2η Σειρά Ασκήσεων

Φίλιππος Δουραχαλής, 3170045

Άσκηση 1)

$T(R) = 1.000.000$ εγγραφές

$B(R) = 20.000$ σελίδες

$T(R)/B(R) = 50\ {}^{εγγραφές}/_{σελίδα}$

$V(R,a) = n$

1. i. Το ευρετήριο είναι ευρετήριο συστάδων.

   a. Έστω $W = \sigma_{\alpha=2}(R)$
   Γνωρίζουμε πως εάν οι τιμές είναι ομοιόμορφα κατανεμημένες, το πλήθος πλειάδων που περιμένουμε να επιστρέψει ο τελεστής επιλογής είναι: T(R) / V(R,a), επομένως:

   $$T(W) = \frac{T(R)}{V(R,a)} = \frac{1.000.000}{n} \text{ εγγραφές}$$

   Οι εγγραφές αυτές καταλαμβάνουν

   $$B(W) = \frac{T(W)}{50} = \frac{20.000}{n} \text{ σελίδες}$$

   Εφόσον το ευρετήριο είναι ευρετήριο συστάδων, σημαίνει πως οι τιμές του πεδίου ευρετηριοποίησης είναι διατεταγμένες. Σε αυτή την περίπτωση θα χρειαστεί να ακολουθήσουμε τον δείκτη προς το μπλοκ που περιέχει την πρώτη εμφάνιση της τιμής α=2 και να διαβάσουμε διαδοχικά όλα τα επόμενα μπλοκ με εγγραφές που θα περιέχουν την ίδια τιμή. Άρα θα χρειαστεί να διαβάσουμε συνολικά 20.000/n σελίδες, δηλαδή να πραγματοποιήσουμε $\boldsymbol{\frac{20.000}{n}}$ **I/Os.**

   b. Έστω $U = \sigma_{K \le \alpha \le L}(R)$
   Γνωρίζουμε πως η συνθήκη ικανοποιείται για n/10 τιμές του α.
   Άρα για να βρούμε το συνολικό πλήθος πλειάδων αρκεί να εφαρμόσουμε τον τύπο του υποερωτήματος α και για τις n/10 τιμές. Επειδή οι τιμές είναι

ομοιόμορφα κατανεμημένες, ο τύπος αυτός θα επιστρέψει το ίδιο πλήθος για κάθε τιμή. Άρα:

$$T(U) = \frac{n}{10} * \frac{1.000.000}{n} = 100.000$$

$$B(U) = \frac{100.000}{50} = 2.000$$

Λόγω του ευρετηρίου συστάδων, όπως και προηγουμένως, ακολουθούμε τους δείκτες προς την πρώτη εμφάνιση των τιμών στα μπλοκ, και θα διαβάσουμε συνολικά 2.000 σελίδες, δηλαδή θα πραγματοποιηθούν **2.000 I/Os.**

ii. Το ευρετήριο είναι απλό.

a. Όπως και στην περίπτωση του ευρετηρίου συστάδων, το ερώτημα περιμένουμε να μας επιστρέψει $\frac{1.000.000}{n}$ εγγραφές.

Ωστόσο οι εγγραφές αυτές μπορεί να μην βρίσκονται σε διαδοχικά block καθώς οι τιμές στις σελίδες δεν είναι διατεταγμένες στις σελίδες, όπως θα ήταν σε ένα ευρετήριο συστάδων. Στη χειρότερη περίπτωση κάθε τιμή θα βρίσκεται σε διαφορετική σελίδα, άρα θα πρέπει να προσπελαστούν $\frac{1.000.000}{n}$ σελίδες, αφού στο απλό ευρετήριο έχουμε δείκτες προς εγγραφές κι όχι δείκτες μπλοκ και θα γίνουν συνολικά $\boldsymbol{\frac{1.000.000}{n}}$ **I/Os.**

b. Και πάλι υποθέτοντας ομοιόμορφη κατανομή, το ερώτημα θα μας επιστρέψει 100.000 εγγραφές ($^{1.000.000}/_{n}$ εγγραφές για κάθε μια από τις $^{n}/_{10}$ διακριτές τιμές του α). Οι εγγραφές δεν είναι διατεταγμένες ως προς το πεδίο α στα μπλοκ της σχέσης. Και πάλι, στην χειρότερη περίπτωση κάθε μια θα βρίσκεται σε ξεχωριστό μπλοκ. Τότε και πάλι θα πρέπει να ακολουθήσουμε κάθε pointer προς κάθε μια εγγραφή ξεχωριστά. Το συνολικό πλήθος I/O που θα γίνουν είναι **100.000 I/Os**

2. a. $\sigma_{\alpha=2}(R)$

Ένα table scan θα διαβάσει διαδοχικά όλα τα block της σχέσης, δηλαδή θα γίνουν 20.000 I/O. Θέλουμε τα I/O που υπολογίσαμε στο ερώτημα 1. ii. a. να είναι λιγότερα από αυτά που θα έκανε το Table Scan. Εφόσον το ευρετήριο είναι στη μνήμη δεν προσμετράται το κόστος της ανάγνωσης των σελίδων του ευρετηρίου που περιέχουν τους δείκτες. Άρα πρέπει:

$$\frac{1.000.000}{\mathrm{n}} < 20.000 \Leftrightarrow 1.000.000 < 20.000n \Leftrightarrow \mathrm{n} > 50$$

b. $\sigma_{K \leq \alpha \leq L}(R)$

Όπως και στο προηγούμενο ερώτημα, θέλουμε η ποσότητα που υπολογίστηκε στο υποερώτημα 1. ii. b. να είναι μικρότερη του 20.000. Όμως, βάσει της ποσότητας που υπολογίσαμε παρατηρούμε πως αυτή είναι σταθερή και δεν εξαρτάται από το n. Άρα για αυτό το επερώτημα, λόγω του απλού ευρετηρίου που έχουμε, καταλήγουμε στο συμπέρασμα πως το Table Scan είναι πάντα πιο αποδοτικό από το Index Scan αφού 20.000 < 100.000

Άσκηση 2) **SELECT** a,c **FROM** R,S **WHERE** R.b = S.b

Στο ερώτημα υπολογίζουμε ουσιαστικά την σύζευξη των δύο πινάκων. Γνωρίζουμε πως το πλήθος αποτελεσμάτων που θα προκύψουν μπορεί να βρίσκονται στο εύρος $[0, \mathrm{T(R)} * \mathrm{T(S)}]$.
Αυτό σημαίνει πως για κάθε πλειάδα της R με κάποια τιμή R.b μπορεί να υπάρχουν 0 ή περισσότερες πλειάδες της S με τις οποίες να γίνεται join.

a. Εφόσον έχουμε το ίδιο εύρος τιμών σε κάθε κουβά, αν οι τιμές των R.b και S.b είναι ομοιόμορφα κατανεμημένες σε κάθε bucket τότε αυτό σημαίνει ότι περιμένουμε για κάθε ένα από αυτά, να υπάρχουν T(S)/V(S,a) τιμές της S με τις οποίες κάνει Join κάθε τιμή της R (όπου το V(S,a) ισούται με το εύρος των bucket). Αυτό συμβαίνει για κάθε τιμή ενός bucket, δηλαδή:

   $\sum_{i=1}^{n} \frac{\mathrm{T(R)}_i * \mathrm{T(S)}_i}{\mathrm{Range}}$,όπου n ο αριθμός των buckets (εδώ n=5).

   Άρα τελικά ο συνολικός αριθμός εγγραφών που θα επιστραφούν είναι

   $\frac{(0*10)+(80*100)+(100*60)+(20*60)+(30*0)}{20}$ = **760 πλειάδες**

b. Έστω ότι οι τιμές είναι ομοιόμορφα κατανεμημένες στη σχέση και δεν έχουμε στη διάθεση μας το ιστόγραμμα. Κάθε σχέση περιέχει T(R) = T(S) = 230 εγγραφές. Οι διακριτές τιμές των σχέσεων είναι V(R,b) = V(S,b) = 100 και υποθέτουμε πως όλες εμφανίζονται στις σχέσεις.

   Το πλήθος των πλειάδων που εκτιμάμε ότι θα επιστραφούν είναι:
   $\frac{T(R)*T(S)}{\max\{V(R,b),V(S,b)\}} = \frac{230*230}{100} = \frac{52.900}{100} = \mathbf{5.290}$ **εγγραφές**

Άσκηση 3)

$\mathrm{T(R)} = 20.000$

$\mathrm{B(R)} = \frac{20.000}{25} = 800$

$T(S) = 45.000$

$B(S) = \frac{45.000}{30} = 1.500$

M = 41

1.

a. Για να εκτελεστεί ο BNLJ με αποδοτικό τρόπο, επιλέγουμε ως εξωτερική σχέση την μικρότερη και της δίνουμε τις M-1=40 σελίδες από τις 41 διαθέσιμες. Έτσι μειώνουμε την πολυπλοκότητα του αλγορίθμου καθώς η εσωτερική σχέση θα διαβαστεί λιγότερες φορές.

   Εφόσον B(R) < B(S), επιλέγουμε την S για εξωτερική σχέση. Αυτή θα διαβαστεί μια φορά ολόκληρη σε τμήματα των 40 block. Για κάθε τμήμα $\frac{B(R)}{m-1} = \left\lceil \frac{800}{40} \right\rceil = 20$ σελίδων, διαβάζουμε διαδοχικά ολόκληρη την S, ένα block την φορά (συνολικά B(S) σελίδες).

   Άρα συνολικά θα διαβαστούν

   $\mathrm{B(R)} + \frac{\mathrm{B(R)}}{\mathrm{M}-1} * \mathrm{B(S)} = 1500 + 20 * 1500 = 30.800$ σελίδες

   και θα γίνουν **30.800 I/Os**

b. Έχουμε ότι $\max\{B(R), B(S)\} = B(S) = 1500 \leq M^2 = 1.681$, άρα μπορεί να εκτελεστεί ο αλγόριθμος.

   Σε μία απλοϊκή εκτέλεση του αλγορίθμου, στη πρώτη φάση (Sort) θα διαβαστούν οι σχέσεις R και S σε τμήματα 41 σελίδων τη φορά και θα δημιουργηθούν

   $\left\lceil \frac{800}{41} \right\rceil = 20$ και $\left\lceil \frac{1500}{41} \right\rceil = 37$ sublists αντίστοιχα. Ταξινομούμε τα sublists με τον τρόπο που ορίζεται στον αλγόριθμο 2-Phase-Sort, δηλαδή αφού διαβάσουμε[1] και ταξινομήσουμε τις εγγραφές των M block κάθε φορά, τα γράφουμε[2] πίσω στον δίσκο ως μια sublist. Στη συνέχεια διαβάζουμε[3] ξανά ένα block από κάθε λίστα και τα συγχωνεύουμε. Γράφουμε[4] τις συγχωνευμένες sublists ως ταξινομημένη σχέση στον δίσκο. Τέλος διαβάζουμε[5] τις

ταξινομημένες σχέσεις και τις δίνουμε στη φάση Merge-Join του αλγόριθμου SMJ. Η διαδικασία τρέχει ξεχωριστά για κάθε σχέση. Σε κάθε στάδιο που αναφέρθηκε διαβάζουμε και γράφουμε αντίστοιχα στην ουσία τόσα blocks όσες και οι σελίδες της σχέσης. Άρα το συνολικό κόστος του αλγορίθμου θα είναι:
$B(R) + B(S) + B(R) + B(S) + B(R) + B(S) + B(R) + B(S) + B(R) + B(S) = 5\big(B(R) + B(S)\big) = 5(800 + 1.500) = \mathbf{11.500\ I/Os}$

c. Διαλέγοντας την μικρότερη σχέση, δηλαδή την R, ως εξωτερική, η απαιτούμενη μνήμη για τον hash join είναι $(M-1)^2 \geq B(R) \Longrightarrow 40^2 \geq 800$, που ισχύει. Στον Hash-Join, αρχικά διαβάζουμε και τις δύο σχέσεις ολόκληρες και δημιουργούμε μέσω κάποιας συνάρτησης κατακερματισμού τα buckets H_i και G_i (εφόσον M=41, μπορούμε να έχουμε εώς 40 buckets των 40 σελίδων το καθένα) των σχέσεων R και S αντίστοιχα. Στη συνέχεια γράφουμε αυτά τα buckets πίσω στον δίσκο. Και για τις δύο σχέσεις, ο συνολικός αριθμός σελίδων σε όλα τα buckets θα είναι ίσος με το πλήθος σελίδων της σχέσης, άρα θα γράψουμε τόσες σελίδες. Μετά διαβάζουμε διαδοχικά τα αντίστοιχα buckets H_i και G_i και ελέγχουμε τις εγγραφές σε κάθε σελίδα τους. Αν οι εγγραφές έχουν τις ίδιες τιμές,
R.x = S.x, γίνονται join. Άρα συνολικά διαβάζουμε 2 φορές και τις δύο σχέσεις και γράφουμε από μια φορά την καθεμία, δηλαδή το κόστος θα είναι $3\big(B(R) + B(S)\big) = 3(800 + 1.500) = \mathbf{6.900\ I/O}$

2. a. Ένας τρόπος να βελτιωθεί ο αλγόριθμος SMJ είναι να ενσωματώσουμε τη 2η φάση του αλγορίθμου 2-phase-sort στην φάση Merge-Join του αλγορίθμου SMJ. Δηλαδή αντί να συγχωνεύσουμε τις ταξινομημένες λίστες και να τις γράψουμε πίσω στο δίσκο, ώστε να τις διαβάσουμε αργότερα και να τις δώσουμε στον SMJ, αφού δημιουργήσουμε τα 20+37=57 sublists, κρατάμε έναν buffer για κάθε λίστα και τις ανοίγουμε όλες ταυτόχρονα στον SMJ. Έτσι αφαιρούμε στην ουσία το κόστος της εγγραφής και μετά της ανάγνωσης της ταξινομημένης σχέσης, όμως αυξάνονται οι απαιτήσεις μνήμης αφού χρειαζόμαστε πλέον 57 buffers. Γενικά, αφού και οι λίστες και των δύο σχέσεων ανοίγουν ταυτόχρονα και καταλαμβάνουν συνολικά B(R) + B(S) σελίδες, πρέπει $M \geq \sqrt{B(R) + B(S)}$.
Ωστόσο παρατηρούμε ότι
$B(R) + B(S) = 2.300 \geq M^2 = 1.681$, άρα η αποδοτική έκδοση του αλγορίθμου δεν μπορεί να τρέξει.
Σε κάθε περίπτωση, το νέο κόστος που προκύπτει από τον βελτιωμένο αλγόριθμο θα είναι ίσο με το κόστος του να διαβάσουμε διαδοχικά τις δύο σχέσεις σε τμήματα των 41 σελίδων, να ταξινομήσουμε και να γράψουμε πίσω τα sublists στον δίσκο, και στη συνέχεια να φορτώσουμε (διαβάσουμε) παράλληλα όλα τα sublists για να συγχωνευτούν στον SMJ. Μέσα στον SMJ

δεν υπάρχει κάποιο επιπλέον κόστος, καθώς ο αλγόριθμος εκτελείται στην μνήμη και υποθέτουμε ότι δεν απαιτούνται επιπλέον αναγνώσεις από τον δίσκο όσο εκτελείται.
Επομένως, το συνολικό κόστος είναι $3\big(B(R) * B(S)\big) = 3(800 + 1.500) =$ **6.900 I/O.**
Όμως ο μόνος τρόπος να πετύχουμε τόσο χαμηλό κόστος είναι να αυξήσουμε το μέγεθος της διαθέσιμης μνήμης.

b. Ένας δεύτερος τρόπος να εκτελέσουμε αποδοτικά τον SMJ είναι αν συγχωνεύσουμε τα sublists της μικρότερης σχέσης R (ώστε να γίνουν λιγότερα I/Os κατά τη συγχώνευση) πριν εκτελέσουμε τον αλγόριθμο. Οι λίστες της μεγαλύτερης σχέσης S φορτώνονται στους buffers για να δοθούν στον SMJ όπως προηγουμένως . Έτσι μειώνουμε τις απαιτήσεις μνήμης καθώς πλέον απαιτούνται 1 + 37 = 38 buffers μνήμης, αντί για 57, και άρα η μνήμη επαρκεί για να τρέξει όμως αυξάνεται σχετικά το κόστος σε I/O σε σύγκριση με την προηγούμενη υλοποίηση αφού για την μικρότερη σχέση πρέπει να εκτελεστούν όλα τα βήματα του απλού SMJ. Άρα ο συνδυασμός αυτός απλοϊκού και βελτιωμένου SMJ έχει κόστος:
$5B(R) + 3B(S) = 5 * 800 + 3 * 1.500 =$ **8.500 I/O**
Αυτό είναι το χαμηλότερο δυνατό κόστος που μπορούμε να πετύχουμε χωρίς να αυξήσουμε την διαθέσιμη μνήμη.

Σε κάθε περίπτωση, δεν μπορούμε να κάνουμε κάτι καλύτερο από $3(B(R) + B(S))$, υποθέτοντας ότι η μνήμη επαρκεί.

Άσκηση 4)

$\mathrm{T}(\Delta\mathrm{ANEIZOMENOI}) = 10.000$

$\mathrm{B}(\Delta\mathrm{ANEIZOMENOI}) = 1.000,\ {}^{10.000}/_{1.000} = 10\ {}^{εγγραφές}/_{σελίδα}$

$\mathrm{T}(\mathrm{BIBΛIA}) = 50.000$

$\mathrm{B}(\mathrm{BΊBΛIA}) = 5.000,\ {}^{50.000}/_{5.000} = 10\ {}^{εγγραφές}/_{σελίδα}$

$\mathrm{T}(\Delta\mathrm{ANEIΣMOI}) = 300.000$

$\mathrm{B}(\Delta\mathrm{ANEIΣMOI}) = 15.000,\ {}^{300.000}/_{15.000} = 20\ {}^{εγγραφές}/_{σελίδα}$

$\mathrm{V}(\mathrm{BIBΛIA}, \mathrm{Εκδότης}) = 500$

$\mathrm{V}(\Delta\mathrm{ANEIZOMENOI}, \mathrm{Ηλικία}) = 18$

$\mathrm{M} = 20$

I(ΒΙΒΛΙΑ.Εκδότης)

I(ΔΑΝΕΙΣΜΟΙ.ΚΒ)

1) $\mathbf{X = \sigma_{Εκδότης='Σαββάλας'}(ΒΙΒΛΙΑ)}$

Η επιλογή αναμένουμε να μας επιστρέψει

$$T(X) = \frac{T(BIBΛIA)}{V(BIBIΛIA,Εκδότης)} = \frac{50.000}{500} = \mathbf{100\ εγγραφές}$$

Δεν υπάρχει clustered index στο γνώρισμα «Εκδότης», άρα θα διαβάσουμε κάθε εγγραφή ξεχωριστά μέσω του index. Άρα θα γίνουν **100 I/O**[1]

$$B(X) = \frac{T(X)}{10} = 10 \text{ σελίδες}$$

2) $\mathbf{Y = X \bowtie ΔΑΝΕΙΣΜΟΙ}$

Υπολογίζουμε για κάθε πλειάδα της Χ με πόσες πλειάδες της σχέσης "ΔΑΝΕΙΣΜΟΙ" κάνει join. Προφανώς, επειδή το πεδίο ΚΒ είναι πρωτεύον κλειδί στη σχέση "ΒΙΒΛΙΑ", άρα και στη Χ, θα ισχύει ότι V(X, ΚΒ) = 100. Υποθέτοντας ότι η κατανομή είναι ομοιόμορφη και πως όλες οι τιμές του πεδίου ΚΒ εμφανίζονται στην "ΔΑΝΕΙΣΜΟΙ", V(X, ΚΒ) = 50.000 η σύζευξη θα επιστρέψει:

$$\frac{T(X) * T(ΔANEIΣMOI)}{\max\{V(X,KB),\ V(ΔANEIΣMOI,\ KB)\}} = \frac{100 * 300.000}{\max\{100,\ 50.000\}}$$

$$= \frac{30.000.000}{50.000} = \mathbf{600\ εγγραφές}$$

Άρα κάθε εγγραφή της σχέσης Χ θα κάνει join κατά Μ.Ο. με $\frac{600}{100}$ =6 εγγραφές της σχέσης ΔΑΝΕΙΣΜΟΙ.

Για το INLJ ως εσωτερική σχέση επιλέγεται εκείνη στην οποία υπάρχει το ευρετήριο, δηλαδή η "ΔΑΝΕΙΣΜΟΙ". Οι εγγραφές της εξωτερικής σχέσης Χ περιέχονται σε 10 σελίδες, και άρα χωράνε στην μνήμη, επομένως δίνονται μια-μια στον αλγόριθμο και δεν προσμετράται το κόστος ανάγνωση της Χ. Για κάθε μια εγγραφή αναζητούμε στο ευρετήριο τις εγγραφές με τις οποίες κάνει join, οι οποίες θα είναι 6. Ώστόσο Επειδή το ευρετήριο είναι συστάδων, αυτές οι εγγραφές θα βρίσκονται μαζί σε $\left\lceil \frac{6}{20} \right\rceil = 1$ σελίδα, άρα διαβάζοντας 1 σελίδα παίρνουμε όλες τις εγγραφές με τις οποίες γίνεται join μια εγγραφή της Χ. Άρα τελικά το κόστος θα είναι:

$Cost(X) * 1 = 100 * 1 = \mathbf{100\ I/O}$[2]

3) $\mathbf{Y \bowtie \Delta ANEIZOMENOI}$

Έστω Y = $\Pi_{K\Delta}$(X $\bowtie$ ΔΑΝΕΙΣΜΟΙ)

Από προηγούμενο ερώτημα έχουμε βρει ότι T(Y) = 600.

Το κόστος του τελεστή προβολής είναι 0, καθώς το αποτέλεσμα μπορεί να υπολογιστεί αμέσως μετά το προηγούμενο βήμα χωρίς επιπλέον διαβάσματα από το δίσκο, Cost(Y) = 100 I/O.

Ως εξωτερική σχέση στο BNLJ επιλέγουμε τη μικρότερη, δηλαδή την Y.

Εφόσον η διαθέσιμη μνήμη είναι 20 σελίδες, θα διαβάσουμε την Y σε τμήματα των 20-1=19 σελίδων. Επιπλέον, για τη σχέση "ΔΑΝΕΙΖΟΜΕΝΟΙ" ισχύει ότι το πεδίο ΚΔ είναι πρωτεύον κλειδί, που σημαίνει ότι V(ΔΑΝΕΙΖΟΜΕΝΟΙ, ΚΔ) = 10.000, αφού για όλες τις εγγραφές θα έχει μοναδική τιμή. Οι διακριτές τιμές του ΚΔ στη σχέση Y θα είναι σίγουρα λιγότερες ή το πολύ ίσες με 10.000 δηλαδή

$V(Y, K\Delta) \leq V(\Delta ANEIZOMENOI, K\Delta)$. Με άλλα λόγια, κάθε πλειάδα της Y αντιστοιχεί σε πολύ μια πλειάδα της "ΔΑΝΕΙΖΟΜΕΝΟΙ", αφού το ΚΔ είναι πρωτεύον κλειδί σε αυτή και ξένο κλειδί στην Y. Οι πλειάδες που θα επιστραφούν από τη σύζευξη είναι:

$$T(Z) = \frac{T(Y) * T(\Delta ANEIZOMENOI)}{max\{V(Y, K\Delta), V(\Delta ANEIZOMENOI, K\Delta)\}} = \frac{600 * 10.000}{10.000} = \mathbf{600}$$

Στον BNLJ διαβάζουμε την εξωτερική σχέση Y σε τμήματα των M-1 σελίδων και για κάθε τμήμα διαβάζουμε ολόκληρη την σχέση "ΔΑΝΕΙΖΟΜΕΝΟΙ". Η συγκεκριμένη σχέση δεν έχει ευρετήριο, άρα θα διαβαστούν όλες οι σελίδες της. Το κόστος του BNLJ είναι:

$$\begin{aligned} Cost\ (Z) &= Cost(Y) + \left\lceil \frac{B(Y)}{M-1} \right\rceil * Cost(\Delta ANEIZOMENOI) = \\ &= Cost(Y) + \left\lceil \frac{B(Y)}{19} \right\rceil * B(\Delta ANEIZOMENOI) \\ &= \mathbf{100} + \left\lceil \frac{\boldsymbol{B(Y)}}{\mathbf{19}} \right\rceil * \mathbf{10.000} \end{aligned}$$

4) $\boldsymbol{\sigma_{12<Ηλικία<20}(Z)}$

Για να εκτελεστεί η επιλογή μπορεί να γίνει αναζήτηση απευθείας πάνω στη σχέση που δίνει ως αποτέλεσμα το προηγούμενο βήμα, επομένως δεν απαιτείται κάποιο έξτρα I/O. Το κόστος του βήματος αυτού είναι 0.

Στο προηγούμενο βήμα επιστρέφονται οι δανειζόμενοι που έχουν δανειστεί βιβλία που έχουν εκδοθεί από τις εκδόσεις Σαββάλα. Υποθέτοντας ξανά πως κάθε τιμή του πεδίου ηλικία περιέχεται στη σχέση Z,

V(Z, ηλικία) = 18

Οι εγγραφές που θα επιστρέψει η επιλογή είναι:

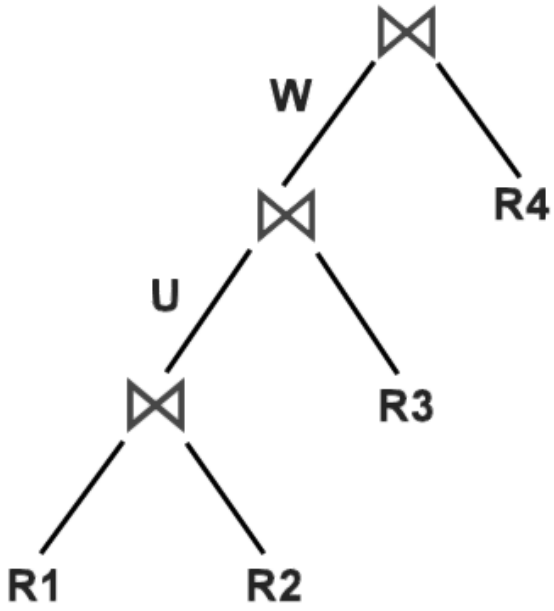
$$T(W) = (19 - 13 + 1) * \frac{T(Z)}{V(Z,Ηλικία)} = 7 * \frac{600}{18} = 234 \text{ εγγραφές}$$

Όπως και προηγουμένως, ο τελεστής επιλογής δεν έχει κάποιο επιπλέον κόστος. Άρα συνολικά το κόστος εκτέλεσης του επερωτήματος είναι

$$\mathbf{100 + \left\lceil \frac{B(Y)}{19} \right\rceil * 10.000 \text{ I/Os}}$$

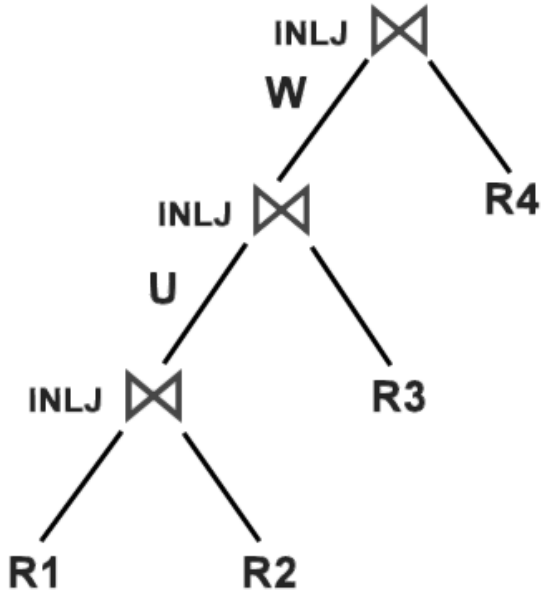
Άσκηση 5)

1. Το λογικό πλάνο αριστερού βάθους θα είναι:



2. Εφόσον έχουμε στην διάθεση μας ένα ευρετήριο συστάδων για κάθε σχέση, ο πιο αποδοτικός αλγόριθμος ισοσύνδεσης είναι ο Indexed Nested Loop Join καθώς το κόστος για ένα Join είναι $\mathrm{B(R)} + \mathrm{T(R)} * \left\lceil \frac{\mathrm{X}}{\text{πλήθος εγγραφών/σελίδα της S}} \right\rceil$, όπου X το πλήθος εγγραφών της S που γίνονται join με πλειάδες της R.
Το πλάνο είναι αποδοτικό γιατί σε κάθε βήμα (σύζευξη) προστίθεται μόνο το κόστος της ανάκτησης των πλειάδων που γίνονται join μέσω του index. Με άλλα λόγια, σε αντίθεση με τους υπόλοιπους αλγορίθμους ισοσύνδεσης, δεν χρειάζεται να διαβάσουμε ολόκληρη την εσωτερική σχέση. Η μόνη σχέση

που θα διαβαστεί ολόκληρη είναι η R1, που θα είναι η εξωτερική σχέση στην σύζευξη με την R2, για την οποία πραγματοποιείται index scan (seek). Η σχέση που προκύπτει δίνεται ως εξωτερική στον INLJ με την R3. Τότε θα πραγματοποιηθεί ξανά index scan πάνω στην R3, που προστίθεται στο κόστος του προηγούμενου υπολογισμού. Ομοίως μετά το αποτέλεσμα θα δοθεί στη σύζευξη με την R4 όπου και εκεί θα πραγματοποιηθεί ένα αποδοτικό index scan. Το φυσικό πλάνο θα είναι:



3. Το κόστος, όπως περιγράφηκε στο προηγούμενο ερώτημα θα είναι:

$$Cost(U) = B(R1) + T(R1) * \left\lceil \frac{\text{A}}{\text{πλήθος εγγραφών/σελίδα της R2}} \right\rceil$$

$$Cost(W) = Cost(U) + B(U) * \left\lceil \frac{\text{B}}{\text{πλήθος εγγραφών/σελίδα της R3}} \right\rceil$$

άρα τελικά το συνολικό κόστος:

$$Cost(X) = Cost(W) + B(W) * \left\lceil \frac{\text{C}}{\text{πλήθος εγγραφών/σελίδα της R4}} \right\rceil$$