

תרגיל 4 : Join Algorithms

תאריך הגשה : 55 : 23, 15.05.22.

הוראות הגשה:

בתרגיל זה אתם נדרשים להגיש קובץ zip בודד שיכלול את הקבצים הבאים :

- ex4.pdf עם התשובות מפורטות לשאלות. יש לפרט חישובים לא רק תשובה סופית!
- README שמכיל שורה בודדת ובו ה-login של הסטודנט שמגיש את התרגיל. אם התרגיל מוגש בזוגות, על שורה זאת להכיל את שני ה-login מופרדים בפסיק.

שימו לב:

- נא לקרוא על הדרישות המנהליות של הקורס בלינק באתר הקורס כדי למלא את ההוראות להגשה של קבצים סרוקים!
- תרגיל מוקלד יזכה ב- 2 נקודות בונוס!

שאלה 1 (40 נקודות):

נתונים שני היחסים הבאים מתוך מסד נתונים של CSRankings (זהים ליחסים בתרגיל 2):

authors (name, conference, year, institution, count, adjustedcount)

conferences (conference, area, subarea)

נניח:

- השדות הנומריים adjustedcount, count, year תופסים כל אחד 4 בייט.
- השדות הטקסטואליים: name, conference, institution, area, subarea תופסים כל אחד 10 בייט.
- בטבלה authors יש 165,000 שורות.
- בטבלה conferences יש 20,000 שורות.
- גודל בלוק הוא 8192 בייט.
- גודל החוצץ (buffer) הוא 25 בלוקים.

נרצה לחשב עלות של צירוף (join) של הטבלאות authors \bowtie conferences.

1. מה תהיה עלות החישוב של הביטוי לפי כל אחד מהאלגוריתמים הבאים?
אם החישוב לא אפשרי, הסבירו למה.

א. Block-nested-loops?

כל שורה של הטבלה authors תופסת 42 בייט.

בכל בלוק נכנסות $195 = \lceil 8192/42 \rceil$ שורות.

הטבלה תופסת סה"כ $B(\text{authors}) = \lceil \frac{165000}{195} \rceil = 847$ בלוקים.

כל שורה של הטבלה conferences תופסת 30 בייט.

בכל בלוק נכנסות $273 = \lceil 8192/30 \rceil$ שורות.

הטבלה תופסת סה"כ $B(\text{conferences}) = \lceil \frac{20000}{273} \rceil = 74$ בלוקים.

$$B(\text{conferences}) + \left\lceil \frac{B(\text{conferences})}{M-2} \right\rceil \times B(\text{authors}) = 74 + \left\lceil \frac{74}{25-2} \right\rceil \times 874$$

$$\underline{\underline{= 3462 \text{ I/O}}}$$

ב. Hash-join ?

$$B(\text{authors}) = 847, B(\text{conferences}) = 74$$

נבדוק את התנאי עבור הטבלה הקטנה יותר:

$$\left\lceil \frac{B(\text{conferences})}{M-1} \right\rceil < M-1 \rightarrow \left\lceil \frac{74}{25-1} \right\rceil = 4 < 24$$

התנאי מתקיים ולכן נשתמש בנוסחה:

$$3B(\text{authors}) + 3B(\text{conferences}) = 3 \times 847 + 3 \times 74$$

$$\underline{\underline{= 2763 \text{ I/O}}}$$

ג. Sort-merge-join ?

$$B(\text{authors}) = 847, B(\text{conferences}) = 74$$

נבדוק את התנאי עבור הטבלה הגדולה יותר:

$$\left\lceil \frac{B(\text{authors})}{M} \right\rceil < M \rightarrow \left\lceil \frac{847}{25} \right\rceil = 34 \nless 25$$

התנאי לא מתקיים ולכן אי אפשר להשתמש בSMJ.

2. כעת הניחו שגודל החוצץ הוא 30, איך הייתה משתנה העלות שחישבתם בסעיף 1?

א. Block-nested-loops?

$$B(\text{authors}) = 847, B(\text{conferences}) = 74$$

החוצץ יותר גדול ולכן כמובן שאפשר לבצע BNL

$$B(\text{conferences}) + \left\lceil \frac{B(\text{conferences})}{M-2} \right\rceil \times B(\text{authors}) = 74 + \left\lceil \frac{74}{30-2} \right\rceil \times 847$$

$$= 2615 \text{ I/O}$$

ב. Hash-join?

$$B(\text{authors}) = 847, B(\text{conferences}) = 74$$

החוצץ גדול ולכן בוודאי שהתנאי עדיין מתקיים ולכן נשתמש בנוסחה:

$$3B(\text{authors}) + 3B(\text{conferences}) = 3 \times 847 + 3 \times 74$$

$$= 2763 \text{ I/O}$$

התוצאה לא השתנתה

ג. Sort-merge-join?

$$B(\text{authors}) = 847, B(\text{conferences}) = 74$$

נבדוק את התנאי עבור הטבלה הגדולה יותר:

$$\left\lceil \frac{B(\text{authors})}{M} \right\rceil < M \rightarrow \left\lceil \frac{847}{30} \right\rceil = 29 < 30$$

כעת נבדוק אם אפשר לבצע צירוף בשלב השני:

$$\left\lceil \frac{B(\text{authors})}{M} \right\rceil + \left\lceil \frac{B(\text{conferences})}{M} \right\rceil < M \rightarrow \left\lceil \frac{847}{30} \right\rceil + \left\lceil \frac{74}{30} \right\rceil = 32 > 30$$

התנאי לא מתקיים ולכן העלות היא :

$$5B(\text{authors}) + 5B(\text{conferences}) = 5 \times 847 + 5 \times 74$$

$$= 4605 \text{ I/O}$$

3. מה גודל החוצץ המינימלי הנדרש כדי שיהיה ניתן לחשב כל אחד מהאלגוריתמים?

א. Block-nested-loops? 3

ב. Hash-join? 11

צריך לקיים:

$$\left\lceil \frac{B(\text{conferences})}{M-1} \right\rceil < M-1 \rightarrow \left\lceil \frac{74}{M-1} \right\rceil < M-1 \rightarrow 74 < (M-1)^2$$

$$\left\lceil \frac{74}{11-1} \right\rceil = 8 < 11-1 \quad 11=M \text{ ניקח}$$

מכיוון ש conferences יותר קטן, זה מספיק .

ג. Sort-merge-join? 30

צריך לקיים:

$$\left\lceil \frac{B(\text{authors})}{M} \right\rceil < M \rightarrow \left\lceil \frac{847}{M} \right\rceil < M \rightarrow 847 < M^2$$

$$\left\lceil \frac{847}{30} \right\rceil = 29 < 30 \quad 30=M \text{ ניקח}$$

מכיוון ש conferences יותר קטן, זה בטוח מספיק גם עבור מיון של conferences .

ד. Sort-merge join בשימוש באופטימיזציה שמאפשרת חישוב יעיל יותר (הנמנעת ממיון מלא של היחסים)? **32**

צריך לקיים:

$$\left\lceil \frac{B(\text{authors})}{M} \right\rceil + \left\lceil \frac{B(\text{conferences})}{M} \right\rceil < M \rightarrow \left\lceil \frac{847}{M} \right\rceil + \left\lceil \frac{74}{M} \right\rceil < M$$

$$\left\lceil \frac{847}{32} \right\rceil + \left\lceil \frac{74}{32} \right\rceil = 30 < 32 \quad \mathbf{32=M}$$

ניקח **M=32**

שאלה 2 (25 נקודות):

רוצים לחשב את הביטוי $\sigma_{A < 25 \wedge B = 6} (R(A, C) \bowtie S(B, C))$.
 גודלי היחסים הם $B(R)=1500$, $B(S)=200$. בכל בלוק של R יש 60 שורות, ובכל בלוק של S יש 150 שורות.
 ליחס S יש שני אינדקסים עם עלות גישה זניחה: אחד על אטריבוט C ואחד על אטריבוט B . כמו כן, ידוע ש C הוא מפתח ביחס S , וכן $V(S, B)=250$, $V(R, C)=50$. בחוצץ (buffer) יש 10 בלוקים.

(הערה: הכוונה ב"עלות גישה זניחה" היא שעלות הגישה לאינדקס - הירידה בו וטיול על העלים - זניחה, ולכן עלות השימוש באינדקס הוא שליפה של בלוקים מהטבלה בלבד. זה מתאים מאד למקרה בו מסד הנתונים שומר את מבנה האינדקס בזיכרון המרכזי)

א. העריכו את גודל התוצאה בבלוקים של הביטוי $\sigma_{B=6} S(B, C)$

$$\begin{aligned} E_S &= \sigma_{B=6} S(B, C) \quad \text{נסמן} \\ T(S) &= 200 \times 150 = 30000 \quad \text{מספר השורות של } S \\ T(E_S) &= \frac{T(S)}{V(S, B)} = \frac{30000}{250} = 120 \quad B = 6 \quad \text{מספר השורות ב } S \text{ שבהן } B = 6 \\ B(E_S) &= \frac{120}{150} = 1 \quad \text{מספר הבלוקים בתוצאת הביטוי } \end{aligned}$$

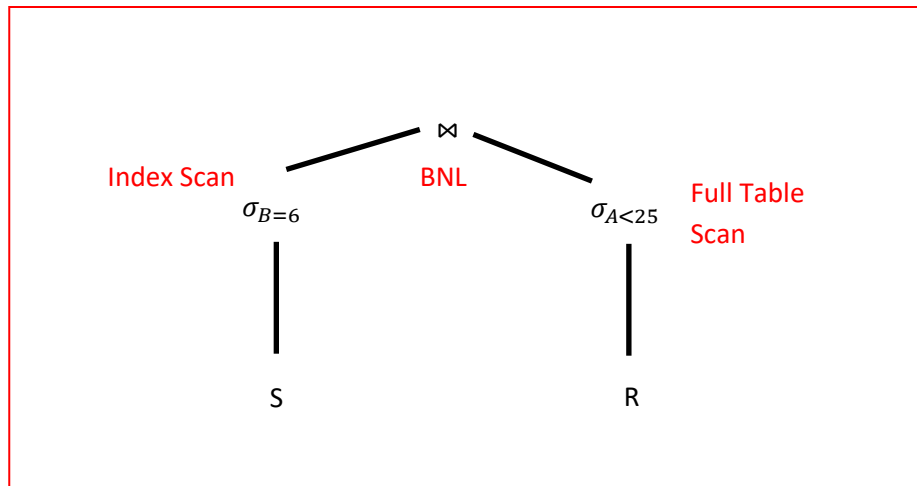
ב. העריכו את גודל התוצאה בבלוקים של הביטוי $\sigma_{A < 25} R(A, B)$

$$\begin{aligned} E_R &= \sigma_{A < 25} R(A, B) \quad \text{נסמן} \\ T(R) &= 1500 \times 60 = 90000 \quad \text{מספר השורות של } R \\ T(E_R) &= \frac{90000}{3} = 30000 \quad A < 25 \quad \text{מספר השורות ב } R \text{ שבהן } A < 25 \\ B(E_R) &= \frac{30000}{60} = 500 \quad \text{מספר הבלוקים בתוצאת הביטוי } \end{aligned}$$

ג. העריכו את מספר השורות בתוצאה של הביטוי כולו $\sigma_{A < 25 \wedge B = 6} (R(A, C) \bowtie S(B, C))$

$$\begin{aligned} V(S, C) &= T(S) = 30,000 \quad \text{ידוע ש } C \text{ הוא מפתח ב } S \text{ ולכן} \\ \frac{T(E_R) \times T(E_S)}{\max\{V(R, C), V(S, C)\}} &= \frac{30000 \times 120}{30000} = 120 \end{aligned}$$

ד. מהו האלגוריתם הכי יעיל לחישוב התוצאה? ציירו את עץ query plan.



ה. מה עלות החישוב היעיל ביותר?

נחשב את עלות הקריאה של E_S בעזרת האינדקס:

מכיוון שיש $T(E_S) = 120$ ועלות גישה של האינדקס זניחה, אז עבור כל שורה בתוצאת הבחירה נקרא לכל היותר בלוק 1.

עלות קריאה של הטבלה כולה זה $B(S) = 200$

ולכן עדיף השימוש באינדקס ויוצא $read(E_S) = 120$

לטבלה R אין אינדקס ולכן $read(E_R) = B(R) = 1500$

כעת נציב בנוסחה של BNL:

$$read(E_S) + \left\lceil \frac{B(E_S)}{M-2} \right\rceil \times read(E_R) = 120 + \left\lceil \frac{1}{10-2} \right\rceil \times 1500 = 1620 \text{ I/O}$$

שאלה 3 (20 נקודות):

רוצים לחשב את הביטוי $(R(A, B, C) \bowtie S(C, D)) \bowtie \pi_{A,D} \sigma_{B=15 \wedge D < 4}$. ההטלה היא ללא מחיקת כפילויות. גודלי היחסים הם $B(S)=3,000$, $B(R)=1,000$. גודל כל אחד מהאטריבוטים הוא 10 bytes וגודל בלוק הוא 1,500 bytes. אין אינדקסים ואסור לבנות אותם. כמו כן $V(S,C)=500$, $V(R,B)=5$ וידוע ש C הוא מפתח ביחס R. בחוצץ (buffer) יש 20 בלוקים.

א. מה יהיה מספר השורות בתוצאה?

כל שורה של R תופסת $30 = 3 * 10$ בייט, בב्लוק יש 1500 בייט, ולכן יש 50 שורות בב्लוק.

$$T(R) = 1000 \times 50 = 50000$$

כל שורה של S תופסת $20 = 2 * 10$ בייט, בב्लוק יש 1500 בייט, ולכן יש 75 שורות בב्लוק.

$$T(S) = 3000 \times 75 = 225000$$

בגלל ש C מפתח בר $T(R) = 50000$ $V(R, C)$

$$\frac{T(R) \times T(S)}{\max\{V(R, C), V(S, C)\} \times V(R, B) \times 3} = \frac{50000 \times 225000}{50000 \times 5 \times 3} = 15,000$$

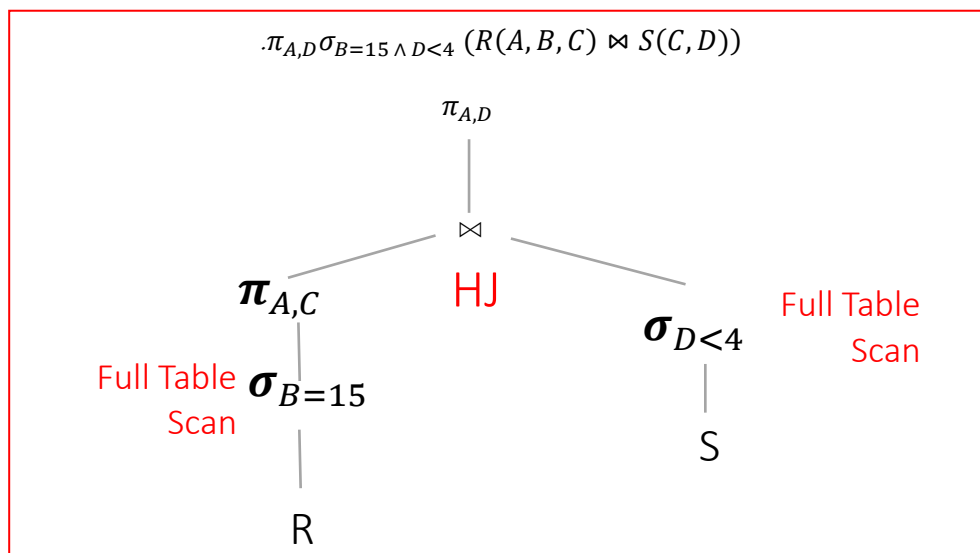
הנוסחה לגודל התוצאה: 15,000

ב. מה יהיה גודל התוצאה בב्लוקים?

כל שורה בתוצאה היא בגודל 20 בייט, כי יש הטלה. ולכן יש 75 שורות בב्लוק.

$$\left\lceil \frac{15,000}{75} \right\rceil = 200$$

ג. מהו האלגוריתם הכי יעיל לחישוב התוצאה? ציירו את עץ ה query plan.



ד. מה עלות החישוב היעיל ביותר?

נגדיר $E_R = \pi_{A,C} \sigma_{B=15} R(A, B, C)$, $E_S = \sigma_{D < 4} S(C, D)$

כעת, $Read(E_R) = B(R) = 1000$, $Read(E_S) = B(S) = 3000$

$$TSize(E_R) = \frac{T(R)}{V(R, B)} = \frac{50000}{5} = 10000, BSize(E_R) = \left\lceil \frac{10000}{75} \right\rceil = 134$$

$$TSize(E_S) = \frac{T(S)}{3} = \frac{225000}{3} = 75,000, BSize(E_S) = \frac{75000}{75} = 1000$$

נבדוק את התנאי של HJ עבור הטבלה הקטנה יותר:

$$\left\lceil \frac{BSize(E_R)}{M-1} \right\rceil < M-1 \rightarrow \left\lceil \frac{134}{20-1} \right\rceil = 8 < 20 -$$

התנאי מתקיים ולכן נשתמש בנוסחת העלות:

$$Read(E_R) + Read(E_S) + 2(BSize(E_R) + BSize(E_S)) =$$

$$1000 + 3000 + 2(134 + 1000) = 6268$$

שאלה 4 (15 נקודות):

מטרת שאלה זו היא התנסות עם כתיבה יעילה של שאילתות ושימוש באינדקס להתייעלות.

נתון היחס :

authors (name, conference, year, institution, count, adjustedcount)

ורוצים לחשב את השאילתה הבאה המוצאת עבור כל שנה שמופיעה בטבלה authors את השורה/שורות עם הערך adjustedcount הכי גבוהה.

```
select distinct *
from authors a1
where adjustedcount = (select max(adjustedcount)
                        from authors a2
                        where a2.year = a1.year);
```

לצורך מענה על הסעיפים הבאים, יש להשתמש בטבלה authors שהוגדרה בתרגילים קודמים.

(אם מחקתם כבר את הטבלה, בבקשה צרו אותה מחדש וטענו בנתונים לפי ההוראות בתרגיל 2.)

כעת ענו על השאלות הבאות:
הערה: כדי למדוד זמן ריצה של שאילתה, יש להריץ אותה עם פקודת *explain analyse* וזמן הריצה המבוקש הוא זמן התכנון + זמן הביצוע.

א. הריצו את השאילתה. כמה זמן לקח להריץ?
(אם הריצה הופסקה וקיבלתם הודעת *timeout*, זה בסדר, כתבו זאת בתשובה).
הריצו פקודת *explain*, שמראה את ה*query plan* של השאילתה וצרפו אותה לתשובות.
שימו לב שפקודת *'explain'* בשונה מפקודת *'explain analyse'* לא מריצה את השאילתה, רק מציגה את ה*query plan*.

הריצה הופסקה עם הודעת *timeout*

```
QUERY PLAN
-----
Unique  (cost=644479792.25..644479806.67 rows=806 width=57)
-> Sort  (cost=644479792.25..644479794.31 rows=824 width=57)
    Sort Key: al.name, al.institution, al.conference, al.count, al.adjustedcount, al.year
    -> Seq Scan on authors al  (cost=0.00..644479752.34 rows=824 width=57)
        Filter: (adjustedcount = (SubPlan 1))
        SubPlan 1
            -> Aggregate  (cost=3910.08..3910.09 rows=1 width=32)
                -> Seq Scan on authors a2  (cost=0.00..3902.30 rows=3110 width=5)
                    Filter: (year = al.year)

JIT:
  Functions: 10
  Options: Inlining true, Optimization true, Expressions true, Deforming true
(12 rows)
```

ב. נסו לשפר את זמן הריצה ע"י שינוי בתחביר השאילתה.
(ה*timeout* המוגדר במערכת הוא של 45 שניות, שאילתה משופרת אמורה לרוץ בהרבה פחות מזה)
כתבו את השאילתה החדשה, וכמה זמן לקח להריץ אותה.
הריצו את השאילתה עם פקודת *explain analyse*, שמראה את ה*query plan* של השאילתה החדשה, צרפו אותה לתשובות.
נסו לשער מה גרם לשיפור בזמן הריצה.

```
select distinct *
from authors a1
where (adjustedcount,year) in
      (select max(adjustedcount),year
       from authors a2
       group by year);
```

```

QUERY PLAN
-----
Unique (cost=8677.54..8683.84 rows=360 width=57) (actual time=410.073..410.279 rows=98 loops=1)
-> Sort (cost=8677.54..8678.44 rows=360 width=57) (actual time=410.070..410.133 rows=98 loops=1)
    Sort Key: a1.name, a1.institution, a1.conference, a1.count, a1.adjustedcount, a1.year
    Sort Method: quicksort Memory: 38kB
-> Hash Join (cost=4306.60..8662.25 rows=360 width=57) (actual time=167.885..409.780 rows=98 loops=1)
    Hash Cond: ((a1.adjustedcount = (max(a2.adjustedcount))) AND (a1.year = a2.year))
-> Seq Scan on authors a1 (cost=0.00..3490.24 rows=164824 width=57) (actual time=0.012..115.016 rows=164824 loops=1)
-> Hash (cost=4305.81..4305.81 rows=53 width=36) (actual time=164.076..164.077 rows=53 loops=1)
    Buckets: 1024 Batches: 1 Memory Usage: 11kB
-> Finalize GroupAggregate (cost=4298.39..4305.28 rows=53 width=36) (actual time=163.708..164.016 rows=53 loops=1)
    Group Key: a2.year
-> Gather Merge (cost=4298.39..4304.48 rows=53 width=36) (actual time=163.690..163.909 rows=106 loops=1)
    Workers Planned: 1
    Workers Launched: 1
-> Sort (cost=3298.38..3298.51 rows=53 width=36) (actual time=140.426..140.466 rows=53 loops=2)
    Sort Key: a2.year
    Sort Method: quicksort Memory: 27kB
    Worker 0: Sort Method: quicksort Memory: 27kB
-> Partial HashAggregate (cost=3296.33..3296.86 rows=53 width=36) (actual time=140.314..140.315 rows=53 loops=2)
    Group Key: a2.year
-> Parallel Seq Scan on authors a2 (cost=0.00..2811.55 rows=96955 width=9) (actual time=0.012..62.342 rows=82412 loops=2)
Planning Time: 0.215 ms
Execution Time: 410.456 ms

```

נראה שהשיפור בזמן ריצה נובע מכך שבשאלתה החדשה אין צורך לבצע את תת השאלתה עבור

כל שורה בטבלת *authors*, אלא מחושבת פעם אחת ואז נבדקת מול כל שורה ב*authors*.

ג. האם אפשר לשפר את זמן הריצה של השאלתה המקורית (לפני השינוי מסעיף ב') ע"י הוספת אינדקס?

בדקו אפשרויות שונות לאינדקס.

(שימוש באינדקס מתאים אמור לעזור לשאלתה לרוץ בהרבה פחות מ45 שניות)

כתבו איזה אפשרות של אינדקס שבניתם היה הכי יעיל,

כתבו את זמן הריצה החדש, הריצו את השאלתה עם פקודת *explain analyse*, שמראה את ה *query plan* של השאלתה, צרפו אותה לתשובות.

נסו להסביר את השינוי בזמן הריצה.

מידע שימושי:

- מומלץ לתת לאינדקסים שמות בזמן היצירה שלהם.
- פקודה פשוטה למחיקת אינדקס היא מהצורה: *drop index indexname*; כאשר *indexname* הוא שם האינדקס.
- כדי ראות רשימה של כל האינדקסים הקיימים יש להריץ את הפקודה: *\di*

האינדקס הכי יעיל הוא

create index on authors(year, adjustedcount);

```

QUERY PLAN
-----
Unique  (cost=437116.68..437131.10 rows=806 width=57) (actual time=2054.795..2055.017 rows=98 loops=1)
-> Sort (cost=437116.68..437118.74 rows=824 width=57) (actual time=2054.793..2054.856 rows=98 loops=1)
    Sort Key: al.name, al.institution, al.conference, al.count, al.adjustedcount, al.year
    Sort Method: quicksort  Memory: 38kB
    -> Seq Scan on authors al (cost=0.00..437076.78 rows=824 width=57) (actual time=36.503..2054.435 rows=98 loops=1)
        Filter: (adjustedcount = (SubPlan 2))
        Rows Removed by Filter: 164726
        SubPlan 2
            -> Result (cost=2.62..2.63 rows=1 width=32) (actual time=0.010..0.011 rows=1 loops=164824)
                InitPlan 1 (returns $1)
                    -> Limit (cost=0.42..2.62 rows=1 width=5) (actual time=0.007..0.008 rows=1 loops=164824)
                        -> Index Only Scan Backward using authors_year_adjustedcount_idx2 on authors a2 (cost=0.42..6836.52 rows=3110 width=5) (actual time=0.006..0.006 rows=1 loops=164824)
                            Index Cond: ((year = al.year) AND (adjustedcount IS NOT NULL))
                            Heap Fetches: 164824
Planning Time: 0.227 ms
JIT:
  Functions: 10
  Options: Inlining false, Optimization false, Expressions true, Deforming true
  Timing: Generation 1.529 ms, Inlining 0.000 ms, Optimization 0.313 ms, Emission 5.910 ms, Total 7.752 ms
Execution Time: 2056.878 ms

```

נראה שעכשיו תת השאילתה משתמש רק באינדקס ולא צריכה בכלל גישה לטבלה ולכן יש שיפור משמעותי בזמן הריצה.

בהצלחה!