

### מעבדה מס' 3

#### המשך שינויים בטבלה

מחיקת שורות מטבלה :

אנא מחקו את טבלת Account ו-Customer2 והריצו את הסקריפט insert.txt מהמעבדה הקודמת.

```
DELETE FROM Account WHERE AccNo=100;  
DELETE FROM Customer2 WHERE Name='Moshe Choen' OR SSNo=22233;
```

אגב, מעצם ההגדרה, הפקודה :

```
DELETE FROM Emp;
```

תמחק את כל רשומות הטבלה (נא לא לנסות פקודה זו).

עדכון שורות בטבלה :

אנא מחקו את טבלת Account ו-Customer2 והריצו את הסקריפט insert.txt מהמעבדה הקודמת.

```
UPDATE Account SET Balance=Balance+100 WHERE Balance>0; // sets balans = balans+100 ,  
where .....
```

```
UPDATE Customer2 SET Name='Mr. ' || Name, Phone='972' || phone; || - siman shirshur le Name  
akayam , klomar ba tavla ihie: Mr. (Name kayam) , 972 (phone kayam)
```

#### אפשרויות נוספות בשאילתה

```
SELECT AccNo AS Account_Code, Balance+200 AS Bonused_Balance FROM Account  
WHERE AccNo>=100; // only for this select Balanc+200 – no changes in table
```

```
SELECT AccNo, Balance, 'Shekels' AS in_Shekels, Type FROM Account; // fictive column  
// shekels – const for all rows
```

#### השוואת מחרוזות

כפי שראינו, ניתן לבצע השוואת מחרוזות לקסיקוגרפית. לדוגמה :

```
SELECT * FROM Emp WHERE Job>'CLERK';
```

אגב, כפי ששמנו לב, על אף ששדה מסוג DATE או TIME מיוצג כמחרוזת, ההשוואה מתבצעת בצורה הגיונית.

ניתן גם לבצע השוואת מחרוזות בעזרת הסימנים % ו- (קו תחתון), כאשר % מייצג 0 תווים או יותר ו- \_ מייצג תו בודד אחד. דוגמאות :

```
SELECT * FROM Customer2 WHERE Name LIKE 'Moshe _____' רווח ו-5 קווים תחתונים אחרי משה  
// LIKE - just for comparing strings , kav tahton – like asteriks  
SELECT * FROM Customer2 WHERE Name LIKE '%Azrieli';  
SELECT * FROM Customer2 WHERE Name LIKE '%"%';
```

כדי לייצג את הסימן % (או \_ ) נכתוב :

```
SELECT * FROM Customer2 WHERE Name LIKE 'x%x%' ESCAPE 'x';
```

נקבל את כל השמות שמתחילים ומסתיימים ב-%.

```
SELECT * FROM Account WHERE AccNo<=100 ORDER BY Balance;  
SELECT * FROM Account WHERE AccNo<=100 ORDER BY Balance, Type;  
SELECT * FROM Account WHERE AccNo<=100 ORDER BY Balance DESC;
```

**דרכים שונות להגדרת מפתח ראשי**

נחזור לטבלת חשבונות בנק מהמעבדה הקודמת ונראה 4 דרכים (ישנן יותר) להגדרת מפתח ראשי:

```
CREATE TABLE Account (Accno INTEGER PRIMARY KEY NOT NULL,  
                        Balance FLOAT,  
                        Type CHAR(10));
```

```
CREATE TABLE Account (Accno INTEGER NOT NULL,  
                        Balance FLOAT,  
                        Type CHAR(10),  
                        PRIMARY KEY (Accno));
```

```
CREATE TABLE Account (Accno INTEGER NOT NULL,  
                        Balance FLOAT,  
                        Type CHAR(10),  
                        CONSTRAINT cc_num PRIMARY KEY (Accno));
```

```
CREATE TABLE Account (Accno INTEGER NOT NULL,  
                        Balance FLOAT,  
                        Type CHAR(10));
```

```
ALTER TABLE Account ADD PRIMARY KEY (Accno);
```

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 הראשי, על מנת שנוכל לבטלו בהמשך. הדרך הרביעית מאפשרת הגדרת המפתח הראשי לאחר הגדרת הטבלה, וכמובן מאפשרת מפתח של יותר משדה אחד.