


פרויקט בסיסי נתונים – חלק ג'

יפה





ההנחות שהונחו לERD בחלק ב':

מרכיב	הנחות	הערות
קשר רקורסיבי יחיד-רבים של ישות Category	מימוש של מרכיב sub-category באתר. לכל קטגוריה אב יחיד לכל היותר, לכל קטגוריה עד N בנים.	
המזהה החלקי "Name" Album	ביכולת הלקוח ליצור ולתחזק מספר אלבומים, ולכל אלבום נאכף שם ייחודי.	
השדה [Interests] Customer	תחומי עניין מוגדרים אישית עבור הלקוח, מתעדכן ונעשה בו שימוש ע"י האלגוריתם של מערכת ההמלצות שמפעיל האתר.	
הקשר בין ישויות Product ל Category	כל מוצר שייך לקטגוריה אחת בלבד. לא נמצאה באתר דוגמה נגדית לכך.	בשונה מההגדרה כקשר רבים-רבים בחלק א
קשר רבים-רבים בין ישויות Customer ל Billing Information בקרדינליות 0:N, 0:N	<ul style="list-style-type: none"> לקוחות שונים רשאים להשתמש בכרטיס אשראי וכתובת מגורים זהה. כל לקוח רשאי להגדיר מספר חשבונות תשלום. כל לקוח רשאי לבצע רכישות, גם אם מגדיר את החשבון במעמד הקנייה (checkout) בלבד. 	
המזהה החלקי DT_Updated Project	באתר אפשרי ליצור פרויקט  זההים בשם, וכן מתאפשר לבצע מיון לאוטף הפרויקטים לפי recent/oldest בלבד.	
קשר יחיד-יחיד בין ישויות Order ל Project	נניח כי האתר עושה שימוש בנתוני הפרויקטים שהובילו להזמנות, בעיקר למען שדרוג מערכת הפרויקטים.	נשקל לייצג קשר זה כרבים-רבים, אך הרעיון מנח, למען הפשטות של בסיס הנתונים, וכיוון שקיימים במודל 3 קשרי רבים-רבים אחרים.

-2
(1)

ההערות שניתנו:

1. נראה שמדובר בקשר מיותר, ניתן לקשר תמונה ללקוח שהעלה אותה דרך האלבום.
2. בהנחה הזאת אתם די מערערים את המודל. הקשר בין הזמנה לבין הפרויקט הוא לא רק לשדרוג מערכת אלא כדי לייצג את המוצר המותאם שנרכש בהזמנה. אם נקשר רק למוצר לא נקבל שום פרט על התמונות וההתאמות שנכללו בעיצוב. ומכאן בעייתיות גם בקרדינליות, אם הזמנתי 5 מוצרים ורק פרויקט אחד - קיבלתי מוצר אחד מותאם בלבד מתוכם. הדבר הסביר יותר לעשות הוא לוותר על הקשר בין המוצר לבין ההזמנה ולשנות את הקשר בין הפרויקט להזמנה לרבים לרבים. את המידע לגבי המוצר שהוזמן ניתן לקחת מהפרויקט כי לכל פרויקט רק מוצר אחד.

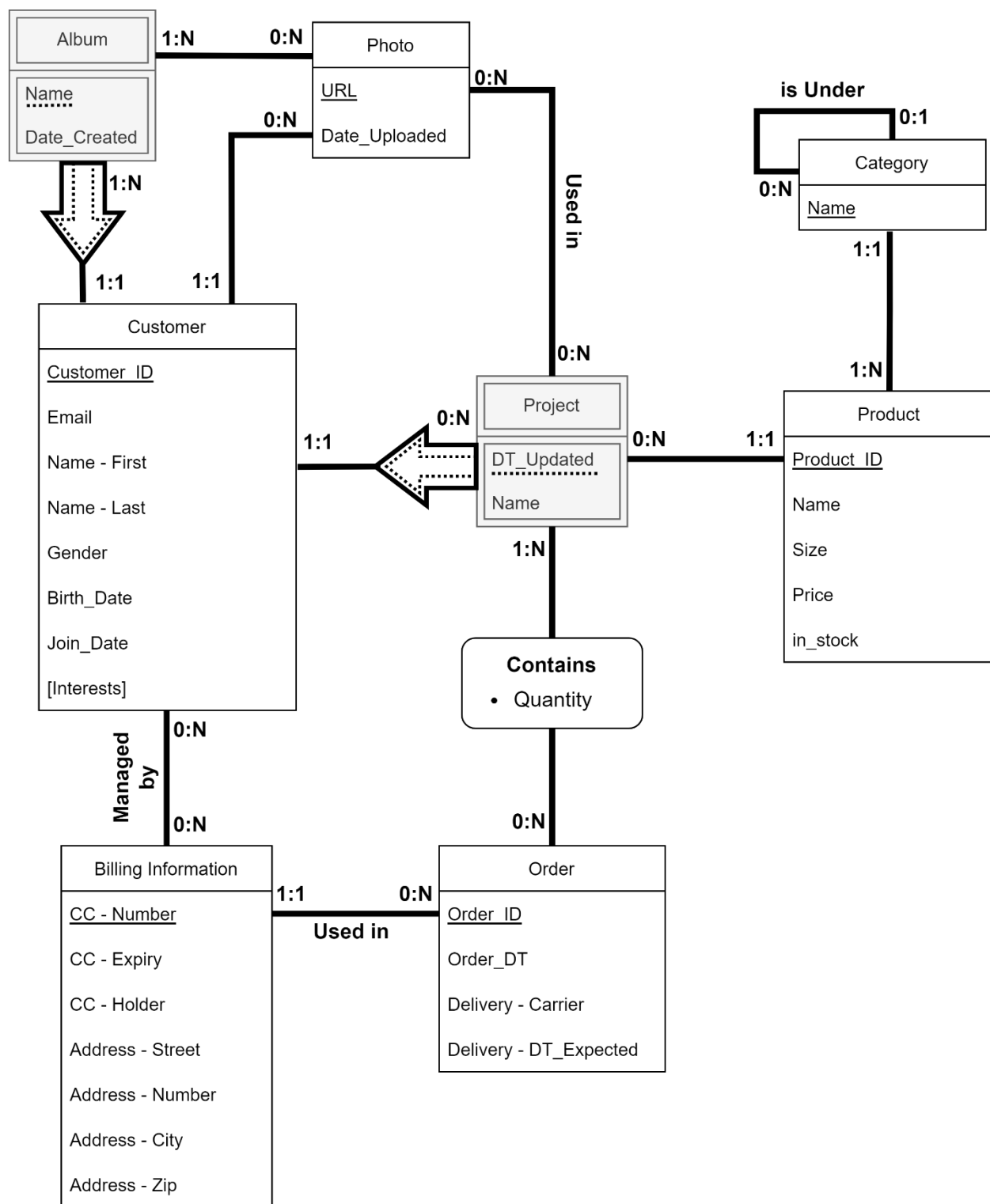


המודל הטבלאי בחלק ב':

- **CUSTOMERS** (Customer ID, Email, First_Name, Last_Name, Gender, Birth_Date, Join_Date)
- **INTERESTS** (Customer ID (CUSTOMERS), Interest)
- **ALBUMS** (Customer ID (CUSTOMERS), Name, Date_Created)
- **PHOTOS** (URL, Date_Uploaded, Customer ID (CUSTOMERS))
- **ALBUM_PHOTO** ({Customer ID, Album Name} (ALBUMS), Photo URL (PHOTOS))
- **CATEGORIES** (Name, Parent_Category (CATEGORIES))
- **PRODUCTS** (Product ID, Name, Size, Price, in_stock, Category (CATEGORIES))
- **PROJECTS** (Customer ID (CUSTOMERS), DT_Updated, Name, Product ID (PRODUCTS))
- **PROJECT_PHOTO** ({Customer, Project_DT} (PROJECTS), Photo URL (PHOTOS))
- **BILLINGS** (CC Number, CC_Expiry, CC_Holder, Street Name, Street Number, City, Zip)
- **CUSTOMER_BILLING** (Customer ID (CUSTOMERS), Credit Card (BILLINGS))
- **ORDERS** (Order ID, Order_DT, Carrier, DT_Expected, {Customer, Project_DT} (PROJECTS), Credit Card (BILLINGS))
- **CONTENTS** (Product (PRODUCTS), Order (ORDERS), Quantity)



מודל ה-ERD המתוקן לחלק זה:





ההנחות למודל זה:

מרכיב	הנחות	הערות
קשר רקורסיבי יחיד-רבים של ישות Category	מימוש של מרכיב sub-category באתר. לכל קטגוריה אב יחיד לכל היותר, לכל קטגוריה עד N בנים.	
הקשר בין ישויות Product ל Category	כל מוצר שייך לקטגוריה אחת בלבד. לא נמצאה באתר דוגמה נגדית לכך.	
המזהה החלקי "Name" בישות Album החלשה	ביכולת הלקוח ליצור ולתחזק מספר אלבומים, ולכל אלבום נאכף שם ייחודי.	
קשר רבים-רבים בין ישות Photo ל Album	לכל תמונה, שיוך אפשרי למספר אלבומים שפותח המשתמש.	
קרדינליות 1:1 בקשר שבין ישות Album ל Customer.	קיים תמיד גם אלבום אחד נסתר מהמשתמש, שנקראו לו Default. אלבום זה נוצר בתאריך הצטרפותו של המשתמש. אלבום זה מכיל אך ורק את התמונות שלא הוקצו לאף אלבום. מרכיב זה התמונות שלא הוקצו לאלבומים, מוצגות למשתמש באתר בחלונית All Photos.	
השדה [Interests] בישות Customer	תחומי עניין מוגדרים אישית עבור הלקוח, מתעדכן ונעשה בו שימוש ע"י האלגוריתם של מערכת ההמלצות שמפעיל האתר.	
המזהה החלקי DT_Updated בישות Project החלשה	באתר אפשרי ליצור פרויקטים הוהים בשם, וכן מתאפשר לבצע מיון לאוסף הפרויקטים לפי recent/oldest בלבד.	
קשר רבים-רבים בין ישות Project ל Order	הזמנה מסוימת יכולה להכיל מספר פרויקטים, וכל פרויקט כ-Quantity פעמים. פרויקט מסוים (מוצלח) יוכל להיות מוזמן שוב ושוב.	נניח כי בנוסף נעשה שימוש מיוחד בנתוני הפרויקטים שהובילו להזמנות, למען שדרוג מערכת הפרויקטים.
קשר רבים-רבים בין ישויות Customer ל Billing Information בקרדינליות 0:N, 0:N	<ul style="list-style-type: none"> לקוחות שונים רשאים להשתמש בכרטיס אשראי וכתובת מגורים זהה. כל לקוח רשאי להגדיר מספר חשבונות תשלום. כל לקוח רשאי לבצע רכישות, גם אם מגדיר את החשבון במעמד הקנייה (checkout) בלבד. 	

תיקונים שהתבצעו:

- הוזז קשר הרבים-רבים של פריטי ההזמנה (המייצגת הטבלה CONTENTS) אל בין הישויות Project Order. זאת בין היתר על מנת לאפשר ללקוח להזמין מחדש את אותם הפרויקטים המוצלחים עליהם עבד בעבר, ומבלי ליצור מופעים חדשים שלהם בבסיס הנתונים.

****** היה ניסיון לוותר על הקשר יחיד-רבים בין Photo ל Customer. אך בלעדי המפתח הזר בצד התמונה, היה מאתגר לאלץ את טבלת ALBUM_PHOTO להכיל תמונות זהות באלבומים שונים, בתנאי שהם של אותו המשתמש. ועל כן החזרנו קשר זה.



המודל הטבלאי (ללא שינויים):

- **CUSTOMERS** (Customer ID, Email, First_Name, Last_Name, Gender, Birth_Date, Join_Date)
- **INTERESTS** (Customer ID (CUSTOMERS), Interest)
- **ALBUMS** (Customer ID (CUSTOMERS), Name, Date_Created)
- **PHOTOS** (URL, Date_Uploaded, Customer_ID (CUSTOMERS))
- **ALBUM_PHOTO** ({Customer ID, Album Name} (ALBUMS), Photo URL (PHOTOS))
- **CATEGORIES** (Name, Parent_Category (CATEGORIES))
- **PRODUCTS** (Product ID, Name, Size, Price, in_stock, Category (CATEGORIES))
- **PROJECTS** (Customer ID (CUSTOMERS), DT_Updated, Name, Product_ID (PRODUCTS))
- **PROJECT_PHOTO** ({Customer, Project DT} (PROJECTS), Photo URL (PHOTOS))
- **BILLINGS** (CC Number, CC_Expiry, CC_Holder, Street Name, Street Number, City, Zip)
- **CUSTOMER BILLING** (Customer ID (CUSTOMERS), Credit Card (BILLINGS))
- **ORDERS** (Order ID, Order_DT, Carrier, DT_Expected, Credit_Card (BILLINGS))
- **CONTENTS** (Order (ORDERS), {Customer, Project DT} (PROJECTS), Quantity)

הנחות והסברים נוספים:

- **נרמול** – ההמרה למודל הטבלאי לפי הכללים שנלמדו דרשה טיפול בשדה מרובה ערכים [Interests] בלבד. לא נמצאו תלויות פונקציונאליות סבירות.
- **שמות הטבלאות שמקורם קשר רבים-רבים** – כיוון שיש שלושה טבלאות המבוססות על קשרי רבים-רבים, אילו היינו מעניקים לכל אחת מטבלאות אלו שם ייחודי המתאר ספציפית את יחסי הגומלין, אזי היה עלול להיווצר בלבול ביניהן. על מנת למנוע בלבול זה, נעשתה בחירה בפורמט טכני אך ברור יותר של שם בצורת ישות_ישות.



מטלה 1 (25%) – שאלות

שתי שאלות SELECT ללא קינן (5%)

שאלתה 1:

לקראת ההטבות שמעניקה החברה לכמה מלקוחותיה, לכבוד יום האהבה, נדרש לדעת מיהם הלקוחות שבנו באתר 5 ויותר מתנות ליקיריהם? (פרויקטים המבוססים מוצרי קטגוריית Gifts) שיזכו בפרס.

```
SELECT C.Customer_ID, [Full Name] = C.First_Name + ' ' + C.Last_Name, [Gift
Projects] = COUNT (*)
FROM CUSTOMERS AS C JOIN PROJECTS AS PRJ ON
C.Customer_ID = PRJ.Customer_ID JOIN PRODUCTS AS PRD ON
PRJ.Product_ID = PRD.Product_ID JOIN CATEGORIES AS CAT ON
PRD.Category = CAT.Category_Name
WHERE CAT.Parent_Category = 'Gifts'
GROUP BY C.Customer_ID, C.First_Name, C.Last_Name
HAVING COUNT (*) >= 5
ORDER BY COUNT (*) DESC, C.Customer_ID DESC
```

הפלט המתקבל:

	Customer_ID	Full Name	Gift Projects
1	120	Lavon Larkins	6
2	84	Talyn Verry	6
3	80	Lennie Nassie	6
4	150	Jessie Close	5
5	124	Ashtin Dries	5
6	77	Memitt MacSkeagan	5
7	54	Jaydyn Lyddy	5



שאלתה 2:

סוף חודש יוני מתקרב, אך המנכ"ל מעוניין לדעת כעת כמה מוצרים נמכרו, לפי קטגוריה?

```
SELECT PRD.Category, Sales = COUNT (*)
FROM PRODUCTS AS PRD JOIN PROJECTS AS PRJ ON PRD.Product_ID = PRJ.Product_ID
      JOIN CONTENTS AS C ON PRJ.Customer_ID = C.Customer AND PRJ.DT_Updated =
C.Project_DT
      JOIN ORDERS AS O ON C.[Order] = O.Order_ID
WHERE Year(O.Order_DT) = 2022 AND MONTH(O.Order_DT) = 6
GROUP BY PRD.Category
ORDER BY Sales DESC
```

הפלט המתקבל:

	Category	Sales
1	Calendars	6
2	Gifts for Mom	5
3	Wall Art	4
4	Wedding Books	4
5	Gifts for Kids	3
6	Gifts for Pets	3
7	Pillows & Blankets	3
8	Thanksgiving	3
9	Travel Books	2
10	Gifts for Him	1
11	Halloween	1
12	Hanukkah	1



שתי שאלות SELECT מקוננות (10%)

שאלתה מקוננת המחזירה סקלר:

מהו השיעור אותו תופסת כל קטגוריית על, בפרויקטים הנבנים באתר?

```
SELECT [Super Category] = Parent_Category, Projects = COUNT(*), Rate =  
CONCAT(ROUND(CAST(COUNT(*) AS float) / (  
    SELECT COUNT(*)  
    FROM PROJECTS  
) , 2) * 100, '%')  
FROM PROJECTS AS PRJ JOIN PRODUCTS AS PRD ON PRJ.Product_ID = PRD.Product_ID  
    JOIN CATEGORIES AS CAT ON PRD.Category = CAT.Category_Name  
GROUP BY Parent_Category  
ORDER BY Rate DESC
```

הפלט המתקבל:

	Super Category	Projects	Rate
1	Home Decor	406	29%
2	Gifts	366	26%
3	Stationery	263	19%
4	Holidays	207	15%
5	Photo Books	158	11%



שאלתה מקוננת המחזירה רשימה:

לצרכי בדיקה מסוימת של מערכת ההמלצות, להלן שאלתה המחזירה את כל הלקוחות המוגדרים כבעלי עניין בKids, ואת מספר ההזמנות של כל אחד מהם עבור מוצרי קטגוריית Gifts for Kids.

```
SELECT C.Customer, [Kid Gifts Orders] = SUM(Quantity)
FROM CONTENTS AS C JOIN PROJECTS AS P ON C.Customer = P.Customer_ID AND
C.Project_DT = P.DT_Updated
WHERE C.Customer IN (
    SELECT Customer_ID
    FROM INTERESTS
    WHERE Interest = 'Kids'
) AND P.Product_ID IN (
    SELECT Product_ID
    FROM PRODUCTS
    WHERE Category = 'Gifts for Kids'
)
GROUP BY C.Customer
ORDER BY [Kid Gifts Orders] DESC
```

הפלט המתקבל (10 מתוך 24 רשומות):

	Customer	Kid Gifts Orders
1	147	16
2	83	12
3	94	7
4	120	6
5	126	5
6	1	5
7	61	5
8	80	5
9	145	5
10	82	4



שתי שאלות מקוננות תוך שימוש במרכיבים נוספים (10%)

שאלת המבצעת חיתוך וחיסור

מציאת מזהי הלקוחות האקטיביים, שמזמינים באופן כללי מעל המשתמש הממוצע, וכן הם שמורים במערכת כלקוחות שמתעניינים בTravel, אך בפועל – הם מזמינים מתחת לממוצע בקטגוריית Travel Books.

```
SELECT Customer_ID
FROM INTERESTS
WHERE Interest = 'Travel'

INTERSECT

SELECT Customer
FROM CONTENTS
GROUP BY Customer
HAVING Sum(Quantity) > (
    SELECT AVG(Items)
    FROM (
        SELECT Customer, Items = SUM (Quantity)
        FROM CONTENTS
        GROUP BY Customer
    ) AS A)

EXCEPT

(
    SELECT Customer
    FROM (
        SELECT C.Customer, Items = SUM(C.Quantity)
        FROM CONTENTS AS C JOIN PROJECTS AS P ON C.Customer = P.Customer_ID
        AND C.Project_DT = P.DT_Updated
        JOIN PRODUCTS AS PRD ON P.Product_ID = PRD.Product_ID
        WHERE PRD.Category = 'Travel Books'
        GROUP BY C.Customer ) as t1
    WHERE Items <
        (
            SELECT AVG(Items)
            FROM (
                SELECT C.Customer, Items = SUM(C.Quantity)
                FROM CONTENTS AS C JOIN PROJECTS AS P ON C.Customer =
P.Customer_ID AND C.Project_DT = P.DT_Updated
                JOIN PRODUCTS AS PRD ON P.Product_ID =
PRD.Product_ID
                WHERE PRD.Category = 'Travel Books'
                GROUP BY C.Customer
```



) AS B
)
)

הפלט המתקבל:

Results		Messages	
	Customer_ID		
1	115		
2	15		
3	43		
4	149		
5	61		
6	147		
7	99		
8	102		
9	60		
10	34		
11	91		
12	103		
13	120		



שאלת המבצעת עדכון

שאלת עדכון סך ההכנסות ממכירות, לכל מוצר.

נוסיף לטבלת המוצרים את השדה המתאים:

```
ALTER TABLE PRODUCTS
ADD Year_Income smallmoney
```

פעולת העדכון:

```
UPDATE PRODUCTS
SET Year_Income = (
    SELECT ISNULL(SUM (C.Quantity) * PRODUCTS.Price,0)
    FROM CONTENTS AS C JOIN PROJECTS AS P ON C.Customer = P.Customer_ID AND C.Project_DT
= P.DT_Updated
    JOIN ORDERS AS O ON C.[Order] = O.Order_ID
    WHERE P.Product_ID = PRODUCTS.Product_ID AND Year(Order_DT) = Year(getDate())
)
```

הטבלה המעודכנת (10 מתוך 50 הרשומות):

```
SELECT *
FROM PRODUCTS
ORDER BY Year_Income DESC
```

	Product_ID	Product_name	Size	Price	in_stock	Category	Year_Income
1	22	Simple Wedding	12x12	29.13	1	Wedding Books	786.51
2	31	So Thankful	6x6	25.72	1	Thanksgiving	745.88
3	2	Water Bottle		32.31	1	Drinkware	516.96
4	35	Indoor Pillow		30.78	1	Pillows & Blankets	492.48
5	46	Canvas Tile		30.20	1	Wall Art	483.20
6	37	Fleece Blanket		26.63	0	Pillows & Blankets	452.71
7	36	Flight Pillow		37.00	1	Pillows & Blankets	444.00
8	34	Elegant Star		21.02	1	Hanukkah	441.42
9	23	Elegant Wedding	12x36	33.07	1	Wedding Books	429.91
10	8	Wall Calendar	12x12	26.12	1	Calendars	417.92



מטלה 2 (25%) – יישומי כלים מתקדמים (5%) View

-- a VIEW used frequently by the marketing team, to build a mailing list for new promos

GO

```
CREATE VIEW v_InterestRecommendation AS
    SELECT I.Interest, C.Email
    FROM CUSTOMERS AS C JOIN INTERESTS AS I ON C.Customer_ID =
I.Customer_ID
    GROUP BY I.Interest, C.Email
```

GO

-- an example usage of the object

```
SELECT Email
FROM v_InterestRecommendation
WHERE Interest = 'Gardening'
```

הפלט המתקבל (10 מתוך 65 הרשומות):

Results		Messages
	Email	
1	thoofe0@hubpages.com	
2	htoms1@usgs.gov	
3	dcatherick2@google.com	
4	rstaddom5@webeden.co.uk	
5	nmainz6@adobe.com	
6	jplewman7@theforest.net	
7	mpickervance8@comsenz.com	
8	kskain9@meriam-webster.com	
9	nlaethamb@networkadvertising.org	
10	rburfielde@freewebs.com	



פונקציות (Functions) (10%)

פונקציה המחזירה טבלה

לצרכי שיווק מותאם גיאוגרפית, פונקציה זו תקבל שם של עיר, ותחזיר את 5 המוצרים שהכי נמכרים בה. הפלט כולל את פרטי כל מוצר, וסך המכירות עבורו.

```
CREATE FUNCTION BestSellers (@INPUT_City varchar(20))
RETURNS TABLE
AS
RETURN
SELECT TOP 5 PRD.Product_ID, PRD.Product_name, PRD.Category,
PRD.Price, [Total Sold] = SUM(C.Quantity)
FROM PRODUCTS AS PRD JOIN PROJECTS AS PRJ ON PRD.Product_ID =
PRJ.Product_ID
JOIN CONTENTS AS C ON C.Customer = PRJ.Customer_ID AND C.Project_DT
= PRJ.DT_Updated
JOIN ORDERS AS O ON C.[Order] = O.Order_ID JOIN BILLINGS AS B ON
O.Credit_Card = B.CC_Number
WHERE B.City = @INPUT_City
GROUP BY PRD.Product_ID, PRD.Product_name, PRD.Category, PRD.Price
ORDER BY [Total Sold] DESC
```

שאלתה לדוגמה להדגמת הפעולה:

```
SELECT *
FROM dbo.BestSellers('Denver')
```

הפלט המתקבל:

Results		Messages			
	Product_ID	Product_name	Category	Price	Total Sold
1	16	My Best Friend	Gifts for Pets	28.74	8
2	13	Collage Puzzle	Gifts for Kids	21.63	7
3	23	Elegant Wedding	Wedding Books	33.07	6
4	44	Metal Tile	Wall Art	32.99	6
5	21	Beer Mug	Gifts for Him	13.77	5



Trigger פשוט (5%)

טריגר פשוט זה מעדכן את מספר המוצרים השייכים לסאב-קטגוריות, כאשר מתבצע כל שינוי שהוא בטבלת המוצרים. כצעד מקדים, נוסיף לכל קטגוריה שדה Total_Products.

```
ALTER TABLE CATEGORIES
ADD Total_Products smallint
```

```
go
CREATE TRIGGER UpdateTotalProductsInCategory
ON
PRODUCTS
FOR
INSERT, UPDATE, DELETE
AS
UPDATE CATEGORIES
SET
Total_Products = (
SELECT COUNT(*)
FROM PRODUCTS AS P
WHERE P.Category = CATEGORIES.Category_Name
)
WHERE Category_Name IN (
SELECT DISTINCT Category_Name FROM inserted
UNION
SELECT DISTINCT Category_Name FROM deleted
)
```

להדגמה התקינה של פעולתו, נוסיף כמה מוצרים:

```
INSERT INTO PRODUCTS (Product_ID, Category)
VALUES
(51, 'Gifts for Mom'),
(52, 'Gifts for Mom'),
(53, 'Travel Books')
```

ובהתאם השינוי בטבלת Categories:

	Results	Messages	
	Category_Name	Parent_Category	Total_Products
1	Calendars	Stationery	6
2	Drinkware	Home Decor	2
3	Family Books	Photo Books	2
4	Gifts	NULL	0
5	Gifts for Him	Gifts	2
6	Gifts for Kids	Gifts	6
7	Gifts for Mom	Gifts	2
8	Gifts for Pets	Gifts	3
9	Halloween	Holidays	3
10	Hanukkah	Holidays	2
11	Holidays	NULL	0
12	Home Decor	NULL	0
13	Journals	Stationery	3
14	Photo Books	NULL	0
15	Pillows & Blankets	Home Decor	3
16	Stationery	NULL	0
17	Thanksgiving	Holidays	2
18	Travel Books	Photo Books	2
19	Wall Art	Home Decor	10
20	Wedding Books	Photo Books	2

(20 rows affected)
(3 rows affected)



Results

Messages

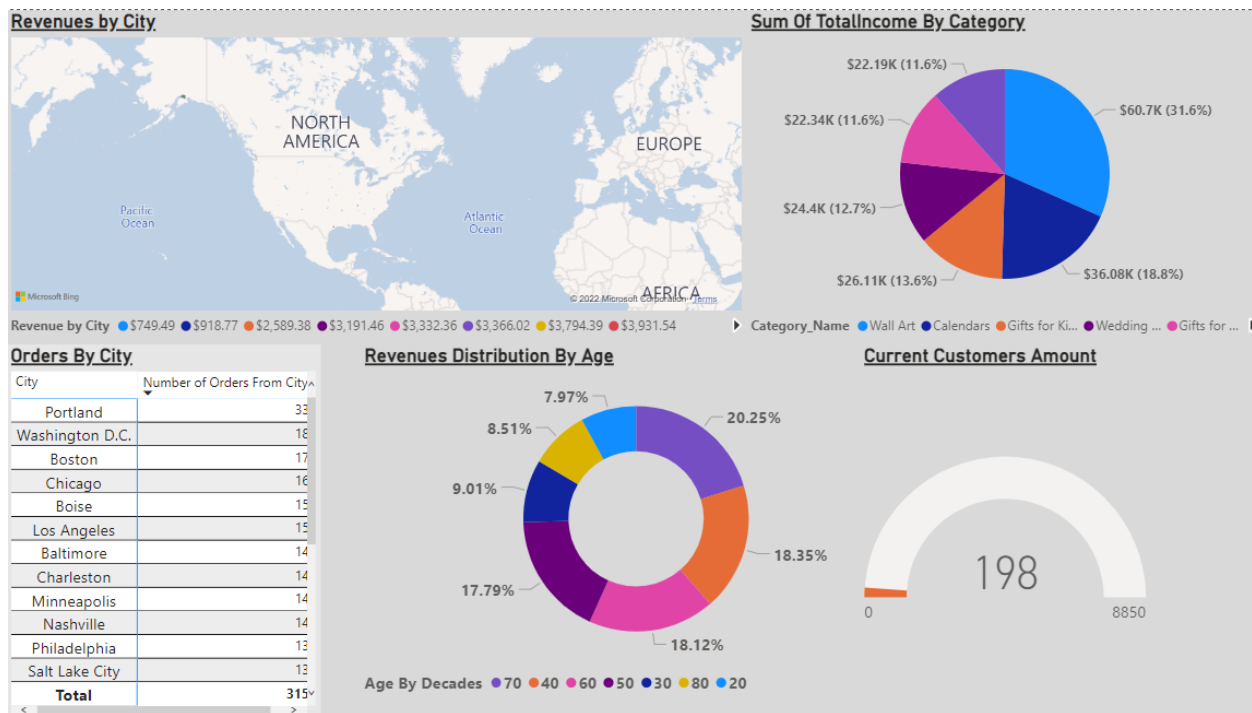
	Category_Name	Parent_Category	Total_Products
1	Calendars	Stationery	6
2	Drinkware	Home Decor	2
3	Family Books	Photo Books	2
4	Gifts	NULL	0
5	Gifts for Him	Gifts	2
6	Gifts for Kids	Gifts	6
7	Gifts for Mom	Gifts	4
8	Gifts for Pets	Gifts	3
9	Halloween	Holidays	3
10	Hanukkah	Holidays	2
11	Holidays	NULL	0
12	Home Decor	NULL	0
13	Journals	Stationery	3
14	Photo Books	NULL	0
15	Pillows & Blankets	Home Decor	3
16	Stationery	NULL	0
17	Thanksgiving	Holidays	2
18	Travel Books	Photo Books	3
19	Wall Art	Home Decor	10
20	Wedding Books	Photo Books	2



מטלה 3 (20%) – כלים להצגת נתונים

דו"ח עסקי

הדוח אשר בחרנו להכין מחולק לשני חלקים, החלק הראשון יוגש למחלקת השיווק של החברה. מטרתה של מחלקת השיווק היא להבין היכן יש לתחזק את הקשרים בערים בעלות השפעה גדולה על הכנסות החברה וכמו כן, לחזק ולהתמקד בערים בעלות הכנסה פחותה לחברה. כמו כן, דרך ניתוח חתך הגילאים להבין היכן יש לשים מאמץ בשיווק ובפרסום ולהגדיל את כמות הלקוחות.



התרשימים בדו"ח:

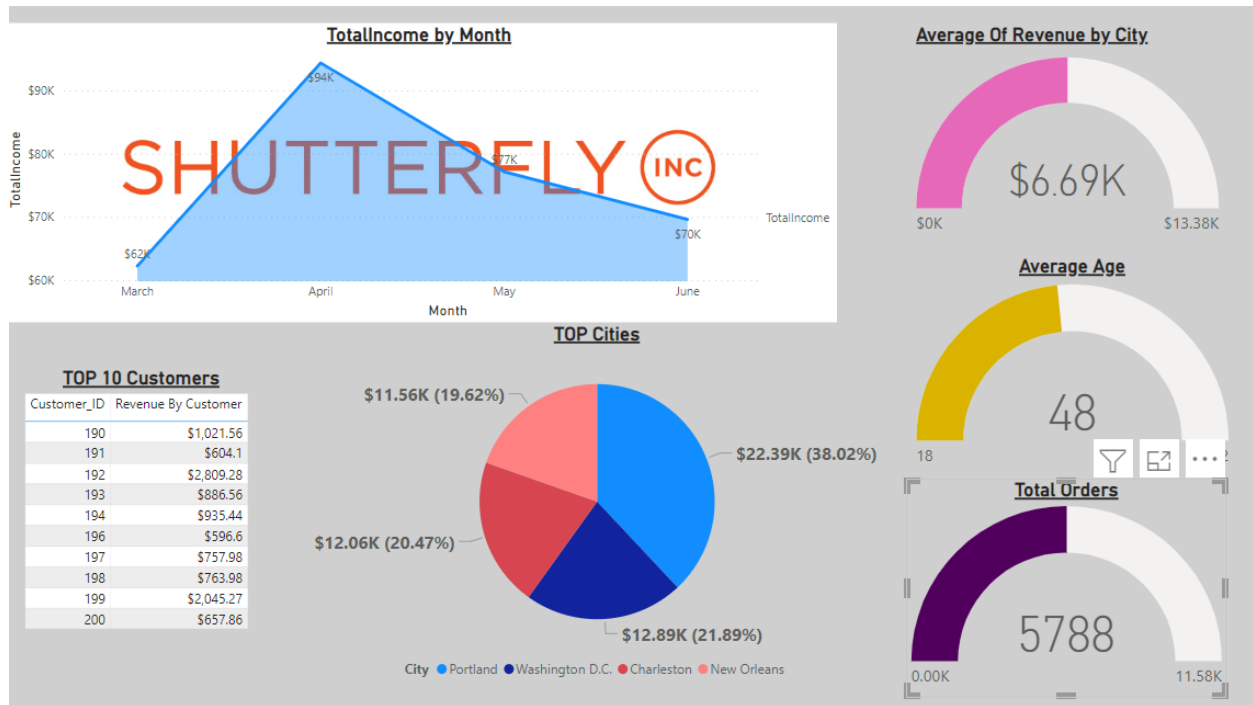
- מפה עם הערים בהן נעשו הזמנות, בכל עיר מצוינת כמות ההכנסות הכוללת.
- מטריצה המכילה אפשרות לעשות "זום אין" לכל אחת מהערים ולידה מספר ההזמנות הכולל בעיר זו
- טבלת פיזור אחוז הכנסה לפי גיל (מחולק לעשורים)
- מד המראה כמות לקוחות נוכחית
- טבלת "פאי" המציגה את הקטגוריות המניבות ביותר ואחוז הרכישה מהן מתוך כלל האפשרויות האחרות



"לוח מחוונים" - דרג ניהולי

דוח זה מיועד לדרג הבכיר של החברה. מטרתו להציג בגרעיניות נמוכה את עיקרי הנתונים הרלוונטיים לצורך התבוננות רחבה על פעילות החברה. על הנתונים להציג את עיקרי הפעילות ולהראות את "השורה התחתונה"

בדוח זה בחרנו להציג נתונים אשר יעזרו להראות את המגמות בחברה לדרג הבכיר:



התרשימים בדו"ח:

- מד גיל ממוצע
- מד סה"כ הזמנות שנעשו
- טבלת עשרת הלקוחות המובילים מבחינת כסף
- טבלת "פאי" של ארבע הערים בהן נעשו הכי הרבה הזמנות וההכנסות מערים אלו
- מד ממוצע הכנסה לפי כל הערים
- גרף הכנסה לפי חודש



לצורך הזנת הנתונים לתוכנת ה־HUE נעזרנו בקובץ אקסל מדויק(ראה נספחים)

הקוד בו השתמשנו להצגת הנתונים בתוכנה:

```
--DROP View Sales_Trend_Line
Create View Sales_Trend_Line as
select O.Order_ID, O.Order_DT , C1.Customer, TotalIncome = C1.Quantity * P2.Price
from ORDERS as O join CONTENTS as C1 on O.Order_ID = C1.[Order] join PROJECTS as P1 on
C1.Customer = P1.Customer_ID
join PRODUCTS as P2 on P1.Product_ID = P2.Product_ID join CATEGORIES as C3 on P2.Category
= C3.Category_Name
```

```
--DROP View Sum_Total_Income_Per_Category
Create View Sum_Total_Income_Per_Category as

select C3.Category_Name , TotalIncome = SUM(C1.Quantity * P2.Price)
from ORDERS as O join CONTENTS as C1 on O.Order_ID = C1.[Order] join PROJECTS as P1 on
C1.Customer = P1.Customer_ID
join PRODUCTS as P2 on P1.Product_ID = P2.Product_ID join CATEGORIES as C3 on P2.Category
= C3.Category_Name
Group By C3.Category_Name
```

```
--DROP VIEW View_Revenue_By_Customer_Age
create view View_Revenue_By_Customer_Age as
select C4.Customer_ID , [Customer Age] = DATEDIFF(year, C4.Birth_Date, GETDATE()),
[Revenue By Customer] = sum(DISTINCT(C1.Quantity * P2.Price))
from ORDERS as O join CONTENTS as C1 on O.Order_ID = C1.[Order]
join PROJECTS as P1 on C1.Customer = P1.Customer_ID join CUSTOMERS as C4
on C4.Customer_ID = P1.Customer_ID join PRODUCTS as P2 on P1.Product_ID = P2.Product_ID
join CATEGORIES as C3 on P2.Category = C3.Category_Name
Group By C4.Customer_ID, C4.Birth_Date
```

```
--DROP VIEW Revenue_By_City
CREATE VIEW Revenue_By_City AS
select B.City as City,
[Revenue by City] = sum(DISTINCT(C1.Quantity * P2.Price)),
[Number of Orders From City] = COUNT(DISTINCT(C4.Customer_ID))
from ORDERS as O join CONTENTS as C1 on O.Order_ID = C1.[Order]
join PROJECTS as P1 on C1.Customer = P1.Customer_ID join CUSTOMERS as C4
on C4.Customer_ID = P1.Customer_ID join PRODUCTS as P2 on P1.Product_ID = P2.Product_ID
join CUSTOMER_BILLING as C5 on C5.Customer_ID = C4.Customer_ID join BILLINGS as B on
b.CC_Number = C5.Credit_Card join CATEGORIES as C3 on P2.Category = C3.Category_Name
Group By B.city
```



```
--DROP VIEW VIEW_max_min_avg_age
CREATE VIEW VIEW_max_min_avg_age as
select
    [Youngest Customer Age] = (
        SELECT MIN(DATEDIFF(year, C.Birth_Date, GETDATE()))
    ),
    [Oldest Customer Age] = (
        SELECT Max(DATEDIFF(year, C.Birth_Date, GETDATE()))
    ),
    [AVG Customers Age] = (
        SELECT AVG(DATEDIFF(year, C.Birth_Date, GETDATE()))
    )

FROM CUSTOMERS as C


--DROP View View_Revenue_By_TOP10_Customer
create view View_Revenue_By_TOP10_Customer as
select TOP 10 C4.Customer_ID ,
[Revenue By Customer] = sum(DISTINCT(C1.Quantity * P2.Price))
from ORDERS as O join CONTENTS as C1 on O.Order_ID = C1.[Order]
join PROJECTS as P1 on C1.Customer = P1.Customer_ID join CUSTOMERS as C4
on C4.Customer_ID = P1.Customer_ID join PRODUCTS as P2 on P1.Product_ID = P2.Product_ID
join CATEGORIES as C3 on P2.Category = C3.Category_Name
Group By C4.Customer_ID
ORDER BY 1 DESC, 2 DESC
```



מטלה 4 (צבירה - עד 35%) – יישומים מתקדמים שאלות עסקיות המשלבות Window Functions (10%)

שאלתה 1

שאלתה זו, העושה שימוש בשתי פונקציות חלון `LAG()`, `RANK()`, מציגה בכל עיר, ובכל שנה עד 2021, את גובה ההכנסות ממכירות אליה, אחוז הגדילה במדד זה בהשוואה לשנה הקודמת, וכן את דירוגה בהכנסות בהשוואה לערים האחרות, באותה השנה.

```
CREATE VIEW V_CityYearIncome
AS
SELECT City, [Year] = YEAR(O.Order_DT), Income = SUM (PRD.Price *
CNT.Quantity)
FROM CONTENTS AS CNT JOIN ORDERS AS O ON CNT.[Order] = O.Order_ID
JOIN BILLINGS AS B ON O.Credit_Card = B.CC_Number
JOIN PROJECTS AS PRJ ON CNT.Customer = PRJ.Customer_ID AND
CNT.Project_DT = PRJ.DT_Updated
JOIN PRODUCTS AS PRD ON PRJ.Product_ID = PRD.Product_ID
WHERE YEAR(O.Order_DT) <= 2021
GROUP BY City, YEAR(O.Order_DT)

SELECT YearlyIncomeRank = (RANK () OVER(PARTITION BY Year ORDER BY Income
DESC)), City, [Year], Income,
YearlyGrowth = CONCAT(CAST((Income / (LAG (Income) OVER(PARTITION BY City ORDER
BY Year)) -1)*100 AS INT), '%')
FROM V_CityYearIncome
ORDER BY Year DESC, YearlyIncomeRank
```

הפלט המתקבל (15 מתוך 90 הרשומות):

	YearlyIncomeRank	City	Year	Income	YearlyGrowth
1	1	Boston	2021	2622.29	61%
2	2	Columbus	2021	2341.65	44%
3	3	San Jose	2021	1929.67	-6%
4	4	San Francisco	2021	1663.59	272%
5	5	Austin	2021	1656.65	-11%
6	6	Jacksonville	2021	1650.10	34%
7	7	Portland	2021	1614.58	-18%
8	8	Charlotte	2021	1406.65	-2%
9	9	Denver	2021	1385.18	248%
10	10	Chicago	2021	1247.46	333%
11	11	El Paso	2021	1172.68	-44%
12	12	Las Vegas	2021	1037.24	52%
13	13	San Diego	2021	1035.04	-35%
14	14	Washington	2021	1027.77	23%
15	15	Fort Worth	2021	1015.15	-15%



שאלתה 2

שאלתה זו, עושה שימוש בשתי פונקציות חלון: LEAD() ו-ROW_NUMBER(), מציגה עבור כל קטגוריה באתר ולפי שנים, את שיעור ההצלחה של פרויקטים שנפתחו תחת אותה קטגוריה באותה שנה, כאשר הצלחה של פרויקט נמדדת בתור הצלחה במכירתו. כמו כן, לצד שיעור ההצלחה מופיעה דרגת השנה של הקטגוריה, המקבלת ערכים בין 1 ל-6, כאשר 1 מציין את השנה המוצלחת ביותר של הקטגוריה. בנוסף, מופיע השינוי בשיעור ההצלחה המשוקלל של הקטגוריה בין כל שנתיים סמוכות, וגם הממוצע הנע של שיעור ההצלחה ב-3 השנים הסמוכות.

```
GO
CREATE VIEW V_OrderedProjects
AS
SELECT DISTINCT PRJ.Customer_ID, PRJ.DT_Updated, PRJ.Product_ID, Ordered = CASE WHEN
(C.[Order] > 0) THEN 1 ELSE 0 END
FROM PROJECTS AS PRJ LEFT JOIN CONTENTS AS C ON PRJ.Customer_ID = C.Customer AND
PRJ.DT_Updated = C.Project_DT
GROUP BY PRJ.Customer_ID, PRJ.DT_Updated, PRJ.Product_ID, C.[Order]
```

```
GO
CREATE VIEW V_CategorySuccessByYear
AS
SELECT P.Category, [Success Rate] = CAST(100.0 * SUM(Ordered) / COUNT(Ordered) AS
DECIMAL(5,2)), [Year] = Year(DT_Updated)
FROM V_OrderedProjects AS V JOIN PRODUCTS AS P ON V.Product_ID = P.Product_ID
GROUP BY P.Category, Year(DT_Updated)
GO
```

```
SELECT Category, [Year],
[Year Success Rate (1-6)] = ROW_NUMBER() OVER(PARTITION BY Category ORDER BY [Success Rate]
DESC),
[Success Rate] = CONCAT([Success Rate], '%'),
[Success Rate Change] = CONCAT([Success Rate] - LEAD([Success Rate], 1) OVER(PARTITION BY
Category ORDER BY [Year]), '%'),
[3 Year Moving Average] = CONCAT(CAST(AVG([Success Rate]) OVER(PARTITION BY Category ORDER
BY [Year] ROWS BETWEEN CURRENT ROW AND 2 FOLLOWING) AS DECIMAL(5,2)), '%')
FROM V_CategorySuccessByYear
ORDER BY Category, [Year]
```

הפלט המתקבל (15 מתוך 88 הרשומות):

	Category	Year	Year Success Rate (1-6)	Success Rate	Success Rate Change	3 Year Moving Average
1	Calendars	2017	5	80.00%	3.08%	79.23%
2	Calendars	2018	6	76.92%	-3.85%	81.65%
3	Calendars	2019	4	80.77%	-6.50%	84.98%
4	Calendars	2020	1	87.27%	0.38%	86.62%
5	Calendars	2021	2	86.89%	1.18%	86.30%
6	Calendars	2022	3	85.71%	%	85.71%
7	Drinkware	2017	1	100.00%	0.00%	91.67%
8	Drinkware	2018	2	100.00%	25.00%	86.40%
9	Drinkware	2019	6	75.00%	-9.21%	83.07%
10	Drinkware	2020	5	84.21%	-5.79%	87.24%
11	Drinkware	2021	3	90.00%	2.50%	88.75%
12	Drinkware	2022	4	87.50%	%	87.50%
13	Family Books	2018	2	100.00%	0.00%	100.00%
14	Family Books	2019	1	100.00%	0.00%	96.67%
15	Family Books	2020	3	100.00%	10.00%	91.11%



שילוב מערכתי של מספר כלים (10%)

בחרנו להכין מערכת אשר מדמה פעילות שעושה אלגוריתם מערכת ההמלצות שבאתר.

```
-- Complex Action
--DROP FUNCTION Average_Items_By_Category
GO
CREATE FUNCTION Average_Items_By_Category(@Category varchar(50))
RETURNS REAL
AS
BEGIN
RETURN
(
    SELECT AVG(T.Items * 1.0) FROM
    (
        SELECT C.Customer, Items = SUM(C.Quantity)
        FROM CONTENTS AS C
        JOIN PROJECTS AS P ON C.Customer = P.Customer_ID AND C.Project_DT =
P.DT_Updated
        JOIN PRODUCTS AS PRD ON P.Product_ID = PRD.Product_ID
        WHERE PRD.Category = @Category
        GROUP BY C.Customer
    ) AS T
)
END
GO

--SELECT dbo.Average_Items_By_Category('Travel Books')

--DROP FUNCTION FrequentlyPurchasingCustomers
GO
CREATE FUNCTION FrequentlyPurchasingCustomers(@Category varchar(50))
RETURNS TABLE
AS
RETURN
(
    SELECT C.Customer
    FROM CONTENTS AS C
    JOIN PROJECTS AS P ON C.Customer = P.Customer_ID AND C.Project_DT = P.DT_Updated
    JOIN PRODUCTS AS PRD ON P.Product_ID = PRD.Product_ID
    WHERE PRD.Category = @Category
    GROUP BY C.Customer
    HAVING SUM(C.Quantity) > dbo.Average_Items_By_Category(@Category)
)
GO

--SELECT * FROM dbo.FrequentlyPurchasingCustomers('Travel Books')

-- DROP TABLE OLD_INTERESTS
CREATE TABLE OLD_INTERESTS
(
    Customer_ID INT NOT NULL,
    Old_Interest varchar(15) NOT NULL

    PRIMARY KEY (Customer_ID, Old_Interest)
)
```



```
--DROP TRIGGER UpdateOldInterests
GO
CREATE TRIGGER UpdateOldInterests
ON INTERESTS
FOR DELETE
AS INSERT INTO OLD_INTERESTS
SELECT D.Customer_ID, D.Interest
FROM DELETED AS D
GO

-- The procedure updates the interest table by getting the frequently
-- purchasing customers of the matching category and deleting
-- the interest from the customer if he is not a frequent buyer any more.
GO
CREATE PROCEDURE UpdateInterestByCategoryPurchases
(
    @Interest varchar(50),
    @Category varchar(50)
)
AS
BEGIN
    DELETE FROM INTERESTS
    WHERE Interest = @Interest AND Customer_ID IN
    (
        SELECT Customer_ID
        FROM INTERESTS
        WHERE Interest = @Interest AND
        Customer_ID NOT IN
        (
            SELECT *
            FROM dbo.FrequentlyPurchasingCustomers(@Category)
        )
    )
END
GO

EXEC UpdateInterestByCategoryPurchases
    @Interest = 'Kids', @Category = 'Gifts for Kids'

SELECT *
FROM OLD_INTERESTS
```