Σειρά Ασκήσεων 2

Πέτσα Γεωργία 3200155

Άσκηση 1

Με βάση την εκφώνηση θα έχουμε έναν πίνακα ως εξής

R

A	B	C	D	E	F

1.000.000 εγγραφές (Records)

[1…10.000] [1…1.000] [1..10]

Έχουμε επίσης τις εξής πληροφορίες:

- ✓ Στα γνωρίσματα A, B, F έχουμε ομοιόμορφη κατανομή, που σημαίνει ότι κάθε τιμή έχει την ίδια πιθανότητα εμφάνισης
- ✓ Οι τιμές μεταξύ τους είναι ανεξάρτητες, δηλαδή μία τιμή σ ’ένα γνώρισμα δεν δίνει κάποια πληροφορία για τις τιμές των υπόλοιπων γνωρισμάτων.
- ✓ Υπάρχει ένα clustered index (A,B) B+ Δέντρο το οποίο βρίσκεται στη μνήμη για γρηγορότερη πρόσβαση στα δεδομένα.
- ✓ Σε μία σελίδα (page) χωράνε 20 records.

A. Αρχικά θα υπολογίσουμε πόσες εγγραφές θα έχουμε για το A=1652
Το 1652 θα βρίσκεται 1 φορά στις 10.000 εγγραφές.
Το 1652 θα βρίσκεται x φορές στις 1.000.000 εγγραφές
Άρα 1 / x = 10.000 / 1.000.000 => x = 100

Έπειτα θα υπολογίσουμε πόσες εγγραφές θα έχουμε για το B > 500
Το B > 500 θα βρίσκεται 500 φορές στις 1.000 εγγραφές.
Το B > 500 θα βρίσκεται x φορές στις 1.000.000 εγγραφές
Άρα 500 / x = 1.000 / 1.000.000 => x = 500.000

Τώρα θα υπολογίσουμε ποια είναι η πιθανότητα να έχουμε εγγραφές που θα ισχύει B > 500
Αφού έχουμε ομοιόμορφη κατανομή, θα είναι P(B > 500) = (1.000 τιμές – 500 τιμές)/ 1.000 τιμές = ½ => P(B > 500) = ½

Έπειτα θα υπολογίσουμε πόσες εγγραφές θα έχουμε για το F > 2
Το F > 2 θα βρίσκεται 8 φορές στις 10 εγγραφές.
Το F > 2 θα βρίσκεται x φορές στις 1.000.000 εγγραφές
Άρα 8 / x = 10 / 1.000.000 => x = 800.000

Τώρα θα υπολογίσουμε την πιθανότητα μία εγγραφή του F να είναι > 2, οπότε θα έχουμε P(F > 2) = (10 τιμές – 2 τιμές)/10 τιμές = 8/10 = 4/5 => P(F > 2) = 4/5

Σχηματικά θα ισχύει δηλαδή το εξής:

R

A	B	C	D	E	F
1651					
1652	1				
1652	2				
…	500				1
1652	1000				10
10.000					
[1…10.000]	[1…1.000]				[1..10]

100 εγγραφές

1.000.000 εγγραφές (Records)

Επειδή έχουμε 100 εγγραφές στο A, 500.000 στο B και 800.000 στο F, και το ερώτημα ζητάει τομή αυτών, σίγουρα το ανώτατο όριο εγγραφών που μπορούμε να έχουμε είναι 100. Οπότε σκεφτόμαστε ως εξής:

Από τις 100 εγγραφές που θα γυρίσει το A, θέλουμε τις μισές από αυτές για να έχουμε B > 500 (Αφού P(B > 500) = 1/2). Δηλαδή έχουμε 100*1/2 = 50 εγγραφές που ικανοποιούν τα πρώτα 2 κριτήρια. Για να έχουμε και το F > 2 παίρνουμε τις 50 εγγραφές που έχει γυρίσει η τομή του A,B και το πολλαπλασιάζουμε με την πιθανότητα και το F > 2. Δηλαδή έχουμε 50* 4/5 = 40. Δηλαδή το συγκεκριμένο ερώτημα θα επιστρέψει περίπου 40 εγγραφές.

B. Αρχικά, με βάση την προαναφερθείσα λογική και το γεγονός ότι έχουμε ένα clustered index στα γνωρίσματα (A, B) θα έχουμε ότι:
Το ευρετήριο βρίσκεται στη μνήμη οπότε θα αναγνώσουμε τις σελίδες του ευρετηρίου για την ανάκτηση των δεικτών προς τις σελίδες του δίσκου που θα περιέχουν τις εγγραφές με A=1652 και B > 500. Επειδή έχουμε clustered index αυτές οι τιμές θα είναι ταξινομημένες στον πίνακα R και άρα θα είναι κοντινές μεταξύ τους, οπότε θα διαβάσουμε 50 εγγραφές. Για τις τιμές του F δεν υπάρχει κάποιο ευρετήριο οπότε δεν γνωρίζουμε τη σειρά με την οποία θα είναι τοποθετημένες οι τιμές του, οπότε θα διαβαστούν και οι 50 που ικανοποιούν τα προηγούμενα 2 κριτήρια.
Οπότε θα έχουμε 50 εγγραφές που θα ανακτηθούν / 20 εγγραφές που χωράνε σε μια σελίδα = $\lceil 5/2 \rceil$ = 3 I/O

Άσκηση 2

Χρησιμοποιούμε clustered index όταν θέλουμε να ανακτήσουμε μια συνεχόμενη περιοχή δεδομένων. Επειδή το clustered index ταξινομεί τις γραμμές του πίνακα της βάσης δεδομένων μπορούμε να έχουμε ΕΝΑ ανά πίνακα. Είναι πιο αποδοτικό να χρησιμοποιείται B+ Δέντρο για ένα clustered index και όχι hash index. Ένα clustered index είναι χρήσιμο όταν η συχνότητα αναζήτησης είναι υψηλή και οι ερωτήσεις συχνά πραγματοποιούνται με βάση αυτό το κλειδί.

Χρησιμοποιούμε non-clustered index όταν θέλουμε να έχουμε γρήγορη αναζήτηση βάσει συγκεκριμένων στηλών ή συνδυασμών στηλών

Τα B+ Δέντρα είναι χρήσιμα για range queries και για ταξινόμηση ORDER BY . (Γι' αυτό και είναι πιο αποδοτική δομή για clustered index).

Τα hash indexes είναι αποδοτικότερα όταν θέλουμε να ανακτήσουμε συγκεκριμένες τιμές, όταν έχουμε JOINS, GROUP BYs, duplicate elimination, counting

Με βάση τα παραπάνω λοιπόν, μπορούμε να απαντήσουμε στις ομάδες ερωτημάτων:

ΟΜΑΔΑ Α

E1. SELECT * FROM T WHERE B < ? (FREQ: 100000)
E2. SELECT * FROM T WHERE C= ? (FREQ: 10000)

Θα δημιουργήσουμε 2 ευρετήρια.

α) IndexE1: στο γνώρισμα B. IndexE2: στο γνώρισμα C.
β) IndexE1: CLUSTERED. INDEXE2: NON-CLUSTERED
γ) IndexE1: B+ Tree. IndexE2: Hash-index

Αιτιολόγηση για το α.
E1. Η συνθήκη του επερωτήματος είναι B<? οπότε το ευρετήριο θα δημιουργηθεί σε αυτό το γνώρισμα.
E2. Η συνθήκη του επερωτήματος είναι C=? οπότε το ευρετήριο θα δημιουργηθεί σε αυτό το γνώρισμα.

Αιτιολόγηση για το β.
E1. Η συχνότητα εκτέλεσης του E1 είναι 100.000 (>>10.000). Έχει αναφερθεί ότι clustered index χρησιμοποιείται όταν έχουμε υψηλή συχνότητα αναζήτησης.
E2. Η συχνότητα εκτέλεσης του E2 είναι 10.000 (<<100.000). Έχει αναφερθεί ότι clustered index χρησιμοποιείται όταν έχουμε υψηλή συχνότητα αναζήτησης. Σε αυτή τη περίπτωση δε θα ήταν αποδοτικό να έχουμε ένα clustered index για την εκτέλεση του E2 γιατί αυτό θα σήμαινε ότι δεν θα είχαμε clustered index στο γνώρισμα B για την εκτέλεση του E1 που εκτελείται συχνότερα. Άρα η συνολική απόδοση των ερωτημάτων στην ΟΜΑΔΑ Α θα ήταν μικρότερη.
Αιτιολόγηση για το γ.
E1. Όπως αναφέρθηκε παραπάνω χρησιμοποιούμε B+ tree όταν έχουμε επερωτήματα εύρους. Το συγκεκριμένο είναι B>? άρα είναι range query άρα είναι προτιμότερο το B+tree.
E2. Όπως αναφέρθηκε παραπάνω χρησιμοποιούμε hash index όταν έχουμε επερωτήματα αναζήτησης συγκεκριμένης τιμής. Το συγκεκριμένο είναι C=? άρα είναι αναζήτηση συγκεκριμένης τιμής άρα είναι προτιμότερο το Hash index.

ΟΜΑΔΑ Β

E1. SELECT * FROM T WHERE B < ? AND C=? (FREQ: 100.000)
E2. SELECT * FROM T WHERE D=? (FREQ: 10.000)
E3. SELECT * FROM T WHERE A=? (FREQ: 1.000)

Θα δημιουργήσουμε 2 ευρετήρια.

α) IndexE1: στο γνώρισμα C,B. IndexE2: στο γνώρισμα D.
β) IndexE1: CLUSTERED. INDEXE2: NON-CLUSTERED
γ) IndexE1: B+ Tree. IndexE2: Hash-index

Αιτιολόγηση για το α.

E1. Η συνθήκη του επερωτήματος είναι B<? ΚΑΙ C=? οπότε το ευρετήριο θα δημιουργηθεί ως προς αυτά τα γνωρίσματα. Όσον αφορά τη σειρά επιλέχθηκε το C,B διότι στατιστικά είναι πιθανότερο το C = ? να επιστρέψει λιγότερα αποτελέσματα (και άρα να περιοριστούν οι εγγραφές που θα εξεταστούν περεταίρω στο B) σε σχέση με το B < ?
E2. Η συνθήκη του επερωτήματος είναι D=? οπότε το ευρετήριο θα δημιουργηθεί σε αυτό το γνώρισμα. (Επιλέχθηκε η δημιουργία index στο γνώρισμα D και όχι στο A (για το επερώτημα E3) διότι η συχνότητα εκτέλεσης του E2 είναι πολύ μεγαλύτερη της συχνότητας εκτέλεσης του E3 και λόγω περιορισμού της εκφώνησης μπορούμε να δημιουργήσουμε το πολύ 2 ευρετήρια).

Αιτιολόγηση για το β.

E1. Η συχνότητα εκτέλεσης του E1 είναι 100.000 (>>10.000). Έχει αναφερθεί ότι clustered index χρησιμοποιείται όταν έχουμε υψηλή συχνότητα αναζήτησης.
E2. Η συχνότητα εκτέλεσης του E2 είναι 10.000 (<<100.000). Έχει αναφερθεί ότι clustered index χρησιμοποιείται όταν έχουμε υψηλή συχνότητα αναζήτησης. Σε αυτή τη περίπτωση δε θα ήταν αποδοτικό να έχουμε ένα clustered index για την εκτέλεση του E2 γιατί αυτό θα σήμαινε ότι δεν θα είχαμε clustered index στο γνώρισμα B για την εκτέλεση του E1 που εκτελείται συχνότερα. Άρα η συνολική απόδοση των ερωτημάτων στην ΟΜΑΔΑ Β θα ήταν μικρότερη.

Αιτιολόγηση για το γ.

E1. Όπως αναφέρθηκε παραπάνω χρησιμοποιούμε B+ tree όταν έχουμε επερωτήματα εύρους. Το συγκεκριμένο είναι B>? άρα είναι range query άρα είναι προτιμότερο το B+tree. Μόλις βερθούν οι τιμές που ικανοποιούν το B>? θα αναζητούνται οι τιμές για τις οποίες ισχύει C=?
E2. Όπως αναφέρθηκε παραπάνω χρησιμοποιούμε hash index όταν έχουμε επερωτήματα αναζήτησης συγκεκριμένης τιμής. Το συγκεκριμένο είναι D=? άρα είναι αναζήτηση συγκεκριμένης τιμής άρα είναι προτιμότερο το Hash index.

ΟΜΑΔΑ Γ

E1. SELECT A,C FROM T WHERE B < ? (FREQ: 100.000)
E2. SELECT * FROM T WHERE D < ? (FREQ: 10.000)

Θα δημιουργήσουμε 2 ευρετήρια.

α) IndexE1: στο γνώρισμα B, το οποίο θα κάνει include τις στήλες A,C. IndexE2: στο γνώρισμα D.
β) IndexE1: CLUSTERED. INDEXE2: NON-CLUSTERED
γ) IndexE1: B+ Tree. IndexE2: B+Tree

Αιτιολόγηση για το α.

E1. Η συνθήκη του επερωτήματος είναι B<? οπότε το ευρετήριο θα δημιουργηθεί ως προς αυτά τα γνωρίσματα.
E2. Η συνθήκη του επερωτήματος είναι D<? οπότε το ευρετήριο θα δημιουργηθεί σε αυτό το γνώρισμα.

Αιτιολόγηση για το β.

E1. Η συχνότητα εκτέλεσης του E1 είναι 100.000 (>>10.000). Έχει αναφερθεί ότι clustered index χρησιμοποιείται όταν έχουμε υψηλή συχνότητα αναζήτησης.
E2. Η συχνότητα εκτέλεσης του E2 είναι 10.000 (<<100.000). Έχει αναφερθεί ότι clustered index χρησιμοποιείται όταν έχουμε υψηλή συχνότητα αναζήτησης. Σε αυτή τη περίπτωση δε θα ήταν αποδοτικό να έχουμε ένα clustered index για την εκτέλεση του E2 γιατί αυτό θα σήμαινε ότι δεν θα είχαμε clustered index στο γνώρισμα B για την εκτέλεση του E1 που εκτελείται συχνότερα. Άρα η συνολική απόδοση των ερωτημάτων στην ΟΜΑΔΑ Γ θα ήταν μικρότερη.

Αιτιολόγηση για το γ.

E1. Όπως αναφέρθηκε παραπάνω χρησιμοποιούμε B+ tree όταν έχουμε επερωτήματα εύρους. Το συγκεκριμένο είναι B<? άρα είναι range query άρα είναι προτιμότερο το B+tree.
E2. Όπως αναφέρθηκε παραπάνω χρησιμοποιούμε B+ tree όταν έχουμε επερωτήματα εύρους. Το συγκεκριμένο είναι D<? άρα είναι range query άρα είναι προτιμότερο το B+tree.

Άσκηση 3

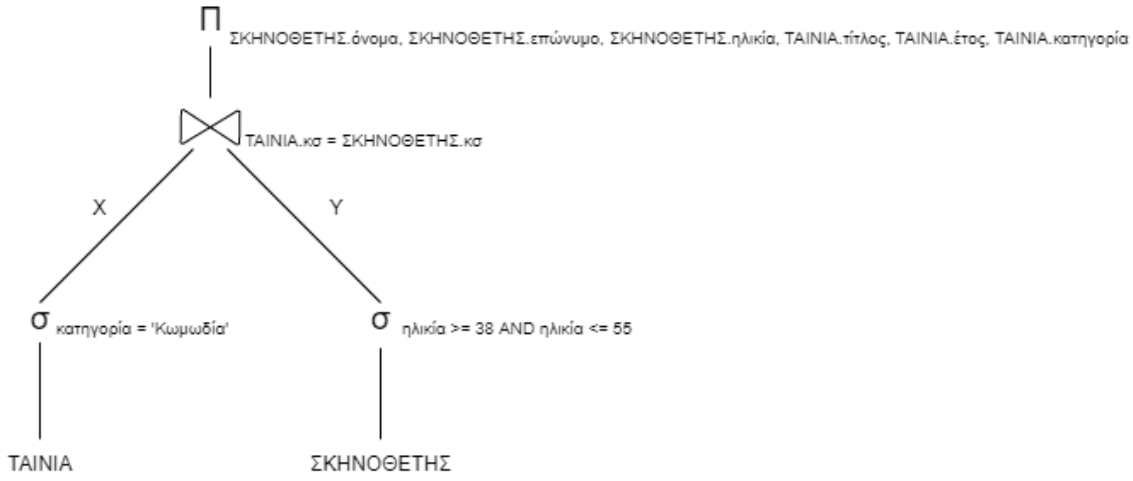
A. Αρχικά γνωρίζουμε ότι έχουμε 2 indexes, ένα clustered στο πεδίο ΣΚΗΝΟΘΕΤΗΣ.ηλικία και ένα non-clustered στο πεδίο ΤΑΙΝΙΑ.κατηγορία. Οπότε για να έχουμε ένα βέλτιστο λογικό πλάνο αυτό θα πρέπει να κάνει χρήση αυτών των 2 ευρετηρίων.

Αρχικά θα εφαρμόσουμε το non-clustered index στο πεδίο ΤΑΙΝΙΑ.κατηγορία για να φιλτράρουμε τις ταινίες με κατηγορία "Κωμωδία". Έτσι, θα ανακτήσουε ένα υποσύνολο των εγγραφών της σχέσης ΤΑΙΝΙΑ που ικανοποιούν το κριτήριο αυτό.

Έπειτα θα εφαρμόσουμε το clustered index στο πεδίο ΣΚΗΝΟΘΕΤΗΣ.ηλικία για να φιλτράρουμε τους σκηνοθέτες με ηλικία μεταξύ 38 και 55. Έτσι θα ανακτήσουμε ένα υποσύνολο των εγγραφών της σχέσης ΣΚΗΝΟΘΕΤΗΣ που ικανοποιούν αυτό το κριτήριο.

Αφού ανακτήσουμε λοιπόν τις παραπάνω εγγραφές θα εκτελέσουμε τη σύζευξη στο πεδίο ΚΣ των 2 σχέσεων.

Το λογικό πλάνο δηλαδή θα είναι το εξής



B.

Αρχικά θα υπολογίσουμε τα pages κάθε πίνακα. Για τον πίνακα ΣΚΗΝΟΘΕΤΗΣ θα έχουμε 4.000records/40(records/page) = 100 pages
Για τον πίνακα ΤΑΙΝΙΑ θα έχουμε 20.000records/20(records/page) = 1.000 pages

Ορίζω S την σχέση ΣΚΗΝΟΘΕΤΗΣ και R τη σχέση ΤΑΙΝΙΑ.

Αρχικά θα υπολογίσουμε τα κόστη για το X, Y.
Για το X:
Για να βρούμε τις εγγραφές όπου θα ισχύει κατηγορία = 'Κωμωδία' έχουμε 2 τρόπους:

i. Table Scan
   Cost $_{\text{table scan}}$ = B(R) = 1.000
ii. Index Seek (using the non-clustered index on κατηγορία)
   Cost $_{\text{index seek}}$ = T(X) = T(R)/V(R, κατηγορία) = 20.000/10 = 2.000
   T(X) = 2.000

Άρα επιλέγουμε να κάνουμε table scan για το X, και έχουμε COST(X) = 1.000

Για το Y:
Για να βρούμε τις εγγραφές όπου θα ισχύει ηλικία >= 38 ΚΑΙ ηλικία <= 55 έχουμε 2 τρόπους:

a) Table Scan
   Cost $_{\text{table scan}}$ = B(S) = 100
b) Index Seek (using the clustered index on ηλικία)
   T(Y) = 2*(1.000/10) + 1.500 + 6 * (500/10) = 200 + 1.500 + 300 = 2.000
   => T(Y) = 2.000
   Cost $_{\text{index seek}}$ = B(Y) = 2.000/40 = 50

Άρα επιλέγουμε να κάνουμε index seek για το Y, και έχουμε COST(Y) = 50

a) $COST_{SMJ}$

Αρχικά θα πρέπει $B(X) + B(Y) < 16^2$ => 2.000/20 + 2.000/40 < 256 => 100 + 50 < 256 => 150 < 256 που ισχύει άρα μπορούμε να τρέξουμε την αποδοτική μέθοδο του SMJ.
Άρα το κόστος θα είναι:
$COST_{SMJ} = COST(X) + COST(Y) + 2*B(X) + 2*B(Y) = 1.000 + 50 + 2 * 100 + 2 * 50 = 1.050 + 200 + 100 = 1350$ => $\boxed{COST_{SMJ} = 1350}$

b) $COST_{NLJ}$

Έχουμε 2 επιλογές για την επιλογή της εξωτερικής σχέσης.

i. Εξωτερική να είναι η Χ. Οπότε

$COST_{NLJ} = COST(X) + \lceil B(X)/M-1 \rceil * COST(Y) = 1.000 + \lceil 100/15 \rceil * 50 = 1.000 + 7 * 50 = 1350$
$\boxed{COST_{NLJ} = 1350}$

ii. Εξωτερική να είναι η Υ. Οπότε

$COST_{NLJ} = COST(Y) + \lceil B(Y)/M-1 \rceil * COST(X) = 50 + \lceil 50/15 \rceil * 1.000 = 50 + 4 * 1.000 = 4.050$
$\boxed{COST_{NLJ} = 4.050σ}$

Επομένως η βέλτιστη επιλογή είναι να πάρουμε ως εξωτερική σχέση τη Χ. Οπότε το τελικό κόστος θα είναι $\boxed{COST_{NLJ} = 1350}$

Άσκηση 4

A.

Για το 1:

Έχουμε ότι γίνεται table scan. Άρα το κόστος θα είναι Β(ΜΑΘΗΤΕΣ), δηλαδή $\boxed{Cost_1 = 100}$

Για το 2:

Έχουμε ότι γίνεται table scan. Άρα το κόστος θα είναι Β(ΣΧΟΛΕΙΑ), δηλαδή $\boxed{Cost_2 = 120}$

Για το 3:

Γίνεται SMJ στο πεδίο ΚΣ.

Αρχικά, γνωρίζουμε από την εκφώνηση ότι τα επιστημονικά πεδία είναι 4 και οι μαθητές κατανέμονται ομοιόμορφα στα 4 επιστημονικά πεδία. Άρα οι εγγραφές που θα έχουν ανακτηθεί από το 1 θα είναι $T_{πεδιο=4}$ (ΜΑΘΗΤΕΣ) = 1.200/4 => $\boxed{T_{πεδιο=4}\text{ (ΜΑΘΗΤΕΣ)} = 300}$

Σκεπτόμενοι με τον ίδιο τρόπο γνωρίζουμε ότι το 5% των του συνόλου των λυκείων είναι ιδιωτικά. Άρα $T_{κατηγορια=ιδιωτικο}$ (ΣΧΟΛΕΙΑ) = 600 * 5% => $\boxed{T_{κατηγορια=ιδιωτικο}\text{ (ΣΧΟΛΕΙΑ)} = 30}$

Για να υπολογίσουμε σε πόσα blocks χωράνε οι παραπάνω εγγραφές σκεφτόμαστε ως εξής:

Γνωρίζουμε:

Η σχέση ΜΑΘΗΤΕΣ έχει 1200 εγγραφές οι οποίες χωρούν σε 100 σελίδες. Άρα 1.200/100 = 12 records/σελίδα

Η σχέση ΣΧΟΛΕΙΑ έχει 600 εγγραφές οι οποίες χωρούν σε 120 σελίδες. Άρα 600/120 = 5 records/σελίδα

Επομένως για τους μαθητές με 4ο επιστημονικό πεδίο θα έχουμε

$B_{πεδιο=4}$ (ΜΑΘΗΤΕΣ) = 300 / 12 = > $B_{πεδιο=4}$ (ΜΑΘΗΤΕΣ) = 25

Και για τα ιδιωτικά σχολεία θα έχουμε $B_{κατηγορια=ιδιωτικο}$ (ΣΧΟΛΕΙΑ) = 30 / 5 => $B_{κατηγορια=ιδιωτικο}$ (ΣΧΟΛΕΙΑ) = 6

Στη συνέχεια ελέγχουμε αν μπορεί να τρέξει η αποδοτική μέθοδος του SMJ. Ελέγχουμε δηλαδή εάν $B_{πεδιο=4}$ (ΜΑΘΗΤΕΣ) + $B_{κατηγορια=ιδιωτικο}$ (ΣΧΟΛΕΙΑ) < M^2 => 25+6 < 25 που δεν ισχύει άρα δεν μπορεί να τρέξει η αποδοτική μέθοδος. Επομένως το κόστος θα είναι

5*($B_{πεδιο=4}$ (ΜΑΘΗΤΕΣ) + $B_{κατηγορια=ιδιωτικο}$ (ΣΧΟΛΕΙΑ))

Όμως για τον υπολογισμο του 1, οι υπολιστες που δημιουργουνται γραφονται στο δισκο, επειτα ξανα διαβαζονται, και συγχωνευονται σε μια (ώστε να γινει η ταξινομηση και αρα να μπορει να τρεξει ο SMJ), επειτα ξανα γραφεται στο δισκο και τελος θα ξανα διαβαστουν για την συζευξη με το 2. Άρα έχουμε 4* $B_{πεδιο=4}$ (ΜΑΘΗΤΕΣ)

Τώρα, για τον υπολογισμο του 2, έχουμε ότι οι υπολίστες που δημιουργούνται γράφονται στο δίσκο, μετά ξανα διαβάζονται για τη συγχώνευση με το 1. Οπότε θα

έχουμε 2 * $B_{κατηγορια=ιδιωτικο}$ (ΣΧΟΛΕΙΑ)). Δηλαδή συνολικά:

4* $B_{πεδιο=4}$ (ΜΑΘΗΤΕΣ) + 2 * $B_{κατηγορια=ιδιωτικο}$ (ΣΧΟΛΕΙΑ)) = 4*25 + 2*6 => $COST_{SMJ}$ = 112

Για το 4:

Το κόστος του INLJ θα είναι T(SMJ_3) * X όπου X ο αριθμός των εγγραφών που κατά μέσο όρο επιστρέφει το non-clustered index του πεδίου ΔΗΛΩΣΕΙΣ.ΑΜ.

Για τον υπολογισμό του X αρχικά γνωρίζουμε ότι το πεδίο ΔΗΛΩΣΕΙΣ.ΑΜ είναι ξένο κλειδί το οποίο αναφέρεται (references) στο πεδίο ΜΑΘΗΤΕΣ.ΑΜ και ότι κάθε μαθητής μπορεί να δηλώσει μέχρι και 10 τμήματα του επιστημονικού πεδίου που διαγωνίστηκε. Με δεδομένη λοιπόν την ομοιόμορφη κατανομή θεωρούμε ότι κάθε μαθητής δηλώνει 10 τμήματα (αφού αυτός είναι ο μέγιστος αριθμός και επομένως το X θα είναι 10.

Συνεχίζουμε με τον υπολογισμό του T(SMJ_3)

Γενικά υπάρχουν διάφοροι τρόποι να το σκεφτούμε. Αρχικά θα μπορούσαν και οι 300 μαθητές που επιλέχθηκαν να ανήκουν όλοι σε ιδιωτικά σχολεία και άρα το T(SMJ_3) να ήταν 300. Από την άλλη θα μπορούσε να θεωρηθεί ότι λόγω της ομοιόμορφης κατανομής θα υπάρχει αντίστοιχο ποσοστό και στους μαθητές (δηλαδή οι μαθητές που πάνε σε ιδιωτικό να αποτελούν το 5% των μαθητών που βρίσκονται στο πεδίο 4). Επειδή η εκφώνηση αναφέρει ότι όπου απαιτείται να υποθέσουμε ότι τα δεδομένα κατανέμονται ομοιόμορφα, θα ακολουθήσουμε τον δεύτερο συλλογισμό, οπότε έχουμε ότι T(SMJ_3) = 300*5% => T(SMJ_3) = 15

Επομένως το κόστος του 4 θα είναι $COST_{INLJ4}$ = T(SMJ_3) * X = 10*15 => $COST_{INLJ4}$ = 150

Για το 5 και το 6:

Δεν έχουνε κάποιο επιπλέον κόστος εφόσον οι εγγραφές βρίσκονται ήδη στη μνήμη μετά την εκτέλεση και του 4

Το συνολικό κόστος όλου του ερωτήματος λοιπόν θα είναι το άθροισμα των επί μερους κοστών, δηλαδή:

ΣΥΝΟΛΙΚΟ ΚΟΣΤΟΣ = 100 + 120 + 112 + 150 + 0 + 0 = 482 I/Os.

B.

Με δεδομένο ότι το M είναι 32 αντί για 5 αρχικά θα μπορούμε να τρέξουμε τον αποδοτικό αλγόριθμο για το SMJ (αφού 25+6 < 32 => 31 < 32). Δηλαδή το κόστος θα ήταν

3*($B_{\text{πεδιο=4}}$ (ΜΑΘΗΤΕΣ) + $B_{\text{κατηγορια=ιδιωτικο}}$ (ΣΧΟΛΕΙΑ)). Όμως αυτά θα υπάρχουν ήδη στη μνήμη λόγω του 1, 2 και άρα δε θα χρειαστεί να διαβαστούν, να γραφτούν και να ξαναδιαβαστούν. Επομένως το κόστος του smj θα μηδενιστεί.

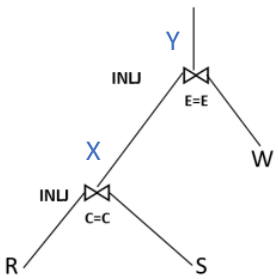
Όσον αφορά ότι το ευρετήριο θα είναι clustered θα έχουμε ότι ο αριθμός των εγγραφών που κάθε φορά επιστρέφονται από την αναζήτηση (το X δηλαδή) θα είναι μικρότερος. Πιο συγκεκριμένα αρχικά θα βρούμε σε πόσες σελίδες θα χωράνε αυτές οι 10 εγγραφές που επιστρέφονται από το index probe. Έχουμε λοιπόν 12.000 εγγραφές/ 400σελίδες = 30 εγγραφές ανά σελίδα. Επομένως οι 10 εγγραφές θα χωράνε σε μία σελίδα και επομένως το νέο X θα είναι 1.

Το συνολικό λοιπόν κόστος του ερωτήματος μεταβάλλεται ως εξής:

ΝΕΟ ΣΥΝΟΛΙΚΟ ΚΟΣΤΟΣ = 100 + 120 + 0 + 15 * 1 + 0 + 0 => ΝΕΟ ΣΥΝΟΛΙΚΟ ΚΟΣΤΟΣ = 235 I/Os

Άσκηση 5

A. Αρχικά θα υπολογίσουμε τον αριθμό των εγγραφών στο πρώτο INLJ. (X)



Σύζευξη των σχέσεων R και S στο πεδίο C:

Για το στάδιο αυτό, πρέπει να υπολογίσουμε τον αριθμό των εγγραφών που θα παράγει η σύζευξη R και S.

Από τη στιγμή που το πεδίο R.C είναι ξένο κλειδί το οποίο αναφέρεται στο πεδίο S.C γνωρίζουμε εκ των προτέρων ότι όλες οι τιμές του θα μπορούν να συζευκτούν με ακριβώς μία τιμή το S.C (αφού είναι κλειδί). Άρα

Αριθμός εγγραφών T(X) = T(R) => T(X) = 1.000

Στη συνέχεια, προχωράμε στο δεύτερο στάδιο, δηλαδή στην σύζευξη του αποτελέσματος του πρώτου σταδίου με τη σχέση W στο πεδίο E με INLJ:

Αντίστοιχα, γνωρίζουμε ότι το πεδίο S.E (Άρα και το X.E) είναι ξένο κλειδί το οποίο αναφέρεται στο πεδίο W.E, οπότε κάθε εγγραφή του X θα μπορεί να συζευκτεί με ακριβώς μία εγγραφή εγγραφή από το W, δηλαδή

Αριθμός εγγραφών T(Y) = T(X) => T(Y) = 1.000

Άρα ο συνολικός αριθμός εγγραφών στην έξοδο της σύζευξης των σχέσεων R, S και W είναι 1.000.

B.

Πάλι θα υπολογίσουμε το κόστος σε 2 στάδια. Για το πρώτο INLJ:
Αρχικά κάθε ερώτηση στο ευρετήριο θα επιστρέφει ακριβώς μία τιμή (αφού το πεδίο S.C είναι κλειδί)
Άρα έχουμε Cost(X) = B(R) + T(R) * 1 = 100 + 1.000 * 1 => Cost(X) = 1.100
Για το δευτερο INLJ, το κόστος θα ήταν
Cost(Y) = Cost(X) + B(X) + T(X) * 1 (αφού πάλι έχουμε ευρετήριο στο πεδίο W.E το οποίο είναι κλειδί)
Το B(X) όμως έχει ήδη διαβαστεί και υπάρχουν στη μνήμη, οπότε το κόστος θα είναι:
Cost(Y) = Cost(X) + T(X) * 1 = 1.100 + 1.000 = 2.100

Άρα το συνολικό κόστος του ερωτήματος θα είναι
Cost(Y) = 2.100

C.

Αρχικά
Η σύζευξη R με S γίνεται στο πεδίο C με χρήση του απλού ευρετηρίου στο πεδίο S.C. Αυτό σημαίνει ότι η απόδοση της σύζευξης R με S θα εξαρτηθεί κυρίως από την απόδοση του ευρετηρίου στο πεδίο S.C και όχι από το εάν υπάρχει ευρετήριο στο πεδίο R.C. Εφόσον το ευρετήριο στο πεδίο S.C είναι non-clustered και βρίσκεται στη μνήμη, η απόδοση της σύζευξης R με S θα είναι ήδη αποτελεσματική.

Όμως,
Με την δημιουργία ενός clustering index στο πεδίο R.C σημαίνει ότι οι εγγραφές της σχέσης R θα οργανωθούν φυσικά στο δίσκο σύμφνωνα με τις τιμές αυτού του πεδίου. Δηλαδή οι εγγραφές που έχουν ίδιες τιμές θα βρίσκονται φυσικά η μία φίπλα στην άλλη στον δίσκο και έτσι η ανάκτηση αυτών των εγγραφών και η αναζήτηση στο non-clustered index θα είναι αποδοτικότερη σε σύγκριση με το scan του ολόκληρου του πίνακα. Δηλαδή όταν διαβάζεται μία τιμή της εξωτερικής σχέσης θα γνωρίζουμε το πλήθος των εγγραφών που έχουν ίδιες τιμές και άρα θα αρκεί μία αναζήτηση στο non-clustered index για αυτές τις εγγραφες (ενώ πριν είχαμε πολλαπλές αναζητήσεις εφόσον η διάταξή τους στο δίσκο ήταν τυχαία).

Άρα η δημιουργία του clustered index στο πεδίο R.C θα επιταχύνει τον υπολογισμό της παραπάνω σύζευξης

*Σημείωση: Σε όλες τις ασκήσεις όλα τα κόστη έχουν ως μονάδα μέτρησης τα I/Os.