תרגיל 4 : Join Algorithms

תאריך הגשה: 23:55, 15.05.22.

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!

<u>שאלה 1 (40 נקודות):</u>

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

authors (name, conference, year, institution, count, adjustedcount)

conferences (conference, area, subarea)

נניח:

- השדות הנומריים adjustedcount, count, year תופסים כל אחד 4 בייט.
- השדות הטקסטואליים: name, conference, institution, area, subarea תופסים כל אחד 10 בייט.
- בטבלה authors יש 165,000 שורות.
- בטבלה conferences יש 20,000 שורות.
- גודל בלוק הוא 8192 בייט.
- גודל החוצץ (buffer) הוא 25 בלוקים.

נרצה לחשב עלות של צירוף (join) של הטבלאות $authors \bowtie conferences$.

1. מה תהיה עלות החישוב של הביטוי לפי כל אחד מהאלגוריתמים הבאים? אם החישוב לא אפשרי, הסבירו למה.

   א. Block-nested-loops?

   ב. Hash-join?

ג. Sort-merge-join?

2. כעת הניחו שגודל החוצץ הוא 30, איך הייתה משתנה העלות שחישבתם בסעיף 1?

א. Block-nested-loops?

ב. Hash-join?

ג. Sort-merge-join?

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

א. Block-nested-loops?

ב. Hash-join?

ג. Sort-merge-join?

ד. Sort-merge join בשימוש באופטימיזציה שמאפשרת חישוב יעיל יותר (הנמנעת ממיון מלא של היחסים)?

<u>שאלה 2 (25 נקודות):</u>

רוצים לחשב את הביטוי $\sigma_{A<25 \wedge B=6}\ (R(A,C) \bowtie S(B,C))$.
גודלי היחסים הם B(S)=200 , B(R)=1500. בכל בלוק של R יש 60 שורות, ובכל בלוק של S יש 150 שורות.
ליחס S יש שני אינדקסים עם עלות גישה זניחה: אחד על אטריבוט C ואחד על אטריבוט B. כמו כן, ידוע שC הוא מפתח ביחס S, וכן V(S,B)=250, V(R,C)=50. 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 $\sigma_{B=6}\ S(B,C)$

ב. העריכו את גודל התוצאה בבלוקים של הביטוי $\sigma_{A<25}\ R(A,C)$

ג. העריכו את מספר השורות בתוצאה של הביטוי כולו $\sigma_{A<25 \wedge B=6}\ (R(A,C) \bowtie S(B,C))$

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

ה. מה עלות החישוב היעיל ביותר?

שאלה 3 (20 נקודות):

רוצים לחשב את הביטוי $\pi_{A,D}\sigma_{B=15 \wedge D<4}\ (R(A,B,C) \bowtie S(C,D))$. ההטלה היא ללא מחיקת כפילויות. גודלי היחסים הם B(S)=3,000, B(R)=1,000 . גודל כל אחד מהאטריבוטים הוא 10 bytes וגודל בלוק הוא 1,500 bytes. אין אינדקסים ואסור לבנות אותם. כמו כן V(S,C)=500,V(R,B)= 5 וידוע שC הוא מפתח ביחס R. בחוצץ (buffer) יש 20 בלוקים.

א. מה יהיה מספר השורות בתוצאה?

ב. מה יהיה גודל התוצאה בבלוקים?

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

ד. מה עלות החישוב היעיל ביותר?

שאלה *4* (15 נקודות):

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

נתון היחס :

authors (name, conference, year, institution, count, adjustedcount)

ורוצים לחשב את השאילתה הבאה המוצאת עבור כל שנה שמופיעה בטבלה authors את השורה/שורות עם הערך adjustedcount הכי גבוהה.

```
select distinct *
from authors a1
where adjustedcount = (select max(adjustedcount)
                       from authors a2
                       where a2.year = a1.year);
```

לצורך מענה על הסעיפים הבאים, יש להשתמש בטבלה *authors* שהוגדרה בתרגילים קודמים.

(אם מחקתם כבר את הטבלה, בבקשה צרו אותה מחדש וטענו בנתונים לפי ההוראות בתרגיל 2.)

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 הריצה הופסקה וקיבלתם הודעת *timeout*, זה בסדר, כתבו זאת בתשובה).

הריצו פקודת *explain*, שמראה את ה*query plan* של השאילתה וצרפו אותה לתשובות.
שימו לב שפקודת '*explain*' בשונה מפקודת '*explain analyse*' לא מריצה את השאילתה, רק מציגה את ה*query plan*.

ב. נסו לשפר את זמן הריצה ע"י שינוי בתחביר השאילתה.
(ה*timeout* המוגדר במערכת הוא של 45 שניות, שאילתה משופרת אמורה לרוץ בהרבה פחות מזה)
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 של השאילתה המקורית (לפני השינוי מסעיף ב') ע"י הוספת אינדקס?
בדקו אפשרויות שונות לאינדקס.
(שימוש באינדקס מתאים אמור לעזור לשאילתה לרוץ בהרבה פחות מ45 שניות)

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.

מידע שימושי:

- מומלץ לתת לאינדקסים שמות בזמן היצירה שלהם.
- פקודה פשוטה למחיקת אינדקס היא מהצורה: drop index *indexname*
  כאשר *indexname* הוא שם האינדקס.
- כדי ראות רשימה של כל האינדקסים הקיימים יש להריץ את הפקודה: \di

בהצלחה!