

הרצאה מס' 5

נושאים לשיעור זה:

(א) אלגברת יחסים – סיום הנושא

- אלגברת יחסים – הפעולה התשיעית והאחרונה: חילוק.
- שאלת סיכום ב-אלגברת יחסים

(ב) שפת SQL

- פונקציות הקבצה פשוטות
- פונקציות הקבצה על אוסף קבוצת של ערכים - GROUP BY

$$R1 \div R2$$

9) חילוק (מנה) *Division*: \div

זו היא פעולה בינארית הבוחרת מתוך היחס $R1$ את בלוק הרשומות המתאימות לכל הרשומות ב $R2$ שהינן בעלי קידומת וסיימת זהות ומחזירה את התכונות הקיימות בעמודות של $R1$ אך לא קיימות ב $R2$.

זו פעולה המתאימה לשאלות בהן מופיעה המילה לכל / בכל.

הגדרה פורמאלית:

יהיו $R1(S1)$, $R2(S2)$ יחסים כך שמתקיים $S1 \subset S2$, היחס $R1 : R2$ הוא-בעל סכמה $S1-S2$.
רשומה בשם t ימצא בתוך יחס החילוק $R1 : R2$ אם ורק אם מתקיים:

$$t \in \Pi_{S1-S2}(R1)$$

(2) לכל רשומה, $t2$ ב $R2$ יש רשומה, $t1$ ב $R1$ המקיימים את שני התנאים הבאים:

- I) $t1[S2] = t2[S2]$
- II) $t1[S1-S2] = t$

הסבר סכמתי:

R1:

D	X	Y	Z
d	A	B	C
d	A	B1	C
d	A1	B1	C
d1	A1	B1	C

R2:

X	Y	Z
A	B	C
A	B1	C
A1	B1	C

$R1 \div R2$:

D
d

נוכל לראות שב $R1$ עבור הקידומת d קיימות 3 שורות ולאור העובדה ש 3 שורות אלו מופיעות גם ב $R2$ אזי (תנאי I מתקיים) בתוצאת החילוק יופיע הערך d המייצג את שלשת הרשומות הנ"ל (תנאי II מתקיים).

מכיוון שעבור הקידומת $d1$ יש רק שורה אחת ב $R1$ אך ב $R2$ יש 3 שורות אזי בעת ביצוע החילוק לא נקבל התאמה מלאה ולכן $d1$ לא יופיע בתוצאה.

דוגמא מסכמת לצירוף טבעי:

בסיס הנתונים לדוגמא (מכיל 4 טבלאות שעליהן נסביר את כל הדוגמאות)

Applicant – רשימת מועמדים לעבודה		
a_id	a_name	a_address
460480	Oren	Tel-Aviv
495332	Dana	Netanya
572460	Batya	Herzeliia
487725	Shalom	Hadera
552731	Rami	Tel-Aviv

Jobs – רשימת משרות	
job_no	Job_name
202	Programmer
223	Garden
230	Secretary
240	Librarian
242	Teacher

Qualified – רשימת המקצועות להם המועמדים מתאימים	
a_id	job_no
460480	242
460480	223
495332	230
572460	230
572460	223
572460	242
487725	202
552731	242

Wishes – רשימת המקצועות בהם המועמדים מעוניינים	
a_id	job_no
460480	223
495332	240
495332	242
572460	230
487725	202
487725	242
552731	202

שאלה: מצא את שמות המועמדים שמעוניינים להיות מתכנתים.

← לשם פתרון השאלה נצטרך להשתמש ב 3 היחסים : Applicant , Jobs , Wishes .

שלב 1: נרצה לקבל את רשימת המועמדים שמעוניינים בעבודה כלשהיא ומהו שם העבודה בה הם מעוניינים:

Wishes \bowtie Jobs

a_id	Job_no	Job_name

↑
שם העבודה
המשותפת בצירוף

שלב 2: נרצה לקבל את שמות המועמדים בעבודה כלשהיא , לשם כן נבצע צירוף עם טבלת Applicant המכילה את שמות העובדים:

Applicant \bowtie (Wishes \bowtie Jobs)

a_id	a_name	a_address	Job_no	Job_name

↑
שם העבודה
המשותפת בצירוף

שלב 3: נבחר את הרשומות מתוך רשימת המועמדים שהינם Programmer :

$\sigma_{\text{job_name} = \text{"programmer"}} (\text{Applicant} \bowtie (\text{Wishes} \bowtie \text{Jobs}))$

שלב 4: כעת נבצע הטלה על רשימה זו בכדי לקבל את רשימת שמות המתכנתים:

$\Pi_{\text{a_name}} (\sigma_{\text{job_name} = \text{"programmer"}} (\text{Applicant} \bowtie (\text{Wishes} \bowtie \text{Jobs})))$

סיום נושא אלגברת יחסים !!!

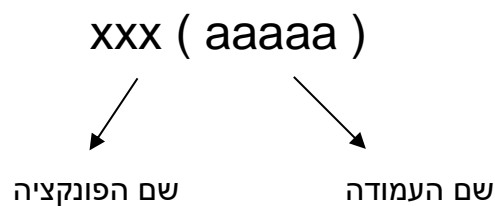
SQL:

(1) פונקציות הקבצה

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



לכל פונקציה שנשתמש ב SQL יש פורמט כתיבה קבוע:



אנו נלמד את הפונקציות הבאות:

- Avg מציאת ערך ממוצע.
- Min מציאת ערך מינימאלי.
- Max מציאת ערך מקסימאלי.
- Sum מציאת סכום ערכים.
- Count מציאת מספר האיברים בקבוצה.

הערה: אסור לבצע הרכבה של פונקציות: avg(min(amount)).

דוגמאות:

Talmidim					
ChnNum	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) סכום הכסף המינימאלי היושב בכל חשבונות הסטודנטים.

```
SELECT min(Amount)
FROM Talmidim
```

Amount
700

2) נרצה למצוא את כמות הכסף הממוצעת שיושבת בחשבונות הסטודנטים הגרים בת"א:

```
SELECT avg(Amount)
FROM Talmidim
WHERE City = 'Tel-Aviv'
```

Amount
3000

(3) מציאת סכום הכסף המקסימאלי מתוך כל חשבונות הסטודנטים לאחר הורדה של 10% עמלה חודשית מהסכום.

```
SELECT max(Amount – Amount * 0.10)
FROM Talmidim
```

Amount
3600

(4) סך כל הכסף היושב בכל חשבונות הסטודנטים ששם מכיל את האות 'i':

```
SELECT sum(Amount)
FROM Talmidim
WHERE Name like '%i%'
```

Amount
10000

(5) שליפה המבצעת ספירה של מספר הסטודנטים הקיימים ברשימת הסטודנטים של הבנק הגרים ברחוב המתחיל באות H ומרוויחים מעל 1200 ₪.

```
SELECT count (Name)
FROM Talmidim
WHERE (Street like 'H%')
and (Amount>1200)
```

Count
2

הערות לגבי פונקציות ההקבצה:

- 1) הפעולה `count(a)` תבצע **ספירה של כל הערכים הקיימים השונים מ NULL** בעמודה a (תהיה ספירה גם של חזרות); על אותו משקל כל הפעלת פעולת הקבצה לא תספור את ערכי ה NULL.
- 2) פעולת `count(*)` מבצעת ספירה של מספר השורות בטבלה (**כולל** שורות שמכילות ערכי NULL).
- 3) בדוגמא למעלה, אם סטודנט היה מופיע פעמיים כאשר בכל מקום הייתה לו כתובת שונה או שהוא היה בעל מספר חשבונות, היינו סופרים אותו פעמיים ולכן בשביל למנוע תופעה זו נצטרך להוסיף את רכיב מבטל הכפילויות - DISTINCT.

```
SELECT count (DISTINCT Talmidim.Name)
FROM Talmidim;
```

ע"י הוספת ה DISTINCT אנו בעצם מבטלים סטודנטים כפולים ורק לאחר מכן מבצעים את הספירה, ←
ז"א שנקבל את מספר הסטודנטים השונים הקיימים בטבלה.

4) הפעולה `count(DISTINCT *)` – אינה חוקית !!!

הערות כלליות:

- ✓ **תזכורת:** אסור לבצע הרכבה של פונקציות, לדוגמא: `avg (count ())`.
- ✓ כל הפונקציות פועלות על ערכים נומריים.
- ✓ הפונקציות MIN, MAX פועלות גם על מחרוזות בסדר לקסיקוגרפי.
- ✓ הפונקציה COUNT גם כן פועלות על מחרוזות.
- ✓ הפונקציות AVG, SUM לא פועלות על מחרוזות ואם יופעלו עליהן יחזירו ערך 0.

GROUP BY (2)

לפעמים נרצה להפעיל את פונקציית ההקבצה לא על קבוצה אחת של ערכים, אלא על אוסף קבוצות של ערכים, במקרה כזה נשתמש ב **GROUP BY**.



פעולת GROUP BY מחלקת את הטבלה לקבוצות לפי התכונות שמצוינות בפסוקית זו, כאשר שורות עם אותו הערך בעמודות שמופיעות בפסוק GROUP BY מקובצות לקבוצה אחת.

דוגמא:

הסכום הממוצע בחשבונות הסטודנטים בחלוקה לפי ערים:

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

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

ניתן לציין מספר עמודות בפסק GROUP BY (מופרדות ע"י פסיק), כאשר עבור כל צירוף של ערכי העמודות תיווצר קבוצה.

דוגמא:

רשימת היתרות המקסימליות בחשבונות הסטודנטים שגרים בתל אביב או גרים בירושלים:

```
SELECT City , MAX(Amount) AS MaxAmount
FROM Talmidim
WHERE (City = 'Tel-Aviv') or (City = 'Jerusalem')
GROUP BY City;
```

City	MaxAmount
Tel-Aviv	4000
Jerusalem	1000