|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מס' קבוצה | אתר | | | תאריך הגשה |
| 24 | [www.simplyrecipes.com](http://www.simplyrecipes.com) | | | 05/06/2019 |
| חברי הצוות - מספרי ת.ז | | | | |
| 123456789 | | 311605497 | 312469083 | |

## פרויקט בסיסי נתונים – חלק ג'

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

## מטלה 1 – שאילתות

# שתי שאילתות SELECT ללא קינון

1. שאילתה: הצג את שמות המחברים שנרשמו למערכת החל מ2018 ושיש להם לפחות מתכון אחד שממוצע ציונו גדול מ-3

הגיון עסקי: אנו נרצה להבין מי הם המחברים החדשים אשר אחד ממתכונם או יותר מדורג בציון 4 ומעלה. נתון זה יעזור לנו לפלח את אוכלוסיית המחברים ולהגיע לכותבים שאנו רואים בהם פוטנציאל עתידי, וכן לשקול להגדיל את חשיפתם באתרנו או להעניק להם תמריץ כספי על מנת לשמר אותם אצלנו.

קוד:

SELECT distinct U.FirstName , U.LastName

FROM Composers AS c JOIN Users AS U ON

c.Email = U.Email JOIN Recipes AS RE ON

Re. UploadedBy = c.Email JOIN Ranking AS RA ON

RA. RecipeName = RE. RecipeName

WHERE YEAR([DateOfRegistration])>2018

GROUP BY U.FirstName , U.LastName , RA. RecipeName

HAVING avg (score)>3

2. שאילתה: הצג את שמות המתכונים הכשרים אשר מתאימים לקיץ ומכילים לפחות חמישה רכיבים.

הגיון עסקי: לקראת הקיץ נרצה להגיע לפלח שוק של האוכלוסייה היהודית, נעשה זאת על ידי פרסום מתכונים קיציים כשרים המתאימים לאוכלוסייה היהודית. בנוסף, נרצה לשווק מתכונים בהם מספר רב של רכיבים, זאת על מנת לעודד קנייה גדולה מתוך האתר דבר שיגדיל את ההכנסות.

קוד:

SELECT distinct R.RecipeName

FROM Recipes AS R JOIN RecipeSpecialDiet AS RS ON

R.RecipeName = RS.RecipeName JOIN Contain AS C ON

R.RecipeName = C.RecipeName

Where R.season = 'summer' AND RS.specialDiet = 'kosher'

GROUP BY R.RecipeName

HAVING count(\*)>=5

# שתי שאילתות SELECT מקוננות

3. שאילתה: שאילתה מקוננת המחזירה ערך סקלאר - מחזירה את שמות המתכונים, המחברים שלהם, ציונם הממוצע ותאריך העלאה של כל המתכונים שממוצע ציונם גבוה מהציון הממוצע של כלל המתכונים, שהצביעו להם מעל 3 פעמים והועלו לפני פחות משנתיים

הגיון עסקי- אנו רוצים לזהות מהם מהמתכונים המוצלחים ביותר באתר, מיהו המחבר שלהם ופרטים על המתכון, הפרמטרים שנקבעו לכך מוודאים שמספיק אנשים הצביעו למתכון ושהוא לא ישן מידי. כך נוכל לחזק מתכונים אילו ולצ'פר את המחברים שלהם.

קוד:

select r.RecipeName, AverageScore = avg (score) , recipeComposer = U.FirstName + ' ' + U.LastName , R.UploadDate

from Recipes as r join Ranking as ra on ra.RecipeName=r.RecipeName join Users as u on u.Email = r.UploadedBy

WHERE R.UploadDate > GETDATE() - 730

group by r.RecipeName, U.FirstName , U.LastName, R.UploadDate

having avg (score) > (select AVG (score)

from Ranking)

and count (score) > 3

4. שאילתה: הצג את הכותבים שמרוויחים מתחת למשכורת הממוצעת, אשר ממוצע הדירוג שלהם הוא בין חמשת הראשונים.

הגיון עסקי: על ידי כתיבת שאילתה זו, אנו נרצה לקבל את הכותבים שממוצע הדירוג שלהם הוא בין חמשת הראשונים ומרוויחים מתחת לממוצע המשכורות. נתון זה יסייע לנו להבין מי הם המחברים האהובים ביותר, מתכונים מוצלחים ואהובים גורמים לחשיפה רחבה של האתר, שימור ציבור המשתמשים וכן עידוד קניית מוצרים. מחברים אלו חשובים לחברה ובכדי לשמר אותם נרצה להציע להם משכורת הולמת יותר המתאימה לאיכות תפוקתם.

קוד:

SELECT u.FirstName, u.LastName, salary

FROM Composers as C join users as U on c.Email = u.Email join

(select top 5 r.UploadedBy, avrageScore = avg(score)

from Recipes as R join Ranking as Ran on r.RecipeName = ran.RecipeName

group by r.UploadedBy

order by 2 DESC ) as newTable on C.Email = newTable.UploadedBy

where salary < (select avg(salary)

from Composers)

# שתי שאילתות מקוננות תוך שימוש במרכיבים נוספים

5. שאילתה: שאילתת UPDATE- שאילתה המוסיפה (לעמודה חדשה שנוסיף) לכל מצרך אזהרת "הרבה קלוריות" אם מספר הקלוריות שלו מעל ממוצע הקלוריות.

הגיון עסקי: כבכירי החברה, אנו מקבלים עדכונים שוטפים מהמומחים שלנו אודות "טרנדים" בעולם בתחום הקולינריה. לידי החברה הגיע לאחרונה ההבנה כי תזונה בריאה צוברת פופולריות בכל תחומי המזון בעולם, ממסעדנות ועד ספרי ואתרי מתכונים. לכן, על מנת להישאר בקו אחד עם לקוחותינו, החלטנו לרשום על מוצרים רבי קלוריות אזהרה. דבר זה ייצור אמון באתר אצל הלקוחות (דבר הכרחי בשימור לקוחות), ואף יביא לקוחות חדשים שדבקים בתזונה בריאה ומחפשים מתכונים בריאים, בעבורם האתר שלנו הוא הכתובת.

קוד:

alter table Ingredients

add CaloriesWarning VARCHAR(100)

update Ingredients

set CaloriesWarning = 'high calorie'

WHERE NumOfCalories > (SELECT avg(NumOfCalories)

FROM Ingredients)

6. שאילתה: הצג את כל המצרכים שתוקפם פג בשלושת החודשים הקרובים, וגם ממוצע המחיר שלו גדול מממוצע המחיר של כל המצרכים.

הגיון עסקי: אנו מעוניינים להיעזר בשאילתה זו על מנת לעשות מבצע על כל המצרכים שעתידים להיזרק בעוד שלושה חודשים או פחות. בנוסף, למצרכים יקרים פוטנציאל רווחי גדול יותר ולכן כדאי להימנע מזריקתם. אי לכך, אנו נמצא מצרכים אלו מהשאילתה, ועליהם נעשה מבצע מיוחד כדי לעודד את קנייתם במחיר שאמנם נמוך ביחס למקורי, אך עדיין רווחי לחברה.

קוד:

(SELECT I.IngredientName

FROM Ingredients as I

where year(ExpressionDate) = 2019 and

MONTH(ExpressionDate) < 9 and

MONTH(ExpressionDate) > 5)

Intersect

(SELECT O.IngredientName

FROM Offer as O

group by o.IngredientName, o.Price

having avg(o.Price) > (select avg(Price)

from Offer )

)

## מטלה 2 – העשרת בסיס הנתונים

# :View

אנו ניצור VIEW שייתן את כל פרטי המשתמשים- שאינם מחברים- ללא פרטיהם האישיים (פרטי אשראי, סיסמה וכו')

הגיון עסקי: יצירת הVIEW תתן לנו אופציה של הסתרת נתונים ממשתמשים לא-מורשים וכדומה. בדוגמא זו לקוחות/עובדים בחברה אשר רוצים לראות את מאגר המשתמשים של האתר, שאין אנו רוצים שייחשפו לפרטים רגישים של הלקוחות, יוכלו להשתמש בVIEW זה וכך אנו ממדרים אותם מנתונים חסויים/שאנו לא רוצים שיחשפו אליהם.

קוד:

CREATE VIEW view\_ShowPublicProfile

AS

select U.FirstName, U.LastName, u.country, u.Gender, u.DateOfBirth, u.About, U.Email , u.DateOfRegistration

from Users as U

except

select U.FirstName, U.LastName, u.country, u.Gender, u.DateOfBirth, u.About, U.Email , u.DateOfRegistration

from Users as U join Composers as c on u.Email = c.Email

# פונקציות (Functions)

1. פונקציה: פונקציה המכניסים אליה שנה מסויימת, והיא מוציאה את כלל הספקים אשר לא רכשו מהם ולו קנייה אחת באותה שנה.

הגיון עסקי: בתור בכירי החברה אשר שואפים לשפר ולייעל בכל זמן נתון את פעילות האתר, אנו מבינים שיש צורך לשמור את היד על הדופק בכל הנוגע לפעילות הספקים, שכן בפעילותם טמון חלק נכבד מרווחיות החברה המקבלת אחוז יחסי מסכום הרכישות הנעשות דרך האתר. לכן, כאשר נרצה לבחון את טיב הספקים בדו"ח השנתי, נרצה להיעזר בפונקציה שבאמצעותה נבדוק באותה שנה האם ישנם ספקים שלא מכרו כלל. דבר זה הינו חריג ויש לבדוק מול אותם ספקים האם מצרכיהם אינם אטרקטיביים ביחס לספקים האחרים, ואם מתבקש שינוי כלשהו לקראת השנה הבאה.

קוד:

CREATE FUNCTION dbo.YearDidntProvidesNothing (@year int)

RETURNS TABLE

AS RETURN

(SELECT SupplierName

FROM Suppliers

Except

SELECT ProvidedBy

FROM Purchases as p

WHERE YEAR([DateOfPurchase])= @year

Group by p.ProvidedBy

)

2. פונקציה: פונקציה המקבלת אימייל של לקוח ומחזירה את הסכום הכולל שהלקוח שילם.

הגיון עסקי: אנו נרצה להגיע דרך פונקציה זו לסכום הכסף שלקוח ספציפי השקיע מאז הצטרפותו לאתר. פונקציה תסייע לנו לדוגמא להבין בעת פעולה חריגה באתר של לקוח מסויים, האם סכום הרכישות שביצע עד כה ודפוס קנייתו מתאים לפעולה זו, וכך לדעת כיצד להתנהל בהתאם.

קוד:

CREATE FUNCTION dbo.userTotalSpent (@Email varchar(30))

RETURNS TABLE

AS RETURN

(

select moneySpend = SUM(t2.toCost)

from (select t.NumOfPurchase,toCost = sum(t.totalcost)

from (SELECT b.NumOfPurchase, totalcost = ( B.Quantity\*o.Price)

FROM Buys as B join Purchases as p on b.NumOfPurchase = p.NumOfPurchase join Offer as o on o.SupplierName = p.ProvidedBy

Where b.IngredientName = o.IngredientName and p.MadeBy = @Email

) as t

group by t.NumOfPurchase

) as t2

# Trigger פשוט

טריגר: הטריגר מופעל כאשר נכנס מתכון חדש לרשימת מתכונים, והוא מעדכן בהתאם את מספר המתכונים החדש של כל מחבר.

הגיון עסקי: אנו רוצים שהנתונים לגבי המחברים יישארו מעודכנים ולכן כשמחבר מעלה מתכון חדש או קובץ (BATCH) של מתכונים בפעימה, הדבר ייכנס לDB של המתכונים, יקפיץ את הTRIGGER ויעדכן אוטומטית את טבלת מחברים (עמודת מספר מתכונים לכל מחבר).

קוד:

CREATE TRIGGER UploadRecipes

ON Recipes

FOR INSERT

AS

UPDATE Composers

SET numOfRecipes = numOfRecipes + (SELECT COUNT(\*)

FROM INSERTED

WHERE INSERTED.UploadedBy = Composers.Email)

UPDATE Composers

SET LastUploadeDate =ISNULL ((SELECT [UploadDate]

FROM INSERTED

WHERE INSERTED.UploadDate = Composers.Email),

LastUploadeDate)

# פרוצדורה שמורה ((Stored Procedure פשוטה

פרוצדורה: פרוצדורה שמאפשרת לנו לשנות את המחיר בהתאם לצורך.

הגיון עסקי: במקרים והספק רוצה לעלות/להוריד את מחיר אחד המוצרים (בעקבות מכירת חיסול, מבצע לחג וכו') אנו, כאתר שמתנהל באופן שוטף, חייבים אמצעי נוח לשינוי מחירים לפי אחוזים. ניעזר בפרוצדורה זו על מנת לתת לספק לממש את השינוי במחיר.

קוד:

CREATE PROCEDURE UpdatePrice

@ingredientName varchar(20), @supplierName varchar(30), @Action Varchar(10), @Rate Real

AS

IF ( @Action = 'raise' ) BEGIN

UPDATE Offer

SET Price = (1 + @Rate)\*Price

WHERE SupplierName = @supplierName and IngredientName = @ingredientName

END

ELSE IF (@Action = 'discount') BEGIN

UPDATE Offer

SET Price = (1 - @Rate)\*Price

WHERE SupplierName = @supplierName and IngredientName = @ingredientName

END

## מטלה 3 (20%) – כלים להצגת נתונים

**דו"ח עסקי:**

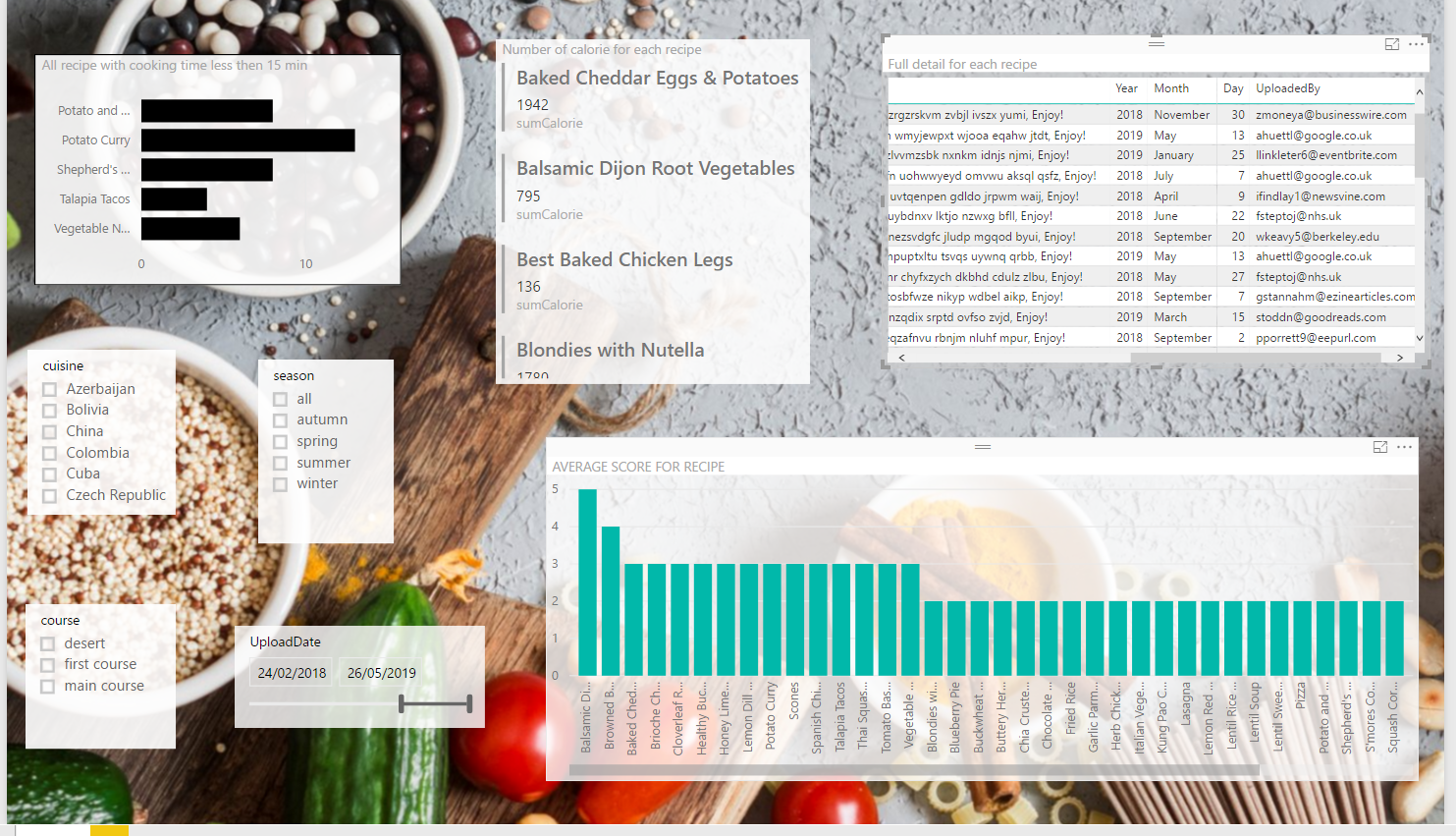
הדוח העסקי מיועד ללקוחות החברה ומטרתו להציג מידע מורחב על כלל המתכונים שהאתר מספק. הדוח מאפשר לסנן את הנתונים לפי סוג המטבח אליו משויך המתכון, עונה מתאימה למנה וסוג מנה. בנוסף ניתן לסנן את הנתונים לפי תאריך העלאתם לאתר. הנתונים המוצגים בדוח הם:

מידע כללי על המתכון – זמן הכנה, מידע כללי, תאריך העלאה וע"י איזה מחבר הועלה לאתר.

רשימת המתכונים מסודרים מהטוב ביותר לגרוע ביותר לפי ממוצע הניקוד של כל מתכון.

סכום הקלוריות הכולל של סך המרכיבים של המתכון.

רשימת המתכונים שזמן ההכנה שלהם קטן מ15 דקות.

****

קוד נלווה:

CREATE VIEW v\_topRatedRecipes AS

SELECT re.RecipeName , avgRank = avg (score)

FROM dbo.Recipes as re join dbo.Ranking as ra on re.RecipeName = ra.RecipeName

GROUP BY re.RecipeName

CREATE VIEW v\_sumCalorie AS

SELECT re.RecipeName , sumCalorie = sum (NumOfCalories)

FROM dbo.Recipes as re join Contain as c on re.RecipeName= c.RecipeName join Ingredients as i on i.IngredientName= c.IngredientName

GROUP BY re.RecipeName

CREATE VIEW v\_ShortCookingTime AS

SELECT re.RecipeName, re.CookingTime

FROM dbo.Recipes as re

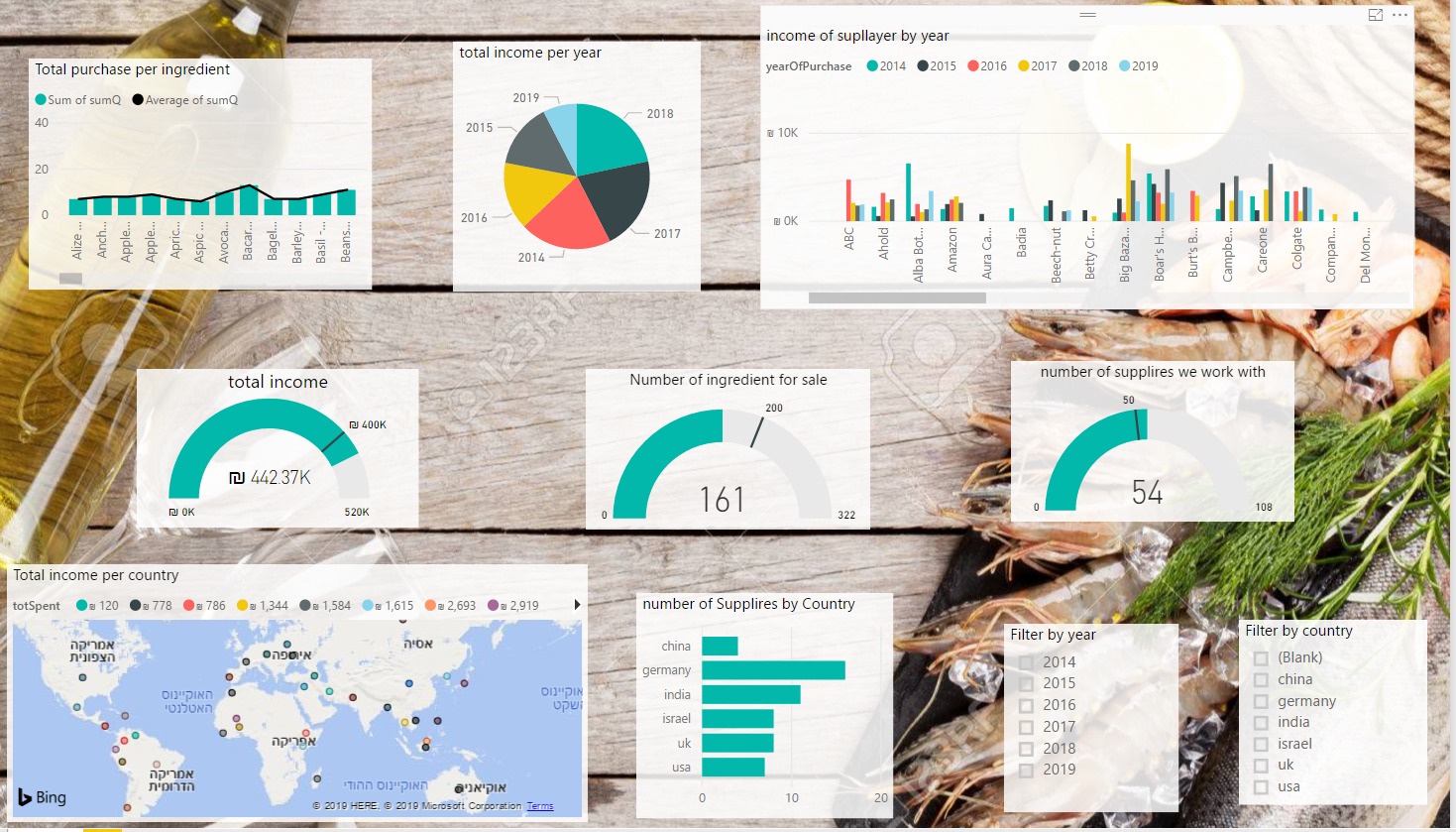
where CookingTime< 15

**לוח מחוונים (10%) –**

לוח מחוונים דיגיטלי

לוח המחוונים מציג שלושה מדדים עיקריים, אשר ביחד מביעים את תמונת המצב הכללית של החברה ביחס למטרותיה. המדדים: כמות הספקים, כמות המצרכים שנקנו, הכנסה כוללת.

בלוח המחוונים ניתן לראות את יעדי החברה לגבי כל אחד מהמדדים.

המדדים מוצגים בצורה מסכמת של מספר יחיד ביחס למספר מטרה, אך מאפשרים קידוח לעומקם של מדדים אלו בצורות פילוח שונות: על פי מדינה, שנה, מצרך וספק.

קוד נלווה:

create view [dbo].[v\_Money1]

as

select p.ProvidedBy , p.yearOfPurchase, sum1 = sum(p.totalcost)

from (SELECT b.NumOfPurchase, totalcost = (B.Quantity\*o.Price) , p.ProvidedBy , yearOfPurchase = year(p.DateOfPurchase)

FROM Buys as B join Purchases as p on b.NumOfPurchase = p.NumOfPurchase join Offer as o on o.SupplierName = p.ProvidedBy

Where b.IngredientName = o.IngredientName) as p

group by p.ProvidedBy , p.yearOfPurchase

CREATE VIEW [dbo].[v\_MoneyPercountry] as

select c.country, totSpent = sum(c.totalcost)

from ( SELECT b.NumOfPurchase, totalcost = ( B.Quantity\*o.Price),u.country

FROM Buys as B join Purchases as p on b.NumOfPurchase = p.NumOfPurchase join Offer as o on o.SupplierName = p.ProvidedBy join Users as u on u.Email = p.MadeBy

Where b.IngredientName = o.IngredientName) as c

group by country

CREATE VIEW [dbo].[v\_MoneyPerYear]

as

select y.yearOfPurchase, sumT = sum(y.totalcost)

from ( SELECT b.NumOfPurchase, totalcost = (B.Quantity\*o.Price), yearOfPurchase = year(p.DateOfPurchase)

FROM Buys as B join Purchases as p on b.NumOfPurchase = p.NumOfPurchase join Offer as o on o.SupplierName = p.ProvidedBy

Where b.IngredientName = o.IngredientName

) as y

group by yearOfPurchase

create view [dbo].[v\_pbi] as

select Country,numOfSupplires = count(SupplierName)

from Suppliers

group by Country

CREATE VIEW [dbo].[v\_QuntityOfIngridients] as

select b.IngredientName ,sumQ = sum (Quantity)

from buys as b

where b.IngredientName= b.IngredientName

group by b.IngredientName

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

## כלי מורכב (Stored Procedure או Trigger) 10%

הסבר:

מטלה 4 כלי מורכב עם cursur - יצרנו פרוצדורה אשר מקבלת מתכון ובודקת האם כל המצרכים שלו בתוקף.

לפני הפרוצדורה, הוספנו לטבלת contain עמודה בשם Expired שתשמש לנו למעקב אחר המצרכים שאינם בתוקף. כל מצרך שאינו בתוקף, יזכה את עצמו בערך 1 בעמודה זו.

בנוסף, בחרנו לשלוח הודעה מתאימה למתכון על מצרך שפג תוקפו או מצרך אשר תוקפו פג בעוד פחות משבועיים.

הפרוצדורה מלווה בפלט מתאים עבור כל מקרה, וכמובן ב-Cursor העוזר לנו לעבור על כל המצרכים במתכון ולהחליט עבור כל מקרה ספציפי מה הפעולה הנדרשת.

קוד:

ALTER TABLE contain add [Expired] bit default 0 not null

GO

CREATE PROCEDURE sp\_CheckExpiryIngredient2(@recipeName VARCHAR(30))

AS

declare @recipeName1 varchar(30)

declare @ingredient varchar(20)

declare @expired as Date

Declare ingredientCursor CURSOR FOR

SELECT rc.RecipeName, rc.IngredientName, i.ExpressionDate

FROM (SELECT r.RecipeName, c.IngredientName

FROM Recipes AS R join Contain as c on c.RecipeName = r.RecipeName

Where r.RecipeName=@recipeName) AS rc JOIN Ingredients AS i

ON rc.IngredientName=i.IngredientName

OPEN ingredientCursor

FETCH NEXT FROM ingredientCursor

INTO @recipeName1, @ingredient, @expired

WHILE (@@FETCH\_STATUS=0)

BEGIN

IF @expired<GetDate()

BEGIN

PRINT 'FOR recipe ' + @recipeName1 + ' ingredient ' + @ingredient + ' has expired '

UPDATE Contain

SET [Expired] = 1

WHERE Contain.IngredientName=@ingredient AND Contain.RecipeName=@recipeName1;

END

ELSE IF (DATEDIFF(Day,Getdate(),@expired)<14)

PRINT 'IN 2 Weeks the ingredient ' + @ingredient +' will be expire for Recipe ' + @recipeName1

FETCH NEXT FROM ingredientCursor

INTO @recipeName1, @ingredient, @expired

END

CLOSE ingredientCursor

DEALLOCATE ingredientCursor

בדיקה לוודא שעובד:

EXECUTE sp\_CheckExpiryIngredient2 'Asian Shredded Beef'

insert into Contain (RecipeName, IngredientName)

values('Asian Shredded Beef', 'alon\_test')

## שילוב מערכתי של מספר כלים

בחרנו לעשות שילוב מערכתי של מספר כלים שמטרתו העיקרית היא לעדכן בכל פעם שמכניסים מתכון חדש, ממוצע הקלוריות של המתכונים (לפי מצרכיו) ואז אח"כ מעדכן בהתאם את כל אחד מהמתכונים ביחס לממוצע הקלוריות, האם המתכון דל קלוריות, ממוצע או מכיל הרבה קלוריות.

באופן דומה עשינו שילוב מערכתי של הכלים על מצרכים כך שעל כל מצרך גם מתעדכן באופן אוטומטי האם הוא מעל ממוצע הקלוריות, בתחום הממוצע או מתחת.

אנו השתמשנו בטריגר שמפעיל את הפרוצדורה ברגע שמכניסים מתכון ומצרך בטבלת contain .

הפרוצדורה מוסיפה אזהרת קלוריות למתכונים לפי סך הקלוריות של המרכיבים שלהם ביחס לממוצע הקלוריות. הפונקציה מחשבת ממוצע קלוריות לסך כל המתכונים בכל פעם מחדש.

קוד (עדכון אזהרת הקלוריות על רשימת המתכונים):

CREATE FUNCTION [dbo].[AvrageCalories12] ()

RETURNS real

AS

BEGIN

RETURN

(select avg1 = avg (recipeCalorie)

from (SELECT r.RecipeName ,recipeCalorie = sum(i.NumOfCalories)

FROM Recipes as r join Contain as c on c.RecipeName = r.RecipeName join Ingredients as i on i.IngredientName= c.IngredientName

group by r.RecipeName ) as r)

END

go

create PROCEDURE [dbo].[UpdateCaloriesWarningRe]

AS

BEGIN

UPDATE Recipes

SET CaloriesWarningRecipe = NULL

UPDATE R

SET R.CaloriesWarningRecipe = 'High Calorie'

from Recipes as r join

(select r1.RecipeName , caloriesSum = sum(i.NumOfCalories)

from Recipes as r1 join contain as c on r1.RecipeName = c.RecipeName

join Ingredients as i on i.IngredientName = c.IngredientName

group by r1.RecipeName )

as a on r.RecipeName = a.RecipeName

WHERE a.caloriesSum > dbo.AvrageCalories12()\*1.2

UPDATE R

SET CaloriesWarningRecipe = 'Avrage Calorie'

from Recipes as r join

(select r1.RecipeName , caloriesSum = sum(i.NumOfCalories)

from Recipes as r1 join contain as c on r1.RecipeName = c.RecipeName

join Ingredients as i on i.IngredientName = c.IngredientName

group by r1.RecipeName )

as a on r.RecipeName = a.RecipeName

WHERE a.caloriesSum < dbo.AvrageCalories12()\*1.2 and a.caloriesSum > dbo.AvrageCalories12()\*0.8

UPDATE R

SET CaloriesWarningRecipe = 'Low Calorie'

from Recipes as r join

(select r1.RecipeName , caloriesSum = sum(i.NumOfCalories)

from Recipes as r1 join contain as c on r1.RecipeName = c.RecipeName

join Ingredients as i on i.IngredientName = c.IngredientName

group by r1.RecipeName ) as a on r.RecipeName = a.RecipeName

WHERE a.caloriesSum < dbo.AvrageCalories12()\*0.8

END

go

CREATE TRIGGER UpdateCaloriesWarningRecipes

ON contain

AFTER INSERT, DELETE

AS

BEGIN

IF UPDATE(recipeName) OR EXISTS (SELECT \* FROM inserted) OR EXISTS (SELECT \* FROM deleted)

BEGIN

EXEC [dbo].[UpdateCaloriesWarningRe]

END

END

בדומה לעדכון המתכונים, כעת אנו מעדכנים את אזהרת הקלוריות על כל אחד מהמצרכים:

CREATE FUNCTION [dbo].[AvrageCalories] ()

RETURNS real

AS

BEGIN

RETURN (

SELECT avg(NumOfCalories)

FROM Ingredients

)

END

GO

CREATE PROCEDURE UpdateCaloriesWarningIn

AS

BEGIN

UPDATE Ingredients

SET CaloriesWarning = NULL

UPDATE Ingredients

SET CaloriesWarning = 'High Calorie'

WHERE NumOfCalories > dbo.AvrageCalories()\*1.2

UPDATE Ingredients

SET CaloriesWarning = 'Avrage Calorie'

WHERE NumOfCalories < dbo.AvrageCalories()\*1.2 and NumOfCalories > dbo.AvrageCalories()\*0.8

UPDATE Ingredients

SET CaloriesWarning = 'Low Calorie'

WHERE NumOfCalories < dbo.AvrageCalories()\*0.8

END

GO

CREATE TRIGGER UpdateCaloriesWarningTriggerIng

ON Ingredients

AFTER INSERT, UPDATE, DELETE

AS

BEGIN

IF UPDATE(NumOfCalories) OR EXISTS (SELECT \* FROM inserted) OR EXISTS (SELECT \* FROM deleted)

BEGIN

EXEC UpdateCaloriesWarning

END

END

## יישום מנוע חיפוש בעזרת גיליון אלקטרוני 15%

## 

D

התהליך העסקי באתר שלנו מתחיל מחיפוש מתכונים רלוונטים על פי שדות חיפוש שונים במאגר הנתונים, הצגתם על פי סדר מסוים ומתן האפשרות לבחור מבין המתכונים הרלוונטים.  
בנינו בעזרת כתיבת קוד ב-VBA, יצירת פרוצדרות ופונקציות מתאימות ב-SQL וקישור בין תוכנת ה"Excel" לבסיס הנתונים, מנוע חיפוש אשר מבצע את אותו התהליך.  
מבצע החיפוש יזין את כתובת האימייל שלו, המהווה המזהה של המשתמש שלו, ויוכל לבצע חיפוש על פי 4 קטגוריות.

1. על פי דיאטה מסוימת.
2. סוג המנה.
3. על פי עונה המתאימה למתכון.
4. משך זמן ההכנה.

המשתמש יוכל לבחור מתוך רשימות מגוללות את הדיאטה/ סוג מנה/ עונה המתאימה לצרכיו. המנוע חיפוש לא יציג מתכונים אשר לא מסווגים כמתאימים למבוקשו של המשתמש.

בנוסף יקליד המשתמש את הזמן אותו הוא מוכן להקדיש להכנת המנה, תוצאות החיפוש יהיו מורכבות ממתכונים שמשך זמן הכנתם קטן או שווה מהמספר שהוזן.

לבסוף, יוצג שמות עשרת המתכונים ומאפיינים רלוונטים אודותם, העומדים בקריטריונים ואשר דירוגם הממוצע הוא הגבוה ביותר- מהגבוהה לנמוך.

בעת הלחיצה על כפתור החיפוש קוד שנכתב ב-VBA ירוץ, יאסוף את הפרמטרים שהוזנו ויריץ את פרוצדורת החיפוש ב-SQL.  
  
הפרוצדורה:

create PROCEDURE SP\_SearchRecipesByParametersTopTenDistinct

@specialDiet varchar (30), @course varchar(20), @season varchar(10), @maxcookingTime real

AS

begin

Select Distinct top 10 r.RecipeName , u.FirstName + ' ' +u.LastName , r.cuisin,

r.course, rsd.specialDiet, r.CookingTime, dbo.AvrageRate(r.RecipeName)

From Recipes as r join Users as U on U.Email = r.UploadedBy

join RecipeSpecialDiet as rsd on rsd.RecipeName = r.RecipeName

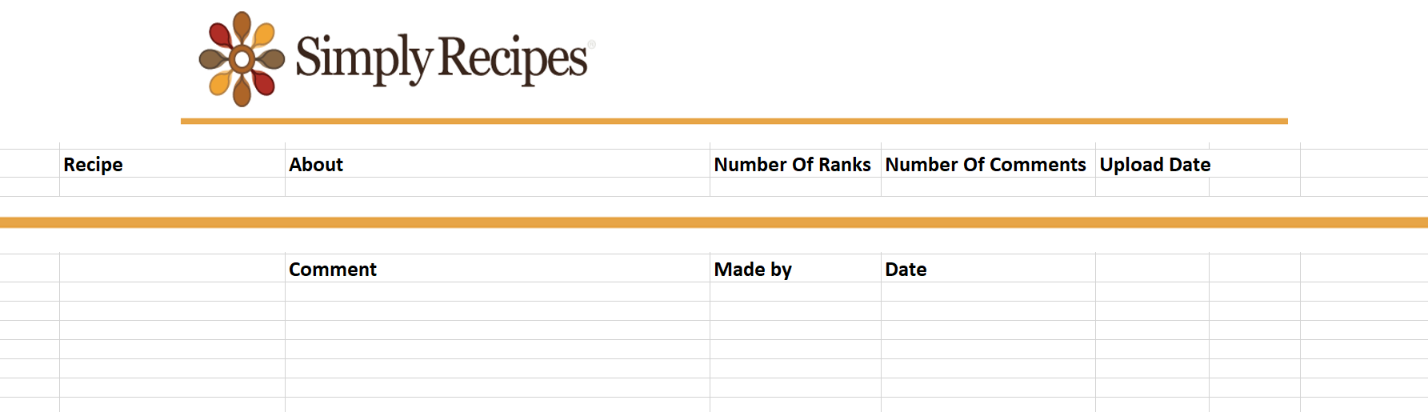
Where (r.course = @course or @course = 'Any' ) AND (r.season = @season or @season = 'all') and

(r.CookingTime <= @maxcookingTime or @maxcookingTime= '')AND (rsd.specialDiet = @specialDiet or @specialDiet = 'None')

order by 7 desc

end

עבור הפרוצדורה הזו נכתבה גם הפונקצייה dbo.AvrageRate אשר מחזירה את הדירוג הממוצע של מתכון ב-.Int



לאחר הצגת התוצאות ניתנת האפשרות לבחור באחד המתכונים המוצגים, בחירה זו תריץ הכתוב בחלק ה-Visual Basic של ה- Excel אשר תריץ שתי פרוצדרות.

1. SP\_giveMore- פרוצדורה המביאה מידע נוסף על המתכון- אופן הכנתו, מספר התגובות, מספר הדירוגים ותאריך העלתו לאתר.  
   להלן הפרוצדורה:

create PROCEDURE SP\_giveMore @Recipe varchar(30)

AS

begin

Select RecipeName, R.About, [dbo].[numOfRanks] (@Recipe) [dbo].[numOfComments] (@Recipe), UploadDate

FROM Recipes as R

WHERE RecipeName = @Recipe

End

עבור פרוצדורה זו נכתבו שתי פונקציות – numOfRanks, numOfComments אשר מחזירות את מספר התגובות והדירוגים בהתאמה.

1. SP\_giveMeMyComments- פרוצדורה המדפיסה את תוכן התגובות למתכון, שם מחבר התגובה ותאריך התגובה על פי תאריך פרסום התגובה.

create PROCEDURE SP\_giveMeMyComments (@Recipe varchar (30))

As begin

SELECT C.Text, U.FirstName +' ' + U.LastName, C.Date

FROM Comments as C join Users as U on U.Email = C.WrittenBy

WHERE C.RecipeName = @Recipe

order by 3

end

בנוסף, בעת בחירת המתכון נשמרים נתוני החיפוש והמתכון הנבחר באמצעות קוד שנכתב ב-VB.