|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מס' קבוצה | אתר | | | תאריך הגשה |
| 37 | [www.kickstarter.com](http://www.kickstarter.com) | | | 12/06/2021 |
| חברי הצוות - מספרי ת.ז | | | | |
| 315523456 | | 208489674 | 316426857 | |

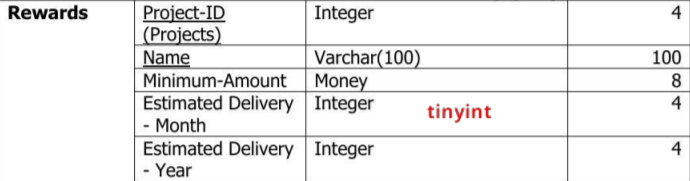
## פרויקט בסיסי נתונים – חלק ג'

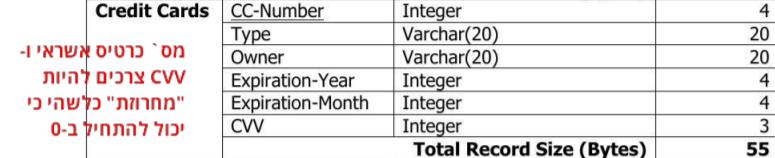
## מטלת חובה מקדימה – תיקון ה-ERD והעשרת בסיס נתונים

## תרשים ה-ERD מחלק ב'

Diagram, engineering drawing, schematic

Description automatically generated





**התרשימים והטבלאות עליהם הסתמכנו בחלק ג':**

**Diagram, schematic

Description automatically generated**

* **Users** (E-mail, Name, Password, Join-Date, Country)
* **Websites** (User E-mail (Users), Website)
* **Projects** (Project-ID, Title, Sub-title, Sub-Category (Categories), Funding-Goal, Duration, Start Date, Country, Creator E-mail (User))
* **Categories** (Sub-Category, Category)
* **Rewards** (Project-ID (Projects), Name, Minimum Amount, Estimated Delivery – Month, Estimated Delivery – Year)
* **Pledges** (Pledge-ID, DT, Amount, User E-mail (User), CC-Number (Credit Cards), {Project-ID, Name} (Rewards), {Address Of User E-mail, Address Number} (Addresses))
* **Credit Cards** (CC-Number, Type, Owner, Expiration, CVV)
* **Searches** (IP Address, DT, Text, Sub-Category, Collection, Country, Sorted By, E-mail (User))
* **Comments** (Project-ID (Projects), Comment Number, DT, Text, {Comment On Project-ID, Commented On Comment Number} (Comments), Commentor (Users))
* **Retrieves** (Project-ID (Projects), {IP Address, DT} (Searches))
* **Collaborated by** (Project-ID (Projects), User E-mail (Users))
* **In Favorites of** (Project-ID (Projects), User E-mail (Users), DT)
* **Follows** (E-mail - Follower (Users), E-mail - Followed (Users), DT)
* **Saves** (User E-mail (Users), CC-Number (Credit Cards))
* **Addresses** (User E-mail (Users), Number, Zipcode, Country, City, Street - Name, Street - Number)

הערות:

* בקשר USES אנחנו מניחים שיש כרטיסי אשראי שכמה אנשים שונים משתמשים בהם.
* לכל פרויקט יש קטגוריה ותת קטגוריה, אך מפני שתת קטגוריה מגדירה חד-ערכית את הקטגוריה, ישנה הפרת נרמול מסדר שלישי, ולכן תת קטגוריה הינה המפתח הראשי של טבלת קטגוריות.
* בטבלת Projects שדה Country מתייחס למדינה בה הפרויקט מנוהל, ואינו קשור לCountry של המשתמש שמנהל את הפרויקט, ולכן אין הפרת נרמול.
* בטבלת Searches, אין תלות בין IP-Address ל-Email (User) מפני שמשתמש בודד יכול לבצע חיפוש ממחשבים שונים ולכן יהיו לו IP-Address שונים.

|  |  |  |  |
| --- | --- | --- | --- |
| **Table** | **Field** | **Data Type** | **Size (Bytes)** |
| **Users** | E-mail | Varchar(30) | 30 |
| Name | Varchar(20) | 20 |
| Password | Varchar(20) | 20 |
| Join-Date | DateTime | 8 |
| Country | Varchar(40) | 40 |
| **Total Record Size (Bytes)** | | **118** |
| **Websites** | User E-mail (Users) | Varchar(30) | 30 |
| Website | Varchar(30) | 30 |
| **Total Record Size (Bytes)** | | **60** |
| **Projects** | Project-ID | Integer | 4 |
| Title | Varchar(100) | 100 |
| Sub-Title | Varchar(100) | 100 |
| Sub-Category (Categories) | Varchar(20) | 20 |
| Funding-Goal | Money | 8 |
| Duration | Integer | 4 |
| Start-Date | DateTime | 8 |
| Country | Varchar(40) | 40 |
| Creator E-mail (Users) | Varchar(30) | 30 |
| **Total Record Size (Bytes)** | | **314** |
| **Categories** | Sub-Category | Varchar(20) | 20 |
| Category | Varchar(20) | 20 |
| **Total Record Size (Bytes)** | | **40** |
| **Rewards** | Project-ID (Projects) | Integer | 4 |
| Name | Varchar(100) | 100 |
| Minimum-Amount | Money | 8 |
| Estimated Delivery - Month | Tinyint | 1 |
| Estimated Delivery - Year | Smallint | 2 |
| **Total Record Size (Bytes)** | | **115** |
| **Pledges** | Pledge-ID | Integer | 4 |
| DT | DateTime | 8 |
| Amount | Money | 8 |
| User E-mail (Users) | Varchar(30) | 30 |
| CC-Number (Credit Cards) | Integer | 4 |
| Project-ID (Rewards) | Integer | 4 |
| Name (Rewards) | Varchar(100) | 20 |
| Address of User E-mail (Addresses) | Varchar(30) | 30 |
| Address Number (Addresses) | Integer | 4 |
| **Total Record Size (Bytes)** | | **197** |
| **Credit Cards** | CC-Number | Integer | 4 |
| Type | Varchar(20) | 20 |
| Owner | Varchar(20) | 20 |
| Expiration - Month | Tinyint | 1 |
| Expiration - Year | Smallint | 2 |
| CVV | Varchar(3) | 3 |
| **Total Record Size (Bytes)** | | **50** |
| **Searches** | IP Address | Varchar(15) | 15 |
| DT | DateTime | 8 |
| Text | Varchar(280) | 280 |
| Sub-Category | Varchar(20) | 20 |
| Country | Varchar(40) | 40 |
| Sorted By | Varchar(20) | 20 |
| E-mail (Users) | Varchar(30) | 30 |
| **Total Record Size (Bytes)** | | **393** |
| **Comments** | Project-ID (Projects) | Integer | 4 |
| Comment Number | Integer | 4 |
| DT | DateTime | 8 |
| Text | Varchar(280) | 280 |
| Comment on Project-ID (Comments) | Integer | 4 |
| Commented On Comment Number  (Comments) | Integer | 4 |
| Commentor (Users) | Varchar(30) | 30 |
| **Total Record Size (Bytes)** | | **334** |
| **Retrieves** | Project-ID (Projects) | Integer | 4 |
| IP Address (Searches) | Varchar(15) | 15 |
| DT (Searches) | DateTime | 8 |
| **Total Record Size (Bytes)** | | **27** |
| **Collaborated by** | Project-ID (Projects) | Integer | 4 |
| User E-mail (Users) | Varchar(30) | 30 |
| **Total Record Size (Bytes)** | | **34** |
| **In Favorites of** | Project-ID (Projects) | Integer | 4 |
| User E-mail (Users) | Varchar(30) | 30 |
| DT | DateTime | 8 |
| **Total Record Size (Bytes)** | | **42** |
| **Follows** | E-mail - Follower (Users) | Varchar(30) | 30 |
| E-mail - Followed (Users) | Varchar(30) | 30 |
| DT | DateTime | 8 |
| **Total Record Size (Bytes)** | | **68** |
| **Saves** | User E-mail (Users) | Varchar(30) | 30 |
| CC-Number (Credit Cards) | Integer | 4 |
| **Total Record Size (Bytes)** | | **34** |
| **Addresses** | User E-mail (Users) | Varchar(30) | 30 |
| Number | Integer | 4 |
| Zipcode | Varchar(20) | 20 |
| Country | Varchar(40) | 40 |
| City | Varchar(20) | 20 |
| Street - Name | Varchar(20) | 20 |
| Street - Number | Integer | 4 |
| **Total Record Size (Bytes)** | | **138** |

**הסברים על השינויים:**

* הוספנו הנחה על מזהה ישות Comment.
* הוספנו בקשר בין לקוח לפרויקט שדה DT שמציין את התאריך והזמן בו הלקוח הוסיף את הפרויקט למועדפים.
* שינינו data type לפי ההערות במקרים הבאים:
  + Estimated Delivery – Month / Year -> Tinyint / Smallint
  + CCV -> Varchar(3)

## מטלה 1 (25%) – שאילתות

בשאילתות בהן אין הגבלה למספר הרשומות, הצגנו בפלט את 10 הרשומות הראשונות.

# שתי שאילתות SELECT ללא קינון 5%

1. החברה רוצה לבחון אילו מדינות הניבו לאחרונה את סכום התרומות הגדול ביותר ולאחר מכן תשקול האם לאמץ את מדיניות החברה במדינות הללו למדינות אחרות בשביל להביא להגדלה בתרומות. לשם כך נכתבה שאילתה: הצג את 5 המדינות המובילות אשר אצלן משתמשים שנרשמו אחרי 2017 תרמו הכי הרבה כסף כאשר עבור אותה מדינה היו מעל 5 תרומות שונות.

SELECT TOP 5 A.Country, Users = COUNT(DISTINCT U.[E-mail]),

[Number of Pledges] = COUNT(P.[Pledge-ID]), [Total Pledges] = SUM(P.Amount)

FROM Users AS U JOIN Addresses AS A ON U.[E-mail]=A.[User E-mail]

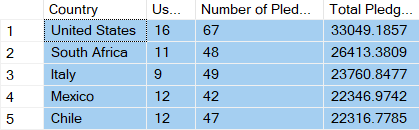
JOIN Pledges AS P ON U.[E-mail]=P.[User E-mail]

WHERE year(U.[Join-Date]) > 2017

GROUP BY A.Country

HAVING COUNT(P.[Pledge-ID]) > 5

ORDER BY [Total Pledges] DESC



2. קיקסטארטר רוצה לסווג את משתמשיה האקטיביים (אשר תרמו לפרויקט אחד לפחות בשנתיים האחרונות) לפי סכום התרומות שלהם.

Select U.[E-mail],PledgeAmount = Sum(Pl.Amount),

Rank = Case When Sum(Pl.Amount)<100 Then 'F'

When Sum(Pl.Amount)>=100 and Sum(Pl.Amount)<500 then 'D'

When Sum(Pl.Amount)>=500 and Sum(Pl.Amount)<1000 then 'C'

When Sum(Pl.Amount)>=1000 and Sum(Pl.Amount)<1500 then 'B'

When Sum(Pl.Amount)>=1500 and Sum(Pl.Amount)<2500 then 'A'

When Sum(Pl.Amount)>=2500 then 'S' end

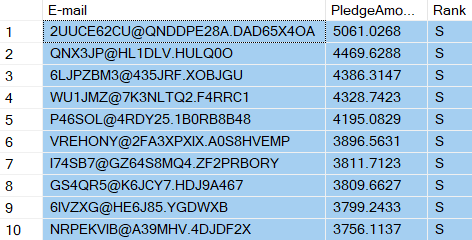
From Users as U Join Pledges as Pl

On U.[E-mail] = Pl.[User E-mail]

Where year(getDate()) - year(Pl.DT) <=2

Group By U.[E-mail]

Order By 2 Desc



# שתי שאילתות SELECT מקוננות 10%

1. החברה מתכננת לצאת בקמפיין פרסום חדש. לשם הצגת נתונים בפרסומות החברה רוצה לבדוק מהו אחוז הפרויקטים שהשיגו את יעד המימון שלהם.

SELECT [Success Ratio] =

(

SELECT Sum = COUNT(\*)

FROM (

SELECT Pr.[Project-ID], Received = SUM(Pl.Amount) - Pr.[Funding-Goal]

FROM Projects as Pr LEFT JOIN Rewards as R ON

Pr.[Project-ID] = R.[Project-ID]

JOIN Pledges as Pl ON

(R.Name = Pl.Name AND R.[Project-ID] = Pl.[Project-ID])

GROUP BY Pr.[Project-ID], Pr.[Funding-Goal]

HAVING SUM(Pl.Amount) - Pr.[Funding-Goal] >= 0

) as Q

)

/

CAST(COUNT(DISTINCT [Project-ID]) as Float)

FROM Projects



# 2. בכדי ליצור חווית משתמש מוגברת אתר קיקסטארטר יצא במיזם משתמשים פופולריים. החברה רוצה לבדוק: מתוך 10 המדינות בעלות כמות התרומות הגדולה ביותר, מי הם הלקוחות שבשנתיים האחרונות תרמו סכום כולל של מעל 1000$.

Select [MvpCountry].[E-mail],Users.Country,AmountOfPledges = Sum(Pl.Amount)

From (

Select us.[E-mail],Us.name, Us.Country

From Users As Us

Where Us.Country IN

(Select Distinct temp.Country

From Users As U join (Select top 10 Us.Country, AmountOfPledges = count(Distinct [Pledge-ID])

From Pledges As Pl Join Users As Us On Pl.[User E-mail] = Us.[E-mail]

Group By Us.Country

Order By 2 desc) As temp

On temp.Country=U.country)

Group by us.[E-mail],Us.Name,Us.Country

) As MvpCountry

Join Users On Users.[E-mail] = MvpCountry.[E-mail] Join Pledges As Pl

On pl.[User E-mail] = MvpCountry.[E-mail]

Where year(getDate()) - year(Pl.DT) <= 2

Group By [MvpCountry].[E-mail],Users.country

Having Sum(Pl.Amount) > 1000

Order By 3 DESC



# שתי שאילתות מקוננות תוך שימוש במרכיבים נוספים 10%

1. עבור הפרויקטים שכמעט השיגו את סכום המטרה (Funding Goal, 90-100%), הורד את הסכום המינימלי עבור כל המתנות המוצעות בפרויקט זה.

שימוש אפשרי בשאילתה הוא כאשר החברה רוצה לעודד תורמים לתמוך בפרויקט שקרוב להשגת היעד, היא תעשה זאת על ידי הנגשת המתנות בכך שתוריד את המחיר המינימלי הנדרש עבורן.

UPDATE Rewards

SET [Minimum-Amount] = [Minimum-Amount]\*0.9

WHERE Rewards.[Project-ID] IN

(

SELECT P1.[Project-ID]

FROM Pledges AS P1 JOIN Projects as P2 ON P1.[Project-ID]=P2.[Project-ID]

GROUP BY P1.[Project-ID], P2.[Funding-Goal]

HAVING SUM(P1.Amount) BETWEEN P2.[Funding-Goal]\*0.9 AND P2.[Funding-Goal]

)

לבדיקת שאילתת ה- UPDATE השתמשנו בשאילתת SELECT לפני ואחרי השימוש בשאילתת ה- UPDATE.

SELECT Name, [Minimum-Amount]

FROM REWARDS

WHERE [Project-ID] IN (

SELECT P2.[Project-ID]

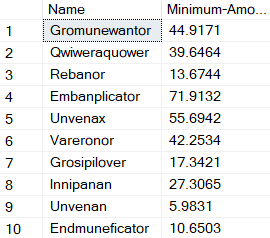
FROM Pledges AS P1 JOIN Projects as P2 ON P1.[Project-ID]=P2.[Project-ID]

GROUP BY P2.[Project-ID], P2.[Funding-Goal], P2.[Funding-Goal]

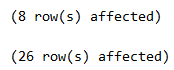
HAVING SUM(P1.Amount) BETWEEN P2.[Funding-Goal]\*0.9 AND P2.[Funding-Goal]

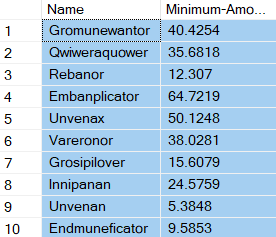
)

לפני:



אחרי:





2. במטרה לזהות פרויקטי "רפאים", כלומר פרויקטים לא פעילים, החברה רוצה לבדוק מהם הפרויקטים שלא התבצעה אליהם אף תרומה.

SELECT [Project-ID], Title, [Start-Date]

FROM PROJECTS

EXCEPT

SELECT Pr.[Project-ID], Pr.Title, Pr.[Start-Date]

FROM Projects as Pr JOIN Rewards as R ON Pr.[Project-ID] = R.[Project-ID]

JOIN Pledges as Pl ON (R.Name = Pl.Name AND R.[Project-ID] = Pl.[Project-ID])

ORDER BY [Start-Date] ASC

(פלט ריק)

## מטלה 2 (25%) – יישומי כלים מתקדמים

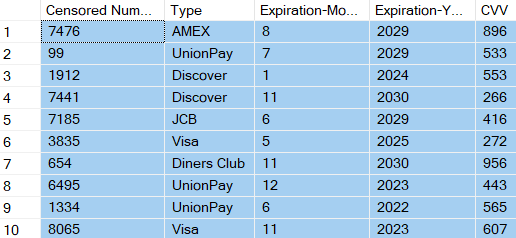
# View 5%

View זה מסתיר נתוני כרטיס אשראי ממשתמשים שאינם מורשים. הוא מציג רק את 4 הספרות האחרונות של כרטיס האשראי ומסתיר את שם הבעלים.

CREATE VIEW V\_CensoredCC

SELECT [Censored Number] = [CC-Number] % 10000, [Type], [Expiration-Month], [Expiration-Year], CVV

FROM [Credit Cards]



# פונקציות (Functions) 10%

1. הפונקציה מקבלת מזהה פרויקט ומחזירה את סכום התשורות עבור פרויקט בודד.

Create Function InvestmentAmount ( @ProjectID int )

Returns int

as Begin

Declare @OutPut\_TotalAmount int

Select @OutPut\_TotalAmount = Sum(Pl.Amount)

From Projects as Pr Join Pledges as Pl

On Pr.[Project-ID] = Pl.[Project-ID]

Where Pr.[Project-ID] = @ProjectID

Group by Pr.[Project-ID]

Return @OutPut\_TotalAmount

End

הקוד שמדגים את השימוש בפונקציה: מהו סכום התרומות עבור פרויקט מס' 5.

Select Amount = dbo.InvestmentAmount([Project-ID])

FROM Projects

WHERE [Project-ID]=5



2. הפונקציה PledgesByCC מקבלת כקלט מספר כרטיס אשראי ומחזירה את פרטי כל התרומות שהתבצעו באמצעות כרטיס אשראי זה. כדוגמא לשימוש בפונקציה, הראינו שאילתת SELECT שמחזירה את פרטי המשתמשים והכתובות שעשו שימוש בכרטיס אשראי מסויים.

CREATE FUNCTION PledgesByCC(@CCN integer)

RETURNS TABLE

AS RETURN

SELECT P.[Pledge-ID], P.DT, P.Amount, P.[Address of User E-mail], P.[Address Number]

FROM Pledges as P JOIN [Credit Cards] as C ON P.[CC-Number] = C.[CC-Number]

WHERE P.[CC-Number] = @CCN

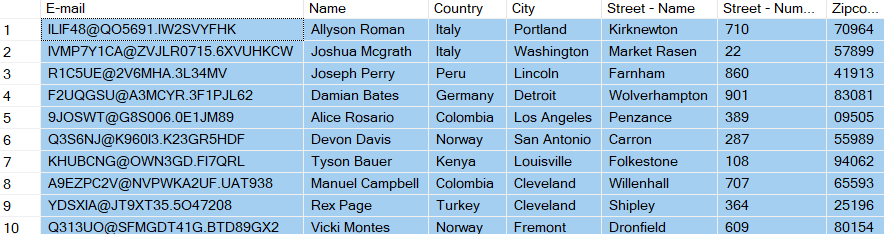
הקוד שמדגים את השימוש בפונקציה: הצג את פרטי הכתובות והמשתמשים המשוייכים לכרטיס אשראי מס' 13110654.

SELECT U.[E-mail], U.Name, A.Country, A.City, A.[Street - Name], A.[Street - Number], A.Zipcode

FROM dbo.PledgesByCC(13110654) as P JOIN Addresses as A

ON (P.[Address of User E-mail] = A.[User E-mail] AND P.[Address Number] = A.Number)

JOIN Users as U ON A.[User E-mail] = U.[E-mail]



# Trigger פשוט 5%

ראשית, הוספנו בטבלת Projects שדה ‘Amount Received’ שמהווה את הסכום שנאסף עד כה בפרויקט. הטריגר UpdateAmountReceived פועל בכל פעם שמתבצע שינוי בטבלת Pledges, וסוכם מחדש את כל התרומות שהתבצעו עבור הפרויקטים אשר תרומות אליהם מטבלת Pledges שונו.

ALTER TABLE Projects

ADD [Amount Received] float Null

CREATE TRIGGER UpdateAmountReceived

ON Pledges

FOR INSERT, DELETE, UPDATE

AS

UPDATE Projects

SET [Amount Received] = (

SELECT SUM(Pl.Amount)

FROM Projects as Pr JOIN Rewards as R ON Pr.[Project-ID] = R.[Project-ID]

JOIN Pledges as Pl ON (R.[Project-ID] = Pl.[Project-ID] AND R.Name = Pl.Name)

WHERE Pr.[Project-ID] = Projects.[Project-ID]

)

WHERE [Project-ID] IN (

SELECT DISTINCT [Project-ID] FROM INSERTED

UNION

SELECT DISTINCT [Project-ID] FROM DELETED

)

הקוד שמדגים את השימוש בטריגר:

SELECT [Amount Received]

FROM Projects

WHERE [Project-ID]=110

UPDATE Pledges

SET Amount = 2000

WHERE [Pledge-ID]=1

SELECT [Amount Received]

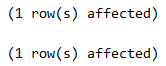
FROM Projects

WHERE [Project-ID]=110

לפני:



אחרי (לפרויקט 110 יש תרומה מס' 1 שסכומה 2000$. עדכנו את הסכום להיות 3000 ולכן השדה המחושב גדל ב- 1000$):





# פרוצדורה שמורה ((Stored Procedure פשוטה 5%

מטרת הפרוצדורה – הוספת ימים למשך הזמן של הפרויקט עד להשגת היעד. פעולה זו תבוצע רק על ידי בעלי הרשאות מסוימות שכן היא משפיעה בצורה מהותית על עתיד הפרויקט וסיכוייו להצלחה.

CREATE PROCEDURE sp\_AddDuration @Days int, @ProjectID int

AS BEGIN

UPDATE Projects

SET Duration = Duration + @Days

WHERE [Project-ID]=@ProjectID

END

הקוד שמדגים את השימוש בפונקציה:

SELECT [Project-ID], Duration

FROM Projects

WHERE [Project-ID]=1

EXECUTE sp\_AddDuration 5,1

Before: After:

## מטלה 3 (20%) – כלים להצגת נתונים

הדו"ח העסקי מיועד לישיבת סיכום שנת 2020 עם סמנכ"ל שיווק.

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

## שאילתות עסקיות המשלבות Window Functions 10%

1. השאילתה נותנת דירוג לכל תת-קטגוריה לפי סכום התרומות שהתקבלו עבור פרויקטים באותה תת-קטגוריה. הדירוג הוא כפול: ישנו דירוג של כלל תתי-הקטגוריות וישנו דירוג פנימי בתוך הקטגוריה המתאימה.

Select [General Rank] = RN.[Rows By Pl Amount],RN.Category,Pr.[Sub-Category],

[Category Rank] = Rank() Over(Partition by RN.Category Order By Sum( RN.SumOfPledges ) DESC),RN.SumOfPledges

From(

Select Cat.Category,Cat.[Sub-Category],SumOfPledges = Sum( pl.Amount ), [Rows By Pl Amount] = Row\_Number() Over( Order By Sum( pl.Amount ) DESC )

From Projects as Pr Join Pledges as Pl

On pr.[Project-ID] = pl.[Project-ID] Join Categories as Cat

On Cat.[Sub-Category] = Pr.[Sub-Category]

Where Pr.[Start-Date] >= 2020

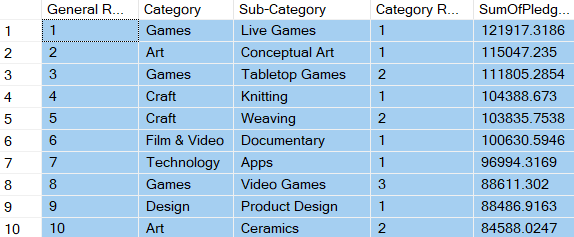
Group By Cat.Category,Cat.[Sub-Category]

) As RN Join Projects As Pr

On Pr.[Sub-Category]= RN.[Sub-Category]

Group By [Rows By Pl Amount],RN.Category,Pr.[Sub-Category],RN.SumOfPledges

Order By 1



1. החברה החליטה לצאת במבצע מימון לסניפיה השונים בעולם. מפני שהחברה דוגלת במצויינות, המימון ילך רק לסניף (עיר) המוביל בכל מדינה. לשם כך, בדקה החברה מהו הסניף המוביל (לפי סכום התרומות שהתקבלו באותה עיר) בכל מדינה. את כל הסניפים המובילים במדינתם, חילקה החברה ל- 15 "רמות", כאשר ככל שהרמה יותר נמוכה (דירוג גבוה) הסניף יקבל חבילת מימון איכותית יותר.

Select CCr.Country,CCR.City,CCR.[Pledge Amount], [Package Number] = Ntile(15) Over(Order By CCR.[Pledge Amount] DESC)

From (

Select U.Country,ad.City,[Pledge Amount] = Sum(Pl.Amount),[Rank] = Dense\_Rank() Over(Partition by U.Country Order by Sum(Pl.Amount))

From Projects as Pr Join Pledges as Pl

On pr.[Project-ID] = pl.[Project-ID] Join Categories as Cat

On Cat.[Sub-Category] = Pr.[Sub-Category] Join Users as U On

U.[E-mail] = Pl.[User E-mail] Join Addresses as Ad

On Ad.[User E-mail] = Pl.[User E-mail]

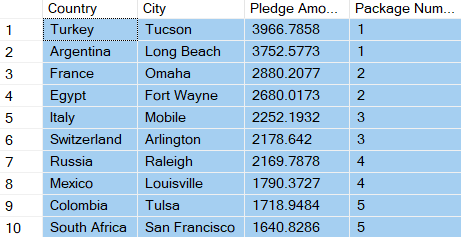
Where year(Pr.[Start-Date]) = 2020

Group By U.Country,Ad.City

) as CCR

Where CCR.[Rank] = 1

Group By CCR.Country,City,CCR.[Pledge Amount]



## שילוב מערכתי של מספר כלים 10%

בקוד זה ניתן לראות פרוצדורה שמורה שמשתמשת בפונקציה כדי לשנות טבלה, וכתוצאה מכך מופעל טריגר.

הסיפור הכללי: החברה רוצה לאפשר למשתמשים להקטין/ להגדיל את הסכום המינימלי לתרומה (Minimum Amount) עבור כל תשורה (Reward), בתנאי שלתשורה זו לא התבצעו עוד תרומות. בנוסף, מתחזקת החברה נתון על כל פרויקט שאומר מהו הסכום המינימלי לתשורה מבין כל התשורות של אותו פרויקט (MinReward).

לשם כך נכתבו:

1. פונקציה שמקבלת את מזהה התשורה (מס' פרויקט + שם התשורה) ובודקת כמה תרומות התבצעו לאותה תשורה (שימושי גם במצבים אחרים).
2. פרוצדורה שמורה שמקבלת את האחוז שבו רוצים לשנות את הסכומים המינימליים לכל התשורות (1.1 🡨 110%) ואת מזהה הפרויקט עבורו רוצים לעשות זאת. הפרוצדורה משנה את הסכום המינימלי לכל תשורה (Minimum-Amount), ובתנאי שלא התבצעו תרומות לתשורה זו.
3. טריגר שמעדכן את סכום התשורה המינימלי בפרויקט (MinReward) בכל פעם שמתבצע שינוי בתשורות של אותו פרויקט.

CREATE FUNCTION PledgesForReward(@PID int, @N varchar(100))

RETURNS Integer

AS BEGIN

DECLARE @Pledges Integer

SELECT @Pledges = COUNT(DISTINCT P.[Pledge-ID])

FROM REWARDS as R LEFT JOIN Pledges as P ON (R.[Project-ID] = P.[Project-ID] AND R.Name = P.Name)

WHERE P.[Project-ID] = @PID AND R.Name = @N

GROUP BY R.[Project-ID], R.Name

RETURN @Pledges

END

ALTER TABLE Projects

ADD MinReward Real

CREATE TRIGGER UpdateMinReward

ON Rewards

FOR INSERT, UPDATE, DELETE

AS

UPDATE Projects

SET MinReward = (SELECT MIN([Minimum-Amount])

FROM Rewards

WHERE Projects.[Project-ID] = Rewards.[Project-ID]

)

WHERE [Project-ID] IN (SELECT DISTINCT [Project-ID]

FROM INSERTED

UNION

SELECT DISTINCT [Project-ID]

FROM DELETED

)

CREATE PROCEDURE SP\_UpdateMinAmount @UpdateBy Real, @PID int

AS

UPDATE Rewards

SET [Minimum-Amount] = [Minimum-Amount] \* @UpdateBy

WHERE Rewards.[Project-ID] = @PID

AND (dbo.PledgesForReward(Rewards.[Project-ID], Rewards.Name)) IS NULL

קוד לבדיקה:

SELECT \*

FROM Projects

WHERE [Project-ID] = 73

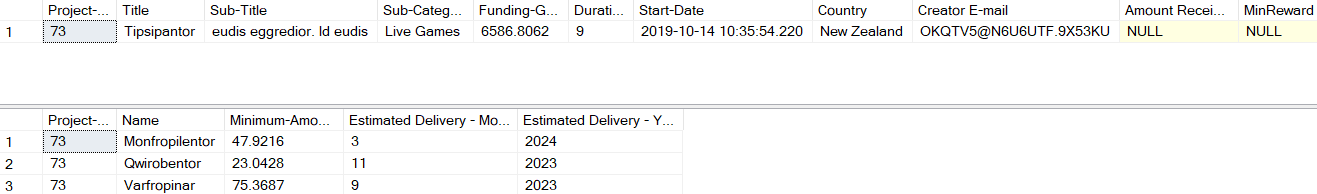
SELECT \*

FROM Rewards

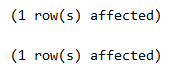
WHERE [Project-ID] = 73

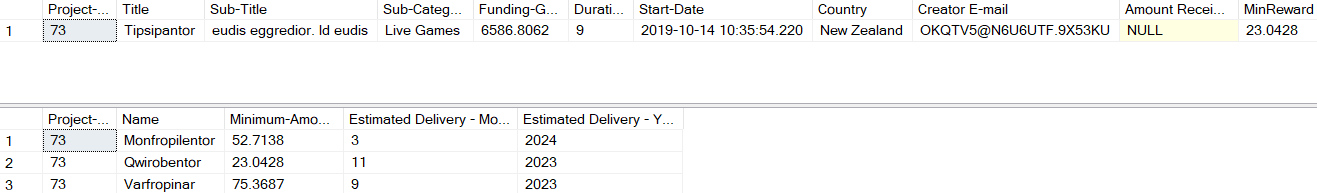
EXECUTE SP\_UpdateMinAmount 1.1, 73

לפני הביצוע:



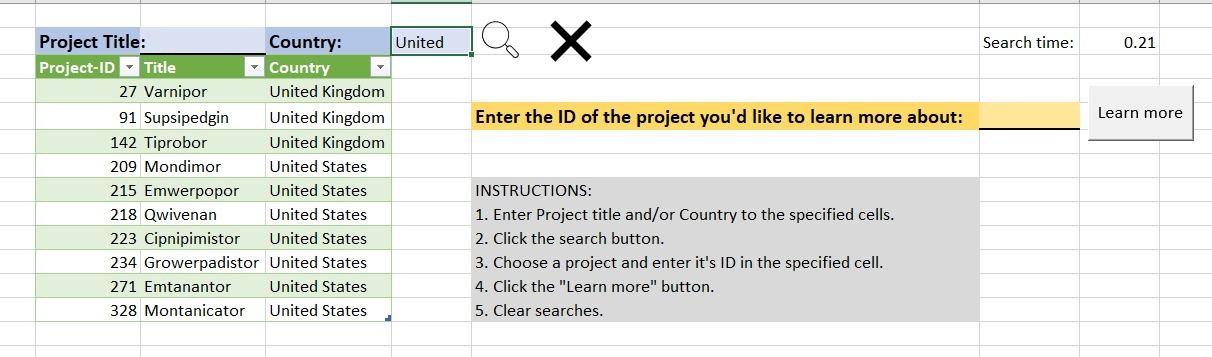
אחרי (עודכן הסכום המינימלי לתשורה הראשונה ועודכנה התשורה המינימלית של הפרויקט):

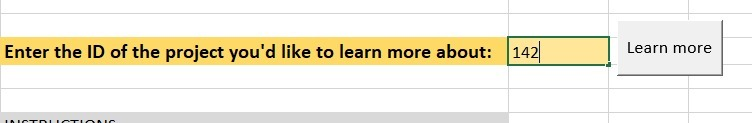




## יישום מנוע חיפוש בעזרת גיליון אלקטרוני 15%

מצורף קובץ אקסל.







# בהצלחה לנו!