



הרצאה מספר 6

נושאים לשיעור זה:

- תזכורת: פונקצית הקבצה למול פקודת GROUP BY ✓
- שילוב פקודת DISTINCT עם פקודת COUNT ✓
- פקודת HAVING ✓
- דוגמאות פשוטות ומורכבות לשימוש בפקודת HAVING ✓
- 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 :

פונקציית הקבצה - פונקציות המקבלות כקלט אוסף של ערכים ומחזירות ערך יחיד.

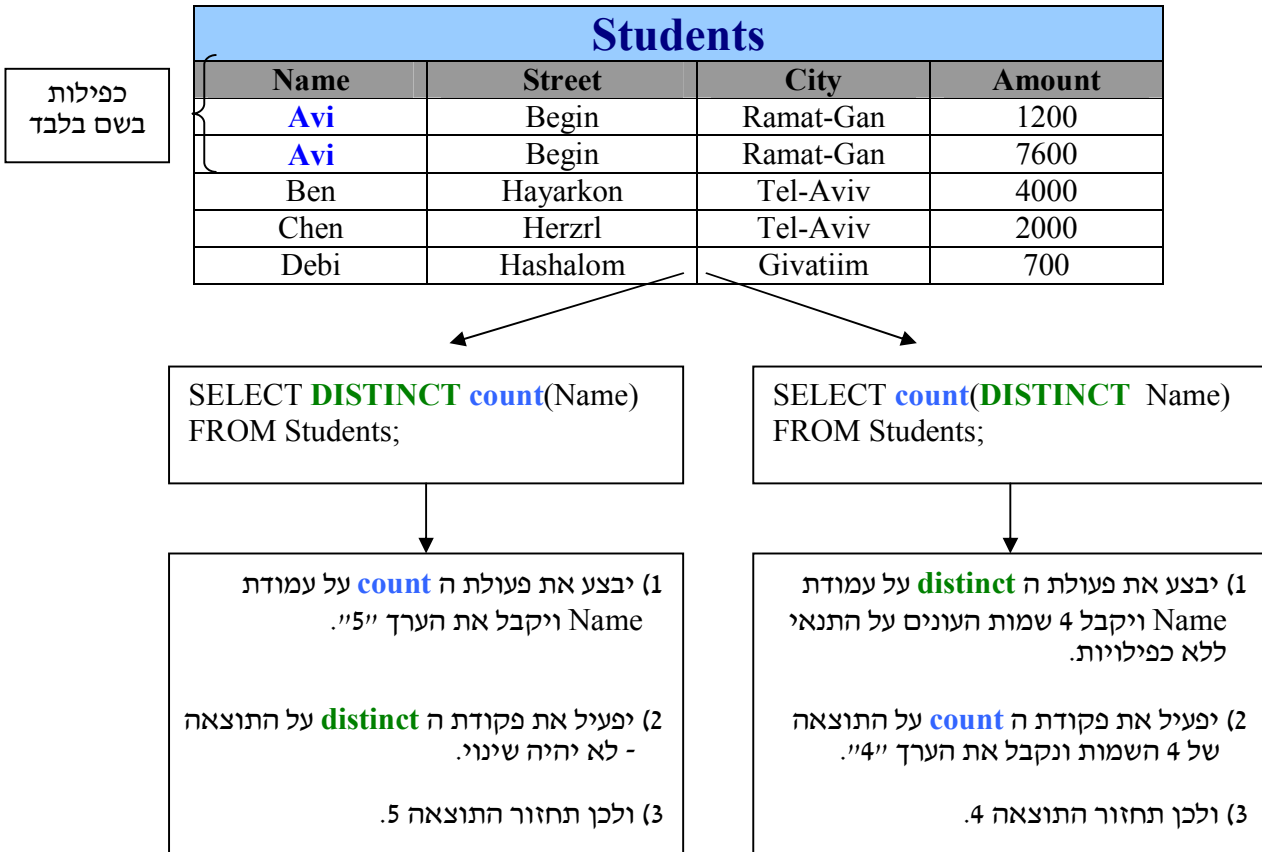


פקודת GROUP BY – נשתמש בה כאשר נרצה להפעיל את פונקציית ההקבצה על אוסף קבוצות של ערכים, (ולא על קבוצה אחת של ערכים).





שילוב פקודת DISTINCT ופקודת COUNT

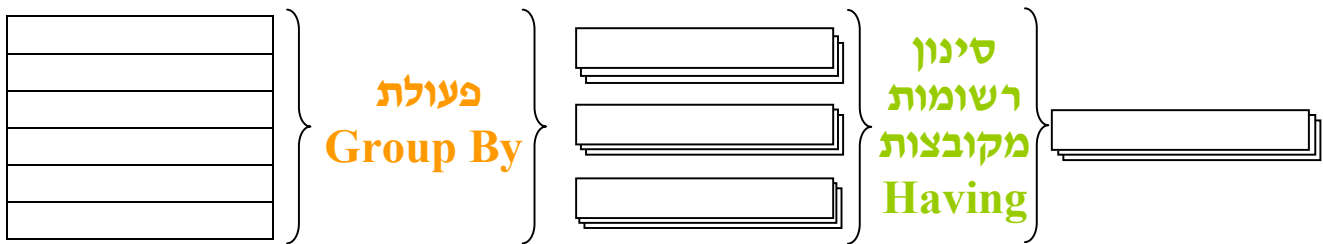


מסקנה: מיקום פקודת ה **DISTINCT** בביצוע פעולת **COUNT** משנה את התוצאות



HAVING פקודת

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



שילוב פקודת HAVING באוסף הפקודות שנלמדו

SELECT
FROM
WHERE

City, Count(*)
Students AS S, Course AS C
S.StudentID = C.StudentID

AND

S.StudentID BETWEEN 3344 AND 3355

GROUP BY
HAVING
ORDER BY

City
Count(*) >2
Name

הסבר רכיבי הפקודה

- **SELECT**
 - **FROM**
 - **WHERE**
 - **GROUP BY**
 - **HAVING**
 - **ORDER BY**
- 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 מ שלב 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.



מצורפת טבלה מתוך מסד נתונים לדוגמא שתמחיש את השימוש בפקודת ה **HAVING** בדוגמאות הבאות:

Students					
ChnNum	Name	Street	Number	City	Amount
1	Avi Cohen	Begin	21	Ramat-Gan	1200
2	Avi Cohen	Begin	4	Ramat-Gan	3600
3	Ben Levi	Hayarkon	147	Tel-Aviv	4000
4	Chen Levin	Herzrl	71	Tel-Aviv	2000
5	Debi Dvir	Hashalom	93	Givatiim	700
6	Zvi Kaner	Vaitzman	17	Givatiim	3500
7	Haim Itzhak	Haela	65	Jerusalem	1000

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 Students
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 Students
GROUP BY City
Having AVG(Amount) > 2500
```

City	AvgAmount
Tel-Aviv	3000



דוגמא 2 : מציאת היתרה הממוצעת לכל סטודנט שגר ברמת-גן | שיש לו מעל שני חשבונות :

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

```
SELECT Name, Avg(Amount) AS AvgAmount
FROM Students
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 :

```
SELECT Name, Avg(Amount) AS AvgAmount
FROM Students
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** לסטודנט בכל עיר, כאשר יוצגו רק הערים בהם **הממוצעת** היא מעל 2000:

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

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

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

קבוצה (Group) →

חישוב זמני (Temporary Calculation) ←

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

```
SELECT City, Max(Amount) AS MaxAmount
FROM Students
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.



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