**תת שליפה מקושרת - Correlated Subquery**

מעולה! אז עכשיו לאחר שלמדנו מהי תת שליפה ומהו CTE, נלמד מהי תת שליפה מקושרת.

**מהי תת שליפה מקושרת?**

כאשר ננסה להבין את הצורך מתוך שאלה ונזהה שעלינו להשתמש בתת שליפה, ייתכן שנחשוב על דרך שבה נצטרך להסתמך בתוך תת השליפה על ערכים מתוך השליפה החיצונית, בדומה לשפות קוד שבהן הקונספט אפשרי. והאם זה אפשרי ב- SQL? התשובה היא כמובן כן. וזוהי ההגדרה של תת שליפה מקושרת - תת שליפה שבה נתייחס דרך הערכים לשליפה הראשית, החיצונית אליה.

דוגמה לשליפה שבה נוכל להשתמש בתת שליפה מקושרת: שליפה שבה נרצה לקבל את כל המוצרים שמחירם מעל המחיר הממוצע בקטגוריה בה הם נמצאים.

במקרה כזה, ישנם מוצרים תחת קטגוריות שונות, ולכל קטגוריה יהיה ממוצע שונה משלה. לכן אנחנו נרצה להתייחס בתת שליפה בכל פעם לממוצע הקטגוריה הרלוונטי לכל מוצר – הקטגוריה שתחתיה המוצר מתויג. והשליפה תיראה כך:

SELECT currprod.product\_name, cat.category\_name, currprod.unit\_price

FROM products currprod

LEFT JOIN categories cat ON

cat.category\_id = currprod.category\_id

WHERE currprod.unit\_price > (SELECT AVG(prodavg.unit\_price)

FROM products prodavg

WHERE prodavg.category\_id = currprod.category\_id)

ORDER BY cat.category\_name;

שימו לב לתנאי שבשורה 7: תת השליפה שולפת רק מוצרים שמספר הקטגוריה שלהם זהה למספר הקטגוריה של המוצר **שכרגע** שולפים עבורו מידע, בשליפה הגדולה.

בשורה 2 הגדרנו שאת הנתונים נשלוף מהטבלה productsוקראנו לה בשם currprod. בשורה 7 אמרנו למסד הנתונים שייקח רק רשומות בהם מספר הקטגוריה שווה למספר הקטגוריה של המוצר שאת פרטיו אנחנו שולפים בשליפה הגדולה, ע"י ציון השם הזה.

כאשר השליפה רצה, לכל רשומה מהטבלה currprod נבדק מה ערכו של category\_id, לאחר מכן מתבצעת תת השליפה ואז מתבצעת השליפה עצמה עבור אותה רשומה.

כלומר, היא תישלף בכל פעם מחדש לכל רשומה, בניגוד לתת שליפה רגילה אשר מתבצעת **פעם אחת בלבד** לפני ביצוע השליפה החיצונית.

ברגע שמסד הנתונים מזהה שבתת השליפה משתתף ערך מטבלה חיצונית לתת השליפה, (כמו currprod) היא תת שליפה שנחשבת Correlated Subquery.

**יעילות תת השליפה המקושרת**

משום ששליפה מקושרת מבצעת את תת השליפה מחדש עבור כל רשומה, התהליך יכול לפגוע בביצועי השליפה. אם נחזור להסתכל על הדוגמה הקודמת, שבה חישבנו בעבור כל קטגוריית מוצר את המחיר הממוצע בה – יכולנו גם לפתור אותה ללא תת שליפה מקושרת.

איך? באמצעות יצירת CTE שמכיל טבלה עם רשומה לכל קטגוריה ולמחיר הממוצע בה, ואז לקשר לכל מוצר את המחיר הממוצע באמצעות עמודת מזהה קטגוריה.

למעשה, ברוב המקרים שבהם אנחנו נרצה או נחשוב לפתור שאלה עם תת שליפה מקושרת, נגלה שניתן לפתור אותה בדרכים אחרות. ולרוב, הן יהיו יעילות יותר ועדיפות יותר – אנחנו יודעים שתת שליפה מקושרת דורשת חישוב מחדש בעבור כל רשומה, אך מה קורה עם הטבלה שלנו מכילה כמות רבה של רשומות? אפילו טבלה בעלת 100 רשומות תדרוש ביצוע של 100 תתי שליפות, במקום יצירת CTE וחיבור JOIN פשוטים.

לכן, על אף שהיא יכולה להיות יעילה ושימושית אנחנו נעדיף להימנע ממנה לחלוטין.

**אז למה אנחנו לומדים מהי תת שליפה מקושרת?**

**מתי כן נחליט שנכון להשתמש בה על פני חלופות אחרות?**

על אף שהיא לרוב לא תהיה יעילה, חשוב ללמוד ולהכיר את סל הכלים שלנו ולדעת שהיא אופציה שקיימת. מלבד זאת, עדיין תיתכן אפשרות שתהיה שליפה שבה תת שליפה מקושרת תהיה יעילה, או אפילו הפתרון היחידי לבעיה שלנו, אך ככל הנראה רק במקרים חריגים וקיצוניים. אנחנו לא נתעסק בתתי שליפות מקושרות בקורס ואתם לא נדרשים לבצע אותן, אך אתם מוזמנים להתנסות לכתוב אחת כזאת, או לחקור עליהן עוד בעצמכם!