

## מעבדה מס' 5

### איחוד, חיתוך והפרש של שאלות

מחקו את טבלאות Table1 ו-Table2. לאחר מכן הריצו את הסקריפט Operations.txt מהמעבדה הקודמת. כדי לבצע הפרש בין שאלות, נכתוב:

(SELECT Num1, Num3 FROM Table1 WHERE Num3>4) MINUS (SELECT Num1, Num3 FROM Table2);

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

### יצירת VIEWS (מבטים)

מבט הוא למעשה relation שאינו קיים פיזית בבסיס הנתונים. הוא מוגדר בדומה לשאלתה. אך מצד שני, ניתן לתשאל מבטים כאילו הם קיימים פיזית.

```
CREATE VIEW FirstView AS  
SELECT Num1, Num3 FROM Table1 WHERE Num2>3;
```

נתשאל את המבט החדש שיצרנו:

```
SELECT Num3 FROM FirstView WHERE Num3>9 AND Num3<20;
```

נשים לב כי רק Num1 ו-Num3 יכולים להשתתף בתשאול של המבט.

בדוגמה מסובכת יותר, ניתן לכתוב:

```
CREATE VIEW SecondView AS  
SELECT EmpNo, Dept.DeptNo, DName, Loc FROM Emp, Dept WHERE  
Emp.DeptNo=Dept.DeptNo;
```

ונתשאל:

```
SELECT * FROM SecondView WHERE AND EmpNo>=7900;
```

נשים לב, כי הפלט זהה למה שקיבלנו בשאלתה מהמעבדה הקודמת:

```
SELECT EmpNo, Dept.DeptNo, DName, Loc FROM Emp, Dept WHERE Emp.DeptNo=Dept.DeptNo  
AND EmpNo>=7900;
```

כדי למחוק מבט נכתוב (נשים לב כי הטבלאות הרלוונטיות למבט לא השתנו):

```
DROP VIEW FirstView;
```

### הוספת מספר שורות כפלט שאלתה לטבלה כלשהי

בצעו את השאלתה הבאה:

```
SELECT Num3, Num2, Num1 FROM TABLE2;
```

וכעת נוסיף פלט זה לטבלת Table1:

```
INSERT INTO Table1  
SELECT Num3, Num2, Num1 FROM TABLE2;
```

הוספנו 3 שורות לטבלה Table1.

הערה: בדף העבודה פעולה זו נקראת **insert a subquery**.

## הגדרת מפתח זר

```
CREATE TABLE Students (sname VARCHAR(20) NOT NULL,  
                        snumber INTEGER PRIMARY KEY,  
                        Department VARCHAR(20) NOT NULL);
```

```
CREATE TABLE Courses (cname VARCHAR(20) NOT NULL,  
                      cnumber INTEGER PRIMARY KEY,  
                      Department VARCHAR(20));
```

```
CREATE TABLE Student_in_Course (ssnumber INTEGER,  
                                 ccnumber INTEGER,  
                                 PRIMARY KEY (ssnumber, ccnumber),  
                                 FOREIGN KEY (ssnumber) REFERENCES Students (snumber),  
                                 FOREIGN KEY (ccnumber) REFERENCES Courses (cnumber));
```

זהו קשר מסוג רבים לרבים. ה-attribute ששמו ssnumber הוא מפתח זר, שמתייחס לשדה snumber של טבלת Students. כנ"ל לגבי ה-attribute ששמו ccnumber. באופן כללי, 2 דברים צריכים להתקיים:

(א) המפתח הזר מתייחס לשדה/ות בטבלה אחרת (או באותה טבלה במקרים מיוחדים). השדה/ות האלו חייבים להיות מפתח ראשי בטבלה שבה נמצאים

(ב) לא ניתן להוסיף ערכים למפתח הזר, שאינם מופיעים בשדה/ות בטבלה אליה מתייחס המפתח. אך ניתן לשים ערך NULL בשדה המפתח הזר, ללא צורך בערך NULL בשדה/ות בטבלה אליה מתייחס המפתח (לא יכול להיות שם NULL, הרי זה מפתח ראשי). אגב, נוכל לאלץ את המפתח הזר להיות NOT NULL (כפי שקורה לעתים ב-E/R).

מלאו את שני הטבלאות בארבע שורות לפחות לכל אחת, לדוגמה:

```
INSERT INTO Students VALUES ('Moshe Cohen', 5, 'SE');  
INSERT INTO Courses VALUES ('Databases', 111, 'SE');
```

כעת נמלא את הטבלה השלישית. יש לשים בכל שורה מס' סטודנט ומס' קורס על מנת לציין שהסטודנט רשום לקורס. לדוגמה:

```
INSERT INTO Student_in_Course VALUES (5, 111);
```

הכניסו לטבלה זו כשבע שורות.

**הערה:** לא ניתן להוסיף שורה בטבלת Student\_in\_Course הכוללת מס' סטודנט שאינו קיים בטבלת הסטודנטים ו/או מס' קורס שאינו קיים בטבלת הקורסים.

נבצע את השאילתה הבאה:

```
SELECT Stdudents.sname, Students.snumber FROM Students, Student_in_Course  
WHERE Students.snumber=Student_in_Course.ssnumber AND Student_in_Course.ccnumber=111;
```

השאילתה תחזיר את שמות ומספרי הסטודנטים הרשומים לקורס שמספרו 111 (בסיסי נתונים). באופן דומה ניתן לכתוב:

```
SELECT Students.sname, Students.snumber FROM Students, Courses, Student_in_Course  
WHERE cname='Databases'  
AND cnumber=ccnumber  
AND Students.snumber=Student_in_Course.ssnumber  
AND snumber>4;
```

אז נקבל את שמות ומספרי הסטודנטים הרשומים לקורס בסיסי נתונים ושמספרם גדול מ-4.

## תרגיל לבית:

בנה שאילתות למציאת:

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