

תרגיל 4 : Join Algorithms

תאריך הגשה : 55:23, 20.12.20.

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 אחר ההוראות להגשה של קבצים סרוקים!
- תרגיל מוקלד יזכה ב- 2 נקודות בונוס!

שאלה 1 (40 נקודות):

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

Movies (movieId, title, rating, year, duration, genre) = 36b, 227 rows in block, B(Movies)=45

Actors (actorId, name, byear, dyear) = 22b, 372 rows in block, B(Actors)=135

PlaysIn (movieId, actorId, character) = 18b, 455 rows in block, B(PlaysIn)=220

נניח:

- השדות הנומריים: movieId, rating, year, duration, actorId, byear, dyear תופסים כל אחד 4 בייט.
- השדות הטקסטואליים: title, genre, name, character תופסים כל אחד 10 בייט.
- בטבלה Movies יש 10,000 שורות. **45 בלוקים**
- בטבלה Actors יש 50,000 שורות. **135 בלוקים**
- בטבלה PlaysIn יש 100,000 שורות. **171 בלוקים**
- גודל בלוק הוא 8192 בייט.
- גודל החוצץ (buffer) הוא 15 בלוקים.

נרצה לחשב עלות של צירוף (join) של הטבלאות $Movies \bowtie PlaysIn$.

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 movies תופסת 36 בייט.

בכל בלוק נכנסות $\lceil 8192/36 \rceil = 227$ שורות.

הטבלה תופסת סה"כ $B(Movies) = \lceil \frac{10,000}{227} \rceil = 45$ בלוקים.

כל שורה של הטבלה PlaysIn תופסת 18 בייט.

בכל בלוק נכנסות $\lceil 8192/18 \rceil = 455$ שורות.

הטבלה תופסת סה"כ $B(PlaysIn) = \lceil \frac{100,000}{455} \rceil = 220$ בלוקים.

$$B(Movies) + \left\lceil \frac{B(Movie)}{M-2} \right\rceil \times B(PlaysIn) = 45 + \left\lceil \frac{45}{15-2} \right\rceil \times 220$$

כעת נשתמש בנוסחה:

$$\underline{\underline{= 925 I/O}}$$

ב. Hash-join?

$$B(PlaysIn) = 220, B(Movies) = 45$$

נבדוק את התנאי עבור הטבלה הקטנה יותר:

$$\left\lceil \frac{B(Movies)}{M-1} \right\rceil \leq M-2 \rightarrow \left\lceil \frac{45}{14} \right\rceil = 4 \leq 12$$

התנאי מתקיים ולכן נשתמש בנוסחה:

$$3B(Movies) + 3B(PlaysIn) = 3 \times 45 + 3 \times 220$$

$$\underline{\underline{= 795 I/O}}$$

ג. Sort-merge-join?

$$B(PlaysIn) = 220, B(Movies) = 45$$

נבדוק את התנאי עבור הטבלה הגדולה יותר:

$$\left\lceil \frac{B(PlaysIn)}{M} \right\rceil < M \rightarrow \left\lceil \frac{220}{15} \right\rceil = 15 \nless 15$$

התנאי לא מתקיים ולכן אי אפשר להשתמש בSMJ.

2. כעת הניחו שגודל החוצץ הוא 16, איך הייתה משתנה העלות שחישבתם בסעיף 1?

א. *Block-nested-loops*?

$$B(PlaysIn) = 220, B(Movies) = 45$$

החוצץ יותר גדול ולכן כמובן שאפשר לבצע BNL

$$B(Movies) + \left\lceil \frac{B(Movie)}{M-2} \right\rceil \times B(PlaysIn) = 45 + \left\lceil \frac{45}{16-2} \right\rceil \times 220$$

$$\underline{\underline{= 925 I/O}}$$

התוצאה לא השתנתה

ב. *Hash-join*?

$$B(PlaysIn) = 220, B(Movies) = 45$$

החוצץ גדל ולכן בוודאי שהתנאי עדיין מתקיים ולכן נשתמש בנוסחה:

$$3B(Movies) + 3B(PlaysIn) = 3 \times 45 + 3 \times 220$$

$$\underline{\underline{= 795 I/O}}$$

התוצאה לא השתנתה

ג. *Sort-merge-join*?

$$B(PlaysIn) = 220, B(Movies) = 45$$

נבדוק את התנאי עבור הטבלה הגדולה יותר:

$$\left\lceil \frac{B(PlaysIn)}{M} \right\rceil < M \rightarrow \left\lceil \frac{220}{16} \right\rceil = 14 < 16$$

כעת נבדוק אם אפשר לבצע צירוף בשלב השני:

$$\left\lceil \frac{B(PlaysIn)}{M} \right\rceil + \left\lceil \frac{B(Movies)}{M} \right\rceil < M \rightarrow \left\lceil \frac{220}{16} \right\rceil + \left\lceil \frac{45}{16} \right\rceil = 14 + 3 = 17 > 16$$

התנאי לא מתקיים ולכן העלות היא :

$$5B(Movies) + 5B(PlaysIn) = 5 \times 45 + 5 \times 220$$

$$\underline{\underline{= 1,325 I/O}}$$

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

א. *Block-nested-loops*? 3

ב. *Hash-join*? 9

צריך לקיים:

$$\left\lceil \frac{B(\text{Movies})}{M-1} \right\rceil < M-1 \rightarrow \left\lceil \frac{45}{M-1} \right\rceil < M-1 \rightarrow 45 < (M-1)^2$$

$$\left\lceil \frac{45}{8} \right\rceil = 6 < 8 \quad \mathbf{9=M}$$

ניקה $\mathbf{9=M}$ מכיוון ש *Movies* יותר קטן, זה מספיק .

ג. *Sort-merge-join*? 16

צריך לקיים:

$$\left\lceil \frac{B(\text{PlaysIn})}{M} \right\rceil < M \rightarrow \left\lceil \frac{220}{M} \right\rceil < M \rightarrow 220 < M^2$$

$$\left\lceil \frac{220}{16} \right\rceil = 14 < 16 \quad \mathbf{16=M}$$

ניקה $\mathbf{16=M}$ מכיוון ש *Movies* יותר קטן, זה בטוח מספיק גם עבור מיון של *Movies*.

ד. *Sort-merge join* בשימוש באופטימיזציה שמאפשרת חישוב יעיל יותר (הנמנעת ממיון מלא

של היחסים)? 17

צריך לקיים:

$$\left\lceil \frac{B(\text{PlaysIn})}{M} \right\rceil + \left\lceil \frac{B(\text{Movies})}{M} \right\rceil < M \rightarrow \left\lceil \frac{220}{M} \right\rceil + \left\lceil \frac{45}{M} \right\rceil < M$$

$$\left\lceil \frac{220}{17} \right\rceil + \left\lceil \frac{45}{17} \right\rceil = 13 + 3 = 16 < 17 \quad \mathbf{17=M}$$

שאלה 2 (25 נקודות):

רוצים לחשב את הביטוי $\sigma_{A < 10 \wedge C = 8} (R(A, B) \bowtie S(B, C))$. גודלי היחסים הם $B(R)=300$, $B(S)=1,000$. בכל בלוק של R יש 100 שורות, ובכל בלוק של S יש 50 שורות. ליחס S יש שני אינדקסים עם עלות גישה זניחה: אחד על אטריבוט C ואחד על אטריבוט B. כמו כן, ידוע ש B הוא מפתח ביחס S, וכן $V(S, C)=200$, $V(R, B)=100$ 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 $\sigma_{C=8} S(B, C)$

$$\begin{aligned} E_S &= \sigma_{C=8} S(B, C) \text{ נסמן} \\ T(S) &= 1,000 \times 50 = 50,000 \text{ מספר השורות של } S \\ T(E_S) &= \frac{T(S)}{V(S, C)} = \frac{50,000}{200} = 250 \text{ מספר השורות ב } S \text{ שבהן } C=8 \\ B(E_S) &= \frac{250}{50} = 5 \text{ מספר הבלוקים בתוצאת הביטוי} \end{aligned}$$

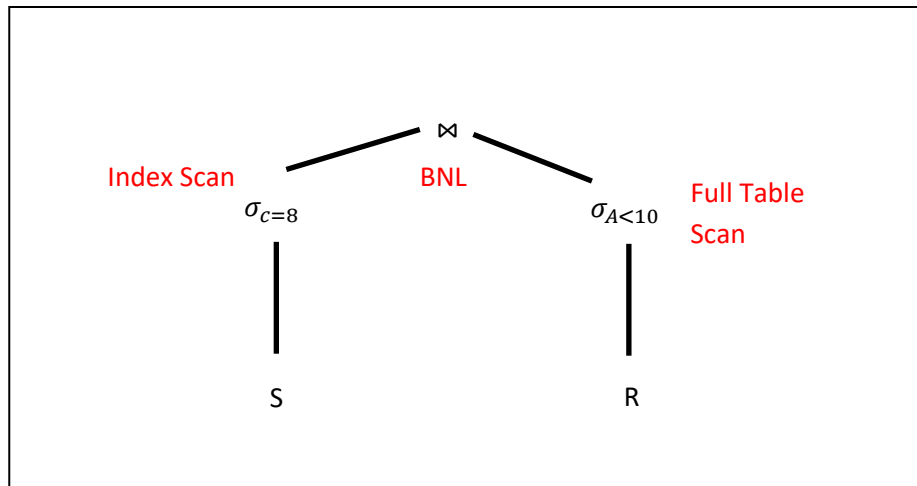
ב. העריכו את גודל התוצאה בבלוקים של הביטוי $\sigma_{A < 10} R(A, B)$

$$\begin{aligned} E_R &= \sigma_{A < 10} R(A, B) \text{ נסמן} \\ T(R) &= 300 \times 100 = 30,000 \text{ מספר השורות של } R \\ T(E_R) &= \frac{30,000}{3} = 10,000 \text{ מספר השורות ב } R \text{ שבהן } A < 10 \\ B(E_R) &= \frac{10,000}{100} = 100 \text{ מספר הבלוקים בתוצאת הביטוי} \end{aligned}$$

ג. העריכו את מספר השורות בתוצאה של הביטוי כולו $\sigma_{A < 10 \wedge C = 8} (R(A, B) \bowtie S(B, C))$

$$\begin{aligned} V(S, B) &= T(S) = 50,000 \text{ ידוע ש } B \text{ הוא מפתח ב } S \text{ ולכן} \\ \frac{T(E_R) \times T(E_S)}{\max\{V(R, B), V(S, B)\}} &= \frac{250 \times 10,000}{50,000} = 50 \end{aligned}$$

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 $T(E_S) = 250$ ועלות גישה של האינדקס זניחה, אז עבור כל שורה בתוצאת הבחירה נקרא לכל היותר בלוק 1.

עלות קריאה של הטבלה כולה זה $B(S) = 1000$

ולכן עדיף השימוש באינדקס ויוצא $read(E_S) = 250$

לטבלה R אין אינדקס ולכן $read(R) = B(R) = 300$

כעת נציב בנוסחה של BNL :

$$read(E_S) + \left\lceil \frac{B(E_S)}{M-2} \right\rceil \times read(E_R) = 250 + \left\lceil \frac{5}{10-2} \right\rceil \times 300 = 550 \text{ I/O}$$

שאלה 3 (20 נקודות):

רוצים לחשב את הביטוי $(R(A, B) \bowtie S(A, C, D)) \pi_{A,D} \sigma_{B=20 \wedge D < 5}$. ההטלה היא ללא מחיקת כפילויות. גודלי היחסים הם $B(S)=1,200$, $B(R)=4,000$. גודל כל אחד מהאטריבוטים הוא 10 bytes וגודל בלוק הוא 2,000 bytes. אין אינדקסים ואסור לבנות אותם. כמו כן $V(S,A)=1000$, $V(R,B)=10$ וידוע ש A הוא מפתח ביחס R . בחוצץ (buffer) יש 70 בלוקים.

א. מה יהיה מספר השורות בתוצאה?

כל שורה של R תופסת 20 בייט, בב्लוק יש 2000 בייט, ולכן יש 100 שורות בב्लוק.

$$T(R) = 4000 \times 100 = 400,000,$$

כל שורה של S תופסת 30 בייט, בב्लוק יש 2000 בייט, ולכן יש 66 שורות בב्लוק.

$$T(S) = 1200 \times 66 = 79,200$$

בגלל שA מפתח בר $V(R, A) = T(R) = 400,000$

$$\frac{T(R) \times T(S)}{\max\{V(R, A), V(S, A)\} \times V(R, B) \times 3} = \frac{400,000 \times 79,200}{400,000 \times 10 \times 3} = 2,640$$

הנוסחה לגודל התוצאה: **2,640**

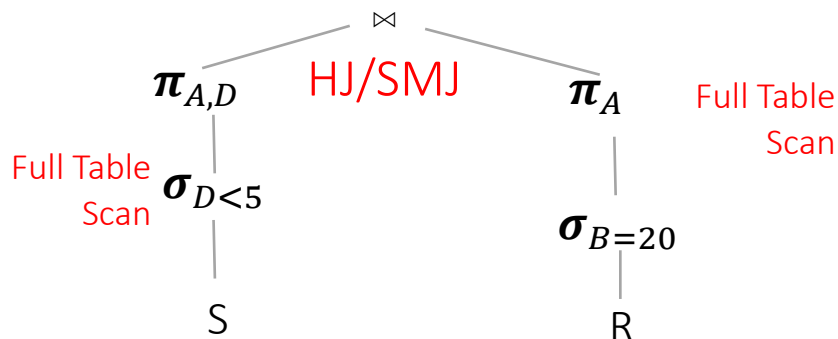
ב. מה יהיה גודל התוצאה בב्लוקים?

כל שורה בתוצאה היא בגודל 20 בייט, כי יש הטלה. ולכן יש 100 שורות בב्लוק.

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

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

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



ד. מה עלות החישוב היעיל ביותר?

$$\begin{aligned} E_R &= \pi_{A \sigma_{B=20}} R(A, B), E_S = \pi_{A,D \sigma_{D < 5}} S(A, C, D) \\ \text{כעת, } Read(E_R) &= B(R) = 4000, Read(E_S) = B(S) = 1200 \\ TSize(E_R) &= \frac{T(R)}{V(R, B)} = \frac{400,000}{10} = 40,000, BSize(E_R) = \frac{40,000}{200} = 200 \\ TSize(E_S) &= \frac{T(S)}{3} = \frac{79,200}{3} = 26,400, BSize(E_S) = \frac{26,400}{100} = 264 \\ \text{נוסחת העלות: } &Read(E_R) + Read(E_S) + 2(BSize(E_R) + BSize(E_S)) = \\ &4,000 + 1,200 + 2(200 + 264) = \mathbf{6,128} \end{aligned}$$

שאלה 4 (15 נקודות):

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

נתון היחס

Movies (movieId, title, rating, year, duration, genre)

ורוצים לחשב את השאילתה הבאה:

```
select distinct *
from Movies M1
where duration = (select min(duration)
                  from Movies M2
                  where M2.year = M1.year);
```

לצורך מענה על הסעיפים הבאים, יש לטעון את הנתונים מהקובץ *moviesBig.csv* הנמצא באתר הקורס לתוך מסד הנתונים במחשב לפי ההוראות הבאות:

1. היכנסו למסד הנתונים (*psql -h dbcourse public*) והשתמשו בפקודה הבאה ליצירת הטבלה:

```
create table movies(
movieId integer primary key,
title varchar(150) not null,
rating numeric check (rating >= 0 and rating <= 10),
year integer check (year > 0),
duration integer check (duration > 0),
genre varchar(50));
```


הערה: אם עדיין קיימת הטבלה משימוש בתרגילים קודמים, מומלץ למחוק אותה (ואת שאר הטבלאות) באמצעות הקובץ `drop.sql` וליצור מחדש.

2. צאו ממסד הנתונים, והריצו את הפקודה הבאה :

```
cat Movies-file-path/moviesBig.csv | psql -hdbcourse public -c "copy Movies FROM STDIN DELIMITER ',' CSV
HEADER"
```

כאשר `Movies-file-path` הוא שם התיקיה שבה מיקמת את הקובץ `moviesBig.csv`.

3. חזרו לתוך מסד הנתונים.

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 לוקח יותר משתי דקות, אפשר להפסיק את ההרצה ולענות: יותר מ-2 דקות).

הריצו פקודת `explain`, שמראה את ה-`query plan` של השאילתה וצרפו אותה לתשובות.

ב. נסו לשפר את זמן הריצה ע"י שינוי בתחביר השאילתה.

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.

ג. האם אפשר לשפר את זמן הריצה ע"י הוספת אינדקס?

בדקו אפשרויות שונות לאינדקס.

כתבו איזה אפשרות של אינדקס שבניתם היה הכי יעיל,

כתבו את זמן הריצה החדש, הריצו את השאילתה עם פקודת `explain analyse`, שמראה את ה-`query`

`plan` של השאילתה, צרפו אותה לתשובות.

נסו להסביר את השינוי בזמן הריצה.

בהצלחה!