2η Σειρά Ασκήσεων

Δημήτριος Τσιομπίκας AM : 3180223

Άσκηση 1

Ερώτημα 1

i) a)

1η Επερώτηση σε SQL :

```
SELECT *
FROM R
WHERE a = 2;
```

Έχουμε :

T(R) = 1000000

B(R) = 20000

V(R,A) = n

Έστω B(X) best case scenario

Άρα B(X) = B(R)/V(R,A) = 20000/n (για clustered index scan)

b)

2η Επερώτηση σε SQL :

```
SELECT *
FROM R
WHERE k <= a AND a <= l;
```

Εφόσον ανήκουν n/10 τιμές εδώ το best case scenario θα είναι :
B(X) = B(R)/V(R,A)/10 = 20000/n/10 = 200000/n (για clustered index scan)

ii)

a)

Εδώ θα πάρουμε το worst case scenario (Έστω T(X)).

Άρα T(X) = T(R) / V(R,A) = 1000000/n (με non-clustered index scan για την 1η επερώτηση)

b)

T(X) = T(R) / V(R,A) / 10 = 1000000/n/10 = 10000000/n (non-clustered index scan , 2η επερώτηση)

Ερώτημα 2

Γνωρίζουμε ότι το Table Scan έχει κόστος B(R).

a)

Για την 1^{η} επερώτηση , μας συμφέρει να το χρησιμοποιήσουμε στην εξής περίπτωση :

T(R) / V(R,A) < B(R) => 1000000/n < 20000 => 20000*n > 1000000 =>

n > 50. Δηλαδή όταν οι διακριτές τιμές είναι περισσότερες από 50.

b)

Για την 2^{η} επερώτηση έχουμε το εξής :

T(R) / V(R,A)/10 < B(R) => (1000000/1) / n/10 < 20000 => 10000000/n < 20000 => 20000*n > 10000000 => n > 500.

Άρα το απλό ευρετήριο συμφέρει αν έχουμε περισσότερες από 500 διακριτές τιμές.

Άσκηση 2

Ερώτημα 1

Έστω ότι η επερώτηση λέγεται Q.

Άρα ψάχνουμε το T(Q).

Εφόσον το διάστημα τιμών στους πίνακες είναι 20 διακριτές τιμές μπορούμε να πούμε ότι :

V(R,B) = 20

V(S,B) = 20

Επίσης μπορούμε να βρούμε το T των πινάκων ως εξής :

T(R) = 0 + 80 + 100 + 20 + 30 = 230 εγγραφές

T(S) = 10 + 100 + 60 + 60 + 0 = 230 εγγραφές

a)

Άρα με βάσει τα στατιστικά των ιστογραμμάτων έχουμε :

T(Q) = T(R)[0-20] * T(S)[0-20] / max{V(R,B),V(S,B)} + T(R)[21-40] * T(S)[21-40] / max{V(R,B),V(S,B)} + T(R)[41-60] * T(S)[41-60]/max{V(R,B),V(S,B)} + T(R)[61-80] * T(S)[61-80] / max{V(R,B),V(S,B)} + T(R)[81-100] * T(S)[81-100] / max{V(R,B),V(S,B)} = 0 * 10 / 20 + 80*100 / 20 + 100*60/20 +

20*60/20 + 30 * 0 /20 = 0 + 400 + 300 + 60 = 760 πλειάδες θα εκτυπώσει.

b)

Και με ομοιόμορφη κατανομή έχουμε :

T(Q) = T(R)* T(S) / V(R,B) = 230 * 230 / 100

= 52900/100 = 529 πλειάδες θα εμφανίσει.

(Χρησιμοποιούμε το V(R,B) επειδή V(R,B) = V(S,B) οπότε δεν έχει σημασία ποιο από τα 2 θα χρησιμοποιήσουμε, το V(R,B) γίνεται 100 γιατί τώρα δεν μας νοιάζουν τα διαστήματα).

Άσκηση 3

Ερώτημα 1

Γνωρίζουμε ότι :

T(R) = 20000

T(S) = 45000

B(R) = 25

B(S) = 30

M = 41

a)

NLJ (Block based) Cost = B(R) + (B(R) / M-1) * B(S) = 25 + (25/40) * 30 = 25 + 1 * 30 = 55 I/Os

b)

SMJ Cost = 5 * (B(R) + B(S)) = 5 * 25 + 30 = 155 I/Os

c)

Hash Join Cost = 3 * (B(R) + B(S)) = 3 * 25 + 30 = 105 I/Os

Ερώτημα 2

1^{ος} τρόπος :

Μπορούμε να χρησιμοποιήσουμε τη βελτιωμένη έκδοση του SMJ με το εξής κόστος :

SMJ Cost = 3*(B(R) + B(S)) = 3 * 25 + 30 = 105 I/Os

2^{ος} τρόπος :

Μπορούν οι σχέσεις εισόδου να ταξινομηθούν προτού εκτελεστεί ο αλγόριθμος με αποτέλεσμα να χουμε το εξής κόστος :

SMJ Cost = B(R) + B(S) = 25 + 30 = 55 I/Os

Άσκηση 4

Έχουμε τα εξής δεδομένα :

T(ΔΑΝΕΙΖΟΜΕΝΟΙ) = 10000

B(ΔΑΝΕΙΖΟΜΕΝΟΙ) = 1000

T(ΒΙΒΛΙΑ) = 50000

B(ΒΙΒΛΙΑ) = 5000

T(ΔΑΝΕΙΣΜΟΙ) = 300000

B(ΔΑΝΕΙΣΜΟΙ) = 15000

V(ΒΙΒΛΙΑ,Εκδότης) = 500

M = 20

1) I/O Cost = T(ΒΙΒΛΙΑ) / V(ΒΙΒΛΙΑ,Εκδότης) = 50000/500 = 100 I/Os (ίδιο με τις πλειάδες)

Έστω ότι το Query λέγεται Q.

T(Q) = T(ΒΙΒΛΙΑ) / V(ΒΙΒΛΙΑ,Εκδότης) = 50000/500 = 100 πλειάδες θα εκτυπώσει.

2) Αριθμός εγγραφών ανα σελίδα του πίνακα ΔΑΝΕΙΣΜΟΙ = T(ΔΑΝΕΙΣΜΟΙ) / B(ΔΑΝΕΙΣΜΟΙ) = 300000/15000 = 20 εγγραφές ανά σελίδα

Ας υποθέσουμε ότι το κόστος ανάγνωσης μέσω ευρετηρίου (έστω X) = 1.
I/O Cost = B(Q) + T(Q) * (X / αριθμός εγγραφών ανα σελίδα του πίνακα ΔΑΝΕΙΣΜΟΙ) = 100 + 100 * (1/20) = 200 I/Os (στρογγυλοποιούμε το 1/20 σε 1)

Έστω πως η σύζευξη λέγεται J
T(J) = T(Q) * T(ΔΑΝΕΙΣΜΟΙ) / max{V(Q,KB),V(ΒΙΒΛΙΑ,KB)} = 100 * 300000 / 50000 = 600 πλειάδες θα εκτυπώσει.

3) Ας υποθέσουμε ότι B(ΠΚΔ) = 1

I/O Cost = I/O cost(INLJ) + (B(ΠΚΔ)/M-1) * B(ΔΑΝΕΙΖΟΜΕΝΟΙ) = 200 + (1/19) * 1000 = 200 + 1*1000 = 1200 I/Os

Έστω L το όνομα της σύζευξης.

T(L) = T(J) * T(ΔΑΝΕΙΖΟΜΕΝΟΙ) / max{V(J,ΚΔ),V(ΔΑΝΕΙΖΟΜΕΝΟΙ,ΚΔ)} = 600 * 10000 / 10000 = 600 πλειάδες θα εκτυπώσει.

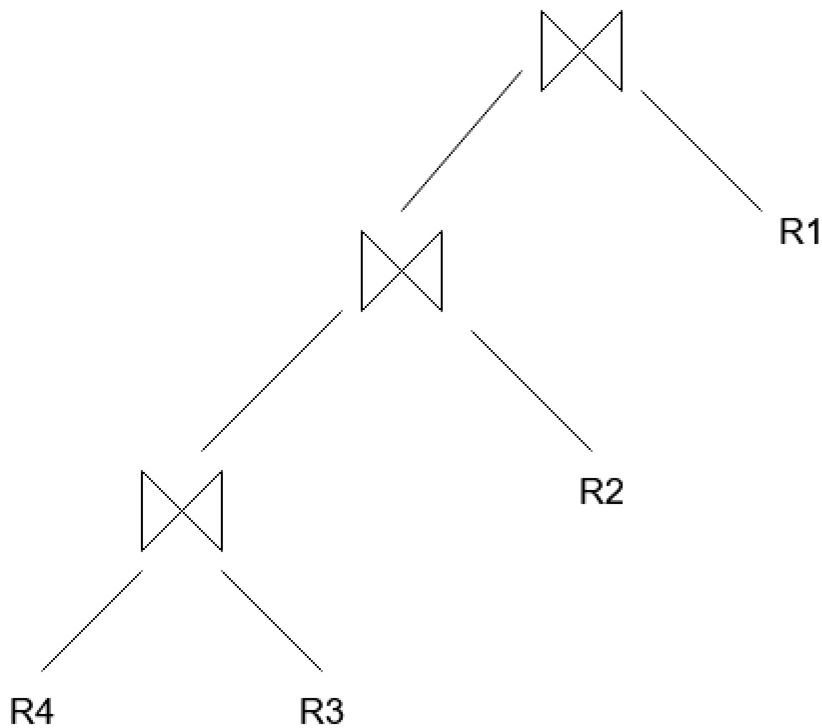
4) Εφόσον οι ηλικίες κυμαίνονται στα 7 εως 24 έτη θα έχουμε V(L,Ηλικία) = 24-7+1 = 18. Όμως , εμείς θέλουμε μόνο το εξής διάστημα [13,19] άρα τις 7 από τις 18 τιμές. Γνωρίζουμε ότι για να πάρουμε μία από τις 18 διακριτές τιμές πρέπει να κάνουμε το εξής :
T(L) / V(L,Ηλικία) = 600/18 = 34 πλειάδες.

Όμως εμείς θέλουμε τις 7 από τις 18 άρα καταλήγουμε στο εξής αποτέλεσμα :
Έστω W η επερώτηση :
T(W) = 7 * 34 = 238 πλειάδες θα εκτυπώσει.

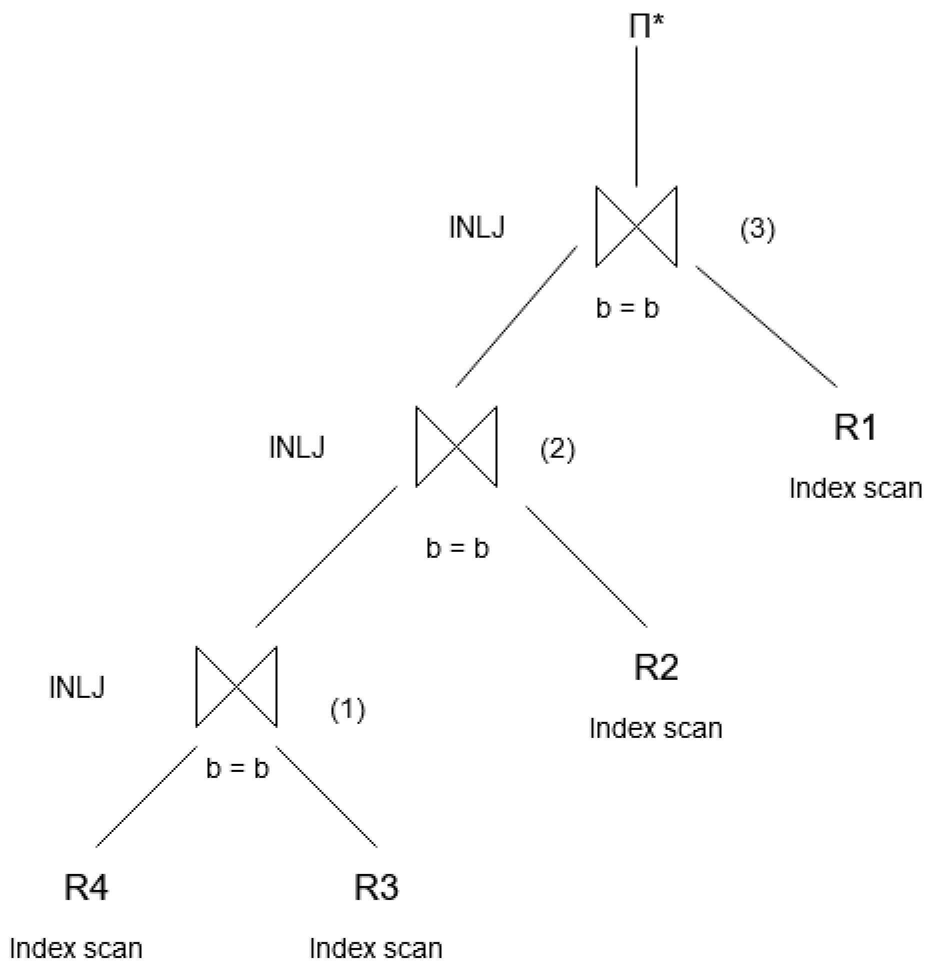
I/O Cost = 7 * 34 = 238 I/Os.

Άσκηση 5

Ερώτημα 1



Ερώτημα 2



Το πλάνο που πρότεινα είναι αποδοτικό διότι εφόσον κάθε γνώρισμα b έχει ευρετήριο συστάδων , ο αλγόριθμος INLJ (Indexed Nested Loop Join) είναι ο αποδοτικότερος για τις συζεύξεις.

Ερώτημα 3

I/O Cost (1) = B(R4) + (B(R4) / M-1) * B(R3)

I/O Cost (2) = B(1) + (B(1) / M-1) * B(R2)

I/O Cost (3) = B(2) + (B(2) / M-1) * B(R1)

Υποθέτω ότι ο R4 είναι το Outer relation και το R3 το inner relation , και ξεκινάω να βρίσκω τα I/Os με την περίπτωση INLJ για clustered indexes.