# פונקציות טוריות

## מהי פונקציית טור?

פונקצית טור מכונה גם פונקציה אגרגטיבית, פונקצית עמודה, או פונקציה מסכמת.

פונקציות טור הנן פונקציות הפועלות על מספר שורות ומחזירות ערך בודד, המבוסס עליהן. כמעט כל פונקציות הטור "מתעלמות" מערכי NULL ואינן מביאות אותם בחשבון בחישוב התוצאה.

למרות זאת, הפונקציות תחזרנה ערך של NULL במקרים הבאים:

1. לא היו שורות שענו על הקריטריונים (אותם מציינים ב-WHERE).
2. כל השורות שענו על הקריטריונים הכילו NULL בטור עליו מפעילים את הפונקציה.

ניתן לציין את מילת המפתח DISTINCT לרוב פונקציות הטור, על מנת לגרום לפונקציה לעבוד על "עותק" אחד מכל ערך.

## הפונקציה SUM

הפונקציה מחזירה את סכום כל הערכים בטור מסוים בטבלה.

### סכום המשכורות במחלקה:

SELECT  SUM(salary)

FROM    liat.t\_empl

WHERE   deptno = 'A00';

בדוגמא זו, מחשבים את סכום המשכורות שמקבלים העובדים במחלקה A00. תחילה נבדק תנאי ה - WHERE על כל רשומה בטבלה, כך שרק הרשומות הרלוונטיות נשארות, ולאחר מכן מופעלת פונקצית הסכימה על הרשומות שנותרו.

## הפונקציה AVG

הפונקציה מחזירה את ממוצע הערכים של טור מסוים בטבלה.

### ממוצע המשכורות השונות:

SELECT  AVG(DISTINCT salary)

FROM    liat.t\_empl

WHERE   hiredate >= '01-JAN-96';

בדוגמא זו, מחשבים את ממוצע המשכורות הייחודיות שקיבלו עובדים שהועסקו החל מינואר, 1996. בחישוב הממוצע, משכורת זהה למספר עובדים, תיכנס פעם אחת לתוצאה. כלומר, למספר העובדים שקיבלו אותו שכר, לא תהיה השפעה על התוצאה(DISTINCT בתוך הסוגריים).

ערכי NULL בטור SALARY לא "ישתתפו" בחישוב התוצאה. למרות זאת, יתכן ותוצאת הפונקציה תהיה NULL (אם לא יהיו רשומות שימלאו את התנאי שהצבנו בפסוקית הWHERE).

כמו כן, התוצאה חוזרת כשורה בודדת (ולא ערך המשוכפל כמספר השורות בטבלה). מכיוון שכפי שאמרנו, מטרתן של הפונקציות האגרגטיביות היא לסכם טור בעל כמות שורות לערך בודד.

**שימו ♥ - כי לעיתים הערכים יוסתרו לנו בטבלת התוצאה ויש להרחיב את התא המכיל את התוצאה שאנו רוצים לראות.**

## הפונקציה MIN

הפונקציה מחזירה את הערך הנמוך (הקטן) ביותר בטור מסוים בטבלה.

### תאריך תחילת העבודה של העובד הוותיק ביותר במחלקה כלשהי:

SELECT  MIN(hiredate)

FROM    liat.t\_empl

WHERE   deptno = 'B01';

בדוגמא זו, מוצאים את תאריך הצטרפותו לחברה, של העובד הוותיק ביותר במחלקה B01 – כלומר העובד שנשכר בתאריך המוקדם ביותר.

## הפונקציה MAX

הפונקציה מחזירה את הערך הגבוה (הגדול) ביותר בטור מסוים בטבלה.

### המשכורת הגבוהה ביותר מכל העובדים:

1   SELECT  MAX(salary)

2   FROM    liat.t\_empl;

בדוגמא זו, מוצאים את המשכורת הגבוהה ביותר.

## הפונקציה COUNT

הפונקציה מחזירה את מספר השורות העונות על שאילתה כלשהי.

### מספר העובדים במחלקה

1   SELECT  COUNT(empno)

2   FROM    liat.t\_empl

3   WHERE   deptno = 'A10';

בדוגמא זו, מוצאים את מספר העובדים במחלקה 10A, במקרה הזה הוא 0.

### מספר המנהלים - שימוש ב-DISTINCT

1   SELECT  COUNT(DISTINCT mgrno)

2   FROM    liat.t\_depart;

בדוגמא זו, מוצאים את מספר המנהלים בארגון. DISTINCT מאפשר לצמצם ערכים כפולים, ללא DISTINCT נקבל את מספר השורות המלא (כולל כפולים), מאחר שפונקציית COUNT ללא DISTINCT סופרת כל שורה בה הערך בטור הוא לא NULL ללא התייחסות אם הערך עצמו חוזר על עצמו או לא.

### שימוש ב-(\*)COUNT

מה לדעתכם תיתן שליפה שמשתמשת ב (\*)COUNT?

כמו שאר פונקציות הטור, גם הפונקציה COUNT מתעלמת מערכי NULL ואינה מביאה אותם בחישוב מספר השורות. להבדיל משאר הפונקציות שהכרנו, ניתן "לבקש" מהפונקציה להתייחס גם לשדות המכילים NULL. בקשה זו מציינים על-ידי השימוש ב-(\*)COUNT.

### מספר העובדים בארגון

1   SELECT  COUNT(\*)

2   FROM    liat.t\_empl;

בדוגמא זו, נשלפת כמות העובדים בארגון.

אם היינו מפעילים את הפונקציה COUNT על טור ספציפי (לדוגמא salary), הרשומות בהן היה ערך NULL עבור שדה זה (salary) לא היו נכנסות לחישוב. בעזרת ה - \* אנו סופרים את כל הרשומות בטבלה.

עם זאת, אנחנו **לא** נשתמש ב COUNT(\*) כאשר נרצה לספור את כל הרשומות בטבלה.

ממבט ראשון היא נראית כמו אופציה קצרה ונוחה יותר, אבל בגלל הניסוח הכללי שלה היא לא תעמוד בפני שינויים במבנה הטבלה או בנתונים עצמם. תמיד נשאף להיות מדויקים בדרישות שלנו, ולציין מה אנחנו רוצים לספור.

ולכן כאשר נצטרך לספור רשומות ללא ערכי ה- NULL, נבצע את ה- COUNT על העמודות שרלוונטיות לנו ובמידה שיעלה לנו צורך לספור את כל הרשומות, האופציה העדיפה יותר היא לבצע את ה- COUNT על עמודת המפתח הראשי. כאשר סופרים עמודה שהיא מפתח ראשי, לא ניתן להתעלם מערכי ה-NULL ואנו מציינים בפירוט את העמודה שאנחנו מבקשים לספור.

השימוש ב- COUNT(\*) נפוץ וייתכן שתיתקלו בו לא מעט, ולכן חשוב לדעת שאופציות אחרות הן נכונות יותר ובהן נשתמש במהלך הקורס.