ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΥΕ041 - ΠΛΕ081: Διαχείριση Σύνθετων Δεδομένων

(ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2017-18)

ΕΡΓΑΣΙΑ 1 – Αλγόριθμοι Αποτίμησης Συνενώσεων

Προθεσμία: 21 Μαρτίου 2018, 9μ.μ.

Στόχος της εργασίας είναι η ανάπτυξη και ο έλεγχος αλγορίθμων για αποτίμηση (evaluation) σύνθετων ερωτημάτων σε βάσεις δεδομένων τα οποία περιλαμβάνουν συνενώσεις (joins).

Μέρος 1 (γραπτό)

Θεωρείστε τους παρακάτω τελεστές σχεσιακής άλγεβρας. Περιγράψτε πως μπορεί ο καθένας από αυτούς να αποτιμηθεί (1) με χρήση ταξινόμησης (sorting) και (2) με χρήση κατακερματισμού (hashing).

a. Semijoin: παίρνει σαν είσοδο δύο σχέσεις r και s και επιστρέφει το σύνολο των πλειάδων στην r οι οποίες συνενώνονται με τουλάχιστο μια πλειάδα στο s. Για παράδειγμα θεωρείστε ότι η r έχει σχήμα R(A,B), η s έχει σχήμα S(A,C), η r περιέχει τις πλειάδες {(1,2), (1,4), (2,5)} και η s έχει τις πλειάδες {(1,‘a’), (1,‘c’), (3,‘a’)}. Το αποτέλεσμα του semijoin(r,s) είναι {(1,2), (1,4)} γιατί για καθεμιά από αυτές τις πλειάδες υπάρχει τουλάχιστον μια πλειάδα στην s η οποία να συνενώνεται με αυτή.
b. Anti-semijoin: παίρνει σαν είσοδο δύο σχέσεις r και s και επιστρέφει το σύνολο των πλειάδων στην r οι οποίες δεν συνενώνονται με καμιά πλειάδα στο s. Για παράδειγμα θεωρείστε ότι η r έχει σχήμα R(A,B), η s έχει σχήμα S(A,C), η r περιέχει τις πλειάδες {(1,2), (1,4), (2,5)} και η s έχει τις πλειάδες {(1,‘a’), (1,‘c’), (3,‘a’)}. Το αποτέλεσμα του antisemijoin(r,s) είναι {(2,5)} γιατί για καθεμιά από αυτές τις πλειάδες δεν υπάρχει πλειάδα στην s η οποία να συνενώνεται με αυτή. Παρατηρήστε ότι antisemijoin(r,s) = r - semijoin(r,s)

Στις περιγραφές σας θα πρέπει να εξηγήσετε με σαφήνεια πως μπορούμε να αλλάξουμε τους αλγορίθμους sort-merge join και hash-join που μάθαμε ώστε να αποτιμούν τον καθένα από τους τελεστές. Δώστε λεπτομερή ψευδοκώδικα και θεωρείστε ότι οι σχέσεις r και s είναι αποθηκευμένες σε αρχεία χωρισμένα σε blocks.

Μέρος 2 (προγραμματιστικό)

Γράψτε ένα πρόγραμμα, το οποίο θα εφαρμόζει τον τελεστή semijoin πάνω σε 2 σχέσεις αφού έχει προηγηθεί επιλογή (selection) σε μια από αυτές. Τα δεδομένα που θα χρησιμοποιήσετε (δεδομένα πτήσεων) μπορείτε να τα κατεβάσετε από το ecourse. Κατεβάστε

τα αρχεία airports.dat και routes.dat από το ecourse. Λεπτομερείς περιγραφές των δεδομένων αυτών των αρχείων μπορείτε να βρείτε στο https://openflights.org/data.html . Προσοχή, μην κατεβάσετε τα αρχεία από το σύνδεσμο γιατί αυτά που υπάρχουν στο ecourse είναι «καθαρισμένα».

Στο αρχείο airports.dat κάθε γραμμή αντιστοιχεί σε ένα αεροδρόμιο. Το πεδίο που μας ενδιαφέρει κυρίως είναι το 1ο πεδίο, το οποίο είναι ο κωδικός του αεροδρομίου. Στο αρχείο routes.dat κάθε γραμμή αντιστοιχεί σε μια διαδρομή (δηλ. πτήση), όπου αναγράφονται μεταξύ άλλων οι κωδικοί των αεροδρομίων αναχώρησης και άφιξης. Το τελευταίο πεδίο είναι ο τύπος του αεροσκάφους.

Γράψτε ένα πρόγραμμα, το οποίο θα παίρνει σαν παράμετρο (command-line argument) τον τύπο ενός αεροσκάφους t και τυπώνει κάθε αεροδρόμιο (όλη την πλειάδα) για το οποίο υπάρχει τουλάχιστον μια διαδρομή που γίνεται με αεροσκάφος τύπου t και έχει προορισμό το αεροδρόμιο. Για παράδειγμα υπάρχουν 34 αεροδρόμια στα οποία πετούν σκάφη τύπου SU9 ενώ υπάρχουν 517 στα οποία πετούν σκάφη τύπου 737. Το κοινό πεδίο της συνένωσης είναι το 1ο από το αρχείο airports.dat και το 6ο από το αρχείο routes.dat (κωδικός αεροδρομίου προορισμού).

Το πρόγραμμά σας πρέπει να υλοποιεί παραλλαγή του sort-merge join αλγορίθμου. Μπορείτε να επωφεληθείτε από το γεγονός ότι στο αρχείο airports.dat τα αεροδρόμια είναι ήδη ταξινομημένα με βάση τον κωδικό τους (1ο πεδίο) αλλά θα πρέπει να ταξινομήσετε τις διαδρομές που ικανοποιούν τον περιορισμό του τύπου αεροσκαφών. Αν το πρόγραμμά σας δεν ακολουθεί τη λογική του merge-join θα αφαιρεθούν βαθμοί.

Μέρος 3 (γραπτό)

Η μέθοδος pipelining χρησιμοποιείται για την αποφυγή της προσωρινής αποθήκευσης ενδιάμεσων αποτελεσμάτων σε σύνθετα ερωτήματα. Έστω ότι πρέπει να υπολογίσουμε την συνένωση f με σχήμα F(A,B,C,D) τριών σχέσεων r,s,t με σχήματα R(A,B), S(A,C), T(A,D). Αν οι σχέσεις r, s, t είναι ήδη ταξινομημένες ως προς το πεδίο A, περιγράψτε με ποιο τρόπο μπορεί να εφαρμοστεί pipelining σε συνδυασμό με τον merge-join αλγόριθμο για τον υπολογισμό του αποτελέσματος (π.χ. υπολογίζοντας πρώτα το join(r,s) και μετά το join του αποτελέσματος με το t). Επίσης, προτείνετε έναν sort-merge αλγόριθμο ο οποίος εφαρμόζεται σε τρεις εισόδους ταυτόχρονα (concurrently) και δίνει απευθείας το αποτέλεσμα f. Περιγράψτε τα βήματα του αλγορίθμου αυτού.