קורס SQL למתחילים

הרצאה 6 – תרגיל – פתב"ס

זקני השבט

כתבו שאילתה המחזירה את ה-Id וה- DisplayName של כל המשתמשים שזקנים בלפחות עשר שנים מממוצע הגיל של אנשים מאותו המקום ממנו הם מגיעים (ה- Location בטבלת Users). החזירו את ה- Id של המשתמש, המיקום ממנו הוא הגיע, הגיל והגיל הממוצע במקום.

פיתרון:

```
with cte as (
       SELECT Location, AverageAge = AVG(Age)
       FROM Users
       WHERE Location IS NOT NULL
       GROUP BY Location
)
SELECT Users.Id,
              Users.DisplayName,
              Users.Location,
              Users.Age,
              cte.AverageAge
FROM Users
JOIN cte ON cte.Location = Users.Location AND Users.Age >= cte.AverageAge + 10
```

תגים

חלק ראשון

כל שאלה ב- StackOverflow יכולה להיות מקושרת לתג אחד או יותר. שמות התגים שקיימים נמצאים בטבלת Tags. הקישור בין פוסט ל-tag נמצא בטבלת PostsToTags, כאשר כאמור post בודד יכול להיות מקושר למספר tags שונים.
כתבו שאילתה שמציגה את כל התגיות שיש להן מעל 25,000 שאלות. השליפה צריכה לכלול את מזהה התג, שם התג ומספר השאלות תחת אותו תג.
הרשימה צריכה להיות ממויינת לפי מס' השאלות שמשוייכות לתג.

פיתרון:

```
SELECT TagId,
              TagName,
              NumQuestions=COUNT(*)
FROM [SO-2016].[dbo].[PostsToTags]
JOIN Posts ON Posts.Id = PostsToTags.PostId AND Posts.PostTypeId=1
JOIN Tags ON Tags.Id = [PostsToTags].TagId
GROUP BY TagId, TagName
HAVING COUNT(*)>25000
ORDER BY NumQuestions DESC
```

נשים לב שבמקרה הזה יש "יתירות" ב- GROUP BY. לא היה נדרש שם גם ה- Id וגם השם, אולם אם הייתי שם רק את ה- Id לא הייתי יכול להביא את השם ישירות (אפשר היה עם JOIN לאחר מכן), ואם הייתי שם את השם לא הייתי יכול להביא את ה- Id. ניתן במקרים כאלה להוסיף key נוסף בשביל הנוחות, רק צריך לוודא שלא מגזימים ומוסיפים מיליון Keys שהם תוצאה של JOIN-ים שונים, רק כדי שאפשר יהיה להכניס אותם לשליפה (זה יכול להשפיע *במצבים מסוימים* לרעה על הביצועים, לעומת פשוט לעשות JOIN אח"כ).

חלק שני

לכל אחת מהתגיות הללו הציגו גם את כותרת, מספר הצפיות ותאריך השאלה הכי פופולרית (עם הכי הרבה Views) שנשאלה תחת אותו תג ואת כותרת, מספר הצפיות ותאריך השאלה הכי פחות פופולרית שנשאלה תחת אותו התג.
בנוסף, הוסיפו לכל תג את מספר הצפיות הכולל של שאלות שנשאלו תחת אותו תג.

פיתרון:

```
with cte as(
	SELECT TagId,
			TagName,
			NumQuestions=COUNT(*),
			TotalViews = SUM(Posts.ViewCount)
	FROM [SO-2016].[dbo].[PostsToTags]
	JOIN Posts ON Posts.Id = PostsToTags.PostId AND Posts.PostTypeId=1
	JOIN Tags ON Tags.Id = [PostsToTags].TagId
	GROUP BY TagId, TagName
	HAVING COUNT(*)>25000

)
SELECT cte.*,
		MostPopularQTitle = mostPopularQuestion.Title,
		MostPopularQCreationDate = mostPopularQuestion.CreationDate,
		MostPopularQViewCount = mostPopularQuestion.ViewCount,
		LessPopuarQTitle = lessPopularQuestion.Title,
		LessPopularQCreationDate = lessPopularQuestion.CreationDate,
		LessPopularQViewCount = lessPopularQuestion.ViewCount
FROM cte
OUTER APPLY	(
				SELECT TOP 1 Posts.Title, Posts.CreationDate, ViewCount
				FROM PostsToTags
				JOIN Posts ON Posts.Id = PostsToTags.PostId
				WHERE PostsToTags.TagId=cte.TagId AND Posts.PostTypeId=1
				ORDER BY Posts.ViewCount DESC
			) mostPopularQuestion
OUTER APPLY	(
				SELECT TOP 1 Posts.Title, Posts.CreationDate,ViewCount
				FROM PostsToTags
				JOIN Posts ON Posts.Id = PostsToTags.PostId
				WHERE PostsToTags.TagId=cte.TagId AND Posts.PostTypeId=1
				ORDER BY Posts.ViewCount ASC
			) lessPopularQuestion
ORDER BY NumQuestions DESC
```

הדיון ער

לכל פוסט ב-StackOverflow אפשר לפרסם תגובות. הטבלה שמכילה את ה- metadata של התגובות (ללא התוכן) היא טבלת Comments.

אנחנו מגדירים שאלה שאלה פוסט שהיה בו **דיון ער** אם לפחות 3 משתמשים שונים הגיבו בו לפחות 2 תגובות שונות כל אחד. כתבו שליפה המחזירה את כל השאלות שהיה בהן דיון ער, ממויינות לפי כמות התגובות מלמעלה למטה. עבור כל פוסט שאלה שלפו את כותרת השאלה, תאריך הפרסום, מספר התגובות הכולל, מספר המשתמשים שהשתתפו בדיון, תאריך התגובה הראשונה ותאריך התגובה האחרונה.

שימו לב – יש לנו **שני תנאים** כדי שדיון על שאלה ייחשב "דיון ער" אנחנו רוצים גם ש-3 משתמשים שונים ישתתפו בדיון, וגם שכל אחד מהנם בנפרד הגיב לפחות 2 תגובות שונות. אם יש שאלה שמשתמש אחד הגיב עליה 50 פעמים, ו-3 משתמשים שונים הגיבו עליה פעם אחת – היא לא צריכה להיכלל.

פיתרון:

השאלה הזאת קצת מאתגרת, אז נבין מראש מה השלבים שנעשה בה:

1. נעשה groping של הטבלה Comments לפי PostId ו-UserId. כלומר, עבור פוסט בודד יהיו לנו כמה שורות בשלב הזה – שורה עבור כל משתמש שהגיב לפוסט הנ"ל ונפלטר רק למשתמשים שהגיבו מעל 2 תגובות לאותו הפוסט
2. נשלוף מהסט שיצרנו בשלב 1 ונעשה grouping נוסף לפי הפוסט, כדי להביא רק פוסטים שמעל 3 משתמשים (שקיימו את התנאי הקודם, כלומר שהגיבו 2 תגובות ומעלה) הגיבו להם
3. בשלב הזה יש לנו רק את הפוסטים הרלוונטיים והנתונים שלהם – נצרף לזה את נתוני ה- post וזהו, סיימנו.

ממליץ בחום לעבור בזהירות על השליפה הזאת, ולהבין מה התפקיד של כל אחד מהשלבים.

```
with postUsersGrouping as (
	SELECT PostId,
			UserId,
			FirstUserCommentDate = MIN (Comments.CreationDate),
			LastUserCommentDate = MAX (Comments.CreationDate),
			NumComments = COUNT(*)
	FROM Comments
	GROUP BY PostId, UserId
	HAVING COUNT(*) >= 2
), postsGrouping as (
	SELECT PostId,
			NumUsers = COUNT(DISTINCT UserId),
			NumReplies = SUM(NumComments),
			FirstPostCommentDate = MIN(FirstUserCommentDate),
			LastPostCommentDate = MAX(LastUserCommentDate)
	FROM postUsersGrouping
	GROUP BY PostId
	HAVING COUNT(DISTINCT UserId) >=3
)

SELECT Posts.Id,
		Posts.Title,
		CreationDate,
		NumReplies,
		NumUsers,
		FirstPostCommentDate,
		LastPostCommentDate
FROM postsGrouping
JOIN Posts ON Posts.Id = postsGrouping.PostId AND PostTypeId=1
ORDER BY NumReplies DESC
```