פונקציות SQL

# NTILE(n)

משמשת לחלוקת ערכת תוצאות למספר מוגדר של קבוצות או דליים. הוא מפיץ את השורות בערכת התוצאות באופן שווה בין הקבוצות הללו. הפונקציה מקצה מספר קבוצה לכל שורה בהתבסס על מספר הקבוצות שצוין וסידור השורות.

להלן התחביר עבור הפונקציה NTILE():

NTILE(n) OVER (ORDER BY column)

הסבר:

* **N** הוא המספר הרצוי של קבוצות או דליים.
* **ORDER BY column** הוא משפט אופציונלי המציין את העמודה לפיה יש לסדר את השורות לפני חלוקתן לקבוצות. אם לא מסופק, השורות מסודרות באופן שרירותי.

הפונקציה NTILE() פועלת באופן הבא:

1. השורות בערכת התוצאות מסודרות על סמך העמודה שצוינה (אם מסופקת) או סדר שרירותי.
2. השורות מחולקות לקבוצות  **n** שוות בגודלן. אם המספר הכולל של השורות אינו מתחלק באופן שווה ב- **n** , ייתכן שלקבוצות מסוימות יש שורה אחת יותר מאחרות.
3. לכל שורה מוקצה מספר קבוצה מ-1 עד **n**  המציין לאיזו קבוצה היא שייכת. מספר הקבוצה מוקצה על סמך מיקום השורה בתוך ערכת התוצאות המסודרת.

באמצעות הפונקציה NTILE() תוכל לבצע פעולות שונות בערכת התוצאות בהתבסס על מספר הקבוצה שהוקצה. זה יכול לכלול דירוג, ניתוח נתונים בתוך כל קבוצה, או הפצה שווה של הנתונים לעיבוד מקביל או עימוד, בין מקרי שימוש אחרים.

## דוגמה לשימוש בפונקציה:

נניח שיש לנו טבלה בשם "עובדים" עם העמודות הבאות: "מזהה עובד", "שם פרטי", "שם משפחה" ו"שכר". אנחנו רוצים לחלק את העובדים לשלוש קבוצות לפי שכרם.

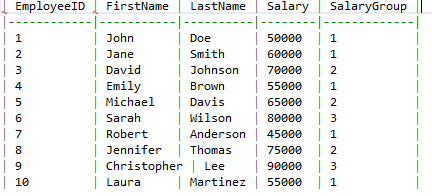
SELECT EmployeeID, FirstName, LastName, Salary, NTILE(3) OVER (ORDER BY Salary) AS SalaryGroup

FROM Employees;

שאילתה זו תקצה כל עובד לאחת משלוש הקבוצות (1, 2 או 3) על סמך השכר שלו. הפונקציה NTILE(3) מחלקת את ערכת התוצאות לשלוש קבוצות שוות בגודלן. הסעיף ORDER BY משמש כדי לציין את העמודה לפיה יש לבצע את הקיבוץ, במקרה זה, "שכר".

התוצאה תכלול את העמודות המקוריות מטבלת "עובדים" יחד עם עמודה נוספת "קבוצת שכר" המייצגת את הקבוצה אליה משתייך כל עובד.

להלן פלט לדוגמה:



כפי שניתן לראות, העובדים מחולקים לשלוש קבוצות בהתאם לשכרם. קבוצה 1 מכילה משכורות נמוכות יותר, קבוצה 2 מכילה משכורות ביניים, וקבוצה 3 מכילה משכורות גבוהות יותר.

# SUBSTR()

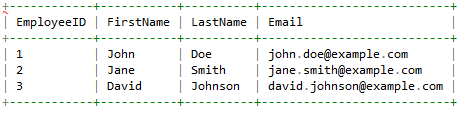
משמשת לחילוץ מחרוזת משנה ממחרוזת נתונה. פונקציה זו יכולה להיות שימושית במספר תרחישים:

1. **מניפולציה של נתונים: SUBSTR()** מאפשר לך לחלץ חלק מסוים של מחרוזת על סמך מיקום ההתחלה והאורך. זה מועיל כאשר אתה צריך לשנות או לשנות את הנתונים בתוך עמודת מחרוזת. לדוגמה, אתה יכול לחלץ את השם הפרטי מעמודת שם מלא או לחלץ חלק מסוים של מחרוזת המכילה מידע כגון תאריכים, קודים או מזהים.
2. **מיסוך נתונים:** במקרים בהם יש צורך להסוות מידע רגיש, ניתן להשתמש ב-SUBSTR() כדי לחלץ חלק ממחרוזת ולהחליפו בתווים מוסווים או מעורפלים. זה עוזר להגן על נתונים רגישים ועדיין מאפשר חשיפה חלקית או אנונימיזציה.
3. **אימות נתונים:** ניתן להשתמש ב-SUBSTR() כדי לאמת או לבדוק את הפורמט של מחרוזת על ידי חילוץ חלק מסוים והחלת כללי אימות. לדוגמה, אתה יכול לחלץ חלק ממספר טלפון ולאמת את הפורמט שלו או לחלץ חלק מסוים מכתובת דוא"ל כדי לוודא שהוא עוקב אחר פורמט חוקי.
4. **דיווח נתונים: SUBSTR()** יכול לסייע ביצירת דוחות או סיכומים מותאמים אישית על ידי חילוץ מידע רלוונטי מעמודת מחרוזת. לדוגמה, אתה יכול לחלץ את השנה ממחרוזת תאריך כדי ליצור דוח סיכום שנה או לחלץ מילות מפתח ספציפיות משדה טקסט להמשך ניתוח או סיווג.
5. **אופטימיזציה של שאילתות:** במקרים מסוימים, שימוש ב-SUBSTR() יכול לסייע באופטימיזציה של ביצועי השאילתה. על ידי חילוץ מחרוזת משנה מעמודה גדולה מראש ואחסנה בעמודה נפרדת, תוכל להימנע מפעולות משנה חוזרות ונשנות במהלך השאילתה, וכתוצאה מכך זמן ביצוע שאילתה משופר.

בסך הכל, הפונקציה SUBSTR() מספקת גמישות במניפולציה וחילוץ נתונים ממחרוזות, ומאפשרת לך לבצע טרנספורמציות, אימותים וניתוחים שונים של נתונים ביעילות.

## דוגמה לשימוש בפונקציה:

נניח שיש לנו טבלה בשם "עובדים" עם העמודות הבאות: "מזהה עובד", "שם פרטי", "שם משפחה" ו"אימייל".



אנו רוצים לחלץ את שם הדומיין מכתובות המייל של כל עובד. אנו יכולים להשתמש בפונקציה SUBSTR() כדי להשיג זאת.

SELECT EmployeeID, FirstName, LastName, SUBSTR(Email, INSTR(Email, '@') + 1) AS Domain

FROM Employees;

בדוגמה זו:

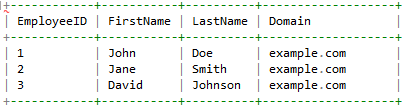
* הפונקציה **INSTR()**

משמשת כדי למצוא את המיקום של הסמל '@' בתוך כתובת האימייל.

* הפונקציה **SUBSTR()**

משמשת לאחר מכן כדי לחלץ את המחרוזת המשנה החל מהמיקום שאחרי '@' עד סוף כתובת הדואר האלקטרוני, ואחזר למעשה את שם הדומיין.

ערכת התוצאות תהיה:



כפי שניתן לראות, הפונקציה SUBSTR() מחלצת את שם הדומיין מכתובות האימייל, ומאפשרת לנו לנתח או לדווח על הדומיין בנפרד.

# INSTR()

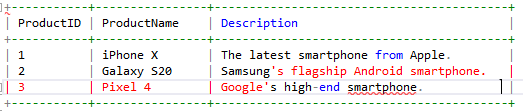
משמשת כדי למצוא את המיקום של מחרוזת משנה בתוך מחרוזת. פונקציה זו יכולה להיות שימושית בתרחישים שונים:

1. **מניפולציה של מחרוזת: INSTR()** מאפשרת לאתר תת-מחרוזת ספציפית בתוך מחרוזת גדולה יותר. זה יכול להיות מועיל כאשר אתה צריך לתפעל או לחלץ חלקים של מחרוזת על סמך מיקומה. לדוגמה, אתה יכול למצוא את המיקום של תו או מילה ספציפיים בתוך מחרוזת ולאחר מכן להשתמש במיקום זה כדי לבצע מניפולציות נוספות של מחרוזת.
2. **אימות נתונים:** ניתן להשתמש ב-INSTR() כדי לאמת נוכחות או היעדר של מחרוזת משנה בתוך מחרוזת. אתה יכול לבדוק אם תת-מחרוזת מסוימת קיימת בתוך ערך עמודה, ובהתבסס על התוצאה, לבצע פעולות או אימותים ספציפיים. לדוגמה, אתה יכול להשתמש ב-INSTR() כדי לבדוק אם מילת מפתח נדרשת קיימת בשדה טקסט או אם קיים דפוס מסוים במחרוזת.
3. **חיפוש וסינון: INSTR()** יכול לסייע בחיפוש וסינון נתונים על סמך התרחשות או מיקומה של מחרוזת משנה. לדוגמה, אתה יכול להשתמש ב-INSTR() כדי לסנן רשומות המכילות תת-מחרוזת מסוימת או כדי לחפש שורות שבהן תת-מחרוזת מסוימת מופיעה בהתחלה, בסוף או בכל מיקום בתוך מחרוזת.
4. **ניתוח ודיווח מחרוזות:** ניתן להשתמש ב-INSTR() כדי לבצע ניתוח על מחרוזות. אתה יכול למצוא את המיקום של תו או מילה כדי לחשב מדדים כגון תדירות ההתרחשות, אורך של מחרוזת משנה מסוימת או המרחק בין שתי מחרוזות משנה בתוך טקסט. מידע זה יכול לשמש לדיווח או לניתוח נתונים נוסף.
5. **אופטימיזציה של שאילתות:** במקרים מסוימים, שימוש ב INSTR() יכול לעזור לייעל את ביצועי השאילתה. על ידי בדיקת המיקום של מחרוזת משנה לפני ביצוע פעולות מורכבות יותר, תוכל למנוע חישובים מיותרים או להקטין את גודל מערך הנתונים לצורך עיבוד נוסף.

בסך הכל, הפונקציה INSTR() מספקת גמישות במניפולציה וניתוח של מחרוזות, ומאפשרת לך לבצע משימות כגון מניפולציות מחרוזות, אימות, חיפוש ודיווח ביעילות.

## דוגמה לשימוש בפונקציה:

נניח שיש לנו טבלה בשם "מוצרים" עם העמודות הבאות: "מזהה מוצר", "שם מוצר" ו"תיאור".



אנחנו רוצים למצוא מוצרים שהמילה "סמארטפון" מוזכרת בתיאור שלהם. אנו יכולים להשתמש בפונקציה INSTR() כדי להשיג זאת.

SELECT ProductID, ProductName, Description

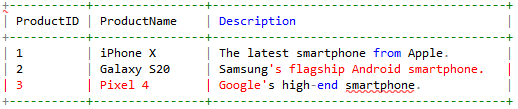
FROM Products

WHERE INSTR(Description, 'smartphone') > 0;

בדוגמה זו:

* הפונקציה **INSTR()** משמשת למציאת המיקום של המחרוזת המשנה 'סמארטפון' בתוך העמודה "תיאור".
* הסעיף **WHERE** מסנן את ערכת התוצאות כך שתכלול רק את השורות שבהן העמודה "תיאור" מכילה את המילה 'סמארטפון'.

התוצאה תהיה:



כפי שניתן לראות, הפונקציה INSTR() עוזרת לזהות את המוצרים שהמילה 'סמארטפון' מוזכרת בתיאור שלהם. זה מאפשר לנו לסנן את ערכת התוצאות ולאחזר את המוצרים הרצויים על סמך נוכחות של תת מחרוזת ספציפית בתוך עמודת המחרוזת.