|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| מס' קבוצה | אתר | | | תאריך הגשה |
| 16 | <seedsheets.com> | | | 11/03/2024 |
| חברי הצוות - מספרי ת.ז | | | | |
| 318334901 | | 211680509 | 318255882 | |

## פרויקט בסיסי נתונים – חלק ג'

## פרק ראשון

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

**התרשים בהגשה הקודמת**

A diagram of a product

Description automatically generated

**המודל הטבלאי בהגשה הקודמת**

A screenshot of a computer

Description automatically generated

**ההערות שניתנו להגשה הקודמת**

1. בצורת המידול הזו, לא בא לידי ביטוי המוצר המעוצב הסופי ויותר מזה, אין התייחסות לאפשרות לחזרתיות- תהיה מחיקה של נתונים.  
   ניתן דוגמא: מוצר בשם A (נניח יריעה מסוג א באתר) מקושרת ל8 מוצרי seed שונים- b,c,d,e,f,g,h,i שכל קשר מכיל כמות 1.  
   ומחר בבוקר נקנה מוצר מעוצב נוסף שגם מורכב ממוצר A עם קשר לSeed בשם b כמות של 8 פעמים.  
   בטבלת הchosen יצטרך להימחק לנו הכמות הקודמת של מוצר A עם seed מסוג B בכמות 1.  
   ועכשיו, איך נדע איזה מוצרים מעוצבים סופיים יש לנו?
2. הפתרון כאן הוא שימוש בישות חלשה. כפי שכתבנו לכם במייל, להיעזר בשקופית 74 מההרצאה.  
   שימו לב לפני חלק ג` לשנות את המודל בהתאם.
3. יש להקפיד להיות תמציתיים, ולא לציין הנחות שמובנות מאליהן.  
   לא הורדו נקודות אבל הקפידו על כך לקראת חלק ג
4. על הדף זה נשמע נורא יפה, אבל זה לא בא לידי ביטוי בERD שלכם.  
   שימו לב שהקשר של הלקוח הוא למוצר מסוג garden שבפועל המזהה שלו הוא שם. כלומר אם מחר הוא יקנה יריעה מסוג A עם קשר chosen מסויים ומחרתיים הוא יקנה יריעה מסוג A עם קשר chosen אחר, עדיין היריעה תיהיה אותה יריעה.  
   הבעיה היא הקשר בין יריעה לSEED.
5. שימו לב שהתבקשתם להגיש סקריפט מלא שכולל את כל השלבים בסדר הנכון.  
   לא הורדו נקודות כי סידרתם את העבודה נכון כך שיכולנו לבחון את הסדר. בפעם הבאה ירדו נקודות.
6. חסר אילוץ בדיקה לסיסמא: עליכם להגדיר אורך סיסמא מינימלי (לא הגיוני שסיסמא תהיה באורך 1), תווים שהם חובה וכו`.  
   חסר טבלת חיפוש לשיטות השילוח.

## עיצוב קונספטואלי מתוקן (תרשים ישויות-קשרים)

A diagram of a product

Description automatically generated

## הנחות שתוקנו או הוספו בהתאם למודל ה-ERD המתוקן

|  |  |  |
| --- | --- | --- |
| **מרכיב** | **הנחות** | **הערות נוספות** |
| הקשר "Planted" בין SEED וGARDEN   * quantity | * כל זרע מסוים יכול להיכלל באינספור גינות מעוצבות מראש או באף אחת מהן. לכן הקרדינליות של קשר זה בצד הישות GARDEN היא 0:N. * בכל גינה מעוצבת מראש מסוימת ייכללו מראש בין 1 ל8 זרעים, ובכל גינה בעיצוב אישי לא ייכלל מראש אף זרע. לכן הקרדינליות של קשר זה בצד הישות SEED היא 0:N. * quantity – כמות יחידות של זרע מסוים הכלול בגינה מעוצבת מראש.  שדה זה יכיל מספרים שלמים גדולים מ0 וקטנים מ9. |  |
| Week Entity: DESIGN   * designID | * מפתח חלקי: designID – לכל עיצוב מסוים של גינה בעיצוב אישי מסוימת קיים מספר סידורי ייחודי. * לכל גינה בעיצוב אישי יכולים להתקיים אינספור עיצובים או אף לא אחת. לכן הקרדינליות של הקשר של הישותGARDEN עם הישות החלשה DESIGN אשר תחת חסותה בצד הישות DESIGN היא 0:N. |  |
| הקשר "Chosen" בין SEED וDESIGN   * quantity | * כל זרע מסוים יכול להיבחר להיכלל באינספור עיצובים או באף אחד מהם. לכן הקרדינליות של קשר זה בצד הישות DESIGN היא 0:N. * בכל עיצוב מסוים של גינה בעיצוב אישי מסוימת יכולים להיבחר בין 1 ל8 זרעים. לכן הקרדינליות של קשר זה בצד הישות SEED היא 1:8. * quantity – כמות יחידות של זרע מסוים שנבחר להיכלל בעיצוב מסוים. שדה זה יכיל מספרים שלמים גדולים מ0 וקטנים מ9. |  |
| הקשר "Designed for" בין DESIGN וORDER   * Quantity | * בכל הזמנה מסוימת יכולים להיכלל אינספור גינות בעיצוב אישי או אף לא אחת. לכן הקרדינליות של קשר זה בצד הישות DESIGN היא 0:N. * כל עיצוב מסוים של גינה בעיצוב אישי מסוימת יכול להיכלל בהזמנה אחת בלבד. לכן הקרדינליות של קשר זה בצד הישות ORDER היא 1:1. * quantity – כמות יחידות של גינה בעיצוב אישי מסוימת בעיצוב מסוים הנכללות בהזמנה מסוימת. שדה זה יכיל מספרים שלמים גדולים מ0. |  |

## מודל טבלאי מתוקן

* **PRODUCTS** ( **Name**, Price, Discount )
* **USERS** ( **Email**, Password)
* **SEARCHES** ( **SearchDT**, **IP\_address**, Email ( USERS ), Search\_text )
* **RELATIONS** ( **Product1** ( PRODUCTS ), **Product2** ( PRODUCTS ) )
* **RESULTS** ( **Name** ( PRODUCTS ), { **SearchDT**, **IP\_address** } ( SEARCHES ) )
* **GARDENS** ( **Name** ( PRODUCTS ), Small\_count, Large\_count )
* **SEEDS** ( **Name** ( PRODUCTS ), Size, Season, Sun\_amount )
* **SEED\_TYPES** ( **Name** ( SEEDS ), **Type** )
* **PLANTEDS** ( **Garden** ( GARDENS ), **Seed** ( SEEDS ), Quantity )
* **DETAILS** ( **Name**, **Address**, Company, Phone# )
* **DETAILS\_OF** ( **Email** ( USERS ), { **Name**, **Address** } ( DETAILS ) )
* **ORDERS** ( **OrderID**, Email ( USERS ), { Name, Address } ( DETAILS ), OrderDate,  
   Shipping\_method, Payment\_type )
* **DESIGNS** ( **Name** ( GARDENS ), **DesignID**, OrderID ( ORDERS ), Quantity )
* **CHOSENS** ( { **Garden**, **Design** } ( DESIGNS ), **Seed** ( SEEDS ), Quantity )
* **INCLUSIONS** ( **OrderID** ( ORDERS ), **Name** ( PRODUCTS ), Quantity )

**הנחות שתוקנו או הוספו:**

* **ביטוי הקשרים Included שבין ORDER ל-PRODUCT ו-Designed for שבין DESIGN ל-ORDER** – הקשר Included הוא קשר רבים לרבים. בדומה לקשרים קודמים, ניצור טבלה חדשה כדי לייצג את הקשר. בנוסף למפתחות הזרים אשר ביחד יהוו מפתח ראשי, יהיה שדה עבור התכונה quantity המצוינת במודל הERD-.  
  בקשר Included השדה הראשון יהווה מפתח זר לPRODUCT, והשדה השני יהווה מפתח זר לORDER.  
  הקשר Designed for הוא קשר יחיד לרבים. לכן מבוטא ע"י שדה Designed\_for בטבלת DESIGNS אשר מהווה מפתח זר לORDER.  
  חשוב לציין כי על אף שגינה בעיצוב אישי היא מוצר המוגדר כמו כל שאר המוצרים DESIGN אינו מוצר אלא שירות שניתן ללקוח בעת ביצוע הזמנה מסוימת הכוללת בתוכה גינה בעיצוב אישי, זאת לעומת כל מוצר אחר אשר לא מקבל התייחסות אישית מהלקוח בעת הזמנתו.  
  בכך לשירות העיצוב של גינה מסוימת קיימת ההצדקה של קשר כמעט זהה לקשר קיים אחר.
* **ביטוי הקשרים Planted שבין GARDEN ל-SEED ו-Chosen שבין DESIGN ל-SEED** – קשרי רבים לרבים. הטבלה של הקשר Planted תכיל שלושה שדות: הראשון מפתח זר לGARDEN, השני מפתח זר לSEED כך ששניהם יחד יהיו המפתח הראשי, ושדה שלישי עבור התכונה quantity המצוינת במודל ה-ERD.הטבלה של הקשר Chosen תכיל 4 שדות: השניים הראשונים יהוו מפתח זר לDESIGN, השלישי מפתח זר לSEED כך ששלושתם יחד יהיו המפתח הראשי ושדה רביעי עבור התכונה quantity המצוינת במודל ה-ERD.  
  בדומה לדמיון בין הקשרים Included וDesigned for גם קשר זה מקבל את ההצדקה מעצם היות עיצוב מסוים של גינה בעיצוב אישי שירות.
* **ביטוי מודל ההורשה שבין PRODUCT ל-SEED ול-GARDEN** – מודל הורשה בדיד וחלקי. על פי חוקי המעבר למודל הטבלאי קיימת הבחירה בין טבלה אחת לכלל הישויות או טבלה לכל אחת מהישויות במודל ההורשה, במקרה שלנו באמצעות שלוש טבלאות.  
  הדרך שנבחרה לביטוי מודל ההורשה הינה באמצעות טבלה לכל ישות, כדי להימנע מערכי NULL רבים שבהכרח יהיו במימוש באמצעות טבלה אחת. זאת מכיוון שערכי ה-NULL הרבים יתפסו מקום בזכירון ויעמדו בדרכנו בעת תחקור מסד הנתונים.

## פירוט שדות בסיס הנתונים שתוקנו או הוספו

| **Table** | **Field** | | **Data Type** | **Size (Bytes)** |
| --- | --- | --- | --- | --- |
| **PLANTEDS** | Garden **(GARDENS)** | | Varchar(80) | 80 |
| Seed **(SEEDS)** | | Varchar(80) | 80 |
| Quantity | | Tinyint | 1 |
| **Total Record Size (Bytes)** | | | **161** |
| **DESIGNS** | Garden **(GARDENS)** | | Varchar(80) | 80 |
| DesignID | | Int | 4 |
| Designed\_for **(ORDERS)** | | Int | 4 |
| Quantity | | Tinyint | 1 |
| **Total Record Size (Bytes)** | | | **89** |
| **CHOSENS** | Garden | **(DESIGNS)** | Varchar(80) | 80 |
| Design | Int | 4 |
| Seed **(SEEDS)** | | Varchar(80) | 80 |
| Quantity | | Tinyint | 1 |
| **Total Record Size (Bytes)** | | |  |

## מטלה 1 (%70 מציון הפרק הראשון) – שאילתות

# שתי שאילתות SELECT ללא קינון (8%)

## בעבור כל זרע, מה היא הכמות שהוזמנה ממנו כחלק מגינה מעוצבת אישית בחודש האחרון ובאותו חודש שנה לפני כן? מטרת השאילתה היא תמונת מצב של ביצועי מכירות של הזרעים בהשוואה לאותה תקופה בשנה שעברה.

SELECT [Seed Name] = C.Seed,

[Seed Quantitity Of Last Month] = SUM(CASE WHEN DATEDIFF(MONTH, O.OrderDate, GETDATE()) = 1 THEN C.Quantity\*DSG.Quantity ELSE 0 END),

[Seed Quantitity Of Year Before] = SUM(CASE WHEN DATEDIFF(MONTH, O.OrderDate, GETDATE()) = 13 THEN C.Quantity\*DSG.Quantity ELSE 0 END)

FROM dbo.ORDERS AS O

JOIN dbo.DESIGNS as DSG

ON O.OrderID = DSG.OrderID

JOIN dbo.CHOSENS AS C

ON DSG.Name = C.Garden AND DSG.DesignID = C.Design

GROUP BY C.Seed

HAVING SUM(CASE WHEN DATEDIFF(MONTH, O.OrderDate, GETDATE()) = 1 THEN C.Quantity\*DSG.Quantity ELSE 0 END) > 0

OR SUM(CASE WHEN DATEDIFF(MONTH, O.OrderDate, GETDATE()) = 13 THEN C.Quantity\*DSG.Quantity ELSE 0 END) > 0

ORDER BY [Seed Quantitity Of Last Month] DESC, [Seed Quantitity Of Year Before]

לכל סוג של של גינה בעיצוב אישי, כמה נמכרו בין התאריך של היום לאותו תאריך שנה לפני כן?  
מטרת השאילתה היא להעריך כמה גינות צריך להכין מכל סוג לצורך השנה הקרובה.

SELECT D.Name,

[Total Orders] = COUNT(\*)

FROM DBO.DESIGNS AS D JOIN DBO.ORDERS AS O ON D.OrderID = O.OrderID

WHERE DATEDIFF(DAY,O.OrderDate, GETDATE()) <= 365

GROUP BY D.Name

ORDER BY [Total Orders] DESC

# שתי שאילתות SELECT מקוננות (15%)

מה הן 15 המילים שהשתמשו בהן הכי הרבה כחלק מחיפוש באתר?  
מטרת השאילתה היא כדי לזהות מוצרים פופלריים וזיהוי של טרנדים שניתן לשלב לתוך שיווק מוצרי החברה

SELECT TOP 15

W.searchWord,

Appearances = COUNT(\*)

FROM (

SELECT searchWord = LOWER(value)

FROM dbo.SEARCHES

CROSS APPLY STRING\_SPLIT(Search\_text, ' ')

) AS W

WHERE LEN(W.searchWord) > 3

GROUP BY W.searchWord

ORDER BY Appearances DESC

# באיזה חודשים בשנה יש הכי הרבה הזמנות. מטרת השאילתה היא כדי לדעת מתי יהיה הכי נכון לשווק הנחות ומוצרים ומתי לחזק את השיווק כדי להגביר קניה בתקופות החלשות יותר

SELECT [Average Orders] = AVG(Orders),

[Month]

FROM (

SELECT Orders = COUNT(\*),

[Year] = YEAR(O.OrderDate),

[Month] = DATENAME(MONTH, O.OrderDate)

FROM dbo.ORDERS AS O

WHERE DATEDIFF(MONTH, O.OrderDate, GETDATE()) <= 60 /\* orders from the last 5 years \*/

GROUP BY YEAR(O.OrderDate), DATENAME(MONTH, O.OrderDate)

) As OMonth

GROUP BY [Month]

HAVING AVG(Orders) > ( /\* select months that their average orders is above the average of all months \*/

SELECT AVG(Orders1)

FROM (

SELECT Orders1 = COUNT(\*),

[Year1] = YEAR(O1.OrderDate),

[Month1] = DATENAME(MONTH, O1.OrderDate)

FROM dbo.ORDERS AS O1

WHERE DATEDIFF(MONTH, O1.OrderDate, GETDATE()) <= 60 /\* orders from the last 5 years \*/

GROUP BY YEAR(O1.OrderDate), DATENAME(MONTH, O1.OrderDate)

) As OAVG

)

ORDER BY [Average Orders] DESC

# שתי שאילתות מקוננות תוך שימוש במרכיבים נוספים (15%)

מחיקת קשרים בין מוצרים שלא נקנו יחד 3 שנים ברציפות  
מטרת השאילתה היא לייעל את תהליך הפרסום דרך הקשרים הללו של מוצרים ולאפשר להצעות יותר סבירות להופיע ללקוח, זאת בהנחה שאם 3 שנים לא נקנו שתי מוצרים יחד באותה הזמנה ומתועד שיש ביניהם קשר אז גם לא ייקנו אותם יחד.

DELETE FROM dbo.RELATIONS

WHERE CONCAT(Product1, Product2) NOT IN (

SELECT CONCAT(RLT.Product1, RLT.Product2)

FROM dbo.RELATIONS AS RLT

JOIN dbo.INCLUSIONS AS I1

ON RLT.Product1 = I1.Name

JOIN dbo.INCLUSIONS AS I2

ON RLT.Product2 = I2.Name

JOIN dbo.ORDERS AS O

ON I1.OrderID = O.OrderID

WHERE I1.OrderID = I2.OrderID AND DATEDIFF(YEAR,O.OrderDate, GETDATE()) BETWEEN 1 AND 3

)

דוח שנתי רבעוני של ביצועי זרעים מסוג שורש  
לצורך בחינת ביצועיהם של זרעים מסוג זה לפי רבעון

SELECT [Seed\_Name] = Seed,

[Year] = YEAR(OrderDate),

[Quarter] = DATEPART(QUARTER, OrderDate),

[Total Quantity] = SUM(Quantity),

[Total Orders] = COUNT(OrderID),

[Avg Quantity per Order] = ROUND(AVG(CAST(Quantity AS FLOAT)),2)

FROM (

SELECT Seed = C.Seed,

OrderID = O.OrderID,

OrderDate = O.OrderDate,

Quantity = C.Quantity\*DSG.Quantity,

[Type] = ST.[Type]

FROM dbo.CHOSENS AS C

JOIN dbo.DESIGNS AS DSG

ON C.Design = DSG.DesignID

JOIN dbo.ORDERS AS O

ON DSG.OrderID = O.OrderID

JOIN dbo.SEEDS AS SD

ON C.Seed = SD.Name

JOIN dbo.SEED\_TYPES AS ST

ON SD.Name = ST.Name

UNION

SELECT Seed = I.Name,

OrderID = O.OrderID,

OrderDate = O.OrderDate,

Quantity = I.Quantity,

[Type] = ST.[Type]

FROM dbo.INCLUSIONS AS I

JOIN dbo.ORDERS AS O

ON I.OrderID = O.OrderID

JOIN dbo.SEEDS AS SD

ON I.Name = SD.Name

JOIN dbo.SEED\_TYPES AS ST

ON SD.Name = ST.Name

UNION

SELECT Seed = PLT.Seed,

OrderID = O.OrderID,

OrderDate = O.OrderDate,

Quantity = PLT.Quantity\*I.Quantity,

[Type] = ST.[Type]

FROM dbo.PLANTEDS AS PLT

JOIN dbo.GARDENS AS G

ON PLT.Garden = G.Name

JOIN dbo.PRODUCTS AS PRD

ON G.Name = PRD.Name

JOIN dbo.INCLUSIONS AS I

ON PRD.Name = I.Name

JOIN dbo.ORDERS AS O

ON I.OrderID = O.OrderID

JOIN dbo.SEEDS AS SD

ON PLT.Seed = SD.Name

JOIN dbo.SEED\_TYPES AS ST

ON SD.Name = ST.Name

) AS SDByOrd

WHERE DATEDIFF(QUARTER, OrderDate, GETDATE()) <= 4 AND DATEDIFF(QUARTER, OrderDate, GETDATE()) > 0 AND [Type] = 'Root Vegetables'

GROUP BY Seed, YEAR(OrderDate), DATEPART(QUARTER, OrderDate)

ORDER BY YEAR(OrderDate), DATEPART(QUARTER, OrderDate), [Total Quantity] DESC

# שאילתות עסקיות המשלבות Window Functions (%16)

דירוג ערים לפי הכנסה כוללת מאותו עיר, בנוסף להראות את ההתפלגות המצטברת של הכנסת הערים.  
זאת לצורך דוח מצב להנהלה ולצורכי צוותי השיווק.

SELECT ordcities.State, ordcities.City, ordcities.Orders\_per\_City,

city\_rank\_by\_orders = ROW\_NUMBER() over (Partition BY ordcities.State Order BY ordcities.Sales\_per\_City DESC),

cume\_dist = CUME\_DIST() OVER (PARTITION BY State ORDER BY Orders\_per\_City)

FROM

(

/\* Calculate Sales per City \*/

SELECT State, City, Orders\_per\_City = COUNT(OrderID), Sales\_per\_City = SUM(order\_price)

FROM

( /\* extract geography (State, City) for each order and calculate order price \*/

SELECT OrderID, State, City, order\_price = SUM(product\_price)

FROM

(

SELECT ord.OrderID,

CAST(COALESCE(LTRIM(CAST(('<X>'+REPLACE(Address,',' ,'</X><X>')+'</X>') AS XML).value('(/X)[2]', 'varchar(128)')), '') AS varchar(128)) AS State,

CAST(COALESCE(LTRIM(CAST(('<X>'+REPLACE(Address,',' ,'</X><X>')+'</X>') AS XML).value('(/X)[3]', 'varchar(128)')), '') AS varchar(128)) AS City,

product\_price = inc.Quantity \* (prd.Price - prd.Discount)

FROM dbo.ORDERS AS ord

INNER JOIN dbo.INCLUSIONS AS inc

ON ord.OrderID = inc.OrderID

INNER JOIN dbo.PRODUCTS AS prd

ON inc.Name = prd.Name

) AS ords

GROUP BY OrderID, State, City

) AS ordState

GROUP BY State, City

) AS ordcities

ORDER BY State, city\_rank\_by\_orders

אמידה של מספר הימים עד להזמנה הבאה של כל לקוח מסוים, ללא לקוחות שביצעו הזמנה יחידה וללא לקוחות שהפסיקו להזמין לאחר תקופת הזמן שנאמדה.  
לצורך ניהול היצע וביקוש של סחורה ולצורך שיקולי שיווק.

SELECT \*

FROM

(

SELECT DISTINCT Email, Last\_Order\_Date = MAX(OrderDate) OVER (PARTITION BY Email),

Orders\_Per\_User = COUNT(OrderID) OVER (PARTITION BY Email),

Total\_Avg\_orders\_time\_gap = AVG(Difference\_in\_Days) OVER (),

Avg\_User\_Orders\_time\_gap = AVG(Difference\_in\_Days) OVER (PARTITION BY Email),

Days\_from\_Last\_User\_Order,

Estimated\_Days\_to\_Next\_Order = AVG(Difference\_in\_Days) OVER (PARTITION BY Email) - Days\_from\_Last\_User\_Order

FROM

(

SELECT Ord.Email, Ord.OrderID, Ord.OrderDate, LEAD(Ord.OrderDate) Over(Partition BY Email ORDER BY Email) AS Next\_Order\_Date,

DATEDIFF(day, ord.OrderDate, LEAD(Ord.OrderDate) Over(Partition BY Email ORDER BY Email)) AS Difference\_in\_Days,

Last\_User\_Order\_Date = LAST\_VALUE(OrderDate) OVER (PARTITION BY Email ORDER BY Email),

Days\_from\_Last\_User\_Order = DATEDIFF(day, LAST\_VALUE(OrderDate) OVER (PARTITION BY Email ORDER BY Email), GETDATE())

FROM dbo.ORDERS AS ord

WHERE ord.Email IS NOT NULL

) AS next\_ords

) AS C

WHERE Estimated\_Days\_to\_Next\_Order > 0

ORDER BY Estimated\_Days\_to\_Next\_Order DESC

# דו"ח המושתת על שימוש בפסקת WITH מורכבת (%16)

דוח רווחיות זרעים ברבעון האחרון, השוואה בין מחיר בסיס כמוצר בודד לבין מחיר ממוצע כחלק מקנייה, מחיר ממוצע כחלק מגינה בעיצוב אישי ומחיר ממוצע כחלק מגינה מעוצבת מראש.  
מטרת הדוח היא הצגת תמונת מצב של ביצועי הזרעים השונים בתקופת הזמן האחרונה שהייתה ובעקבות תמונת המצב לראות כיצד ניתן להגדיל את הרווחים באמצעות שיווק של גינות מעוצבות מראש או בעיצוב אישי בהתאם.

WITH

income\_per\_seed\_from\_designs\_per\_order AS (

SELECT Seed = C.Seed,

TotalIncome = (PRD.Price - PRD.Discount) \* DSG.Quantity \* C.Quantity / SUM(C.Quantity) OVER (PARTITION BY DSG.Name, DSG.DesignID),

QuantitySold = C.Quantity\*DSG.Quantity

FROM DESIGNS AS DSG

JOIN dbo.CHOSENS AS C

ON DSG.DesignID = C.Design AND DSG.Name = C.Garden

JOIN dbo.GARDENS AS G

ON DSG.Name = G.Name

JOIN dbo.PRODUCTS AS PRD

ON G.Name = PRD.Name

JOIN dbo.ORDERS AS O

ON DSG.OrderID = O.OrderID

WHERE DATEDIFF(QUARTER, O.OrderDate, GETDATE()) = 1

),

income\_per\_seed\_from\_premade\_per\_order AS (

SELECT Seed = PLT.Seed,

TotalIncome = (PRD.Price - PRD.Discount) \* I.Quantity \* PLT.Quantity / SUM(PLT.Quantity) OVER (PARTITION BY O.OrderID, I.Name),

QuantitySold = PLT.Quantity\*I.Quantity

FROM dbo.INCLUSIONS AS I

JOIN dbo.PRODUCTS AS PRD

ON I.Name = PRD.Name

JOIN dbo.GARDENS AS G

ON I.Name = G.Name

JOIN dbo.PLANTEDS AS PLT

ON G.Name = PLT.Garden

JOIN dbo.ORDERS AS O

ON I.OrderID = O.OrderID

WHERE DATEDIFF(QUARTER, O.OrderDate, GETDATE()) = 1

),

income\_per\_seed\_as\_simple\_product AS (

SELECT Seed = I.Name,

TotalIncome = (PRD.Price - PRD.Discount) \* I.Quantity,

QuantitySold = I.Quantity

FROM dbo.INCLUSIONS AS I

JOIN dbo.PRODUCTS AS PRD

ON I.Name = PRD.Name

JOIN dbo.ORDERS AS O

ON I.OrderID = O.OrderID

WHERE I.Name IN (SELECT Name FROM dbo.SEEDS) AND DATEDIFF(QUARTER, O.OrderDate, GETDATE()) = 1

),

OverallStat AS (

SELECT Seed = SeedIncome.Seed,

TotalIncome = SUM(SeedIncome.TotalIncome),

QuantitySold = SUM(SeedIncome.QuantitySold),

AvgPriceUnit = SUM(SeedIncome.TotalIncome)/SUM(SeedIncome.QuantitySold),

BasePrice = PRD.Price - PRD.Discount

FROM

(

SELECT Seed,

TotalIncome,

QuantitySold

FROM income\_per\_seed\_from\_designs\_per\_order

UNION

SELECT Seed,

TotalIncome,

QuantitySold

FROM income\_per\_seed\_from\_premade\_per\_order

UNION

SELECT Seed,

TotalIncome,

QuantitySold

FROM income\_per\_seed\_as\_simple\_product

) AS SeedIncome

JOIN dbo.PRODUCTS AS PRD

ON SeedIncome.Seed = PRD.Name

GROUP BY Seed, PRD.Price - PRD.Discount

),

DesignStat AS (

SELECT Seed = design.Seed,

TotalIncome = SUM(design.TotalIncome),

QuantitySold = SUM(design.QuantitySold),

AvgPriceUnit = SUM(design.TotalIncome)/SUM(design.QuantitySold),

BasePrice = PRD.Price - PRD.Discount

FROM income\_per\_seed\_from\_designs\_per\_order AS design

JOIN dbo.PRODUCTS AS PRD

ON design.Seed = PRD.Name

GROUP BY Seed, PRD.Price - PRD.Discount

),

PremadeStat AS (

SELECT Seed = premade.Seed,

TotalIncome = SUM(premade.TotalIncome),

QuantitySold = SUM(premade.QuantitySold),

AvgPriceUnit = SUM(premade.TotalIncome)/SUM(premade.QuantitySold),

BasePrice = PRD.Price - PRD.Discount

FROM income\_per\_seed\_from\_premade\_per\_order AS premade

JOIN dbo.PRODUCTS AS PRD

ON premade.Seed = PRD.Name

GROUP BY Seed, PRD.Price - PRD.Discount

)

SELECT Seed = OST.Seed,

[Total Income] = OST.TotalIncome,

[Quantity Sold] = OST.QuantitySold,

[Avg Price per Unit] = OST.AvgPriceUnit,

[Profit %] = CONCAT(CAST((OST.AvgPriceUnit - OST.BasePrice) / OST.BasePrice \* 100 AS varchar(7)), '%'),

[Designed - Avg Price per Unit] = CASE WHEN DST.QuantitySold IS NULL THEN OST.BasePrice ELSE DST.AvgPriceUnit END,

[Designed - Profit %] = CONCAT(CAST((DST.AvgPriceUnit - DST.BasePrice) / DST.BasePrice \* 100 AS varchar(7)), '%'),

[Premade - Avg Price per Unit] = PST.AvgPriceUnit,

[Premade - Profit %] = CONCAT(CAST((PST.AvgPriceUnit - PST.BasePrice) / PST.BasePrice \* 100 AS VARCHAR(7)), '%')

FROM OverallStat AS OST

LEFT JOIN DesignStat AS DST

ON OST.Seed = DST.Seed

LEFT JOIN PremadeStat AS PST

ON OST.Seed = PST.Seed

ORDER BY OST.AvgPriceUnit DESC

## מטלה 2 (%30 מציון הפרק הראשון) – כלים להצגת נתונים

**דו"ח עסקי** – הדו"ח העסקי מיועד למחלקת השיווק של החברה

**A screenshot of a computer

Description automatically generated**מסך ראשי:

**A screenshot of a graph

Description automatically generated**

**A screenshot of a computer

Description automatically generatedA screenshot of a graph

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**לוח מחוונים** – לוח המחוונים מיועד למנהל החברה

מסך ראשי:

A close-up of a graph

Description automatically generated