

שאלה 1

ראשית נבדוק את כמות הבלוקים בכל יחס:

members:

בשורה/tuple כמות הביטים הינה $2 * 4 + 5 * 10 + 1 = 59$.

$$B(\text{members}) = \left\lceil \frac{1000}{\left\lceil \frac{256}{59} \right\rceil} \right\rceil = \left\lceil \frac{1000}{4} \right\rceil = 250$$

לכן כמות הבלוקים תהיה:

memberInKneset:

בשורה/tuple כמות הביטים הינה $2 * 4 + 10 = 18$.

$$B(\text{memberInKneset}) = \left\lceil \frac{3000}{\left\lceil \frac{256}{18} \right\rceil} \right\rceil = \left\lceil \frac{3000}{14} \right\rceil = 215$$

לכן כמות הבלוקים תהיה:

א) memberInKneset הוא היחס הקטן מבין השניים, כלומר תופס פחות בלוקים, לכן נשים אותו בחיצוני ולכן עלות החישוב

$$B(\text{memberInKneset}) + B(\text{members}) * \left\lceil \frac{B(\text{memberInKneset})}{M - 2} \right\rceil$$

$$= 215 + 250 * \left\lceil \frac{215}{18} \right\rceil = 3215$$

ב) התנאי מתקיים שכן: $12 \leq 17 = M - 2$ $\left\lceil \frac{B(\text{memberInKneset})}{M-1} \right\rceil = \left\lceil \frac{215}{19} \right\rceil = 12$ לכן אפשרי.

$$3 * (B(\text{memberInKneset}) + B(\text{members})) = 3 * (215 + 250) = 1395$$

ג) אפשרי שכן $11 < 20 = M$ $\left\lceil \frac{B(\text{memberInKneset})}{M} \right\rceil = \left\lceil \frac{215}{20} \right\rceil = 11$ וגם $\left\lceil \frac{B(\text{members})}{M} \right\rceil = \left\lceil \frac{250}{20} \right\rceil = 13$ $11 + 13 = 23 > 20 = M$

עלות החישוב לפיכך תהיה:

$$5 * (B(\text{memberInKneset}) + B(\text{members})) = 2325$$

שאלה 2

א) כאשר memberInKneset החיצוני נקבל

$$B(\text{memberInKneset}) + B(\text{members}) * \left\lceil \frac{B(\text{memberInKneset})}{M - 2} \right\rceil$$

$$= 215 + 250 * \left\lceil \frac{215}{28} \right\rceil = 2215$$

אבל במקרה ההפוך:

$$B((\text{members})) + B(\text{memberInKneset}) * \left\lceil \frac{B((\text{members}))}{M - 2} \right\rceil$$

$$= 250 + 215 * \left\lceil \frac{250}{28} \right\rceil = 2185$$

שזה טוב יותר, לכן נבחר בריצה של האלגוריתם כאשר members הוא החיצוני והעלות תהיה 2185.

ב) התנאי עדיין מתקיים שכן: $8 \leq 28 = M - 2$ $\left\lceil \frac{B(\text{memberInKneset})}{M-1} \right\rceil = \left\lceil \frac{215}{29} \right\rceil = 8$ לכן אפשרי.

עלות החישוב תישאר זהה שכן לא תלויה בגודל הבאפר, כלומר **1395**.

ג) אפשרי שכן $M = 30 > 8 = \left\lceil \frac{B(\text{memberInKnesset})}{M} \right\rceil = \left\lceil \frac{215}{30} \right\rceil$ וגם $\left\lceil \frac{B(\text{members})}{M} \right\rceil = \left\lceil \frac{250}{30} \right\rceil = 9$.
 הגרסה האפשרית היא הקצרה שכן $M = 30 < 17 = 8 + 9$.
 עלות החישוב לפיכך תהיה:
 $1395 = (B(\text{memberInKnesset}) + B(\text{members})) * 3$.

שאלה 3

א) 3. אחד לקריאת בלוק מיחס חיצוני, אחד מיחס פנימי ואחד לכתיבת התוצאה.
 ב) נבדוק מינימלי שמקיים אחד מהמקרים:

$$\left\lceil \frac{B(\text{memberInKnesset})}{M-1} \right\rceil = \frac{215}{M-1} \leq M-2$$

נשים לב ש-17 המינימלית שמקיימת את המשוואה.
 עבור המקרה השני,

$$\left\lceil \frac{B(\text{members})}{M-1} \right\rceil = \frac{250}{M-1} \leq M-2$$

נשים לב ש-18 המינימלית שמקיימת את המשוואה. כיוון שמספיק שרק אחת מהמשוואות ממקיים כדי לאפשר אלגוריתם זה, אז 17 היא המינימלית.

ג) צריך שגם $\left\lceil \frac{B(\text{memberInKnesset})}{M} \right\rceil = \left\lceil \frac{215}{M} \right\rceil < M$ וגם $\left\lceil \frac{B(\text{members})}{M} \right\rceil = \left\lceil \frac{250}{M} \right\rceil < M$.
 ו- $M=17$ המינימלית שמקיימת זאת.

ד) צריך שיתקיים $\left\lceil \frac{215}{M} \right\rceil + \left\lceil \frac{250}{M} \right\rceil < M$ ואז בפרט גם התנאים בסעיף קודם יתקיימו שכן זה חיבור של מספרים אי שליליים. 23 היא המינימלית שמקיימת זאת.

חלק 2

א) אם מניחים שהבלוקים של S מלאים, אז $T(S) \approx B(S) * \text{rows_per_block} = 200 * 30 = 6000$.

אין מידע על טווח הערכים של עמודה C ב-S ולכן מניחים שמספר השורות שנקבל בהפעלת הביטוי הינו $\frac{T(S)}{3} = \frac{6000}{3} = 2000$. לכל בלוק נכנסים 30 שורות של S ולכן נעריך שגודל

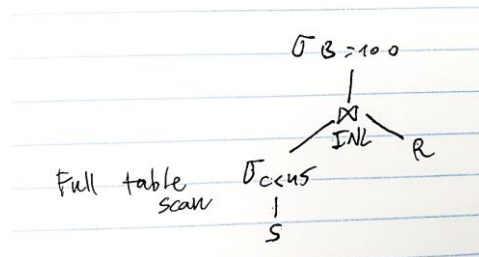
הפלט בבלוקים הינו $\left\lceil \frac{2000}{30} \right\rceil = 67$. לחלופין, אפשר פשוט לעשות $\left\lceil \frac{B(S)}{3} \right\rceil = \left\lceil \frac{200}{3} \right\rceil = 67$.

ב) אנו יודעים של-R יש 2000 בלוקים וכמות הערכים הייחודיים של שדה B הינו 20, לכן נקבל $\frac{B(R)}{V(R,B)} = \frac{2000}{20} = 100$.

ג) נפתור זאת בשלבים. ראשית, עבור המכפלה הקרטזית, נשים לב שכיוון ש A מפתח B אז יש $T(S) = 6000$ שורות פלט. לאחר מכן נכפיל ב- $\frac{1}{V(R,B)}$ בעבור התנאי R ו- $\frac{1}{3}$ בעבור

התנאי על S. סה"כ נקבל: $\frac{6000}{3 * V(R,B)} = 100$.

ד) INL



ה) $read(S_{\sigma_{C<45}}) + T(S_{\sigma_{C<45}}) * cost_{of_read}$ קריאת S והבחירה עם full table scan יעלה
 $T(S_{\sigma_{C<45}}) = \frac{B(S) * rows_per_block}{3} = \text{סך השורות} = read(S_{\sigma_{C<45}}) = B(S) = 200$
 $\frac{200 * 30}{3} = 2000$ נזכיר שחילקנו ב-3 כי אין מידע על שדה C ב-S. A הוא אינדקס ב-R וכן
עלות הגישה זניחה, לכן צריך לגשת רק לערך בודד, לכן $cost_of_read=1$ סה"כ קיבלנו
 $read(S_{\sigma_{C<45}}) + T(S_{\sigma_{C<45}}) * cost_{of_read} = 200 + 2000 * 1 = 2200$

חלק 3

א) במכפלה קרטזית, לכל שורה מ R תתאים רק אחת מ S שכן שדה צירוף B הוא מפתח ב S, לכן זה $T(R)$ שורות.

מתוכן, בהפעלת בחירה על A יש 100 ערכים ייחודיים, לכן יש $\frac{T(R)}{V(R,A)}$ שורות.

מתוכן, בהפעלת בחירה על D, אין לנו מידע על D ולכן מניחים שיש שליש, כלומר סה"כ

שורות $\frac{T(R)}{V(R,A) * 3}$

נמצא את $T(R)$:

$$\left\lceil \frac{T(R)}{\left\lfloor \frac{2048}{8+8} \right\rfloor} \right\rceil = 6000 = b(r)$$

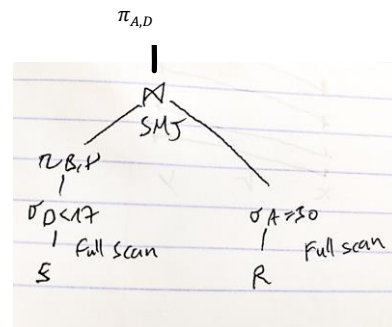
$$\left\lceil \frac{T(R)}{128} \right\rceil = 6000$$

במקרה הגרוע ביותר, $T(R) = 768000$. נציב ונחשב:

$$\frac{T(R)}{V(R,A) * 3} = \frac{768000}{100 * 3} = 2560$$

ב) לאחר הטלה נשארים רק עם עמודות A ו D שגודלן 18, לכן נקבל $\left\lceil \frac{2560}{\left\lfloor \frac{2048}{18} \right\rfloor} \right\rceil = \left\lceil \frac{2560}{114} \right\rceil = 23$

ג) SMJ Full scan בבחירה.



ד) לאחר מכן נקבל $9000 + 3000 = 12000$ של הטבלאות תעלה .

ולאחר ההטלה נקבל: D כי אין מידע על שדה 1000 $B(\sigma_{D<17}S) = \frac{3000}{3} =$

$$T(\pi_{B,D} \sigma_{D<17}S) = T(\sigma_{D<17}S) = 1000 * rowsPerBlock = 1000 * 73 = 73000$$

$$\left\lceil rowsPerBlock = \frac{bytesInBlock}{bytesPerRow} = \frac{2048}{28} = 73 \right\rceil$$

$$B(\pi_{B,D} \sigma_{D<17}S) = \left\lceil \frac{T(\pi_{B,D} \sigma_{D<17}S)}{\left\lfloor \frac{2048}{18} \right\rfloor} \right\rceil = \left\lceil \frac{73000}{113} \right\rceil = 647$$

$$B(\sigma_{A=30}R) = \left\lceil \frac{B(R)}{V(R,A)} \right\rceil = \left\lceil \frac{6000}{100} \right\rceil = 60.$$

וכן עבור $9000 + 2 * (647 + 60) = 10414$ סה"כ

```

QUERY PLAN
-----
Unique (cost=11293.63..11293.71 rows=10 width=39) (actual time=324.177..324.231 rows=22 loops=1)
  → Sort (cost=11293.63..11293.66 rows=10 width=39) (actual time=324.175..324.190 rows=22 loops=1)
    Sort Key: e1.year, e1.eng name
    Sort Method: quicksort Memory: 27kB
    → Hash Join (cost=5534.37..11293.47 rows=10 width=39) (actual time=117.059..324.096 rows=22 loops=1)
      Hash Cond: ((e1.year)::text = (enrollment.year)::text) AND ((e1.students5_estimated)::text = (max(enrollment.students5_estimated)::text)))
      → Seq Scan on enrollment e1 (cost=0.00..5041.96 rows=136596 width=43) (actual time=0.011..90.562 rows=136596 loops=1)
      → Hash (cost=5534.14..5534.14 rows=15 width=37) (actual time=112.196..112.196 rows=15 loops=1)
        Buckets: 1024 Batches: 1 Memory Usage: 9kB
        → Finalize GroupAggregate (cost=5530.19..5533.99 rows=15 width=37) (actual time=112.045..112.169 rows=15 loops=1)
          Group Key: enrollment.year
          → Gather Merge (cost=5530.19..5533.69 rows=30 width=37) (actual time=112.030..112.134 rows=45 loops=1)
            Workers Planned: 2
            Workers Launched: 2
            → Sort (cost=4530.17..4530.21 rows=15 width=37) (actual time=83.665..83.676 rows=15 loops=3)
              Sort Key: enrollment.year
              Sort Method: quicksort Memory: 25kB
              Worker 0: Sort Method: quicksort Memory: 25kB
              Worker 1: Sort Method: quicksort Memory: 25kB
              → Partial HashAggregate (cost=4529.73..4529.88 rows=15 width=37) (actual time=83.607..83.619 rows=15 loops=3)
                Group Key: enrollment.year
                → Parallel Seq Scan on enrollment (cost=0.00..4245.15 rows=56915 width=9) (actual time=0.025..33.855 rows=4553
2 loops=3)
Planning Time: 0.191 ms
Execution Time: 324.376 ms
(24 rows)

```

זמן הריצה החדש הינו $0.191 + 324.376 = 324.567 \text{ ms}$

(3) בקודמת הייתה תת שאילתה שחושבה לכל איטרציה על שורה מ-enrollment שמחשבת ערך מקסימלי של רשומים עבור השנה הנוכחית בעוד בשאילתה החדשה אין צורך בתת שאילתה, אלא מתחילים מליצור יחס עם with ששומר את הערך המקסימלי בכל שנה ככה שבשלב הבא יש פשוט hash join. בעצם במקורי היה בתת שאילתה כל פעם סריקה מחדש של הטבלה ובמימוש שלי רק פעם אחת.

שאלה ב

(1) CREATE INDEX on enrollment(year, students5_estimated);

(2) זמן הריצה החדש הינו $0.146 + 2553.409 = 2553.555 \text{ ms} = 2.55 \text{ seconds}$

```

QUERY PLAN
-----
Unique (cost=305697.24..305702.36 rows=668 width=39) (actual time=2551.617..2551.673 rows=22 loops=1)
  → Sort (cost=305697.24..305698.94 rows=683 width=39) (actual time=2551.615..2551.629 rows=22 loops=1)
    Sort Key: e1.year, e1.eng name
    Sort Method: quicksort Memory: 27kB
    → Seq Scan on enrollment e1 (cost=0.00..305665.08 rows=683 width=39) (actual time=69.993..2551.522 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 enrollment_year_students5_estimated_idx on enrollment e2 (cost=0.42..15004.61 rows=8485 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: 0.146 ms
JIT:
  Functions: 12
  Options: Inlining false, Optimization false, Expressions true, Deforming true
Timing: Generation 1.631 ms, Inlining 0.000 ms, Optimization 0.781 ms, Emission 11.769 ms, Total 14.181 ms
Execution Time: 2553.409 ms
(20 rows)

```

(3) הפעם אין סריקה מחדש של e2 enrollment אלא מבנה האינדקס מחפש את השורות בהן השנה זהה על ידי מבנה האינדקס. בלבד זאת, כיוון שהאינדקס גם על students5_estimated אז מספיק לו לגשת לעלים (index only scan) כדי לקבל את המידע, ללא קריאות נוספות מהדיסק. זה קיצר משמעותית את זמן הריצה שכן הפעם הוא כולל רק את הקריאה של הטבלה למען השאילתה החיצונית וכן קריאת העץ.