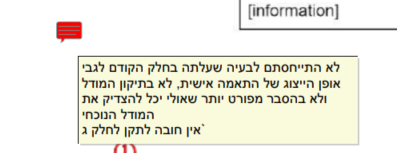
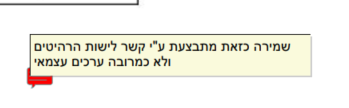
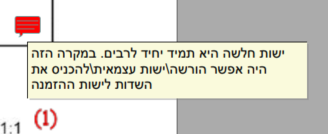
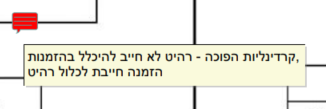
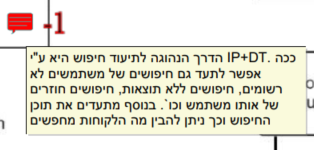
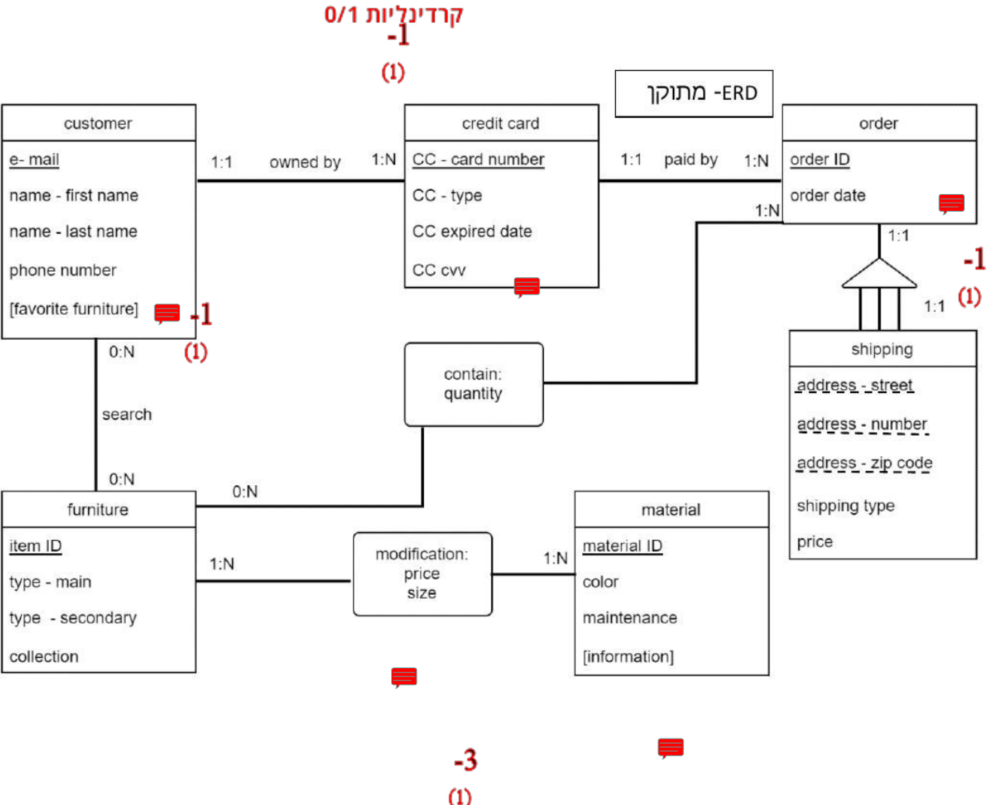
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מס' קבוצה | אתר | | | תאריך הגשה |
| 45 | www.boconcept.com | | | 21/6/22 |
| חברי הצוות - מספרי ת.ז | | | | |
| 315421883 | | 207935255 | 207128513 | |

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



נספח 1 ERD מחלק ב'- הערות

**ERD סופי לאחר תיקון הערות-**

Diagram

Description automatically generated

עיקרי התיקונים-

* איחוד הישות החלשה לתוך הישות order.
* תיקון הקרדינליות בין הרהיט להזמנות
* הוספת ישות חיפוש

טבלת הנחות הסברים-

|  |  |  |
| --- | --- | --- |
| **מרכיב** | **הנחות** | **הערות נוספות** |
| Entity: Customer   * [favorite furniture] | * favorite furniture - האתר מאפשר ללקוח לשמור רהיטים שאהב לצורך הזמנה עתידית, מאפשר איפיון טעם לקוח וכך לטרגט אותו לפרסומות/הנחות ייעודיות לו. |  |
| הקשר בין "Credit Card" ל"Customer" | * האתר דורש שבעת הרישום הלקוח ימסור מספר כרטיס אשראי. כמו כן, האתר מאפשר שיוך של מספר כרטיסי אשראי לכל לקוח – לפיכך הקרדינאליות בין "Credit Card" ל"Customer" הוגדרה כ- 1:1 ו-N:1. |  |
| Entity: material   * material ID * color | * לכל סוג של חומר יש אך ורק צבע אחד מכיוון שהצבע הוא חלק בלתי נפרד מתכונות החומר. | לדוגמה: עץ אלון שחור יהיה בעל material ID שונה מעץ אלון לבן. |
| הגדרת קדינאליות בין  "Credit Card" ל"Customer" | * אין אפשרות לקנות באתר ללא כרטיס אשראי לכן הקרדינאליות בין הלקוח לכרטיס האשראי היא N:1. |  |
| Entity: search | * לכל חיפוש יש תוצאה מכיוון שהחיפוש נשמר כאשר המשתמש מקליק על הרהיט   . | \*חיפוש של רהיט שלא קיים לא נשמר בטבלאות. |
| Entity: material   * information | * הנחת יסוד יכול להיות רהיט שמורכב מכמה חומרים שונים ובנוסף גם מיוצר בכמה מדינות מכיוון שרהיט מורכב חלקים רבים. |  |
| Entity: furnitures   * collection | * יש אוספים שמכילים רק סוג רהיט אחד מפני שהאוסף זה ליין של סוג רהיט אחד |  |

## עיצוב לוגי (מודל נתונים טבלאי)

**Costumers (E-mail,** firstName, lastName , phoneNumber**)**

**Favorites ( E-mail (Costumers)**, **itemID (Furnitures) )**

**CreditCards (card number**, type, expired date, cvv, E-mail (Costumers ))

**Furnitures ( itemID,** mainType, secondaryType, collection)

**Orders** ( **orderID**, orderDate, card number (CreditCards), country, street, number, zipCode, shippingType, price )

**Contains** (**orderID (Orders)**, **itemID(Furnitures),** quantity)

**Search**(**IP adress**, **DT**, itemID (Furnitures), E-mail (Costumers))

**Materials** ( **MaterialID**, color, maintenance)

**Infomations** ( **MaterialID** (Materials), information)

**Modifications** (**MaterialID (Materials)**, **itemID(Furnitures)**, price, size)

קוד SQL להקמת ישות החיפוש (לא קיים בחלק ב')-

Create Table Searches(

IP Varchar(15) not null,

DT DateTime not null,

ItemID int null,

Email varchar(30),

primary key(IP,DT),

constraint FK\_ItemID\_Se

foreign key(ItemID)

references Furnitures(ItemID))

קוד SQL לאחר איחוד הטבלת המשלוחים וההזמנות-

--תיקון ישות ההזמנות

create table Orders (

OrderID int not null,

OrderDate date not null,

CardNumber varchar(16) not null,

Country Varchar(20) not null,

Street varchar(20) not null,

Number integer not null,

ZipCode varchar(20) not null,

ShippingType varchar(20) not null,

price money not null,

primary key(OrderID),

constraint FK\_CreditCard\_Od

foreign key(CardNumber)

references CreditCards(CardNumber))

alter table Orders

add constraint Ck\_price\_Or

check (price >= 0) -- (=) coz refunds

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **able** | **Field** | | **Data Type** | **Size (Bytes)** |
| Customers | E-mail | | Varchar(30) | 30 |
| firsName | | Varchar(20) | 20 |
| lastName | | Varchar(20) | 20 |
| phoneNumber | | Varchar(20) | 20 |
| Total Record Size (Bytes) | | | **90** |
| CreditCards | card number | | Varchar(16) | 16 |
| type | | Varchar(20) | 20 |
| expired date | | Varchar(5) | 5 |
| CCV | | Varchar(3) | 3 |
| E-mail | | Varchar(30) | 30 |
| Total Record Size (Bytes) | | | **74** |
| Furnitures | itemID | | Integer | 4 |
| mainType | | Varchar(20) | 20 |
| secondaryType | | Varchar(20) | 20 |
| collection | | Varchar(20) | 20 |
| Total Record Size (Bytes) | | | **64** |
| Orders | orderID | | Integer | 4 |
| OrderDate | | DateTime | 8 |
| card number | | Varchar(16) | 16 |
| country | | Varchar(20) | 20 |
| street | | Varchar(20) | 20 |
| number | | Integer | 4 |
| zipcode | | Varchar(20) | 20 |
| shippingType | | Varchar(20) | 20 |
| price | | money | 8 |
| Total Record Size (Bytes) | | | **120** |
| contains | orderID | | Integer | 4 |
| itemID | | Integer | 4 |
| quantity | | Integer | 4 |
| Total Record Size (Bytes) | | | **12** |
| Search | IP adress | | Varchar(15) | 15 |
| DT | | DateTime | 8 |
| itemID | | Integer | 4 |
| E-mail | | Varchar(30) | 30 |
| Total Record Size (Bytes) | | | **57** |
| materials | MaterialID | | Integer | 4 |
| color | | Varchar(20) | 20 |
| maintenance | | Varchar(20) | 20 |
| Total Record Size (Bytes) | | | **44** |
| information | MaterialID | | Integer | 4 |
| Information | | Varchar(20) | 20 |
| Total Record Size (Bytes) | | | **24** |
| modifications | MaterialID | | Integer | 4 |
| itemID | | Integer | 4 |
| price | | money | 8 |
| size | | Varchar(20) | 20 |
| Total Record Size (Bytes) | | | **36** |
| favorites | E-mail | Varchar(30) | | 30 |
| itemID | Integer | | 4 |
| Total Record Size (Bytes) | | | **34** |

\*zipcode = varchar- might contain letters

\*phoneNumer= varchar- will contain “+” (country code)

## מטלה 1 (25%) – שאילתות

# שתי שאילתות SELECT ללא קינון (5%)

הרעיון העסקי- החברה רוצה להקטין את מבחר הרהיטים בישראל עקב בעיות ההובלה שנגרמו בגלל הקורונה ולכן היא בודקת מה השלושה הרהיטים שהכניסו הכי מעט במדינה.

select top 3 f.itemID, 'net value' = sum (m.price\*c.quantity)

from furnitures as f join modifications as m on f.itemId=m.itemId

join [Contains] as c on f.itemID= C.itemId

join Orders as o on c.OrderID = o.OrderID

where o.country= 'israel'

group by f.itemID

having count (\*) < 20

order by 'net value'

Table

Description automatically generated

הרעיון העסקי – החברה מעוניינת לאתר את הסניפים בהם יש הכי הרבה הזמנות שדרך ההספקה שלהם היא איסוף עצמי כדי לבדוק אם יש את האפשרות לצמצם את מערך הלוגיסטי.

select top 3 o.country, pickups = count (\*)

from Customers as c join creditcards as cc on c.email= cc.email

join orders as o on cc.cardnumber= o.cardnumber

where o.shippingtype= 'pick up'

group by o.country

ORDER BY pickups DESC

Table

Description automatically generated

# שתי שאילתות SELECT מקוננות (10%)

הרעיון העסקי- החברה רוצה לבדוק איזה אוסף בין שני אוספים יותר נמכר בכל מדינה כדי ללמוד מכך את העדפות של הלקוחות השונים לטובת שיתופי פעולה עם מעצבים עתידיים.

select b.country , bergamo , fermo, gap = bergamo - fermo

from (

select o.Country,fermo = sum(c.quantity)

from furnitures as f

join [Contains] as c on f.itemID= C.itemId

join Orders as o on c.OrderID = o.OrderID

where f.collection= 'fermo'

group by o.Country

A screenshot of a computer

Description automatically generated with medium confidence)as a join(

select o.Country, count(\*) as bergamo

from furnitures as f

join [Contains] as c on f.itemID= C.itemId

join Orders as o on c.OrderID = o.OrderID

where f.collection= 'bergamo'

group by o.Country

)as b on a.Country=b.Country

הרעיון העסקי- החברה מעוניינת לדעת באיזה מדינות מכרו ליותר לקוחות יחודיים (לא לקוח חוזר) והשאילתה מציגה את המדינות שנמצאות מעל הממוצע העולמי לטובות חלוקת בונוס לעובדים במדינה.

select o.Country , count(distinct c.Email) as total

from Customers as c join creditcards as cc on c.email= cc.email

join orders as o on cc.cardnumber= o.cardnumber

where year(o.OrderDate) = 2020

group by o.Country

having (select av=AVG(a.total)

Table

Description automatically generatedfrom(

select o.Country , count(distinct c.Email) as total

from Customers as c join creditcards as cc on c.email= cc.email

join orders as o on cc.cardnumber= o.cardnumber

where year(o.OrderDate) = 2020

group by o.Country

)as a) < count(distinct c.Email)

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

**שאילתת UPDATE**

רעיון עסקי- החברה החליטה לתת לכל לקוח מטבע לשימוש באתר שהוא מחושב כך- בשביל לקבל מטבע אחד צריך לקנות ב10$ באתר הצבירה גם מתבצעת על ההזמנות הקודמות של הלקוח עקב מבצע השקה.

-- ALTER TABLE Customers DROP COLUMN Fcoins

alter table Customers add FCoins money

update Customers set Fcoins =

(select sum(o.price)\*0.1

from orders as o join CreditCards as cc on o.CardNumber = cc.CardNumber

where Customers.Email = cc.Email

group by cc.Email

)

select \*

from Customers

order by fcoins desc

אחרי העדכון לפני העדכון

Graphical user interface, text

Description automatically generatedTable

Description automatically generated

**שאילתת חיסור**

רעיון עסקי- החברה מעוניינת לשלוח קופון הנחה להזמנות לכל הלקוחות שסימנו לפחות שני פריטים במועדפים ואך הם מעולם עוד לא ביצעו הזמנה באתר כדי לדחוף אותם לקניה.

select \*

from Customers as c

where (select count(cu.Email)

from Customers as cu join Favorites as f   
 on cu.Email = f.Email

where cu.Email = c.Email

) > 2

except

select c.Email, c.firstName, c.lastName , c.phone

from Customers as c join creditcards as cc on c.email= cc.email

join orders as o on cc.cardnumber= o.cardnumber

group by c.Email, c.firstName, c.lastName , c.phone

Table

Description automatically generated

## מטלה 2 (25%) – יישומי כלים מתקדמים

# View (5%)

רעיון עסקי- החברה רוצה לספק טבלה חדשה לעובדים הרבים עובדים אצלה אך אינם צריכים נתונים מלאים על לקוחות החברה ובנוסף הטבלה נוצרה גם לטובת הגברת אבטחת מידע.   
לכן החברה מעוניינת להסתיר את הכתובת המדויקת של הלקוח, תוקף וCCV של כרטיס האשראי.

create view view\_CustomerOrder as

select c.Email, c.firstName, c.lastName , c.phone,

o.OrderID, o.OrderDate, o.CardNumber, o.Country , o.ShippingType,o.price

from Customers as c join creditcards as cc on c.email= cc.email

join orders as o on cc.cardnumber= o.cardnumber

select \*

from view\_CustomerOrder

# Graphical user interface, table Description automatically generated with medium confidence

# פונקציות (Functions) (10%)

**פונקציית סקלאר**

רעיון עסקי- הפונקציה מקבלת אימייל של לקוח ומחזירה את סך העלות של כל ההזמנות שביצע. כך אם נרצה לעשות פעולות השוואה שונות בין לקוחות מסוימים לפי כמות ההוצאות שלהם נוכל לקבל בקלות את הנתונים.

create function ordersCosts (@email varchar (30))

returns money

as begin

declare @ordersCosts money

select @ordersCosts= (sum(o.price))

from Customers as c join creditcards as cc on c.email= cc.email

join orders as o on cc.cardnumber= o.cardnumber

where c.Email = @email

group by c.Email

return @ordersCosts

end

select distinct c.Email , [orders cost]= dbo.ordersCosts(c.Email)

from Customers as c join creditcards as cc on c.email= cc.email

join orders as o on cc.cardnumber= o.cardnumber

order by [orders cost] desc

Text

Description automatically generated with medium confidence

**פונקציית טבלה**-

רעיון עסקי- החברה רוצה לדעת כמה מכירות היא ביצעה בכל מדינה בין תאריכים שונים ואם אפשרות לפלטור סוג ההספקה, משלוח או איסוף, לטובת הסקת מסקנות לגבי המערך הלוגיסטי בכל מדינה.

-- drop function function\_CountrySellesByShippingType

create function function\_CountrySellesByShippingType ( @Country varchar(20), @from date, @until date, @ShipingType varchar(20))

returns table as return

select c.ItemID ,count(distinct o.OrderID) as total\_orders, sum(c.quantity) as total\_amount

from Orders as o join [Contains] as c on o.OrderID = c.OrderID

where (o.OrderDate between @from and @until) and o.Country = @Country and o.ShippingType = @ShipingType

group by c.ItemID

select \*

from dbo.function\_CountrySellesByShippingType('denemrk','2021-12-12','2022-12-12' ,'Pick Up')

A picture containing table

Description automatically generated

# Trigger פשוט (5%)

רעיון עסקי- כאשר החברה מפסיקה לייצר מוצר והיא רוצה למחוק אותו מטבלת הרהיטים, הטריגר נכנס לפעולה ומוחק את כל הרשומות בטבלת MODIFICATION שקשורות לרהיט הנמחק. (פותר את בעיית סדר מחיקה עם מפתחות זרים).

-- drop trigger Update\_modifications

CREATE TRIGGER Update\_modifications

ON furnitures instead of delete

as begin

delete Modifications

where ItemID in (select d.ItemID from deleted as d) and ItemID in (select m.ItemID from Modifications as m)

delete Furnitures

where ItemID in (select d.ItemID from deleted as d) and ItemID in (select f.ItemID from Furnitures as f)

end

select \*

from Modifications as m

where m.ItemID = 108

delete from Furnitures where Furnitures.ItemID = 108

אחרי הפעלת הטריגר לפני הפעלת הטריגר

Graphical user interface, application

Description automatically generatedTable

Description automatically generated

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

רעיון עסקי- החברה צריכה נתוני מכירה עדכניים מכל מדינה לטובת פעולות חיזוי שונות.   
לכן הפרוצדורה מקבלת מדינה ומחזירה כמה רהיטים נמכרו לפני חודשיים.

create procedure sp\_furniture\_supply (@country varchar (20))

as begin

select c.itemid, total\_amount= sum(c.quantity)

from orders as o join [Contains] as c on o.orderid= c.orderid

where year (o.orderdate)= year (getdate()) and month(o.orderdate)= month(getdate())-2

and o.country = @country

group by c.itemid,c.quantity

end

execute sp\_furniture\_supply 'israel'

Chart

Description automatically generated with medium confidence

## מטלה 3 (20%) – כלים להצגת נתונים

**דוח עסקי**- הדוח מיועד למעצב הראשי של החברה. מטרת הדוח, היא להציג למעצב את כל המידע על התפלגות המכירות על פי סוג הרהיט ועל פי סוג האוספים מהקמת החברה וגם על המכירות של כל סוג רהיט לפי שנים כדי שהוא יוכל לראות את השינוי במגמות לאורך הזמן.

הדוח כולל –

כמות מכירות בכל מדינה בכללי וגם על פי סוג מוצר.  
גודל נתח השוק על פי אוספים כלל עולמי.  
כמות מכירות של כל סוג רהיט לפי שנים.

**Chart

Description automatically generated**

ה-view שעזר לנו לבנות את הדוח ואת הלוח מחוונים-

CREATE VIEW V\_Sales as

select c.Email as [customer email], o.OrderID as [order ID], o.OrderDate as [order date], o.Country as [country], o.ShippingType as [Shipping Type], o.price as [order price], fu.ItemID as [furniture ID], co.quantity as [quantity] ,fu.collection as [furniture collection], fu.MainType as [furniture Main Type]

from Customers as c full join creditcards as cc on c.email= cc.email

full join orders as o on o.cardnumber=cc.cardnumber

full join Favorites as f on f.Email = c.Email

full join [Contains] as co on co.OrderID = o.OrderID

full join Furnitures as fu on fu.ItemID = co.ItemID

group by c.Email, cc.CardNumber, o.OrderID, o.OrderDate, o.Country, o.ShippingType, o.price, fu.ItemID, fu.collection, fu.MainType, co.quantity

**לוח מחוונים**- החברה מעמידה מטרות עסקיות מדי שנה, והיא מעוניינת שתהיה לה את היכולת לראות האם הם עמדו ביעדים ביחד עם סיכום כולל של כל הנתונים הרלוונטיים כדי, שההנהלה הבכירה תוכל לראות היכן החברה עומדת ביחס ליעדים שהציבה לעצמה ובכללי.

הלוח כולל-

כל ההכנסות השנתיות כולל היעד השנתי שייעדנו.  
כמות ההזמנות הכוללת, כולל היעד השנתי שייעדנו.  
פילוגים של רווחים על פי סוג משלוח.  
רווח כולל של כל רבעון על פי מדינה.

**Graphical user interface, application

Description automatically generated**

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

## שאילתות עסקיות המשלבות Window Functions (10%)

רעיון עסקי- השאילתה נותנת לחברה יכולת ניתוח יותר גבוהה של ההכנסות של כל רהיט ורהיט ביחד עם האפשרות לראות את יחס כמותי של פריטים שנמכרו, ובכך אנו משגים את הניתוח המעמיק בחלוקה לאחוזונים מבחינת ההכנסות מכל פריט ודירוג אחוזון מ-1-100 לכמות הפרטים שנמכרו. אנו יצרנו את השאילתה הזו לטובת הפקת דוחות עסקיים לדרג בכיר.

--drop view view\_payments

create view view\_payments as

select c.ItemID, payment = sum(o.price), [counter] = count (\*)

from orders as o join [contains] as c on o.OrderID = c.orderID

group by c.ItemID

select itemID, profits = payment,

NTILE (10) OVER (ORDER BY payment) as [Deciles],

[counter] as [Number of sold items],

(round (PERCENT\_RANK() OVER (ORDER BY [counter]), 3))\*100 as [Distribution by percentages of item sales]

from view\_payments

order by 2 desc

Table

Description automatically generated

רעיון עסקי- השאילתה הזו באה להשלים את השאילתה הקודמת, לטובת הדוחות העסקיים לרמה בכירה, אנו מודדים את הרווח של כל אוסף ביחס לכמות הפרטים. תוך כדי חלוקה לדירוגים הן מבחינת הכנסות כלליות והן יחס ודירוג אחוזון מ-1-100 ליחס הטוב ביותר.

create view view\_selledCollections as

SELECT [collection], payments = sum(o.price),

(select count(fu.collection)

from Furnitures as fu

where f.collection = fu.collection) as [Number of items]

from Furnitures as f join [Contains] as c on f.ItemID = c.ItemID

join orders as o on o.OrderID = c.OrderID

group by [collection]

select [collection], [Avg income per item] = payments/[Number of items] ,profits = payments,

rank() over (order by payments/[Number of items] desc) as [rank by ratio],

round(cume\_dist() over(order by payments/[Number of items] asc),3)\*100 as [Distribution by percentages of collection sales],

rank() over (order by payments desc) as [rank by profits]

from view\_selledCollections

order by [rank by ratio]

Graphical user interface, table

Description automatically generated with medium confidence

## שילוב מערכתי של מספר כלים (10%)

רעיון עסקי- החברה מעוניינת לדעת מה-X הרהיטים הכי פחות נמכרים וכדי לעודד את קנייתם היא מבצעת הנחה על מחיר הבסיס של המוצר.  
הפרוצדורה שלנו מקבלת את מספר המוצרים הכי פחות נמכרים ואת אחוז ההנחה שהחברה מעוניינת לתת.  
לצורך הפרוצדורה יצרנו view שמחשב לנו כמה מכירות ביצעו מכל מוצר ספציפי ומדרג מהמוצר הכי פחות נמכר למוצר שהכי נמכר. בנוסף יצרנו פונקציה אשר מחזירה טבלה עם ה-X מוצרים הרלוונטיים (לדוגמא תחזיר את חמשת המוצרים הכי פחות נמכרים אם תקבל 5) מטבלת ה- view.  
הפרוצדורה מקבל את אחוז הנחה ואת כמות המוצרים שאנחנו מעוניינים לבצע את הנחה עליהם. והיא תעדכן את המחיר של המוצרים בטבלת Modifications לפי אחוז ההנחה שתקבל.

create view v\_furnitures\_rank as

select c.ItemID, quantitiy = sum(c.quantity),RANK() over (order by sum(c.quantity)) rank

from orders as o join [Contains] as c on o.orderid= c.orderid

join Furnitures as f on f.ItemID=c.ItemID

group by c.ItemID

create function function\_items\_quan ( @num int)

returns table as return

select \*

from v\_furnitures\_rank

where rank<= @num

-------drop procedure sp\_make\_disc

create procedure sp\_make\_disc (@disc as float , @num as int)

as begin

update Modifications

set price = price\*(1-@disc)

where Modifications.ItemID in (select ItemID from dbo.function\_items\_quan(@num))

end

execute sp\_make\_disc 0.1 , 5

אחרי הפעלת הפרוצדורה לפני הפעלת הפרוצדורה

A picture containing table

Description automatically generatedA picture containing table

Description automatically generated

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

הדוח שלנו הוא דוח סיכומי, אשר מטרתו לשקף את מצב החברה על פי חלוקה למדינות. הדוח מציג משמאל לימין עבור כל מדינה, את הרווח של כל מדינה, כמה לקוחות רשומים יש בה, אחוז הרווח שהמדינה מפיקה לחברה מכלל הרווחים של החברה, את כמות האיסופים ואת כמות המשלוחים בכל מדינה.

with

-- סכום מכירות של כול מדינה

countrySale (countryS ,[sum country Sales]) as

(select o.Country , sum(o.price)

from orders as o

group by o.Country),

--כמות לקוחות בכל מדינה

countryCostomers (countryC, [coustemer amount]) as

(select o.country, count(distinct c.Email)

from orders as o join CreditCards as cc on o.CardNumber = cc.CardNumber

join Customers as c on c.Email = cc.Email

group by o.Country),

--כלל הרווחים של החברה

totalP as

(select sum(o.price) as [total profit]

from orders as o),

--אחוז רווח מכלל הרווחים של החברה

countryProfitP (countryP, [profit percent]) as

(select cs.countryS, [profit precent] = cs.[sum country Sales]/[total profit]

from countrySale as cs cross join totalP),

-- כמות האיסופים

countryPickUps (countryPU, [Pick Ups amount]) as

(select o.country, pickups = count (\*)

from Customers as c join creditcards as cc on c.email= cc.email

join orders as o on cc.cardnumber= o.cardnumber

where o.shippingtype= 'pick up'

group by o.country),

-- כמות המשלוחים

countryShipping (countrySh, [Shipping amount]) as

(select o.country, Shipping = count (\*)

from Customers as c join creditcards as cc on c.email= cc.email

join orders as o on cc.cardnumber= o.cardnumber

where o.shippingtype= 'Shipping'

group by o.country)

select cs.countryS, cs.[sum country Sales], cc.[coustemer amount], cp.[profit percent], cpu.[Pick Ups amount], csh.[Shipping amount]

from countrySale as cs join countryCostomers as cc on cs.countryS = cc.countryC

join countryProfitP as cp on cs.countryS = cp.countryP

join countryPickUps as cpu on cs.countryS = cpu.countryPU

join countryShipping as csh on cs.countryS = csh.countrySh

Table

Description automatically generated