

מעבדה מס' 6

פתרונות לתרגיל מהשיעור הקודם

נתבקשנו למצוא:

- רשימת הקורסים שלקחה דנה
- רשימת הסטודנטים שנרשמו ל-c++.
- רשימת התלמידים במחלקה 5
- רשימת הקורסים של מחלקה 5.

והפתרונות:

```
SELECT cname FROM Courses, Students, Student_in_Course
WHERE sname='Dana'
AND snumber=ssnumber
AND cnumber=ccnumber;
```

```
SELECT sname FROM Courses, Students, Student_in_Course
WHERE cname='c++'
AND snumber=ssnumber
AND cnumber=ccnumber;
```

```
SELECT sname FROM Students WHERE Department='5';
```

```
SELECT cname FROM Courses WHERE Department='5';
```

דרכים להגדרת מפתח זר

נראה 4 דרכים (ישנן יותר) להגדרת מפתח זר (בהנחה ש-Table1 מוגדרת כנדרש):

```
CREATE TABLE Table2 (j INTEGER REFERENCES Table1(i));
```

```
CREATE TABLE Table2 (j INTEGER,
                      FOREIGN KEY (j) REFERENCES Table1 (i));
```

```
CREATE TABLE Table2 (j INTEGER,
                      CONSTRAINT cc FOREIGN KEY (j) REFERENCES Table1 (i));
```

```
CREATE TABLE Table2 (j INTEGER);
ALTER TABLE Table2 ADD CONSTRAINT cc FOREIGN KEY (j) REFERENCES Table1(i);
```

נשים לב כי הדרך הראשונה היא הפשוטה ביותר אך אינה מאפשרת מפתח של יותר משדה אחד. הדרך השנייה כן מאפשרת זאת. הדרך השלישית מאפשרת מפתח של יותר משדה אחד ונותנת שם לאילוץ המפתח הזר, על מנת שנוכל לבטלו בהמשך. הדרך הרביעית מאפשרת הגדרת המפתח הזר לאחר הגדרת הטבלה, וכמובן מאפשרת מפתח של יותר משדה אחד.

תת שאילתות

הדוגמה היחידה שראינו של תת שאילתות, היתה תת שאילתה להכנסת שורות לטבלה. נדון בנושא זה יותר בפירוט.

תת שאילתה המחזירה סקלר

אם לדוגמה, ננסה לקבל בצורה שונה את רשימת הקורסים שלקחה דנה, נוכל לכתוב:

```
SELECT cname FROM Courses, Student_in_Course WHERE
ssnumber = (SELECT snumber FROM Students WHERE sname='Dana') AND
ccnumber=cnumber;
```

הערה: אנחנו מניחים כי השאילתה הפנימית `SELECT snumber FROM Students WHERE sname='Dana'` תחזיר ערך בודד. **אחרת תתרחש שגיאה.** לכן, השאילתה שכתבנו תעבוד רק אם יש רק סטודנטית אחת ששמה דנה.

שאילתות המחזירות מספר שורות

נחזור ל-Operations.txt, שם הגדרנו:

Table1				Table2		
Num1	Num2	Num3		Num1	Num2	Num3
1	4	3		1	3	5
1	1	5		1	4	3
1	10	11		1	10	12

כעת נכתוב מספר שאילתות:

1. `SELECT * FROM Table1 WHERE Num3 > ALL (SELECT Num2 FROM Table1);`
2. `SELECT * FROM Table1 WHERE Num2 > ANY (SELECT Num3 FROM Table1);`
3. `SELECT * FROM Table1 WHERE Num3 IN (SELECT Num3 FROM Table2 WHERE Num3<4);`
4. `SELECT * FROM Table1 WHERE NUM2<10 AND EXISTS (SELECT * FROM Table2 WHERE Num3>=12);`
5. `SELECT * FROM Table1 WHERE NUM2<10 AND EXISTS (SELECT * FROM Table2 WHERE Num3>=13);`
6. `SELECT * FROM Table1 WHERE NUM2<10 AND NOT EXISTS (SELECT * FROM Table2 WHERE Num3>=12);`
7. `SELECT * FROM Table1 WHERE NUM2<10 AND NOT EXISTS (SELECT * FROM Table2 WHERE Num3>=13);`
8. `SELECT * FROM Table1 WHERE NUM2>= ALL (SELECT Num2 FROM Table1);`
9. `SELECT * FROM Table1 WHERE NUM2<= ALL (SELECT Num2 FROM Table1);`
10. `SELECT * FROM Table1 WHERE Num2 <> ALL (SELECT Num2 From Table2);`
11. `SELECT * FROM Table1 WHERE Num2 NOT IN (SELECT Num2 From Table2);`
12. `SELECT * FROM Table1 WHERE Num3 = ANY (SELECT Num3 FROM Table2);`
13. `SELECT * FROM Table1 WHERE Num3 IN (SELECT Num3 FROM Table2);`
14. `SELECT Num3 FROM Table1 WHERE (Num1, Num2) IN (SELECT Num1, Num2 FROM Table2);`

נסכם. האופרטורים שניתן להשתמש בהם:

1. EXISTS R – זהו תנאי עם ערך אמת אם R אינו ריק (שאילתות 4-7).
2. s IN R – נכון רק אם s שווה לאחד מערכי R. R יכול להיות אונרי (דוגמאות 3,11,13) או רב-שדה (דוגמה 14).
3. s>ALL R – נכון רק אם s גדול מכל ערכי R. ניתן להשתמש בכל אופרטור השוואה אחר. R **חייב** להיות אונרי. (דוגמאות 1,8,9,10).
4. s>ANY R – נכון רק אם s גדול לפחות מערך אחד של R. ניתן להשתמש בכל אופרטור השוואה אחר. R **חייב** להיות אונרי. (דוגמאות 2,12).

נראה דוגמה מוחשית לעבודה עם תת-שאילתות: יהיו ה-Relations הבאים:

Movie (title, year, length, inColor, studioName, producer#)

StarsIn (movieTitle, movieYear, starName)

MovieExec(name, address, cert#, netWroth)

```

SELECT name FROM MovieExec
WHERE cert# IN
  (SELECT producer# FROM Movie
   WHERE (title, year) IN
     (SELECT movieTitle, movieYear FROM StarsIn
      WHERE starName='Harrison Ford'));

```

נקבל את כל המפיקים של סרטים בהם הריסון פורד שיחק.
אגב מה יהיה פלט השאילתה הבאה?

```

SELECT name FROM MovieExec, Movie, StarsIn
WHERE cert#=producer# AND title=movieTitle and year=movieYear AND starName='Harrison Ford';

```

תת-שאילתות מתואמות (Correlated Subqueries)

לעתים אנו רוצים שתת השאלתה תחושב הרבה פעמים, פעם אחת לכל השמה של ערך לביטוי בתת השאילתה – ערך המגיעה מחוץ לתת השאלתה. תת שאלתה כזו נקראת מתואמת.

נוסיף לטבלה Table2 את שתי שורות:

```

INSERT INTO Table2 VALUES (2,4,7);
INSERT INTO Table2 VALUES (5,3,17);

```

כעת נריץ את השאילתה הבאה:

```

SELECT Num2 FROM Table2 Dup WHERE Num1 < ANY (SELECT Num1 FROM Table2 WHERE
Num2=Dup.Num2);

```

מה קיבלנו? את כל הערכים של Num2 שמופיעים יותר מפעם אחת.

לבית: עמ' 269 – 5.3.1, 5.3.2. מומלץ מאוד