

SQL 2025 לחשבונאים

רועי עידן

מקצוען בהכנת סטודנטים למבחני התואר

מומחה למבחני המכללה למנהל

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

~~330 שח~~

המחיר כיום 240 שח בלבד.

הכנה מלאה תוך

כ 7 שעות צפייה בלבד!

קישור לקורס המקוון המלא:

<https://roy-idan.co.il/tduq>



עמוד כל המבצעים שבאתר, סמסטר ב שנה א חשבונאות: <https://roy-idan.co.il/rrk7>

שנה	שם הקורס
א	סטטיסטיקה ב לחשבונאים
א	יסודות המימון
א	אנליסט SQL
א	כלכלה לחשבונאים ב

אורך ממוצע לקורס מקוון כ 8 שעות בלבד!

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

מבנה המבחן לחשבונאים 2025, לראשונה:

15 שאלות אמריקאיות. ללא חומר עזר.

מבחן ברמה די קלה, הוגן- ברמה של המבחן לדוגמה שפורסם.

כדאי להדפיס את מסד הנתונים עם תרגום לעברית

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

ממועד א 2024: 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
```

ממן הטעויות/השגיאות בשאילתה הנתונה של אורן-

יש למצוא לפחות 3 טעויות/השגיאות ?

טעות ראשונה:

לא ניתן להשתמש ב־MonthOrder שכתבנו ב־SELECT בשורת ה־WHERE .

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

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

מהו הכינוי MonthOrder לכן הוא לא יריץ כלום.

שימוש בכינוי שנתנו ב־SELECT אפשרי רק בשורת ה־ORDER BY...

טעות שנייה:

בשורה הראשונה כתבו:

```
SELECT C.CompanyName,  
       C.CustomerID  
FROM Customers AS C  
WHERE C.CustomerID = 9
```

כלומר ביקשו להשתמש בשדה 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

כל שדה אשר מופיע ב־SELECT חייב להופיע ב־GROUP BY (מלבד שדות חישוביים: sum avg min max count)

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 התבקשה 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.

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:

- עבור כל קוד קטגוריה מוצר, את מחיר המחירון המקסימאלי ליחידת מוצר, סה"כ הכמות במלאי מהמוצרים המשתייכים לקטגוריה, והכמות הממוצעת במלאי מהמוצרים המשתייכים לקטגוריה
- 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.
- 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.
- 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.
- אף אחת מהתשובות אינה נכונה

נעת 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 קטגורית 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.

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 3:

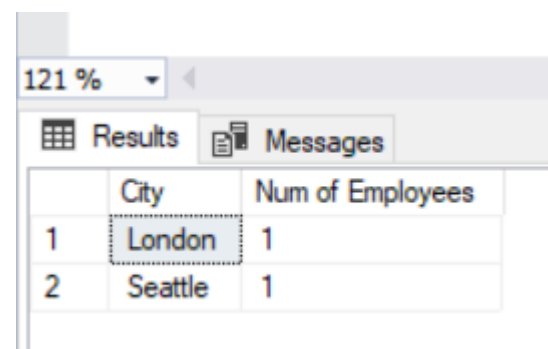
הפעם הסינון אומר להשאיר את כל העובדים ששם משפחתם מכיל את האות D כלומר:

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...

```

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 יצבו 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

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

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;

121 %

	ShipCity	Count of Orders
1	Barcelona	5
2	Berlin	6

קוד מקורי מהשאלה לפני השינויים שלנו:

```
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;

	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	Costa Mesa	10

שאלות חזרה - אנליסט SQL לחשבונאים

1. מחלקת המחקר של החברה ביקשה להפיק דו"ח לשיבת הנהלה הקרובה על בסיס מסד הנתונים Northwind. הדו"ח יציג את שם המוצר, קוד המוצר וקודי ההזמנות שבהן נרכש אותו מוצר. הדוח יכלול רק מוצרים שעבורם ממוצע הכמות במלאי גבוה מממוצע של כמות המלאי של מוצרים ששםם מתחיל באות M. איזו מבין הדרכים הבאות מתאימה ביותר לכתיבת השאילתה?

- א. שימוש בתת שאילתה בתוך פסוקית Where.
- ב. שימוש ב-Union עם פסוקיות Where ו-Having בכל אחת מהשאילתות.
- ג. שימוש בתת שאילתה בתוך פסוקית Having.
- ד. שאילתה רגילה הכוללת קיבוץ (Group by), תוך שימוש בפסוקיות Where ו-Having.
- ה. שאילתה רגילה, ללא קיבוץ (Group by), תוך שימוש בפסוקיות Where ו-Having.

"יציג" הוא החלק של ה SELECT :

יציג את שם המוצר, קוד המוצר וקודי ההזמנות שבהן נרכש אותו מוצר.

QUANTITY AVG

הדוח יכלול רק מוצרים שעבורם ממוצע הכמות במלאי

גבוה

מהממוצע של כמות המלאי של מוצרים ששםם מתחיל באות M

אין לנו את המספר של המסגרת הירוקה: לכן תת שאילתה

בנוסף, מדובר על משהו ספציפי ומדויק: כמות מלאי של מוצרים ששםם מתחיל באות M לכן מדובר בתת שאילתה

"רק" היא מילת סינון

WHERE היא סינון שורות (תצפיות)

HAVING היא סינון קבוצות

WHERE לא כוללת חישוב

HAVING תמיד כוללת חישוב

מזהים שתנאי הסינון כולל חישוב (AVG) לכן מדובר ב HAVING

(חוק: תנאי שכולל חישוב מדובר ב HAVING)

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

❌ 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

❌

ביקשו

QUANTITY AVG

רק מוצרים שעבורם ממוצע הכמות במלאי

רק (מילת סינון) + חישוב (AVG) אז בטוח מדובר ב HAVING

דרך נוספת:

תת השאילתה מושווית מול "ממוצע הכמות המלאי"

כלומר בהשוואה לחישוב (ממוצע)

לכן בטוח שתת השאילתה היא ב HAVING AVG

תשובה ה שגויה כי HAVING תמיד מופיע מתחת ל GROUP BY

מדובר במבנה קבוע

כלומר אם אין קיבוץ GROUP BY אז גם בטוח שאין כאן HAVING

לסיכום מדובר בשאילתה עם תת שאילתה

```

use Northwind
SELECT
    P.ProductID,
    P.ProductName,
    AVG(OD.Quantity) AS AvgQuantityOrdered
FROM
    Products AS P
JOIN
    [Order Details] AS OD ON P.ProductID = OD.ProductID
GROUP BY
    P.ProductID, P.ProductName
HAVING
    AVG(OD.Quantity) >
        (SELECT AVG(UnitsInStock)
         FROM Products
         WHERE ProductName LIKE 'M%');

```

	ProductID	ProductName	AvgQuantityOrdered
1	23	Tunnbrød	29
2	46	Spegesild	20
3	69	Gudbrandsdalsost	23
4	29	Thüringer Rostbratwurst	23
5	75	Rheinischer Klosterbier	25
6	15	Genen Shouyu	20
7	9	Mishi Kobe Niku	19
8	3	Aniseed Syrup	27
9	52	Filo Mix	17
10	72	Mozzarella di Giovanni	21
11	66	Louisiana Hot Spiced Okra	29
12	32	Mascarpone Fabioli	19
13	26	Gumbir Gummibirchen	23
14	12	Queso Manchego La Pastora	24
15	35	Steeleye Stout	24
16	63	Veggie-spread	26
17	6	Grandma's Boysenberry Spread	25

2. להלן תוצאה נדרשת משאילתה המתבססת על טבלאות הזמנות ועובדים בחברה:

FirstName	Last Date	AvgQuantity
שיר	2/2/2005	56
מיכל	1/1/1999	12
דניאל	3/2/1996	20
מורן	30/3/2012	55

כאשר:

- העמודה FirstName מתייחסת לשם הפרטי של העובד
- העמודה Last Date מתייחסת לתאריך הזמנה אחרון שהוזמן על ידי אותו העובד
- העמודה AvgQuantity מתייחסת לממוצע הכמויות שהוזמנו

במהלך ארוחת הצהריים התקיים דיון בין שלושה מעובדי מחלקת המחקר והפיתוח.
להלן שלוש טענות שעלו:

גילי:	"ניתן להציג את הטבלה הזו באמצעות GROUP BY לפי קוד העובד ושם העובד"
יעל:	"ניתן להציג את הטבלה הזו ללא שימוש ב-GROUP BY"
דנה:	"ניתן להציג את הטבלה הזו באמצעות GROUP BY לפי תאריך הזמנה בלבד"

מי מעובדי המחלקה צודק?

- רק דנה.
- רק יעל ודנה.
- רק דנה וגילי.
- רק גילי.
- שלושתם טועים.

חוק חמורבי:

ברגע שמשתמשים בפונקציה חישובית כמו sum avg count min max וגם יש שדה רגיל ללא חישוב

חובה יהיה לבצע קיבוץ לקבוצות GROUP BY

הפונקציות החישוביות דורשות שיהיה קיבוץ לקבוצות.

שדה חישובי max(orderdate)	
שדה רגיל	
FirstName	Last Date
שיר	2/2/2005
מיכל	1/1/1999
דניאל	3/2/1996
מורן	30/3/2012

למעשה אנו מבקשים לחשב

לכל עובד (לפי הקוד והשם) את ממוצע הכמויות ותאריך ההזמנה האחרון.

אנו רוצים תוצאות לכל עובד בנפרד,

לכן אנו מקבצים לפי כל עובד

(למשל המקסימום מבין כל ההזמנות ששייכות "לקבוצה" מסוימת.... כלומר לקבוצת העובד "שיר")

```
use Northwind
SELECT
    E.FirstName,
    MAX(O.OrderDate) AS [Last Date],
    AVG(OD.Quantity) AS AvgQuantity
FROM
    Employees AS E
JOIN
    Orders AS O ON E.EmployeeID = O.EmployeeID
JOIN
    [Order Details] AS OD ON O.OrderID = OD.OrderID
GROUP BY
    E.EmployeeID, E.FirstName
```

Results		Messages	
	FirstName	Last Date	AvgQuantity
1	Anne	1998-04-29 00:00:00.000	24
2	Janet	1998-04-30 00:00:00.000	24
3	Michael	1998-04-23 00:00:00.000	20
4	Robert	1998-05-06 00:00:00.000	26
5	Nancy	1998-05-06 00:00:00.000	22
6	Margaret	1998-05-06 00:00:00.000	23
7	Steven	1998-04-22 00:00:00.000	25
8	Andrew	1998-05-05 00:00:00.000	25
9	Laura	1998-05-06 00:00:00.000	22

3. להלן שאילתת SQL שבוצעה על טבלת הספקים מתוך מסד הנתונים Northwind:

```
SELECT COUNT(S.SupplierID)
FROM Suppliers AS S
```

המנכ"ל ביקש לעדכן את השאילתה כך שתציג את מספר הספקים בכל אזור בנפרד. אילו מהמרכיבים הבאים יש להוסיף לשאילתה כדי שתענה לדרישה?

- א. אף אחת מהתשובות האחרות אינה נכונה.
- ב. שדות S.Region בפסקית ה-SELECT ואת הפקודה: WHERE S.City like 'london'
- ג. שדה S.Region בפסקית ה-SELECT ואת הפקודה: GROUP BY S.Region
- ד. שדה S.Country בפסקית ה-SELECT
- ה. שדה S.Country בפסקית ה-SELECT ואת הפקודה: GROUP BY S.Country

במצב המוצא השאילתה פשוט סופרת באופן כללי את מספר הספקים.

```
SELECT COUNT(S.SupplierID)
FROM Suppliers AS S
```

Results		Messages
(No column name)		
1	29	

לאחר העדכון מעוניינים להציג את מספר הספקים לפי איזור, למשל:

איזור	כמות ספקים
צפון	64
מרכז	41
דרום	22

חוק חמורבי:

ברגע שמשתמשים בפונקציה חישובית כמו max min count avg sum וגם יש שדה רגיל ללא חישוב

חובה יהיה לבצע קיבוץ לקבוצות GROUP BY

הפונקציות החישוביות דורשות שיהיה קיבוץ לקבוצות.

שדה חישובי
count(supplier.id)
שדה רגיל

	Region	NumSuppliers
1	NULL	20
2	Asturias	1
3	LA	1
4	MA	1

לאחר התוספת:

```
SELECT
    S.Region,
    COUNT(S.SupplierID) AS NumSuppliers
FROM
    Suppliers AS S
GROUP BY
    S.Region;
```

	Region	NumSuppliers
1	NULL	20
2	Asturias	1
3	LA	1
4	MA	1
5	MI	1
6	NSW	1
7	OR	1
8	Qu?bec	2
9	Victoria	1

4. במהלך ההתמחות שלך במחלקה הכלכלית של משרדי O&D נתבקשת להפיק דוח שיציג עבור כל מדינה של לקוח ולכל עיר של לקוח את מספר ההזמנות שבוצעו על ידי לקוחות מאותה מדינה ועיר. הדו"ח יכלול רק מדינות וערים בהן בוצעו לפחות 5 הזמנות. בהתאם לכך, מהו השימוש הנכון בפסוקיות GROUP BY ו-HAVING במסגרת השאילתה?

א. GROUP BY C.Country, C.City
HAVING COUNT(*)>=5

ב. GROUP BY C.Country
HAVING COUNT(C.Country)>=5

ג. GROUP BY C.Country AND C.City
HAVING COUNT(*)=5

ד. GROUP BY C.City
HAVING COUNT(C.City)>=5

ה. אף אחת מהתשובות האחרות אינה נכונה

"לכל" / "עבור" אומר לנו GROUP BY

עבור כל מדינה ולכל עיר:

GROUP BY C.Country, C.City

"מספר ההזמנות" = פעולה חישובית של ספירה count

"רק" היא מילת סינון...סינון זה WHERE או HAVING.

מבקשים "רק ספירה של לפחות 5 הזמנות"

הסינון כולל חישוב (COUNT) לכן מדובר ב HAVING

הרי אין דבר כזה where count וברור שיש having count

```
HAVING SUM(Price) > 1000
```

```
HAVING AVG(Price) < 50
```

```
HAVING MIN(Price) >= 5000
```

```
HAVING MAX(Price) < 40
```

```
HAVING COUNT(*) > 3
```

❌ 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
```

❌

ג שגויה כי ההפרדה היא עם פסיקים ב GROUP BY ואסור AND או OR ב GROUP BY

```
SELECT
    C.Country,
    C.City,
    COUNT(O.OrderID) AS NumOrders
FROM
    Customers AS C
JOIN
    Orders AS O ON C.CustomerID = O.CustomerID
GROUP BY
    C.Country, C.City
HAVING
    COUNT(O.OrderID) >= 5
ORDER BY
    C.Country, C.City;
```

	Country	City	NumOrders
1	Argentina	Buenos Aires	16
2	Austria	Graz	30
3	Austria	Salzburg	10
4	Belgium	Bruxelles	7
5	Belgium	Charleroi	12
6	Brazil	Campinas	9
7	Brazil	Resende	9
8	Brazil	Rio de Janeiro	34
9	Brazil	Sao Paulo	31
10	Canada	Montreal	13
11	Canada	Tsawassen	14
12	Denmark	Copenhagen	11
13	Denmark	Copenhagen	7
14	Finland	Helsinki	7
15	Finland	Oulu	15
16	France	Lille	5
17	France	Lyon	10

```

select P.ProductName,
       AVG(OD.Quantity) AS 'AVG quantities'
FROM
  Products AS P
  INNER JOIN [Order Details] AS OD ON P.ProductID = OD.ProductID
WHERE
  P.CategoryID = (SELECT CategoryID
                  FROM Products AS P
                  WHERE P.ProductName LIKE 'Queso Cabrales' )

GROUP BY
  P.ProductID, P.ProductName
HAVING
  SUM(OD.Quantity) <
    (SELECT SUM(OD.Quantity)
     FROM Products AS P
     INNER JOIN [Order Details] AS OD ON P.ProductID = OD.ProductID
     WHERE P.ProductName LIKE 'CHAI' )

order by p.ProductName

```

מהי הבעיה העסקית שאותה באה השאילתה לפתור?

- דיויד מנהל המחסן של החברה ביקש פלט של מוצרים המשתייכים לקטגוריה של המוצר CHAI. הוא דרש להציג את רשימת שמות המוצרים הללו וממוצע הכמויות שנמכרו מהם, רק עבור מוצרים שהכמות שנמכרה מהם נמוכה מסך הכמות שנמכרה מהמוצר CHAI. את הדו"ח יש להציג לפי שם מוצר בסדר עולה.
- דיויד מנהל המחסן של החברה ביקש פלט של מוצרים המשתייכים לקטגוריה של המוצר Queso Cabrales. הוא דרש להציג את רשימת שמות המוצרים הללו וממוצע הכמויות שנמכרו מהם, רק עבור מוצרים שסך הכמות שנמכרה מהם נמוכה מסך הכמות שנמכרה מהמוצר CHAI. את הדו"ח יש להציג לפי שם מוצר בסדר עולה.
- דיויד מנהל המחסן של החברה ביקש פלט של מוצרים המשתייכים לקטגוריה של המוצר Queso Cabrales. הוא דרש להציג את רשימת שמות המוצרים הללו וממוצע הכמויות שנמכרו מהם, רק עבור מוצרים שנמוצע הכמות שנמכרה מהם נמוכה מממוצע הכמות שנמכרה מהמוצר CHAI. את הדו"ח יש להציג לפי שם מוצר בסדר יורד.
- דיויד מנהל המחסן של החברה ביקש פלט של מוצרים המשתייכים לקטגוריה של המוצר CHAI. הוא דרש להציג את רשימת שמות המוצרים הללו וסך הכמויות שנמכרו מהם, רק עבור מוצרים שסך הכמות שנמכרה מהם גבוהה מממוצע הכמות שנמכרה מהמוצר Queso Cabrales. את הדו"ח יש להציג לפי שם מוצר בסדר עולה.
- כל התשובות אינן נכונות.

Results			Messages	
	ProductName	AVG quantities		
1	Geitost	23		
2	Gudbrandsdalsost	23		
3	Mascarpone Fabioli	19		
4	Mozzarella di Giovanni	21		
5	Queso Cabrales	18		
6	Queso Manchego La Pastora	24		

השאילתה מציגה לכל שם מוצר את ממוצע הכמות שנמכרה ממנו

ה WHERE הוא סינון של שורות כך שיישארו רק שורות מוצרים שהקטגוריה שלהם היא קסו קבראלס.

מדובר בתת שאילתה ב WHERE.

יצרו קבוצות של כל מוצר ומוצר בעזרת GROUP BY

ה HAVING מסנן קבוצות.

יש כאן תת שאילתה ב HAVING:

יישארו רק קבוצות מוצרים שסכום הכמויות אצלן

נמוך

מסכום הכמות של המוצר CHAI .

א תיפסל כי:

מדובר על מוצרים המשתייכים לקטגוריה של קסו קבראלס

ולא לקטגוריית המוצר CHAI.

ג תיפסל כי אמרו בסוף "בסדר יורד" כשברירת המחדל היא "סדר עולה".

ד תיפסל כי:

מדובר על מוצרים המשתייכים לקטגוריה של קסו קבראלס

ולא לקטגוריית המוצר CHAI.

גם אמרו "גבוהה" במקום "נמוכה" ואמרו "ממוצע" במקום "סכום"

והחישוב של "סכום" צריך להיות על מוצר CHAI ולא על מהמוצר קסו קבראלס.

6. להלן קוד שאילתת SQL אשר הורץ על בסיס מסד הנתונים Northwind:

```
SELECT C.CompanyName, MIN(O.OrderDate) ,  
DATEDIFF(DAY,(SELECT MIN(O.OrderDate)  
FROM Orders AS O),MIN(O.OrderDate)) AS 'Diff'  
FROM Customers AS C INNER JOIN ORDERS AS O  
ON C.CustomerID=O.CustomerID  
GROUP BY C.CustomerID, C.Companyname
```

מהי הקביעה הנכונה ביותר?

- א. השאילתה תקינה והיא מציגה עבור כל לקוח את תאריך ההזמנה הראשונה שביצע ואת הפער בימים בין מועד זה לבין תאריך ההזמנה הראשונה שבוצעה בחברה.
- ב. השאילתה אינה תקינה, כיוון שהשימוש בתת שאילתה בתוך פונקציה הינו שגוי.
- ג. השאילתה תקינה, אך חסרה פסוקית HAVING בשל השימוש ב-GROUP BY.
- ד. השאילתה אינה תקינה, משום שלא ניתן להציג לכל לקוח את תאריך ההזמנה הראשונה שלו.
- ה. השאילתה אינה תקינה, כיוון שתת השאילתה כוללת שימוש של פונקציה מתמטית (MIN) דבר שאינו נתמך בפסוקית SELECT.

```
SELECT  
    C.CompanyName,  
    MIN(O.OrderDate),  
    DATEDIFF(  
        DAY,  
        (SELECT MIN(O.OrderDate) FROM Orders AS O),  
        MIN(O.OrderDate)  
    ) AS 'Diff'  
FROM  
    Customers AS C  
    INNER JOIN Orders AS O ON C.CustomerID = O.CustomerID  
GROUP BY  
    C.CustomerID, C.CompanyName;
```

קיבצו לקבוצות של לקוחות

```
GROUP BY  
    C.CustomerID, C.CompanyName;
```

עבור כל לקוח-

מציגים את תאריך ההזמנה הראשון (הכי ישן, הכי מינימלי) שבוצעה כל חברה.

Datediff היא פונקציה שתבצע הפרש בין תאריכים:

ההפרש יהיה במונחי ימים DAY

יש כאן תת שאילתה ב SELECT :

תמיד תת שאילתה ב SELECT מציגה ערך קבוע (שלא משתנה).

הערך הקבוע הוא תאריך ההזמנה הראשון (MIN) אי פעם בכל החברה כולה (זה לא משתנה מלקוח ללקוח)

לדוגמה, אם ההזמנה הראשונה בתולדות החברה בוצעה ב- 1996-07-04 זהו התאריך שיחזור כאן בכל שורה

פחות

תאריך ההזמנה המינימלי הראשון של אותו לקוח נוכחי שנבדק (כל פעם התאריך יהיה אחר כי כל פעם הלקוח יהיה אחר)

מחשבים אותו בכל שורת תוצאה של לקוח אחר – כי הוא בתוך GROUP BY C.CustomerID

עבור כל לקוח-

הפער בימים בין תאריך ההזמנה הראשון אי פעם בכל החברה לבין תאריך ההזמנה הראשון של אותו לקוח נוכחי שנבדק

א נכונה.

ב שגויה:

יש כאן שימוש תקין בתת שאילתה ב SELECT

ג שגויה:

GROUP BY

יכול להופיע לבדו (ללא HAVING)

דוגמה:

```
SELECT City, COUNT(*)  
FROM Customers  
GROUP BY City;
```

אם יש HAVING הוא חייב להגיע מתחת ל GROUP BY:

```
SELECT City, COUNT(*)  
FROM Customers  
GROUP BY City  
HAVING COUNT(*) > 5;
```

ד שגויה:

MIN עוזרת להציג את התאריך הראשון

ה שגויה

```

]SELECT
    C.CompanyName,
    MIN(O.OrderDate),
    DATEDIFF(
        DAY,
        (SELECT MIN(O.OrderDate) FROM Orders AS O),
        MIN(O.OrderDate)
    ) AS 'Diff'
FROM
    Customers AS C
    INNER JOIN Orders AS O ON C.CustomerID = O.CustomerID
GROUP BY
    C.CustomerID, C.CompanyName;

```

	CompanyName	(No column name)	Diff
1	Alfreds Futterkiste	1997-08-25 00:00:00.000	417
2	Ana Trujillo Emparedados y helados	1996-09-18 00:00:00.000	76
3	Antonio Moreno Taquer?a	1996-11-27 00:00:00.000	146
4	Around the Horn	1996-11-15 00:00:00.000	134
5	Berglunds snabbk?p	1996-08-12 00:00:00.000	39
6	Blauer See Delikatessen	1997-04-09 00:00:00.000	279
7	Blondesddsl p?re et fils	1996-07-25 00:00:00.000	21
8	B?lido Comidas preparadas	1996-10-10 00:00:00.000	98
9	Bon app'	1996-10-16 00:00:00.000	104
10	Bottom-Dollar Markets	1996-12-20 00:00:00.000	169
11	B's Beverages	1996-08-26 00:00:00.000	53
12	Cactus Comidas para llevar	1997-04-29 00:00:00.000	299
13	Centro comercial Moctezuma	1996-07-18 00:00:00.000	14
14	Chop-suey Chinese	1996-07-11 00:00:00.000	7
15	Com?rcio Mineiro	1996-08-27 00:00:00.000	54
16	Consolidated Holdings	1997-02-04 00:00:00.000	215
17	Frankenb?lt Delikatessen	1996-11-26 00:00:00.000	145

7. לביא מעוניין להריץ שאילתה על טבלת לקוחות, כך שתוצג בה אך ורק רשימת לקוחות ששם החברה שלהם מכיל את האות S או את האות M בלבד. איזה מבין התחבירים הבאים עונה על דרישתו של לביא?

- א. WHERE C.CompanyName LIKE '[MS]%'
- ב. WHERE C.CompanyName LIKE '[MS] '
- ג. WHERE C.CompanyName LIKE '%[MS] '
- ד. HAVING C.CompanyName LIKE '[MS]%'
- ה. WHERE C.CompanyName LIKE '%[MS]%'

הביטוי '%[MS]%' LIKE ב-SQL מחפש מחרוזות שמכילות לפחות אחת מהאותיות M או S, במיקום כלשהו במחרוזת.  דוגמאות שיחזירו תוצאה ('%[MS]%' LIKE):

למה זה מתאים?	מחרוזת
מכיל את האות M	MiX Corp
מכיל את האות S	Easy Solutions
מכיל גם M וגם S	MegaSoft

הסוגר המרובע משמעותו "או" בין האותיות שבסוגריים

8. אילו מהפסקוקיות הבאות תלויה בפסקוקית GROUP BY כדי לקבל שאילתת SQL תקינה?

- א. FROM
- ב. HAVING
- ג. ORDER BY
- ד. WHERE
- ה. INNER JOIN

GROUP BY

יכול להופיע לבדו (ללא HAVING)
דוגמה:

```
SELECT City, COUNT(*)
FROM Customers
GROUP BY City;
```

אם יש HAVING הוא חייב להגיע מתחת ל GROUP BY:

```
SELECT City, COUNT(*)
FROM Customers
GROUP BY City
HAVING COUNT(*) > 5;
```

9. שי כתב את השאילתה הבאה:

```
select year(o.OrderDate) as 'Year', month(o.OrderDate) as 'Month',  
sum(o.Freight) as 'Sum'  
from Orders as o  
where datepart(QUARTER,o.OrderDate)%2!=0 and Year=1997  
group by year(o.OrderDate)
```

ענת, עמיתה לצוות, עיינה בשאילתה וטענה כי קיימות בה מספר שגיאות.
כמה שגיאות יש בשאילתה?

- א. 3
- ב. 5
- ג. 2
- ד. 1
- ה. 0

שגיאה 1:

אסור להשתמש בכינוי YEAR בפסוקית ה WHERE

רק בפסוקית ה ORDER BY ניתן להשתמש בכינוי...

שגיאה 2:

חוק:

כל שדה רגיל (ללא חישוב) שיש ב SELECT

חייב להופיע גם ב GROUP BY

ב SELECT יש שני שדות רגילים (YEAR וגם MONTH)

אך ב GROUP BY יש רק YEAR, חסר MONTH

הקוד לאחר 2 התיקונים:

```
SELECT  
    YEAR(o.OrderDate) AS 'Year',  
    MONTH(o.OrderDate) AS 'Month',  
    SUM(o.Freight) AS 'Sum'  
FROM Orders AS o  
WHERE  
    DATEPART(QUARTER, o.OrderDate) % 2 != 0  
    AND YEAR(o.OrderDate) = 1997  
GROUP BY  
    YEAR(o.OrderDate), MONTH(o.OrderDate);
```

הערות כלליות:

תמיד מותר לרשום יותר שדות ב-GROUP BY ממה שמופיע ב-SELECT

אז האם כל שדה מ-GROUP BY חייב להופיע גם ב-SELECT? לא!

דוגמה

```
use Northwind
SELECT CategoryID
FROM Products
GROUP BY CategoryID, ProductName, UnitPrice, UnitsInStock
```

33 %

Results Messages

	CategoryID
1	1
2	1
3	1
4	1
5	1
6	1

חוק:

כל שדה מה-SELECT (מלבד חישובים count max min avg sum) חייב להופיע ב-group by:

```
SELECT CategoryID, ProductName, sum(UnitPrice), count(UnitsInStock), AVG(SupplierID)
FROM Products
GROUP BY CategoryID, ProductName
```

33 %

Results Messages

	CategoryID	ProductName	(No column name)	(No column name)	(No column name)
1	6	Alice Mutton	39.00	1	7
2	2	Aniseed Syrup	10.00	1	1
3	8	Boston Crab Meat	18.40	1	19
4	1	C?te de Blaye	263.50	1	18
5	4	Camembert Pierrot	34.00	1	28
6	8	Camarvon Tigers	62.50	1	7
7	1	Chai	18.00	1	1
8	1	Chang	19.00	1	1
9	1	Chartreuse verte	18.00	1	18
10	2	Chef Anton's Cajun Seasoning	22.00	1	2

השגיאה היא שיש שדה ב SELECT שהוא לא ב GROUP BY:

❌ דוגמה שגויה:

Edit Copy

sql

```
SELECT CategoryID, ProductName
FROM Products
GROUP BY CategoryID;
```

זאת צורת רישום תקינה כי לא עברנו על החוק:

חוק:

כל שדה מה SELECT (מלבד חישובים count max min avg sum) חייב להופיע ב group by:

```
SELECT CategoryID, count(ProductName)
FROM Products
GROUP BY CategoryID;
```

CategoryID	(No column name)
1	12
2	12
3	13
4	10
5	7

10. להלן קוד של שאילתת SQL אשר הורץ על בסיס מסד הנתונים Northwind:

```
select c.*  
from Customers as c left join Orders as o  
    on o.CustomerID=c.CustomerID  
where o.Orderid is null
```

מהי הקביעה הנכונה ביותר?

- א. השאילתה מציגה את פרטי ההזמנות של לקוחות החברה שביצעו הזמנות.
- ב. השאילתה אינה תקינה, מאחר שחסרה בה פסוקית GROUP BY.
- ג. השאילתה אינה תקינה כלל בשל אופן השימוש בפסוקית WHERE.
- ד. השאילתה מציגה את כל לקוחות החברה גם אלה שהזמינו וגם אלה שלא הזמינו.
- ה. השאילתה מציגה את כל פרטי הלקוחות שלא הזמינו כלל.

ה נכונה.

LEFT JOIN

נותן תמיד את כל הנתונים מכל הטבלאות (גם השורות המשותפות וגם הלא משותפות בין שתי טבלאות)

```
SELECT O.*  
FROM Customers AS c  
LEFT JOIN Orders AS o  
    ON o.CustomerID = c.CustomerID  
WHERE o.Orderid IS NULL;
```

where o.Orderid is null

אומר להשאיר רק את השורות שבהן קוד ההזמנה הוא 0 (בפועל לא היו הזמנות באותן שורות)

א שגויה כי עקב ה WHERE יוצגו פרטי הזמנות של לקוחות שלא ביצעו הזמנות

LEFT JOIN

נותן תמיד את כל הנתונים מכל הטבלאות (גם השורות המשותפות וגם הלא משותפות בין שתי טבלאות)

דוגמא

1. Customers

CustomerName	CustomerID
דני	1
יעל	2
מיכל	3

2. Orders

CustomerID	OrderID
1	101
1	102
2	103

◆ כלומר:

- דני (1) ביצע 2 הזמנות
- יעל (2) ביצעה 1
- מיכל (3) לא ביצעה בכלל

▼ נקבל:

OrderID	CustomerID
101	1
102	1
103	2
NULL	3

● רואים שמיכל (3) מופיעה — אבל אין לה הזמנה, לכן `OrderID = NULL`.

```
WHERE o.OrderID IS NULL;
```

OrderID	CustomerID
NULL	3

כלומר: מצאנו את מיכל — הלקוחה שאין לה אף הזמנה 

ומה יקרה אם נשתמש ב־ INNER JOIN ?

Edit ✎ Copy 📄

sql

```
SELECT c.CustomerID, o.OrderID
FROM Customers AS c
INNER JOIN Orders AS o ON o.CustomerID = c.CustomerID
WHERE o.OrderID IS NULL;
```

INNER JOIN מחבר רק שורות שיש להן התאמה (לקוחות שיש להם הזמנות)

התוצאה: 🔍

OrderID	CustomerID
101	1
102	1
103	2

WHERE o.OrderID IS NULL;

אתה בעצם אומר:

תביא לי את השורות שבהן אין מספר הזמנה.

התוצאה של כל השאילתה: ▼

OrderID	CustomerID
(שום דבר)	(שום דבר)

מה קורה אם <code>o.OrderID IS NULL</code>	מי מוצג	סוג JOIN
✓ מאתר את מי שאין לו הזמנה	כל הלקוחות (גם בלי הזמנה)	LEFT JOIN
✗ לא יהיו בכלל NULL, ולכן התנאי מחזיר כלום	רק לקוחות עם הזמנה	INNER JOIN

11. נתונה הפסוקית הבאה:

WHERE YEAR(E.BIRTHDATE) BETWEEN 1990 AND 1995 AND YEAR(E.BIRTHDATE)%5=0
איזו/אילו מהשנים הבאות תופיע/תופענה בתוצאות השאילתה?

- א. 1990, 1992, 1994
- ב. 1990, 1995
- ג. לא תתקבלנה תוצאות
- ד. 1990, 1991
- ה. 1990

YEAR(E.BIRTHDATE)%5=0

אם השנה תחולק ב 5 אז לא תהיה שארית

1990

1991

1992

1993

1994

1995

ב נכונה

12. במהלך דיון בקורס, שלוש סטודנטיות העלו טענות לגבי השימוש בסימן המודולו % בשאלות SQL:
- סטודנטית א' טענה כי ניתן להשתמש בסימן % רק כאשר מבקשים לייצר קריטריונים על מחרוזות טקסט בפסקית WHERE.
 - סטודנטית ב' טענה כי ניתן להשתמש בסימן % גם במחרוזות וגם בחישובים על שדות מספריים.
 - סטודנטית ג' טענה כי ניתן להשתמש בסימן % רק בחישוב תאריכים.
- מי מבין הסטודנטיות צודקת?

- רק סטודנטית ב' צודקת.
- רק סטודנטית א' צודקת.
- סטודנטית א' וסטודנטית ב' צודקות.
- שלושת הסטודנטיות טועות.
- רק סטודנטית ג' צודקת.

מודולו ניתן לשימוש במחרוזות או בבדיקת שארית על מספרים

```
SELECT 8 % 2 AS Result;      -- תוצאה: 0 (כי 8 מתחלק ב-2)
SELECT 9 % 4 AS Result;      -- תוצאה: 1 (כי 1 + 2*4 = 9)
```

מחפש לקוחות ששםם מתחיל באות A או B

```
WHERE CustomerName LIKE '[AB]%'
```

Avi

Ben

Amitai

Boris

סימן %:

- לא שייך רק לתאריכים
- אפשר להשתמש בו עם כל עמודה מספרית
- בתאריכים הוא מופיע רק כשמוציאים מהם חלק מספרי (כמו חודש או יום) ומשווים

לכן: הטענה של סטודנטית ג' שגויה. 

13. לקראת הכנס השנתי שייערך בקליפורניה בחודש ספטמבר הקרוב, נתבקשת להפיק דו"ח אחד בלבד אשר מציג רשימה שמית של שמות הלקוחות ושמות הספקים מקליפורניה ולצידה רשימה של כתובות הדוא"ל (מייל) של אותם לקוחות וספקים, בהתאמה. באילו מהמרכיבים הבאים יש להשתמש כחלק מקוד השאילתה כדי להפיק דו"ח כזה?

- א. פסוקית HAVING
- ב. פסוקית UNION
- ג. שימוש בריבוי טבלאות על ידי INNER JOIN
- ד. שימוש בריבוי טבלאות על ידי LEFT JOIN
- ה. לא ניתן להפיק דוח אחד מסוג זה.

א תיפסל כי אין כאן סינון חישובי (SUM COUNT AVG MIN MAX)

ב נכונה:

UNION אכן יוצר רשימה שמית

שאילתה אחת לספקים מקליפורניה

שאילתה שנייה ללקוחות מקליפורניה

נבצע UNION (איחוד) בין שתי השאילתות וכן נקבל את הרשימה השמית בדוח אחד בלבד בדיוק כפי שביקשו

```
USE Northwind;
```

```
SELECT
```

```
    CompanyName AS Name,  
    ContactName,  
    ContactTitle,  
    City,  
    Region,  
    Phone,  
    'Customer' AS Type
```

```
FROM Customers
```

```
WHERE Region = 'CA'
```

```
UNION
```

```
SELECT
```

```
    CompanyName AS Name,  
    ContactName,  
    ContactTitle,  
    City,  
    Region,  
    Phone,  
    'Supplier' AS Type
```

```
FROM Suppliers
```

```
WHERE Region = 'CA';
```

Results

Messages

Name	ContactName	ContactTitle	City	Region	Phone	Type
Let's Stop N Shop	Jaime Yorres	Owner	San Francisco	CA	(415) 555-5938	Customer

הורדתי את סינון ה WHERE כדי לקבל יותר תוצאות בפלט וכך נוכל להמחיש ולהבין טוב יותר את תרומת ה UNION:

	Name	ContactName	ContactTitle	City	Region	Phone	Type
1	Alfreds Futterkiste	Maria Anders	Sales Representative	Berlin	NULL	030-0074321	Customer
2	Ana Trujillo Emparedados y helados	Ana Trujillo	Owner	M?xico D.F.	NULL	(5) 555-4729	Customer
3	Antonio Moreno Taquer?a	Antonio Moreno	Owner	M?xico D.F.	NULL	(5) 555-3932	Customer
4	Around the Horn	Thomas Hardy	Sales Representative	London	NULL	(171) 555-7788	Customer
5	Aux joyeux eccl?siastiques	Guy?ne Nodier	Sales Manager	Paris	NULL	(1) 03.83.00.68	Supplier
6	B?lido Comidas preparadas	Mart?n Sommer	Owner	Madrid	NULL	(91) 555 22 82	Customer
7	Berglunds snabbk?p	Christina Berglund	Order Administrator	Lule?	NULL	0921-12 34 65	Customer
8	Bigfoot Breweries	Cheryl Saylor	Regional Account Rep.	Bend	OR	(503) 555-9931	Supplier
9	Blauer See Delikatessen	Hanna Moos	Sales Representative	Mannheim	NULL	0621-08460	Customer
10	Blondesddsl p?re et fils	Fr?d?rique Citeaux	Marketing Manager	Strasbourg	NULL	88.60.15.31	Customer
11	Bon app'	Laurence Lebihan	Owner	Marseille	NULL	91.24.45.40	Customer
12	Bottom-Dollar Markets	Elizabeth Lincoln	Accounting Manager	Tsawassen	BC	(604) 555-4729	Customer
13	B's Beverages	Victoria Ashworth	Sales Representative	London	NULL	(171) 555-1212	Customer
14	Cactus Comidas para llevar	Patricio Simpson	Sales Agent	Buenos Aires	NULL	(1) 135-5555	Customer
15	Centro comercial Moctezuma	Francisco Chang	Marketing Manager	M?xico D.F.	NULL	(5) 555-3392	Customer
16	Chop-suey Chinese	Yang Wang	Owner	Bern	NULL	0452-076545	Customer
17	Com?rcio Mineiro	Pedro Afonso	Sales Associate	Sao Paulo	SP	(11) 555-7647	Customer
18	Consolidated Holdings	Elizabeth Brown	Sales Representative	London	NULL	(171) 555-2282	Customer
19	Cooperativa de Quesos 'Las Cabras'	Antonio del Valle Saavedra	Export Administrator	Oviedo	Asturias	(98) 598 76 54	Supplier
20	Die Wandemde Kuh	Rita M?ller	Sales Representative	Stuttgart	NULL	0711-020361	Customer
21	Drachenblut Delikatessen	Sven Ottlieb	Order Administrator	Aachen	NULL	0241-039123	Customer

בתמונה הבאה אפשר לראות מה היה קורה אם לא היינו מבצעים UNION

כלומר

היינו מקבלים שני פלטים נפרדים עבור שתי שאלות נפרדות (שאלת לקוחות, שאלת ספקים)

```

USE Northwind;

-- לקוחות מקליפורניה
SELECT
    CompanyName AS Name,
    ContactName,
    ContactTitle,
    City,
    Region,
    Phone,
    'Customer' AS Type
FROM Customers
WHERE Region = 'CA';

-- ספקים מקליפורניה
SELECT
    CompanyName AS Name,
    ContactName,
    ContactTitle,
    City,
    Region,
    Phone,
    'Supplier' AS Type
FROM Suppliers
WHERE Region = 'CA';

```

Name	ContactName	ContactTitle	City	Region	Phone	Type
Let's Stop N Shop	Jaime Yorres	Owner	San Francisco	CA	(415) 555-5938	Customer

ג + ד ייפסלו כי שימוש בריבוי טבלאות ב FROM

```

use Northwind
USE Northwind;

SELECT
    CompanyName AS Name,
    ContactName,
    ContactTitle,
    City,
    Region,
    Fax,
    'Customer' AS Type
FROM Customers
WHERE Region = 'CA'

UNION

SELECT
    CompanyName AS Name,
    ContactName,
    ContactTitle,
    City,
    Region,
    Fax,
    'Supplier' AS Type
FROM Suppliers
WHERE Region = 'CA';

```

33 %

Results Messages

	Name	ContactName	ContactTitle	City	Region	Fax	Type
1	Let's Stop N Shop	Jaime Yorres	Owner	San Francisco	CA	NULL	Customer

14. בחברה נהוג לעקוב בקפדנות אחר תהליך קידום העובדים. מנהלת מחלקת משאבי אנוש פנתה אליך בבקשה להפיק דו"ח שנתי לעובדים, בו יופיעו עבור כל עובד ובכל שנת הזמנה:
- שם מלא של עובד בשדה בשם FullName
 - השנה בה בוצעה 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 לכלול בדו"ח אך ורק:

1. הזמנות בהן הפדיון הוא מתחת ל-1,000 או מעל 3,000 אשר בוצעו בחודשים אי-זוגיים וקיים בהן מיקוד (שדה ShipPostalCode)
 2. הזמנות שבהן המוצרים משתייכים לקטגוריות משקאות (beverages) וממתקים (confections) ושם המוצר מכיל את האותיות I, S, R
- איזה קוד SQL ייתן מענה לדרישה של מנהלת מחלקת משאבי אנוש?

א.

```
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
OR
    And O.ShipPostalCode is not Null)
(C.CategoryName in ('beverages','confections')
And P.ProductName like '%[ISR]%')
```

Group By E.EmployeeID, E.FirstName, E.LastName, Year(O.OrderDate)

ב.

```
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=0

And O.ShipPostalCode is not Null)

OR (C.CategoryName in ('beverages','confections')

And P.ProductName like '%[ISR]%)

Group By Year(O.OrderDate)

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

Group By E.EmployeeID, E.FirstName, E.LastName, Year(O.OrderDate)

Having (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]%)

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 or 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, FullName, Year(O.OrderDate)

ה.

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 or 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, FullName, Year(O.OrderDate)

15. להלן קוד שאילתת SQL:

SELECT P.ProductName, AVG(P.UnitPrice)

FROM Products AS P INNER JOIN [Order Details] AS OD

ON P.ProductID = OD.ProductID

GROUP BY P.ProductName

מהו ההיגד הנכון ביותר?

- א. תוצאת השאילתה הייתה זהה גם ללא מיזוג טבלת פרטי הזמנה.
- ב. תוצאת השאילתה הייתה זהה גם ללא השימוש בפסוקית GROUP BY.
- ג. השאילתה לא תעבוד מכיוון שלא ניתן כינוי לאחר הפונקציה AVG.
- ד. כל שאילתה מחייבת שימוש בפסוקית WHERE, ולכן במקרה זה השאילתה תציג שגיאה.
- ה. השאילתה תחזיר תוצאה זהה גם אם נעשה שימוש בפסוקית ה-WHERE בתנאי למוצרים המתחילים באות A

א נכונה:

אנו לא עושים שימוש בשום שדה מטבלת ORDER DETAILS

לכן

הצירוף של טבלת ORDER DETAILS הוא מיותר.

מיד נראה ששני הפלטים זהים (עם או ללא טבלת ORDER DETAILS)

פלט עם טבלת ORDER DETAILS:

```
SELECT P.ProductName, AVG(P.UnitPrice)
FROM Products AS P
INNER JOIN [Order Details] AS OD
ON P.ProductID = OD.ProductID
GROUP BY P.ProductName;
```

	ProductName	(No column name)
1	Alice Mutton	39.00
2	Aniseed Syrup	10.00
3	Boston Crab Meat	18.40
4	C'te de Blaye	263.50
5	Camembert Pierrot	34.00
6	Camarvon Tigers	62.50
7	Chai	18.00
8	Chang	19.00
9	Chartreuse verte	18.00
10	Chef Anton's Cajun Seasoning	22.00
11	Chef Anton's Gumbo Mix	21.35
12	Chocolade	12.75
13	Escargots de Bourgogne	13.25
14	Filo Mix	7.00
15	Flotemysost	21.50
16	Geitost	2.50
17	Genen Shouyu	15.50
18	Gnocchi di nonna Alice	38.00
19	Gorgonzola Telino	12.50
20	Grandma's Boysenberry Spread	25.00

פלט ללא טבלת ORDER DETAILS:

```
SELECT P.ProductName, AVG(P.UnitPrice)
FROM Products AS P
GROUP BY P.ProductName;
```

33 %

Results Messages

	ProductName	(No column name)
1	Alice Mutton	39.00
2	Aniseed Syrup	10.00
3	Boston Crab Meat	18.40
4	C?te de Blaye	263.50
5	Camembert Pierrot	34.00
6	Camarvon Tigers	62.50
7	Chai	18.00
8	Chang	19.00
9	Chartreuse verte	18.00
10	Chef Anton's Cajun Seasoning	22.00
11	Chef Anton's Gumbo Mix	21.35
12	Chocolade	12.75
13	Escargots de Bourgogne	13.25
14	Filo Mix	7.00
15	Flotemysost	21.50
16	Geitost	2.50
17	Genen Shouyu	15.50
18	Gnocchi di nonna Alice	38.00
19	Gorgonzola Telino	12.50
20	Grandma's Boysenberry Spread	25.00

ב שגויה:

יש כאן תפקיד חשוב ל GROUP BY והוא להציג תוצאות לכל שם מוצר, לפי קבוצות של המוצרים השונים.
כמו כן,

חובה להשתמש ב GROUP BY כי יש כאן שדה רגיל לצד שדה חישובי

שדה רגיל	שדה חישובי
P.ProductName,	AVG(P.UnitPrice)
ProductName	avg
Alice Mutton	39.00
Aniseed Syrup	10.00
Boston Crab Meat	18.40
C?te de Blaye	263.50
Camembert Pierrot	34.00
Camarvon Tigers	62.50
Chai	18.00
Chang	19.00
Chartreuse verte	18.00

ה שגויה כי שימוש בסינון שורות כנראה יצמצם את התוצאות כי שורות יימחקו.

יישארו נתוני שורות רק של מוצרים שמתחילים ב A.

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 צריך להשתמש ב group by שכן ישנה פעולה מתמטית (שימוש בפונקציית סיכום)
הדר טוענת שבדו"א זה צריך להשתמש ב group by וגם ב having שכן ישנה פעולה מתמטית וגם סינון.
הביעו דעתכם על טענותיהם.

חוק 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

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 שניציג 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:

שאלתה 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

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

הציגו את התוצאה שתתקבל בכל אחת משתי השאלות

שאלתה 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

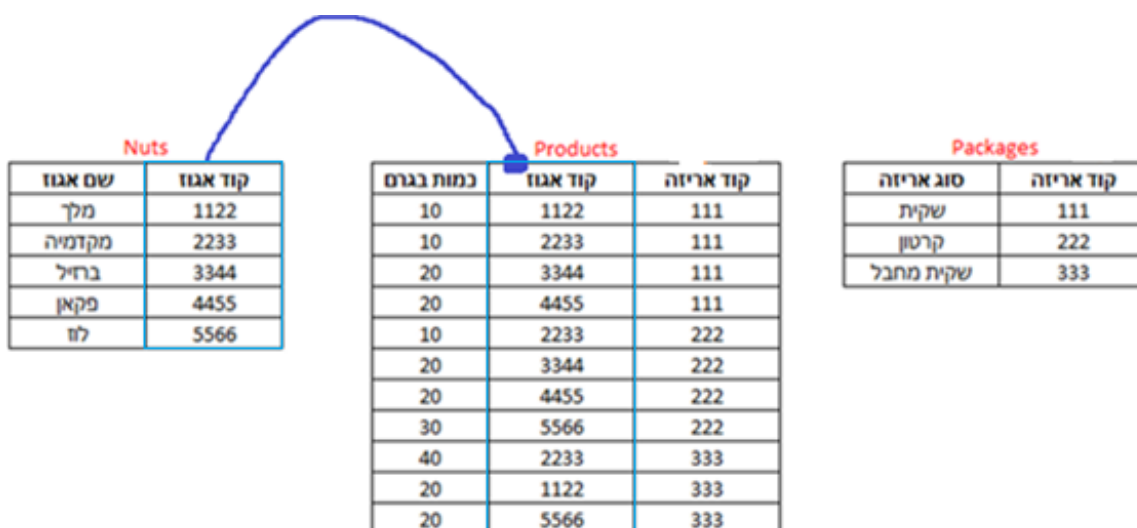
בחברת "פיצוחי הממלכה" בע"מ, מייצרים אריזות ובהן שילוב של סוגי אגוזים שונים
לפניהם מוצגות 3 הטבלאות הבאות: אגוזים (Nuts), מוצרים ומכירות (Products) ואריזות (Packages)

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



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 של (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