

## מה ההבדלים בין: INNER JOIN ו- LEFT OUTER, RIGHT OUTER FULL JOIN ?

בעיקרון משתמשים בסוגים אחרים של JOIN (אחרים מ- INNER) כאשר יש מקרים שלא בטוחים באמינות הנתונים או שהנתונים באים ממקורות שונים ורוצים לתאם ביניהם, או שרוצים לראות באותה תוצאה מקרים חיוביים ושליילים גם יחד.

דוגמה לשימוש:

נתונות הטבלאות הבאות:

| Employee |         | Location |                  |
|----------|---------|----------|------------------|
| EmpID    | EmpName | EmpID    | EmpLoc           |
| 13       | Jason   | 13       | San Jose         |
| 8        | Alex    | 8        | Los Angeles      |
| 3        | Ram     | 3        | Pune, India      |
| 17       | Babu    | 17       | Chennai, India   |
| 25       | Johnson | 39       | Bangalore, India |

שימו לב שבטבלה Employee ישנו עובד בשם Johnson שאיננו נמצא בטבלת המקומות וכן שישנו עובד בעל מספר זיהוי 39 שאינו נמצא בטבלת העובדים.

אם נכתוב Join פנימי ( INNER JOIN ) על מספר זיהוי עובד, יושמטו עובדים 25 ו-39. אבל אנחנו מעוניינים להתאים בין שמות העובדים ומיקומם, ורוצים לדעת גם אם עובד לא ידוע מיקומו. הטכניקה תהיה להשתמש במשהו שניקרא: LEFT JOIN או LEFT OUTER JOIN

לדוגמה:

```
select * from employee left outer join location
on employee.empID = location.empID;
```

אפשר גם להשמיט את מילת המפתח: OUTER ולכתוב:

```
select * from employee left join location
on employee.empID = location.empID;
```

התוצאה תהיה:

| Employee.EmpID | Employee.EmpName | Location.EmpID | Location.EmpLoc |
|----------------|------------------|----------------|-----------------|
| 13             | Jason            | 13             | San Jose        |
| 8              | Alex             | 8              | Los Angeles     |
| 3              | Ram              | 3              | Pune, India     |
| 17             | Babu             | 17             | Chennai, India  |
| 25             | Johnson          | NULL           | NULL            |

המילה LEFT מתיחסת לטבלה שמשמאל (או הראשונה), שממנה ישלפו גם שורות שאינן עונות על התנאי של ה- join.

דבר דומה יקרה בכיוון השני אם ניכתוב:

```
select * from employee right outer join location
on employee.empID = location.empID;
```

| Employee.EmpID | Employee.EmpName | Location.EmpID | Location.EmpLoc  |
|----------------|------------------|----------------|------------------|
| 13             | Jason            | 13             | San Jose         |
| 8              | Alex             | 8              | Los Angeles      |
| 3              | Ram              | 3              | Pune, India      |
| 17             | Babu             | 17             | Chennai, India   |
| NULL           | NULL             | 39             | Bangalore, India |

כלומר מצאנו עובד שממוקם ב- Bangalore, India עובד מספר 39 שאיננו בטבלת העובדים (נתונים לא אמינים). פעם טבלה שמימין (location) היא זו שמספקת שורות שגם מקיימות את תנאי ה- join וגם לא.

תנחשו מה יקרה אם ניכתוב:

```
select * from employee full outer join location
on employee.empID = location.empID;
```

ישנו מצב שבו רוצים אינפורמציה מורכבת מטבלה, הדורשת חיבור טבלה לעצמה בעזרת alias.

### JOIN של טבלה על עצמה

נתונה הטבלה:

employee

| emp_id | emp_name | manager_id |
|--------|----------|------------|
| 1      | Hattie   | 4          |
| 2      | Henry    | 4          |
| 3      | Harry    | 5          |
| 4      | Helen    | NULL       |
| 5      | Heidi    | 4          |
| 6      | Hazel    | 1          |

רוצים לקבל מידע על מי מדווח למי, כלומר לכל עובד או עובדת לתת את המנהל שלו או שלה. זהו מצב שרוצים לקבל שני סטים של אינפורמציה מטבלה אחת.

```
select e.emp_name as employee,
       m.emp_name as manager
from   employee e, employee m
where  e.manager_id = m.emp_id
```

employee manager

---

|        |        |
|--------|--------|
| Hattie | Helen  |
| Henry  | Helen  |
| Harry  | Heidi  |
| Heidi  | Helen  |
| Hazel  | Hattie |

שמתם לב שחסרה שורה?

עשינו inner join, ו-Helen היא בעלת החברה כך שאין לה מנהל. איך נמצא עבור מי עובדת Helen ?

אם נחזור לבסיס הנתונים של הספרים, איך אפשר בשאילתה אחת לראות את שמות הסטודנטים ששאלו לפחות 2 ספרים וגם את אלה שלא?

תרגיל כתיבה

לכתוב שאילתה שנותנת את צמדי הסטודנטים ששאלו אותו ספר, כאשר ה-ID של אחד קטן מה-ID של השני

הרעיון היה לבצע JOIN של טבלה על עצמה, במקרה זה טבלת ה- 'שואלים' (borrow) שבה נימצאים גם מספרי הזהות של הספרים (isbn) וגם מספרי הזהות של הסטודנטים השואלים ספרים (ID)

```
select distinct b1.id as id1, b2.id as id2
from borrow b1, borrow b2
where b1.isbn = b2.isbn
and b1.id < b2.id order by 1, 2;
```

כדי לעקוב אחרי דרך הפעולה של השאילתה, ניצור לעצמנו 2 עותקים זהים של הטבלה: borrow וניצור alias אחד בשם: b1 והשני בשם: b2 ונעבור לכל שורה של עותק אחד, על כל השורות של העותק השני לדוגמה:

| ID  | LNAME     | ISBN       | ID  | LNAME     | ISBN       |
|-----|-----------|------------|-----|-----------|------------|
| 040 | Courant   | 0824774566 | 040 | Courant   | 0824774566 |
| 099 | Central   | 9889950316 | 099 | Central   | 9889950316 |
| 099 | Cooper    | 0824774566 | 099 | Cooper    | 0824774566 |
| 099 | Law       | 1596912839 | 099 | Law       | 1596912839 |
| 099 | Main      | 0140367152 | 099 | Main      | 0140367152 |
| 099 | Microsoft | 052188473X | 099 | Microsoft | 052188473X |
| 295 | Cooper    | 0824774566 | 295 | Cooper    | 0824774566 |
| 295 | Cooper    | 9621747082 | 295 | Cooper    | 9621747082 |
| 295 | Courant   | 1586039474 | 295 | Courant   | 1586039474 |
| 295 | Law       | 1596912839 | 295 | Law       | 1596912839 |
| 295 | Star      | 0312425074 | 295 | Star      | 0312425074 |

ניקח למשל שורה ראשונה מהטבלה משמאל מול שורה ראשונה מהטבלה מימין, אמנם ה- isbn שווה אבל ה- ID זהה (040), כך שהן לא מתאימות ולכן עוברים לשורה הבאה בטבלה מימין, שורה שנייה גם לא מתאימה ה- ID שלה (099) אמנם גדול מ- 040, אבל ה- isbn שונה. השורה השלישית בטבלה מימין כן מתאימה.

ה- isbn הוא : 0824774566 וזה זהה לשורה הראשונה של הטבלה משמאל וגם 040 קטן מ- 099, ולכן בתוצאת השאילתה כבר תהיה השורה: 099 040 גם השורה השביעית בטבלה שמימין מתאימה, גם אותו isbn וגם ה- ID הוא 295 גדול מ- 040 של השורה הראשונה בטבלה משמאל, כך שהשורה הבאה בתוצאה תהיה: 295 040 וכך הלאה.