קורס SQL למתחילים

הרצאה 8 – תרגיל

משתמשים ומיקומים

סעיף א'

כתבו שאילתה המביאה מהטבלה Users את כל השורות שהגיל **אינו** NULL וגם ה- Location אינו NULL וגם ה- Location אינו ריק וגם המשתמש פירסם לפחות שאלה אחת (החלק הזה לא מסתמך על מה שנלמד השיעור הזה, אלא תזכורת משיעורים קודמים).
עבור כל שורה (משתמש) כללו את עמודת ה- Id, DisplayName, Location, Age וכן עמודה של הגיל הממוצע, הגיל המקסימלי והגיל המינימלי של אנשים מאותו ה- Location.
את המידע של הגיל הממוצע, המקסימלי והמינימלי של האנשים הביאו **באמצעות Window Functions** (בפרט – לא באמצעות GROUP BY ולא באמצעות nested queries).

פיתרון:

```
SELECT Id,
              DisplayName,
              Location,
              Age,
              AvgAgeInLocation = AVG(Age)OVER(PARTITION BY Location),
              MaxAgeInLocation = MAX(Age)OVER(PARTITION BY Location),
              MinAgeInLocation = MIN(Age)OVER(PARTITION BY Location)
FROM Users
WHERE  Age IS NOT NULL AND
       Location IS NOT NULL AND Location <> ''
       AND EXISTS (SELECT NULL FROM Posts WHERE PostTypeId=1 AND OwnerUserId = Users.Id)
```

סעיף ב'

הוסיפו עמודה שכוללת גם את ה- Id וה- DisplayName של המשתמש הראשון שנרשם מכל Location.

פיתרון:

```
SELECT Id,
              DisplayName,
              Location,
              Age,
              AvgAgeInLocation = AVG(Age)OVER(PARTITION BY Location),
              MaxAgeInLocation = MAX(Age)OVER(PARTITION BY Location),
              MinAgeInLocation = MIN(Age)OVER(PARTITION BY Location),
              FirstUserIdFromLocation = FIRST_VALUE(Id)OVER(PARTITION BY Location ORDER BY CreationDate),
              FirstUserNameFromLocation = FIRST_VALUE(DisplayName)OVER(PARTITION BY Location ORDER BY CreationDate)
FROM Users
```

© כל הזכויות שמורות לשחר גבירץ.
להרצאות, תרגילים ופתרונות ניתן להיכנס לאתר הקורס.

```
WHERE  Age IS NOT NULL AND
       Location IS NOT NULL AND Location <> ''
       AND EXISTS (SELECT NULL FROM Posts WHERE PostTypeId=1 AND OwnerUserId = Users.Id)
```

סעיף ג'

החזירו רק שורות של משתמשים שמקיימים את התנאים הבאים:

1. במיקום שהם הזינו יש לפחות 100 משתמשים.
2. הגיל שלהם מבוגר בלפחות 10 שנים מהגיל הממוצע במיקום שבו הם גרים

גם את זה השיגו באמצעות Window Functions.

פיתרון:

```
with cte  as(
SELECT Id,
              DisplayName,
              Location,
              Age,
              AvgAgeInLocation = AVG(Age)OVER(PARTITION BY Location),
              MaxAgeInLocation = MAX(Age)OVER(PARTITION BY Location),
              MinAgeInLocation = MIN(Age)OVER(PARTITION BY Location),
              CountInLocation = COUNT(*)OVER(PARTITION BY Location),
              FirstUserIdFromLocation = FIRST_VALUE(Id)OVER(PARTITION BY Location ORDER BY CreationDate),
              FirstUserNameFromLocation = FIRST_VALUE(DisplayName)OVER(PARTITION BY Location ORDER BY CreationDate)
FROM Users
WHERE  Age IS NOT NULL AND
       Location IS NOT NULL AND Location <> ''
       AND EXISTS (SELECT NULL FROM Posts WHERE PostTypeId=1 AND OwnerUserId = Users.Id)
)
select *
from cte
where CountInLocation > 100 AND Age - AvgAgeInLocation>10
```

ניקוי אורוות

טבלת Badges מכילה מידע על Badges שחולקו למשתמשים שונים. כל פעם שמשתמש מקבל Badge, מתווספת שורה לטבלה הזאת. ה-DBA של StackOverflow רוצה לנקות ממנה ערכים מיותרים. מבחינתו, אם משתמש קיבל את אותו ה- Badge יותר מפעם אחת באותו החודש – אז צריך להשאיר רק את השורה האחרונה עבור החודש הזה בטבלה. שורות נוספות שמתארות Badge שניתן לאותו המשתמש באותו החודש – אפשר למחוק.
כתבו שאילתה שמציפה את כל הערכים שניתן למחוק מהטבלה.

פיתרון:

```
with cte as (
SELECT  Name,
              UserId,
```

© כל הזכויות שמורות לשחר גבירץ.
להרצאות, תרגילים ופתרונות ניתן להיכנס לאתר הקורס.

```
            RN=ROW_NUMBER()OVER(PARTITION BY Name, UserId, DATEPART(Year,Date), DATEPART(Month,Date) ORDER BY Date DESC)
  FROM [SO-2016].[dbo].[Badges]

)
select *
from cte
where RN>1
```

דו"ח Badges

על סמך אותה טבלת ה- Badges שהוזכרה בשאלה הקודמת, כתבו שאילתה המחזירה לכל צירוף של משתמש ושל שם של Badge שהוא אי פעם קיבל (מופיע בטבלה בתור Name) את הערכים הבאים:

1. שם ה- Badge
2. ה- Id של המשתמש
3. שם המשתמש
4. הפעם הראשונה שהמשתמש קיבל את ה- Badge (תאריך ושעה)
5. הפעם האחרונה שהמשתמש קיבל את ה- Badge (תאריך ושעה)
6. כמה פעמים המשתמש קיבל את ה- Badge הזה
7. ה- Id של המשתמש הראשון שקיבל את ה- Badge הזה אי פעם
8. שם המשתמש הראשון שקיבל את ה- Badge הזה אי פעם
9. מה ממוצע הפעמים שמשתמש באתר מקבל את ה- Badge הזה

החזירו רק שורות שבהם הערך של (6) גדול מהערך של (9).

<u>רמז:</u>

השאילתה הזאת משלבת (או יכולה לשלב, בחלק מהשיטות שבהן נכתבת) גם GROUP BY וגם Window Functions. ניתן לשים באותו ה- SELECT גם תוצאות של פונקציות אגרגציה של ה- GROUP BY <u>וגם</u> window function. הסיבה לכך פשוטה: ה- window function פועל על השורות אחרי שהוחזרו, ככה שהוא מסתכל למעשה על השורות שחזרו מה-GROUP BY.

למשל, בדוגמא הבאה:

```
SELECT Location,
            NumOfPeople=COUNT(*),
            MaximumNumOfPEopleInAnyLocation = MAX(COUNT(*)) OVER(PARTITION BY 1)
FROM Users
WHERE Location IS NOT NULL AND Location <> ''
GROUP BY Location
```

© כל הזכויות שמורות לשחר גבירץ.
להרצאות, תרגילים ופתרונות ניתן להיכנס לאתר הקורס.

השליפה שלנו מסתכלת בטבלה Users על כל המשתמשים שהזינו Location שאינו ריק, עושה GROUP BY לפי ה- Location ומחזירה שם של Location וכמה אנשים הזינו אותו. העמודה השלישית היא window function שמסתכל בפועל על כל התוצאות של ה-GROUP BY ואומר מה המס' הכי גדול של אנשים שהזינו אותו Location). שימו לב שבעמודה השלישית, הפונקציה MAX פועלת כ- window function על כל הערכים שחוזרים מהפונקציה COUNT (שפועלת פה כפונקציית אגרגציה על הערכים שחזרו מה-GROUP BY).

פיתרון:

```
with usersToBadges as (
       SELECT Name,
                     CurrentUserId = UserId,
                     FirstGot = MIN(Date),
                     LastGot = MAX(Date),
                     FirstEverUserId = FIRST_VALUE(UserId)OVER(PARTITION BY Name ORDER BY MIN(Date)),
                     CountUserHasThisBadge = COUNT(*),
                     AverageCountForUsersToHAveThisBadge = AVG(COUNT(*))OVER(PARTITION BY Name)
       FROM Badges
       GROUP BY Name,UserId
)
SELECT usersToBadges.*,
              CurrentUserName = cur.DisplayName,
              FirstUserName = f.DisplayName
FROM usersToBadges
JOIN Users cur ON cur.Id = CurrentUserId
JOIN Users f ON f.Id = FirstEverUserId
WHERE CountUserHasThisBadge > AverageCountForUsersToHAveThisBadge
```

© כל הזכויות שמורות לשחר גבירץ.
להרצאות, תרגילים ופתרונות ניתן להיכנס לאתר הקורס.