|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מס' קבוצה | אתר | | | תאריך הגשה |
| 32 | www.halloweencostumes.com | | | 17\/06/2023 |
| חברי הצוות - מספרי ת.ז | | | | |
| 316284355 | | 209496843 | 208385930 | |

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

## תרשים ה-ERD שהוגש בחלק ב'

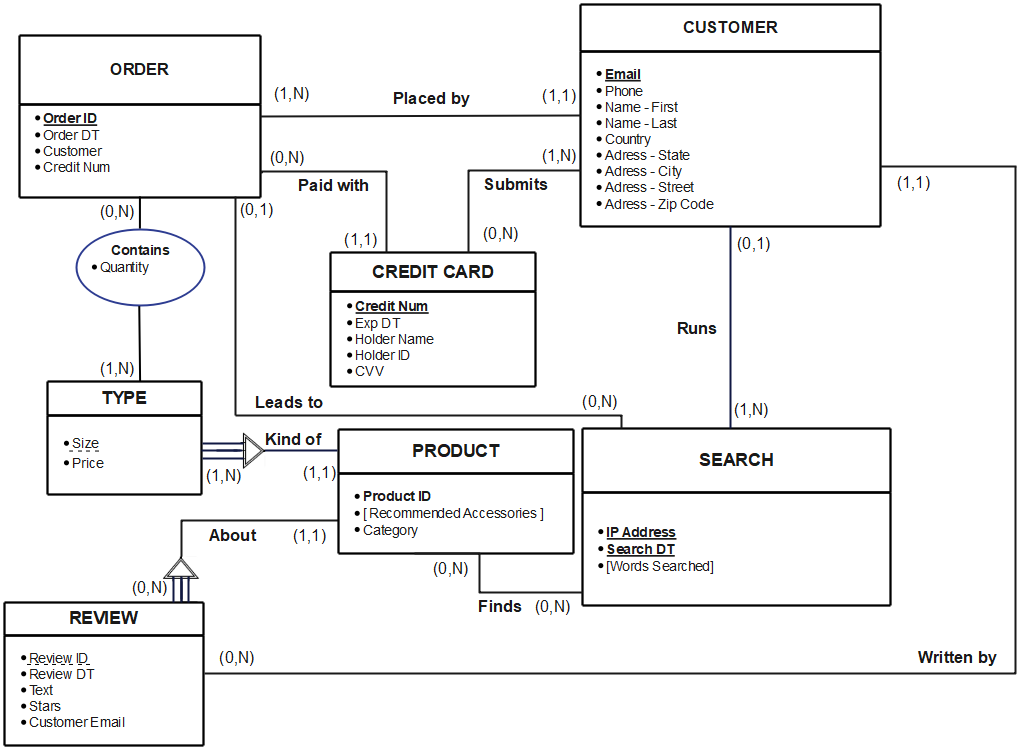
**תמונה שמכילה טקסט, תרשים, תוכנית, מקביל

התיאור נוצר באופן אוטומטי**

**ההערות שניתנו להגשה הקודמת**

* בחלק מהישויות שמנו את המפתחות הזרים- לא צריכים להופיע בתרשים
* חסר שדה CVV בישות כרטיס אשראי
* הקשר בין ORDER ל-PRODUCT לא טוב כיוון שלא ניתן לדעת באיזה סוג תחפושת מדובר
* שדה RATE בישות PRODUCT הוא שדה חישובי שלא צריך להופיע בתרשים
* Runs מופיע עם קרדניליות של N:1 בתרשים ובקרדינליות N:0 בטבלת ההנחות

**תרשים ה-ERD המתוקן + ההנחות שהנחנו:**



|  |  |
| --- | --- |
| **מרכיב** | **הנחות** |
| Connection: Runs | * לא כל חיפוש מוביל להזמנה, לכן קרדינליות של 0:1. * ניתן להזמין מוצר ללא חיפוש, לכן הקרדינליות של N:1. |
| Connection: Leads to | * לא כל חיפוש מוביל למוצר, לכן קרדינליות של 0:N. |
| Entity: Search | * בעת ביצוע החיפוש, נשמרת כתובת הIP של המכשיר. הכתובת IP וזמן שעה של חיפוש יזהו את החיפוש הספציפי (Search DT). |
| Connection: Contains | * בכל הזמנה אפשר לבחור כמה פריטים נרצה להזמין מכל מוצר. |
| Weak entity: TYPE | * הפכנו את הסוג תחפושת לישות חלשה של מוצרת מכיוון שלכל תחפושת יש מחיר שונה בגלל הגודל של המוצר. * מוצר יכול להופיע בגדלים שונים ומחירים שונים. |
| Weak entity: Review | * לכל מוצר יש ביקורת שהיא על מוצר ספציפית וייחודי לו |

## עיצוב לוגי (מודל נתונים טבלאי)

* **CUSTOMERS** (Email , phone, Name-First, Name-Last , Country, Address-State, Address-City, Address-Street, Address-Number, Address-ZIP code)
* **CREDIT CARDS**(Credit Num , Exp DT , Holder Name , Owner ID, CVV)
* **ORDERS** (Order ID, Order DT , Customer Email (CUSTOMERS) , Credit Num (Credit Cards))
* **SEARCHES** (IP Address, Search DT , Words Searched, Order ID (ORDERS), Customer Email (CUSTOMERS))
* **PRODUCTS** (Product ID , Category)
* **TYPES**(Product ID (Products) , Size , price)
* **CONTAINS**(Order ID (ORDERS), Product ID(PRODUCTS), Size (TYPES), Quantity)
* **SUBMITS**(Customers Email (CUSTOMERS) , Credit Num (Credit Cards))
* **RECOMMENDED ACCESSORIES**(Product ID (PRODUCTS) , Recommended accessories)
* **FINDS** (Product ID (PRODUCTS), {IP Address, Search DT} (SEARCHES))
* **REVIEWS** (Product ID (PRODUCTS), Review ID, Review DT , Text , Stars, Customer Email (CUSTOMERS))

# פירוט שדות בסיס הנתונים והערכת גודלם

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Field** | **Data Type** | **Size (Bytes)** |
| **CUSTOMERS** | Email | Varchar(25) | 25 |
| Name-First | Varchar(20) | 20 |
| Name-Last | Varchar(20) | 20 |
| Phone Number | Varchar(10) | 10 |
| Address-State | Varchar(20) | 20 |
| Address-City | Varchar(20) | 20 |
| Address-Street | Varchar(20) | 20 |
| Address-Number | Varchar(20) | 20 |
| Address-ZIP code | Varchar(20) | 20 |
| **Total Record Size (Bytes)** | | **175** |
| **ORDERS** | Order ID | Varchar(20) | 20 |
| Order DT | Date Time | 8 |
| Customer Email | Varchar(25) | 25 |
| Credit Num | BigInt | 8 |
| **Total Record Size (Bytes)** | | **61** |
| **SEARCHES** | IP- Address | Varchar(13) | 13 |
| Search DT | Date Time | 8 |
| Order ID | Varchar(20) | 20 |
| Customer Email | Varchar(25) | 25 |
| Words Searched | Varchar(100) | 100 |
| **Total Record Size (Bytes)** | | **166** |
| **PRODUCTS** | Product ID | Varchar(20) | 20 |
| Category | Varchar(50) | 50 |
| **Total Record Size (Bytes)** | | **70** |
| **CREDIT CARDS** | Credit Num | BigInt | 8 |
| Exp DT | Date | 8 |
| Holder Name | Varchar(20) | 20 |
| Holder ID | Int | 4 |
| CVV | Int | 4 |
| **Total Record Size (Bytes)** | | **44** |
| **TYPES** | Product ID | Varchar(20) | 20 |
| Size | Varchar(20) | 20 |
| Price | Money | 4 |
| **Total Record Size (Bytes)** | | **44** |
| **CONTAINS** | Order ID | Varchar(20) | 20 |
| Product ID | Varchar(20) | 20 |
| Size | Varchar(20) | 20 |
| Quantity | Tinyint | 1 |
| **Total Record Size (Bytes)** | | **61** |

|  |  |  |  |
| --- | --- | --- | --- |
| **SUBMITS** | Email | Varchar(25) | 25 |
| Credit Num | BigInt | 8 |
| **Total Record Size (Bytes)** | | **33** |
| **FINDS** | Product ID | Varchar(20) | 20 |
| IP- Address | Varchar(20) | 20 |
| Search DT | Date Time | 8 |
| **Total Record Size (Bytes)** | | **48** |
| **REVIEWS** | Product ID | Varchar(20) | 20 |
| Review ID | int | 4 |
| Review DT | Date Time | 8 |
| Text | Varchar(500) | 500 |
| Stars | Tinyint | 1 |
| Customer Email | Varchar(50) | 50 |
| **Total Record Size (Bytes)** | | **583** |

## מטלה 1 (25%) – שאילתות

# שתי שאילתות SELECT ללא קינון

1. **מהות השאילתה-** אילו לקוחות ביצעו הזמנות של יותר מ100 מוצרים בחודש הראשון של 2022 וביצעו בחודש השני של אותה השנה פחות הזמנות   
**תרחיש עסקי-** לקוחות שביצעו הזמנות של יותר מ100 מוצרים בחודש הם לקוחות גדולים בעלי חשיבות עסקית גדולה לחברה, ובעזרת השאילתה נוכל לזהות אם לקוחות אלו בדעיכה ולפעול בהתאם על מנת לשמר אותם.

SELECT C.Email, C.[First Name], C.[Last Name], NumOfOrders = COUNT(\*)

FROM CUSTOMERS AS C

JOIN ORDERS AS O ON C.Email = O.[Customer Email]

JOIN [CONTAINS] AS CO ON O.[Order ID] = CO.[Order ID]

WHERE MONTH(O.[ORDER DT]) = 1

GROUP BY C.Email, C.[First Name], C.[Last Name]

HAVING SUM(QUANTITY) > 10

EXCEPT

SELECT C.Email, C.[First Name], C.[Last Name], NumOfOrders = COUNT(\*)

FROM CUSTOMERS AS C

JOIN ORDERS AS O ON C.Email = O.[Customer Email]

JOIN [CONTAINS] AS CO ON O.[Order ID] = CO.[Order ID]

WHERE MONTH(O.[ORDER DT]) = 2

GROUP BY C.Email, C.[First Name], C.[Last Name]

HAVING SUM(QUANTITY) > 10

ORDER BY NumOfOrders;

A screenshot of a computer

Description automatically generated

**פלט השאילתה:**

2. **מהות השאילתה -** חמשת המוצרים שקיבלו את הדירוג הכי גבוה והוזמנו על ידי לקוחות ממדינת קוניטקיט.  
**תרחיש עסקי –** חמשת המוצרים שקיבלו דירוג גבוה וביקורות והוזמנו ע"י לקוחות ממדינת קונטיקט, הם מוצרים מובילים ופופולריים בשוק הלקוחות באזור. בעזרת השאילתה נוכל לדעת איזה מוצרים לפרסם במדינה זו.

SELECT TOP 5 R.[Product ID], R.Stars , AVGStars = AVG(R.Stars)

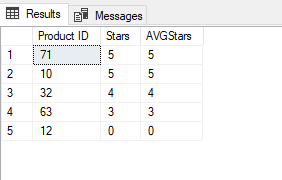
FROM REVIEWS AS R JOIN PRODUCTS AS P ON R.[Product ID] = P.[Product ID] JOIN CUSTOMERS AS C ON R.[Customer Email] = C.Email

WHERE C.[Address-State] = 'Connecticut'

GROUP BY R.[Product ID], R.Stars

ORDER BY AVGStars DESC

**פלט השאילתה:**



# שתי שאילתות SELECT מקוננות (10%)

**שאילתה 1:**  
**מהות השאילתה** – כמה הרווחנו ממוצרים שנמכרו במידה מסוימת ב-3 או יותר הזמנות שונות.   
**תרחיש עסקי –** כיוון שאנו מוצאים אילו מוצרים נמכרו במידה ספציפית ב-3 הזמנות או יותר, עלינו להבין מה מוצלח בהם ולשווק בהתאם את התחפושות הנוספות. בנוסף, אנו יודעים כמה אנו מרוויחים מהן ובהתאם יכולים ליצור מטרות דומות לתחפושות האחרות.

SELECT A.[Product ID],A.Size, [Total Revenue] = Sum(A.Amount\*T.Price)

FROM TYPES AS T JOIN (

SELECT [Product ID], Size, Amount=Sum(Quantity)

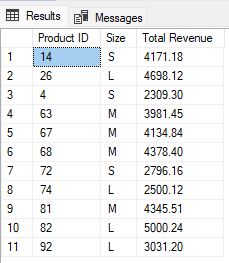
FROM [CONTAINS]

GROUP BY [Product ID], Size

Having count(\*)>3

) AS A ON T.[Product ID]=A.[Product ID]

GROUP BY A.[Product ID], A.Size

**פלט השאילתה:**

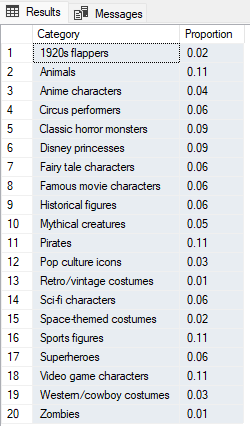
**שאילתה 2:**  
**מהות השאילתה** – מה אחוז ההזמנות שבהן הוזמן מוצר מקטגוריה מסוימת  
**תרחיש עסקי –** ברצוננו לדעת אילו תחפושות רלוונטיות יותר ואילו פחות. חישוב זה יעזור לנו להבין מה שיעור הפופולריות של כל קטגוריית תחפושות, וממנה ליצור יותר תחפושות/להעלות את מחיר התחפושות מאותה קטגוריה כדי למקסם רווחים.

SELECT P.Category,

CAST(COUNT(\*) \* 1.0 / (SELECT COUNT(\*) FROM ORDERS) AS DECIMAL(10, 2)) AS Proportion

FROM PRODUCTS AS P JOIN [CONTAINS] AS C ON P.[Product ID] = C.[Product ID]

GROUP BY P.Category

**פלט השאילתה:**

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

**שאילתה 1:**  
**מהות השאילתה** – שאילתה שחותכת בין לקוחות שגם מהווים נתח של יותר מאחוז מסך כל ההזמנות וגם השאירו יותר מביקורת אחת.  
**תרחיש עסקי –**ברצוננו לדעת אילו לקוחות גם קנו הרבה פריטים וגם השאירו כמות גדולה של ביקורות, כך נוכל לדעת אילו לקוחות נאמנים יש לנו שמבצעים הרבה הזמנות וחשובה להם גם איכות המוצרים. בכך אולי נוכל לדעת איך נוכל לתת להם תמריצים להישאר לקוחות שלנו ממחשבה שלקוחות אילו הינם הכי חשובים לחברה.

SELECT Email

FROM CUSTOMERS AS C JOIN ORDERS AS O ON C.Email = O.[Customer Email]

GROUP BY Email

HAVING COUNT (\*) >0.01 \* (SELECT COUNT (\*) FROM ORDERS)

INTERSECT

SELECT Email

FROM REVIEWS AS R JOIN CUSTOMERS AS C ON C.Email = R.[Customer Email]

GROUP BY Email

HAVING COUNT (\*)>=2

**פלט השאילתה:**

**תמונה שמכילה טקסט, צילום מסך, מספר, גופן

התיאור נוצר באופן אוטומטי**

**שאילתה 2:**  
**מהות השאילתה** מוסיף עמודת סטטוס ללקוחות ומעדכן אותה לכל לקוח אם הוא מרוצה או לא על סמך כמות ממוצעת של כוכבים שנתן למוצרים (אם מתחת ל3 לא מרוצה אחרת מרוצה).  
**תרחיש עסקי –** ברצוננו לדעת אילו לקוחות מרוצים ואילו לא מרוצים מהמוצרים שהזמינו כדי שנוכל לדעת בעתיד איך לשמר לקוחות ואיך לגרום להם כן להיות מרוצים בעתיד

ALTER TABLE CUSTOMERS

ADD [STATUS] VARCHAR(20);

UPDATE CUSTOMERS

SET [STATUS] = CASE

WHEN (

SELECT AVG(R.STARS)

FROM REVIEWS AS R

WHERE R.[Customer Email] = CUSTOMERS.Email

) <= 3 THEN 'unsatisfied'

ELSE 'satisfied'

END;

## מטלה 2 (25%) – יישומי כלים מתקדמים

# View (5%)

CREATE VIEW V\_RevnuePerProductAndSize AS

SELECT T.[Product ID],P.Category,C.[Order ID],C.Size,C.Quantity,T.Price, TotalAmountPerProductSize = T.Price\*C.Quantity

FROM PRODUCTS AS P JOIN [CONTAINS] AS C ON P.[Product ID] = C.[Product ID]

JOIN TYPES AS T ON C.Size = T.Size AND P.[Product ID] = T.[Product ID]

ה-View הנ"ל מציג איחוד בין שלוש טבלאות- Products, Types and [Contains]. השימוש ב-View זה בא לידי ביטוי כמעט בכל View שיצרנו לטובת לוח המכוונים והדו"ח שיצרנו בעזרת ה-Power BI. שילוב הטבלאות הנ"ל מביא לנו את האיחוד הנחוץ של הזמנה, כמות המוצרים שהוזמנו בה, מחיר כל פריט (תוך פילוג של מידות שונות) והמחיר הכולל להזמנה. השימוש של נתונים אלו ביחד רלוונטי לכל חישוב רווחים בחברה בנוסף לחישובים של רווח לפי חלוקה מסוימת- לפי קטגוריה, תאריך, מוצר וכו'.

# פונקציות (Functions)

**פונקציה טבלאית -**   
**מהות הפונקציה -** מקבלת כקלט את המזהה של הלקוח ומחזירה טבלה של כל ההזמנות שאותו הלקוח ביצע.

**תרחיש עסקי –** בעזרת הפונקציה נוכל באופן מהיר ויעיל לזקק אילו הזמנות בוצעו ע"י לקוח ספציפי.

DROP FUNCTION IF EXISTS CUSTOMER\_ORDERS;

GO

CREATE FUNCTION CUSTOMER\_ORDERS (@CID VARCHAR(40))

RETURNS TABLE

AS

RETURN (

SELECT C.Email, C.[First Name], C.[Last Name], O.[Order ID]

FROM CUSTOMERS AS C

JOIN ORDERS AS O ON C.Email = O.[Customer Email]

WHERE C.Email = @CID

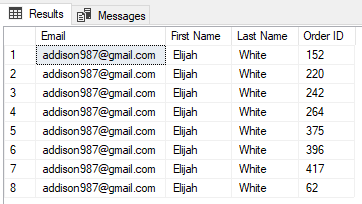
);

**שימוש בפונקציה:**

SELECT \*

FROM dbo.CUSTOMER\_ORDERS('addison987@gmail.com')

**פלט:**



**פונקציה סקלרית -**   
**מהות הפונקציה -** מקבלת כקלט מזהה של מוצר ומחזירה את הציון (הדירוג) הממוצע שלו .

**תרחיש עסקי –** בעזרת הפונקציה נוכל באופן מהיר ויעיל להוציא את הדירוג של מוצר ובכך לדעת אילו מוצרים הם הכי אהובים ואותם לשים בעמוד הראשי ובמומלצים.

CREATE FUNCTION P\_Rate (@PID VARCHAR(20))

RETURNS INT

AS

BEGIN

DECLARE @Rate INT;

SELECT @Rate = AVG(R.stars)

FROM PRODUCTS AS P JOIN REVIEWS AS R ON P.[Product ID] = R.[Product ID]

WHERE P.[Product ID] = @PID;

RETURN @Rate;

END

---Example

SELECT RATE = dbo.P\_Rate('3')

A screenshot of a computer

Description automatically generated with medium confidence**פלט:**

**-TRIGGER-**

**מהות הטריגר –** ניצור עמודת ציון לטבלת המוצרים ונרצה שכל פעם שנוסיף, נעדכן או נמחק ביקורת, לכל מוצר יתעדכן הציון הממוצע שלו .

**תרחיש עסקי –** בעזרת הטריגר הזה אנו נוכל לדעת לכל מוצר מה הציון הממוצע שלו ובכך לבצע בו שינויים כדי להפוך אותו ליותר אטרקטיבי .

ALTER TABLE PRODUCTS

ADD RATE REAL;

GO

DROP TRIGGER trg\_UpdateRate

CREATE TRIGGER trg\_UpdateRate ON REVIEWS

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

UPDATE PRODUCTS

SET RATE = (

SELECT CAST(SUM(REVIEWS.STARS)\*1.0/COUNT(\*) AS DECIMAL(10, 8))

FROM REVIEWS

WHERE PRODUCTS.[Product ID] = REVIEWS.[Product ID]

)

FROM PRODUCTS

WHERE PRODUCTS.[Product ID] IN (

SELECT [Product ID] FROM inserted

UNION

SELECT [Product ID] FROM deleted

);

END;

**קלט לדוגמה:**

---Example

INSERT INTO REVIEWS VALUES

('11','11','2021-05-13 00:00:00.000', 'Was Great', '5', 'scarlett543@gmail.com')

DELETE FROM REVIEWS WHERE [Product ID] = '11' and [Review ID] = '11'

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

הפרוצדורה הבאה מקבלת כמשתנים מהמשתנה את הערכים הבאים: מזהה מוצר ומוצרים נלווים להוספה. הפרוצדורה בודקת קודם כל האם הצירוף הנ"ל קיים, ובמידה והוא לא היא מוסיפה לטבלה מוצרים נלווים לתחפושות את הערכים שהמשתמש הכניס. פרוצדורה זו חיונית במידה ואנו רוצים ליצור/להוסיף מוצרים נלווים לתחפושות

CREATE PROCEDURE SP\_AddAccessory

@ProductID varchar(20),

@Accessory varchar(100)

AS

BEGIN

IF NOT EXISTS (SELECT \* FROM RECOMMENDED\_ACCESSORIES WHERE [Product ID] = @ProductID AND [Recommended Accessories] = @Accessory)

BEGIN

INSERT INTO RECOMMENDED\_ACCESSORIES ([Product ID], [Recommended Accessories])

VALUES (@ProductID, @Accessory);

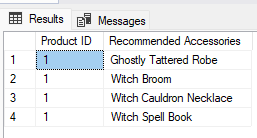
END;

END;

EXEC SP\_AddAccessory '1', 'Witch Broom'

DROP PROCEDURE SP\_AddAccessory

**דוגמה בקוד + פלט השאילתה:**

****SELECT \*

FROM RECOMMENDED\_ACCESSORIES

WHERE [Product ID] = '1'

## מטלה 3 (20%) – כלים להצגת נתונים

**דו"ח עסקי**

הדו"ח שיצרנו הינו דו"ח עסקי המיועד למחלקת השיווק של החברה ומטרתו הצגת נתונים רלוונטיים אשר יעזרו למחלקה לשפר את הביצועים העתידיים שלה ובעקבות כך להגדיל הכנסות ולהגדיל גם כן את קהל הלקוחות. הדו"ח כמובן ישפר את קבלת ההחלטות העתידיות ויעזור לסדר אותן באופן ויזואלי שיתאים ויקל על תהליך זה.

בנוסף, דו"ח זה יעזור למחלקה לשפר את ביצועיה ולהגדיל את רווחי החברה ואת קהל לקוחותיה. בעזרת הדו"ח המחלקה תוכל לתכנן את פעולותיה העתידיות לכל מדינה, תאריך ומדדים נוספים. בחרנו לעשות דו"ח קריא ללא נתונים מיותרים וכפילויות בכדי שיהיה פשוט להבנה ועל מנת לתמצת את המידע הרלוונטי לצורה הנוחה ביותר לעובד.

פירוט על חלקי הדו"ח:

* טבלה אשר מציגה את מספר ההזמנות של כל מדינה- הטבלה תעזור למחלקה לראות כמה לקוחות יש בכל מדינה, לבדוק את רמת ההצלחה שלה שם ולדעת באילו מדינות כדאי להגביר את מאמץ הפרסום בכדי להכניס ממדינה זו עוד לקוחות.
* טבלת מדינות- מאפשרת להם להסתכל על מדינות ספציפיות או על כל המדינות. הטבלה קלה לשימוש וחלק מערכי הדו"ח משתנים לפיה. טבלה זו עוזרת לשלוט בדו"ח בצורה יותר רלוונטית במקרה ונרצה להסתכל על מדינה ספציפית בדוח.
* כרטיסייה אשר מציגה את ההכנסה הממוצעת להזמנה- מספר זה עוזר למחלקה לדעת הזמנה ממוצעת להזמנה מה שעוזר למחלקה מאוד בתכנון של עלות למוצר וכמה לפרסם כל מוצר מבחינת כדאיות השקעה בעלות הפרסום שלו.
* עוגה- מציגה את כמות הפריטים הנמכרים מכל קטגוריה, בעזרת עוגה זו נוכל לדעת אילו מוצרים הכי נמכרים אצלנו ובכך לדעת על אילו קטגוריות יש לנו לשים יותר מאמץ בפרסום או אפילו לדעת שכדאי לנו כל הזמן להוסיף מוצרים לקטגוריה זו בידיעה שימכרו יותר.
* גרף הזמנות לפי תאריך- הגרף מראה את מספר ההזמנות בכל יום, וכך נדע לתכנן באילו ימים אנחנו נרצה להשקיע יותר בפרסום ממוקד ואילו ימים יותר חזקים בחודש. בנוסף, נוכל לתכנן לעתיד ממש על פי יום איך יעבוד הייצור והפרסום.

תמונה שמכילה טקסט, צילום מסך, תרשים, גופן

התיאור נוצר באופן אוטומטי

**לוח מחוונים**

מטרת לוח המחוונים שיצרנו הינה הצגת מידע באופן ברור והנגשה שלו לדרג הארגוני.  
המטרה בלוח זה הינה לצמצם כמה שיותר מידע אשר יוכל לתרום לדרגים הנמוכים יותר ולהביא כאן בעיקר את המידע שיוכל לעזור לדרג הניהולי לקבל את החלטותיו.

פירוט על חלקי הלוח:

* מפה- המפה מציגה את סך ההכנסות בכל מדינה באמצעות הצגת גודל העיגול שנמצא עליה, בכך המנהלים יוכלו לדעת אילו מדינות ואילו יבשות תופסות חלק שוק גדול יותר של מוצרי החברה ובכך לקבל החלטות עתידיות בנוגע לכל מדינה.
* גרף מכירות לפי מוצר- הגרף מציג את כמות המכירות לכל מוצר בחברה כולה. דבר זה נותן למנהלים את האופציה לדעת איך לתכנן את העתיד של כל מוצר בין אם בכמות שמייצרים או אפילו להחליט אם לתת תמריצים לעובדים שמייצרים אותו ועוד.
* גרף הזמנות לפי חודש- הגרף מציג את כמות ההזמנות בכל חודש ונותן למנהלים אינדיקציה למצב המכירות ובכך יכולת תכנון קדימה של התנהלות החברה.
* בנוסף הוספנו סימניות עם נתונים כללים על כל החברה:

כמות הזמנות- מציג למנהלים כמה הזמנות בוצעו בכללי מהחברה.

הכנסה כללית- מציג למנהלים כמה הכנסות יש לחברה, דבר בעל חשיבות עליונה למנהלים, בעזרת כלי זה הם יוכלו לדעת כמה כסף יש לחברה ובכך לדעת איך להשתמש בו.

מספר לקוחות- מציג כמה לקוחות יש לחברה, מספר זה יכול לעזור למנהלים להבין את סדר הגודל של החברה וכמה גדול נתח השוק שהם תופסים מסך העולם.

תמונה שמכילה טקסט, מפה, צילום מסך, מספר

התיאור נוצר באופן אוטומטי

## מטלה 4 – יישומים מתקדמים

## שאילתות עסקיות המשלבות Window Functions (10%)

**מהות השאילתה** – בכל קטגוריה אנחנו ממספרים את המוצרים לפי רמת מכירות וגם מחלקים את כל המוצרים לארבע מחלקות אשר מחולקות לפי סכום כולל שנכנס מהן

**תרחיש עסקי** – השאילתה נותנת לנו מידע עסקי חשוב לגבי המכירות של המוצרים. בעזרתה אנחנו יכולים להבין אילו מוצרים מכל קטגוריה מוכרים הכי הרבה ואילו נמכרים הכי הרבה בכללי.

בעזרת שאילתה זו אנחנו יכולים לבצע הרבה החלטות חשובות בין אם לא לספק יותר מוצר ספציפי או אולי להגביר יצור ממוצר אחר, להפסיק לייצר קטגוריה שלמה ועוד.

SELECT P.[Product ID],P.Category, TOTAL=SUM(Quantity\*T.Price),

Ranking = Rank()

OVER (PARTITION BY Category

ORDER BY SUM(Quantity\*Price)DESC) ,

ProductGroup = Ntile(4)

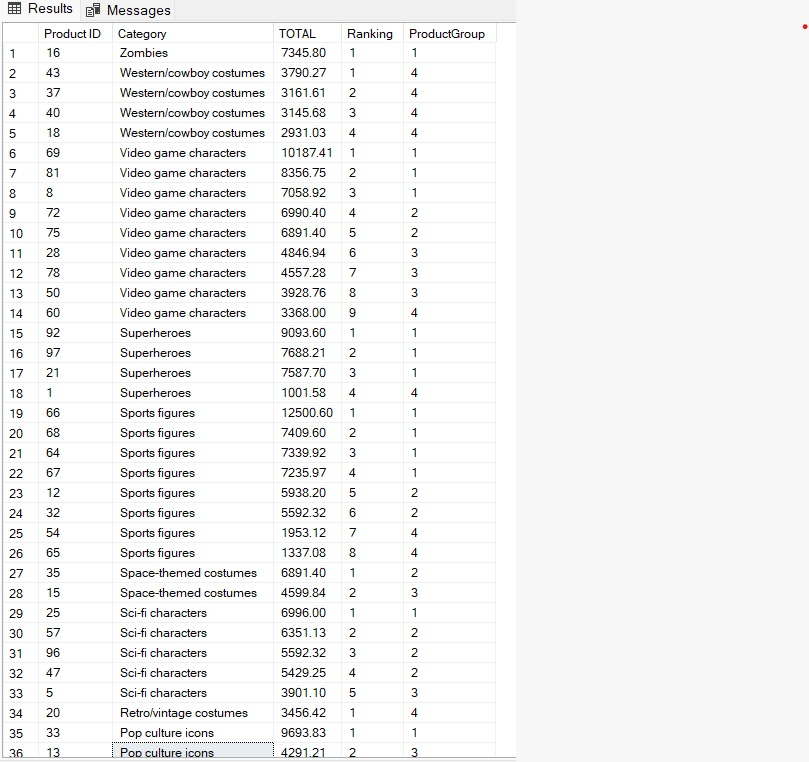
OVER (

ORDER BY (SUM(Quantity\*T.Price))DESC)

FROM PRODUCTS AS P JOIN [CONTAINS] AS C ON P.[Product ID]=C.[Product ID] JOIN ORDERS AS O ON O.[Order ID] = C.[Order ID] JOIN TYPES AS T ON T.[Product ID]=P.[Product ID]

GROUP BY P.Category , P.[Product ID]

ORDER BY P.Category DESC, SUM(Quantity\*T.Price)DESC

**פלט השאילתה:**

**100 רשומות-**

**מהות השאילתה –** השאילתה מציגה טבלה של הזמנות ולקוחות- עמודה של הזמנה נוכחית, עמודה של ממוצע ההזמנות עד עכשיו ועמודה שמציגה את הפער מההזמנה הגדולה ביותר.

**תרחיש עסקי –** לפי עמודת הממוצע נוכל לדעת בערך כמה כל לקוח מוציא ונוכל לדעת להיערך ולחזות לקראת ההזמנה הבאה שלו וכך לדעת פחות או יותר את הכנסות החברה העתידיות.

לגבי עמודת ההפרש מההזמנה הגדולה ביותרת כך נדע אם הלקוח בירידה או בעליה מההזמנה הגדולה שלו, כך אפשר למשל במקרה שלא לתת לו תמריצים והנחות כדי שיקנה עוד וכדומה

SELECT C.Email, C.[First Name], C.[Last Name],

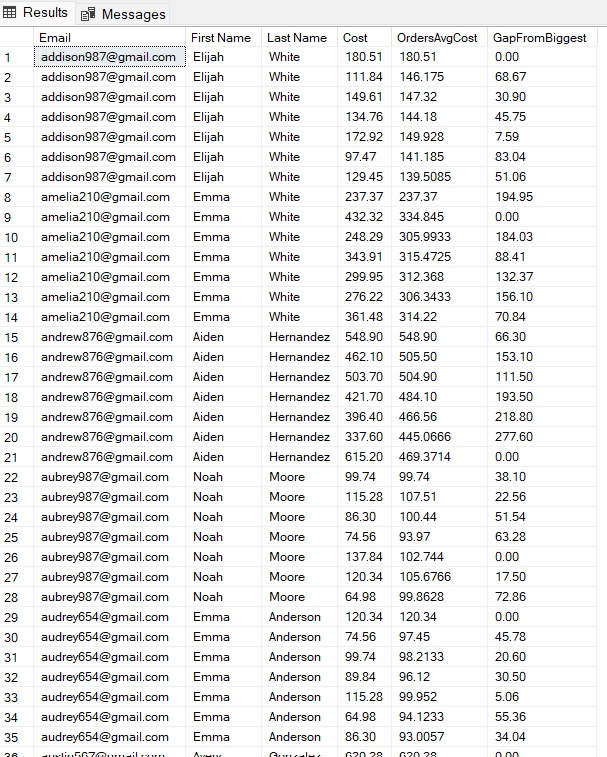
Cost= (Quantity\*T.Price),OrdersAvgCost=

AVG(Quantity\*T.Price) OVER (PARTITION BY C.Email ORDER BY C.Email

ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) , GapFromBiggest=

MAX(Quantity\*T.Price) OVER (PARTITION BY C.Email ORDER BY C.Email )-(Quantity\*T.Price)

FROM ORDERS AS O JOIN CUSTOMERS AS C ON O.[Customer Email]=C.Email JOIN [CONTAINS] AS CO ON O.[Order ID] = CO.[Order ID] JOIN TYPES AS T ON T.[Product ID]=O.[Order ID]



**פלט השאילתה:**

**כ-707 רשומות-**

## שילוב מערכתי של מספר כלים (10%)

**מהות השילוב –** בעזרת שילוב כל הכלים האלו אנו פועלים למען מטרת שינוי המחיר של מוצר מסוים במידה מסוימת. בנוסף, כל אלמנט בנפרד בר שימוש לגורמים שונים אך מהות הפרוצדורה היא השילוב של כולם יחד.

**תרחיש עסקי –** התרחיש העסקי המוצג הוא היכולת של החברה לשנות את מחירי המוצרים, המוצרים נבחרים בהתאם לקטגוריה ולמידה שהוכנסו ,המחירים משתנים בהתאם לאחוז השינוי שהוזנו ובהתאם לכיוון השינוי [0,1].

הפרוצדורה היא זאת שפועלת ראשונה, העסק מחליט לעדכן את מחירי המוצרים לכן בטבלת TYPES המחיר משתנה בהתאם.

CREATE PROCEDURE SP\_PRICECHANGE (@CATEGORY varchar(50) , @SIZE varchar(20) , @CHANGE REAL , @TYPEOFCHANGE BIT )

AS

BEGIN

UPDATE T

SET PRICE = T.PRICE \*

CASE

WHEN @TYPEOFCHANGE = 0 THEN (1 - @CHANGE)

ELSE (1 + @CHANGE)

END

FROM TYPES AS T

INNER JOIN PRODUCTS AS P ON T.[Product ID] = P.[Product ID]

WHERE P.Category = @CATEGORY

AND T.Size = @SIZE;

END;

EXEC SP\_PRICECHANGE 'Anime characters','XL', '0.2', '1'

הפונקציה המוצגת מחשבת את אחוז השינוי מן המחיר הישן למחיר החדש, היא נמצאת בנפרד כי השימוש שלה יכול להיות מגישות שונות ומסיבות שונות.

הפונקציה מקבלת את מס' המוצר, המחיר העדכני ביותר והמחיר הקודם.

לאחר החישוב היא מחזירה את אחוז השינוי, מה שיפעיל את הטריגר בעתיד.

CREATE FUNCTION CALCULATE ( @PRODUCTID varchar(20), @SIZE varchar(20) , @OLDPRICE money )

RETURNS varchar(20)

AS BEGIN

DECLARE @PRECENTEGE varchar(20)

SELECT @PRECENTEGE = ((PRICE - @OLDPRICE ) / @OLDPRICE )

FROM TYPES

WHERE [Product ID] = @PRODUCTID AND SIZE =@SIZE

RETURN @PRECENTEGE

END

Go

של אחוז השינוי .TYPESהוספת עמודה בטבלה

ALTER TABLE TYPES

ADD [PRECENTEGECHANGE] varchar(20);

ALTER TABLE TYPES

DROP COLUMN PRECENTch

GO

DROP TRIGGER dbo.UPDATETYPES;

GO

ברגע שהפרוצדורה מופעלת היא מעדכנת את הטבלה, הדבר מביא להפעלת הטריגר.

הטריגר הוא זה שמעדכן את עמודת השינוי באחוזים ומכניס את הערך המתאים, זאת בזכות הפונקציה אשר

DELETEDמחשבת ושולחת בחזרה לטריגר את הסקלר המתאים. הערך של המחיר הישן נשלח מטבלת

ובכך יעדכן את המחיר בטבלה.

CREATE TRIGGER UPDATETYPES ON TYPES

FOR UPDATE

AS

BEGIN

UPDATE T

SET PRECENTEGECHANGE= C.PERCENTAGE

FROM TYPES AS T

INNER JOIN

(

SELECT D.[Product ID], D.Size, dbo.CALCULATE(D.[Product ID], D.Size, D.Price) AS PERCENTAGE

FROM deleted AS D

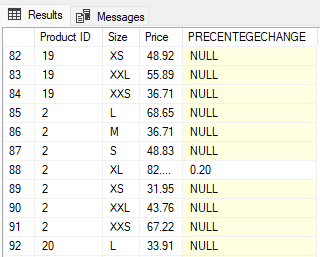
) AS C ON T.[Product ID] = C.[Product ID] AND T.Size = C.Size;

END;

**דוגמה לקלט:**

---Example

EXEC SP\_PRICECHANGE 'Anime characters','XL', '0.2', '1'

**דוגמה לפלט:**

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

הדו"ח שיצרנו הינו דו"ח עסקי שמסתכל ומנתח נתונים שונים לגבי כל מדינה בנפרד. החלטנו לחלק כך את הדו"ח כיוון שהפילוח הזה מאפשר לנו לראות לאילו מדינות אנחנו שולחים יותר משלוחים, באילו מדינות ההשקעה שלנו יותר/פחות כדאית, מה הקטגוריות הנפוצות ביותר בכל מדינה על מנת להגביר את הפרסום שלהן ועוד.

מוצג בזאת פירוט לגבי כל עמודה בדו"ח (למעט עמודת מדינה):

1. ימים מאז הזמנה אחרונה- בחרנו להציג את הנתון הזה כיוון שלדעתנו הוא משקף תמונת מצב עדכנית ונכונה לגבי מידת הרלוונטיות של החברה בכל מדינה. אנחנו כחברה צריכים להבין למה ישנן מדינות שלא הוזמן אליהן משלוח ב-X ימים האחרונים (הפרמטר שאנחנו בחרנו הוא 182 יום- חצי שנה בדיוק). ידע זה יעזור לנו להבין מתי עלינו לתגבר את הפרסום במדינות מסוימות ולפעמים מתי כבר לא כדאי לנו כחברה לשלוח למדינות מסוימות.
2. פעילה/לא פעילה- כפי שהסברנו בסעיף 1, קטגוריה זו בודקת האם נעשה משלוח למדינה בחצי השנה האחרונה ומחלקת את המדינות לשתי קטגוריות- פעילה (נעשה משלוח בטווח זמן זה) ולא פעילה (לא נעשה משלוח בטווח זמן זה). כפי שפורט למעלה, מידע זה יעזור להבין באילו מדינות אנחנו פחות מוכרים לאחרונה ובהתאם להפיק לקחים.
3. מספר ההזמנות לכל מדינה- מדובר בנתון בסיסי מאוד עליו אנו מתבססים בחלקים רבים לאורך העבודה. בחרנו לשלב אותו גם בטבלה זו כיוון שהוא מהווה בסיס להבנה של מידת הרלוונטיות של כל מדינה לחברה שלנו. יש לציין שהמדינות מוצגות לנו בטבלה בסדר הבא: מהמדינה אליה היו הכי הרבה הזמנות אל המדינה אליה הוזמנו הכי פחות הזמנות.
4. אחוז ההזמנות שהגיעו מחיפוש של לקוחות- באתר שלנו יש שורת חיפוש לתחפושות כמו גם אפשרויות שקופצות ישר במסך הבית ללא צורך בחיפוש. כיוון שישנן כמה אפשרויות להגיע למוצר, חשבנו כי עלינו להבין כיצד הלקוחות הגיעו בסופו של דבר להזמנה. עלינו להבין עד כמה שורת החיפוש ממוקמת טוב באתר, עד כמה היא נגישה וגם כמה הפיצ'רים האחרים (כמו למשל מוצרים מועדפים, המלצות האתר וכו') רלוונטיים. נתון זה עוזר לנו להבין איזה חלקים באתר יותר רלוונטיים ושימושיים.
5. מספר הלקוחות מכל מדינה- חשוב לנו כאתר לדעת מה מספר הלקוחות שיש לנו בכל מדינה. לכאורה יש לנו נתון שמספק תוצאה דומה: כמות ההזמנות מכל מדינה, אך אנו רואים ביניהם שוני משמעותי. כחברה, אנחנו צריכים לדעת האם כמות ההזמנות מכל מדינה מתבססת על כמות קטנה או גדולה של לקוחות. במידה ולמשל, כל ההזמנות מהפיליפינים היו מבוצעות על ידי אותו הלקוח, הנתון של כמות ההזמנות היה "שיקרי"- אנחנו חייבים להגביר את הפרסום שם כי אנו כנראה לא מוכרים (וזאת למרות שזו המדינה ממנה הזמינו הכי הרבה לקוחות).
6. קטגוריה מובילה בכל מדינה- שוב, ברצוננו לדעת אילו מהתחפושות והקטגורויות שאנו מציעים נמכרים יותר ואילו פחות. למשל, במידה ואנו מגלים שבמדינות מסוימות הביקוש לתחפושות מקטגוריית דמויות ספורט גדול, נגדיל את ההיצע שאנו מספקים למדינה זו כדי להגדיל רווחים.

**השאילתה:**

WITH DaysSinceLastOrderPerCountry AS (

SELECT C.Country, DaysSinceLastOrder = DATEDIFF (DD, MAX(O.[Order DT]), GETDATE()), Active = (CASE

WHEN DATEDIFF (DD, MAX(O.[Order DT]), GETDATE()) <= 182 THEN 'Active'

ELSE 'Inactive' END)

FROM ORDERS AS O JOIN CUSTOMERS AS C ON O.[Customer Email]=C.Email

GROUP BY C.Country

),

SearchesPerCountry AS (

SELECT C.Country, [Searches Amount] = COUNT(\*)

FROM SEARCHES AS S JOIN CUSTOMERS AS C ON S.[Customer Email] = C.Email

GROUP BY C.Country

),

SearchesToOrdersPerCountry AS (

SELECT S.Country, [Orders by Search Precentage] = CAST(1.0\*S.[Searches Amount] / V.Amount AS DECIMAL(10, 2))

FROM SearchesPerCountry As S JOIN V\_OrdersPerCountry AS V ON S.Country = V.country

),

CustomersPerState AS (

SELECT Country, [Number Of Customers] = COUNT(\*)

FROM CUSTOMERS

GROUP BY Country

),

CategoriesPerState AS (

SELECT CU.Country, P.Category, [Amount From Category] = COUNT(\*)

FROM PRODUCTS AS P JOIN [CONTAINS] AS CO ON P.[Product ID]= CO.[Product ID] JOIN ORDERS AS O ON CO.[Order ID]=O.[Order ID] JOIN

CUSTOMERS AS CU ON O.[Customer Email]=CU.Email

GROUP BY P.Category, CU.Country

),

TopCategoryPerState AS (

SELECT Country, [Top Category] = Category, Row = ROW\_NUMBER () OVER (PARTITION BY Country

ORDER BY [Amount From Category] DESC )

FROM CategoriesPerState

)

SELECT D.Country, D.DaysSinceLastOrder, D.Active,[Number Of Orders] = V.amount,

S.[Orders by Search Precentage], C.[Number Of Customers],

T. [Top Category]

FROM DaysSinceLastOrderPerCountry AS D JOIN SearchesToOrdersPerCountry AS S ON

D.Country = S.Country JOIN

CustomersPerState AS C ON S.Country = C.Country JOIN V\_OrdersPerCountry AS V ON C.Country = V.Country JOIN

TopCategoryPerState AS T ON V.country = T.Country

WHERE T.Row = 1

ORDER BY [Number Of Orders] DESC

**פלט השאילתה:**

A screenshot of a computer

Description automatically generated with low confidence