ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΥΕ041 - ΠΛΕ081: Διαχείριση Σύνθετων Δεδομένων

(ΕΑΡΙΝΟ ΕΞΑΜΗΝΟ 2023-24)

ΕΡΓΑΣΙΑ 1 – Αποτίμηση ερωτημάτων

Προθεσμία: 29 Μαρτίου 2024

Ζητείται να υλοποιήσετε προγράμματα που αποτιμούν τελεστές και ερωτήματα σε μια μικρή βάση δεδομένων με δύο πίνακες R και S, οι οποίοι αποθηκεύονται στα αρχεία R.csv και S.csv αντίστοιχα. Οι πίνακες αυτοί έχουν τρία πεδία (attributes) ο καθένας με αριθμητικά πεδία τιμών (ακέραιοι). Τα αρχεία που αποθηκεύουν τα δεδομένα των πινάκων μπορείτε να τα κατεβάσετε από το ecourse. Για την υλοποίηση μπορείτε να χρησιμοποιήσετε γλώσσα της επιλογής σας (C, C++, Java, Python, κλπ.)

Μέρος 1 (group-by with aggregation): Ζητείται ένα πρόγραμμα, το οποίο υπολογίζει το αποτέλεσμα τελεστών συνάθροισης σε έναν πίνακα με χρήση ταξινόμησης. Το πρόγραμμά σας θα πρέπει να παίρνει σαν **ορίσματα από τη γραμμή διαταγών** (command-line arguments) το όνομα του αρχείου το οποίο περιέχει τον πίνακα (R.csv ή S.csv), ένα attribute ομαδοποίησης (0, 1, ή 2, όπου 0 είναι το πρώτο πεδίο, 1 το δεύτερο πεδίο, κλπ.), ένα attribute όπου εφαρμόζεται συνάθροιση (0, 1, ή 2) και την συνάρτηση συνάθροισης (sum, min, ή max). Το αποτέλεσμα είναι ένας πίνακας με πλειάδες της μορφής (x,y), όπου x είναι μία τιμή του attribute ομαδοποίησης και y είναι το αποτέλεσμα της συνάρτησης συνάθροισης πάνω σε όλες τις τιμές του attribute συνάθροισης που περιλαμβάνονται σε πλειάδες του πίνακα εισόδου οι οποίες έχουν τιμή x στο attribute ομαδοποίησης. Για παράδειγμα, αν τα ορίσματα στη γραμμή διαταγής είναι `R.csv 1 2 max`, αυτό αντιστοιχεί στην παρακάτω ερώτηση σε SQL:

```
SELECT R.1, MAX(R.2)
FROM R
GROUP BY R.1
```

Το πρόγραμμά σας θα πρέπει να διαβάζει εξ ολοκλήρου το αρχείο του πίνακα εισόδου στην κύρια μνήμη. Κατόπιν θα πρέπει να εκτελεί μια παραλλαγή του αλγορίθμου MergeSort (https://en.wikipedia.org/wiki/Merge_sort), ο οποίος ταξινομεί τον πίνακα με βάση το attribute ομαδοποίησης και ταυτόχρονα υπολογίζει τη συνάθροιση. Ο αλγόριθμος αντί να κάνει πλήρη ταξινόμηση του πίνακα, (α) αγνοεί το attribute που δεν είναι ούτε το attribute ομαδοποίησης ούτε το attribute συνάθροισης και δεν το συμπεριλαμβάνει στους ενδιάμεσους πίνακες συγχώνευσης και (β) εφαρμόζει τη συνάθροιση κατά την συγχώνευση, δηλαδή αν δύο πλειάδες που συγκρίνονται έχουν την ίδια τιμή στο attribute ομαδοποίησης τότε συγχωνεύονται σε μία πλειάδα στην έξοδο. Εφαρμόστε δυαδική συγχώνευση και όχι M-way merging όπως στις διαφάνειες. Τα αποτέλεσματα της συνάθροισης θα πρέπει να γραφτούν σε ένα αρχείο O1.csv.

Για παράδειγμα, οι πρώτες 5 γραμμές του αρχείου O1.csv μετά την εκτέλεση του προγράμματος με ορίσματα `R.csv 1 2 max` είναι:

```
1,10
2,10
3,10
4,9
5,10
```

Μέρος 2 (merge join): Θεωρείστε ότι το σχήμα του πίνακα R στο αρχείο R.csv είναι (A,B,C) και το σχήμα του πίνακα S στο αρχείο S.csv είναι (D,A,E). Δηλαδή, το attribute 0 (πρώτη στήλη) στον πίνακα R ονομάζεται A, το attribute 1 (δεύτερη στήλη) ονομάζεται B, κλπ. Το R.A είναι primary key (πρωτεύον κλειδί) στον R και το S.A στο S είναι foreign key (ξένο κλειδί) που «δείχνει» στον πίνακα R. Παρατηρείστε ότι και τα δύο αρχεία είναι ταξινομημένα ως προς το πεδίο A. Γράψτε ένα πρόγραμμα το οποίο διαβάζει ταυτόχρονα τα δύο αρχεία και παράγει την φυσική συνένωση (natural join) τους με σχήμα (A,B,C,D,E). Το πρόγραμμα θα πρέπει να χρησιμοποιεί τη λογική του merge-join αλγορίθμου και θα πρέπει να δημιουργεί και να εμπλουτίζει το αρχείο εξόδου O2.csv με τα περιεχόμενα της συνένωσης. Για παράδειγμα, οι 5 πρώτες γραμμές του O2.csv είναι

```
45,41,7,1,3
45,41,7,2,9
45,41,7,3,7
45,41,7,4,6
45,41,7,5,4
```

Το πρόγραμμά σας **δεν θα πρέπει να διαβάζει εξ ολοκλήρου** το R.csv ή/και το S.csv πριν αρχίσει να παράγει αποτελέσματα.

Μέρος 3 (composite query): Στο τρίτο μέρος, θα γράψετε ένα πρόγραμμα το οποίο αποτιμά μία σύνθετη ερώτηση με πολλούς τελεστές. Οι ερώτηση είναι ${}_{\text{sum(E)}}\gamma_{\text{A}}\ (\sigma_{\text{C=7}}(\text{R}) \bowtie \text{S})$ ή σε SQL:

```
SELECT S.A, SUM(S.E)
FROM R, S
WHERE R.A = S.A AND R.C = 7
GROUP BY S.A
```

Όπως και στο Μέρος 2, θεωρείστε ότι το σχήμα του πίνακα R στο αρχείο R.csv είναι (A,B,C) και το σχήμα του πίνακα S στο αρχείο S.csv είναι (D,A,E). Το R.A είναι primary key στο R και το S.A είναι foreign key που δείχνει στον πίνακα R. Το πρόγραμμά σας θα πρέπει να διαβάσει ταυτόχρονα τα δύο αρχεία και να παράξει το αποτέλεσμα της ερώτησης **με ένα μόνο πέρασμα** πάνω από τα αρχεία. Θα πρέπει να δημιουργήσει και να εμπλουτίσει το αρχείο εξόδου O3.csv με τα αποτελέσματα της ερώτησης. Για παράδειγμα, οι 5 πρώτες γραμμές του O3.csv είναι

```
45,63
47,44
78,60
187,70
501,55
```

Το πρόγραμμά σας **δεν θα πρέπει να διαβάζει εξ ολοκλήρου** το R.csv ή/και το S.csv πριν αρχίσει να παράγει αποτελέσματα.

Παραδοτέα:

Βάλτε σέ ένα zip αρχείο τα προγράμματά σας και ένα PDF αρχείο, το οποίο θα περιέχει πληροφορίες για τα προγράμματά σας. Υποβάλετε το zip αρχείο σας μέσω turnin στο assignment1@mye041

Οδηγίες για τις υποβολές:

1) Αν χρησιμοποιήσετε Java, το πρόγραμμά σας θα πρέπει να γίνεται compile και να τρέχει και εκτός Eclipse στους υπολογιστές του εργαστηρίου. **Μην χρησιμοποιείτε packages**.
2) Αν χρησιμοποιήσετε Python, μην χρησιμοποιήστε τη βιβλιοθήκη pandas και μην υποβάλετε κώδικα για interactive programming (π.χ. ipython)
3) Υποβάλετε τις εργασίες σας σε ένα **zip** αρχείο (**όχι rar**) το οποίο πρέπει να περιλαμβάνει όλους τους κώδικες καθώς και ένα αρχείο τεκμηρίωσης το οποίο να περιγράφει τη μεθοδολογία σας και να περιλαμβάνει το PDF αρχείο. **Μην υποβάλετε αρχεία δεδομένων**.
4) Μην ξεχνάτε να βάζετε το όνομά σας (σε greeklish) και το ΑΜ σε κάθε αρχείο που υποβάλετε.
5) Ο έλεγχος των προγραμμάτων σας μπορεί να γίνει σε άλλα αρχεία εισόδου από αυτά που σας δίνονται, άρα θα πρέπει ο κώδικάς σας να μην εξαρτάται από τα συγκεκριμένα αρχεία εισόδου που σας δίνονται.