

## המשך הדיון בנושא group by והקבצה של שורות עם סיכומים - דיון על תת-שאלות ועוד

דיברנו על פונקציות סיכום הן פונקציות שמחזירות ערך אחד, למשל: count, sum, max, min, avg  
אם למשל נירצה למצוא איזה ספר זמין בכמות הגדולה ביותר בטבלה: Available (יתכנו כמה כאלה)  
ניסיון ראשון:

```
SELECT ISBN, MAX(noOfCopies)
FROM Available
```

מה קורה? למה זה לא עובד? אנחנו מנסים לשלב כאן פונקציה סיכום עם בחירת שורות פשוטה. זה לא ילך.

כדי לפתור את הבעיה, למדנו על אפשרות ל "קבץ" שורות ולסכם עבורן שדה או שדות בעזרת:  
**GROUP BY**

לכן, כדי להשיג את הרצוי במקרה הנ"ל נכתוב:

```
SELECT ISBN, MAX(noOfCopies)
FROM Available GROUP BY ISBN
```

ראינו גם מצב שבו לא נרצה להציג את כל שורות הסיכום, אלא חלק מהן על פי קריטריונים שיתוארו בשאלתה, למשל אם נרצה למצוא מספרי זהות של ספרים שמופיעים במלאי של יותר מספריה אחת (לא משנה באיזו כמות)

מה זה בעצם אומר? אם ספר מוחזק בספריה אחת בלבד, הוא לא יהיה ברשימה (אפילו אם יש ממנו מספר עותקים באותה ספריה).

אם נכתוב למשל

```
SELECT ISBN, COUNT(*)
FROM Available
GROUP BY ISBN
```

זה לא מספיק, כי זה יתן לנו גם מספרי ספרים שנמצאים בספריה בודדת. כדי לסנן את אלה, השתמשנו בפקודת ה- **Having**. זו מקבילה בקירוב לפקודת ה- where בשאלתה ללא סיכומי שורות. כלומר נכתוב:

```
SELECT ISBN, COUNT(*)
FROM Available
GROUP BY ISBN
HAVING COUNT(*) > 1;
```

כאן ה- **having** מתייחס למספר השורות שמופיעות עבור כל ISBN.

כפי שכבר ראינו פקודת ה- **group by** נעשית גם במצב שרוצים לספור שורות שעונות על תנאי ההקבצה **count(\*)** וגם במצב שרוצים לסכם שדה כמותי, כמו למשל: **numOfCopies** בטבלה **available**.

באותה מידה גם הסינון של: **having** יכול לשמש בשני מצבים אלה.

דוגמה:

נניח שרוצים למצוא ספרים (מס זיהוי) שעבורם כמות סך העותקים שנותרה בכל הספריות בהן הם נימצאים, גדול-2. כלומר כאן פונקציה הסיכום (אגרגציה) לא סתם תספור שורות אלא תסכם גם את כמות העותקים בכל השורות בהן הספר מופיע.

לשם כך נכתוב:

```
SELECT ISBN, SUM(noOfCopies)
FROM Available
GROUP BY ISBN
HAVING SUM(noOfCopies) > 2
```

שימו לב שבמקרה זה מפעילים את ה- having על הסכום.

אפשר גם לשלב סינון של שורות בעזרת where ביחד עם: group by. למשל אם נרצה שיופיעו בשאלתה האחרונה רק ספרים בעלי ISBN גדול מ- 5555555555 נכתוב:

```
SELECT ISBN, SUM(noOfCopies)
FROM Available
WHERE ISBN > 5555555555
GROUP BY ISBN
HAVING SUM(noOfCopies) > 2
```

שימו לב שמיקמתי את ה- where לפני ה- group by, אחרת זה לא יעבוד. קודם מסננים את הבחירה הכללית, ורק אחר כך מקבצים לקבוצות.

### תת-שאלות, או שאלות פנימיות או מקוננות (Sub queries)

אלה שאלות שנימצאות בתוך ביטוי ה- WHERE של השאלתה המרכזית (או החיצונית). תת שאלתה מיועדת על מנת להחזיר ערך או ערכים שעל פיהם יכולה השאלתה הראשית לבסס תנאים נוספים לבחירת שורות.

תת-שאלות ניתנות לשימוש במשפטי SELECT, INSERT, UPDATE, DELETE, ויכולות להכיל פעולות כמו =, >, <, <=, >=, IN וכד'.

### כללים לכתיבת תת-שאלתה

היא חייבת להיות מוקפת בין סוגריים רגילים  
לא ניתן להשתמש בפקודה: ORDER BY (אפשר בחיצונית)  
אם תת השאלתה מחזירה יותר משורה אחת, ניתן להתייחס אליה רק עם פעולות מתאימות כגון IN  
ניתן גם לכתוב תת שאלתה על אותה טבלה כמו השאלתה החיצונית.

שאלה - מצא את כל הספרים שכמותם בטבלת הזמינות גבוהה מהכמות הממוצעת של ספר בטבלת הזמינות.

```
SELECT ISBN
FROM Available
WHERE noOfCopies > (
  SELECT AVG(noOfCopies)
  FROM Available)
```

דוגמה נוספת: אם נחזור למה שרצינו בתחילת הדיון כאן: למצוא את הספרים הזמינים בכמות מקסימלית היינו יכולים במקום group by לכתוב:

```
SELECT ISBN
FROM Available
WHERE noOfCopies = (
  SELECT MAX(noOfCopies)
  FROM Available
)
```

בתוך משפט ה- where של השאילתה הראשית 'שתלנו' תת-שאילתה שמחזירה רק ערך אחד.

```
SELECT MAX(noOfCopies)
FROM Available
```

כלומר, אם נסכם אפשר להגיד שתת-השאילתה מחזירה ערך שמאפשר לסנן שורות עבור השאילתה הראשית.

דוגמה נוספת (אגב לעיתים ניתן לבצע join במקום להשתמש בתת-שאילתה)

ישנה טבלת עובדים מהודו כדלקמן:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	35	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00
7	Muffy	24	Indore	10000.00

שלוף את כל הנתונים מהטבלה עבור עובדים שגילם גבוה מממוצע הגילים של אלה שמעל גיל 24. תת-השאילתה תשלוף את הממוצע, והראשית תתיחס אליו:

```
select *
from employee
where age > ( select avg(age)
              from employee
              where age > 24)
```

בהקשר של תת-שאילתות נראה כעת אופרטורים שימושיים נוספים: **(ANY - זה כמו ANY), ALL, ANY**

אלה משמשים ביחד עם סימלי ההשוואות ע"מ לתת תנאי על תת השאילתה במידה והיא מחזירה מספר ערכים. לדוגמה נתונה הטבלה:

EMPNO	ENAME	JOB	SAL	COMM	DEPTNO
7369	SMITH	CLERK	800		20
7499	ALLEN	SALESMAN	1600	300	30
7521	WARD	SALESMAN	1250	500	30
7566	JONES	MANAGER	2975		20
7654	MARTIN	SALESMAN	1250	1400	30
7698	BLAKE	MANAGER	2850		30
7782	CLARK	MANAGER	2450		10
7788	SCOTT	ANALYST	3000		20
7839	KING	PRESIDENT	5000		10
7844	TURNER	SALESMAN	1500		30

שלוף מספר עובד ומשכורת עבור עובדים שמשכורתם גדולה מכל העובדים במחלקה מספר 20.

```
SELECT e1.empno, e1.sal
FROM emp e1
WHERE e1.sal > ALL (SELECT e2.sal
                    FROM emp e2
                    WHERE e2.deptno = 20);
```

טבלת התוצאה תכיל שורה אחת: 7839 5000

### דוגמה לשימוש ב- ANY

שלוף מספרי עובדים ומשכורות עבור עובדים שמשכורתם גבוהה מעובד כלשהו במחלקה 10.

```
SELECT e1.empno, e1.sal
FROM emp e1
WHERE e1.sal > ANY (SELECT e2.sal
                    FROM emp e2
                    WHERE e2.deptno = 10);
```

טבלת התוצאה:

```
7566 2975
7698 2850
7788 3000
7839 5000
```

### עוד מספר מילות מפתח שימושיות

עוד מילת מפתח: **between** שמאפשרת לבחור שורות בטווח מסויים (שני הצדדים כלולים בטווח), למשל:

```
select ISBN from book where isbn between '6666666666' and '9999999999'
```

### שימוש ב- IN

```
SELECT column_name(s)
FROM table_name
WHERE column_name IN (value1,value2,...)
```

שימוש במילת המפתח - **LIKE** - שני תוים בשימוש \_ (קו תחתון) ו- % (קו תחתון מייצג תו בודד כלשהו ו- % מייצג בין 0 לאינספור. למשל, מצא את כל הסטודנטים ששםם מתחיל באותיות: Ma, או מצא את כל הספריות שנימצאות בערים שיש בשם העיר האות y. (LIKE 'Ma%', LIKE '%y%')

יש גם את האפשרות לבחירת תו ממחרות: LIKE '[abcd]%' או c ואח"כ כל דבר. אפשרי גם להוציא תוים מסויימים על ידי: LIKE '[^abc]\_'

### קומבינציה של AND ו- OR

```
select * from CUSTOMER
where LastName='Smith' and (FirstName='John' or FirstName='Smith')
```

עבור שיפור ביצועים

**SELECT TOP number/percent**

```
SELECT TOP 2 * FROM Customers  
SELECT TOP 50 PERCENT * FROM Customers
```

שימוש במילת המפתח **DISTINCT** (ראינו כבר)

```
SELECT DISTINCT column1, column2, .....columnN  
FROM table_name  
WHERE [condition]
```