

## הרצאה מס' 6

### נושאים לשיעור זה:

- תזכורת: פונקציית הקבצה למול פקודת GROUP BY
- פקודת HAVING
- דוגמאות פשוטות ומורכבות לשימוש בפקודת HAVING
- שילוב פקודת DISTINCT עם פקודת COUNT
- פקודות איחוד בשפת SQL

### אוסף הפקודות שלמדנו עד כה:

```
SELECT X, count(Y)
FROM Students
WHERE ( ) AND ( )
GROUP BY X
ORDER BY X
```

תזכורת: פונקציית הקבצה למול פקודת GROUP BY:

Talmidim					
Account	Name	Street	Number	City	Amount
1	Avi	Begin	21	Ramat-Gan	1200
2	Avi	Begin	4	Ramat-Gan	3600
3	Ben	Hayarkon	147	Tel-Aviv	4000
4	Chen	Herzel	71	Tel-Aviv	2000
5	Debi	Hashalom	93	Givatiim	700
6	Zvi	Vaitzman	17	Givatiim	3500
7	Haim	Haela	65	Jerusalem	1000

(1) פונקציית הקבצה – קיימות חמש פונקציות המקבלות כקלט אוסף של ערכים ומחזירות ערך יחיד.



דוגמא: כמות הכסף הממוצעת שיושבת בחשבונות הסטודנטים

```
SELECT avg(Amount) AS AverageAmount
FROM Talmidim
```

AverageAmount
2286

(2) פקודת **GROUP BY** – נשתמש בה כאשר נרצה להפעיל את אחת מחמשת פונקציות ההקבצה על אוסף קבוצות של ערכים (ולא על קבוצה אחת של ערכים).



דוגמא: מציאת מספר הסטודנטים שגרים בכל רחוב בכל עיר

```
SELECT City , Street , count(Name) AS Counter
FROM Talmidim
GROUP BY City , Street
```

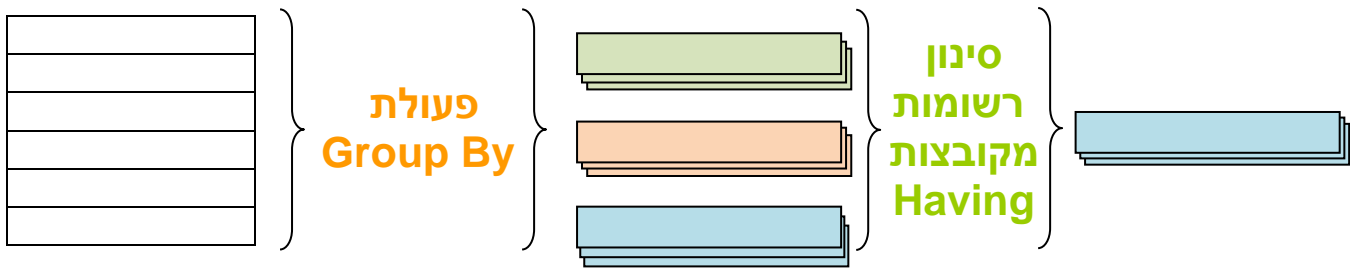
City	Street	Counter
Givatiim	Hashalom St.	1
Givatiim	Vaitzman St.	1
Jerusalem	Haela St.	1
Ramat-Gan	Begin St.	2
Tel-Aviv	Hayarkon St.	1
Tel-Aviv	Herzrl St.	1

כעת, אם נרצה להציג רק את שמות הערים בהם מס' הסטודנטים הוא לפחות 2, באיזה פקודה נשתמש ?

City	Street	Counter
Ramat-Gan	Begin St.	2

## פקודת HAVING

פקודת ה HAVING בוחרת קבוצות המקיימות תנאי מסוים (הצבת תנאי סינון על פונקציית הקבצה) לאחר שבוצעה פעולת הקבצה שהשתמשה ב - Group By.



## שילוב פקודת HAVING באוסף הפקודות שנלמדו

```

SELECT      City, Count(*)
FROM        Students AS S, Course AS C
WHERE       S.StudentID = C.StudentID
AND
S.StudentID BETWEEN 3344 AND 3355
GROUP BY   City
HAVING     Count(*) >2
ORDER BY   Name
    
```

## הסבר רכיבי הפקודה

פקודה המכילה את שמות העמודות (והפונקציות) שנרצה להציג בתוצאת השאילתא. יכיל את שמות הטבלאות שישתפו בשאילתא ומהם נרצה לשאוב את המידע להצגה. יכיל את תנאי הסינון (בשילוב AND / OR) שיופעלו על הטבלאות המוחזרות משלב ה FROM. יבצע קיבוץ של ערכי הטבלה בפילוח לפי שם העמודה המופיעה בו. יכיל תנאי סינון עבור הטבלה המקובצת שהתקבלה ע"י הפעלת GROUP BY. (פקודת HAVING יכולה להופיע רק אם לפנייה הופיעה פקודת GROUP BY). יבצע מיון של רשומות הטבלה על פי העמודה שנבחרה.

– SELECT  
– FROM  
– WHERE  
– GROUP BY  
– HAVING  
– ORDER BY

[דוגמא 1](#): שמות הערים וסכום הכסף **הממוצע** לסטודנט בכל עיר, כאשר יוצגו רק הערים בהם היתרה **הממוצעת** היא מעל 2500:

**שלב א'**: שליפת העיר והסכום הממוצע היושב בחשבונות הסטודנטים בה.

```
SELECT City , AVG(Amount) AS AvgAmount
FROM Talmidim
GROUP BY City
```

City	AvgAmount
Ramat-Gan	2400
Tel-Aviv	3000
Givatiim	2100
Jerusalem	1000

**שלב ב'**: נציג רק את הקבוצות שמקיימות את התנאי שהיתרה הממוצעת בעיר מסוימת היא מעל 2500:

```
SELECT City , AVG(Amount) AS AvgAmount
FROM Talmidim
GROUP BY City
Having AVG(Amount) > 2500
```

City	AvgAmount
Tel-Aviv	3000

**דוגמא 2:** מציאת שם הסטודנט והיתרה הממוצעת לכל סטודנט שגר ברמת-גן | שיש לו מעל שני חשבונות

(הערה: בשאלה זו נתייחס לשתי הרשומות הראשונות בטבלה כרשומות המתייחסות לאותו הסטודנט).

**שלב א':** מציאת היתרה הממוצעת לכל סטודנט שגר ברמת-גן:

```
SELECT Name, Avg(Amount) AS AvgAmount
FROM Talmidim
WHERE City = 'Ramat-Gan'
GROUP BY Name
```

Name	AvgAmount
Avi Cohen	2400

**שלב ב':** סינון של הסטודנטים שגרים ברמת-גן ויש להם מעל לשני חשבונות:

**\*\* בפועל ה DBMS יחשב 3 עמודות למרות שהעמודה השלישית הינה נסתרת מעיני המשתמש \*\***

Name	AvgAmount	Count
Avi Cohen	2400	2

```
SELECT Name, Avg(Amount) AS AvgAmount
FROM Talmidim
WHERE City = 'Ramat-Gan'
GROUP BY Name
Having Count(ChnNum) > 2
```

Name	AvgAmount
------	-----------

אין אפילו סטודנט אחד כזה.

**דוגמא 3 (מורכבת):** שמות הערים וסכום הכסף **המקסימאלי** בכל עיר, כאשר יוצגו רק הערים בהם היתרה **הממוצעת** היא מעל 2000:

**שלב א':** שליפת העיר והסכום המקסימלי היושב בחשבונות הסטודנטים בה.

```
SELECT City, Max(Amount) AS MaxAmount
FROM Talmidim
GROUP BY City
```

City	MaxAmount	AvgAmount
Ramat-Gan	3600	2400
Tel-Aviv	4000	3000
Givatiim	3500	2100
Jerusalem	1000	1000

קבוצה

חישוב זמני

**שלב ב':** נציג רק את הקבוצות שמקיימות את התנאי שהיתרה הממוצעת בעיר מסוימת היא מעל 2000:

```
SELECT City, Max(Amount) AS MaxAmount
FROM Talmidim
GROUP BY City
Having Avg(Amount) > 2000
```

City	MaxAmount
Ramat-Gan	3600
Tel-Aviv	4000
Givatiim	3500

פסוקית ה **HAVING** בעצם מגדירה אלו **קבוצות** נקבל בתשובה הסופית שלנו.

## לסיכום:

- (1) שנרצה להפעיל תנאי על הטבלה/הטבלאות המקורית נשתמש בפסוקית WHERE.
- (2) שנרצה להפעיל תנאי על הטבלה המקובצת נשתמש בפסוקית HAVING.
- (3) אם באותה שאילתא מופיע גם פסוק WHERE וגם פסוק HAVING, התנאי שמופיע בפקודת ה WHERE יתבצע קודם.

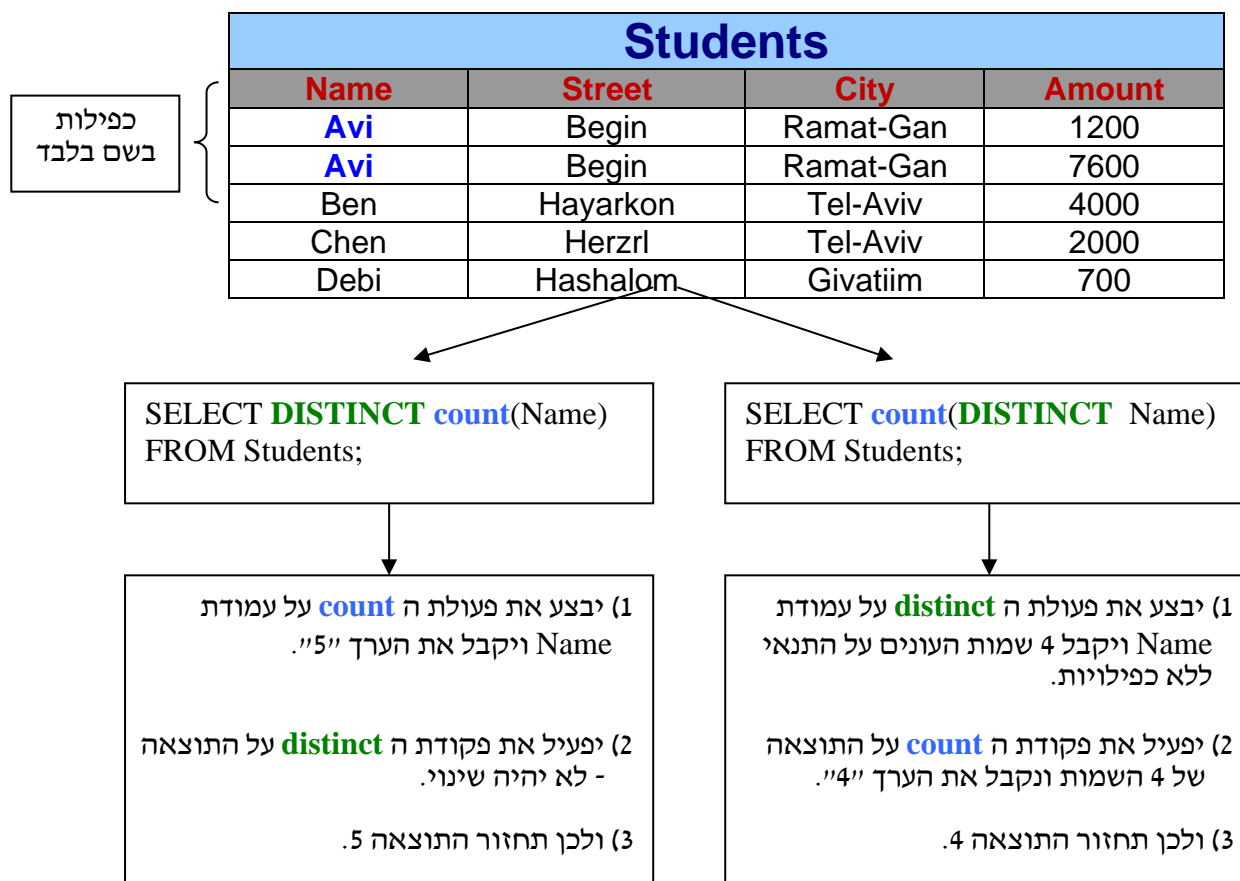
## סדר הפעולות בהרצת שאילתא:

1. נבחרות השורות שמקיימות את תנאי ה WHERE
2. השורות שנבחרו יקובצו לקבוצות לפי העמודות שנרשמו ב GROUP BY
3. נבחרות הקבוצות שמקיימות את התנאי שמופיע ב HAVING.

## סיום פקודת HAVING.



## שילוב פקודת DISTINCT ופקודת COUNT



**מסקנה: מיקום פקודת ה DISTINCT בביצוע פעולת COUNT משנה את התוצאות**

## פעולת איחוד UNION

בדומה לפעולת האיחוד הבינארית (המשלבת 2 טבלאות) שנלמדה באלגברת יחסים שמטרתה היא יצירת טבלה חדשה המכילה את כל השורות המופיעות לפחות באחד משתי טבלאות המקור ("או" ביניהם) קיימת פקודה מקבילה בשפת SQL.

**חשוב:** לפני שמבצעים איחוד, יש לוודא שסכמות היחסים אותן נרצה לאחד תואמות (compatible) כלומר, מתקיימים התנאים:

- 1) לשתי הטבלאות יש את אותו מספר עמודות (אותה דרגה).
- 2) תחום התכונה זהה, ז"א אותו type באיחוד 2 עמודות (לא נוכל לאחד string עם int).
- 3) הטבלאות המאוחדות יכולות להכיל שמות עמודות בעלי שמות שונים.

← התוצאה של פקודת ה UNION תחזיר טבלה שלא יהיו בה שורות כפולות.

**דוגמא:** נניח שקיימות טבלאות סטודנטים שונות באוניברסיטת בר אילן ובאוניברסיטת תל-אביב ושר החינוך היה רוצה לקבל רשימה אחת של כל הסטודנטים בשני המוסדות:

סכמת הסטודנטים בבר-אילן:

StudentID	StudentName	StudentCity	Department
1	a	RG	MA
2	b	BB	CS

סכמת הסטודנטים בתל-אביב:

SID	SName	Faculty	Department	City	Age
3	c	ES	PH	TA	22
4	d	ES	MA	TA	24

הבעיה הנראית לעין היא ששתי הסכמות כאן שונות ובכדי לאחד נתונים נצטרך סכמות זהות ולכן נכתוב את השאילתא הבאה:

**SELECT** StudentID **AS** ID, StudentName **AS** Name  
**FROM** BarIlanStudents

**UNION**

**SELECT** SID **AS** ID, SName **AS** Name  
**FROM** TelAvivStudents

ID	Name
1	a
2	b
3	c
4	d