

## שאלה 1

נחשב תחילה כמה בלוקים כל טבלה תופסת:

כל שורה ב- $members$  תופסת  $59 \text{ bytes}$ .  $2 \cdot 4 + 5 \cdot 10 + 1 = 59$

כל בלוק יכול להכיל  $4 \text{ rows}$ .  $\lfloor \frac{256}{59} \rfloor = 4$

לכן הטבלה  $members$  תופסת סך הכל  $250 \text{ blocks}$ .  $B(members) = \lceil \frac{1000}{4} \rceil = 250$

כל שורה ב- $memberInKnesset$  תופסת  $18 \text{ bytes}$ .  $4 + 4 + 10 = 18$

כל בלוק יכול להכיל  $14$ .  $\lfloor \frac{256}{18} \rfloor = 14$

לכן הטבלה  $memberInKnesset$  תופסת סך הכל  $215 \text{ blocks}$ .  $B(memberInKnesset) = \lceil \frac{3000}{14} \rceil = 215$

1. א. נשתמש בנוסחה שלמדנו בשיעור עבור BNL פעמיים, כאשר פעם אחת  $memberInKnesset$  הוא היחס החיצוני ופעם אחת  $members$  הוא היחס החיצוני:

$$B(memberInKnesset) + B(members) \cdot \left\lceil \frac{B(memberInKnesset)}{M-2} \right\rceil =$$

$$215 + 250 \cdot \left\lceil \frac{215}{20-2} \right\rceil = 3215$$

$$B(members) + B(memberInKnesset) \cdot \left\lceil \frac{B(members)}{M-2} \right\rceil =$$

$$250 + 215 \cdot \left\lceil \frac{250}{20-2} \right\rceil = 3260$$

תשובה סופית: **3,215 פעולות I/O**.

ב. כפי שנאמר בשיעור, אנו מניחים שהנתונים מתפלגים באופן אחיד ופונקציית הגיבוב מפלגת את הערכים באופן אחיד.

נבדוק אם התנאי מתקיים עבור הטבלה הקטנה:

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

$$\left\lceil \frac{215}{19} \right\rceil \leq 18$$

$$12 \leq 18$$

קריאת היחסים ויצירת טבלאות הגיבוב יקחו  $2B(memberInKnesset) + 2B(members)$

קריאת הדליים ופעולות ה-join יקחו  $B(memberInKnesset) + B(members)$

לכן נוכל להשתמש בנוסחה שלמדנו בשיעור עבור HJ:

$$3B(memberInKnesset) + 3B(members) =$$

$$3 \cdot 215 + 3 \cdot 250 = 1,395$$

תשובה סופית: **1,395 פעולות I/O**.

ג. נרצה לבדוק תחילה אם ניתן לבצע מיון לטבלאות באמצעות 2-phase external merge-sort,

לכן נבדוק את התנאי על הטבלה הגדולה יותר:

$$\left\lceil \frac{B(members)}{M} \right\rceil \leq M-1$$

$$\left\lceil \frac{250}{20} \right\rceil \leq 19$$

$$13 \leq 19$$

מאחר והתנאי מתקיים על הטבלה הגדולה, הוא בהכרח מתקיים גם על הטבלה הקטנה ולכן ניתן למיין את שתיהן.

עלות יצירת הרצפים הממוינים עבור הטבלאות היא  $2B(memberInKnesset) + 2B(members)$ ,

לכל טבלה בנפרד צריך לקרוא לזכרון את הבלוק הראשון מכל רצף ממוין ולכתוב את הטבלה הממוינת

לדיסק וזה יקח  $2B(memberInKnesset) + 2B(members)$ ,

לבסוף נבצע join בעלות  $B(memberInKnesset) + B(members)$ , וסה"כ נקבל:

$$5B(memberInKnesset) + 5B(members) =$$

$$5 \cdot 215 + 5 \cdot 250 = 2325$$

נבדוק אם אפשר לשפר את עלות החישוב, כלומר אם אפשר לבצע צירוף בזמן המיזוג:

$$\left\lceil \frac{B(members)}{M} \right\rceil + \left\lceil \frac{B(memberInKnesset)}{M} \right\rceil \leq M - 1$$

$$\left\lceil \frac{250}{20} \right\rceil + \left\lceil \frac{215}{20} \right\rceil \leq 19$$

$$13 + 11 \leq 19 \Rightarrow 24 \not\leq 19$$

לא ניתן לשפר את עלות החישוב והתשובה הסופית היא: **פעולות 2,325 I/O**.

**2. א.** נחזור על החישובים עבור גודל חוצץ של 30:

$$B(memberInKnesset) + B(members) \cdot \left\lceil \frac{B(memberInKnesset)}{M-2} \right\rceil =$$

$$215 + 250 \cdot \left\lceil \frac{215}{30-2} \right\rceil = 2215$$

$$B(members) + B(memberInKnesset) \cdot \left\lceil \frac{B(members)}{M-2} \right\rceil =$$

$$250 + 215 \cdot \left\lceil \frac{250}{30-2} \right\rceil = 2185$$

תשובה סופית: **פעולות 2,185 I/O**.

**ב.** התנאי התקיים לפני כן עבור גודל חוצץ קטן יותר, לכן הוא בהכרח יתקיים גם עכשיו:

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

$$\left\lceil \frac{215}{29} \right\rceil \leq 28$$

$$8 \leq 28$$

עלות החישוב לא תלויה בגודל החוצץ לכן היא לא תשתנה:

$$3B(memberInKnesset) + 3B(members) =$$

$$3 \cdot 215 + 3 \cdot 250 = 1,395$$

תשובה סופית: **פעולות 1,395 I/O**.

**ג.** התנאי על הטבלה הגדולה יותר התקיים לפני כן, לכן הוא בהכרח יתקיים גם עכשיו:

$$\left\lceil \frac{B(members)}{M} \right\rceil \leq M - 1$$

$$\left\lceil \frac{250}{30} \right\rceil \leq 29$$

$$9 \leq 29$$

לכן עדיין ניתן למיין את שתי הטבלאות ולהשתמש בנוסחה שלמדנו בשיעור עבור SMJ:

$$5B(memberInKnesset) + 5B(members) =$$

$$5 \cdot 215 + 5 \cdot 250 = 2325$$

נבדוק שוב אם אפשר לשפר את עלות החישוב:

$$\left\lceil \frac{B(members)}{M} \right\rceil + \left\lceil \frac{B(memberInKnesset)}{M} \right\rceil \leq M - 1$$

$$\left\lceil \frac{250}{30} \right\rceil + \left\lceil \frac{215}{30} \right\rceil \leq 29$$

$$9 + 8 \leq 29 \Rightarrow 17 \leq 29$$

לכן אפשר לקרוא לזכרון את הבלוק הראשון מכל רצף ממוין כדי לבצע join בעלות

$$B(memberInKnesset) + B(members)$$

עלות יצירת הרצפים הממוינים עבור הטבלאות היא עדיין  $2B(memberInKnesset) + 2B(members)$  וסה"כ נקבל:

$$3B(memberInKnesset) + 3B(members) =$$

$$3 \cdot 215 + 3 \cdot 250 = 1395$$

תשובה סופית: **פעולות 1,395 I/O**.

**3. א.** צריך בלוק אחד עבור members, בלוק אחד עבור memberInKnesset ובלוק אחד עבור ה-output, כדי שיהיה ניתן להפעיל BNL עם גודל חוצץ מינימלי.  
תשובה סופית: **3 בלוקים**.

ב.

$$\begin{aligned} \frac{B(memberInKnesset)}{\frac{M-1}{215}} &\leq M-2 \\ \frac{215}{M-1} &< M-1 \\ 215 &< (M-1)^2 \end{aligned}$$

עבור  $M = 16$  נקבל גודל חוצץ מינימלי שמקיים את אי השוויון.  
נבדוק אם זה מקיים את התנאי:

$$\left\lceil \frac{215}{16-1} \right\rceil \leq 16-2 \iff 15 \not\leq 14$$

התנאי לא מתקיים לכן נבדוק את התנאי עבור  $M = 17$ :

$$\left\lceil \frac{215}{17-1} \right\rceil \leq 17-2 \iff 14 \leq 15$$

תשובה סופית: **17 בלוקים.**

ג.

$$\begin{aligned} \frac{B(members)}{M} &\leq M-1 \\ \frac{250}{M} < M &\iff 250 < M^2 \end{aligned}$$

עבור  $M = 16$  נקבל גודל חוצץ מינימלי שמקיים את אי השוויון.  
נבדוק אם זה מקיים את התנאי:

$$\left\lceil \frac{250}{16} \right\rceil \leq 16-1 \iff 16 \leq 15$$

התנאי לא מתקיים לכן נבדוק את התנאי עבור  $M = 17$ :

$$\left\lceil \frac{250}{17} \right\rceil \leq 17-1 \iff 15 \leq 16$$

תשובה סופית: **17 בלוקים.**

ד.

$$\begin{aligned} \frac{B(members)}{M} + \frac{B(memberInKnesset)}{M} &\leq M-1 \\ \frac{250}{M} + \frac{215}{M} &< M \\ 465 &< M^2 \end{aligned}$$

עבור  $M = 22$  נקבל גודל חוצץ מינימלי שמקיים את אי השוויון.  
נבדוק אם זה מקיים את התנאי:

$$\begin{aligned} \left\lceil \frac{B(members)}{M} \right\rceil + \left\lceil \frac{B(memberInKnesset)}{M} \right\rceil &\leq M-1 \\ \left\lceil \frac{250}{22} \right\rceil + \left\lceil \frac{215}{22} \right\rceil &\leq 21 \\ 12 + 10 &\leq 21 \iff 22 \not\leq 21 \end{aligned}$$

התנאי לא מתקיים לכן נבדוק את התנאי עבור  $M = 23$ :

$$\begin{aligned} \left\lceil \frac{250}{23} \right\rceil + \left\lceil \frac{215}{23} \right\rceil &\leq 22 \\ 11 + 10 &\leq 22 \iff 21 \leq 22 \end{aligned}$$

תשובה סופית: **23 בלוקים.**

## שאלה 2

**א.** אנו מעוניינים למצוא את  $B(E_S) = B(\sigma_{C < 45} S(A, C))$   
נחשב את מספר השורות של הטבלה S:

$$T(S) = B(S) \cdot 30 = 200 \cdot 30 = 6000$$

הטווח של C לא נתון לנו לכן נניח כי  $1/3$  מהשורות מקיימות  $C < 45$  ולכן:

$$T(E_S) = T(S) \cdot \frac{1}{3} = 6000 \cdot \frac{1}{3} = 2000$$

$$\downarrow$$

$$B(E_S) = \left\lceil \frac{2000}{30} \right\rceil = 67$$

תשובה סופית: **67 בלוקים.**

**ב.** אנו מעוניינים למצוא את  $B(E_R) = B(\sigma_{B=100} R(A, B))$   
נחשב את מספר השורות של הטבלה R:

$$T(R) = B(R) \cdot 150 = 2000 \cdot 150 = 300,000$$

נתון  $V(R, B) = 20$  כלומר כמות הערכים השונים בעמודה B היא 20,  
לכן מספר השורות בטבלה R שמקיימות  $B = 100$  הוא:

$$T(E_R) = T(R) \cdot \frac{1}{V(R, B)} = 300000 \cdot \frac{1}{20} = 15000$$

$$\downarrow$$

$$B(E_R) = \left\lceil \frac{15000}{150} \right\rceil = 100$$

תשובה סופית: **100 בלוקים.**

**ג.** נתון כי A הוא מפתח ביחס R, לכן  $V(R, A) = T(R) = 300,000$   
נעזר בסעיפים הקודמים ונקבל:

$$T(\sigma_{B=100 \wedge C < 45} (R(A, B) \bowtie S(A, C))) = \frac{T(E_R) \cdot T(E_S)}{\max(V(R, A), V(S, A))} = \frac{15000 \cdot 2000}{\max(300000, 1000)} = 100$$

תשובה סופית: **100 שורות.**

**ד.** נמצא את האופציה הזולה ביותר לביצוע selection:

- יחס R עם Full table scan

יעלה לנו  $B(R) = 2000$

- יחס R עם Index scan על B

נתון כי עלות הגישה זניחה לכן עלות השימוש באינדקס כולל רק שליפה של בלוקים מהטבלה,

והוא  $T(E_R) = 2000 \cdot 150 \cdot \frac{1}{20} = 15000$

- יחס S עם Full table scan

יעלה לנו  $B(S) = 200$

לסיכום, עבור שני היחסים נבצע Full table scan לביצוע selection ונקבל:

$$Read(E_R) = 2000$$

$$B(E_R) = 100$$

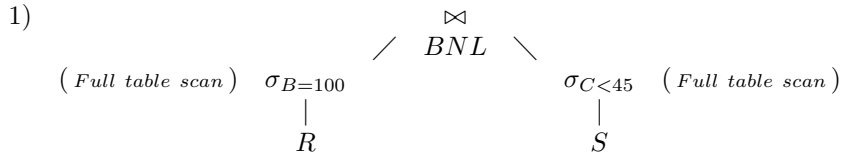
$$T(E_R) = 15000$$

$$Read(E_S) = 200$$

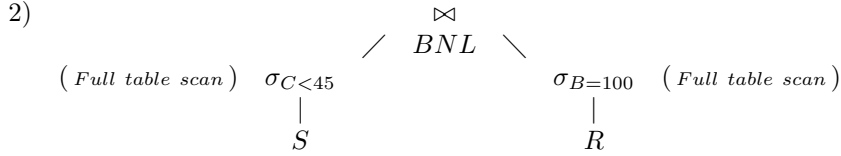
$$B(E_S) = 67$$

$$T(E_S) = 2000$$

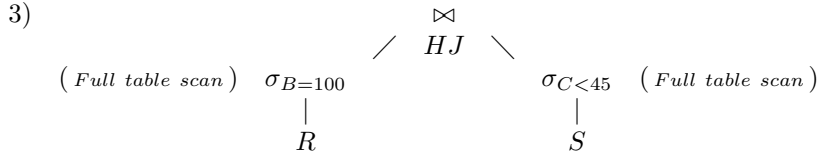
לפיכך נותרנו עם 5 query plans וכעת נמצא את האופציה הזולה ביותר לביצוע join:



$$\begin{aligned}
 cost1 &= Read(E_R) + Read(E_S) \cdot \left\lceil \frac{B(E_R)}{10-2} \right\rceil = \\
 &= 2000 + 200 \cdot \left\lceil \frac{100}{8} \right\rceil = 4600
 \end{aligned}$$



$$\begin{aligned}
 cost2 &= Read(E_S) + Read(E_R) \cdot \left\lceil \frac{B(E_S)}{10-2} \right\rceil = \\
 &= 200 + 2000 \cdot \left\lceil \frac{67}{8} \right\rceil = 18200
 \end{aligned}$$

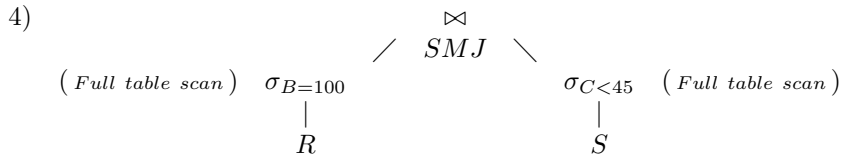


check if applicable:

$$\begin{aligned}
 \left\lceil \frac{B(E_S)}{M-1} \right\rceil &\leq M-2 \\
 \left\lceil \frac{67}{9} \right\rceil &\leq 8 \\
 8 &\leq 8
 \end{aligned}$$

then:

$$\begin{aligned}
 cost3 &= Read(E_R) + Read(E_S) + 2 \cdot [B(E_R) + B(E_S)] = \\
 &= 2000 + 200 + 2 \cdot (100 + 67) = 2534
 \end{aligned}$$



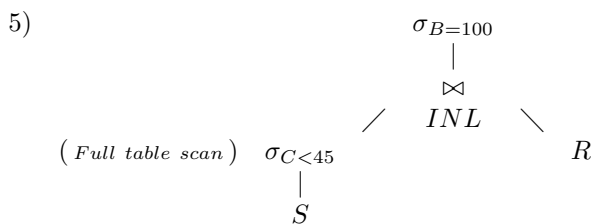
Check if applicable with optimization:

$$\begin{aligned}
 \left\lceil \frac{B(E_R)}{M} \right\rceil + \left\lceil \frac{B(E_S)}{M} \right\rceil &\leq M-1 \\
 \left\lceil \frac{100}{10} \right\rceil + \left\lceil \frac{67}{10} \right\rceil &\leq 9 \\
 17 &\not\leq 9
 \end{aligned}$$

Check if applicable without optimization:

$$\begin{aligned}
 \left\lceil \frac{B(E_R)}{M} \right\rceil &\leq M-1 \\
 \left\lceil \frac{100}{10} \right\rceil &\leq 9 \\
 10 &\not\leq 9
 \end{aligned}$$

Therefore, we can't use SMJ.



עלות האינדקס על R.A היא זניחה, וידוע ש A הוא מפתח ביחס R, ולכן עלות השימוש

באינדקס היא קריאה של בלוק אחד מהטבלה בלבד. כלומר

$$cost\_of\_index(R.A) = 1$$

$$\begin{aligned} cost5 &= Read(E_S) + T(E_S) \cdot costOfSelect = \\ &= 200 + 2000 \cdot 1 = 2200 \end{aligned}$$

תשובה סופית: האלגוריתם הכי יעיל לחישוב הוא Index Nested Loops, ועץ ה-query plan המתאים מופיע למעלה בחישוב 5.

ה. עלות החישוב היעיל ביותר הוא 2200 בלוקים.

### שאלה 3

נחשב את הגדלים הבאים שיעזרו בפתרונות הסעיפים הבאים:

\* גודל כל אחד מהתכונות A,B הוא 8 bytes וגודל בלוק הוא 2048 bytes,

לכן כמות השורות בבלוק ב-R היא  $\lfloor \frac{2048}{8} \rfloor = 128$ . לפיכך:

$$T(R) = 6000 \cdot 128 = 768000$$

$$E_R = \sigma_{A=30} R(A, B)$$

$$T(E_R) = T(R) \cdot \frac{1}{V(R,A)} = 768000 \cdot \frac{1}{100} = 7680$$

$$Read(E_R) = B(R) = 6000$$

גודל כל אחת מהתכונות A,B הוא 8 bytes לכן כמות השורות בכל בלוק

היא  $\lfloor \frac{2048}{8} \rfloor = 128$  ולפיכך:

$$B(E_R) = \lceil \frac{7680}{128} \rceil = 60$$

\* גודל כל אחד מהתכונות B,C,D הוא 8, 10, 10 bytes בהתאמה וגודל בלוק הוא 2048 bytes,

לכן כמות השורות בבלוק ב-S היא  $\lfloor \frac{2048}{28} \rfloor = 73$ . לפיכך:

$$T(S) = 3000 \cdot 73 = 219000$$

$$E_S = \pi_{B,D} \sigma_{D < 17} S(B, C, D)$$

$$T(E_S) = T(S) \cdot \frac{1}{3} = 219000 \cdot \frac{1}{3} = 73000$$

$$Read(E_S) = B(S) = 3000$$

גודל כל אחת מהתכונות B,D הוא 8,10 bytes בהתאמה לכן כמות השורות בכל בלוק

היא  $\lfloor \frac{2048}{18} \rfloor = 113$  ולפיכך:

$$B(E_S) = \lceil \frac{73000}{113} \rceil = 647$$

א. נתון כי B הוא מפתח ביחס S לכן  $V(S, B) = T(S) = 219000$  ובנוסף  $V(R, B) = 20$ .

ונתון כי ההטלה היא ללא מחיקת כפילויות, לכן היא לא משנה את מספר השורות.

נעזר בגדלים שחישבנו ונקבל:

$$\begin{aligned} T(\pi_{A,D} \sigma_{A=30 \wedge D < 17} (R(A, B) \bowtie S(B, C, D))) &= \\ T(\pi_{A,D} (\sigma_{A=30} R(A, B) \bowtie \pi_{B,D} \sigma_{D < 17} S(B, C, D))) &= \\ T(\pi_{A,D} (E_R \bowtie E_S)) &= \\ T(E_R) \cdot T(E_S) \cdot \frac{1}{\max(V(S,B), V(R,B))} &= \\ 7680 \cdot 73000 \cdot \frac{1}{219000} &= 2560 \end{aligned}$$

תשובה סופית: 2560 שורות.

ב. העמודות בתוצאה הן A,D שגודלן 8, 10 bytes בהתאמה,

לכן מספר השורות בבלוק בודד הוא  $\lfloor \frac{2048}{18} \rfloor = 113$ . לפיכך:

$$B(\pi_{A,D} \sigma_{A=30 \wedge D < 17} (R(A, B) \bowtie S(B, C, D))) = \lceil \frac{2560}{113} \rceil = 23$$

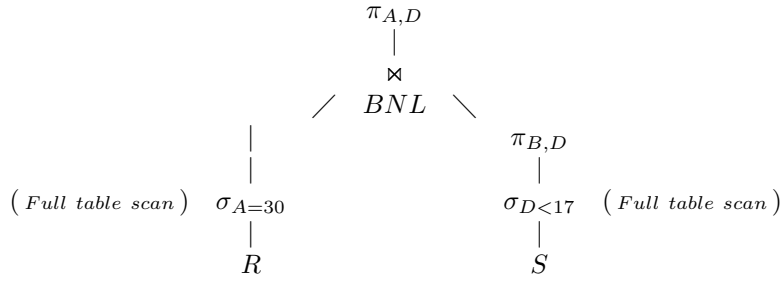
תשובה סופית: 23 בלוקים.

ג. נתון כי אין אינדקסים ואסור לבנות אותם, לכן נשתמש רק ב-Full table scan

בפעולות ה-selection. נותרנו עם 4 query plans ונמצא את האופציה הזולה

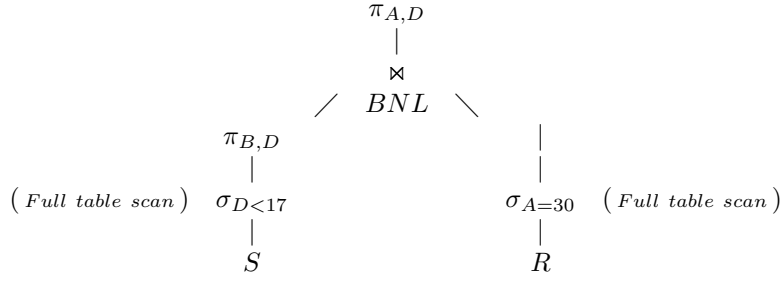
ביותר לביצוע join כאשר נעזר בגדלים שחישבנו כבר:

1)



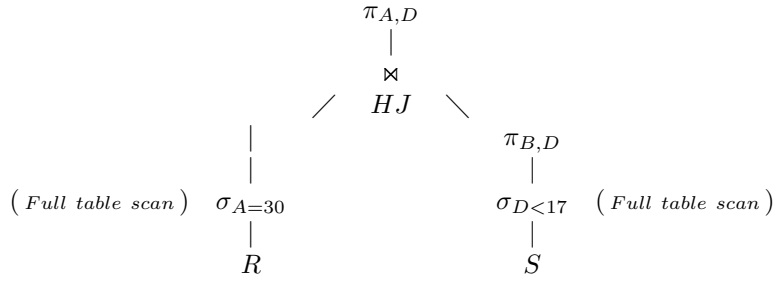
$$\begin{aligned} cost1 &= Read(E_R) + Read(E_S) \cdot \left\lceil \frac{B(E_R)}{50-2} \right\rceil = \\ &= 6000 + 3000 \cdot \left\lceil \frac{60}{48} \right\rceil = 12000 \end{aligned}$$

2)



$$\begin{aligned} cost2 &= Read(E_S) + Read(E_R) \cdot \left\lceil \frac{B(E_S)}{50-2} \right\rceil = \\ &= 3000 + 6000 \cdot \left\lceil \frac{647}{48} \right\rceil = 87000 \end{aligned}$$

3)



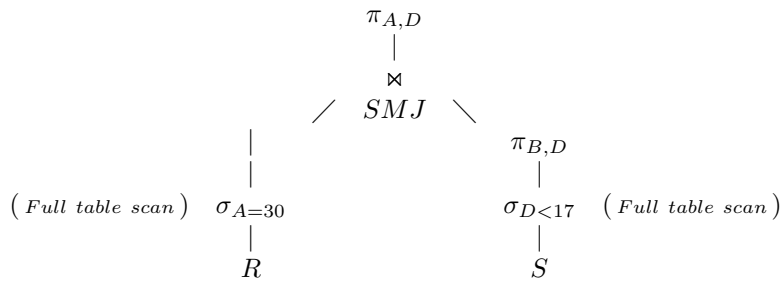
check if applicable:

$$\begin{aligned} \left\lceil \frac{B(E_R)}{M-1} \right\rceil &\leq M-2 \\ \left\lceil \frac{60}{49} \right\rceil &\leq 48 \\ 2 &\leq 48 \end{aligned}$$

then:

$$\begin{aligned} cost3 &= Read(E_R) + Read(E_S) + 2 \cdot [B(E_R) + B(E_S)] = \\ &= 6000 + 3000 + 2 \cdot (60 + 647) = 10414 \end{aligned}$$

4)



check if applicable with optimization:

$$\left\lceil \frac{B(E_R)}{M} \right\rceil + \left\lceil \frac{B(E_S)}{M} \right\rceil \leq M - 1$$

$$\left\lceil \frac{60}{50} \right\rceil + \left\lceil \frac{647}{50} \right\rceil \leq 49$$

$$2 + 13 \leq 49$$

then:

$$\begin{aligned} cost4 &= Read(E_R) + Read(E_S) + 2 \cdot [B(E_R) + B(E_S)] = \\ &= 6000 + 3000 + 2 \cdot (60 + 647) = 10414 \end{aligned}$$

תשובה סופית: קיימים 2 אלגוריתמים הכי יעילים לחישוב התוצאה  
הם - Hash Join OR Sort Merge Join, ועצי ה-query plan מופיעים בחישובים 3,4.

ד. עלות החישוב היעיל ביותר הוא 10,414 בלוקים.

## שאלה 4

א. 1. התשובה ב-sql-improved.

2.

```

QUERY PLAN
-----
Gather Merge (cost=11269.48..11270.42 rows=8 width=39) (actual time=321.000..322.194 rows=22 loops=1)
  Workers Planned: 2
  Workers Launched: 2
  → Sort (cost=10269.46..10269.47 rows=4 width=39) (actual time=317.297..317.303 rows=7 loops=3)
    Sort Key: enrollment.year, enrollment.eng_name
    Sort Method: quicksort Memory: 25kB
    Worker 0: Sort Method: quicksort Memory: 25kB
    Worker 1: Sort Method: quicksort Memory: 25kB
    → Hash Join (cost=5725.46..10269.42 rows=4 width=39) (actual time=251.668..317.234 rows=7 loops=3)
      Hash Cond: (((enrollment.year)::text = (enrollment_1.year)::text) AND ((enrollment.students5_estimated)::text = (max((enrollment_1.students5_estimated)::text))))
      → Parallel Seq Scan on enrollment (cost=0.00..4245.15 rows=56915 width=43) (actual time=0.005..33.505 rows=45532 loops=3)
      → Hash (cost=5725.24..5725.24 rows=15 width=37) (actual time=245.086..245.087 rows=15 loops=3)
        Buckets: 1024 Batches: 1 Memory Usage: 9kB
        → HashAggregate (cost=5724.94..5725.09 rows=15 width=37) (actual time=245.044..245.056 rows=15 loops=3)
          Group Key: enrollment_1.year
          → Seq Scan on enrollment enrollment_1 (cost=0.00..5041.96 rows=136596 width=9) (actual time=0.010..96.993 rows=136596 loops=3)
Planning Time: 0.454 ms
Execution Time: 322.329 ms
(18 rows)

```

זמן הריצה החדש הוא:

$$322.329 + 0.454 = 322.783 \text{ ms}$$

3. אני משער כי בשאלתה המקורית תת השאלתה מחושבת בכל פעם שעוברים על שורה  
enrollment e1 כדי לבצע selection. נשים לב שבשורת ה-WHERE מבצעים e1.year=e2.year  
ולכן תוצאת תת השאלתה משתנה בכל פעם וצריך לחשב אותה מחדש.  
לעומת זאת, בשאלתה החדשה תת השאלתה מחושבת פעם אחת ובאמצעותה בוחרים  
את ה-student5\_estimated המתאימים.

ב. 1. נכתוב את הפקודה לבניית האינדקס:

CREATE INDEX ind\_year\_students5 ON enrollment(year, students5\_estimated);

2.

```

QUERY PLAN
-----
Unique (cost=305522.21..305527.33 rows=669 width=39) (actual time=2547.757..2547.813 rows=22 loops=1)
  → Sort (cost=305522.21..305523.91 rows=683 width=39) (actual time=2547.755..2547.769 rows=22 loops=1)
    Sort Key: e1.year, e1.eng_name
    Sort Method: quicksort Memory: 27kB
    → Seq Scan on enrollment e1 (cost=0.00..305490.05 rows=683 width=39) (actual time=73.227..2547.630 rows=22 loops=1)
      Filter: ((students5_estimated)::text = (SubPlan 2))
      Rows Removed by Filter: 136574
      SubPlan 2
        → Result (cost=2.19..2.20 rows=1 width=32) (actual time=0.016..0.017 rows=1 loops=136596)
          InitPlan 1 (returns $1)
            → Limit (cost=0.42..2.19 rows=1 width=32) (actual time=0.014..0.014 rows=1 loops=136596)
              → Index Only Scan Backward using ind_year_students5 on enrollment e2 (cost=0.42..14999.04 rows=8488 width=32) (actual time=0.013..0.013 rows=1 loops=136596)
                Index Cond: ((year = (e1.year)::text) AND (students5_estimated IS NOT NULL))
                Heap Fetches: 136596
Planning Time: 0.511 ms
JIT:
  Functions: 12
  Options: Inlining false, Optimization false, Expressions true, Deforming true
  Timing: Generation 1.976 ms, Inlining 0.000 ms, Optimization 0.707 ms, Emission 13.727 ms, Total 16.409 ms
Execution Time: 2586.954 ms
(20 rows)

```



זמן הריצה החדש הוא:

$$2586.954 + 0.511 = \mathbf{2587.465\ ms}$$

3. בהתבסס על סעיף א, עיקר הבעיה היא שבתת השאילתה אנחנו ניגשים בכל פעם לטבלה e2 enrollment ובודקים עבור כל שורה אם `e2.year=e1.year`. לכן שימוש באינדקס על `year` יכול לעזור לנו מאוד. בנוסף, נשתמש גם באינדקס שני על `students5_estimated`, כדי שכל המידע שאנחנו צריכים בתת השאילתה יהיה נתון ב-BTree. בצורה כזאת תת השאילתה משתמשת רק באינדקסים ולא ניגשת לטבלה.