# אופרטורים מתקדמים

## האופרטורים IN, BETWEEN, NOT, DISTINCT, LIKE :

אז לפני שנדבר על האופרטורים הנ"ל, נזכר מהו אופרטור...

#### אופרטורים - PREDICATES

ראינו שפסוק ה-WHERE מאפשר לנו לסנן שורות באמצעות ציון תנאי בודד או מספר תנאים שביניהם יחס לוגי (AND/OR). צורת התניה זו (אופרטור לוגי בין שני ביטויים) היא האופרטור הבסיסי ביותר. SQL מאפשרת לנו להשתמש באופרטורים מורכבים יותר, המאפשרים לנו לסנן את השורות בצורה יותר מתוחכמת.

#### האופרטור BETWEEN

האופרטור BETWEEN מאפשר סינון של שורות, כך שבטבלת התוצאה תופענה רק שורות שעבורן הערך אותו בודקים, נע בין שני ערכים כלשהם. השימוש ב-BETWEEN זהה לשימוש בתנאי מורכב המגביל את ערכו של טור מסוים לגבול עליון ותחתון מסוימים.

WHERE ביטוי נבדק BETWEEN ביטוי מינימאלי AND ביטוי מקסימאלי

#### שימוש ב-BETWEEN לסינון שורות

SELECT empno, firstname

FROM liat.t\_empl

WHERE salary BETWEEN 15000 AND 25000;

לאחר השליפה, יופיעו מספרי ושמות כל העובדים שמשכורתם נעה בין 15000 ל-25000 בטבלת התוצאה. יש לזכור כי השימוש ב-BETWEEN הוא לצורכי נוחות ובעצם ה- SELECT הבא ייתן את אותה תוצאה:

SELECT empno, firstname

FROM liat.t\_empl

WHERE (salary >= 15000) AND

         (salary <= 25000);

#### האופרטור IN

האופרטור IN מאפשר לסנן שורות על פי רשימה של ערכים. כלומר, בטבלת התוצאה יופיעו רק שורות שבאחד הטורים שלהן יש ערך המופיע כאחד מהערכים ברשימה. כמובן שסוג הערכים ברשימה צריך להתאים לסוג הביטוי (מספר/מחרוזת/תאריך וכו').

WHERE ביטוי IN (... ,ערך3 ,ערך2 ,ערך1)

#### שימוש ב-IN לסינון שורות

SELECT empno, firstname, salary

FROM liat.t\_empl

WHERE deptno IN ('A01', 'E10', 'D05');

שליפת פרטי כל העובדים במחלקות ‘A01’, ‘E10’, ‘D05’. שליפה זו תהיה זהה לשליפה :

SELECT empno, firstname, salary

FROM liat.t\_empl

WHERE (deptno = 'A01') OR

(deptno = 'E10') OR

(deptno = 'D05');

#### ואם עכשיו אני רוצה לבדוק האם עובד מסוים נמצא במחלקה מתוך רשימה של מחלקות וגם שמשכורתו היא מתוך רשימת המשכורות? זה ייראה כך:

SELECT empno, firstname, salary

FROM liat.t\_empl

WHERE (deptno IN ('A01', 'D21', 'D05')) AND

         (salary IN (3000, 27380, 12352));

דרך נוספת לבצע את אותה הפעולה היא כך:

SELECT empno, firstname, salary

FROM liat.t\_empl

WHERE (deptno, salary) IN (('A01', 3000), ('D21', 27380), ('D05', 12352));

#### שימו לב שדרך זו תבדוק האם עובד מסוים עונה על אחת מהאפשרויות המוצעות.

כלומר: האם העובד נמצא במחלקה A01 וגם מקבל משכורת של 3000, או שהעובד נמצא במחלקה D21 וגם מקבל משכורת של 27380, או שהעובד נמצא במחלקה D05 וגם מקבל משכורת של 12352.

לכן עבור כל וריאציה נוספת עלינו להוסיף עוד אפשרות לרשימה המוצעת.

#### האופרטור NOT

האופרטור NOT מאפשר להשתמש בשלילה לוגית. להבדיל משאר האופרטורים, אופרטור ה-NOT יכול להופיע בצירוף (לפני) כל אופרטור אחר, על מנת לשאול על השלילה של אותו אופרטור.

#### שליפה בעזרת NOT

SELECT firstname

FROM liat.t\_empl

WHERE (NOT firstname = 'SAMMY') AND

(salary < 3000)

בדוגמא זו, נקבל בטבלת התוצאה את שמות כל העובדים ששמם אינו ‘SAMMY’ ואשר ששכרם נמוך מ- 3000. יש לשים לב כי ה -NOT פועל על אופרטור אחד בכל פעם. ניתן לשנות זאת על ידי שימוש שונה בסוגריים (כמובן).

#### האופרטור LIKE

האופרטור LIKE מאפשר סינון של שורות על פי PATTERN (תבנית) כלשהי. לצורך קביעת ה -PATTERN, משתמשים בסימן '%' ובסימן '\_'. הסימן '%' מסמל מספר כלשהו של תווים והסימן '\_' מסמל תו אחד.

WHERE ביטוי LIKE 'pattern'

שמות עובדים המתחילים בתו S:

SELECT firstname, deptno

FROM liat.t\_empl

WHERE    firstname LIKE 'S%';

עובדים שהועסקו בחודש דצמבר:

SELECT firstname, salary

FROM liat.t\_empl

WHERE hiredate LIKE '%DEC%';

עובדים שהאות השנייה בתפקיד שלהם היא A:

SELECT firstname, job

FROM liat.t\_empl

WHERE job LIKE '\_A%'

שים לב כאשר מתייחסים למספר באופרטור LIKE עלינו לשים אותו בין גרשיים, בעצם מתייחסים אליו כאל מחרוזת, ה-DBMS דואג לבצע את ההסבות הנדרשות.

**אופרטור ה-DISTINCT**

אופרטור ה- DISTINCT מאפשר לצמצם את תוכן טבלת התוצאה, כך ששורות זהות, יופיעו פעם אחת בלבד.

#### מבנה של SELECT עם DISTINCT

SELECT DISTINCT ...

FROM ...

#### SELECT עם DISTINCT

SELECT DISTINCT salary

FROM liat.t\_empl

ORDER BY salary;

בדוגמא זו, נקבל בטבלת התוצאה את מגוון המשכורות שמקבלים העובדים, במיון עולה לפי גובה המשכורת, כאשר גם אם מספר עובדים מקבלים אותו שכר, בטבלת התוצאה נקבל שורה אחת בלבד עבור אותו שכר.

DISTINCT עובד כך שהוא יחזיר לנו כל קומבינציה ייחודית של **כל העמודות** שנכתוב אחריו ב SELECT. מה אם נרצה לבצע את הפונקציה רק על עמודה אחת, ועדיין להציג ב SELECT עמודות נוספות שעליהן הDISTINCT לא יפעל?

חפשו באינטרנט על הפונקציה DISTINCT ON, וקראו עליה. נסו להבין מהו ההבדל בין DISTINCT ל- DISTINCT ON ונסו לכתוב שליפה שמשתמשת בפונקציה בכדי להבין כיצד היא עובדת.

**הערך NULL**

שדה עם ערך NULL הוא שדה ללא ערך.

אם שדה בטבלה הוא לא חובה, ניתן להכניס רשומה חדשה או לעדכן רשומה מבלי להוסיף ערך לשדה זה. לאחר מכן, השדה יישמר עם ערך .NULL

**שימו לב:** הערך NULL שונה מערך אפס או שדה המכיל רווחים. שדה עם ערך NULL הוא תחום שהושאר ריק במהלך יצירת הרשומות.

**איך נבדוק שדות עם ערך NULL ?**

לא ניתן לבדוק שדות NULL עם אופרטורים של השוואה כמו  =, <, או <>

במקום זאת נשתמש ב IS NULL וב IS NOT NULL

#### האופרטור IS NULL

האופרטור IS NULL משמש לבדיקת ערכים ריקים (ערכי NULL)

SELECT column\_names

FROM table\_name

WHERE column\_name IS NULL;

#### האופרטור IS NOT NULL

האופרטור IS NOT NULL משמש לבדיקת ערכים שאינם ריקים (לא ערכי NULL)

SELECT column\_names

FROM table\_name

WHERE column\_name IS NOT NULL;

**שימו לב, יש להשתמש באופרטורים אלו בכל הזדמנות בה תראו את הצורך.**

לדוגמא, במקום בדיקת טווח מפורשת בעזרת שני תנאים, נעדיף את האופרטור BETWEEN.

אופרטורים חזקים לחיים נוחים,

צוות מסדי נתונים.