

Ex4

ליאור שפירא וכרמל גרוס

שאלה 1:

נניח:

-	השדות הנומריים moviedl, rating, year, duration, actorld, byear, dyear
-	תופסים 4 בייט
-	השדות הטקסטואליים character, name, genre, title
-	10 תופסים בייט
-	בטבלה Movies יש 10,000 שורות
-	בטבלה Actors יש 50,000 שורות
-	בטבלה PlaysIn יש 100,000 שורות
-	גודל בלוק הוא 8192 בייט.
-	גודל החוצץ הוא $M = 15$ בלוקים

נרצה לחשב את $Movies \bowtie Playsin$

1. ראשית נחשב את מספר הבלוקים שכל טבלה תופסת

$$Movies \rightarrow \frac{8192}{4 + 10 + 4 + 4 + 4 + 10} = \left\lfloor \frac{8192}{36} \right\rfloor = 227 \rightarrow \left\lceil \frac{10,000}{227} \right\rceil = 45$$

$$Playsin \rightarrow \frac{8192}{4 + 10 + 4} = \left\lfloor \frac{8192}{18} \right\rfloor = 455 \rightarrow \left\lceil \frac{10,000}{455} \right\rceil = 220$$

א. עבור block nested-loops נקבל ש-movies יהיה ביחס החיצוני כך שעלות החישוב הינה

$$B(movies) + B(Playsin) \cdot \left\lceil \frac{B(movies)}{M - 2} \right\rceil = 45 + 220 \cdot \frac{45}{13} = 925$$

ב. עבור hash join ראשית נחשב האם אפשר לבצע את האלגוריתם

$$\left\lceil \frac{B(movies)}{15 - 1} \right\rceil = \left\lceil \frac{45}{14} \right\rceil = 4 < 14 = M - 1$$

אזי אפשר ולכן נקבל שעלות החישוב הינה

$$3B(movies) + 3B(Playsin) = 3 \cdot 45 + 3 \cdot 220 = 795$$

ג. עבור sort-merge-join ראשית נחשב האם אפשר לבצע את האלגוריתם

$$\left\lceil \frac{B(Playsin)}{15} \right\rceil = \left\lceil \frac{220}{15} \right\rceil = 15 = M$$

אזי לא ניתן לבצע את האלגוריתם

2. בעת עבור $M = 16$

א. עבור block nested-loops נקבל ש-movies עדין יהיה ביחס החיצוני כך שעלות החישוב הינה

$$B(movies) + B(Playsin) \cdot \left\lceil \frac{B(movies)}{16 - 2} \right\rceil = 45 + 220 \cdot \left\lceil \frac{45}{14} \right\rceil = \mathbf{925}$$

ב. עבור hash join ראשית נחשב האם אפשר לבצע את האלגוריתם

$$\left\lceil \frac{B(movies)}{16 - 1} \right\rceil = \left\lceil \frac{45}{15} \right\rceil = 5 < 14$$

אזי עדיין אפשר להפעיל את האלגוריתם ולכן נקבל שעלות החישוב הינה

$$3B(movies) + 3B(Playsin) = 3 \cdot 45 + 3 \cdot 220 = \mathbf{795}$$

ג. עבור sort-merge-join ראשית נחשב האם אפשר לבצע את האלגוריתם

$$\left\lceil \frac{B(Playsin)}{16} \right\rceil = \left\lceil \frac{220}{16} \right\rceil = 14 < M \wedge \left\lceil \frac{B(movies)}{16} \right\rceil = \left\lceil \frac{45}{16} \right\rceil = 3 < M$$

אזי נוכל להפעיל את האלגוריתם ולמיין גם את *movies and Playsin* אך לא

$$14 + 3 = 17 > M = 16$$

אזי נקבל כי עלות החישוב הינה

$$5B(movies) + 5B(Playsin) = 5 \cdot 45 + 5 \cdot 220 = \mathbf{1325}$$

3. גודל החוצץ המינימלי עבור כל אלגוריתם-

א. עבור block nested-loops נצטרך בלוק ל-movies בלוק ל-playsin ובלוק

ל-output ולכן נצטרך מינימום **3 בלוקים**.

ב. עבור hash join נחשב את מספר הבלוקים המינימלי

$$\left\lceil \frac{B(movies)}{M - 1} \right\rceil < M - 1 \Rightarrow 45 < (M - 1)^2 \Rightarrow M > 8$$

אזי נצטרך לכל הפחות **9 בלוקים**.

ג. עבור sort-merge-join בעלות לא אופטימלית ראינו בשאלה הקודמת כי

עבור $M = 15$ לא יכולנו לחשב אך עבור $M = 16$ כן ולכן נניח כי נצטרך

מינימום **16 בלוקים**.

ד. עבור sort-merge-join בעלות אופטימלית נצטרך

$$\left\lceil \frac{B(movies)}{M} \right\rceil + \left\lceil \frac{B(Playsin)}{M} \right\rceil < M \Rightarrow \frac{45}{M} + \frac{220}{M} < M \Rightarrow$$

$$265 < M^2 \Rightarrow M > 16$$

אזי נצטרך לכל הפחות **17 בלוקים**.

שאלה 2:

נרצה לחשב את הביטוי $\sigma_{A < 10 \wedge C = 8}(R(A, B) \bowtie S(B, C))$

גודל היחסים הם $B(S) = 1000, B(R) = 300$ בכל בלוק של R יש 100 שורות וב-S 50.

ליחס S שני אינדקסים עם עלות גישה נמוכה על B, C וידוע B מפתח ביחס S.

$$V(S, C) = 200, V(R, B) = 100, M_{buffer} = 10$$

א. גודל התוצאה בבלוקים של $\sigma_{C=8}S(B, C)$ הוא

$$T(S) = 50 \cdot 1000 = 50,000 \text{ ראשית נחשב את מס השורות בטבלה-}$$

כמו כן נניח כי ההתפלגות אחידה (שכן אין מידע אחר) אזי נקבל כי מספר השורות

$$\frac{50,000}{V(S, C)} = \frac{50,000}{200} = 250 \text{ המתאימות לתנאי הן -}$$

לבסוף נתון כי בכל בלוק 50 שורות ולכן נקבל כי מספר הבלוקים הינו

$$\frac{250}{50} = \mathbf{5 \text{ blocks}} = \frac{B(S)}{V(S, C)}$$

ב. גודל התוצאה בבלוקים של $\sigma_{A < 10}R(A, B)$ הוא

$$T(R) = 100 \cdot 300 = 30,000 \text{ שוב נחשב את מספר השורות בטבלה-}$$

בהתאם לכלל אצבע שלמדנו בכיתה נניח כי מספר השורות המתאימות הוא שליש

$$\text{ולכן } \frac{30,000}{3} = 10,000, \text{ בעת נתון כי בכל בלוק 100 שורות ולכן נקבל כי מס הבלוקים}$$

$$\frac{10000}{100} = \mathbf{100 \text{ blocks}} = \frac{B(R)}{3}$$

ג. מספר השורות בביטוי כולו הינו כפי שראינו בשיעור מתואר ע"י הנוסחה

כאשר מכיוון ש-B מפתח נקבל כי $V(S, B) = 50,000$ במספר השורות

$$\frac{T(R) \cdot T(S)}{\max\{V(R, B), V(S, B)\}} = \frac{250 \cdot 10,000}{\max(100, 50000)} = \mathbf{50}$$

ד. בעת נתאר את השאילתה לפי חלקים

ראשית נחשב האם כדאי לעשות full table scan או אינדקס לכל אחת משתי

selections.

$$E_S = \sigma_{C=8}S(B, C), E_R = \sigma_{A < 10}R(A, B)$$

מסעיפים קודמים ידוע כי $B(E_S) = 5, B(E_R) = 100$

ב-R אין אינדקס ולכן $read(E_R) = B(R) = 300$ כלומר full table scan

ב-S נעשה אינדקס על C ונראה כי העלות הינה $read(E_S) = T(E_S) = 250$

וללא אינדקס העלות תהיה $read(E_S) = 1000$

אזי נקרא את E_R בעזרת אינדקס ואת E_S ב-full table scan.

בעת נחשב את עלות החישוב של כל אלגוריתם עבור ה-join:

$$BNL: read(E_S) + read(E_R) \cdot \left\lceil \frac{B(E_S)}{M-2} \right\rceil = 250 + 300 \cdot \left\lceil \frac{5}{8} \right\rceil = \mathbf{550}$$

$$INL: read(E_R) + T(E_R) \cdot select = 300 + 10,000 \cdot 1 = 10,300$$

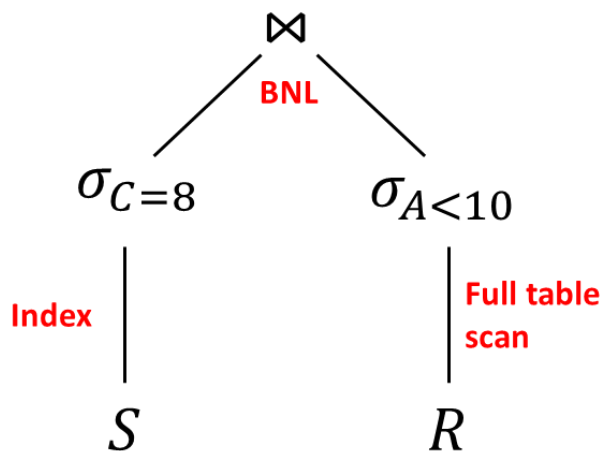
$$HJ: \left\lceil \frac{B(E_R)}{M-1} \right\rceil < M-1 \rightarrow \left\lceil \frac{5}{9} \right\rceil < 9 \checkmark$$

$$read(E_R) + read(E_S) + 2B(E_R) + 2B(E_S) = 250 + 300 + 200 + 10 = \mathbf{760}$$

$$SMJ: \left\lceil \frac{B(E_R)}{M} \right\rceil < M \wedge \left\lceil \frac{B(E_S)}{M} \right\rceil < M \rightarrow \frac{100}{10} < 10$$

אזי SNJ לא יכול לפעול.

אזי נקבל כי האלגוריתם הכי יעיל הינו **BNL**



ה. עלות החישוב היעילה ביותר הינה **550 I/O** כפי שחושב בסעיף קודם.

שאלה 3:

נחשב את הביטוי $\pi_{A,D} \sigma_{B=20 \wedge D < 5}(R(A, B) \bowtie S(A, C, D))$

גודל היחסים הם $B(S) = 1200, B(R) = 4000$ גודל כל אטריביוט הוא 10 בייט וכל בלוק 2000 בייט. אין אינדקסים ואסור לבנות אותם, A מפתח של R.

$$V(S, A) = 1000, V(R, B) = 10, M_{buffer} = 70$$

א. ראשית נחשב כמה שורות יש לכל תת שאילתה

$$4000 \cdot \left(\left\lceil \frac{2000}{20} \right\rceil \right) = \text{מספר השורות הכולל ל-R הינו } \sigma_{B=20} R(A, B) = 400,000$$

עבור תנאי הבחירה נניח התפלגות אחידה על פני 10 ערכים אזי נקבל

$$\left\lceil \frac{400,000}{10} \right\rceil = \mathbf{40000}$$

עבור $\sigma_{D < 5} S(A, C, D)$ מספר השורות הכולל הינו $1200 \cdot \left\lceil \frac{2000}{30} \right\rceil = 79200$
 נניח התפלגות של שליש ונקבל

$$\left\lceil \frac{79200}{3} \right\rceil = 26,400$$

אזי מספר השורות בתוצאה (כאשר $V(R, A) = 400,000$ כי A מפתח ב- R)

$$\frac{T(R) \cdot T(S)}{\max\{V(R, A), V(S, A)\}} = \frac{40000 \cdot 26400}{\max(1000, 400000)} = \mathbf{2640}$$

ב. נעבור על D, A ולכן מספר השורות בבילוק הוא $\left\lceil \frac{2000}{20} \right\rceil = 100$ אזי נקבל כי גודל

$$\left\lceil \frac{2640}{100} \right\rceil = \mathbf{27}$$

התוצאה בבילוקים הוא $\mathbf{27}$

ג. בעת נתאר את השאילתה לפי חלקים
 ראשית מכיוון שאין אינדקסים נוכל להגיד כי כל תת שאילתה של selection הינה
 full table scan

$$E_S = \pi_{A,D} \sigma_{D < 5} S(A, C, D), E_R = \pi_A \sigma_{B=20} R(A, B)$$

$$B(E_S) = \left\lceil \frac{T(E_S)}{\frac{2000}{20}} \right\rceil = \left\lceil \frac{26400}{100} \right\rceil = 264, B(E_R) = \left\lceil \frac{T(E_R)}{\frac{2000}{10}} \right\rceil = \left\lceil \frac{40000}{200} \right\rceil = 200$$

$$read(E_S) = B(S) = 1200, read(E_R) = B(R) = 4000$$

אזי נקרא את E_S ואת E_R כ- full table scan

בעת נחשב את עלות החישוב של כל אלגוריתם עבור ה-join:

$$BNL: read(E_R) + read(E_S) \cdot \left\lceil \frac{B(E_R)}{M-2} \right\rceil = 4000 + 1200 \cdot \left\lceil \frac{200}{68} \right\rceil = \mathbf{7600}$$

INL: no indexes

$$HJ: \left\lceil \frac{B(E_R)}{M-1} \right\rceil < M-1 \rightarrow \left\lceil \frac{200}{69} \right\rceil < 69 \checkmark$$

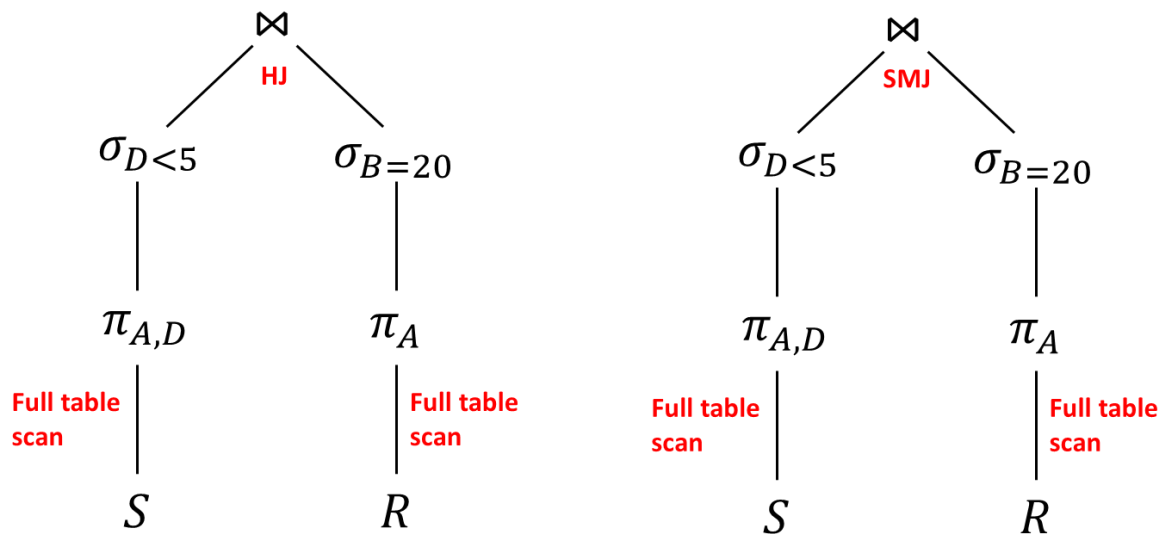
$$read(E_R) + read(E_S) + 2B(E_R) + 2B(E_S) = 1200 + 4000 + 400 + 528 = \mathbf{6128}$$

$$SMJ: \left\lceil \frac{B(E_R)}{M} \right\rceil < M \wedge \left\lceil \frac{B(E_S)}{M} \right\rceil < M \rightarrow \frac{200}{70} < 70 \wedge \frac{264}{70} < 70 \checkmark$$

$$\left\lceil \frac{B(E_R)}{M} \right\rceil + \left\lceil \frac{B(E_S)}{M} \right\rceil < M \rightarrow 4 + 3 < 70 \checkmark$$

$$read(E_R) + read(E_S) + 2B(E_R) + 2B(E_S) = 1200 + 4000 + 400 + 528 = \mathbf{6128}$$

אזי קיבלנו כי SMJ, HJ פועלים בצורה הכי טובה באופן שווה



ד. עלות החישוב היעילה ביותר הינה **6128 I/O** כפי שחושב בסעיף קודם.

שאלה 4:

א. השאילתה רצה ביותר משתי דקות

```

QUERY PLAN
-----
Unique  (cost=54725977.96..54725982.33 rows=250 width=44) (actual time=217780.357..217780.619 rows=116 loops=1)
-> Sort  (cost=54725977.96..54725978.58 rows=250 width=44) (actual time=217780.355..217780.428 rows=116 loops=1)
    Sort Key: ml.movieid, ml.title, ml.rating, ml.year, ml.duration, ml.genre
    Sort Method: quicksort  Memory: 36kB
-> Seq Scan on movies m1  (cost=0.00..54725968.00 rows=250 width=44) (actual time=226.274..217779.985 rows=116 loops=1)
    Filter: (duration = (SubPlan 1))
    Rows Removed by Filter: 49884
    SubPlan 1
        -> Aggregate  (cost=1094.49..1094.50 rows=1 width=4) (actual time=4.348..4.348 rows=1 loops=50000)
            -> Seq Scan on movies m2  (cost=0.00..1093.00 rows=595 width=4) (actual time=1.186..3.774 rows=847 loops=50000)
                Filter: (year = ml.year)
                Rows Removed by Filter: 49153
Planning Time: 2.266 ms
JIT:
  Functions: 10
  Options: Inlining true, Optimization true, Expressions true, Deforming true
Timing: Generation 1.470 ms, Inlining 67.950 ms, Optimization 84.380 ms, Emission 55.791 ms, Total 209.592 ms
Execution Time: 217835.268 ms
(18 rows)

```

ב. השאילתה החדשה היא-

```

SELECT distinct *
FROM Movies Natural join (SELECT year, min(duration) as dur
FROM Movies
GROUP BY year); M2

```

זמן הרצת השאילתה הוא $141.861 + 0.144 = 142.005ms$

להבנתנו, מה שגורם לשיפור בזמן הריצה הוא הימנעות ממעבר על Movies לכל סרט. כך, נעבור על Movies פעם אחת, כאשר קיבצנו את סרטים שיצאו באותה השנה, בחרנו את אורך הסרט המינימלי של כל קבוצה (אותו חישבנו פעם אחת), והשתמשנו בתוצאה על מנת להחזיר את הסרטים הרלוונטים באמצעות natural join.

```
QUERY PLAN
-----
Unique  (cost=2457.94..2460.98 rows=174 width=44) (actual time=141.496..141.734 rows=116 loops=1)
-> Sort (cost=2457.94..2458.37 rows=174 width=44) (actual time=141.494..141.566 rows=116 loops=1)
    Sort Key: ml.year, ml.duration, ml.movieid, ml.title, ml.rating, ml.genre
    Sort Method: quicksort  Memory: 36kB
-> Hash Join (cost=1220.94..2451.46 rows=174 width=44) (actual time=72.867..141.367 rows=116 loops=1)
    Hash Cond: ((ml.year = movies.year) AND (ml.duration = (min(movies.duration))))
-> Seq Scan on movies ml (cost=0.00..968.00 rows=50000 width=44) (actual time=0.009..32.955 rows=50000 loops=1)
-> Hash (cost=1219.68..1219.68 rows=84 width=8) (actual time=72.849..72.849 rows=88 loops=1)
    Buckets: 1024  Batches: 1  Memory Usage: 12kB
-> HashAggregate (cost=1218.00..1218.84 rows=84 width=8) (actual time=72.708..72.771 rows=90 loops=1)
    Group Key: movies.year
-> Seq Scan on movies (cost=0.00..968.00 rows=50000 width=8) (actual time=0.004..32.949 rows=50000 loops=1)

Planning Time: 0.144 ms
Execution Time: 141.861 ms
(14 rows)
```

ג. אפשר לשפר את זמן הריצה ע"י הוספת אינדקס. ניסינו להריץ את השאילתה עם כל מיני אינדקסים למשל- אינדקס על *year* אינדקס על *duration* או על שני השדות. נראה כי האינדקס המוצלח יותר היה *(duration, year)* זאת, משום שהוא שדה משותף בצירוף הטבעי וגם בבחירת משך הסרט המינימלי האינדקס יכול לעזור במקום לעבור אחד אחד על כל משכי הסרטים, וכך אנחנו יכולים לדעת איפה הערך המינימלי יהיה בעץ האינדקס ולגשת אליו בזמן קבוע.

```
QUERY PLAN
-----
Unique  (cost=75583.77..75588.14 rows=250 width=44) (actual time=651.922..652.160 rows=116 loops=1)
-> Sort (cost=75583.77..75584.39 rows=250 width=44) (actual time=651.920..651.992 rows=116 loops=1)
    Sort Key: ml.movieid, ml.title, ml.rating, ml.year, ml.duration, ml.genre
    Sort Method: quicksort  Memory: 36kB
-> Seq Scan on movies ml (cost=0.00..75573.81 rows=250 width=44) (actual time=0.034..651.763 rows=116 loops=1)
    Filter: (duration = (SubPlan 2))
    Rows Removed by Filter: 49884
    SubPlan 2
-> Result (cost=1.48..1.49 rows=1 width=4) (actual time=0.011..0.011 rows=1 loops=50000)
    InitPlan 1 (returns $1)
-> Limit (cost=0.29..1.48 rows=1 width=4) (actual time=0.008..0.009 rows=1 loops=50000)
-> Index Only Scan using movies_year_duration_idx on movies m2 (cost=0.29..599.67 rows=503 width=4) (actual time=0.007..0.007 rows=1 loops=50000)
    Index Cond: ((year = ml.year) AND (duration IS NOT NULL))
    Heap Fetches: 49998

Planning Time: 0.210 ms
Execution Time: 652.291 ms
(16 rows)
```

לאחר הוספת האינדקס זמן הריצה הכולל היה **652.501 ms** אפשר לראות כי לפני הוספת האינדקס המערכת בחרה להשתמש באלגוריתם *HJ* לביצוע השאילתה ולאחר הוספת האינדקס המערכת בחרה באלגוריתם *INL* שכן הוספת האינדקס אפשרה את השימוש בו ועלותו הייתה נמוכה יותר. בנוסף מכיוון שהגדרנו את האינדקס על 2 שדות, אין צורך לגשת לשורות ב-Movies אלא רק להגיע לעלים המתאימים בעץ.

