

SQL 2025 רועי עידן

**הקורס המקוון באתר עולה רק 240 שח
במקום 330 שח (27% הנחה).**

~~nw 330~~

המחיר היום 240 שח בלבד.

קישור לקורס המקוון המלא:

https://roy-

idan.co.il/courses/%d7%90%d7%a0%d7%9c%d7%99%d7%a1%d7%9
/8-sql

עמוד כל המבצעים שבאתר, סמסטר ב שנה א מנע"ס:

<https://roy-idan.co.il/%d7%9e%d7%97%d7%99%d7%a8%d7%95%d7%9f-%d7%a7%d7%95%d7%a8%d7%a1%d7%99%d7%9d-%d7%91%d7%99%d7%94%d7%a1-%d7%9c%d7%9e%d7%a0%d7%a2%d7%a1-%d7%a1%d7%9e%d7%a1%d7%98%d7%a8-%d7%91-2025-%d7%94%d7%9e%d7%9b>

כמות קורסים	אחוז הנחה על כל קורס
1 (קורס יחיד)	0%
2 קורסים	15% הנחה על כל קורס
3 קורסים	20% הנחה על כל קורס
4 קורסים	25% הנחה על כל קורס
5 קורסים	30% הנחה על כל קורס

חשוב:

כל שאילתה שסיימתם, יש להעתיק לתוך קובץ הוורד.

מגישים רק קובץ וורד.

לא מגישים קובץ עם סיומת SQL.

לשמור את קובץ הוורד בכונן D בלבד ולא ב DESKTOP

ללחוץ SAVE AS

לשמור כל 10 דקות את שכתבתם בקובץ הוורד...

קונטרול ואז S כי לעיתים מחשב נכבה או מפעיל עצמו מחדש.

כדאי להיכנס ל SQL עם הסיסמה לפני שזמן המבחן מתחיל

כי אתם בודקים למעשה שהמערכת שלכם תקינה

ולא מגלים זאת רק במבחן עצמו

להדפיס את מסד הנתונים עם תרגום לעברית

להביא מבחנים אחרונים ופתרונות מודפסים- המבחן עם חומר עזר

חלק א:

3שאלות "פתוחות" הכוללות שאילתות:

כל שאלה שווה 15 נקודות וס"ה חלק ב שווה 45 נקודות.

חלק ב:

12שאלות אמריקאיות כאשר כל שאלה שווה 4.5 נקודות.

ס"ה חלק ב שווה 54 נקודות.

המבחן שעתיים וחצי.

האם להריץ את האמריקאיות?

הרי נקבל את הקוד כתמונה ולא הייתי מבזבז זמן על להקליד את מה שרשום לכן את האמריקאי לא כדאי להריץ ב SQL

אלא אם יש לכם ספייר זמן.

האם לתרגל פתוחות? כן, כי כל פתוחה ניתנת להמרה לאמריקאית בקלות.

המון תיאורטיות חוזרות אז אם תביאו מבחנים ופתרונות תוכלו לאתר חלק מהשאלות במה שהבאתם.

כמובן שעדיף כבדיקה ועדיף לדעת החומר בעצמכם.

במודל יש 2 מבחנים לדוגמה + שאלות אמריקאיות שכולם רלבנטיים לחומר של הסמסטר.

גם מרתון האמצע רלבנטי כולו למבחן.

המבחן יכלול רק את הנושאים הבאים:

Select

From

Where

Getdate/Datediff/Datepart

Inner join/on

Order by

Group by

פעילות מתמטית

Having

Count

Distinct

null

נושאים שירדו ולא יופיעו במבחן הסופי:

כל סוגי תתי השאילתות ירדו ולא יהיה בכלל בחומר למבחן:

תת שאילתא ב WHERE, תת שאילתא ב SELECT, תת שאילתא ב HAVING, תת שאילתא ב FROM. כל אלו לא יהיו במבחן.

UNION- לא בחומר למבחן

LEFT JOIN- לא בחומר למבחן

RIGHT JOIN- לא בחומר למבחן

format

לסיכום, צריך לדעת למבחן הסופי רק את הנושאים שנלמדו במסגרת 6 השיעורים הראשונים בסמסטר.

אם נלך לפי הסילבוס אז צריך להכיר את כל הנושאים שיש בסילבוס עד וכולל שבוע 6.

שאלה 2 (5 נקודות)

אורן, בוגר קורס אנליסט SQL, כתב את השאילתה הבאה אשר מציגה את לקוחות החברה ונתונים אודות המכירות:

```
SELECT C.CompanyName,  
       AVG(od.UnitPrice) AS 'TotPrice',  
       MONTH(O.Orderid) AS 'MonthOrder'  
FROM Orders AS O INNER JOIN [Order Details] AS OD  
ON O.OrderID = OD.OrderID  
WHERE 'MonthOrder' = 9  
GROUP BY O.CustomerID
```

מהן הטעויות/השגיאות בשאילתה הנתונה של אורן-

יש למצוא לפחות 3 טעויות/השגיאות ?

טעות ראשונה:

לא ניתן להשתמש בכוני **MonthOrder** שכתבנו ב SELECT בשורת ה WHERE .

כי ה SQL מריץ קודם את ה where **לפני** ה SELECT

ולכן כש SQL נמצא בשורת WHERE הוא עדיין לא מכיר ועדיין לא יודע

מהו הכיוני **MonthOrder** לכן הוא לא יריץ כלום.

שימוש בכוני שנתנו ב SELECT אפשרי רק בשורת ה ORDER BY...

טעות שנייה:

בשורה הראשונה כתבו:

```
SELECT C.CompanyName,  
       CompanyName מטבלת לקוחות customers (טבלה C) בשלב ה SELECT .
```

כרגע ב from חסרה טבלת **customers**.

לכן כרגע לא ניתן להריץ.

יש להוסיף את טבלת **customers** לשורת ה from.

טעות שלישית:

אם כתבנו ב SELECT את שדה ה c.comanyName אז חובה לכתוב את השדה הזה גם ב group by

טעות רביעית:

שליפת החודש בשורה 3 צריכה לקרות משדה תאריך הזמנה ORDERDATE ולא ORDERID

שאלה 3 (5 נקודות)

גבריאלה אנליסטית בחברת ההשקעות "ביחד ננצח" התבקשה להפיק פלט כחלק מהליך הביקורת במשרד. בפלט יוצגו שמות לקוחות החברה וסך הכמות שרכשו ממוצרי החברה **ממוצרים שהקוד שלהם 31 ו-21**. ענת חברתה הטובה של גבריאלה המליצה לה להשתמש בטבלאות -לקוחות, הזמנות ופרטי מוצרים. **גבריאלה התעקשה מניסיונה הרב שצריכים גם את טבלת מוצרים.** הביעו דעתכם על הנאמר.

פתרון:

גבריאלה טועה ואין צורך בטבלת מוצרים כיוון שגם בטבלת פרטי מוצרים ORDER DETAILS יש את קוד המוצר ולכן אין צורך לצרף את טבלת מוצרים במיוחד בשביל קוד המוצר.

דרך טבלת פרטי המוצרים (Order Details) נוכל לחשב את סך הכמויות שנרכשו לכל לקוח כי בטבלה זו קיימים כבר הקודים של המוצרים, וב-2 הטבלאות המקושרות האחרות (טבלת לקוחות וטבלת הזמנות) נוכל להפיק את שאר המידע.

שאלה 1 (5 נקודות)

מנהל מחלקת המכירות של החברה מבקש מהאנליסטים של החברה לדעת מהו **סכום הפדיון הכללי בחברה בלבד**, אשר בוצע במחיר מכירה הגבוה/שווה למחיר המחירון. אודי והדר בוגרי הקורס אנליסט SQL הביעו דעתם לגבי בקשת מנהל המכירות אודי טוען שבדו"א זה יהיה צורך להשתמש ב `group by` שכן ישנה פעולה מתמטית (שימוש בפונקציית סיכום) הדרושה בדו"א זה יהיה צורך להשתמש ב `group by` וגם ב `having` שכן ישנה פעולה מתמטית וגם סינון. הביעו דעתכם על טענותיהם.

חזק חשוב:

אם מדובר **ברמת החברה כולה** והשאלתה כוללת חישוב מתמטי:

אין `GROUP BY`

אין `HAVING`

לפעמים יהיה `WHERE`

תמיד תהיה תוצאה אחת בלבד כי מדובר ברמת החברה.

פתרון השאלה:

שניהם טועים.

מדובר על פדיון ברמת החברה כולה!

אין `GROUP BY` (אין פה צורך ליצור קבוצות, ביקשו את כולם....ברמת החברה).

אין `HAVING` (כי אין `GROUP BY`).

יהיה `WHERE` כדי שנוודא שמחיר המכירה גבוה/שווה למחיר המחירון.

רק כשנעמוד בתנאי של הסינון הפדיון יצטרף לסכימה.

תהיה תוצאה אחת בלבד בפלט.

הסכום יבוצע עם `SUM` בשורת ה `SELECT`.

ב-`FROM` יכתבו הטבלאות הרלוונטיות והקשרים בניהם.

לא ביקשו לכתוב שאילתה, אבל רשמתי לצורך ההבנה:

```
use Northwind
select sum(od.Quantity * od.UnitPrice)
from [Order Details] as od inner join Products as p
on od.ProductID = p.ProductID
where od.UnitPrice >= p.UnitPrice
```

Results		Messages	
(No column name)			
1	976111.59		

use Northwind

```
select sum(od.Quantity * od.UnitPrice)
from [Order Details] as od inner join Products as p
    on od.ProductID = p.ProductID
where od.UnitPrice >= p.UnitPrice
```


סמנכ"ל פיתוח המוצרים בחברה מעוניין לקבל דו"ח אודות מוצרי החברה והקטגוריות אליהן הם משתייכים. לצורך כך, בוצעה שאילתת ה-SQL הבאה על טבלאות מוצרים וקטגוריות.

```
use Northwind
SELECT C.CategoryID, MAX(P.UnitPrice), SUM(P.UnitsInStock), AVG(P.UnitsInStock)
FROM Categories AS C INNER JOIN Products AS P ON C.CategoryID = P.CategoryID
WHERE P.UnitsInStock NOT BETWEEN 5 AND 35
GROUP BY C.CategoryID
```

התוצאות של שאילתה זו יציגו:

1. עבור כל קוד קטגוריה מוצר, את מחיר המחירון המקסימאלי ליחידת מוצר, סה"כ הכמות במלאי מהמוצרים המשתייכים לקטגוריה, והכמות הממוצעת במלאי מהמוצרים המשתייכים לקטגוריה
2. עבור כל שם קטגוריה מוצר, את מחיר המחירון המקסימאלי ליחידת מוצר, סה"כ הכמות במלאי מהמוצרים המשתייכים לקטגוריה, והכמות הממוצעת במלאי מהמוצרים המשתייכים לקטגוריה וזאת רק עבור מוצרים שהכמות במלאי מהם נמוכה מ-5 או גבוהה מ-35.
3. עבור כל קוד קטגוריה מוצר, את מחיר המחירון המקסימאלי ליחידת מוצר, סה"כ הכמות במלאי מהמוצרים המשתייכים לקטגוריה, והכמות הממוצעת במלאי מהמוצרים המשתייכים לקטגוריה וזאת רק עבור מוצרים שהכמות במלאי מהם נמוכה מ-5 או גבוהה מ-35.
4. עבור כל קוד קטגוריה מוצר, את מחיר המחירון המקסימאלי ליחידת מוצר, סה"כ הכמות במלאי מהמוצר, והכמות הממוצעת במלאי מכל מוצר וזאת רק עבור מוצרים שהכמות במלאי מהם נמוכה מ-5 או גבוהה מ-35.
5. אף אחת מהתשובות אינה נכונה

לפני שניתן לתשובות לבלבל אותנו

ננסה להבין מה עושה השאילתה

נתחיל ב FROM

אנו עושים שימוש בטבלאות מוצרים products וקטגוריות categories

שדה unitsinstock הוא כמות היחידות במלאי והוא שייך לטבלת מוצרים products

ליד ה WHERE יש תנאי סינון כאשר כתוב NOT BETWEEN

והמשמעות היא שרוצים להשאיר רק מוצרים שהכמות במלאי מהם נמוכה מ 5 או גבוהה מ 35

כעת עוברים לשורת ה SELECT

ונזהה קוד קטגוריה (אפשר גם לפי ה GROUP BY בו רשום לפי קוד קטגוריה)

לכן נבין שמציגים לכל קוד קטגוריה

לכל קוד קטגוריה מציגים את המקסימום של מחיר המחירון unitprice (של כל המוצרים שנמצאים באותה הקטגוריה)

לכל קוד קטגוריה מציגים גם את הסכום של הכמות במלאי (של כל המוצרים שנמצאים באותה הקטגוריה)

לכל קוד קטגוריה מציגים גם את הממוצע של הכמות במלאי (של כל המוצרים שנמצאים באותה הקטגוריה)

```
SELECT C.CategoryID, MAX(P.UnitPrice), SUM(P.UnitsInStock), AVG(P.UnitsInStock)
FROM Categories AS C INNER JOIN Products AS P ON C.CategoryID = P.CategoryID
WHERE P.UnitsInStock NOT BETWEEN 5 AND 35
GROUP BY C.CategoryID
```

לכל קוד קטגוריה	MAX(P.UnitPrice)	SUM(P.UnitsInStock)	AVG(P.UnitsInStock)
CategoryID	(No column name)	(No column name)	(No column name)
1	18.00	453	75
2	28.50	405	57
3	81.00	269	44
4	55.00	277	69
5	21.00	239	59
6	123.79	115	28
7	10.00	4	4
8	62.50	620	88

1 תיפסל כי גם התעלמו מהסיון...השארתי מי שמעל 35 ונמוך מ 5.

2 תיפסל רק כי אמור להתחיל ב "עבור כל קוד קטגוריה מוצר" וכתבו "עבור כל שם קטגוריה מוצר", שאר 2 נכונה.

3 נכונה!!!

4 תיפסל כי

4. עבור כל קוד קטגוריה מוצר, את מחיר המחירון המקסימלי ליחידת מוצר, סה"כ הכמות במלאי מהמוצר, והכמות הממוצעת במלאי מכל מוצר וזאת רק עבור מוצרים שהכמות במלאי מהם נמוכה מ-5 או גבוהה מ-35.

כי חסר אחרי ה X האדום הראשון את המילים "המשתייכים לקטגוריה".

גם במקום ה X האדום השני צריך היה לרשום את המילים "המשתייכים לקטגוריה" במקום "מכל מוצר".

use Northwind

```
SELECT C.CategoryID, MAX(P.UnitPrice), SUM(P.UnitsInStock), AVG(P.UnitsInStock)
```

```
FROM Categories AS C INNER JOIN Products AS P ON C.CategoryID = P.CategoryID
```

```
WHERE P.UnitsInStock NOT BETWEEN 5 AND 35
```

```
GROUP BY C.CategoryID
```

```
SELECT C.CategoryID, MAX(P.UnitPrice), SUM(P.UnitsInStock), AVG(P.UnitsInStock)
FROM Categories AS C INNER JOIN Products AS P ON C.CategoryID = P.CategoryID
WHERE P.UnitsInStock NOT BETWEEN 5 AND 35
GROUP BY C.CategoryID
```

	CategoryID	(No column name)	(No column name)	(No column name)
1	1	18.00	453	75
2	2	28.50	405	57
3	3	81.00	269	44
4	4	55.00	277	69
5	5	21.00	239	59
6	6	123.79	115	28
7	7	10.00	4	4
8	8	62.50	620	88

שאלה 2:

אחד ממנתחי הנתונים בחברה קמעונאית בנה את שאילתת ה-SQL-הבאה על טבלת עובדים במטרה להציג את הערים של העובדים המועסקים בחברה ואת מספר העובדים שמגיעים מכל עיר.

```
use Northwind
SELECT E.City, COUNT(E.EmployeeID) AS 'Num of Employees'
FROM Employees AS E
WHERE E.LastName LIKE 'D%'
GROUP BY E.City;
```

להלן מובאות מספר טענות:

- טענה 1: השאילתה תציג את מספר העובדים, ששם משפחתם מתחיל באות "D"
- טענה 2: השאילתה תציג את שמות הערים ועבור כל עיר יוצג מספר העובדים ששם משפחתם מתחיל באות "D"
- טענה 3: אם הקריטריון בפסוקית ה-WHERE-היה 'E.LastName LIKE '%D%' היו ככל הנראה מצפים לקבל יותר תוצאות

אילו מבין הטענות נכונה?

1. טענות 2 ו-3 בלבד
2. טענות 1 ו-3 בלבד
3. טענה 3 בלבד
4. טענה 2 בלבד
5. כל הטענות נכונות

עוסקים כאן בטבלת עובדים

Employees

נתחיל בניתוח הסינון

WHERE

נשאיר רק את השורות בהן שם המשפחה מתחיל ב D

ה GROUP BY מקבץ לקבוצות של שמות ערים

מה אנו מציגים :SELECT

שמות הערים

לכל שם עיר ספרנו את כמות העובדים שמגיעים לעבודה מאותה העיר (ספרנו את קוד העובד)

ומדובר רק בעובדים ששם משפחתם מתחיל ב D

לכן טענה 2 נכונה

טענה 1 תיפסל כי לא הזכירו שהכל קובץ לפי שמות הערים

טענה 3:

הפעם הסינון אומר להשאיר את כל העובדים ששם משפחתם מכיל את האות D כלומר:

שם המשפחה מתחיל ב D, שם המשפחה מסתיים ב D, שם המשפחה מכיל באמצע במיקום כלשהוא את D

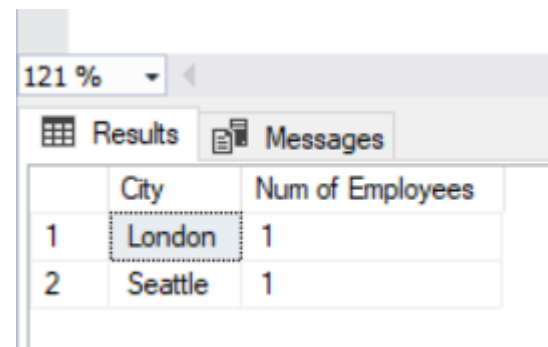
- שם משפחה שמתחיל ב D Davis

- שם משפחה שמסתיים ב D Reed

- שם משפחה שמכיל D באמצע Middleton

טענה 3 נכונה

לסיכום, נסמן תשובה 1 נכונה.



	City	Num of Employees
1	London	1
2	Seattle	1

כאן יש הקלדה של השאילתה המקורית מהשאלה:

```
use Northwind
```

```
SELECT E.City, COUNT(E.EmployeeID) AS 'Num of Employees'
```

```
FROM Employees AS E
```

```
WHERE E.LastName LIKE 'D%'
```

```
GROUP BY E.City;
```

```
use Northwind
SELECT E.City, COUNT(E.EmployeeID) AS 'Num of Employees'
FROM Employees AS E
WHERE E.LastName LIKE 'D%'
GROUP BY E.City;
```

להלן מובאות מספר טענות:

```
use Northwind
SELECT O.CustomerID, MAX(O.OrderDate), SUM(O.Freight), COUNT(O.OrderID)
FROM Orders AS O
GROUP BY O.CustomerID
```

בתוצאות השאילתה ניתן יהיה למצוא בין היתר את:

1. את תאריך ההזמנה הראשון של כל לקוח
2. סך עלויות ההובלה שנגבו מכל לקוח
3. מספר המוצרים בהזמנת לקוח
4. אף אחת מהתשובות האחרות אינה נכונה
5. יש יותר מתשובה אחת נכונה

מדובר על טבלת ההזמנות ORDERS

יוצרים קבוצות של לקוחות לפי קודי לקוחות

מציגים את תאריך ההזמנה האחרון (המקסימלי, שהכי קרוב להיום)...לא הראשון כמו שתשובה 1 טוענת.

מציגים את סכום עלות ההובלה

סופרים את כמות ההזמנות (לפי קודי הזמנות)

הכל לפי קוד לקוח....

אגב, תאריך מינימלי (התאריך הראשון) הוא התאריך שהוא רחוק ככל האפשר (בעבר) מהתאריך של היום.

תשובה 2 נכונה.

3 תיפסל כי סופרים כמות הזמנות לכל לקוח

ואין פה בשום שלב ספירה של מספר מוצרים שהיו בהזמנה של כל לקוח.

```
use Northwind
```

```
SELECT O.CustomerID, MAX(O.OrderDate), SUM(O.Freight), COUNT(O.OrderID)
```

```
FROM Orders AS O
```

```
GROUP BY O.CustomerID
```

```
use Northwind
```

```
SELECT O.CustomerID, MAX(O.OrderDate), SUM(O.Freight), COUNT(O.OrderID)
```

```
FROM Orders AS O
```

```
GROUP BY O.CustomerID
```


121 %

Results Messages

	CustomerID	(No column name)	(No column name)	(No column name)
1	ALFKI	1998-04-09 00:00:00.000	225.58	6
2	ANATR	1998-03-04 00:00:00.000	97.42	4
3	ANTON	1998-01-28 00:00:00.000	268.52	7
4	AROUT	1998-04-10 00:00:00.000	471.95	13
5	BERGS	1998-03-04 00:00:00.000	1559.52	18
6	BLAUS	1998-04-29 00:00:00.000	168.26	7
7	BLONP	1998-01-12 00:00:00.000	623.66	11
8	BOLID	1998-03-24 00:00:00.000	191.17	3
9	BONAP	1998-05-06 00:00:00.000	1357.87	17
10	BOTTM	1998-04-24 00:00:00.000	793.95	14
11	BSBEV	1998-04-14 00:00:00.000	281.31	10
12	CACTU	1998-04-28 00:00:00.000	72.76	6
13	CENTC	1996-07-18 00:00:00.000	3.25	1
14	CHOPS	1998-04-22 00:00:00.000	367.24	8
15	COMMI	1998-04-22 00:00:00.000	187.82	5
16	CONSU	1998-01-22 00:00:00.000	51.02	2

אגב, כל החישובים המתמטיים והפונקציות החישוביות min avg sum max וכדומה

מבוצעות תמיד על הטבלה הנמוכה ביותר בהיררכיה שקיימת בתוך ה from

אם הייתה גם טבלה עם **ORDER DETAILS** ב FROM

אז החישובים היו קורים על השדות של **הטבלה הזאת** כי היא נחשבת לנמוכה.

אז לסיכום, אם למשל רוצים חישוב SUM של עלות הובלה

ויש שתי טבלאות האחת של ORDERS והשנייה ORDER DETAILS

אז החישוב יהיה חייב לקרות על שדה מטבלת ORDER DETAILS כי היא נחשבת לנמוכה בהיררכיה.

אי אפשר יהיה לבצע את החישוב על שדה מ ORDERS במידה ויש טבלה נמוכה יותר כמו ORDER DETAILS בסביבה במקביל.

אגב, מקרה מיוחד שכן יאפשר לבצע חישוב על טבלת ORDERS

למרות שטבלת ORDER DETAILS תהיה גם כן במקביל ב FROM

הוא שימוש ב COUNT עם DISTINCT וזה כן יאפשר לנו לבצע חישוב מהטבלה הגבוהה יותר (ORDERS)

Count distinct...

שאלה 4:

נתונה שאילתת ה-SQL הבאה:

```
use Northwind
SELECT O.ShipCity, COUNT(O.OrderID) AS 'Count of Orders'
FROM Orders AS O
GROUP BY O.ShipCity;
```

מנכ"ל החברה ביקש לקבל ניתוח מדויק יותר של התוצאות ולקבל את הנתונים לגבי ערי המשלוח (**O.ShipCity**) ברלין, ברצלונה ובריסל ורק במידה ומספר ההזמנות **COUNT(O.OrderID)** גדול מ 1.

באילו פסוקיות בשאילתה יש להציב כל אחד מהקריטריונים על מנת לקבל את המבוקש?

1. בפסוקית ה **WHERE**-יוצב הקריטריון של מספר ההזמנות, ואילו בפסוקית ה **HAVING**-יוצב הקריטריון של ערי המשלוח.

2. שני הקריטריונים יוצבו בפסוקית ה **HAVING**-כיוון שנעשה שימוש בשאילתה בפסוקית **GROUP BY**.
3. אין חשיבות למיקום הקריטריון בשאילתה. בכל פסוקית שנציב את הקריטריון תתקבל התוצאה המבוקשת.
4. בפסוקית ה **WHERE**-יוצב הקריטריון של ערי המשלוח, ואילו בפסוקית ה **HAVING**-יוצב הקריטריון של מספר ההזמנות

5. שני הקריטריונים יוצבו בפסוקית ה- **WHERE** ביחס של "וגם" (**AND**) ביניהם

כמו שראינו עד עתה, **GROUP BY** יכול להגיע גם ללא **HAVING**.

כמו שרשום בשאלה הזאת המקורית!

מנגד, **HAVING** תמיד מגיע יחד עם **GROUP BY**.

אם כבר רושמים **HAVING** המבנה חייב להיות כך:

Group by

having

HAVING הולך ביחד עם **GROUP BY** הם שילוב!

רק **HAVING**

ללא **GROUP BY** זהו מצב בלתי אפשרי

תשובה 4 נכונה

❌ שימוש אסור בתוך WHERE

כתיבה ב-WHERE

WHERE SUM(Amount) > 1000



WHERE AVG(Price) < 50



WHERE MAX(Salary) > 10000



WHERE MIN(Date) < '2023-01-01'



WHERE COUNT(*) > 3



כך הקוד נראה לאחר הוספת הפקודות:

```
use Northwind
SELECT O.ShipCity, COUNT(O.OrderID) AS 'Count of Orders'
FROM Orders AS O
WHERE O.ShipCity IN ('Berlin', 'Barcelona', 'Brussels')
GROUP BY O.ShipCity
HAVING COUNT(O.OrderID) > 1;
```

use Northwind

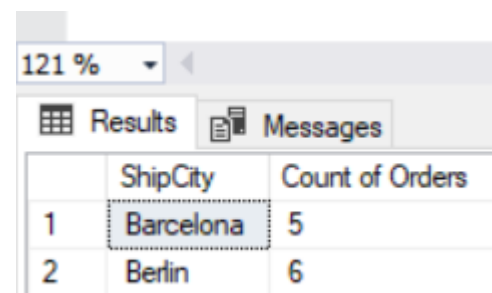
SELECT O.ShipCity, COUNT(O.OrderID) AS 'Count of Orders'

FROM Orders AS O

WHERE O.ShipCity IN ('Berlin', 'Barcelona', 'Brussels')

GROUP BY O.ShipCity

HAVING COUNT(O.OrderID) > 1;



	ShipCity	Count of Orders
1	Barcelona	5
2	Berlin	6


```
use Northwind
SELECT O.ShipCity, COUNT(O.OrderID) AS 'Count of Orders'
FROM Orders AS O
GROUP BY O.ShipCity;
```

use Northwind

SELECT O.ShipCity, COUNT(O.OrderID) AS 'Count of Orders'

FROM Orders AS O

GROUP BY O.ShipCity;

Results			Messages	
	ShipCity	Count of Orders		
1	Aachen	6		
2	Albuquerque	18		
3	Anchorage	10		
4	Århus	11		
5	Barcelona	5		
6	Barquisimeto	14		
7	Bergamo	10		
8	Berlin	6		
9	Bern	8		
10	Boise	31		
11	Bräcke	19		
12	Brandenburg	14		
13	Bruxelles	7		
14	Buenos Aires	16		
15	Butte	3		
16	Campinas	9		
17	Caracas	2		
18	Charleroi	12		
19	Colchester	13		
20	Cook	10		

שאלה 1

בהינתן הטבלאות הבאות של חברת הטכנולוגיה ProTech:

טבלת רכיבי מחשב (Components)

ComponentID	ProductID	UnitPrice	Quantity
66	W	80	27
55	Y	70	48
44	W	90	60
33	Y	40	40
22	V	80	11
11	W	50	9

טבלת מוצרים (Products)

ProductID	Name
Z	Tablet
Y	Phone
X	Laptop
W	Desktop
V	Watch

הציגו את תוצאות השאילתה הבאה:

```
SELECT P.Name, COUNT(C.ComponentID) AS 'Count', MIN(C.Quantity) AS 'Min'
FROM Products AS P inner JOIN Components AS C
ON P.ProductID = C.ProductID
GROUP BY P.ProductID, P.Name
```

שלבים:

מסמנים את העמודה החופפת שחוזרת בשתי הטבלאות: productID

גם מהשאילתה הנתונה ניתן לזהות כי המפתח המקשר הוא productid

ניתן למתוח קשת שמחברת הטבלאות דרך productid.

הולכים לפי הטבלה הראשונה שב FROM ובמילים אחרות לפי הטבלה שאצלה productid הוא המפתח הראשי: טבלת products

ב SELECT ביקשו שנוציג מטבלת המוצרים את שמות המוצרים מטבלת P ולכן:

מינימום הכמות (לכל קבוצת קוד מוצר)	ספירת רכיבים (לכל קבוצת קוד מוצר)	p.name
		Tablet
		Phone
		Laptop
		Desktop
		watch

חשוב לזכור בסוג הזה של השאלות שמותר להשאיר רק מוצרים שהקוד שלהם חזר בשתי הטבלאות!

ניגש לטבלה שיצרנו, נעבור מוצר מוצר ונשאיר אותה רק אם הוא משותף לשתי הטבלאות:

Z מטבלת מוצרים לא מופיע בטבלת רכיבים (התחתונה) ולכן שורת Z תימחק בטבלת המוצרים.

X מטבלת מוצרים לא מופיע בטבלת רכיבים (התחתונה) ולכן שורת X תימחק בטבלת המוצרים.

Y מופיע בשתי הטבלאות.

W מופיע בשתי הטבלאות.

V מופיע בשתי הטבלאות.

נבחין שב GROUP ביקשו לקבץ...להכין קבוצות של המוצרים.

ב SELECT גם ביקשו לכל קבוצת קוד מוצר, לספור כמה רכיבים יש לה (ספירה של componentID)

ב SELECT גם ביקשו לכל קבוצת קוד מוצר, למצוא את המינימום שבכמות (מינימום של quantity)

לקבוצת כל ה PHONE נספור כמות של קודי רכיב ונגלה 2 והמינימום בשדה "כמות" הוא 40

לקבוצת כל ה DESKTOP נספור כמות של קודי רכיב ונגלה 3 והמינימום "כמות" הוא 9

לקבוצת כל ה WATCH נספור כמות של קודי רכיב ונגלה 1 והמינימום "כמות" הוא 11

פתרון

Name	Count	Min
Phone	2	40
Desktop	3	9
Watch	1	11

שאלתה עם HAVING

```
SELECT ProductName, SUM(UnitPrice * Quantity) 'Revenue'
FROM Sales
GROUP BY ProductName
HAVING SUM(UnitPrice * Quantity) >= 200
```

	productname	unitprice	quantity
1	eggs	10	35
2	eggs	10	40
3	eggs	10	10
4	eggs	10	10
5	eggs	10	25
6	tofu	5	10
7	tofu	5	100
8	tofu	5	10
9	tofu	5	20
10	tofu	5	20
11	bread	15	10
12	bread	15	4
13	bread	15	8
14	bread	15	2
15	bread	15	2

שאלתה עם where

```
SELECT ProductName, SUM(UnitPrice * Quantity) 'Revenue'
FROM Sales
WHERE UnitPrice * Quantity >= 200
GROUP BY ProductName
```

	productname	unitprice	quantity
1	eggs	10	35
2	eggs	10	40
3	eggs	10	10
4	eggs	10	10
5	eggs	10	25
6	tofu	5	10
7	tofu	5	100
8	tofu	5	10
9	tofu	5	20
10	tofu	5	20
11	bread	15	10
12	bread	15	4
13	bread	15	8
14	bread	15	2
15	bread	15	2

הציגו את התוצאה שתתקבל בכל אחת משתי השאלות

ראשית, כדאי להוסיף בעצמינו עמודה קיצונית ימנית ולחשב את הפדיון:

שאילתה עם HAVING

```
SELECT ProductName, SUM(UnitPrice * Quantity) 'Revenue'  
FROM Sales  
GROUP BY ProductName  
HAVING SUM(UnitPrice * Quantity) >= 200
```

	productname	unitprice	quantity	revenue
1	eggs	10	35	350
2	eggs	10	40	400
3	eggs	10	10	100
4	eggs	10	10	100
5	eggs	10	25	250
6	tofu	5	10	50
7	tofu	5	100	500
8	tofu	5	10	50
9	tofu	5	20	100
10	tofu	5	20	100
11	bread	15	10	150
12	bread	15	4	60
13	bread	15	8	120
14	bread	15	2	30
15	bread	15	2	30

שאילתה עם where

```
SELECT ProductName, SUM(UnitPrice * Quantity) 'Revenue'  
FROM Sales  
WHERE UnitPrice * Quantity >= 200  
GROUP BY ProductName
```

	productname	unitprice	quantity	revenue
1	eggs	10	35	350
2	eggs	10	40	400
3	eggs	10	10	100
4	eggs	10	10	100
5	eggs	10	25	250
6	tofu	5	10	50
7	tofu	5	100	500
8	tofu	5	10	50
9	tofu	5	20	100
10	tofu	5	20	100
11	bread	15	10	150
12	bread	15	4	60
13	bread	15	8	120
14	bread	15	2	30
15	bread	15	2	30

הציגו את התוצאה שתתקבל בכל אחת משתי השאילתות

שאילתת ה WHERE:

WHERE מסנן שורות

במהלך הסינון רצים על כל הטבלה הגדולה

ומשאירים רק שורות שתוצאת הפדיון שלהם גבוהה מ 200.

יהיו שורות שיימחקו!

לאחר מכן נקבץ לקבוצות עם שמות המוצרים

לכל קבוצת שם מוצר נציג את סכום הפדיון של אותה קבוצה (מתחשבים רק בשורות שנותרו לאחר הסינון)

תוצאות:

שאלתה עם where

```
SELECT ProductName, SUM(UnitPrice * Quantity) 'Revenue'  
FROM Sales  
WHERE UnitPrice * Quantity >= 200  
GROUP BY ProductName
```

	productname	unitprice	quantity	revenue
1	eggs	10	35	350
2	eggs	10	40	400
3	eggs	10	10	100
4	eggs	10	10	100
5	eggs	10	25	250
6	tofu	5	10	50
7	tofu	5	100	500
8	tofu	5	10	50
9	tofu	5	20	100
10	tofu	5	20	100
11	bread	15	10	150
12	bread	15	4	60
13	bread	15	8	120
14	bread	15	2	30
15	bread	15	2	30

productName	revenue
Eggs	1000
tofu	500
bread	

שאלת ה **HAVING**:

שאלת ה **HAVING** עם

```
SELECT ProductName, SUM(UnitPrice * Quantity) 'Revenue'
FROM Sales
GROUP BY ProductName
HAVING SUM(UnitPrice * Quantity) >= 200
```

הפעם אין **WHERE**.

כלומר כל השורות נשארות!

אף שורה לא נמחקת!!!

HAVING לא מגיע לבד.

HAVING מגיע תמיד מתחת ל **GROUP BY**.

המבנה הקבוע הוא תמיד:

GROUP BY ...
HAVING ...

HAVING SUM(Price) > 1000

HAVING AVG(Price) < 50

HAVING MIN(Price) >= 5000

HAVING MAX(Price) < 40

HAVING COUNT(*) > 3

הרעיון הוא שקודם כל GROUP BY תיצור קבוצות,

למשל קבוצות של שמות מוצרים....

ואז HAVING תבצע סכום או ממוצע או מקסימום או מינימום או ספירה לכל הקבוצות

HAVING תבדוק אם החישוב שחל על הקבוצה עומד בתנאי ה HAVING

אם הקבוצה עמדה בתנאי ה HAVING היא תישאר

אם הקבוצה לא עמדה בתנאי ה HAVING היא תסונן (תימחק).

לסיכום,

קודם יוצרים קבוצות עם GROUP BY

נבחר חישוב (למשל סכום) ונפעיל אותו על כל הקבוצות בעזרת HAVING

תמיד ב HAVING יש תנאי...למשל שהתוצאה תהיה גדולה מ 200....

רק הקבוצות שיעמדו בתנאי יישארו

קבוצות שלא עמדו בתנאי יסוננו ויימחקו.

```
GROUP BY ProductName
HAVING SUM(UnitPrice * Quantity) >= 200
```

קבוצת EGGS:

Revenue	Quantity	UnitPrice	ProductName
350	35	10	Eggs
400	40	10	Eggs
100	10	10	Eggs
100	10	10	Eggs
250	25	10	Eggs
1,200	סה"כ Eggs		

קבוצת **כל שורות** קבוצת ה EGGS היא 1200 כלומר מעל 200,

עמדנו בתנאי ולכן הסכום של 1200 לקבוצת EGGS (יוצג כפי שביקשו ב SELECT)

```
HAVING SUM(UnitPrice * Quantity) >= 200
```

קבוצת **כל שורות** ה tofu היא 800 כלומר מעל 200,

עמדנו בתנאי ולכן הסכום של 800 לקבוצת TOFU יוצג (כפי שביקשו ב SELECT)

קבוצת TOFU:

50	10	5	Tofu
500	100	5	Tofu
50	10	5	Tofu
100	20	5	Tofu
100	20	5	Tofu
800	סה"כ Tofu		

קבוצת כל שורות ה BREAD היא 390 וזה מעל 200, עמדנו בתנאי ולכן יוצג כפי שביקשו ב SELECT

150	10	15	Bread
60	4	15	Bread
120	8	15	Bread
30	2	15	Bread
30	2	15	Bread
390	סה"כ Bread		

לכן תוצאת השאילתה התחתונה:

Eggs	1200
Tofu	800
bread	390

כמובן שאם בתנאי היה רשום גדול מ 1000 במקום גדול מ 200 אז רק שורת הביצים הייתה מוצגת!

שאלה מספר 1

א. מה ההבדל בין Where לבין Having?

WHERE הוא סינון של שורות...

רצים על כל השורות.

שורה שלא תעמוד בתנאי הסינון של WHERE: תימחק.

שורות שיעמדו בתנאי הסינון: יישארו!

HAVING הוא סינון מתמטי שמבוצע על קבוצות.

קודם חובה לקבץ לקבוצות עם GROUP BY

ורק אז נוכל לסנן את הקבוצות לפי התנאי שיהיה צמוד ל HAVING.

ב. מדוע בעת שימוש ב Group By נשתמש בקוד לקוח ובשם לקוח?

קוד לקוח הוא שדה ייחודי שבטוח מבדל ומזהה לקוח מלקוח אחר.

בעוד ששם לקוח יכול לחזור על עצמו בין לקוחות שונים.

הרעיון המרכזי בשימוש בקוד ושם לקוח הוא למנוע כפילויות

ולהצליח לזהות בצורה ייחודית כל לקוח.

שאלה מספר 3

נתונה השאילתה הבאה ובה מופיעה שגיאה

א. מהי השגיאה

ב. הסבירו מדוע זוהי שגיאה ותאר כיצד פועלת התוכנה לפי סדר פעולות.

ג. מה תפקיד השדה CustomerID בטבלת ORDERS

(תשובה ללא נימוק לא תזכה בניקוד כלל)

```
USE Northwind
SELECT C.CustomerID, C.CompanyName,
       SUM(OD.Quantity*OD.UnitPrice) AS 'SUM R'
FROM Customers AS C INNER JOIN Orders AS O
     ON C.CustomerID=O.CustomerID
     INNER JOIN [Order Details] AS OD
     ON O.OrderID=OD.OrderID
WHERE OD.Quantity*OD.UnitPrice BETWEEN 150 AND 550
     AND C.Country IN ('UK', 'USA')
GROUP BY C.CustomerID, C.CompanyName
HAVING 'SUM R' > 5000
ORDER BY 'SUM R' DESC
```

פתרון שאלה מספר 3

נתונה השאילתה הבאה ובה מופיעה שגיאה

א. מהי השגיאה

ב- Having עושים שימוש בבינוי לעמודה (הבינוי SUM R) אשר ניתן בשלב ה SELECT.

ב HAVING אסור להשתמש בבינוי SUM R.

כמובן שמותר ב SELECT לרשום כינוי לעמודה

אך מותר יהיה להשתמש בבינוי הזה רק בתוך ה ORDER BY ולא בתוך HAVING.

הבעיה היא שתמיד SQL תיגש קודם ל HAVING (לפני שלב ה SELECT)

ואז SQL לא תכיר את הבינוי (שכן הבינוי עדיין לא נוצר כי הרי רק בהמשך SQL תגיע ל SELECT תיצור ותכיר את הבינוי).

ב. הסבירו מדוע זוהי שגיאה ותאר כיצד פועלת התוכנה לפי סדר פעולות.

רק ב-ORDER BY ניתן לעשות שימוש בבינוי שנותרים.

מדוע זה אפשרי ב OREDR BY?

בכה שמגיעים לשלב המיון ORDER BY אחרי ש SQL יצרה ומכירה כבר עוד מלפני המיון את כל הכיכונים.

סדר ביצוע אמיתי של SQL Query (מאחורי הקלעים):

שלב	שלב ביצוע	תיאור
1	FROM	נטענות הטבלאות ומה שמתלווה אליהן (כמו JOIN)
2	WHERE	סינון שורות לפני קיבוץ (לפני פונקציות כמו SUM)
3	GROUP BY	קיבוץ השורות לפי עמודות
4	HAVING	סינון קבוצות אחרי חישובי פונקציות כמו SUM, COUNT, וכו'
5	SELECT	בחירת העמודות שיופיעו בפלט הסופי
6	ORDER BY	מיון התוצאה

ג. מה תפקיד השדה CustomerID בטבלת ORDERS

מדובר בשדה מפתח משני (מפתח זר) אשר משמש ליצירת קשר גומלין בין טבלת ORDERS לטבלת CUSTOMERS.

FROM Customers AS C INNER JOIN Orders AS O
ON C.CustomerID=O.CustomerID

Orders	הזמנות
OrderID	הזק הזמנה
CustomerID	קוד לקוח
EmployeeID	קוד עובד
OrderDate	תאריך הזמנה
RequiredDate	תאריך צורך
ShippedDate	תאריך משלוח
ShipVia	מספר משלוח
Freight	עלות הובלה
ShipName	שם משלוח
ShipAddress	כתובת משלוח
ShipCity	עיר משלוח
ShipRegion	אזור משלוח
ShipPostalCode	מיקוד משלוח
ShipCountry	מדינת משלוח

Customers	לקוחות
CustomerID	הזק לקוח
CompanyName	שם חברת לקוח
ContactName	איש קשר
ContactTitle	כותרת איש קשר
Address	כתובת
City	עיר
Region	אזור
PostalCode	מיקוד
Country	מדינה
Phone	טלפון
Fax	פקס

שאלה מספר 4

בחברת "פיצוחי הממלכה" בע"מ, מייצרים אריזות ובהן שילוב של סוגי אגוזים שונים
לפניהם מוצגות 3 הטבלאות הבאות: אגוזים (Nuts), מוצרים ומכירות (Products) ואריזות (Packages)

כתבו שאילתה שתציג לכל שם אגוז את סך הכמויות שנמכרו ממנו

Nuts	
שם אגוז	קוד אגוז
מלך	1122
מקדמיה	2233
ברזיל	3344
פקאן	4455
לוז	5566

Products		
קוד אריזה	קוד אגוז	כמות בגרם
111	1122	10
111	2233	10
111	3344	20
111	4455	20
222	2233	10
222	3344	20
222	4455	20
222	5566	30
333	2233	40
333	1122	20
333	5566	20

Packages	
קוד אריזה	סוג אריזה
111	שקית
222	קרטון
333	שקית מחבל

Nuts	
שם אגוז	קוד אגוז
מלך	1122
מקדמיה	2233
ברזיל	3344
פקאן	4455
לוז	5566

Products		
קוד אריזה	קוד אגוז	כמות בגרם
111	1122	10
111	2233	10
111	3344	20
111	4455	20
222	2233	10
222	3344	20
222	4455	20
222	5566	30
333	2233	40
333	1122	20
333	5566	20

Packages	
קוד אריזה	סוג אריזה
111	שקית
222	קרטון
333	שקית מחבל

כתבו שאילתה שתציג לכל שם אגוז את סך הכמויות שנמכרו ממנו

נחבר עם קשת את שדה המפתח שחוזר בשתי הטבלאות שהוזכרו בשאילתה.

למעשה לטבלת אריזות PACKAGES אין בכלל תפקיד בשאילתה שנכתוב.

לכן רק טבלת מוצרים וטבלת אגוזים יופיעו לנו ב FROM ונקשר אותן בעזרת שדה קוד אגוז.

נבין קבוצות של שמות האגוזים כי ביקשו "לכל שם אגוז".

נצטרך להציג את שמות האגוזים בטבלת "אגוזים" nuts.

ב SELECT נצטרך להפעיל סכימה SUM בטבלת מוצרים products על הכמות QUANTITY כי ביקשו זאת.

```

]use Northwind
]SELECT N.nutsname, SUM(P.Quantity) AS 'sum_quantity'
FROM Nuts AS N INNER JOIN
      Products AS P
      ON P.Nutsid=N.Nutsid
GROUP BY N.nutsid, N.nutsname

```

שאלה מספר 5

להלן שני פתרונות אותן הגישו שני סטודנטים. סטודנט A קיבל ציון מצוין וסטודנט B קבל ציון נכשל

הסבר מדוע סטודנט B קיבל ציון נכשל :

פתרון סטודנט A :

```

USE Northwind
SELECT S.SupplierID,S.CompanyName,
SUM(P.UnitsInStock) AS 'Products Units in Stock'
FROM Suppliers AS S INNER JOIN Products AS P
      ON S.SupplierID=P.SupplierID
WHERE S.CompanyName LIKE '[abc]%'
GROUP BY S.SupplierID,S.CompanyName

```

פתרון סטודנט B :

```

USE Northwind
SELECT S.SupplierID,S.CompanyName,
SUM(P.UnitsInStock) AS 'Products Units in Stock'
FROM Suppliers AS S INNER JOIN Products AS P
      ON S.SupplierID=P.SupplierID
      INNER JOIN [Order Details] AS OD
      ON P.ProductID=OD.ProductID
WHERE S.CompanyName LIKE '[abc]%'
GROUP BY S.SupplierID,S.CompanyName

```

פתרון:

סטודנט B צירף לטבלאות ב FROM את טבלת ORDER DETAILS

אבל בפועל במהלך התוכנית הוא לא עושה שימוש בשום שדה מהטבלה הזאת.

הוא לא משתמש בכלל בטבלה שהוא צירף.

טבלת ORDER DETAILS היא מראש טבלה מסוכנת ובעייתית
כי היא יוצרת כפילות ועיוות נתונים (כי היא הטבלה האחרונה בהיררכיה).
למשל, קוד מוצר חוזר גם בטבלת ORDER DETAILS וגם בטבלת PRODUCTS.

שאלות בחינה לדוגמה + פתרון

שאלתה מספר 3 (מועד א' סמסטר ב' תשפ"ג)

במחלקת הזמנות של החברה נתבקשו לבדוק תקופה מסוימת שהתגלתה כמאוד רווחית עבור החברה.

לשם כך הם דרשו בדוח נתונים לגבי עובדי החברה וההזמנות בהן טיפלו.

יוצגו שם מלא של העובד משורשר לתעודת זהות שלך (שם פרטי (רווח) שם משפחה (רווח) ומספר תעודת זהות),

תאריכי ההזמנות בהן טיפלו, רבעון ההזמנות, שנת ההזמנות, חודש ההזמנות ועלות ההובלה.

הנתונים יוצגו רק עבור הזמנות שהוזמנו בין התאריכים ה- 31/1/1997 ל- 30/06/1997.

```
SELECT E.FirstName+' '+E.LastName+' '+123456789 AS 'FullName',
O.OrderDate,
DATEPART(QUARTER,O.OrderDate) AS 'Quarter',
YEAR(O.OrderDate) AS 'YearOrder',
MONTH(O.OrderDate) AS 'MonthOrder',
O.Freight
FROM Employees AS E INNER JOIN Orders AS O
ON E.EmployeeID=O.EmployeeID
WHERE O.OrderDate BETWEEN '1997-01-31' AND '1997-03-30'
```

IX

```
SELECT E.FirstName+' '+E.LastName+' '+123456789 AS 'FullName',
O.OrderDate,
DATEPART(QUARTER,O.OrderDate) AS 'Quarter',
DATEPART(YEAR, O.OrderDate) AS 'YearOrder'
DATEPART(MONTH, O.OrderDate) AS 'MonthOrder'
O.Freight
FROM Employees AS E INNER JOIN Orders AS O
ON E.EmployeeID=O.EmployeeID
WHERE O.OrderDate BETWEEN '1997-01-31' AND '1997-03-30'
```

שאלתה מספר 4 (מועד א' סמסטר ב' תשפ"ג)

לאור התייקרות הדלק ברחבי העולם, מחלקת השילוח של החברה שוקלת לבצע העלאת מחירים.

אך לפני ביצוע העלאה ברצונה לבצע דגימה של סך עלויות ההובלה לכל חודש ושנה

מפניה אלייך היא דרשה 2 דו"חות:

הדו"ח הראשון יציג את סך העלות ההובלה בכל שנה וחודש של תאריך ההזמנה.

הדו"ח יתמקד בהזמנות שהוזמנו בשנת 1997 ועבור רבעונים זוגיים.

בדוח השני יופיע סך עלויות ההובלה הכללי בחברה עבור אותה שנה (שנת 1997) ואותם רבעונים זוגיים.

שאלתה 1

SELECT

```
YEAR(O.OrderDate) AS 'YEAR OF ORDER',  
DATEPART(Month,O.OrderDate) AS 'MONTH OF ORDER',  
SUM(O.Freight) AS 'SUM OF FREIGHT'
```

FROM Orders AS O

WHERE YEAR(O.OrderDate)=1997 AND DATEPART(QUARTER,O.OrderDate)%2=0

GROUP BY YEAR(O.OrderDate), DATEPART(Month,O.OrderDate)

שאלתה 2

SELECT SUM(O.Freight)

FROM Orders AS O

WHERE YEAR(O.OrderDate)=1997 AND DATEPART(QUARTER,O.OrderDate)%2=0

שאלתה מספר 5 (מועד א' סמסטר ב' תשפ"ג)

במחלקת משאבי אנוש החליטו שכחלק מתהליך קידום העובדים בחברה ברצונם לקבל פלט ובו להציג לכל עובד בחברה בכל שנת הזמנה את הנתונים הבאים:

- שם מלא של עובד החברה – FullName בשדה בשם
- השנה בה בוצע ההזמנה של כל עובד בשדה בשם YearOrder
- הפדיון המקסימלי לכל עובד בכל שנה בשדה בשם MaxRev
- כמות המוצרים הייחודיים בהם טיפל כל עובד בהזמנות בכל שנה בשדה בשם Num Unique Products

יש להתייחס להזמנות בהן הפדיון הוא מתחת ל-1,000 או מעל 3,000 אשר בוצעו בחודשים אי-זוגיים ולהזמנות קיים מיקוד (שדה ShipPostalCode)

או

יש לקחת בחשבון מוצרים המשתייכים לקטגוריות משקאות (beverages) ו-ממתקים (confections), אשר שם המוצר מכיל את האותיות I, S, R

```
Select E.FirstName+ ' ' + E.LastName As FullName,
        Year(O.OrderDate) As YearOrder,
        Max(Od.Quantity*OD.UnitPrice) As MaxRev,
        Count(Distinct Od.ProductID) As 'Num Unique Products'
From Employees As E inner join Orders As O
        On E.EmployeeID=O.EmployeeID inner join [Order Details] As OD
        On O.OrderID=OD.OrderID inner join Products As P
        On P.ProductID=Od.ProductID inner join Categories As C
        On C.CategoryID=P.CategoryID
where (Od.Quantity*OD.UnitPrice not between 1000 and 3000
        And Month(O.OrderDate) % 2=1
        And O.ShipPostalCode is not Null)
OR
        (C.CategoryName in ('beverages','confections')
        And P.ProductName like '%[ISR]%')
Group By E.EmployeeID, E.FirstName, E.LastName, Year(O.OrderDate)
```

יש הרבה טבלאות ב FROM

אך הכי נמוכה היא טבלת OD (order details) ולכן חישוב ה MAX וה COUNT יקרו מטבלת OD
אגב שליפת שנה לא נחשב לחישוב אמיתי אלא מדובר רק בשליפה