מסד נתונים – תרגיל מס' 4 2/1/20

דין הסנאר,
ת"ז: 313584401

Login name: Hasenaar

שאלה 1:

א.) ללא אינדקס: עלינו לעבור על כל הטבלה Visit

כמו כן, $T(Visit) = 500{,}000$, וכן:

$$B(Visit) = \left\lceil \frac{500{,}000}{40} \right\rceil = 12{,}500$$

לפיכך, עלות השאילתא הנה: 12,500.

עם שימוש באינדקס: המעבר לאורך העץ יקח: $\log_{10} 500{,}000 = 6$. כמו כן, ערכי fee מתפלגים אחיד בטווח $[1, 1000]$, ולכן 1% מהשורות תהיינה עם ערך fee גדול ממש מ- 990. כלומר, קיימות 5,000 שורות כאלה, וכאשר נמצא אחד, נפסיק את החיפוש, שהרי המטרה שלנו הוא נכון exists.

לפיכך, עלות השאילתא הנה: $6 + 1 = 7$

ב.) ללא אינדקס: עלינו לעבור על כל הטבלה Visit, וחישבנו בסעיף הקודם כי:

$$B(Visit) = 12{,}500$$

לפיכך, עלות השאילתא הנה: 12,500

עם אינדקס: המעבר לאורך העץ יקח: $\log_{10} 500{,}000 = 6$. כמו כן, ערכי fee

שמפתח באופן אחיד בטווח [1,1000], ולכן
1% מהשורות הן עם ערך fee גדול ממש
מ-990, לכן, קיימות 5000
שורות מסוג זה.

בממוצע, ב-5000/10 = 500 בלוקים יש
לשמור 5000 שורות
כלומר, עלינו לעבור על 500 בלוקים
לפחות, כי נצטרך לקרוא את הבלוקים,
שכן אין עקביות לפיזור של המפתח בלבד.
לסיכום, עלות השאילתא הינה: 6+500 = **506**

ג) ללא אינדקס: עלינו לעבור על כל
הטבלה Visit. וכבר חישבנו בסעיף א'
כי: $B(visit) = 12,500$
לסיכום, עלות השאילתא הינה: **12,500**.

עם שימוש באינדקס: המעבר לאורך העץ
ייקח $\log_{10} 500,000 = 6$
כמו כן, ערכי fee שמפתח מתפזרים
בטווח [1,1000], לכן 1% מהשורות
הן עם ערך גדול ממש מ-990.
כלומר קיימות 5,000 שורות מסוג זה.

כמו כן, ניתן לשמור 5000 שורות —

ב- $\frac{5000}{10} = 500$ עמודים, וניתן לעבור ב-

500 עמודים בעלות, כמו כן, נצטרך לעבור

את כל הבלוקים, שכן ברנדומלי שמציר

שדה שאינו לא המפתח של visit

לסיכום, עלות השאילתא הינה:

$6 + 500 + 5000 = 5506$

ב') ללא שימוש באינדקס:

עלינו לעבור על כל בלוקי Patient

כמו כן: $T(Patient) = 40{,}000$, וכן:

$B(Visit) = \left\lceil \frac{40{,}000}{12} \right\rceil = 3334$

לסיכום, עלות השאילתא הינה 3,334.

עם שימוש באינדקס:

המעבר לאורך העץ ייקח $\log_{10} 40{,}000 = 5$

כמו כן, עלינו לקרוא את הבלוק,

שהרי אנו מעוניינים בשדה pname,

והוא לא במפתח של Patient

לסיכום, עלות השאילתא הינה: $5+1+1 = 7$

(מותר ב- pid מפתח, קיים עלה יחיד בלבד עם pid=11111,

הגעה עלה זה עלה 1 וקריאת pname עלה 1, בסך הכל 7).

עמ' 3

שאלה 2: 1) א') Block-nestet-loops:

תחילה נשים לב כי קיימות 2 מליון שורות בטבלה R, וכי בבלוק מכיל 100 שורות, ולכן:

$$B(R) = 2,000,000 / 100 = 20,000$$

באותו אופן, נחשב עבור הטבלה S ונקבל:

$$B(S) = \frac{5000}{20} = 250$$

כעת, נשים לב ש-S יותר קטנה מ-R, ולכן עדיף לשים אותה בלולאה החיצונית, ולפיכך עלות החישוב תהיה:

$$250 + 20,000 \cdot \left\lceil \frac{250}{100} \right\rceil = \boxed{60,250}$$

ב') sort-merge-join:

נשים לב כי התנאי למימוש האלגוריתם אינו מתקיים, שהרי:

$$\left\lceil \frac{B(R)}{M} \right\rceil = 197 > 102$$

ולא ניתן להשתמש בו.

~~ג') Hash-join: [illegible]~~

~~[illegible]~~

~~$\left\lceil \frac{B(R)}{M+1} \right\rceil = 197 > 102$ [illegible]~~

~~ולא ניתן להשתמש בו~~

ג.) Hash-join

נשים לב כי:

עלות הפעולה היא:

$$3B(R) + 3B(S) = 3 \cdot 20{,}000 + 3 \cdot 250 = \boxed{60{,}750}$$

הפעולה תתבצע אם:

$$\left\lceil \frac{B(R)}{M-1} \right\rceil \leq M-2 \quad \text{או} \quad \left\lceil \frac{B(S)}{M-1} \right\rceil \leq M-2$$

במקרה שלנו: ~~[illegible]~~

שקיימת $\left\lceil \frac{250}{100} \right\rceil = 3 \leq 102-2$ ולכן הפעולה תתבצע

ב) ii) Block-nested-loops:

שוב, נשים לב כי S קטנה יותר מ-R,
ולכן S נבחרת להיות בלולאה החיצונית.

עלות החישוב היא: $250 + 20{,}000 \cdot \left\lceil \frac{250}{300} \right\rceil = 20{,}250$.

ג) הפעם נראה שהתנאי של האלגוריתם
כן מתקיים, שהרי:

$$\left\lceil \frac{B(R)}{M} \right\rceil = \left\lceil \frac{20{,}000}{300} \right\rceil = 67$$

וכן:

$$\left\lceil \frac{B(S)}{M} \right\rceil = \left\lceil \frac{250}{300} \right\rceil = 1$$

ולכן:

$$\left\lceil \frac{B(R)}{M} \right\rceil + \left\lceil \frac{B(S)}{M} \right\rceil = 67 + 1 < 300 = M$$

ולכן החישוב אפשרי.

עלות החישוב הינה:

$$3B(R) + 3B(S) = 3 \cdot 20{,}000 + 3 \cdot 250 = 60{,}750$$

ג) החישוב אפשרי אם:

$$\left\lceil \frac{B(R)}{M-1} \right\rceil \leq M-2 \lor \left\lceil \frac{B(S)}{M-1} \right\rceil \leq M-2$$

במקרה שלנו מתקיים:

$$\left\lceil \frac{250}{299} \right\rceil = 1 \leq 300-2 = 298$$

ולכן החישוב אפשרי. עלות החישוב הינה:

6

עלות המשוך הינה:

$3B(R) + 3B(S) = 3 \cdot 20{,}000 + 3 \cdot 250 = \boxed{60{,}750}$

ג.) גודל החוצץ המינימלי:

ד.) **Block-nested-loops:**

נצטרך חוצץ בגודל 3 לפחות (אחד ללולאה החיצונית, אחד לפנימית, ואחד ל-output) כדי לחשב את זה.

ה.) מיון **Sort-merge-join**

עלינו למצוא את ה-X המינימלי שמקיים:

$$\left\lceil \frac{20{,}000}{X} \right\rceil + \left\lceil \frac{250}{X} \right\rceil < X$$

חישוב:

$$20{,}000 + 250 < X^2$$

$$142 < X$$

נבדוק עבור $X = 142$:

$$\left\lceil \frac{20000}{142} \right\rceil + \left\lceil \frac{250}{142} \right\rceil = 143 > 142$$

נבדוק עבור $X = 143$:

$$\left\lceil \frac{20000}{143} \right\rceil + \left\lceil \frac{250}{143} \right\rceil = 142 < 143$$

ולכן גודל החוצץ המינימלי הוא $\boxed{143}$

ג) (3 מיון Hash-join:

נרצה למצוא גודל מקסימלי x כך ש:

$$\left\lceil \frac{B(R)}{x-1} \right\rceil \le x-2 \quad \text{או} \quad \left\lceil \frac{B(S)}{x-1} \right\rceil \le x-2$$

עבור S, נקבל: $250 \le (x-2)(x-1)$

כלומר $x = 18$.

עבור R נקבל: $20000 \le (x-2)(x-1)$

כלומר $x = 143$.

נבדוק עבור $x = 18$: $\left\lceil \frac{250}{17} \right\rceil = 15 < 18-2$

לכן גודל החוצץ המינימלי הוא 18.

שאלה 3: הנוסחה, נשים לב שיש ב-S

כ- $300 \cdot 5 = 1{,}500$ שורות, וכן ב-R

יש $5000 \cdot 10 = 50{,}000$ שורות.

א) אין מספיק מידע, אבל נוכל להעריך את התוצאה כפי שלמדנו בהרצאה:

$$\frac{T(S)}{3} / 5 = 1500/15 = \boxed{100}$$

ב) נניח שקיימת התפלגות אחידה על הערכים, אבל נשלם כפי שלמדנו

עמ' 8

המשך

$$\frac{T(R)}{V(R,A)} / 10 = \frac{50,000}{1000} = \boxed{50}$$

ג) ראשית, נחשב את $V(R,B)$:

אם ידוע כי B הוא מפתח בטבלה R,

לכן שורה ב-R מקבלת ערך B ייחודי,

ולכן: $V(R,B) = 50,000$.

ולאחר הפעולה של האופרציה $\sigma_{A=11}$ נקבל:

$$V(\sigma_{A=11}R, B) = 500$$

וכן:

$$S(B,C) \bowtie R(A,B) = \frac{T(S)\cdot T(R)}{\max\{V(S,B), V(\sigma_{A=11}R, B)\}}$$

$$= \frac{500\cdot 500}{\max\{20, 500\}} = \boxed{500}$$

ד) ראשית נשים לב שעבור כל אלגוריתם,

עלינו לעשות קודם את התהליך ורק

לאחר מכן את המיון.

כעת, עבור הטבלה S, מכיוון שאין

אינדקס על C, עלינו לסרוק את הטבלה

תהיה כמספר הבלוקים בטבלה, הלא הם 300

ועבור הטבלה R, ידוע לנו שקיים אינדקס

ל-A שאינו גישה לשורה, ולכן

הספיקה תהיה בשאלה של מספר השורות
המתאימות לערכי — כמו 500.

<u>כמו כן, מתקיים:</u>

$$B(E_S) = \sigma_{C<3} S(B,C) = 100$$

$$B(E_R) = \sigma_{A=11} R(A,B) = 50$$

כמו כן, גודל החוצץ לא מאפשר שימוש
באלגוריתם merge-sort-join, שהרי

$$\left\lceil \frac{B(E_S)}{m} \right\rceil = 10 = m$$

כאן, נשתמש באלג' hash-join,
שעלותו היא:

$$read(E_S) + read(E_R) + 2B(E_R) + 2B(E_S) =$$

$$= 300 + 500 + 2 \cdot 50 + 2 \cdot 100 = \boxed{1100}$$

כאן, עבור אלג' INLJ, אין לנו
את R עבור הטבלה המשויכת,
שכן יש לך אינדקס על הטבלה B,
לכן, אין האלגוריתם הנ"ל?

עמ' 10

$$read(E_S) + T(E_S) \cdot (1) = 300 + 500 = 800$$

(שהרי עלות השימוש באינדקס נעלמה).

עבור אלג' BNLJ עם S בתור הטבלה החיצונית, העלות תהיה:

$$read(E_S) + read(E_R) \cdot \left\lceil \frac{B(E_S)}{m-2} \right\rceil =$$

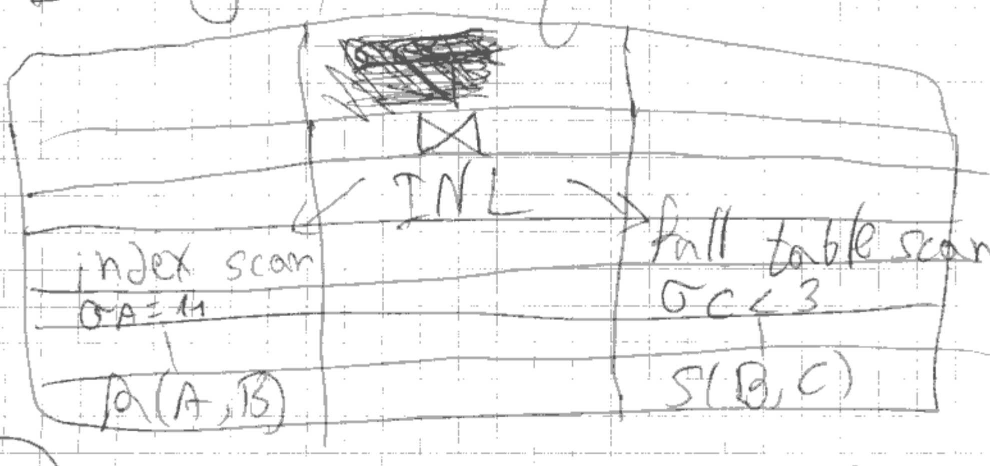
$$= 300 + 500 \cdot \left\lceil \frac{100}{8} \right\rceil = \boxed{6800}$$

עבור אלג' BNLJ עם R בתור הטבלה החיצונית, העלות תהיה:

$$read(E_R) + read(E_S) \cdot \left\lceil \frac{B(E_R)}{m-2} \right\rceil =$$

$$= 500 + 300 \cdot \left\lceil \frac{50}{8} \right\rceil = \boxed{2600}$$

לסיכום, נבחר באלגוריתם INLJ:



(ה) עלות החישוב הכולל במקרה הינו $\boxed{800}$

שאלה 4: רושמים משפט:

$$T(R)=B(R)\cdot\frac{3000}{30}=100{,}000$$

$$T(S)=B(S)\cdot\frac{3000}{20}=90\cdot150=13{,}500$$

כמו כן, B הוא מפתח של R,

ולכן: $V(R,B)=T(R)=100000$

א') (משפט)

$$T(R(A,B,C)\bowtie S(B,D))=\frac{T(S)\cdot T(R)}{\max\{V(S,B),V(R,B)\}}=$$

$$=\frac{13{,}500\cdot100000}{100{,}000}=13{,}500$$

כמו כן, ההטלה אינה מוחקת כפילויות, ולכן אינה משפיעה על התוצאה.

~~[illegible]~~ נשאר את זה

הבחירה שלנו קטנה ונחלק את התוצאה ב-3 לכל בחירה: אזי נקבל:
(כך הכלל של הבחירה)

$$T(\sigma_{A<10\wedge D<5}\pi_{A,D}R(A,B,C)\bowtie S(B,D))=$$

$$=\frac{T(R(A,B,C)\bowtie S(B,D))}{9}=\frac{13{,}500}{9}=\boxed{1500}$$

12 אל

ב.) המשך:

במצב הקודם חישבנו כי יהיו 1500 רשומות,

ולאחר שיבצע הטלה $\pi_{A,D}$ – "יוותרו רק שני"

ערכים, כך הרשומה תהיה מופחתת ל- 20 בתים

כעת, בבלוק יש $\frac{3000}{20} = 150$ שורות,

ולכן גודל התוצאה בבלוקים יהיה: $\frac{1500}{150} = \boxed{10}$.

ג.) נמצא את התוכנית היעילה ביותר לחישוב

התוצאה, ונשתמש בה אל ה-מולך query:

הכלל: נחשב את גודל התוצאות לאחר הפעולות

ונבחר באופטימלית (לפי התוכן):

נסמן:

$$E_R = \sigma_{A<10}\, \pi_{A,B}\, R(A,B,C)$$

$$E_S = \sigma_{D<5}\, S(B,D)$$

אזי:

$$T(E_R) = \frac{T(R)}{3} = 33334$$

$$B(E_R) = T(E_R) \cdot 2 \cdot \frac{10}{3000} = 33334 \cdot \frac{20}{3000} = 223$$

$$T(E_S) = \frac{T(S)}{3} = 4500$$

$$B(E_S) = T(E_S) \cdot 2 \cdot \frac{10}{3000} = 4500 \cdot \frac{20}{3000} = 30$$

13 גל

hash-join / sort-merge-join

$$HJ = SMJ = read(E_R) + read(E_S) + 2B(E_R) + 2B(E_S) =$$
$$= 1000 + 200 + 2 \cdot 30 + 2 \cdot 223 = \boxed{1596}$$

BNLJ, R

$$read(E_R) + read(E_S) \cdot \left\lceil \frac{B(E_R)}{M-2} \right\rceil = 2080$$

BNLJ, S

$$read(E_S) + read(E_R) \cdot \left\lceil \frac{B(E_S)}{M-2} \right\rceil = 2090$$

אלא, האלגוריתם היעיל ביותר הוא Hash-Join

נבדוק שאכן ניתן להשתמש באלגוריתם זה,

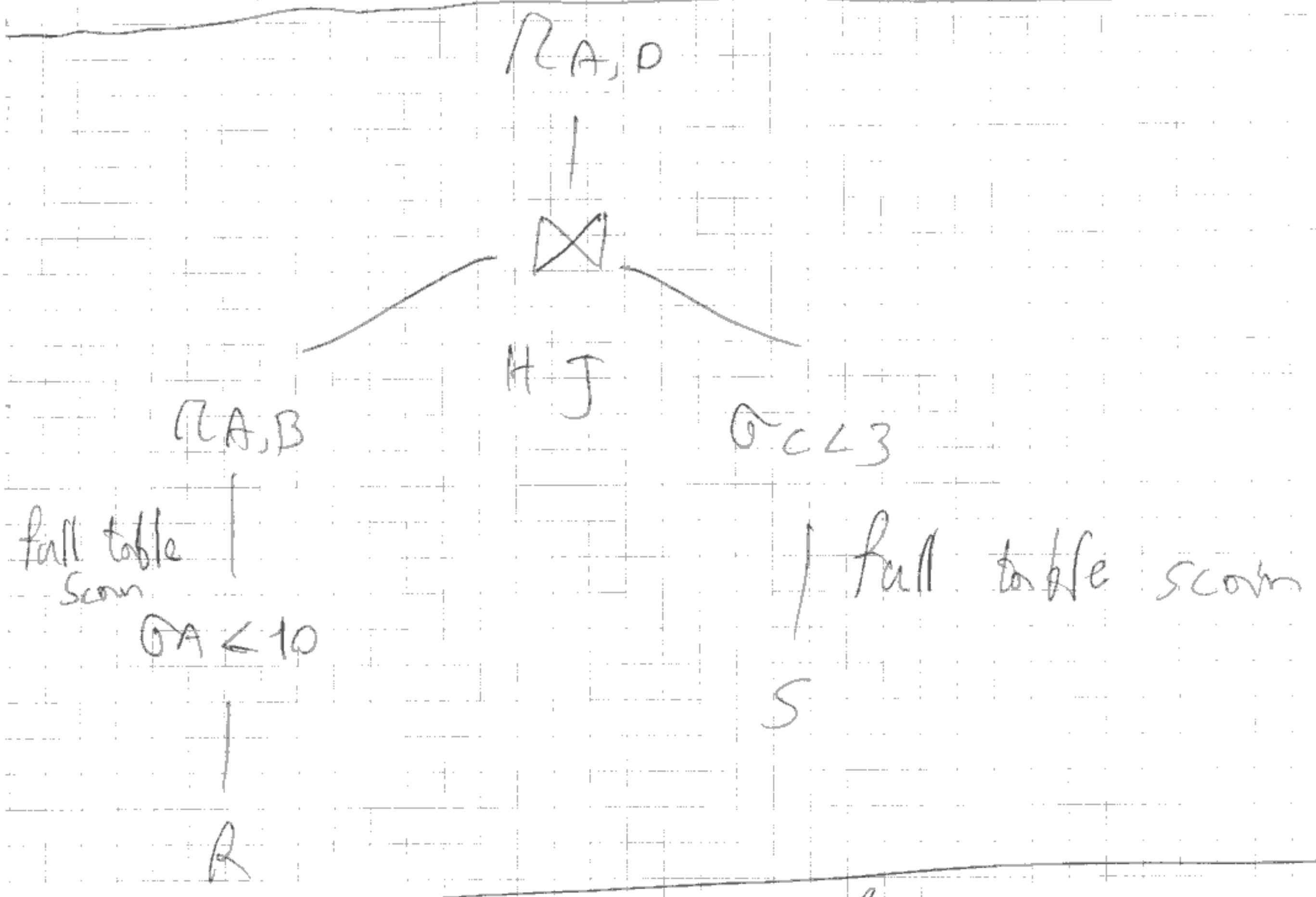
בהשתמש בתנאי הבא:

$$\left\lceil \frac{B(E_S)}{m-1} \right\rceil = \left\lceil \frac{30}{21} \right\rceil = 2 \leq 20 = m-2$$

ולכן ניתן להשתמש באלגוריתם HJ,

שהוא כאמור היעיל ביותר.

כעת, נצייר את עץ ה- query plan:



(ג) לפי העץ שקיבלנו, עלות השאילתה

היעיל ביותר הינה 1,596

שאלה 4: (ה) נתון כעת כי $B(S)=60$

אזי:

$$E_S = \sigma_{C<3} S(B,D)$$

$$Read(E_S) = B(S) = 60$$

$$B(E_S) = 20$$

עכשיו, כמו, INL לא נתון אינדקס, מכיוון שאין אינדקס והוא [illegible] האינדקס

בנוסף הפעם נתון [illegible] תהליך ~~[illegible]~~ SM:

$$\left\lceil \frac{B(E_R)}{m} \right\rceil + \left\lceil \frac{B(E_S)}{m} \right\rceil = 16+3 = 19 < 22 = m$$

כן.

בנוסף הפעם HJ נתון אפשרי:

$$\left\lceil \frac{B(E_S)}{m-1} \right\rceil = 3 \leq 20 = m-2$$

כן:

עלות המימוש שלהם (HJ/SM) תהיה:

$$Read(E_R) + Read(E_S) + 2\cdot(B(E_R)+B(E_S)) = \boxed{1546}$$

$(1000+60+2\cdot 243 = 1546)$

עלות המימוש של BNL תהיה:

$$read(E_S) + read(E_R)\cdot\left(\left\lceil \frac{B(E_S)}{m-2} \right\rceil\right) =$$

$$= 60 + 1000\cdot\left\lceil \frac{20}{20} \right\rceil = \boxed{1060}$$

אזי, האלגוריתם BNL הוא המהיר ביותר, עלותו $\boxed{1060}$

96 נק'

שאלה 5:

א.) לקח לי יותר מ-2 דקות להריץ את השאילתא.

הרצתי פקודת explain, מצ"ב תמונה בעמוד הבא.

```
                                 QUERY PLAN
-------------------------------------------------------------------------------------------
 HashAggregate  (cost=1104419037.50..1104419040.00 rows=250 width=81)
   Group Key: p1.id, p1.name, p1.phone_number, p1.city, p1.country, p1.job_title, p1.bdate
   ->  Seq Scan on people p1  (cost=0.00..1104419020.00 rows=1000 width=81)
         Filter: (bdate = (SubPlan 1))
         SubPlan 1
           ->  Aggregate  (cost=5522.06..5522.07 rows=1 width=4)
                 ->  Seq Scan on people p2  (cost=0.00..5520.00 rows=823 width=4)
                       Filter: ((country)::text = (p1.country)::text)
(8 rows)
```

ב') השאילתא הנדרשת:

```
SELECT distinct id, name, country,
    phone-number, job-title, city, bdate
FROM People natural join
  (SELECT country, min(bdate) as
      bdate-min from People
   group by country) as bdate-min-table
where bdate = bdate-min;
```

הרצתי את השאילתה עם פקודת
explain analyse (ה- query plan
מופיע בתמונה הבאה).

זמן הריצה: 1672.997 ms.

אני משער שהשיפור בזמן הריצה נובע
מכך שבשאילתה החדשה לא חישבנו את
המינימום כמות פעמים אשר תלוי
במספר השורות בטבלה People,
אלא הרבה פחות מכך.

```
                                                              QUERY PLAN
------------------------------------------------------------------------------------------------------------------------------------------
 HashAggregate  (cost=34230.47..34230.70 rows=23 width=81) (actual time=1672.325..1672.567 rows=246 loops=1)
   Group Key: people.id, people.name, people.country, people.phone_number, people.job_title, people.city, people.bdate
   ->  Hash Join  (cost=29356.00..34230.07 rows=23 width=81) (actual time=1590.636..1669.527 rows=1722 loops=1)
         Hash Cond: (((people_1.country)::text = (people.country)::text) AND ((min(people_1.bdate)) = people.bdate))
         ->  HashAggregate  (cost=10535.00..10537.43 rows=243 width=15) (actual time=694.479..694.708 rows=243 loops=1)
               Group Key: people_1.country
               ->  Seq Scan on people people_1  (cost=0.00..8785.00 rows=350000 width=15) (actual time=0.006..300.653 rows=350000 loops=1)
         ->  Hash  (cost=8785.00..8785.00 rows=350000 width=81) (actual time=895.867..895.867 rows=350000 loops=1)
               Buckets: 32768  Batches: 16  Memory Usage: 2798kB
               ->  Seq Scan on people  (cost=0.00..8785.00 rows=350000 width=81) (actual time=0.010..355.528 rows=350000 loops=1)
 Planning time: 0.777 ms
 Execution time: 1672.997 ms
(12 rows)
```

ג) ניתן לשפר את זמן הריצה ע"י
הוספת אינדקס.

נבדוק אינדקס לפי (country, bdate):
זמן הריצה הוא: 1489.523 מילישניות.

נבדוק אינדקס לפי (bdate, country) והפוך
זמן ריצה יותר ירוד: [illegible] ms
901.938
שלוש אחר, אינדקס לפי country בלבד
או bdate בלבד, נקבל זמן רב יותר.

נבחר כי האינדקס הטוב ביותר הינו:
(bdate, country)

אני מניח שהשיפור בזמן הריצה ע"י הוספת אינדקס
נובע מכך שכעת אין צורך לעבור על
כל הטבלה לצורך חישוב התנאים
על ה-bdate, אלא רק הקטע הרלוונטי
של ה-bdate הרלוונטי.
שהבדיקה ממוינת עליה תחילה
מכך השאלה מי יהיה המוקדם של ה-
bdate הרלוונטי, וכל זה לפי...
עם הנחה שכל התנאים מתבצעים בעצמם הבא.

ציון 19

```
                                                                 QUERY PLAN
-----------------------------------------------------------------------------------------------------------------------------------------------
 HashAggregate  (cost=15556.33..15556.62 rows=29 width=81) (actual time=901.416..901.637 rows=246 loops=1)
   Group Key: people.id, people.name, people.country, people.phone_number, people.job_title, people.city, people.bdate
   ->  Nested Loop  (cost=13545.42..15555.82 rows=29 width=81) (actual time=888.955..898.146 rows=2214 loops=1)
         ->  HashAggregate  (cost=13545.00..13547.43 rows=243 width=15) (actual time=888.911..889.132 rows=243 loops=1)
               Group Key: people_1.country
               ->  Seq Scan on people people_1  (cost=0.00..11295.00 rows=450000 width=15) (actual time=0.011..390.381 rows=450000 loops=1)
         ->  Index Scan using bdate_cntry_idx on people  (cost=0.42..8.24 rows=1 width=81) (actual time=0.005..0.020 rows=9 loops=243)
               Index Cond: ((bdate = (min(people_1.bdate))) AND ((country)::text = (people_1.country)::text))
 Planning time: 0.762 ms
 Execution time: 901.938 ms
(10 rows)

~
~
~
~
~
```