

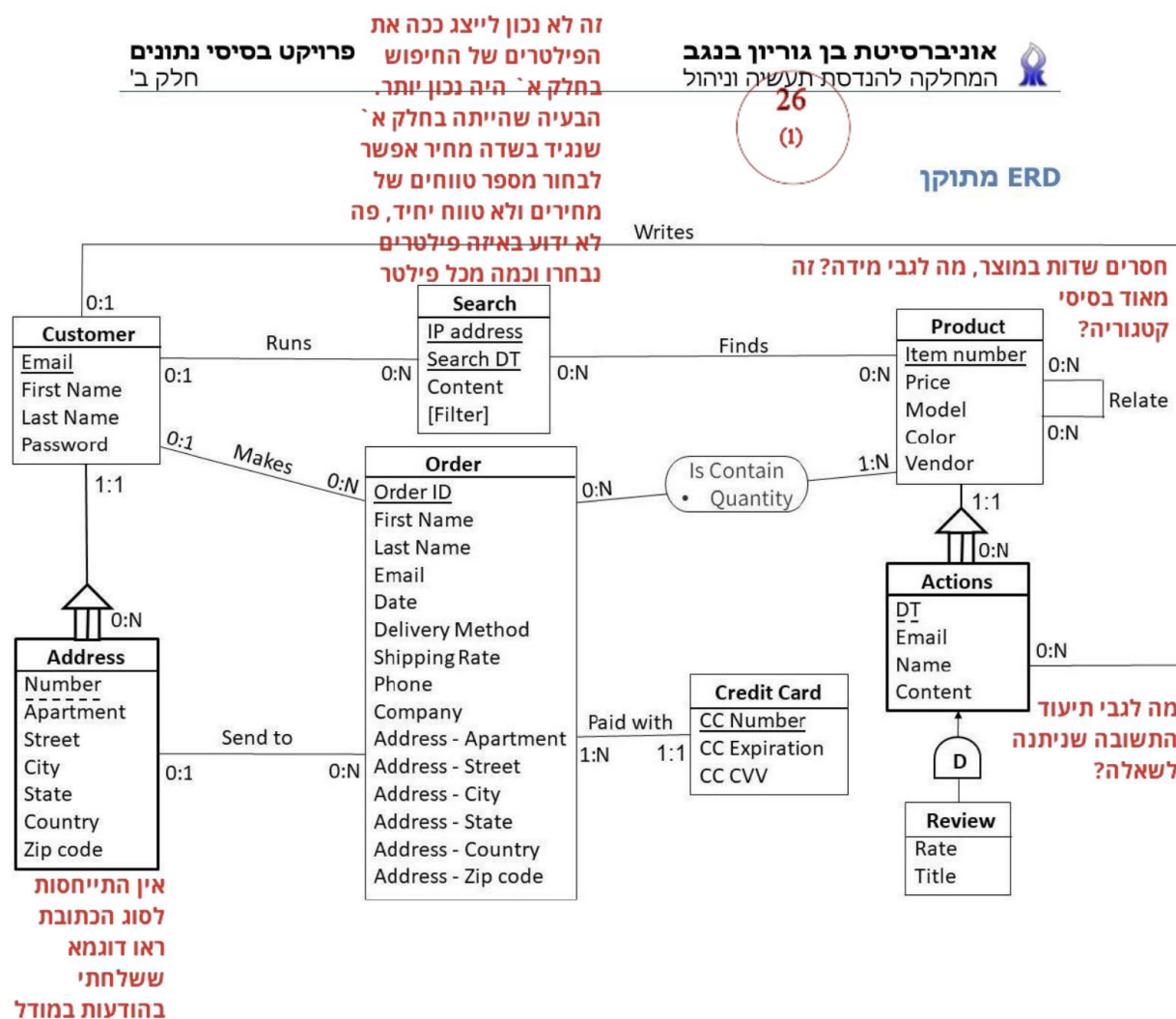


מס' קבוצה	אתר	תאריך הגשה
12	Skates.com	08/06/2021
חברי הצוות - מספרי ת.ז.		
315695643	316297480	316161694

פרויקט בסיסי נתונים – חלק ג'

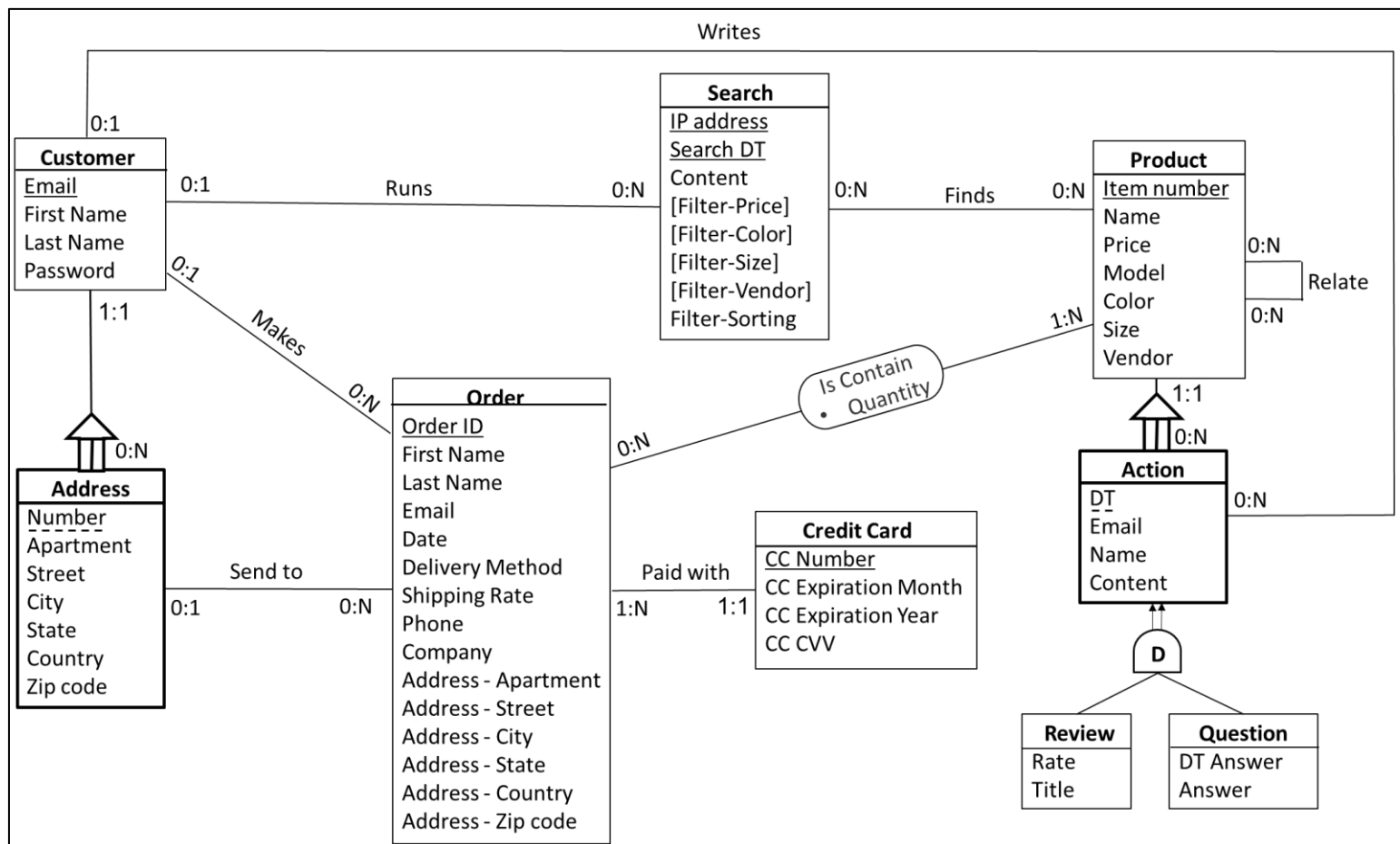
מטלת חובה מקדימה

ERD (חלק ב')





ERD (מעודכן)



עיקרי השינויים:

- התווספו שדות ליישות PRODUCT (SIZE).
- התווספו שדות מרובי ערכים של פילטרים התואמים את השדות המאפיינים את PRODUCT.
- הוספת יישות בן ל ACTION עבור QUESTION ושינוי סוג ההורשה. כמו כן, הוספת שדה תשובה ותאריך התשובה בQUESTION.



הנחות מודל ה-ERD

הנחות	ERD מרכיב
<ul style="list-style-type: none"> Customer מייצג לקוח רשום באתר. בהסתכל על הקרדינליות של הקשרים היוצאים מישות זו, ישנה אופציה לקיום הקשר ע"י משתמש אורח (מיוצג ע"י קרדינליות 0) ולא לקוח רשום. 	<p>הקשרים שיוצרת הישות</p> <p>Customer</p> <p>(Writes, Runs, Makes)</p>
<ul style="list-style-type: none"> הורשה: באתר ישנה האפשרות (Action) לכתובת ביקורת ושאלות שאלות עבור מוצר. מרבית השדות של אפשרויות אלה משותפים, ולכן בחרנו למדל באמצעות הורשה מלאה בה ניתן ייצוג לשדות הייחודיים של כל פעולה. מזהה חלקי (DT) בכל רגע נתון יכולה להתבצע במקסימום פעולה אחת עבור מוצר מסוים. ישות Question: לכל שאלה ניתנת עד תשובה אחת בלבד. 	<p>ישות Action ובניה</p> <p>(Review, Question)</p>
<ul style="list-style-type: none"> בכל חיפוש מוצגים באתר מקסימום שמונה מוצרים נוספים, אך אנו מסיקים כי הוא מקושר ליותר. 	<p>קשר רקורסיבי של ישות</p> <p>Related - Product</p>
<ul style="list-style-type: none"> באתר קיימת יותר מאופציות תשלום אחת, אנו התמקדנו בתשלום באמצעות כרטיס אשראי. 	<p>ישות Credit-Card</p>
<p>במידה ונבחרה אופציית המשלוח (ולא איסוף עצמי), משתמש שאינו רשום מחויב להכניס כתובת. לעומתו משתמש שרשום קיימת האפשרות להשתמש באחת הכתובות מספר הכתובות או לחילופין להזין כתובת אחרת כרצונו.</p>	<p>שיוך כתובת לישות</p> <p>Order</p>
<p>כל שדה בישות Product שעליו ניתן לפלטר ב-Search הינו שדה מרובה ערכים שבהם יופיעו כל הערכים שהמשתמש בחר לסנן לפיהם. כמו כן, שדה Filter-Sorting הינו שדה אטומי המאפשר למיין את התוצאות לפי סינון מוגדר (לדוגמה, עפ"י סדר האלפביתי, מחיר מהזול ליקר וכדומה)</p>	<p>שדות בישות Search:</p> <p>[Filter-Price]</p> <p>[Filter-Color]</p> <p>[Filter-Size]</p> <p>[Filter-Vendor]</p> <p>Filter-Sorting</p>



המודל הטבלאי (חלק ב')

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

26
(3)

PRODUCTS (Item Number, Price, Model, Color, Vendor)

CUSROMERS (Email, First Name, Last Name, Password)

CREDIT_CARDS (CC Number, CC Expiration, CC CVV)

ADDRESSES (Email (CUSTOMERS), Number, Apartment, Street, City, State, Country, Zip Code)

SEARCHES (IP Address, Search DT, Content, Email (CUSROMERS))

FILTERS ({IP Address, Search DT} (SEARCHES), Filter)

ORDERS (Order ID, First Name, Last Name, Email_Order, Date, Delivery Method, Shipping Rate, Phone, Company, Address_Apartment, Address_Street, Address_City, Address_State, Address_Country, Address_Zip_Code, CC Number (CREDIT CARDS), {Email_Customer (CUSROMERS), Number} (ADDRESSES))

ACTIONS (Item Number (PRODUCTS), DT, Email_Action, Name, Content, Email_Customer (CUSROMERS))

REVIEWS ({Item Number, DT} (ACTIONS), Rate, Title)

RELATES (Item Number – A (PRODUCTS), Item Number – B (PRODUCTS))

IS_CONTAINS (Order ID (ORDERS), Item Number (PRODUCTS), Quantity)

FINDS (Item Number (PRODUCTS), {IP Address, Search DT} (SEARCHES))

- בטבלאות ORDEERS ו-ACTIONS ישנן מספר כתובות אימייל ולכן שיננו את שמות המפתחות הזרים המכילים אימייל בהתאם לישות ממנה הגיעו. לדוגמה, אימייל שהינו מפתח זר מטבלת CUSTOMERS יקרא בשם Email_Customer.
- כפי שניתן לראות מתרשים ה-ERD, בעקבות הקשרים של ישות ORDER עם ADDRESS ו-CUSTOMER נוצרו כפילויות בשדות Email למיניהם בטבלת ORDERS. ישנה בעיה לממש זאת בפקודות ה-SQL ולכן, בחרנו לממש את הקשרים בעזרת השדה Email_Customer שהינו חלק משני מפתחות זרים. האחד, לטבלת CUSTOMERS והשני לטבלת ADDRESSES.

נרמול נתונים נוסף

- שדה filter הינו שדה מרובה ערכים של הישות search, בכדי לעמוד בתנאי הנרמול מסדר ראשון הוספנו טבלה חדשה בשם filters עם מפתח זר לטבלת searches. כך, נקבל שתי טבלאות עם שדות אטומיים בלבד.
- קיימת תלות הדדיות בין השדות Item Number <--> Model כל מוצר מופיע באתר פעם אחת בלבד, ולכן Model הוא מפתח אלטרנטיבי של טבלת PRODUCTS. לכן לא קיימת הפרה מסדר שלישי בטבלת זו.

חסר נרמול מסדר שלישי - סוג המשלוח גורר בהכרח את מחיר המשלוח, במידה ואתם לא חושבים ככה - היה צורך לכתוב הנחות בהתאם ולהתייחס לזה בצורה כזו או אחרת.

-2

(3)

- 4 -



המודל הטבלאי (מעודכן)

PRODUCTS (Item Number, Price, Model, Color, Size, Vendor)

NAMES (Item Number (PRODUCTS), Name)

CUSROMERS (Email, First Name, Last Name, Password)

CREDIT_CARDS (CC Number, CC Expiration_Month, CC Expiration_Year, CC CVV)

ADDRESSES (Email (CUSTOMERS), Number, Apartment, Street, City, State, Country, Zip Code)

SEARCHES (IP Address, Search DT, Content, Email (CUSROMERS), Filter Sorting)

FILTERS PRICES ({IP Address, Search DT} (SEARCHES), Filter Price)

FILTERS COLORS ({IP Address, Search DT} (SEARCHES), Filter Color)

FILTERS SIZES ({IP Address, Search DT} (SEARCHES), Filter Size)

FILTERS VENDORS ({IP Address, Search DT} (SEARCHES), Filter Vendor)

ORDERS (Order ID, First Name, Last Name, Email_Order, Date, Delivery Method, Shipping Rate, Phone, Company, Address_Apartment, Address_Street, Address_City, Address_State, Address_Country, Address_Zip_Code, CC Number (CREDIT CARDS), {Email_Customer (CUSROMERS), Number} (ADDRESSES))

QUESTIONS (Item Number (PRODUCTS), DT, Email_Action, Name, Content, DT_Answer, Answer, Email_Customer (CUSROMERS))

REVIEWS (Item Number (PRODUCTS), DT, Email_Action, Name, Content, Rate, Title, Email_Customer (CUSROMERS))

RELATES (Item Number – A (PRODUCTS), Item Number – B (PRODUCTS))

IS_CONTAINS (Order ID (ORDERS), Item Number (PRODUCTS), Quantity)

FINDS (Item Number (PRODUCTS), {IP Address, Search DT} (SEARCHES))

- בטבלאות ORDERS ו-ACTIONS ישנן מספר כתובות אימייל ולכן שיננו את שמות המפתחות הזרים המכילים אימייל בהתאם לישות ממנה הגיעו. לדוגמה, אימייל שהינו מפתח זר מטבלת CUSTOMERS יקרא בשם Email_Customer.
- כפי שניתן לראות מתרשים ה-ERD, בעקבות הקשרים של ישות ORDER עם ADDRESS ו-CUSTOMER נוצרו כפילויות בשדות Email למיניהם בטבלת ORDERS. ישנה בעיה לממש זאת בפקודות ה-SQL ולכן, בחרנו לממש את הקשרים בעזרת השדה Email_Customer שהינו חלק משני מפתחות זרים. האחד, לטבלת CUSTOMERS והשני לטבלת ADDRESSES.
- בטבלת ORDERS - עלות המשלוח (Shipping Rate) נקבע בהתאם ליזמן אספקת המשלוח, ולכן משתנה בהתאם לסוג המשלוח (Delivery Method): משלוח או איסוף עצמי.
- יישום מודל ההורשה על בסיס המרת disjoin-total. (אפשרות א').



מטלה 1 – שאלות

שתי שאלות SELECT ללא קיבון

1. שאלתה: כמה מוצרים נרכשו מכל ספק מתחילת שנת 2021, מתוך הספקים שמכרו לפחות ארבעה מוצרים?
מטרה עסקית: איתור הספקים הדומיננטים במחצית הראשונה של שנת 2021.

```
SELECT P.Vendor , TotalSales= SUM(I.Quantity)
FROM ORDERS AS O JOIN IS_CONTAINS AS I ON O.Order_ID=I.order_ID JOIN
PRODUCTS AS P ON P.Item_Number=I.Item_Number
WHERE YEAR(O.Date)=2021
GROUP BY P.Vendor
HAVING SUM(I.Quantity) > 3
ORDER BY 2 DESC
```

הפלט:

	Vendor	TotalSales
1	Carzy skates	12
2	Ridell	8
3	Oliver Thomas	7
4	Adidas	5
5	Free-fly	5
6	K2	5
7	Oakley	5
8	Champion	4

2. שאלתה: עבור כל מוצר באתר, מי מהמוצרים המקושרים אליו זול ממנו ב-35%?
מטרה עסקית: זיהוי מוצרים בעלי מאפיינים דומים הזולים באופן משמעותי אחד מהשני.

```
SELECT Product=P1.Item_Number, N1.Name, P1.Price, RelatedProduct=P2.Item_Number,
N2.Name , P2.Price
FROM PRODUCTS AS P1 JOIN NAMES AS N1 ON P1.Item_Number=N1.Item_Number JOIN
RELATES AS R ON P1.Item_Number=R.Item_Number_A JOIN PRODUCTS AS P2 ON
P2.Item_Number=R.Item_Number_B JOIN NAMES AS N2 ON
N2.Item_Number=P2.Item_Number
WHERE P1.Price > P2.Price AND P1.Price/P2.Price > 1.35
ORDER BY 1,6
```

הפלט (15 רשומות ראשונות):

Product	Name	Price	RelatedProduct	Name	Price
1	10647 Roces RC2 unisex rollerskates	108.917	10120	Jackson finesse 180 girls figure skates	65.9691
2	10796 K2 preformance men protective gear	39.1885	35591	Rollerblade girls skate helmet	5.1828
3	10796 K2 preformance men protective gear	39.1885	41957	Rollerblade boys skate helmet	20.4414
4	12462 Roller derby driftR artist black men rollerskates	120.7043	10120	Jackson finesse 180 girls figure skates	65.9691
5	13073 Jacson excel girls figure skates	93.0512	10120	Jackson finesse 180 girls figure skates	65.9691
6	14821 Roces RC3 unisex rollerskates	137.9274	10120	Jackson finesse 180 girls figure skates	65.9691
7	15557 Jackson finesse 50 girls figure skates	215.6723	10120	Jackson finesse 180 girls figure skates	65.9691
8	15557 Jackson finesse 50 girls figure skates	215.6723	12462	Roller derby driftR artist black men rollerskates	120.7043
9	15557 Jackson finesse 50 girls figure skates	215.6723	59833	Riedell opal boys figure skates	158.4324
10	16364 Riedell soar girl figure skates	123.9672	56755	Races Moody 2.0 adjustable girls ice skate	89.7903
11	17580 Rollerblase kids skate bag	86.5359	35591	Rollerblade girls skate helmet	5.1828
12	17580 Rollerblase kids skate bag	86.5359	41957	Rollerblade boys skate helmet	20.4414
13	17580 Rollerblase kids skate bag	86.5359	10796	K2 preformance men protective gear	39.1885
14	17592 Rollerblade Macroblade 70 3WD men inline ...	263.4537	73766	Riedell soar boy figure skates	178.3554
15	23212 Rollerblade Macroblade 100 3WD men inlin...	153.2823	92542	Rollerblade Macroblade 80 men inline skates	113.0541



שתי שאלות SELECT מקוננות

1. שאלתה שמחזירה TABLE בFROM: מהם חמשת המוצרים בעלי שיעור הגידול הגבוה ביותר בהיקף המכירות שלהם בין שנת 2019 לשנת 2020?
מטרה עסקית: מידע זה חשוב על מנת לדעת מהם המוצרים המבוקשים ביותר נכון לתקופת הביקוש האחרונה (זיהוי טרנדים) זאת על מנת ולהיערך לשנה הבאה.

```
SELECT TOP 5 X20.Item_Number, Sold2019=X19.Total, Sold2020=X20.Total,
Ratio = cast(X20.total as real) / cast(X19.total as real)
FROM ((SELECT DISTINCT P.Item_Number, Total=SUM(I.Quantity)
FROM PRODUCTS AS P JOIN IS_CONTAINS AS I ON P.Item_Number=I.Item_Number
JOIN ORDERS AS O ON O.Order_ID = I.Order_ID
WHERE year(O.Date) = 2020
GROUP BY P.Item_Number ) AS X20 JOIN
(SELECT DISTINCT P.Item_Number, Total=SUM(I.Quantity)
FROM PRODUCTS AS P JOIN IS_CONTAINS AS I ON P.Item_Number=I.Item_Number
JOIN ORDERS AS O ON O.Order_ID = I.Order_ID
WHERE year(O.Date) = 2019
GROUP BY P.Item_Number ) AS X19 ON X20.Item_Number = X19.Item_Number)
ORDER BY Ratio DESC
```

הפלט:

	Item_Number	Sold2019	Sold2020	Ratio
1	17229	1	5	5
2	10120	1	3	3
3	25016	1	3	3
4	30511	1	3	3
5	62527	1	3	3

2. שאלתה שמחזירה SCALAR בHAVING: מהן הערים שהיקף המכירות שבוצעו בהן הינו נמוך מממוצע המכירות לעיר (הצגה מהגבוה לנמוך).
מטרה עסקית: נרצה לזהות את הערים בהן היקף המכירות של החברה הינו נמוך ביחס לערים אחרות על מנת לחזק את הפרסום שלנו בערים אלו.

```
SELECT City = O.Address_City , Amount = SUM(P.Price * I.Quantity)
FROM PRODUCTS AS P JOIN IS_CONTAINS AS I ON P.Item_Number = I.Item_Number JOIN
ORDERS AS O ON I.Order_ID = O.Order_ID
GROUP BY O.Address_City
HAVING SUM(P.Price*I.Quantity) < (SELECT TotalSum = SUM(P.Price * I.Quantity )
FROM PRODUCTS AS P JOIN IS_CONTAINS AS I ON
P.Item_Number = I.Item_Number JOIN ORDERS AS O
ON I.Order_ID = O.Order_ID )
/(SELECT COUNT(DISTINCT Address_City)
FROM ORDERS)
ORDER BY 2
```



הפלט (15 רשומות ראשונות):

	City	Amount
1	Tucson	103.0239
2	Hialeah	119.6033
3	St. Louis	123.9672
4	Dallas	218.3915
5	Denver	237.1561
6	Oakland	259.4475
7	Baton Rouge	292.229
8	Tacoma	310.8272
9	Rochester	385.335
10	Chicago	476.6278
11	St. Paul	483.3947
12	Birmingham	498.2075
13	Fresno	525.6629
14	Newark	549.7785
15	New Orleans	573.1646

שתי שאלות מקוננות תוך שימוש במרכיבים נוספים

1. שאלת UPDATE: מתן 20% הנחה לעשרת המוצרים הכי פחות נמכרים בחצי שנה האחרונה.

מטרה עסקית: על ידי הורדת המחירים של מוצרים הכי פחות נמכרים, נוכל עודד את הצרכן לקנות אותם ובכך לגרום לעלייה במספר הרכישות שלהם.

```
UPDATE PRODUCTS
SET Price= (
    CASE WHEN PRODUCTS.Item_Number in (SELECT TOP 10 P.Item_Number
    FROM          PRODUCTS AS P JOIN IS_CONTAINS AS
    I ON P.Item_Number = I.Item_Number
    JOIN ORDERS AS O ON
    O.Order_ID = I.Order_ID
    WHERE        datediff(mm,O.Date,GETDATE())<= 6
    GROUP BY    P.Item_Number
    ORDER BY    SUM(I.Quantity))
    THEN PRODUCTS.Price*0.8
    ELSE PRODUCTS.Price END)
```

שאלת הרצה:

```
SELECT      Item_Number, Price
FROM        PRODUCTS
WHERE       Item_Number IN (SELECT TOP 10
    FROM          PRODUCTS AS P JOIN IS_CONTAINS AS I ON
    P.Item_Number = I.Item_Number JOIN ORDERS
    AS O ON O.Order_ID = I.Order_ID
    WHERE        datediff (mm, O.Date, GETDATE()) <= 6
    GROUP BY    P.Item_Number
    ORDER BY    SUM(I.Quantity))

ORDER BY Price
```




פלט אחרי הנחה:

	Item_Number	Price
1	17229	20.9717
2	30768	32.4453
3	31059	80.1994
4	31169	84.251
5	10143	92.9185
6	13073	93.0512
7	30098	106.2462
8	14821	137.9274
9	36537	157.1794
10	31403	212.9057

פלט לפני הנחה:

	Item_Number	Price
1	17229	26.2146
2	30768	40.5566
3	31059	100.2493
4	31169	105.3138
5	10143	116.1481
6	13073	116.314
7	30098	132.8077
8	14821	172.4093
9	36537	196.4743
10	31403	266.1321

2. שאלתה (הפרש): מיהם האנשים (לאו דווקא לקוחות רשומים) אשר שאלו יותר שאלות מהממוצע לאדם, אך לא ביצעו אף הזמנה.
מטרה עסקית: איתור לקוחות אשר מביעים עניין / מתלבטים אך אינם רוכשים בפועל אף מוצר.

```

SELECT      Email = E.Email_Action , Q.Name
FROM(  SELECT DISTINCT      Q1.Email_Action
      FROM
      WHERE
      ((SELECT AV
        FROM (SELECT Email_Action, AV =COUNT(DISTINCT DT)
              FROM QUESTIONS AS Q2
              WHERE Q1.Email_Action = Q2.Email_Action
              GROUP BY Email_Action) AS NUM_Q)
        >
        (SELECT AVRAGE = cast(SUM(TotalQ) AS real) /
          cast (COUNT(DISTINCT Email_Action) AS real)
        FROM (SELECT Email_Action, TotalQ =COUNT(DISTINCT DT)
              FROM QUESTIONS
              GROUP BY Email_Action) AS X))

EXCEPT

SELECT DISTINCT Email_Order
FROM ORDERS) AS E JOIN QUESTIONS AS Q ON E.Email_Action = Q.Email_Action
    
```

	Email	Name
1	aavicz142@hotmail.com	Chanda Todd
2	lfzlcas3@dnmdt.org	Terry West
3	ctzrdj.mxdptwsl@qfyvk.wrukpa.com	Derick Matthews
4	qgrzlspl.dtgw@kiulidgf.-olgjf.org	Tyson Mc Donald
5	ljwsfwh89@ohk-bi.org	Ramon Tyler
6	zhovptgj8@bphjr.sdjql-.com	Vicki Boone
7	csveip101@hotmail.org	Charlene Mercado
8	cxnymnd423@gmail.org	Alexander Cross
9	zhovptgj8@bphjr.sdjql-.com	Jo Hunt
10	aavicz142@hotmail.com	Jessie Galloway
11	zhovptgj8@bphjr.sdjql-.com	Marianne Ibarra
12	qgrzlspl.dtgw@kiulidgf.-olgjf.org	Lawrence Sanc...
13	ctzrdj.mxdptwsl@qfyvk.wrukpa.com	Joan Rice
14	ixpprj867@ctllsf.org	Dina Williamson
15	ixpprj867@ctllsf.org	Cornelius Pache...

הפלט (15 רשומות ראשונות):



מטלה 2 – יישומי כלים מתקדמים View

VIEW: טבלה המציגה את כלל הפריטים ושמותיהן שהוזמנו בכל אחת מהזמנות. זוהי טבלה המשמשת למעקב אחר הזמנות שבוצעו והפריטים שהוזמנו בהן.

```
--DROP VIEW V_PRODUCTS_IN_ORDERS
CREATE VIEW V_PRODUCTS_IN_ORDERS AS
SELECT      Order_ID = O.Order_ID, Order_Date = O.Date ,City = O.Address_City,
            Item_Number = P.Item_Number, Name = N.Name
FROM        PRODUCTS AS P JOIN IS_CONTAINS AS I ON P.Item_Number=I.Item_Number JOIN
            ORDERS AS O ON I.Order_ID=O.Order_ID JOIN NAMES AS N ON
            N.Item_Number= P.Item_Number
```

שימוש ב-VIEW: באילו ערים הזמינו לפחות 2 רולרבליד בשנת 2018?

```
SELECT      City, Num_Rollerbalde = COUNT ( DISTINCT Item_Number)
FROM        V_PRODUCTS_IN_ORDERS
WHERE       Name LIKE '%Skate%' AND YEAR(Order_Date) = 2018
GROUP BY    City
HAVING      COUNT(DISTINCT Item_Number) > 1
ORDER BY    Num_Rollerbalde DESC
```

הפלט (15 רשומות ראשונות):

	City	Num_Rollerbalde
1	Buffalo	5
2	Grand Rapids	4
3	Omaha	4
4	Pittsburgh	4
5	Portland	3
6	Riverside	3
7	Santa Ana	3
8	Toledo	3
9	Greensboro	3
10	Louisville	3
11	Anchorage	3
12	New York	3
13	Norfolk	3
14	Oklahoma	3
15	Arlington	2



פונקציות (Functions)

1. פונקציה המחזירה SCALAR: כמה פעמים השתמשו בפילטר מסוג מסוים בתקופה מוגדרת (קלט: סוג הפילטר, תאריך התחלה ותאריך סיום).
מטרה עסקית: ניתן להבחין כמה פילטר מסוים בשימוש, נתון המראה על נוחיות השימוש בו לצורך חיפוש ועל בסיסו ניתן לשפר או לבטל אותו.

```
--DROP FUNCTION FilterUsage
CREATE FUNCTION FilterUsage ( @Filter Varchar(20) , @DateFrom Date , @DateTo Date )
RETURNS INT
AS BEGIN
    DECLARE @TimesBeenUsed INT
    IF ( @Filter = 'Color' )
        SELECT @TimesBeenUsed = count (*)
        FROM FILTERS_COLORS AS FC
        WHERE FC.Search_DT BETWEEN @DateFrom AND @DateTo

    IF ( @Filter = 'Price' )
        SELECT @TimesBeenUsed = count (*)
        FROM FILTERS_PRICES AS FP
        WHERE FP.Search_DT BETWEEN @DateFrom AND @DateTo

    IF ( @Filter = 'Size' )
        SELECT @TimesBeenUsed = count (*)
        FROM FILTERS_SIZES AS FS
        WHERE FS.Search_DT BETWEEN @DateFrom AND @DateTo

    IF ( @Filter = 'Vendor' )
        SELECT @TimesBeenUsed = count (*)
        FROM FILTERS_VENDORS AS FV
        WHERE FV.Search_DT BETWEEN @DateFrom AND @DateTo

    IF ( @Filter = 'Best Selling' )
        SELECT @TimesBeenUsed = count (*)
        FROM SEARCHS AS S
        WHERE S.Search_DT BETWEEN @DateFrom AND @DateTo AND
        (S.Filter_Sorting = 'Best Selling')

    IF ( @Filter = 'Price, low to high' )
        SELECT @TimesBeenUsed = count (*)
        FROM SEARCHS AS S
        WHERE S.Search_DT BETWEEN @DateFrom AND @DateTo AND
        (S.Filter_Sorting = 'Price, low to high')

    IF ( @Filter = 'Price, high to low' )
        SELECT @TimesBeenUsed = count (*)
        FROM SEARCHS AS S
        WHERE S.Search_DT BETWEEN @DateFrom AND @DateTo AND
        (S.Filter_Sorting = 'Price, high to low')

    IF ( @Filter = ' Alphabetically, A-Z' )
        SELECT @TimesBeenUsed = count (*)
        FROM SEARCHS AS S
        WHERE S.Search_DT BETWEEN @DateFrom AND @DateTo AND
        (S.Filter_Sorting = ' Alphabetically, A-Z')
```



```
IF ( @Filter = 'Alphabetically, Z-A')
    SELECT      @TimesBeenUsed = count (*)
    FROM        SEARCHS AS S
    WHERE       S.Search_DT BETWEEN @DateFrom AND @DateTo AND
                (S.Filter_Sorting = 'Alphabetically, Z-A')

RETURN        @TimesBeenUsed
END
```

שימוש בפונקציה:

```
SELECT      Timeused= dbo.FilterUsage ( 'color' , '01/01/2017' , '01/01/2020 ' )
```

הפלט:

	Timeused
1	106

2. פונקציה המחזירה TABLE: מי הם הלקוחות שקנו מעל סכום מסוים ב-X חודשים האחרונים? (קלט: סכום כסף ומספר החודשים אחורה).
מטרה עסקית: איתור לקוחות ה-VIP אשר רכשו בהכי הרבה כסף בטווח הזמנים המוגדר.

```
CREATE      Function      VIPCustomers (@Amount int, @Months int)
RETURNS    TABLE
AS        RETURN
    SELECT      C.Email, FullName = C.First_Name + ' '+C.Last_Name,
                Amount= SUM (I.Quantity*P.Price)
    FROM        CUSTOMERS AS C JOIN ORDERS AS O ON C.Email=O.Email_Customer
                JOIN IS_CONTAINS AS I ON O.Order_ID=I.Order_ID JOIN
                PRODUCTS AS P ON P.Item_Number=I.Item_Number
    WHERE       Datediff (mm,O.Date, getdate())<= @Months
    GROUP BY    C.Email, C.First_Name, C.Last_Name
    HAVING      SUM (I.Quantity*P.Price) >= @Amount
```

שימוש בפונקציה:

```
SELECT      *
FROM        dbo.VIPCustomers (600,12)
ORDER BY    3 DESC
```

הפלט:

	Email	FullName	Amount
1	bziyrzyh370@hotmail.org	Kristina Wagner	897.2967
2	ahxhvo261@yhao.com	Pamela Matthews	694.5179
3	cbczebuz519@hotmail.com	Janelle Khan	632.9234
4	bigeehgy836@gmail.net	Kellie Campos	624.264



Trigger פשוט

טריגר: כאשר לקוח מבצע הזמנה יעודכן שדה ביישות PRODUCT אשר סוכם את כלל ההכנסות של החברה ממכירת מוצר ספציפי זה, ושדה שמעדכן את יחס המכירות של מוצר זה ביחס להכנסת החברה הכוללת.
מטרה עסקית: עדכון שוטף של עמודת הכנסות ויחסים על פי מוצרים בכדי לדעת אילו מוצרים מהווים את רוב מקור ההכנסות של החברה, ועל מנת לסכום בקלות את הכנסות החברה בעת הצורך.

הטריגר:

```
--DROP TRIGGER Set_Ratio
CREATE TRIGGER Set_Ratio
ON IS_CONTAINS
FOR INSERT

AS
UPDATE PRODUCTS
SET
    TotalIncome = (SELECT SUM (I.Quantity* P.Price)
                    FROM PRODUCTS AS P JOIN IS_CONTAINS AS I
                    ON P.Item_Number= I.Item_Number
                    WHERE P.Item_Number = PRODUCTS.Item_Number )

UPDATE PRODUCTS
SET
    [Ratio (%)] = (SELECT SUM (I.Quantity* P.Price) * 100
                    FROM PRODUCTS AS P JOIN IS_CONTAINS AS I
                    ON P.Item_Number = I.Item_Number
                    WHERE P.Item_Number = PRODUCTS.Item_Number) /
    (SELECT SUM (I.Quantity* P.Price)
     FROM PRODUCTS AS P JOIN IS_CONTAINS AS I ON
     P.Item_Number = I.Item_Number)
```

שאלת הרצה מקדימה:

```
--ALTER TABLE PRODUCTS DROP COLUMN TotalIncome
ALTER TABLE PRODUCTS
ADD TotalIncome real

--ALTER TABLE PRODUCTS DROP COLUMN Ratio
ALTER TABLE PRODUCTS
ADD [Ratio (%)] real
UPDATE PRODUCTS
SET
    TotalIncome=(SELECT SUM (I.Quantity* P.Price)
                  FROM PRODUCTS AS P JOIN IS_CONTAINS AS I ON P.Item_Number=I.Item_Number
                  WHERE P.Item_Number = PRODUCTS.Item_Number )

UPDATE PRODUCTS
SET
    [Ratio (%)]=(SELECT SUM (I.Quantity* P.Price) * 100
                  FROM PRODUCTS AS P JOIN IS_CONTAINS AS I ON P.Item_Number=I.Item_Number
                  WHERE P.Item_Number = PRODUCTS.Item_Number) /
    (SELECT SUM (I.Quantity* P.Price)
     FROM PRODUCTS AS P JOIN IS_CONTAINS AS I ON
     P.Item_Number=I.Item_Number)
```



שימוש בטריגר:

```
SELECT *
FROM PRODUCTS
WHERE Item_number = 10120
```

```
INSERT INTO ORDERS (Order_ID, First_name, Last_name, Email_order, Date , Delivery_method
, shipping_rate , phone , company , Address_Apartment ,Address_Street, Address_City
,Address_State ,Address_Country, Address_Zip_Code, Email_customer , cc_number,
Address_Number)
VALUES (888888, 'Doili', 'Soli' , 'DOLisoli@gmail.com' , '2021-05-28', 'P'
, 0, 05642647353, 'icx', 2, 'rager', 'Norfolk', 'california', 'USA', 77777, NULL ,
'3492555390354670', NULL)
```

```
INSERT INTO IS_CONTAINS (Order_ID,Item_Number ,Quantity )
VALUES(888888 ,10120, 1 )
```

פלט לפני:

	Item_Number	Price	Model	Size	Color	Vendor	TotalIncome	Ratio (%)
1	10120	65.9691	O65B4J	7	white	Chanpion	461.7837	0.5362

פלט אחרי:

	Item_Number	Price	Model	Size	Color	Vendor	TotalIncome	Ratio (%)
1	10120	65.9691	O65B4J	7	white	Chanpion	527.7528	0.6124



פרוצדורה שמורה (Stored Procedure) פשוטה

פרוצדורה: מהם עשרת המוצרים הכי נמכרים בתקופה מוגדרת? (קלט: תאריך התחלה ותאריך סיום).

מטרה עסקית: איתור המוצרים הנמכרים ביותר אשר ישמרו בטבלה ייעודית BEST SELLERS אשר תשמש את החברה לעדכון של הפילטר BEST SELLING באתר.

הפרוצדורה:

```
--DROP PROCEDURE SP_BestSellers
CREATE PROCEDURE SP_BestSellers    @DateFrom Date ,    @DateTo Date
AS BEGIN
    IF (SELECT Object_ID ('BEST_SELLERS')) IS NOT NULL DROP TABLE BEST_SELLERS

    CREATE TABLE BEST_SELLERS (
        Item_Number      Int      PRIMARY KEY    NOT NULL,
        Name              VARCHAR (100)          NOT NULL,
        NumberOfSales     Int                        NOT NULL )

    INSERT INTO    BEST_SELLERS

    SELECT TOP 10  P.Item_Number,N.Name, NumberOfSales= SUM(Distinct I.Quantity)
    FROM          PRODUCTS AS P JOIN IS_CONTAINS AS I ON P.Item_Number= I.Item_Number
                JOIN ORDERS AS O ON O.Order_ID = I.Order_ID
                JOIN NAMES AS N ON P.Item_number = N.Item_number

    WHERE        O.Date BETWEEN @DateFrom AND @DateTo
    GROUP BY     P.Item_number, N.Name
    ORDER BY     NumberOfSales DESC
END
```

שימוש בפונקציה:

```
EXECUTE SP_BestSellers    '01/01/2017' , '01/01/2020'
```

שאלת הרצה:

```
SELECT *
FROM BEST_SELLERS
ORDER BY NumberOfSales DESC
```

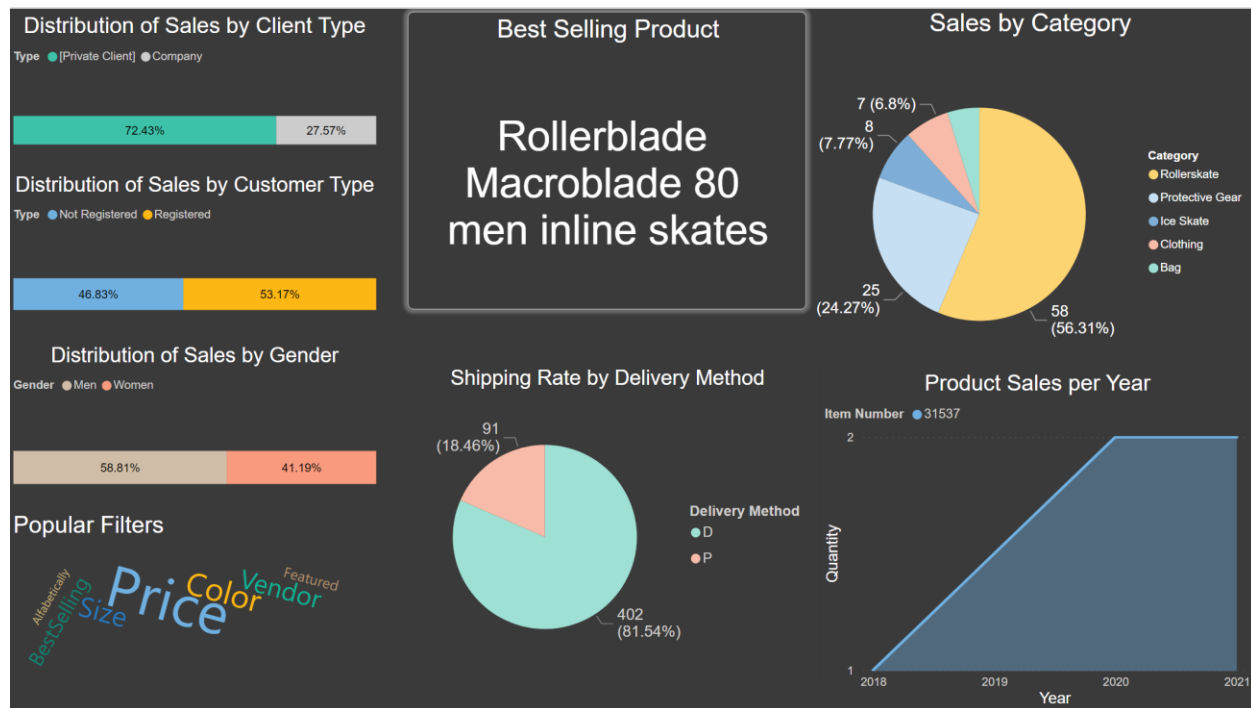
הפלט:

	Item_Number	Name	NumberOfSales
1	92542	Rollerblade Macroblade 80 men inline skates	13
2	96885	Rollerblade Macroblade 81 women inline skates	11
3	10647	Roces RC2 unisex rollerskates	10
4	30511	Rollerblade Macroblade 100 3WD women inline skates	9
5	57748	Rollerblade Macroblade 81 men inline skates	9
6	60602	Roces chuck unisex rollerskates	8
7	99833	Rollerblade 76 mm/80A +SG5 inline skate wheel	8
8	16364	Riedell soar girl figure skates	8
9	25016	Rollerblade Zetrablade elite men inline skates	7
10	50136	Rollerblade inline skates BREAK PAD	7

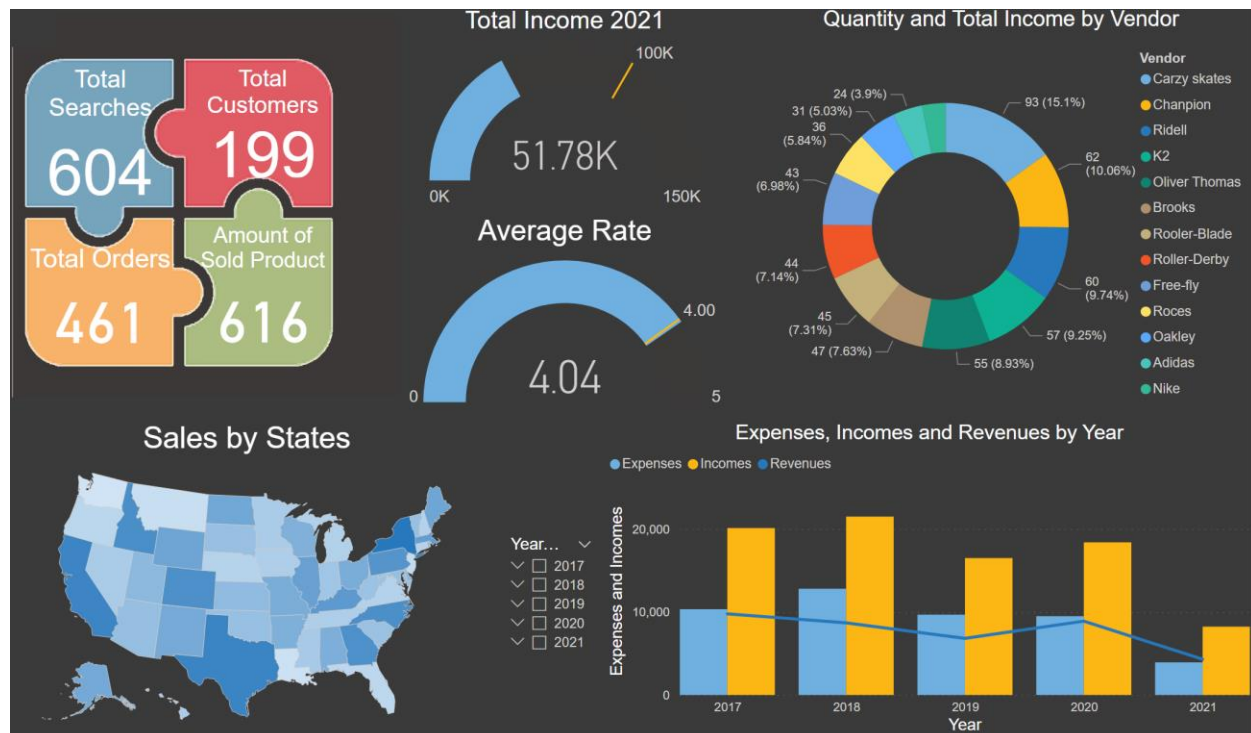


מטלה 3 (20%) – כלים להצגת נתונים

דו"ח עסקי – דו"ח עבור מחלקת מכירות של החברה:



לוח מחוונים:





מטלה 4 – יישומים מתקדמים

Trigger העושה שימוש בסמן (Cursor)

טריגר: בעת ביצוע הזמנה חדשה על ידי לקוח רשום, אם כתובת המשלוח של ההזמנה שבוצעה אינה נמצאת בספר הכתובות של הלווח, נוסף אותה. מטרה עסקית: נרצה להקל על הלקוחות הרשומים בעת ביצוע הזמנות חוזרות, ולאפשר להם להשתמש בכתובות שכבר הזינו, גם אם לא שמרו אותן בעצמן בספר הכתובות שלהם.

```
-- DROP TRIGGER Set_Address
CREATE TRIGGER Set_Address
ON ORDERS
FOR INSERT, UPDATE
AS
    --- הגדרת משתנים של אינסטרטד
    DECLARE @Order_ID int
    DECLARE @First_Name Varchar (20)
    DECLARE @Last_Name Varchar (20)
    DECLARE @Email_Order Varchar (40)
    DECLARE @Date Date
    DECLARE @Delivery_Method Char (1)
    DECLARE @Shipping_Rate smallmoney
    DECLARE @Phone Varchar (10)
    DECLARE @Company Varchar (10)
    DECLARE @Address_Apartment Varchar (5)
    DECLARE @Address_Street Varchar (20)
    DECLARE @Address_City Varchar (20)
    DECLARE @Address_State Varchar (20)
    DECLARE @Address_Country Varchar (20)
    DECLARE @Address_Zip_Code Varchar (20)
    DECLARE @Email_Customer Varchar (40)
    DECLARE @Address_Number tinyint
    DECLARE @CC_Number Varchar (16)

    --- לולאה של אינסטרטד
    DECLARE INSERTED_CURSOR CURSOR FOR
    SELECT Order_ID, First_Name, Last_Name, Email_Order, Date, Delivery_Method,
    Shipping_Rate, Phone, Company, Address_Apartment, Address_Street, Address_City,
    Address_State, Address_Country, Address_Zip_Code, Email_Customer, Address_Number,
    CC_Number
    FROM INSERTED
    WHERE (Email_Customer IS NOT NULL) AND (Address_Number IS NULL)

    BEGIN
    OPEN INSERTED_CURSOR
    FETCH NEXT FROM INSERTED_CURSOR INTO @Order_ID, @First_Name, @Last_Name, @Email_Order,
    @Date, @Delivery_Method, @Shipping_Rate, @Phone, @Company, @Address_Apartment,
    @Address_Street, @Address_City, @Address_State, @Address_Country, @Address_Zip_Code,
    @Email_Customer, @Address_Number, @CC_Number

    DECLARE @X int
    SET @X = (select COUNT (DISTINCT Order_ID) FROM inserted)

    WHILE (@X>0) -- עבור שורה אחת באינסטרטד
    BEGIN
```



```
DECLARE @Email          Varchar (40)
DECLARE @Number          tinyint
DECLARE @Apartment       Varchar (5)
DECLARE @Street          Varchar (20)
DECLARE @City Varchar (20)
DECLARE @State           Varchar (20)
DECLARE @Country         Varchar (20)
DECLARE @Zip_Code       Varchar (20)

DECLARE ADDRESSE_CURSOR CURSOR FOR
SELECT Email, Number, Apartment, Street, City, State, Country, Zip_Code
FROM ADDRESSES
WHERE Email = @Email_Customer

BEGIN
OPEN ADDRESSE_CURSOR
FETCH NEXT FROM ADDRESSE_CURSOR INTO @Email, @Number, @Apartment, @Street,
@City, @State, @Country, @Zip_Code

DECLARE @I          int
SET @I = 0

DECLARE @Y int
SET @Y = (select COUNT (*) FROM ADDRESSES WHERE Email = @Email_Customer)

WHILE (@Y>0)
BEGIN
IF ((@Email_Customer = @Email) AND (@Address_Number IS NULL) AND
(@Address_Apartment = @Apartment) AND (@Address_Street = @Street) AND
(@Address_City = @City) AND (@Address_State = @State) AND
(@Address_Country = @Country) AND (@Address_Zip_Code = @Zip_Code)) SET @I= 1

FETCH NEXT FROM ADDRESSE_CURSOR INTO @Email, @Number, @Apartment, @Street,
@City, @State, @Country, @Zip_Code
SET @Y = @Y -1
END -- END BEGIN

IF ((@Email_Customer = @Email) AND (@Address_Number IS NULL) AND (@I = 0))
INSERT INTO ADDRESSES (Email, Number, Apartment, Street, City, State,
Country, Zip_Code)
VALUES (@Email_Customer, 1 + (SELECT COUNT (*)
FROM ADDRESSES
WHERE Email = @Email_Customer),
@Address_Apartment , @Address_Street, @Address_City, @Address_State,
@Address_Country, @Address_Zip_Code)

CLOSE ADDRESSE_CURSOR
DEALLOCATE ADDRESSE_CURSOR
END -- END BEGIN

SET @X = @X -1
FETCH NEXT FROM INSERTED_CURSOR INTO @Order_ID, @First_Name, @Last_Name, @Email_Order,
@Date, @Delivery_Method, @Shipping_Rate, @Phone, @Company, @Address_Apartment,
@Address_Street, @Address_City, @Address_State, @Address_Country, @Address_Zip_Code,
@Email_Customer, @Address_Number ,@CC_Number
```



```
END -- END BEGIN
CLOSE INSERTED_CURSOR
DEALLOCATE INSERTED_CURSOR
END -- END BEGIN
```

שאלות הרצה:

```
INSERT INTO ORDERS (Order_ID, First_name, Last_name, Email_order, Date , Delivery_method
, shipping_rate , phone , company , Address_Apartment ,Address_Street, Address_City
,Address_State ,Address_Country, Address_Zip_Code, Email_customer ,cc_number ,
Address_Number )
VALUES (9929, 'Sarah', 'Marshall', 'bglartr503@gmail.net', '2021-05-28', 'P'
,0,05642647353,'icx', 2,'rager', 'Norfolk', 'california', 'USA',
77777, 'bglartr503@gmail.net', '349255390354670', NULL)
```

```
select *
from ADDRESSES
where Email='bglartr503@gmail.net'
```

פלט לפני:

	Email	Number	Apartment	Street	City	State	Country	Zip_Code
1	bglartr503@gmail.net	1	AOVQ5	Old Road	Anchorage	Maine	USA	41003

פלט אחרי:

	Email	Number	Apartment	Street	City	State	Country	Zip_Code
1	bglartr503@gmail.net	1	AOVQ5	Old Road	Anchorage	Maine	USA	41003
2	bglartr503@gmail.net	2	2	rager	Norfolk	california	USA	77777

שאלות עסקיות המשלבות Window Functions

שאלתה ראשונה – שימוש ב PRECENT RANK,RANK : שאלתה המדרגת את מידת הכנסתם ביחס למוצרים אחרים בחברה.

מטרה עסקית: הבנה מיהם המוצרים המשפיעים ביותר על הכנסות החברה ("מודל-2080").

```
-- DROP VIEW V_ORDER_IN_PRODUCTS
CREATE VIEW V_ORDER_IN_PRODUCTS AS
SELECT P.Item_Number, N.Name ,[Total Income] = SUM (P.Price * I.Quantity)
FROM ORDERS AS O JOIN IS_CONTAINS AS I ON O.Order_ID=I.Order_ID JOIN PRODUCTS AS P ON
P.Item_Number=I.Item_Number JOIN NAMES AS N ON N.Item_Number=P.Item_Number
GROUP BY P.Item_Number, N.Name
```

```
-- DROP VIEW V_ranktot
CREATE VIEW V_ranktot
AS
SELECT
Item_Number,Name, [Total Income], RANK () OVER (ORDER BY [Total Income] DESC ) [Rank],
[Income Percent] = [Total Income] / (select sum([Total Income]) from V_ORDER_IN_PRODUCTS)
FROM V_ORDER_IN_PRODUCTS
```



```
-- DROP FUNCTION TOTAL_PERCENT
CREATE FUNCTION TOTAL_PERCENT (@RANK1 INT)

RETURNS REAL
AS
BEGIN

DECLARE @R INT
SET @R = @RANK1
DECLARE @TOTAL_PERCENT REAL
SET @TOTAL_PERCENT = 0

WHILE (@R > 0)
BEGIN
SET @TOTAL_PERCENT = @TOTAL_PERCENT + ISNULL((SELECT [Income Percent] from V_ranktot
where [Rank] = @R) ,0)
SET @R = @R -1
END

RETURN @TOTAL_PERCENT
END
```

שאלת הרצה:

```
SELECT Item_Number, Name, [Total Income], [Rank] , [Precent Rank]=( (1/cast((select
count(*) from V_ORDER_IN_PRODUCTS) as real)) * [Rank] ),
[Income Percent] , [Total Percent] =
CAST(dbo.TOTAL_PERCENT([Rank]) AS float)

FROM V_ranktot as v1
```

פלט (15 רשומות ראשונות):

	Item_Number	Name	TOTAL	rank_no	percent_rank	Income_Percent	Total_Percent
1	82826	Roces kolossal unisex rollerskates	2668.9509	1	0	0.0315	0.0315000005066395
2	30511	Rollerblade Macroblade 100 3WD women inline skates	2104.8006	2	0.00980392156862745	0.0249	0.0564000010490417
3	60602	Roces chuck unisex rollerskates	1990.4868	3	0.0196078431372549	0.0235	0.0798999965190887
4	86226	Roces aces unisex rollerskates	1897.2488	4	0.0294117647058824	0.0224	0.102300003170967
5	75256	Riedell pearl girls figure skates	1802.668	5	0.0392156862745098	0.0213	0.123599991202354
6	50136	Rollerblade inline skates BREAK PAD	1738.122	6	0.0490196078431373	0.0205	0.144099995493889
7	69212	Sure grip prism motion indoor rollerskates unisex roll...	1700.3992	7	0.0588235294117647	0.0201	0.164199993014336
8	45221	Rollerblade moonbeams led 72/82A inline skate wheel	1593.6318	8	0.0686274509803922	0.0188	0.18299999833107
9	92542	Rollerblade Macroblade 80 men inline skates	1582.7574	9	0.0784313725490196	0.0187	0.201700001955032
10	17592	Rollerblade Macroblade 70 3WD men inline skates	1580.7222	10	0.0882352941176471	0.0187	0.220400005578995
11	40885	Rollerblade Macroblade 70 3WD women inline skates	1565.6952	11	0.0980392156862745	0.0185	0.238900005817413
12	99833	Rollerblade 76 mm/80A +SG5 inline skate wheel	1550.7664	12	0.107843137254902	0.0183	0.257200002670288
13	15557	Jackson finesse 50 girls figure skates	1509.7061	13	0.117647058823529	0.0178	0.275000005960464
14	33086	Free fly breeze 6in mens shorts	1495.4945	14	0.127450980392157	0.0176	0.292600005865097
15	72726	Sure grip prism motion outdoor rollerskates unisex ro...	1482.222	15	0.137254901960784	0.0175	0.310100018978119



שאלתה שנייה – שימוש ב CUME_DIST, NTILE חלוקת המדינות על פי הכנסותיהם לאחוזונים ושלישים.

מטרה עסקית: על מנת לדעת באילו מדינות היקף המכירות הינו גבוה/נמוך, בכדי לנקוט צעדים עסקיים בתחום הפרסום שיווק וכו'.

```
SELECT State=O.Address_State, [Total Price ] = SUM (I.Quantity * P.Price ) ,
ROUND(CUME_DIST() OVER ( ORDER BY SUM (I.Quantity * P.Price ) DESC),2) [Top] ,
NTILE (3) OVER (ORDER BY SUM (I.Quantity * P.Price ) DESC) [Third] -- 1 Top Third , 2
Middle Third , 3 Lower Third
FROM ORDERS AS O JOIN IS_CONTAINS AS I ON O.Order_ID = I.Order_ID JOIN PRODUCTS AS P ON
P.Item_Number = I.Item_Number
GROUP by O.Address_State
order by O.Address_State
```

פלט (15 רשומות ראשונות):

	State	Total Price	Top	Third
1	Alabama	1894.0443	0.42	2
2	Alaska	2342.7052	0.24	1
3	Arizona	1454.2926	0.5	2
4	Arkansas	1987.1989	0.38	2
5	California	3273.1845	0.06	1
6	Colorado	3053.1481	0.12	1
7	Connecticut	748.4967	0.88	3
8	Delaware	1706.8567	0.44	2
9	Florida	611.2212	0.92	3
10	Georgia	3035.994	0.14	1
11	Hawaii	1005.6847	0.72	3
12	Idaho	3121.1086	0.08	1
13	Illinois	2475.8576	0.22	1
14	Indiana	1204.4474	0.66	2
15	Iowa	929.9459	0.76	3



שאלתה שלישית – שימוש ב LAG: חישוב הגידול בהכנסות החברה בכל חודש של שנה נתונה ביחס לחודש הקודם לו.
מטרה עסקית: על ידי שימוש בנתונים אלו, נוכל להבין מגמות קנייה של הצרכנים וביצוע תחזיות לחודשים הבאים.

```
-- DROP VIEW V_ORDER1_IN_PRODUCTS
CREATE VIEW V_ORDER1_IN_PRODUCTS AS
SELECT yy=YEAR(o.Date), mm =MONTH(o.date) ,TOTAL = SUM (P.Price * I.Quantity)
FROM ORDERS AS O JOIN IS_CONTAINS AS I ON O.Order_ID=I.Order_ID JOIN PRODUCTS AS P ON
P.Item_Number=I.Item_Number
GROUP BY YEAR(o.date), MONTH(o.Date)

SELECT [Year]=vvv.yy,[Month]=vvv.mm,[This Month]=TOTAL ,
LAG(TOTAL,1) OVER (ORDER BY yy,mm) [Previous Month],
[Growth Rate] = ROUND((TOTAL/cast(LAG(TOTAL,1) OVER (ORDER BY yy,mm) as real )),2)

FROM V_ORDER1_IN_PRODUCTS as vvv
ORDER BY 1,2
```

פלט (15 שורות):

	Year	Month	This Month	Previous Month	Growth Rate
1	2017	1	1434.7674	NULL	NULL
2	2017	2	1384.8911	1434.7674	0.97
3	2017	3	322.1563	1384.8911	0.23
4	2017	4	2640.6369	322.1563	8.2
5	2017	5	2529.6095	2640.6369	0.96
6	2017	6	2682.4756	2529.6095	1.06
7	2017	7	904.3729	2682.4756	0.34
8	2017	8	950.7438	904.3729	1.05
9	2017	9	1501.4416	950.7438	1.58
10	2017	10	1405.7894	1501.4416	0.94
11	2017	11	1287.7991	1405.7894	0.92
12	2017	12	3276.2087	1287.7991	2.54
13	2018	1	2205.6881	3276.2087	0.67
14	2018	2	1024.5033	2205.6881	0.46
15	2018	3	1339.4471	1024.5033	1.31



דו"ח המושתת על שאלתה מקוננת מורכבת

שאלתה: איתור המוצרים הנמכרים ביותר תוך מתן הסבר מפורט על שמם, תאריך החיפוש האחרון שבוצע עבורם, תאריך הקנייה האחרון בו רכשו את המוצר, דירוג ממוצע, מדד אמינות הדירוג*, מספר שאלות הנשאלו לגבי אותו מוצר, מספר הזמנות וכמות הפריטים הנמכרו כמה מלקוחותינו קנו ממנו הכנסותיו והאם היה Best Seller. **מטרה עסקית:** דו"ח המאחד את כלל המדדים הרלוונטים עבור כלל המוצרים בחברה.

*אמינות הדירוג = $\frac{\text{מספר האנשים המדרגים}}{\text{דירוג ממוצע}}$ אמינות הדירוג. ככל שיחס המדד גדול יותר ככה הדירוג אמין יותר.

```
-- DROP VIEW V_OrderAndQAndP
CREATE VIEW V_OrderAndQAndP AS
SELECT O.Order_ID ,I.Item_Number,O.Email_Customer, O.Email_Order, O.Date ,I.Quantity
FROM ORDERS AS O JOIN IS_CONTAINS AS I ON O.Order_ID = I.Order_ID

SELECT P.Item_Number ,N.Name,
      [Last Search] = (SELECT (Max (F.Search_DT))
                      FROM FINDS AS F
                      WHERE F.Item_Number= P.Item_Number ) ,
      [Last Sell] = (SELECT (Max (v.Date))
                    FROM V_OrderAndQAndP as v
                    WHERE v.Item_Number= P.Item_Number ) ,
      [Avrage Rate] = (SELECT AVG (R.Rate)
                      FROM REVIEWS AS R
                      WHERE P.Item_Number = R.Item_Number ) ,
      [Rate Credabilty]= (SELECT (COUNT(DISTINCT R.DT)/CAST(AVG (R.Rate)AS REAL))
                        FROM REVIEWS AS R
                        WHERE P.Item_Number = R.Item_Number ) ,
      Questions = (SELECT COUNT( DISTINCT Q.DT )
                  FROM QUESTIONS AS Q
                  WHERE Q.Item_Number =P.Item_Number ) ,
      Orders = (SELECT TOT=COUNT (distinct V.Order_ID)
                FROM V_OrderAndQAndP AS V
                WHERE V.Item_Number = P.Item_Number ) ,
      [Products Sold] = (SELECT TOT =SUM(V.Quantity)
                        FROM V_OrderAndQAndP AS V
                        WHERE V.Item_Number = P.Item_Number),
      [Customers Bought] = (SELECT Count (Distinct V.Email_Order)
                          FROM V_OrderAndQAndP AS V
                          WHERE V.Item_Number = P.Item_Number ) ,
      Income = (SELECT SUM (I.quantity * P1.Price )
                FROM IS_CONTAINS AS I JOIN PRODUCTS AS P1 ON
                  I.Item_Number=P1.Item_Number
                WHERE I.Item_Number = P.Item_Number ) ,
      [Best Selling]= ( CASE WHEN (P.Item_Number IN (SELECT B.Item_Number
                                                    FROM BEST_SELLERS AS B))
                            THEN 'Yes' ELSE 'No' END

FROM PRODUCTS AS P JOIN NAMES AS N ON P.Item_Number = N.Item_Number
GROUP BY P.Item_Number , N.Name
ORDER BY Income DESC
```



פלט (15 שורות):

	Item Number	Name	Last Search	Last Sell	Average Rate	Rate Credability	Questions	Orders	Products Sold	Customers Bought	Income	Best Selling
1	82826	Roces kolossal unisex rollerskates	2020-04-10 20:15:00.000	2021-01-21	5	0.2	3	7	11	7	2668.9509	No
2	30511	Rollerblade Macroblade 100 3WD women inline skates	2020-05-04 06:16:00.000	2021-03-11	2	0.5	3	10	14	10	2104.8006	Yes
3	60602	Roces chuck unisex rollerskates	2020-05-11 09:26:00.000	2020-05-02	3	0.3333333	3	7	9	7	1990.4868	Yes
4	86226	Roces aces unisex rollerskates	2021-05-27 06:18:00.000	2021-02-01	5	0.2	3	6	8	6	1897.2488	No
5	75256	Riedell pearl girls figure skates	2019-11-19 01:10:00.000	2021-04-13	5	0.2	3	7	10	7	1802.668	No
6	50136	Rollerblade inline skates BREAK PAD	2019-11-29 10:43:00.000	2021-04-30	5	0.2	2	9	10	9	1738.122	Yes
7	69212	Sure grip prism motion indoor rollerskates unisex roll...	2020-10-21 09:41:00.000	2021-02-10	5	0.2	3	6	8	6	1700.3992	No
8	45221	Rollerblade moonbeams led 72/82A inline skate wheel	2020-07-24 01:02:00.000	2021-01-16	4	0.25	1	4	6	4	1593.6318	No
9	92542	Rollerblade Macroblade 80 men inline skates	2021-03-26 06:28:00.000	2020-04-06	5	0.2	2	10	14	10	1582.7574	Yes
10	17592	Rollerblade Macroblade 70 3WD men inline skates	2020-04-30 08:36:00.000	2020-09-28	4	0.25	3	4	6	4	1580.7222	No
11	40885	Rollerblade Macroblade 70 3WD women inline skates	2020-08-10 01:04:00.000	2020-12-17	4	0.25	3	4	6	4	1565.6952	No
12	99833	Rollerblade 76 mm/80A +SG5 inline skate wheel	2019-12-21 01:17:00.000	2019-10-10	NULL	NULL	1	4	8	4	1550.7664	Yes
13	15557	Jackson finesse 50 girls figure skates	2020-06-20 16:24:00.000	2020-07-26	5	0.2	3	6	7	6	1509.7061	No
14	33086	Free fly breeze 6in mens shorts	2021-01-30 17:20:00.000	2021-03-03	5	0.2	1	3	5	3	1495.4945	No
15	72726	Sure grip prism motion outdoor rollerskates unisex ro...	2020-12-10 19:03:00.000	2020-12-09	4	0.25	4	5	7	5	1482.222	No

בהצלחה!