מטלה 4 - אופטמיזציה

שאלה 1:

נתונות טבלה R לא ממוינת שגודלה 100 בלוקים וטבלה S שגודלה 1000 בלוקים. מעוניינים לבצע Hash-Join בין שתי הטבלאות. לא מוגדרים אינדקסים על הטבלאות. הזיכרון הפנימי כולל מקום פנוי ל-5 בלוקים.

1. (1 נק') לכמה מקסימום קבוצות יכולה פונקציית הגיבוב לחלק את הטבלאות בשלב א של HashJoin בהינתן מגבלת הזיכרון הנתונה ?
2. (2 נק') ידוע כי החלוקה בשלב א' היא מאוזנת לחלוטין. כמה עוד מקום פנוי בזיכרון נדרש במדויק אם מעוניינים להשלים את שלב א ואת שלב ב.
3. (7 נק') בהנחה כי לא ניתן להקצות את הבלוקים הנוספים – הציעו שיטה חלופית יעילה לביצוע שלב ב' (הניחו ששלב א של HashJoin בוצע בהצלחה במגבלת הזיכרון). מה הסיבוכיות במונחי פעולות קריאה וכתיבה הנדרשות להשלמת הפעולה?

א.מכיוון שבזיכרון הפנימי מקום ל5 בלוקים, ניתן לחלק למקסימום 4 קבוצות כאשר הקבוצה החמישית שמורה לאינפוט.

ב.נשתמש בחישוב לזיכרון כאשר f במקרה שלנו שווה לאחד בגלל איזון מוחלט- $B > \frac{fB_R}{B-1} + 2$, B_R=100

נראה כי תוצאת החישוב היא בערך 11.5 בלוקים ולכן נצטרך 12 בלוקים של זיכרון. על כן, בנוסף ל5, נצטרך 7 בלוקים נוספים של זיכרון.

ג.נבדוק את Block nested loop , ישנן 4 קבוצות חלוקה כאשר בכל קבוצה מספר בלוקים Bs=250 , Br=25.

החישוב הינו *$B_R + B_S * ceil(B_R/(B-2))$ כאשר הרלציה הקטנה יותר היא החיצונית ולכן*

*25+250*ceil(25/5-2)=2275. החישוב מתבצע עבור כל קבוצה ולכן סה"כ הזמן הנדרש הינו 9100.*

נבדוק Sort Merge לכל חלוקה –

$$2*25*\log_4 25+2*250\log_4 250+25+250=2382.35$$

נבצע לכל 4 הקבוצות ונקבל 9530.

לכן במקרה זה היה עדיף לבצע Block nested loop

שאלה 2:

נתון בסיס נתונים לניהול הרישום של סטודנטים לקורסים שונים הכולל את הטבלאות הבאות:

Students(std_id, first_name, last_name, city)

Courses (course_id, course_name, points)

Enrolment (course_id, std_id, year, semester)

וכן השאילתא הבאה:

SELECT last_name

FROM Students NATURAL **JOIN** Enrolment NATURAL **JOIN** Courses

WHERE course_name = "Databases" AND city = "Beer-Sheva"

א. שרטטו עץ שאילתא ראשוני (2 נק')

ב. שרטטו עץ שאילתא אופטימאלי ככל האפשר בהתבסס על כללי אופטימיזציה היוריסטית. (8 נק').

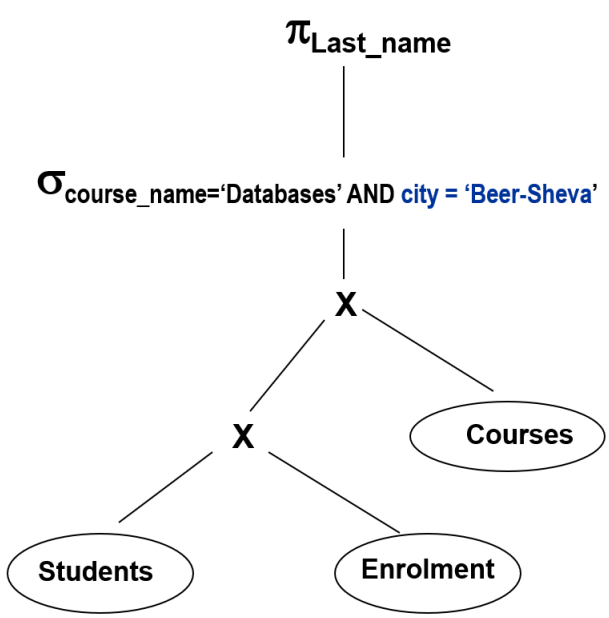
ג. הסבירו מה צריך לשנות בפעולת Hash – Join כדי לאפשר ביצוע הפעולה (3 נק'):

S LEFT OUTER JOIN R

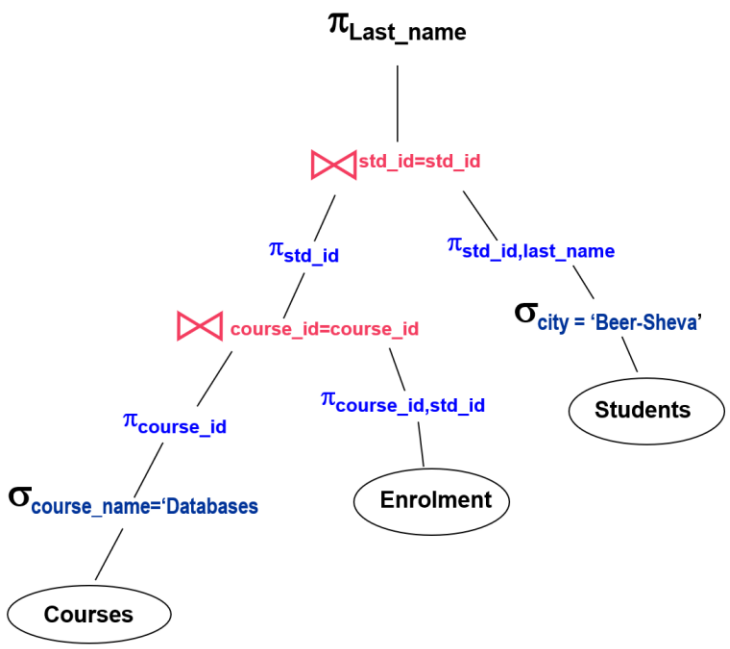
ד. בהמשך לסעיף ג' הסבירו כיצד, אם בכלל, משתנה התנאי לכמות הזיכרון המינימאלית הנדרשת לביצוע הפעולה (2 נק')

2א.

Initial query tree:



2.ב



ג.

-חילוק שתי הרלציות באמצעות hash ל-B-1 מחיצות.

-חילוק כל קבוצה של R לתתי קבוצות באמצעות פונקציית hash חדש ונשים בזיכרון הראשי

-לכל קבוצת חלוקה של S קוראים את הקבוצות המתאימות של R ומנסים למצוא התאמות.

-נעבור על כל הקבוצות בS ונרצה להשאיר את כולן, נבדוק אם הן מופיעות כבר בקבוצת ההתאמות על ידי שימוש בפונקציית הערבול המתאימה עליהן, אם מופיעות אז ניקח את החלק עם ההתאמות, אם לא מופיעות ניקח אותן איך שמופיעות בS.

ד. לא משתנה התנאי, בכל מקרה בכל פעם יש לקרוא מ B-1 S בלוקים לביצוע ההתאמות ולהכניס לזיכרון.