# COALESCE

**הפונקציה COALESCE** מחזירה את הביטוי הראשון שאינו שווה ל- NULL מתוך הפרמטרים שקיבלה. היא שימושית בעיקר להתמודדות עם ערכי NULL בטבלת התוצאה.

**תחביר הפונקציה:**

COALESCE(expression1, expression2, ..., expressionN)

* הפונקציה יכולה לקבל מספר לא מוגבל של פרמטרים.
* אם כל הביטויים שהיא קיבלה שווים NULL היא תחזיר NULL.

**איך היא עובדת?**

הפונקציה עוברת על כל אחד מהביטויים משמאל לימין עד הרגע שבו היא מוצאת את הביטוי הראשון שאינו שווה ל-NULL. ברגע שהיא מצאה את ביטוי זה היא מפסיקה לרוץ ולא עוברת על שאר הביטויים הנותרים.

דוגמה פשוטה לשימוש בפונקציה:

SELECT COALESCE(NULL, 5, 7);

במקרה זה הפלט יהיה:

5

דוגמה נוספת:

SELECT COALESCE (1, 2);

במקרה זה הפלט יהיה:

1

**איך היא יכולה לשמש אותנו?**

* נוכל להשתמש בה להגדרת ערך ברירת מחדל במקום ערכי ה-NULL.

נסתכל על הדוגמה הבאה:

SELECT item\_name

discount

FROM items

בשליפה זו עבור כל מוצר ישלף שם המוצר וההנחה שיש עליו באחוזים.

תוצאת השליפה תראה כך:

A white background with black text

Description automatically generated

נוכל להשתמש בפונקציה COALESCE על מנת להציג בטבלת התוצאה ערך אחר במקום NULL שמופיע בעמודת discount:

SELECT item\_name

COALESCE(discount, 0)

FROM items

כעת עבור מוצרים שבהם ערך ההנחה שווה NULL תוצאת הפונקציה תהיה 0 וזה יהיה הערך שיוצג בטבלת התוצאה.

A white grid with black text

Description automatically generated

**\*נשים לב** שערך ברירת המחדל שנגדיר צריך להיות מאותו סוג נתונים של העמודה שאותה אנחנו רוצים לשלוף, אחרת תעלה שגיאה.

במידה ונרצה להגדיר ערך ברירת מחדל לעמודה טקסטואלית כמו לדוגמה item\_name נוכל לעשות זאת בצורה הבאה:

SELECT COALESCE(item\_name, ‘NO\_VALUE’)

FROM items

במקרה זה במידה ויופיעו ערכי NULL בעמודת item\_name הם יוחלפו בערך ‘NO\_VALUE’ בטבלת התוצאה.

A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated

**\*נשים לב** כי הערך שיתקבל מפונקציית ה-COALESCE יחליף את ערך ה-NULL בטבלת התוצאה בלבד. ערך ברירת החדל מוגדר באופן זמני עבור השליפה ואינו משנה את הטבלאות המקוריות ב-DB.

* השימוש בפונקציה מאפשר לנו לבצע חישובים על עמודות מבלי שתוצאת החישוב תפגע מהופעתם של ערכי NULL.

נראה זאת בדוגמה הבאה:

SELECT item\_name,

price,

discount,

Price \* (1 - discount) AS total\_price

FROM items

בשליפה זו נחשב את המחיר הכולל של מוצר לאחר הנחה.

תוצאת השליפה תיראה כך:

A white sheet with black text

Description automatically generated

משום שערך ההנחה של חטיף ה-twix מוגדר כ-NULL תוצאת החישוב של המחיר הכולל מחושבת אוטומטית גם היא כ-NULL.

COALESCE מציעה לנו דרך להתמודדות עם בעיה זו.

SELECT item\_name

Price \* (1 - COALESCE(discount, 0)) AS total\_price

FROM items

כפי שמוצג בשליפה זו, נוכל להשתמש בפונקציה להגדרת ערכי ה-NULL בעמודת discount כ-0. כך יוכל להתבצע החישוב בעמודת total\_price מבלי שהוא יפגע ויחשב אוטמטית כ-NULL עבור רשומות שלא הוגדרה בהם הנחה.

כעמת טבלת התוצאה תיראה כך:

A white paper with black text

Description automatically generated

**CASE vs COALESCE**

נציין ששימוש בביטוי CASE גם מאפשר מענה דומה לאותן בעיות בדומה לCOALESCE- בשניהם נוכל להשתמש עבור הגדרת ערכי ברירת מחדל שיחליפו את ערכי ה-NULL. אין הבדל מהותי ביניהן גם מבחינת היעילות כך שתוכלו לבחור מה לשלב בשאילתה שלכם.

דוגמה לשימוש בCASE לצד שימוש ב-COALESCE למטרה זהה:

SELECT item\_name

COALESCE(discount, 0) AS using\_coalesce

CASE

WHEN discount IS NULL THEN 0

ELSE discount

END AS using\_case

FROM items

מה בכל זאת כן מבדיל ביניהם? הכתיבה של COALESCE היא משמעותית יותר קצרה וההבדל באורך הקוד ניכר בייחוד אם נרצה להציב מספר ביטויים בפונקציה.

**לסיכום**, פונקציית COALESCE מציעה לנו דרך להתמודדות עם ערכי NULL בטבלת התוצאה. נוכל באמצעותה להגדיר ערכי ברירת מחדל שיחליפו את ערכי ה-NULL ובכך לאפשר גם חישובים על עמודות אלו. נשים לב ששינויים אלה מופיעים בטבלת התוצאה בלבד ולא משנות את הערכים בטבלאות המקוריות בבסיס הנתונים.