT AllOrders = count(\*),

SelfPickUpOrders = (SELECT COUNT(\*)

FROM ORDERS as O

WHERE O.[Delivery Method] = 'self-pick up' AND

DATEDIFF(yy,O.[Order DT],GETDATE())<= 1),

proportion = ROUND(CAST(CAST((SELECT COUNT(\*)

FROM ORDERS as O

WHERE O.[Delivery Method]='self-pick up'

AND

DATEDIFF(yy,O.[Order DT],

GETDATE())<= 1)

AS REAL)/CAST(COUNT(\*) AS REAL) AS REAL),3)

FROM ORDERS as O

WHERE DATEDIFF(yy ,O.[Order DT],GETDATE()) <= 1

**פלט השאילתה כולל רשומה 1 המכילה את הערכים הנדרשים :**



**שאילתה 2 :**

השאילתה מאתרת את חמשת הלקוחות הרווחיים ביותר לחברה.

כלומר מציגה את חמשת הלקוחות שסכום הזמנתם הכולל הוא הגבוה ביותר.

חישוב סכום ההזמנה הכולל יתבצע על ידי סכימה של מחיר כלל הפריטים המצויים בכל בהזמנה.

עבור כל לקוח נציג את מזהה הלקוח, שם הלקוח, מספר ההזמנה הרלוונטי ומחיר ההזמנה הכולל.

מבחינה עסקית , החברה תרצה לאתר לקוחות מובילים לשם בדיקות שביעות רצון , שימור לקוחות רווחיים, מתן תמריצים ושיווק מוכוון אוכלוסייה.

**נוסח השאילתה ב - SQL :**

SELECT TOP 5 CU.Email , FullName = CU.[name-first name] + ' ' + CU.[name-last name],

O.[Order ID],

Total\_Revenue = ISNULL( TOTALS\_V.VersionTotalCost ,0) + ISNULL (TOTALS\_P.PersonalTotalCost,0)

FROM dbo.CUSTOMERS AS CU JOIN CREDITCARDS AS CC ON CU.Email = CC.EMAIL

JOIN ORDERS AS O ON O.[Credit Card] = CC.Number

JOIN (SELECT C.[Order ID] ,

VersionTotalCost=SUM(C.quantity \* V.[Version Price])

FROM VERSIONS AS V JOIN CONTAIN AS C ON V.[Surfboard ID] = C.[Surfboard ID] AND V.[Version ID]=C.[Version ID]

GROUP BY C.[Order ID]) AS TOTALS\_V ON O.[Order ID]=TOTALS\_V.[Order ID]

JOIN (SELECT I.[Order ID], PersonalTotalCost = SUM(I.Units \* P.[PD Price])

FROM PERSONALDESIGNS AS P JOIN [INCLUDE] AS I

ON P.[Surfboard ID]=I.[Surfboard ID] AND

P.[Version ID]=I.[Version ID] AND P.Email=I.Email AND

P.[Fin system]=I.[Fin system] AND P.[Deck color]=I.[Deck color] AND P.[Bottom color]=I.[Bottom color] AND P.[Rail color]=I.[Rail color] AND P.[Band color] = I.[Band color] AND P.Gear = I.Gear

GROUP BY I.[Order ID]) AS TOTALS\_P

ON O.[Order ID] = TOTALS\_P.[Order ID]

ORDER BY Total\_Revenue DESC

# פלט השאילתה מכיל 5 רשומות כמצופה :

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

# שתי שאילתות מקוננות תוך שימוש במרכיבים נוספים 10%

**שאילתת עדכון - UPDATE**

החברה תרצה לעדכן עבור כל הזמנה את הסכום הכולל ששולם עבורה.

העלות הכוללת של ההזמנה מייצגת את הרווח עבור החברה מההזמנה ואת הסכום לתשלום עבור הלקוח , לכן זהו פריט מידע חיוני ועליו להופיע יחד עם פרטי ההזמנה.

**נוסח השאילתה ב - SQL :**

(תחילה נוסיף עמודה בטבלת ההזמנות שתשמור את הסכום הכולל עבור כל הזמנה.)

ALTER TABLE dbo.ORDERS ADD TotalCost MONEY

UPDATE dbo.ORDERS

SET TotalCost = (

SELECT Total\_Revenue = ISNULL( TOTALS\_V.VersionTotalCost ,0) +

ISNULL (TOTALS\_P.PersonalTotalCost,0)

FROM dbo.CUSTOMERS AS CU JOIN CREDITCARDS AS CC

ON CU.Email = CC.EMAIL JOIN ORDERS AS O

ON O.[Credit Card] = CC.Number

JOIN (SELECT C.[Order ID] ,

VersionTotalCost =

SUM(C.quantity \* V.[Version Price])

FROM VERSIONS AS V JOIN CONTAIN AS C ON

V.[Surfboard ID] = C.[Surfboard ID]

AND V.[Version ID] = C.[Version ID]

GROUP BY C.[Order ID])

AS TOTALS\_V ON O.[Order ID] = TOTALS\_V.[Order ID]

JOIN (SELECT I.[Order ID],

PersonalTotalCost = SUM(I.Units \* P.[PD Price])

FROM PERSONALDESIGNS AS P JOIN [INCLUDE] AS I

ON P.[Surfboard ID]=I.[Surfboard ID]

AND P.[Version ID]=I.[Version ID]

AND P.Email=I.Email

AND P.[Fin system]=I.[Fin system]

AND P.[Deck color]=I.[Deck color]

AND P.[Bottom color]=I.[Bottom color]

AND P.[Rail color]=I.[Rail color]

AND P.[Band color] = I.[Band color]

AND P.Gear = I.Gear

GROUP BY I.[Order ID])

AS TOTALS\_P ON O.[Order ID] = TOTALS\_P.[Order ID]

WHERE orders.[Order ID] = O.[Order ID]

)

**הצגת הטבלה לאחר העדכון :**

SELECT \*

FROM dbo.ORDERS

**הפלט יציג את טבלת ORDERS לאחר העדכון :**

(הטבלה המעודכנת מכילה 500 רשומות , כפי שצפינו השאילתה עדכנה את המחיר הכולל עבור כל אחת מההזמנות בבסיס הנתונים , נציג את 10 הראשונות)

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

**שאילתה המשלבת הפרש - EXEPT :**

נרצה לאתר את פרטי הלקוחות (אימייל ושם מלא) שביצעו מספר הזמנות שקטן ממספר ההזמנות הממוצע ללקוח. החברה תרצה לאתר לקוחות אלה כדי לשלוח להם פרסומות / קופונים / הודעות על מבצעים או הנחות על מנת לעודד אותם לרכוש שוב באתר.

**נוסח השאילתה ב – SQL :**

SELECT Email , full\_name = [name-first name]+ ' ' + [name-last name]

FROM dbo.CUSTOMERS

EXCEPT

SELECT C.Email ,

full\_name= C.[name-first name] + ' ' + C.[name-last name]

FROM ORDERS AS O JOIN CREDITCARDS AS CC

ON O.[Credit Card] = CC.Number

JOIN CUSTOMERS AS C

ON C.Email = CC.EMAIL

GROUP BY c.Email,

c.[name-first name],

c.[name-last name]

HAVING COUNT(\*) > (SELECT AvgNumOfOrders = AVG(NumOfOrders)

FROM (SELECT C.Email ,

full\_name= C.[name-first name] + ' ' + C.[name-last name],

NumOfOrders = COUNT(\*)

FROM ORDERS AS O JOIN CREDITCARDS AS CC

ON O.[Credit Card] = CC.Number

JOIN CUSTOMERS AS C ON C.Email = CC.EMAIL

GROUP BY c.Email,

c.[name-first name],

c.[name-last name]) AS A)

**פלט השאילתה מכיל 318 רשומות :**

(נציג את עשרת הרשומות הראשונות)

תמונה שמכילה טקסט, שולחן

התיאור נוצר באופן אוטומטי

## מטלה 2 (25%) – יישומי כלים מתקדמים

# View 5%

**VIEW מעקב מלאי –**

הטבלה שה – VIEW מציג תשמש כדו"ח מעקב אחר המלאי במחסני החברה , הדו"ח מציג את ההזמנות שהתקבלו בחודש האחרון לצד הפריטים (מק"ט וכמות) שיש להוציא מהמלאי כדי לספק את ההזמנות.

הדו"ח מיועד לשימוש עובדי המחסן ומנהלי המלאי של החברה ומכיל מידע ייעודי לתפקידם תוך מידור מידע רגיש כמו מספר כרטיס האשראי של הלקוח המצורף לפרטי ההזמנה.

**נוסח יצירת ה- VIEW ב – SQL :**

CREATE VIEW StockKeeper\_VIEW AS

SELECT O.[Order ID] , O.[Order DT] ,

CatalogNumber = CAST(C.[Surfboard ID] AS VARCHAR) + '.' + CAST(c.[Version ID] AS VARCHAR),

C.quantity

FROM dbo.ORDERS AS O JOIN dbo.CONTAIN AS C ON O.[Order ID] = C.[Order ID]

WHERE DATEDIFF (mm , O.[Order DT] , GETDATE() ) <= 1

SELECT \*

FROM StockKeeper\_VIEW

**הפלט משאילתת ה – SELECT של ה – VIEW :**

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

# פונקציות (Functions) 10%

**פונקציה שמחזירה טבלה –**

הפונקציה מקבלת מספר הזמנה ומחזירה את פרטי ההזמנה (מספר הזמנה , תאריך ההזמנה , ושיטת המשלוח) לצד פרטי הלקוח שביצע אותה (אימייל ושם מלא).

הפונקציה תאפשר לעובדי החברה לאתר בקלות פרטי הזמנה ופרטי לקוחות באמצעות מספרי הזמנה ותוכל לשמש עובדים שלא צריכים להיות חשופים לפרטים רגישים כמו פרטי האשראי של הלקוח אך כן זקוקים למידע אודות לקוחות והזמנות.

**יצירת הפונקציה ב – SQL :**

--DROP FUNCTION Orders\_Customers\_table

CREATE FUNCTION Orders\_Customers\_table (@OrderID INT)

RETURNS TABLE

AS RETURN

SELECT O.[Order ID] , O.[Order DT] , O.[Delivery Method],

C.Email ,

FullName = C.[name-first name] + ' ' + C.[name-last name]

FROM dbo.ORDERS AS O JOIN dbo.CREDITCARDS AS CC

ON CC.Number = O.[Credit Card]

JOIN dbo.CUSTOMERS AS C ON C.Email = CC.EMAIL

WHERE O.[Order ID] = @OrderID

**נדגים הפעלת הפונקציה :**

SELECT \*

FROM dbo.Orders\_Customers\_table(1)

**פלט השאילתה :**



**פונקציה שמחזירה סקלר -**

הפונקציה מקבלת תאריך וסוג גלשן ומחזירה את מספר הגלשנים מדגם זה שנמכרו בטווח התאריכים הנבחר (מהתאריך הנבחר ועד היום).

הפונקציה מאפשרת לבחון את הפופולריות של דגם מסוים לאורך תקופות שונות.

**יצירת הפונקציה ב – SQL :**

CREATE FUNCTION UnitsSold (@SurfboardID INT , @fromDate DATE)

RETURNS INT

AS BEGIN

DECLARE @Amount INT

SELECT @Amount = SUM(C.quantity)

FROM dbo.CONTAIN AS C JOIN dbo.ORDERS AS O

ON O.[Order ID] = C.[Order ID]

WHERE @SurfboardID = C.[Surfboard ID] AND

O.[Order DT] BETWEEN @fromDate AND GETDATE()

GROUP BY C.[Surfboard ID]

RETURN ISNULL(@Amount,0)

END

**נדגים הפעלת הפונקציה :**

SELECT UnitsSold = dbo.UnitsSold (47 , '2021-10-27')

**הפלט מהפעלת הפונקציה :**



# Trigger פשוט 5%

מטרת הטריגר היא לבצע עדכון של סכום ההזמנה הכולל בכל הוספת/ הסרת פריט מההזמנה.

עבור כל פריט שמתווסף להזמנה הטריגר יעדן בטבלת ההזמנות את הסכום הכולל העדכני של ההזמנה הרלוונטית. נזכיר כי הוספנו עמודה בטבלת ההזמנות ששומרת את הסכום הכולל עבור כל הזמנה , זהו שדה מחושב ונרצה שהטריגר יעדכן באופן אוטומטי שדה זה בכל הוספת/הסרת פריט. (הטריגר המוצג בסעיף זה רלוונטי להוספת פריטים שהם גלשנים מוכנים אך בקלות נוכל להוסיף טריגר נוסף שיהיה שייך לטבלה המתאימה (INCLUDE) ויבצע את אותו העדכון עבור הוספת גלשנים מעוצבים אישית). עלות ההזמנה הכוללת רלוונטית הן עבור החברה והן עבור הלקוח. עבור הלקוח עלות ההזמנה הכוללת מציגה את המחיר הכולל לתשלום ועבור החברה הנתון יציג את רווחי החברה מכל הזמנה באופן שתוכל לסכום בקלות את רווחיה בתקופה מסוימת, להסיק על ממוצע רווח להזמנה ולזהות הזמנות גדולות המעידות על לקוחות רווחיים שיש לשמר.

**נוסח יצירת הטריגר ב – SQL :**

CREATE TRIGGER UPDATE\_TotalCost

ON dbo.CONTAIN

FOR INSERT , DELETE , UPDATE

AS

UPDATE dbo.ORDERS

SET TotalCost = (

SELECT Total\_Revenue =

ISNULL( TOTALS\_V.VersionTotalCost ,0) +

ISNULL (TOTALS\_P.PersonalTotalCost,0)

FROM dbo.CUSTOMERS AS CU JOIN CREDITCARDS AS CC

ON CU.Email = CC.EMAIL JOIN ORDERS AS O

ON O.[Credit Card] = CC.Number

JOIN (SELECT C.[Order ID] ,

VersionTotalCost = SUM(C.quantity \*V.[Version Price])

FROM VERSIONS AS V JOIN CONTAIN AS C

ON V.[Surfboard ID]=

C.[Surfboard ID]

AND V.[Version ID]=C.[Version ID]

GROUP BY C.[Order ID])

AS TOTALS\_V ON O.[Order ID] = TOTALS\_V.[Order ID]

JOIN (SELECT I.[Order ID],

PersonalTotalCost = SUM(I.Units

\*P.[PD Price])

FROM PERSONALDESIGNS AS P

JOIN [INCLUDE] AS I

ON P.[Surfboard ID]=

I.[Surfboard ID] AND

P.[Version ID]=I.[Version ID]

AND P.Email=I.Email

AND P.[Fin system]=I.[Fin system]

AND P.[Deck color]=I.[Deck color]

AND P.[Bottom color]=

I.[Bottom color]

AND P.[Rail color]=I.[Rail color]

AND P.[Band color]=I.[Band color]

AND P.Gear = I.Gear

GROUP BY I.[Order ID])

AS TOTALS\_P ON O.[Order ID] = TOTALS\_P.[Order ID]

WHERE orders.[Order ID] = O.[Order ID]

)

WHERE dbo.ORDERS.[Order ID] IN (SELECT DISTINCT [Order ID]

FROM INSERTED

UNION

SELECT DISTINCT [Order ID]

FROM Deleted)

**נדגים את פעולת הטריגר :**

נוסיף רשומה לטבלת – CONTAIN , כלומר נוסיף להזמנה מסויימת גלשן מהמלאי שמציעה החברה ונצפה שסכום ההזמנה הכולל יתעדכן בטבלת ההזמנות.

INSERT INTO dbo.CONTAIN

(

[Surfboard ID],

[Version ID],

[Order ID],

quantity

)

VALUES

( 1, -- Surfboard ID - int

121, -- Version ID - int

2, -- Order ID - int

2 -- quantity - int

)

**טבלת ההזמנות לפני הוספת הפריט :**

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

**טבלת ההזמנות לאחר הוספת הפריט :**

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

כמצופה מחיר ההזמנה הכולל התעדכן בהתאם להוספת הפריטים.

# פרוצדורה שמורה ((Stored Procedure פשוטה 5%

הפרוצדורה מקבלת מזהה לקוח של לקוח כלשהו ממאגר הלקוחות ושומרת עבורו את מספר ההזמנות שביצע באתר עד כה.

מספר ההזמנות שביצע כל לקוח הוא מידע חיוני עבור החברה בהיבטים שיווקיים ובהיבטי שימור לקוחות - הפרוצדורה נועדה לשם תחזוקה של נתון זה. שמירת הנתון הנ"ל כחלק מטבלת הלקוחות תאפשר את קבלת המידע על מספר ההזמנות לכל לקוח מבלי שיש צורך בגישה לנתוני ההזמנות והאשראי של הלקוחות.

באופן כזה מחלקות שונות (כדוגמת קשרי לקוחות/ מחלקת שיווק) שזקוקות למידע אודות הלקוחות בלבד יוכלו להשתמש במספר ההזמנות שביצע כל לקוח מבלי להיות חשופים למידע אודות פרטי ההזמנות והאשראי של הלקוח.

**נוסח יצירת הפרוצדורה ב- SQL :**

(כפעולה מקדימה ליצירת הפרוצדורה נוסיף עמודה המייצגת את מספר ההזמנות שביצע לקוח בטבלת הלקוחות.)

ALTER TABLE dbo.CUSTOMERS ADD Num\_Of\_Orders int

CREATE PROCEDURE SP\_UpdateNumOfOrders @Email VARCHAR(30)

AS

UPDATE dbo.CUSTOMERS

SET Num\_Of\_Orders =

(SELECT COUNT(\*)

FROM ORDERS AS O JOIN dbo.CREDITCARDS AS CC

ON O.[Credit Card] = CC.Number

JOIN dbo.CUSTOMERS AS C ON C.Email = CC.EMAIL

WHERE C.Email = @Email

GROUP BY C.Email)

WHERE dbo.CUSTOMERS.Email = @Email

**הפעלת הפרוצדורה :**

EXECUTE dbo.SP\_UpdateNumOfOrders @Email = '00EGIA6Y@8JTG2X.WHECCWWGR'

**הרשומה בטבלת לקוחות לפני ההפעלה** **:**

**הרשומה בטבלת לקוחות אחרי ההפעלה:** 

## מטלה 3 (20%) – כלים להצגת נתונים

**דו"ח עסקי**

נייצר דו"ח שמיועד לשימוש מחלקת המכירות והשיווק של החברה.

מטרת הדו"ח להציג נתוני מכירות רלוונטיים למחלקה במהלך תקופה מסוימת.

הדו"ח יספק מידע שיאפשר לעובדי המחלקה לבצע החלטות שיווקיות שיקדמו את האינטרסים של החברה ויגדילו את רווחיה.

הדו"ח יציג נתונים סיכומיים שיאפשרו לעובדי המחלקה להתוות מדיניות שיווק מיטבית ולבנות אסטרטגיות לתקופות הבאות על סמך נתונים מהשטח.

**תוכן הדו"ח :**

* מספר הלקוחות הכולל בחברה – המחלקה תוכל להציב יעדים להרחבת מספר הלקוחות , לבחון את הגידול במספר הלקוחות לאורך תקופות שונות ובמדינות שונות בעזרת הפילטרים.
* מספר ההזמנות הכולל שבוצע באתר – נתון זה משקף את כמות המכירות של החברה, ניתן לבחון את השינוי במספר ההזמנות לאורך התקופות השונות ובמדינות השונות באמצעות הפילטרים. מחלקת השיווק תוכל לבחון האם תקופות מסוימות פופולריות יותר , האם קמפיין שהתקיים בתקופה מסוימת היה מוצלח ולקבל אינדיקציה מתי והיכן יש לשווק את המוצרים.
* טבלה מסכמת המציגה עבור כל גרסת מוצר המוצעת באתר את מספר הפעמים שהוזמן , את הכמות הנמכרת בכל הזמנה ואת המדינה בה בוצעה ההזמנה.
* תצוגה של התפלגות המכירות עבור פריטים בעיצוב אישי אל מול פריטים מוכנים המוצעים באתר – תרשים העוגה מאפשר לצפות בפילוח בין שתי הקטגוריות באופן ברור ומאפשר למחלקת המכירות לנתח את הפופולריות והנחיצות של מכירת מוצרים בעיצוב מותאם אישית לאורך התקופות השונות ובמדינות השונות.
* מצורפת מפה שמאפשרת לבחון באופן ויזואלי את מספר ההזמנות לפי אזורים / מדינות. מדובר בחברה בינלאומית לכן מחלקת השיווק תרצה לדעת באילו מדינות כדאי להשקיע במאמצי הפרסום.
* דיאגרמה המציגה את התפלגות ההזמנות לפי החודשים השונים ומאפשרת זיהוי של תקופות שיא ושפל במכירות.
* מסנן לפי תאריכים מאפשר לצפות בכל טווח התאריכים בהם בוצעו הזמנות מהחברה.
* מסנן לפי מדינות יכיל את רשימת כלל המדינות מהן ניתן לבצע הזמנות.

המסננים מאפשרים ניתוח מעמיק יותר של הנתונים על פני תקופות שונות ובמדינות שונות .

בהפעלת המסננים כל הגרפים יעודכנו לתקופה / למדינה הרלוונטית.

**תצוגת הדו"ח :**

Graphical user interface, application

Description automatically generated

**לשם יצירת הדו"ח מימשנו את ה- VIEW הבא:**

(ה – VIEW יוצר טבלה סיכומית המאפשרת לנו להשתמש בנתונים ולסנכרן ביניהם בקלות.)

CREATE VIEW Summary\_view

AS

SELECT o.[Order ID],

o.[Order DT],

o.[Credit Card],

[Address – country] = ISNULL( s.[Address – country], 'United States'),

o.TotalCost,

CatalogNumber = CAST(dbo.CONTAIN.[Surfboard ID] as varchar(20)) + '.' +

CAST(dbo.CONTAIN.[Version ID] AS varchar(20)),

vertion\_quantity = dbo.CONTAIN.quantity,

personal\_design\_units = dbo.[INCLUDE].Units

FROM orders AS o LEFT JOIN dbo.SHIPPING AS s ON s.[Order ID] = o.[Order ID]

JOIN dbo.CONTAIN ON CONTAIN.[Order ID] = O.[Order ID]

JOIN dbo.[INCLUDE] ON [INCLUDE].[Order ID] = O.[Order ID]

**לוח מחוונים**

לוח המחוונים נועד כדי לשרת את הדרג הניהולי של החברה , מטרתו לתת תמונת מצב על פעילות החברה בהתייחס לתקופה מסוימת. הדו"ח יבחן את הפעילות הכלכלית של החברה ויהווה בסיס להחלטות אסטרטגיות בנוגע לעתיד החברה ומדיניותה.

הדו"ח יבחן את מצב החברה באספקטים שונים אל מול היעדים שהחברה הציבה לעצמה ויאפשר ראייה רחבה וספציפית יותר באמצעות סננים.

הנתונים בדו"ח המחוונים ישמשו את הדרג הניהולי של החברה לשם התווית מדיניות וקבלת החלטות אסטרטגיות להמשך.

**תוכן לוח המחוונים :**

* סך הכל הכנסות תקופתיות אל מול היעד שהציבה החברה הפילטרים מאפשרים בחינת הכנסות תקופתיות לפי טווח תאריכים ולפי מדינות.
* מספר ההזמנות הכולל אל מול היעד שהציבה החברה.
* מספר הלקוחות הפעילים באתר החברה אל מול היעד שהציבה לעצמה החברה.
* עבור שלושת הנתונים לעיל החברה הציבה יעד של גידול ב-10%.
* מפה שמאפשרת לראות עבור כל מדינה את מספר ההזמנות שהתבצעו ממנה ואת ההכנסה ממכירות באותה מדינה.
* גרף הכנסות לפי רבעונים ושנים מציג את ההכנסות בכל תקופה בשלוש השנים האחרונות. גרף זה ישמש כדי לבחון את תקופות השיא והשפל מבחינת רווחיות החברה.
* מסנן לפי תאריכים מציג את כל טווח התאריכים בהם בוצעו הזמנות מהחברה.
* מסנן לפי מדינות מציג את רשימת המדינות מהן ניתן לבצע הזמנות.

נרצה לאפשר סינון של כלל הנתונים בלוח לפי תאריכים ולפי מדינות מה שיאפשר ניתוח מעמיק יותר של הנתונים הן מבחינת התקופות השונות והן מבחינת הסקה גיאוגרפית על כדאיות התרחבות או צמצום שלוחות החברה באזורים מסוימים.

**לשם יצירת לוח המחוונים מימשנו את ה- VIEW'S הבאים:**

CREATE VIEW Target\_orders\_view

AS

SELECT Orders\_Target = CAST((COUNT([Order ID])\*1.1) AS int)

FROM dbo.ORDERS

CREATE VIEW Target\_Customers\_view

AS

SELECT Customers\_Target = CAST((COUNT(DISTINCT[Credit Card])\*1.1) AS int)

FROM dbo.ORDERS

CREATE VIEW Target\_revenue\_view

AS

SELECT TotalCostTarget = (SUM(TotalCost)\*1.1)

FROM dbo.ORDERS

**תצוגת לוח המחוונים :**

Graphical user interface, application

Description automatically generated

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

## שאילתות עסקיות המשלבות Window Functions 10%

**שאילתה 1 :**

נסווג את דגמי הגלשנים המוצעים באתר ל – 5 קטגוריות על פי המחירים שלהם - כאשר טווח המחירים המסומן ב- 1 הוא הזול ביותר וטווח המחירים המסומן ב-5 הוא היקר ביותר.

נבחן עבור כל קטגוריה כמה יחידות נמכרו ממנה ונדרג את קטגוריות המוצרים על פי מספר המכירות. הפלט יאפשר לנו לבחון האם יש קשר בין טווח המחירים לבין כמות המכירות , כלומר נראה האם יש הבדל ניכר במכירות בטווחי מחירים שונים.

באמצעות הדירוג נדע איזו קטגוריית מחירים היא הנמכרת ביותר ואיזו קטגוריה נמכרת פחות.

(השאילתה משתמשת בפונקציות RANK ו- NUTLE)

**נוסח השאילתה ב – SQL :**

SELECT R.[PRICE-RANGE],

R.UnitSoldFromCategory,

RANK () OVER (

ORDER BY UnitSoldFromCategory DESC

) sales\_rank

FROM

(SELECT B.[PRICE-RANGE],

UnitSoldFromCategory = SUM(C.quantity)

FROM dbo.CONTAIN AS C JOIN (SELECT [Surfboard ID],

[Version ID],

[Version Price],

NTILE (5) OVER (

ORDER BY [Version Price]

) [PRICE-RANGE]

FROM dbo.VERSIONS) AS B

ON B.[Surfboard ID] = C.[Surfboard ID]

AND B.[Version ID] = C.[Version ID]

GROUP BY B.[PRICE-RANGE]) AS R

**פלט השאילתה :**

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

**שאילתה 2 :**

ניצור פונקציה שמקבלת חודש ושנה ומחזירה פירוט של רווחי החברה בחודש הקודם לחודש המבוקש , את רווחי החברה בחודש המבוקש , את היעד שהציבה לעצמה החברה לחודש העוקב ואת הרווח של החברה בחודש העוקב.

הפונקציה מאפשרת לקבל תמונת מצב סיכומית של רווחי החברה ברבעון מסויים בשנה מסוימת , יכולה להועיל בניתוח מגמות עליה/ירידה בין החודשים ובהשוואה בין תקופות זהות בשנים שונות.

**נוסח השאילתה ב – SQL :**

CREATE FUNCTION QUARTER\_PROFITS(@MONTH INT , @YEAR INT)

RETURNS TABLE

AS RETURN

SELECT K.[Month] ,

[previous month Profit] ,

[Monthly profit],

[Next Month Destenation],

[next month Profit]

FROM (SELECT [Month] ,

LAG([Monthly profit],1) OVER (

ORDER BY month

) [previous month Profit],

[Monthly profit] ,

[Next Month Destenation] = [Monthly profit]\*1.05, LEAD([Monthly profit],1) OVER ( ORDER BY month

) [next month Profit]

FROM (SELECT [Month] = MONTH([Order DT]) ,

[Monthly profit] = SUM(TotalCost)

FROM ORDERS

WHERE YEAR([Order DT]) = @YEAR

GROUP BY MONTH([Order DT])) AS R) AS K

WHERE [Month] = @MONTH

**הדגמה של הפעלת הפונקציה :**

SELECT \*

FROM dbo.QUARTER\_PROFITS(4,2021)

**הפלט המתקבל :**

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

## שילוב מערכתי של מספר כלים 10%

## נרצה לאפשר ביטול הזמנות שלא חלפו 7 ימים ממועד הזמנתם.

עבור הזמנות שבוטלו נרצה לשמור את פרטי ההזמנה והלקוח כדי לנסות לנתח את סיבת הביטול וליצור קשר עם הלקוח בעת הצורך.

הפרוצדורה תבצע בדיקה האם לאפשר את מחיקת ההזמנה באמצעות מספר פונקציות שונות הבודקות את פרטי ההזמנה , במידה ופרטי ההזמנה מאפשרים זאת הפרוצדורה תמחק את ההזמנה מרשימת ההזמנות ותפעיל טריגר שמכניס את ההזמנה שנמחקה לטבלת ההזמנות שנמחקו.

ניצור טבלה עבור ההזמנות שנמחקו :

CREATE TABLE DeletedOrders (

[ORDER ID] INT NOT NULL,

[Order DT] DATETIME NOT NULL,

[Delivery Method] VARCHAR(20) null,

[Credit Card] VARCHAR(20) null,

[TotalCost] MONEY NULL,

[cancellation Date] datetime NULL,

[Customer email] varchar(30) NULL

CONSTRAINT pk\_DeletedOrders PRIMARY KEY ([ORDER ID], [Order DT])

)

פונקציה שמחזירה את מספר הימים שעברו מביצוע ההזמנה :

CREATE FUNCTION checkDaysFromOrder (@orderID INT)

RETURNS INT

AS BEGIN

DECLARE @answer INT

SELECT @answer = DATEDIFF(DD ,O.[Order DT] , GETDATE())

FROM dbo.ORDERS AS O

WHERE O.[Order ID] = @orderID

RETURN @answer

END

פונקציה שבודקת האם קיימת הזמנה עם מספר ההזמנה שהזין הלקוח ובנוסף בודקת האם כרטיס האשראי שהוזן עבור הזמנה תואם את כרטיס האשראי שהזין הלקוח :

(מחזירה 1 אם כן ו 0 אם לא ) :

CREATE FUNCTION CHECK\_DETAILS (@orderID INT , @CC\_number VARCHAR(20))

RETURNS INT

AS BEGIN

DECLARE @answer INT

SELECT @answer = COUNT([Order ID])

FROM dbo.ORDERS

WHERE [Order ID] = @orderID AND [Credit Card] = @CC\_number

RETURN ISNULL(@answer,0)

END

פונקציה שמקבלת מספר כרטיס אשראי ומחזירה את המייל של בעל הכרטיס :

CREATE FUNCTION getEmailFromCCnumber (@CCnumber VARCHAR(20))

RETURNS VARCHAR(30)

AS BEGIN

DECLARE @email VARCHAR(30)

SELECT @email = c.Email

FROM dbo.CREDITCARDS AS cc JOIN dbo.CUSTOMERS AS c ON c.Email = cc.EMAIL

WHERE cc.Number = @CCnumber

RETURN @email

END

טריגר שמעדכן את טבלת ההזמנות שנמחקו בעת מחיקה הזמנה :

CREATE TRIGGER Update\_deleted\_orders

ON dbo.ORDERS

FOR DELETE

AS

INSERT INTO DeletedOrders

SELECT Deleted.[Order ID] , Deleted.[Order DT] , Deleted.[Delivery Method],

Deleted.[Credit Card] , Deleted.TotalCost , NULL,NULL

FROM Deleted

הפרוצדורה שמשתמשת בשלל הכלים המוצגים לעיל כדי לאפשר פעולת מחיקה של הזמנה :

CREATE PROCEDURE DELETE\_ORDER (@ORDERID INT , @CCnumber VARCHAR(20))

AS

IF ( [dbo].[CHECK\_DETAILS](@ORDERID , @CCnumber) = 0)

BEGIN

PRINT 'Wrong input or order does not exist,try again'

END

ELSE IF ([dbo].[checkDaysFromOrder](@ORDERID) > 7)

BEGIN

PRINT 'More then 7 days left, please phone to customer service'

END

ELSE

BEGIN

DELETE FROM dbo.ORDERS WHERE [Order ID] = @ORDERID

PRINT 'Your order,' + CAST(@ORDERID AS VARCHAR(20)) +' ,has been successfully deleted'

UPDATE [dbo].[DeletedOrders]

SET [Customer email] = (dbo.getEmailFromCCnumber (@CCnumber)),

[cancellation Date] = GETDATE()

WHERE [dbo].[DeletedOrders].[ORDER ID] = @ORDERID

END

נדגים את השימוש בפרוצדורה :

לשם הדוגמה נוסיף הזמנה שניתן למחוק לטבלת ההזמנות :

INSERT INTO dbo.ORDERS

(

[Order ID],

[Order DT],

[Delivery Method],

[Credit Card],

TotalCost

)

VALUES

( 505, -- Order ID - int

GETDATE(), -- Order DT - datetime

'self-pick up', -- Delivery Method - varchar(20)

'0006 2813 8918 0945', -- Credit Card - varchar(20)

NULL -- TotalCost - money

)

הפעלת הפרוצדורה :

EXECUTE [dbo].[DELETE\_ORDER] 505 , '0006 2813 8918 0945'

הפלט לאחר הפעלת הפרוצדורה :

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטיטבלת הזמנות לפני המחיקה :

טבלת ההזמנות שנמחקו לפני המחיקה (טבלה ריקה) :

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטיטבלת ההזמנות לאחר ביצוע הפרוצדורה (ההזמנה נמחקה):

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי

טבלת ההזמנות שנמחקו לאחר ביצוע הפרוצדורה :

(ההזמנה שנמחקה נוספה לטבלת ההזמנות שנמחקו)



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

**דו"ח עסקי המסכם פעילות חודשית באתר :**

דו"ח שנתי המפרט עבור כל חודש בשנה מסוימת את מספר ההזמנות שבוצעו מהאתר במהלך החודש , מספר הלקוחות הפעילים באתר באותו החודש , הרווח הכולל של החברה באותו חודש, מספר היחידות המעוצבות אישית שנמכרו במהלך החודש ואימייל של הלקוח הרווחי ביותר עבור החברה באותו החודש (הלקוח שסכום הזמנתו היה הגדול ביותר).

איסוף הנתונים לשם יצירת הדו"ח יתבצע באמצעות שימוש בפונקציות שונות המפורטות להלן.

פונקציה שמחזירה את מספר ההזמנות בחודש הרלוונטי :

CREATE FUNCTION ORDERS\_PER\_MONTH (@month VARCHAR(2) , @year INT)

RETURNS INT

AS BEGIN

DECLARE @OrdersNum INT

SELECT @OrdersNum = COUNT([Order ID])

FROM ORDERS

WHERE MONTH([Order DT])= @month AND YEAR([Order DT])= @Year

RETURN @OrdersNum

END

פונקציה שמחזירה את הרווח הכולל של החברה באותו חודש :

CREATE FUNCTION TOTAL\_REVENUE\_PER\_MONTH (@month VARCHAR(2) , @year INT)

RETURNS MONEY

AS BEGIN

DECLARE @revenue MONEY

SELECT @revenue = SUM(TotalCost)

FROM ORDERS

WHERE MONTH([Order DT])= @month AND YEAR([Order DT])= @Year

RETURN @revenue

END

פונקציה שמחזירה את מספר הלקוחות שרכשו באתר החברה בחודש המבוקש :

CREATE FUNCTION Customers\_per\_month (@month VARCHAR(2) , @year INT)

RETURNS INT

AS BEGIN

DECLARE @customers INT

SELECT @customers = COUNT (DISTINCT C.Email)

FROM ORDERS AS O JOIN dbo.CREDITCARDS AS CC ON CC.Number = O.[Credit Card]

JOIN dbo.CUSTOMERS AS C ON C.Email = CC.EMAIL

WHERE MONTH([Order DT])= @month AND YEAR([Order DT])= @Year

RETURN @customers

END

פונקציה שמחזירה את מספר הפריטים המעוצבים אישית שנמכרו בחודש נתון :

CREATE FUNCTION PersonalDesigns\_Sold\_per\_Month (@month VARCHAR(2) , @year INT)

RETURNS INT

AS BEGIN

DECLARE @UnitsSOLD INT

SELECT @UnitsSOLD = SUM(I.Units)

FROM ORDERS AS O JOIN dbo.INCLUDE AS I ON I.[Order ID] = O.[Order ID]

WHERE MONTH([Order DT])= @month AND YEAR([Order DT])= @Year

RETURN @UnitsSOLD

END

פונקציה שמחזירה את ה – BEST CLIENT בחודש נתון – הלקוח שרכש את ההזמנה היקרה ביותר:

CREATE FUNCTION CLIENT\_OF\_THE\_MONTH (@month VARCHAR(2) , @year INT)

RETURNS VARCHAR(30)

AS BEGIN

DECLARE @CLIENT VARCHAR(30)

SELECT @CLIENT = C.EMAIL

FROM ORDERS AS O JOIN dbo.CREDITCARDS AS CC ON CC.Number = O.[Credit Card]

JOIN dbo.CUSTOMERS AS C ON C.Email = CC.EMAIL

WHERE (SELECT maxCost = MAX(O.TotalCost)

FROM ORDERS AS O JOIN dbo.CREDITCARDS AS CC ON CC.Number =

O.[Credit Card] JOIN dbo.CUSTOMERS AS C ON C.Email = CC.EMAIL

WHERE MONTH([Order DT])= @month AND YEAR([Order DT])= @year) = O.TotalCost

RETURN @CLIENT

END

**הדו"ח הכולל – נפיק את הדו"ח עבור שנת 2021 ונקבל עבור כל חודש בשנה זו את כלל פריטי המידע המבוקשים:**

SELECT DISTINCT [Month] = MONTH(OD.[Order DT]) ,

[Orders] = dbo.ORDERS\_PER\_MONTH (CAST (MONTH(OD.[Order DT]) AS VARCHAR(2)), 2021),

[Customers] = dbo.Customers\_per\_month (CAST (MONTH(OD.[Order DT]) AS VARCHAR(2)), 2021),

[Total Revenue] = dbo.TOTAL\_REVENUE\_PER\_MONTH(CAST (MONTH(OD.[Order DT]) AS VARCHAR(2)), 2021),

[Personal Designed products] = dbo.PersonalDesigns\_Sold\_per\_Month(CAST (MONTH(OD.[Order DT]) AS VARCHAR(2)), 2021),

[Best Client's Email] = dbo.CLIENT\_OF\_THE\_MONTH(CAST (MONTH(OD.[Order DT]) AS VARCHAR(2)), 2021)

FROM dbo.ORDERS AS OD JOIN dbo.CREDITCARDS AS CR ON CR.Number = OD.[Credit Card]

JOIN dbo.CUSTOMERS AS CU ON CU.Email = CR.EMAIL

ORDER BY MONTH(OD.[Order DT])

תמונה שמכילה שולחן

התיאור נוצר באופן אוטומטי