# JOIN Basic

### לפני שנתחיל בשיעור ניזכר מהו מפתח ראשי ומהו מפתח משני:

מפתח ראשי – שדה בטבלה המחויב להיות בעל ערך ייחודי לכל רשומה.

מפתח משני – שדה בטבלה המשמש כמפתח ראשי בטבלה אחרת.

(יכולים לחזור ולהסתכל על מפתחות ראשיים ומשניים ב – ERD ב – DB של ליאת).

### נתונות שתי הטבלאות הבאות:

טבלת המקצועות:

|  |  |
| --- | --- |
| **תיאור המקצוע** | **קוד המקצוע** |
| תוכניתן | 1800 |
| מפעיל | 1802 |

טבלת החיילים:

|  |  |  |
| --- | --- | --- |
| **שם החייל** | **קוד המקצוע** | **מספר חייל** |
| דורון | 1800 | 001 |
| אביבית | 1802 | 002 |
| לירן | 1802 | 003 |
| גיא | 1800 | 004 |

נרצה לשלוף עבור כל חייל את תיאור המקצוע שלו. איך נעשה את זה?

הדרך הראשונה שתעלה לנו לראש וכולנו מכירים אותה היא באופן ידני. נוכל להסתכל עבור כל חייל מה קוד המקצוע שלו ואז להתאים את המקצוע. אבל מה היינו עושים אם היו 100000 חיילים?

כמה זמן זה היה לוקח לנו?

הדרך השנייה היא בעזרת **JOIN!**

**שימו לב** שהדרך היחידה שבא נצליח לקשר בין כל חייל לבין מקצועו היא בעזרת קוד המקצוע. קוד המקצוע הוא זה שמציג עבור כל חייל את תפקידו (קוד המקצוע הוא מפתח ראשי בטבלת המקצועות ומפתח משני בטבלת החיילים).

המידע שיישלף יראה ככה (במידה ונבחר להציג את כל העמודות הללו):

|  |  |  |  |
| --- | --- | --- | --- |
| **שם החייל** | **תיאור המקצוע** | **קוד המקצוע** | **מספר חייל** |
| דורון | תוכניתן | 1800 | 001 |
| אביבית | מפעיל | 1802 | 002 |
| לירן | מפעיל | 1802 | 003 |
| גיא | תוכניתן | 1800 | 004 |

## פסוקית JOIN

עד עכשיו למדנו כיצד לשלוף מטבלאות יחידות רק מהטורים שנמצאים בטבלה. כעת נכיר פקודה חדשה שנקראת JOINשמאפשרת לנו **לשלוף מידע משתי טבלאות** (חיתוך).

**הגדרה**: דרך לשלוף שילוב של עמודות מטבלה אחת או יותר, תוך התבססות על מפתחות בין הטבלאות (בדרך כלל מפתח ראשי בטבלה אחת ומפתח משני בשנייה).

כך נוכל ליצור חיתוכים ולבצע שליפות מורכבות התלויות בישויות שונות במסד הנתונים.

הסוג הראשון והבסיסי ביותר של JOIN נקרא INNER JOIN ואותו נלמד ראשון.

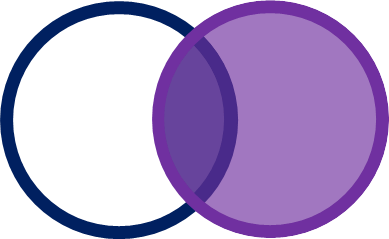
### INNER JOIN

שליפת המשותף לשתי הטבלאות לפי תנאי.

INNER JOIN היא פסוקית ה - JOIN הנפוצה ביותר וגם הדיפולטית (ניתן לרשום רק JOIN אך הסטנדרט הוא לרשום INNER JOIN).

פסוקית זו תחזיר לנו רק את המשותף בין 2 הטבלאות (נשתמש בה כשנרצה חיתוך מושלם בין טבלאות בעלות קשרי ONE TO MANY או MANY TO MANY).

**בתמונה – כל עיגול מייצג טבלה, ורק המידע המשותף ביניהן הוא זה שיישלף.**



**נחזור לטבלת המקצועות:**

|  |  |
| --- | --- |
| **תיאור המקצוע** | **קוד המקצוע** |
| תוכניתן | 1800 |
| מפעיל | 1802 |
| **מגן סייבר** | **1804** |

**שימו לב** שהוסף מקצוע לטבלה (טבלת החיילים נשארה אותו הדבר).

במידה ונבצע INNER JOIN בין שתי הטבלאות האם התוצאה תישאר אותו הדבר כמו בהתחלה?

**כן!**

אין שום חייל בתפקיד מגן סייבר, ולכן זהו לא מידע שמשותף בין שתי הטבלאות (בהמשך נלמד איך כן להציג מידע כזה – סוגים שונים של JOIN).

אחרי שהכרנו את הסוג הראשון והבסיסי ביותר, הגיע הזמן להסתכל על מבנה של שליפה עם JOIN:

SELECT col1,

col2

FROM table1

JOIN\_COMMAND table2 ON

join\_condition

WHERE additional\_condition [optional];

* אחרי ה**SELECT -**  יבואו שמות העמודות שנרצה לקבל מהטבלאות.
* אחרי ה **FROM -**  יהיה שם הטבלה הראשונה (מרכזית).
* **JOIN\_COMMAND** תהיה סוג פקודת הJOIN - שנרצה לבצע (נכיר את סוגיה השונים בהמשך) ואחריה תבוא הטבלה השנייה (הנוספת).
* אחרי ה - **ON** יבוא התנאי שייקשר בין העמודות שבטבלאות השונות.
* אחרי ה**WHERE -**  יש לנו אופציה להוסיף תנאים נוספים.

**איך זה נראה?**

SELECT  שם חייל,

תיאור המקצוע

FROM טבלת החיילים

JOIN\_COMMAND טבלת המקצועות ON

קוד המקצוע בטבלת החיילים שווה לקוד המקצוע בטבלת המקצועות

### שליפת JOIN על הטבלה של ליאת:

SELECT empno,

deptname

FROM liat.t\_empl

INNER JOIN liat.t\_depart ON

deptno = deptno;

איזו שגיאה עלולה לעלות מהשליפה?



ERROR: column reference "deptno" is ambiguous

LINE 4: ON deptno = deptno;

^

משמע, שם העמודה הוא דו משמעי (יכול להצביע ל2 עמודות שונות)

כאשר אנחנו מבצעים join בין טבלאות שיש להן עמודות בעלות שמות זהים נוצרת לנו בעיה – איך השליפה שלנו תדע לאיזה עמודה לגשת?

הסר את המלבן הלבן בשביל לחשוף את התשובה

הפתרון – נרשום לפני את שם העמודה של הטבלה ולאחריה נקודה.

SELECT liat.t\_empl.empno,

liat.t\_depart.deptname

FROM liat.t\_empl

INNER JOIN liat.t\_depart ON

     liat.t\_empl.deptno = liat.t\_depart.deptno;

הסר את המלבן הלבן בשביל לחשוף את התשובה

למעשה, כשאנחנו עושים JOIN בין שתי טבלאות, האיחוד נעשה על פי טורים מסוימים בכל טבלה, ושניהם ימצאו לנו בטבלת התוצאה של השליפה, גם אם הם זהים לחלוטין.

לדוגמה, אם נשלוף את כל העמודות בשתי הטבלאות – t\_empl ו t\_depart, והקישור יתבצע על סמך עמודת deptno בכל אחת מהטבלאות, נראה שהעמודה מופיעה פעמיים – אחת לקוחה מטבלת העובדים ואחת מטבלת המחלקות.

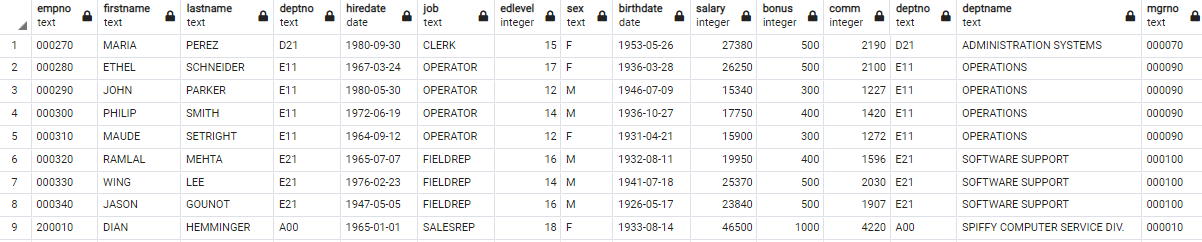
כאן העמודות יהיו זהות לחלוטין מאחר שעשינו INNER JOIN – כלומר יוצגו רק הערכים שמשותפים לשתי הטבלאות. נזכיר ששימוש בסימן כוכבית (\*) בפסוקית הSELECT שולף את כל העמודות בטבלת התוצאה הזמנית.

SELECT \*

FROM liat.t\_empl

INNER JOIN liat.t\_depart ON

deptno = deptno;



**liat.t\_depart**

**liat.t\_empl**

וכדי לסדר את הבלגן, נציין בפסוקיות ה – JOIN וה – FROM, שמות חדשים וקצרים יותר הנקראים - **QUALIFIERS**.

כך נוכל מעכשיו להשתמש בכינוי במקום לרשום את שם הטבלה:

SELECT emp.empno,

dpt.deptname

FROM liat.t\_empl emp

INNER JOIN liat.t\_depart dpt ON

emp.deptno = dpt.deptno;

כמה דגשים עבור QUALIFIERS:

* יש להשתמש ב - QUALIFIERS בכל שליפה בעלת **2 טבלאות או יותר**.
* יש לפנות **לכל** הטורים בשליפה, גם אם הם בעלי שמות שונים, באמצעות ה - QUALIFIERS המתאימים.
* QUALIFIERS יהיו שמות בעלי 3 אותיות **לפחות** (אך קצרים ככל הניתן).
* QUALIFIERS ייכתבו **באותיות קטנות** בלבד, ללא מספרים או תווים נוספים)מלבד \_).

צפו בסרטון: <https://www.youtube.com/watch?v=2X_qXnPg6G0&t=121s>

**מבנה הטבלאות**

כמו שצוין למעלה מבנה של שליפה בסיסית עם פסוקית JOIN נראה כך:

SELECT col1,

col2

FROM table1

JOIN\_COMMAND table2 ON

join\_condition

WHERE additional\_condition [optional]

### אבל מה זה בעצם table1 וtable2 - ?

בעת שימוש ב - JOIN אנחנו מסתכלים על הטבלאות בשליפה שלנו כטבלה ימנית וטבלה שמאלית. הטבלה השמאלית תהיה זו שב-FROM, ואילו הימנית ב - JOIN.

**כלומר:**

TABLE 1 תהיה הטבלה השמאלית – טבלה זו היא בעצם טבלת התוצאה והיא תכיל את כל מה שנמצא FROM.

TABLE 2 תהיה הטבלה הימנית – הטבלה "המצטרפת", הטבלה הימנית תמיד תהיה הטבלה ממנה נצרף מידע.

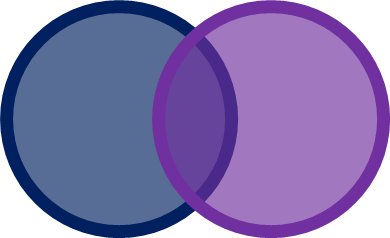
**השאלה המתבקשת היא**: מה היא הטבלה השמאלית או הימנית כאשר יש לנו יותר מ – JOIN אחד?

### LEFT JOIN (OUTER)

שליפת המשותף לשתי הטבלאות + הרשומות הייחודיות לטבלה השמאלית.

LEFT JOIN היא פסוקית ה – OUTER הנפוצה יותר, היא תחזיר לנו את כל הרשומות בטבלה השמאלית ("טבלת התוצאה") ותוסיף לה מהטבלה הימנית את כל השורות המתאימות (שורות שיש בהן ערך מהטבלה השמאלית).

רשומה מהטבלה השמאלית שאין לה שורה מתאימה מהטבלה הימנית תוחזר עם צירוףNULL בעמודות של הטבלה הימנית.



**חשבו:** כיצד היינו יכולים להציג רק את הרשומות שקיימות רק בטבלה השמאלית?

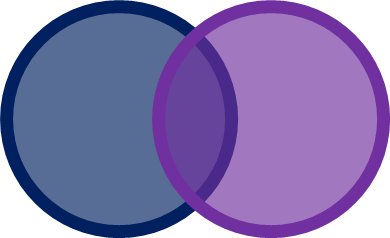
(פתרון יופיע בהמשך).

**RIGHT JOIN (OUTER)**

RIGHT JOIN הוא סוג נוסף של JOIN.

שליפת המשותף לשתי הטבלאות + הרשומות הייחודיות לטבלה הימנית.

RIGHTJOIN היא פסוקית ה - OUTER ההפוכה מ - LEFT, היא תחזיר לנו את כל הרשומות בטבלה הימנית ("טבלת המצטרפת") ותוסיף לה מהטבלה השמאלית את כל השורות המתאימות (שורות שיש בהן ערך מהטבלה הימנית), רשומה מהטבלה הימנית שאין לה שורה מתאימה מהטבלה השמאלית תוחזר עם צירוף NULL בעמודות של הטבלה השמאלית.



**נחזור אל טבלת המקצועות ואל טבלת החיילים:**

|  |  |
| --- | --- |
| **תיאור המקצוע** | **קוד המקצוע** |
| תוכניתן | 1800 |
| מפעיל | 1802 |
| מגן סייבר | 1804 |

|  |  |  |
| --- | --- | --- |
| **שם החייל** | **קוד המקצוע** | **מספר חייל** |
| דורון | 1800 | 001 |
| אביבית | 1802 | 002 |
| לירן | 1802 | 003 |
| גיא | 1800 | 004 |
| **רועי** | **1806** | **005** |

**שימו לב שהוסף חייל לטבלת החיילים.**

עכשיו נרצה להציג עבור **כל** החיילים את מקצועם. כיצד נעשה זאת?

נשתמש ב – LEFT JOIN:

|  |  |  |
| --- | --- | --- |
| **שם החייל** | **תיאור המקצוע** | **קוד המקצוע** |
| דורון | תוכניתן | 1800 |
| אביבית | מפעיל | 1802 |
| לירן | מפעיל | 1802 |
| גיא | תוכניתן | 1800 |
| רועי | **NULL** | 1806 |

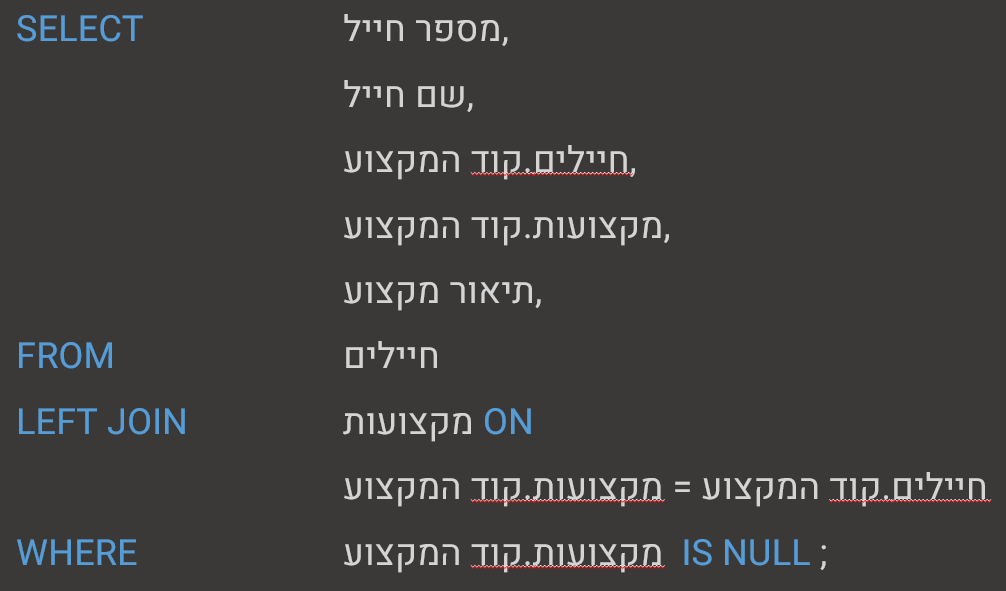
במקצוע של החייל רועי מופיע NULL. מדוע?

**תשובה:**

קוד המקצוע של החייל רועי הוא 1806. קוד זה אינו קיים בטבלת המקצועות ולכן לא ידוע איזה מקצוע יש לחייל רועי. בחרנו לבצע LEFT JOIN לכן כל החיילים שנמצאים בטבלה השמאלית מופיעים(בחרנו גם שהטבלה של החיילים היא הטבלה השמאלית).

בשתי הפסוקיות LEFT וRIGHT ניתן להוסיף את פסוקית ה- **WHERE column IS NULL** ולקבל רק את הייחוד של הטבלה הימנית/שמאלית לפי כיוון הJOIN – (כפי שניתן לראות בציור).

**לדוגמה**, אם נרצה להציג רק את החיילים שהמקצוע שלהם **לא** מופיע בטבלת המקצועות או שאין להם מקצוע, נוכל לעשות זאת בעזרת השליפה הבאה:



בשליפה זו ישלפו לנו כל החיילים שהערך שלהם בעמודות שלקוחות מטבלת המקצועות הוא NULL, כלומר שלא קיימים בטבלה הימנית. אלו חיילים שאין להם קוד מקצוע או שקוד המקצוע שלהם לא קיים בטבלת המקצועות, כמו רועי.

מאחר שקוד המקצוע של רועי לא מופיע בטבלת המקצועות, אז הערך שיופיע בעמודת קוד המקצוע של טבלת המקצועות יהיה NULL. ובאמת, טבלת התוצאה של השליפה הזו תהיה רק הרשומה של רועי.

נעדיף להשתמש בתנאי זה בעמודות שהן מפתח ראשי/ משני בטבלה. זאת מכיוון שעמודות אחרות יכולות להכיל NULL מסיבות אחרות (לדוגמה, אם כן יש מקצוע שקיים, רק שכחו להזין את תיאור המקצוע שלו)

****

### FULL OUTER JOIN

שליפת כל הנתונים בטבלאות.

FULL JOIN הוא שילוב של RIGHT ו LEFT ולמעשה שולף את **כל** המידע ב2 הטבלאות, מצליב מידע מ - 2 הטבלאות היכן שמתאים והיכן שלא מחזיר רשומה בצירוף NULL בעמודות הטבלה השנייה (עבור כל טבלה).

היקף השימוש ב FULL JOIN די מצומצם ולרוב נשתמש בו להסתכלות והשוואה בין 2 טבלאות ואו חילוץ המידע הלא משותף בין הטבלאות (כל מה שלא נכלל ב INNER JOIN) בעזרת WHERE column IS NULL.

SELECT   emp.firstname || ‘ ‘ || emp.lastname AS emp\_name,

dpt.deptname

FROM      liat.t\_depart dpt

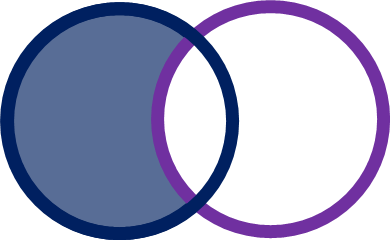
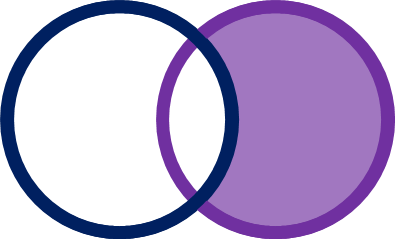
FULL JOIN liat.t\_empl emp ON

         emp.deptno = dpt.deptno;

מה שנשלף:

כל פרטי העובדים והמחלקות שקיימים במפעל של ליאת. **זאת** **ללא קשר אם יש במחלקה עובדים או אם המחלקה של עובד מסוים לא נמצאת בטבלת המחלקות** (עבור מה שלא קיים הטורים יישלפו כ -NULL ).

שליפה עם WHERE column IS NULL OR column IS NULL.



(באותה דרך היה ניתן לפתור את הבעיה ב – LEFT JOIN).