



הרצאה מספר 11

נושאים לשיעור זה:

שאלת רענון בצירוף טבלאות ✓

פקודות DDL: ✓

○ מפתח ראשי (Primary Key)

○ מפתח זר (חיצוני – Foreign Key)

סיכום הפקודות עד לשיעור זה

```
SELECT id, count(id) + (S-Q)
FROM Students AS S
LEFT OUTER JOIN Course AS C, (S-Q) AS SQ
ON S.studentID = C.studentID
WHERE (id=3) AND (name is not NULL)
AND X = (S-Q)
GROUP BY id
HAVING count(id) > 1 / count(id) = (S-Q)
ORDER BY id, name
```

*** (S-Q) = מקומות בהם ניתן לשלב תתי שאילתות



שאלת רענון בצירוף טבלאות:

נתונות 2 טבלאות, מה יהיה הפלט ע"י הרצת 2 השאילתות הנ"ל ?

t1		t2	
a	b	b	c
1	a	b	3
2	b	c	4

SELECT
FROM
ON

*
t1 LEFT JOIN t2
(t1.b = t2.b);



a	b	b	c
1	a	Null	Null
2	b	b	3

לעומת זאת אם היינו מבצעים את הפקודה הבאה:

SELECT
FROM
ON
WHERE

*
t1 LEFT JOIN t2
(TRUE)
(t1.b = t2.b);

a	b	b	c
2	b	b	3

מתבצע צירוף חיצוני, כאשר כל השורות עומדות בתנאי הצירוף (כולם מקיימים את TRUE) ז"א שנקבל בעצם מכפלה קרטזית של הטבלאות (4 שורות). כעת, מכיוון שאף שורה לא נפלה, אין צורך בביצוע ריפוד עם ערכי NULL.

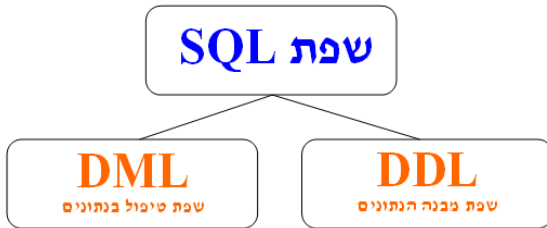
a	b	b	c
1	a	b	3
1	a	c	4
2	b	b	3
2	b	c	4

מסקנה: יש משמעות למיקום התנאי, נוכל לראות שהתוצאות שונות אם התנאי מופיע

בפקודת ה ON או בפקודת ה USING לעומת התוצאה שתתקבל אם יופיע בתנאי ה WHERE.



פקודות DML:



בשיעורים האחרונים עסקנו בפקודות DML הנוגעות בנתונים וראינו שיש פקודות העוסקות בשליפה, הוספה, עדכון ומחיקה של נתונים/רשומות בטבלה. השיעור נתחיל ללמוד את החלק השני בשפת ה SQL – פקודות DDL העוסקות במבנה.

פקודות DDL:

1) מפתח ראשי (Primary Key)

- **מפתח ראשי** – שדה (או אוסף שדות) המזהה באופן חד ערכי (ומינימאלי) רשומה מסוימת בטבלה.

דוגמא: מפתח ראשי הבנוי משדה בודד - מספר תעודות זהות בטבלת סטודנטים.

Students		
StudentID	FirstName	LastName
111	Avi	Cohen
222	Dan	Israeli
333	Ofer	Bar

Primary Key

דוגמא: מפתח ראשי הבנוי מאוסף שדות – נבנה טבלה המציגה רשימת ספקים ורשימת מוצרים,

כאשר נתון:

- שמות הספקים יהיו: אסם, שטראוס ותנובה
- המוצרים המיוצרים על ספקים אלו הם: קוטג', קטשופ וגבינה לבנה

Vendor	Product	Price	Weight (gr)
Osem	Ketchup	10.00	1000
Strauss	Cottage	6.20	250
Tnuva	Cottage	6.30	250
Strauss	Cheese	5.80	200
Tnuva	Cheese	5.70	200



מי יהיה שדה המפתח בטבלה ?

נוכל לראות שלא ניתן להגדיר את הספק כמפתח כי יש ספקים החוזרים על עצמם, על אותו משקל נראה שלא ניתן להגדיר את המוצר כמפתח כי יש מוצרים בעלי אותו שם לספקים שונים.

← לכן, הפתרון הוא להגדיר מפתח משולב (מפתח המורכב ממספר שדות) המכיל את השדות ספק+מוצר. שילוב של שדות אלו מבטיח בצורה חד ערכית שהשילוב של ספק+מוצר הינו ייחודי ומינימאלי עבור כל שורה בטבלה.

Primary Key

Vendor	Product	Price	Weight (gr)
Osem	Ketchup	10.00	1000
Strauss	Cottage	6.20	250
Tnuva	Cottage	6.30	250
Strauss	Cheese	5.80	200
Tnuva	Cheese	5.70	200

הערות

- **כפילות ערכים בשדה מפתח** (בהמשך לדוגמא הקודמת) - אם מפתח ראשי מורכב מיותר משדה אחד (כמו בדוגמא לעיל), אזי ערכי **אחד השדות** יכולים להיות כפולים בתוך עמודה נתונה (לדוגמא: בעמודת ספק נוכל לראות שתנובה מופיעה מספר פעמים) אבל השילוב של כל שדות המפתח הראשי חייבים להיות ייחודיים ברמת טבלה.

הגדרת מפתח ראשי

- **הגדרת מפתח ראשי** - אנו נגדיר את המפתח הראשי בטבלה בשלב של יצירת הטבלה ע"י הוספת הפקודה Primary Key. והשדות שמרכיבים את המפתח.

*** הערה, בעקרון בטבלה יוצג מספרי הספק וקוד המוצר ולא השמות שלהם, אך בדוגמא זו על מנת להבהיר את הנקודה הוצגו הנתונים בשמותיהם ולא בקודים המסמלים אותם. ***

```
CREATE TABLE Example
(
  Vendor TEXT NOT NULL,
  Product TEXT NOT NULL,
  Price FLOAT NOT NULL,
  Weight FLOAT,
  PRIMARY KEY (Vendor, Product)
);
```

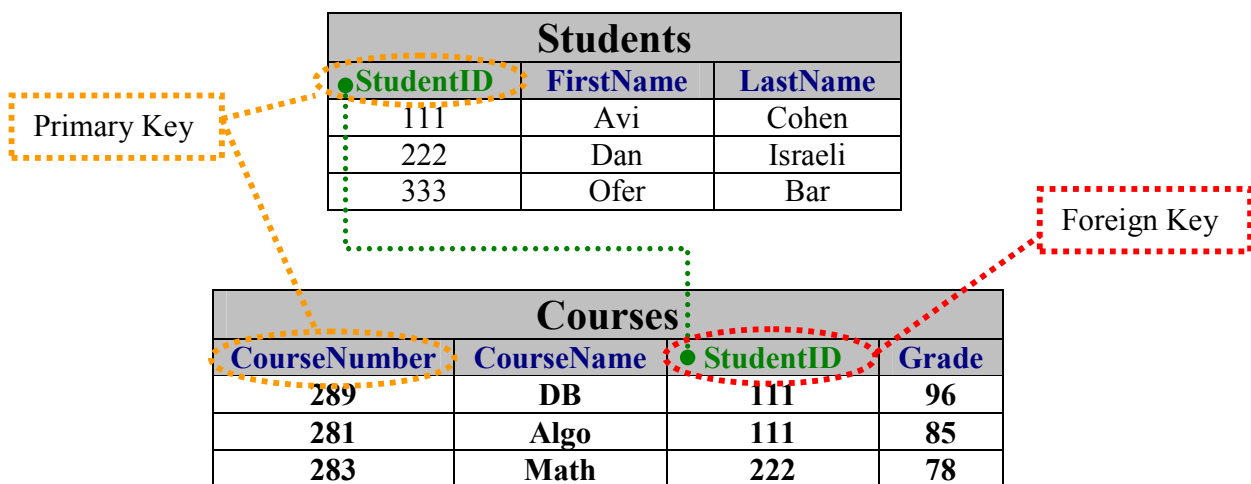
- **כמות מפתחות** - כל טבלה יכולה שיהיה לה **רק מפתח ראשי אחד** (שיכול להיות מורכב ממספר שדות) והוא אינו יכול להכיל ערכי NULL מכיוון ששדות המפתח חייבים להכיל ערכים ייחודיים.



(2) מפתח זר (חיצוני – Foreign Key)

- **מפתח זר** – שדה (או אוסף שדות) המצביע על מפתח ראשי בטבלה אחרת.
- מפתח זר הינו שדה (עמודה) בטבלה אשר ערכיו האפשריים (Domain) נשאבים משדה מקביל בטבלה אחרת אשר בה הוא משמש כמפתח ראשי.
- ערכי המפתח הזר אינם בלעדיים (Not Unique) בטבלת הבן ויכולים לחזור על עצמם (לדוגמא שדה מספר סטודנט).

דוגמא: בטבלת קורסים, ת.ז. של הסטודנט הינה מפתח זר לטבלת הסטודנטים (כיוון ששדה זה מהווה מפתח ראשי בטבלת סטודנטים).



- נוכל לראות שבהפניה של מפתח זר, נוצר קישור (לינק) בין 2 טבלאות (בירוק) כאשר בצד אחד מופיע שדה המהווה מפתח ראשי בטבלה אחת (מוקף בכתום) – שדה זה הופך להיות מפתח זר בטבלת השנייה (מוקף באדום).



הערות

- מטרה נוספת של מפתח זר היא שמירה על שלמות הנתונים [Data integrity]

ניתן להגדיר קשר זה כאילוץ (Constraint) בהגדרת הטבלה, כך שמערכת ניהול בסיס הנתונים (DBMS) תמנע מצב שיוכנסו ערכים בשדה המפתח הזר שאינם עונים לאילוץ, כלומר אינם נמצאים בטבלת האב. כתוצאה מהגדרת אילוץ כזה, לא נוכל למחוק רשומה מטבלת האב כל עוד ישנן רשומות בטבלת הבן המכילות את מפתח הרשומה כמפתח זר.

על ידי שימוש במפתח זר מגדירים, ולמעשה יוצרים, את הקשרים בין הטבלאות השונות בבסיס הנתונים.

לדוגמא: נניח שהיינו מוחקים את הסטודנט שמספרו 222 מתוך טבלת סטודנטים, מחיקה מסוג זה תיצור בעיה לאור העובדה שקיימת רשומה לסטודנט מספר 222 גם בטבלת קורסים; בשלב המחיקה מטבלת הסטודנטים המפתח הזר יגרור את הופעתה של הודעת שגיאה המונעת את מחיקת שורה זו מטבלת סטודנטים אלא אם קודם לכן נמחק את הקורס אליו רשום סטודנט 222 (שזו כמובן בקשה הגיונית לאור העובדה שאנו מתכוונים למחוק את הסטודנט מהמאגר).