ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ 2

ΑΣΚΗΣΗ 1:

1)

Τ(Α).Για να υπολογίσουμε το Τ(Α) θα θεωρήσουμε το οτι οι τιμές κατανέμονται ομοιόμορφα.Ψάχνουμε τον αριθμό εγγραφών εξόδου όταν σ(country=greece) στον πίνακα cities. Έχουμε 100 χώρες και 5000 πόλεις Αγού θεωρούμε ομοιόμορφη κατανομή έχουμε 5000/100=50 οπότε προκύπτει 50 πόλεις ανα χώρα.Αρα και για την ελλάδα θα επιστρέψει **50** εγγραφές.

Τ(Β).Πάλι θεωρούμε ομοιόμορφη κατανομή .Ψάχνουμε τον αριθμό εγγραφών εξόδου όταν σ(duration>=2 και duration <6) για τον πίνακα flights. Αρχικά ο πίνακας της εκφώνησης λέει ότι για duration αποτελει το 30% των εγγραφών οπότε εμείς επιλέγουμε το μισό για ψάχνουμε για =2. Μετά ο πίνακα λεει οτι πτήσεις με duration 3-4 πιάνουν το 40% που περιέχεται και αυτο το >=2 και <6.Τέλος έχουμε το κομμάτι 5-7 ώρες που αποτελεί το 15% των εγγραφών. Λογο της ομοιόμορφης κατανομής θα πάρουμε το ⅓ γιατι μας ενδιαφέρουν μόνο οι τιμές <6 οπότε 5%. Αρα προκύπτει ότι 40%+15%+5%=60% είναι το ποσοστό των εγγραφών από τις συνολικές που μας ενδιαφέρουν.Οι συνολικες εγγραφές στον πίνακα flights είναι 100.000 οπότε έχουμε 100.000*0.60=**60.000** συνολικές εγγραφές εξόδου στο Τ(Β).

Τ(Γ).Σε αυτό το ερώτημα πραγματοποιείται μεταξύ των αποτελεσμάτων των Τ(Α) και Τ(Β) με συνθήκη Cities.cid = Flights.from_cid. Στο Τ(Α) έχουμε 50 εγγραφές. Αν υποθέσουμε ότι οι πτήσεις είναι ομοιόμορφα κατανεμημένες ως προς το from_cid, τότε η πιθανότητα μία πτήση να ξεκινά από ελληνική πόλη είναι 50 / 5.000 = 0,01. Άρα από τις 60.000 εγγραφές του Τ(Β) μόνο το 1% θα έχουν from_cid ελληνική πόλη δηλαδή 60.000 *0,01 = **600** εγγραφές.

Τ(Δ). Όσον αφορά το τελευταίο ερώτημα γίνεται σύζευξη μεταξύ του αποτελέσματος του Τ(Γ) και του πίνακα Airlines στο χαρακτηριστικό aid.Το aid πρόκειται για το προτεύον κλειδί του πίνακα airlines και το ξένο κλειδί για τον πίνακα που προέκυψε από την σχέση Τ(Γ).Συμπερασματικά αφου δεν υπάρχει κάποια άλλη συνθήκη και η σύζευξη γίνεται μεταξύ ξένου κλειδιού και πρωτεύον κλειδιού δεν θα υπάρχουν αλλαγές, δηλαδή οι εγγραφές θα παραμείνουν **600.**

2)Αρχικά για αυτό το ερώτημα θα πάρουμε ότι οι εγγραφές στο fid είναι ομοιόμορφα κατανεμημένες.Οπότε το πρώτο select θα επιστρέψει 50.000 εγγραφές. Έπειτα το δεύτερο select έχει υλοποιηθεί στο ερώτημα Τ(Β) και γνωρίζουμε ότι επιστρέφει 60.000 χιλιάδες εγγραφές.Για να βρούμε το αποτέλεσμα του union θεωρούμε ότι οι δύο συνθήκες είναι ανεξάρτητες, δηλαδή δεν επηρεάζει η μια την άλλη.Οι πιθανότητα νά έχουμε fid>50.000 είναι 50.000/100.000=0.5. Η πιθανότητα το duration να είναι >=2 και <6 είναι

60.000/100.000=0.6.Άρα η πιθανότητα να υπάρχει διπλότυπη εγγραφή θα είναι 0.5*0.6*100.000=30.000
Άρα 60.000+50.000-30.000=**80.000**

ΑΣΚΗΣΗ 2:

1) ΤΟ SMJ χωρίζεται σε δύο φάσεις, της ταξινόμησης και του merge.Δηλαδή ταξινόμηση R1 Ταξινόμηση,R2 και merge R1,R2.Δεν μπορούμε να χρησιμοποιήσουμε του κλασικούς τύπους υπολογισμού κόστους καθώς το m δεν καλύπτει τις ανάγκες.Θα δούμε την κάθε φάση αναλυτικά.

R1 Ταξινόμηση:

Η R1 έχει 600 blocks,έχουμε 5 θέσεις μνήμεις οπότε 600/5=120 blocks αρχικά runs.Για να ενώσουμε τα merge έχουμε εως 4 runs (5-1).Οπότε θα πάρουμε τον τύπο $\log_4(120)=4$ φάσεις merge των runs.

Κόστος ταξινόμησης R1:

2 × B(R1) × (1 + αριθμός φάσεων merge) = 2 × 600 × (1 + 4) = **6000 I/O.**

R2 Ταξινόμηση:

Η R1 έχει 400 blocks,οπότε 400/5=80 blocks αρχικά runs.Για να ενώσουμε τα merge έχουμε εως 4 runs (5-1).Οπότε θα πάρουμε τον τύπο $\log_4(80)=4$ φάσεις merge των runs.

Κόστος ταξινόμησης R2:

2 × B(R2) × (1 + αριθμός φάσεων merge) = 2 × 400 × (1 + 4) = **4000 I/O.**

Κόστος σύζευξης (merge phase):

B(R1) + B(R2) = 600 + 400 = 1000 I/O.

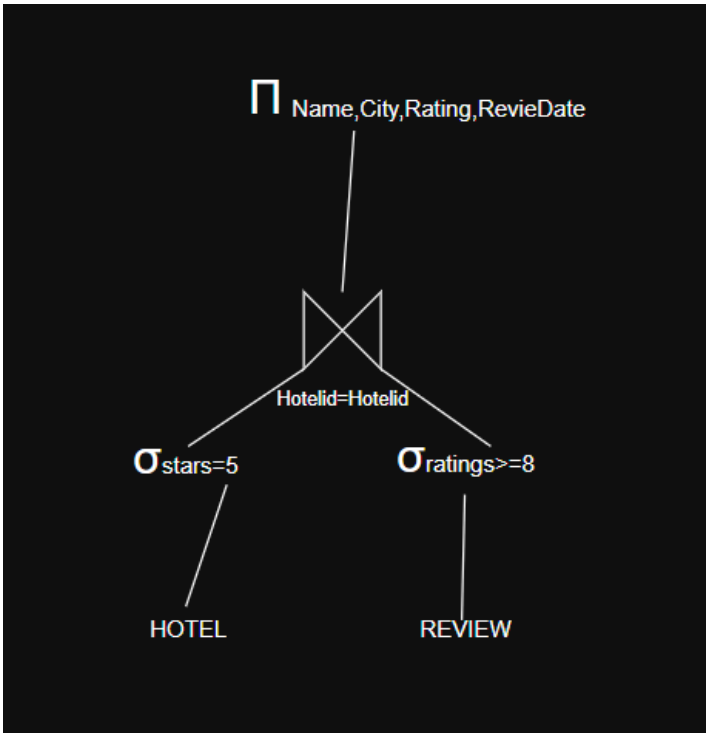
Οπότε το συνολικό κόστος I/O =4800+3200+1000=9000 I/O.

2)Θα φτιάξουμε 2 ευρετήρια. Ενα b+tree clustered index στο γνώρισμα b την σχέσης R2 και ενα b+tree clustered index στο γνώρισμα b της σχέσης R1.Επειδή είναι clustered οι σχέσεις αποθηκεύονται ταξινομημένες ως προς το b άρα η φάση της ταξινόμησης παραλείπετε.Το μόνο κόστος που υπάχει είναι του merge δηλαδή (R1) + B(R2) = 600 + 400 = 1000 I/Os.

ΑΣΚΗΣΗ 3:

Α)

Αρχικά υλοποιούμε το διάγραμμα ετσι ωστε χρησιμοποιουνται αποδοτικότερα τα ευρετήρια που έχουν δημιουργηθεί στα πεδία review και hotels.Οι επιλογές εκτελούνται νωρίς και με αυτό τον τρόπο μειώνουν τα δεδομένα.



Α)

a)

Αρχικά θα υπολογίσουμε πόσα blocks χρειάζονται για κάθε πεδίο.Γνωρίσουμε οτι τα ξενοδοχεία με stars=5 πιάνουν το 10% των συνολικών εγγραφών της σχέσης hotel δηλαδή 20.000*0.1=2000. Σε κάθε σελίδα χωράνε 50 εγγραφές.Οπότε 2000/50=40 blocks.

Θεωρούμε ότι το γνώρισμα rating της σχέσης review είναι ομοιόμορφα κατανεμημένο. Οπότε οι τιμές 8,9,10 (>=8) αποτελούν το 30% των συνολικών εγγραφών,δηλαδή 400.000*0.3=120.000.Κάθε σελίδα χωράει 500 εγγραφές της σχέσης οπότε 120.000/50=240 blocks.

Επειδή υπάρχει ήδη ένα ευρετήριο στην στον γνωρίσμα rating δεν θα χρειαστεί να κάνουμε ταξινόμηση στο review αλλά μόνο στο hotel.

Δηλαδή μας ενδιαφέρει μόνο το sort(hotel)+merge.Επειδή 21>=sqrt(40) υπολογίζουμε το κόστος του sort(hotel) ως: 3*B(hotel)=3*40=120 I/O.

και το merge έχει κόστος I/O 40+240=280. Αρά τελικό κόστος για SMJ=120+280=400 I/O

b)
Για τον BNJ:
Εξωτερική σχέση είναι η Hotels γιατί χρειάζεται λιγότερα blocks ενώ εσωτερική ή reviews.Η Hotels έχει 40 blocks ενώ η review 240. Οι διαθέσιμες μνήμες ισούνται M = 21 οπότε τα μπλοκ του HOTEL που χωράνε = M - 1 = 20.Άρα για να φορτώσει όλα τα μπλοκ χρειάζεται 2 επαναλήψεις.Σε κάθε επανάληψη χρειάζεται διαβάζει και τα 240 block της review. Οπότε 2*240=480 και διαβάζει και μία φορά τα blocks της hotels οπότε το συνολικό I/O cost είναι 480+40=520 I/O cost

ΑΣΚΗΣΗ 4:

Γνωρίζουμε ότι η σχέση υπάλληλος καταλαμβάνει 10.000 blocks στον δίσκο.
Το 10% των πλειάδων της σχέσης ικανοποιούν την συνθήκη ειδικότητα = ηλεκτρολόγος δηλαδή 10.000*0.1=1.000 βlocks.
Το 10% των πλειάδων της σχέσης ικανοποιούν τη συνθήκη Τμήμα=Παραγωγή δηλαδή 10.000*0.1=1.000.
Το 5% των πλειάδων της σχέσης ικανοποιούν και τις δύο συνθήκες ταυτόχρονα δηλαδή 10.000*0,05=500.

1)Γνωρίζουμε ότι το κόστος για το index seek πάνω σε ένα γνώρισμα όταν υπάρχει ευρετήριο συστάδων ισούται με B(X),δηλαδή με τον αριθμό τών blocks που καταλαμβάνει η σχέση X μετά την συνθήκη ειδικότητα = ηλεκτρολόγος. Στην προκειμένη περίπτωση B(X) = 1.000 για το γνώρισμα Ειδικότητα = Ηλεκτρολόγος οπότε και το I/O=1000.Ο έλεγχος της συνθήκης για το τμήμα δεν καταλαμβάνει παραπάνω I/O καθώς δεν διαβάζει παραπάνω σελίδες.

2)Στο συγκεκριμένο ερώτημα έχουμε ευρετήριο συστάδων και για τα δύο γνωρίσματα που ικανοποιούν τις συνθήκες οπότε το ποσοστό των block που καταλαμβάνουν είναι 500.Πάλι το κόστος I/O ισούνται με το B(X) που στην προκειμένη περίπτωση είναι 500 οπότε Κόστος I/O=500.

3)Πάλι στο συγκεκριμένο υποερώτημα το ευρετήριο συστάδων συμπεριλαμβάνει και τα δυο γνωρισματα που μας ενδιαφέρουν και με σωστή σειρά.Οπότε πάλι το κόστος ισούνται με τον αριθμό των block που καταλαμβάνει η σχέση δηλαδή του B(X) . B(X)=500 οπότε Κόστος I/O = 500. Το ότι υπάρχει μέσα στο ευρετήριο το γνώρισμα επώνυμο δεν επηρεάζει κάπως το κόστος γιατί είναι τελευταίο σε σειρά και δεν μας απασχολεί στο ερώτημα μας

4)Το ευρετήριο στο συγκεκριμένο ερώτημα δεν βοηθάει κάπως καθώς πρόκειται για clustered και επώνυμο είναι πρώτο οπότε θα χρειαστεί να κάνει index scan δηλαδή να σαρώσει όλο τον πίνακα για να βρει τις πλειάδες που ικανοποιούν τις ανάλογες συνθήκες. Για το index scan γνωριζουμε οτι το I/O cost ισούται με το σύνολο των blocks του πίνακα οπότε I/O cost=10.000.