

## מבוא ל - Operator overloading ב-C# ודיון על שאילתות SQL

הקמנו את בסיס הנתונים של 5 הטבלאות לנהל סטודנטים שמשאילים ספרים מספריות והכנסנו לתוכן נתונים.

התחלנו בדיון על שאילתות פשוטות עם פונקציות אגרגציה כמו: count(\*) ו-sum(column) והמשכנו לדון בשאילתות קצת יותר מורכבות.

ראינו שאילתה שכוללת שילוב של שתי טבלאות. ע"מ למצוא את מספרי הזיהוי של הספרים שמצד אחד הושאלו על ידי מישהו, כלומר מופיעים בטבלה: Borrow, ומצד שני עדיין קיימים עותקים שלהם זמינים באחת או יותר מהספריות.

```
select ISBN
from Available a, Borrow b
where a.ISBN = b.ISBN
and a.noOfCopies > 0
```

זה בעצם אומר תן לי את הספרים (מספר זיהוי של) שמופיעים בשתי הטבלאות ועבור כל צירוף כזה בדוק אם מספר העותקים שנוותר עבור הספר בטבלה Available הוא גדול מאפס (כלומר יש עותק פנוי).

זוהי פעולה שניקראת Join, של שתי טבלאות על השדה ISBN שבמקרה זה הוא חלק מהמפתח בשתי הטבלאות (אבל לא חייב להיות).

ראינו כאן עוד אספקט שבו כענענו לשם קיצור את הטבלאות a ו-b, זה ניקרא Alias (או שם נירדף), כדי לחסוך בכתיבת שם הטבלה על מנת לזהות שדות במדויק. יש לציין שאם השדה מופיע רק בטבלה אחת (כמו במקרה של noOfCopies) אפשר להשמיט את שם אוכינוי הטבלה ורק לכתוב:

```
and noOfCopies > 0
```

בפעולת Join אנחנו מקבלים מכפלה קרטזית של שורות, כלומר כל השורות התואמות בין הטבלאות יופיעו בתוצאה. לדוגמה בטבלה: Available הספר בעל ISBN של: 6666666666 זמין גם בספריה Main וגם בספריה Spy (עותק אחד בכ"א). אם הוא היה מופיע בטבלה: Borrow אפילו רק מספריה אחת מהן, בתוצאה של השאילתה הנ"ל הוא היה מופיע פעמיים.

בשיעור הבא נילמד איך להתגבר על תופעה כזאת.

התחביר של שאילתת ה-Join הנ"ל אינו היחיד האפשרי, כפי שנראה מנוע ה-SQL ישכתב אותו בצורה קצת שונה.

פעולת ה-Join היא פעולה מאד בסיסית ב-SQL ולמעשה מדגימה את העוצמה שיש בבסיס נתונים יחסי להרכיב צירוף נתונים רצוי על ידי שילוב של מספר טבלאות (אין הגבלה מעשית למספר הטבלאות ב-join)

תחשבו איך לשנות את השאילתה הנ"ל ע"מ להציג את שם הספר במקום את מספר הזיהוי שלו.

1. לכתוב שאילתה (Query) שתתן לנו את כל שמות הספרים שאינם נימצאים במלאי של אף ספרייה. כלומר אינם מופיעים בכלל (לא רק עם מספר עותקים  $< 0$ )

2. לכתוב שאילתה (Query) שתתן את רשימת שמות כל הסטודנטים ששאלו ספר של הסופר: Sherman Chow או Bernard Cushing של הסופר.

3. לכתוב את הפונקציות של חיבור תאריכים עבור המחלקה date כ- Operator, והשוואה בין תאריכים ( $==$ ), כך שנוכל למשל לכתוב:

```
date d1 = new date (16, 9, 2014);
d1.print();
date d2 = new date (16, 9, 2015);
d2.print();
date d3 = d1 + d2;
d3.print();
if (d1 == d2)
    Console.WriteLine ("d1 and d2 are equal");
else
    Console.WriteLine ("d1 and d2 are NOT equal");
```

אגב, הועלה רעיון בכיתה לכתוב את פונקצית קידום התאריך ביום ללא החזר של עצם מסוג date, אבל זה לא אפשרי בפונקציות של operator - הן אינן יכולות להיות בעלות החזר void. מהבחינה הזאת שפת ++C בעלת גמישות גדולה יותר.