# פקודת SELECT בסיסית

**פקודת ה-SELECT**

פסוקית ה-SELECT ב-SQL מאפשרת שליפה של נתונים מתוך טבלאות ומחזירה טבלה הנקראת

**טבלת התוצאה**.

**מבנה פקודת SELECT**

המבנה הבסיסי של SELECT מורכב משתי הפסקאות הראשונות:

1. SELECT - אילו טורים (שדות) לשלוף לתוך טבלת התוצאה.
2. FROM - מאיזו טבלה לשלוף את ערכי השדות.

לא לשכוח ; בסוף פקודת ה-SELECT (כמו בסוף כל פקודת SQL).

**ערכים חוקיים ב-SELECT**

1. **שם טור** – במידה ויש כמה, נפריד ע"י פסיקים וירידת שורה.

**דוגמה:** שליפת מספרי ושמות של כל העובדים מטבלת העובדים (liat.t\_empl).

SELECT empno,

firstname

FROM liat.t\_empl;

1. **ציון \* ב -SELECT** **:** זהה להשמת שמות כל הטורים המוגדרים בטבלה שצוינה ב-FROM.

אם מציינים \* אין לציין שמות טורים נוספים.

**דוגמה:** שליפת כל הטורים המוגדרים בטבלת t\_empl. עבור כל שורה בטבלת המקור (FROM) נקבל שורה בטבלת התוצאה עם כל הטורים המוגדרים בטבלה.

SELECT \*

FROM liat.t\_empl;

**נשתדל להימנע משימוש בשליפת כל הטורים בשליפות שלנו, ובפרט בשימוש ב-\*, אלא אם יש בכך צורך (פירוט נוסף בקובץ הסטנדרטים).**

1. **ביטוי** - הביטוי יכול להיות ביטוי חשבוני המכיל קבועים מספריים ו/או טורים המכילים ערכים מספריים או קבועי מחרוזת ו/או טורים המכילים ערכים מחרוזתיים.

SELECT empno,

firstname,

1.10 \* salary

FROM liat.t\_empl;

**דוגמה:** שליפת משכורות כל העובדים, לאחר תוספת של 10% לשכרם.

**שימו לב אין מדובר בשינוי הנתונים המקוריים בטבלה**. ה-DBMS מחשב את התוצאה המתאימה לכל שורה ומכניס אותה לטבלת התוצאה, כאילו היה מדובר בטור מן הטבלה המקורית.

אם נציין קבוע כאחד הטורים (למשל הקבוע המחרוזתי ‘PostgreSQL Master') אותו קבוע יופיע עבור כל השורות בטבלת התוצאה (נסו זאת).

מספר הטורים בטבלת התוצאה זהה תמיד למספר הטורים שנבחר ב-SELECT.

#### ערכים חוקיים ב-FROM

ב - FROM נרשום את שם הטבלה ממנה מבצעים את השליפה. שם הטבלה צריך להתאים לחוקי שמות טבלה במסד הנתונים איתו עובדים. כמובן שהטבלה צריכה להיות מוגדרת וקיימת ב - DB שלנו.

**פסוקית ה-ORDER BY**

**מהו פסוקית ה-ORDER BY?**

משפט ה -SELECT מאפשר לבקש מה - DBMS למיין את טבלת התוצאה על פי אחד או יותר מהטורים בטבלת התוצאה, על ידי שימוש בפסוקית ORDER BY.

### איך נבצע זאת?

בפסוקית ה-ORDER BY מציינים את שמות הטורים שעל פיהם צריך ה-DBMS למיין את טבלת התוצאה. לכל שדה מיון, ניתן לציין אם מעוניינים במיון עולה (ASC - ברירת המחדל) או יורד (DESC).

#### מבנה של SELECT עם ORDER BY

SELECT       ...

FROM         ...

ORDER BY    שדות מיון;

#### שליפת פרטי כל העובדים - מיון עפ"י שם פרטי

SELECT empno,

firstname

FROM liat.t\_empl

ORDER BY firstname DESC, empno;

שליפת מספרי ושמות כל העובדים, כאשר טבלת התוצאה תמוין במיון ראשי לפי שם פרטי בסדר יורד. אם מופיע אותו שם פרטי פעמיים, יופיעו מספרי העובד בסדר עולה (ברירת המחדל).

**פסוקית ה-WHERE**

**מהי פסוקית ה-WHERE ?**

משפט ה-SELECT מאפשר לציין תנאים המגדירים סינון של שורות, המתבצע על ידי שימוש בפסוקית ה-WHERE.

**מבנה של SELECT עם WHERE**

SELECT  ...

FROM    ...

WHERE   תנאי

ORDER BY    שדות מיון;

כתנאי בפקודת ה-WHERE ניתן לרשום כל ביטוי מורכב המחזיר תוצאה דוגמה לתנאים שניתן לציין בפסוקית ה-WHERE

* + 1. (salary >= 5)
    2. (job = 'CLERK')
    3. (y > x)
    4. (y = x)

## תנאים מורכבים

ניתן לעשות תנאים מורכבים באמצעות המילים השמורות AND ו OR

* AND – רשומה תישלף אך ורק אם היא מקיימת את שני התנאים
* OR – רשומה תישלף אם היא מקיימת לפחות אחד מהתנאים

#### שליפה עפ"י תאריך תחילת עבודה ושכר

SELECT   empno,

firstname,

hiredate,

salary

FROM     liat.t\_empl

WHERE    (hiredate >= '01-JAN-1981') AND

  (salary > 2000);

שליפת פרטי העובדים שנתקבלו לעבודה בשנת 1981 ומעלה ושכרם עולה על 2000. בטבלת התוצאה תופענה רק שורות העונות על הקריטריונים הנ"ל.

**NULL -**  מסמן לי שלא הוזן הערך של התא. ניתן לבדוק זאת בעזרת הפקודה . **IS NULL**

SELECT deptno,

deptname,

admrdept

FROM liat.t\_depart

WHERE admrdept IS NULL;

העבודה עם NULL היא שונה – לא נשתמש בתנאי רגיל, כלומר **Deptno = NULL** לא יעבוד!

#### שליפה מורכבת של עובדות משכילות

שלוף את שמה הפרטי , שם משפחתה והשכלתה של עובדת אשר למדה יותר מ- 10 שנים, ואשר קוד מחלקתה איננו מתחיל באות 'C', בטבלת התוצאה השורות תהיינה ממוינות לפי שם משפחה בסדר יורד.

SELECT firstname,

lastname,

edlevel

FROM liat.t\_empl

WHERE (sex = 'F') AND

(deptno NOT LIKE 'C%') AND

(edlevel > 10)

ORDER BY lastname DESC;