ΜΥΕ041 - ΠΛΕ081: Διαχείριση Σύνθετων Δεδομένων

ΕΡΓΑΣΙΑ 1 – Αποτίμηση ερωτημάτων

Νεφέλη-Ελένη Κατσιλέρου

ΑΜ : 4385

Μέρος 1 (group-by with aggregation)

Αρχικά απο τη γραμμή εντολών παίρνουμε το input_file του csv αρχείου που θέλουμε ως είσοδο καθως επίσης τα attributes για την ταξινόμηση και τη συνάθροιση και τη συνάρτηση aggregation_function (max, min, sum). Μετά καλώ τη συνάρτηση group_by_aggregation.

Η συνάρτηση group_by_aggregation(input_file, clustering_attribute, aggregation_attribute, aggregation_function) διαβάζει δεδομένα από το αρχείο csv που δίνεται μέσω του input_file και τα μετατρέπει σε μια λιστα απο ints. Διαβάζει κάθε γραμμή του αρχείου csv , τη μετατρέπει σε μια λίστα ακεραίων και την βάζει στη λιστα data. Στη συνέχεια καλείται η συνάρτηση my_merge_sort με τα data αυτά και τα υπόλοιπα ορίσματα που απαιτούνται.

Η συνάρτηση my_merge_sort(data, clustering_attribute, aggregation_attribute, aggregation_function) χρησιμοποιειται για να χωριζει το συνολο απο τα data που δινονται ως input σε 2 υποσυνολα left και right. Μετα καλει αναδρομικα τον εαυτο της για καθε υποσυνολο μεχρι τα δεδομένα να έχουν μόνο ένα στοιχείο ή ειναι κενα. Τοτε καλει την my_merge για να κανει απο “κατω” προς τα “πανω” merge τα left και right υποσυνολα.

Στη συνάρτηση my_merge(left, right, clustering_attribute, aggregation_attribute, aggregation_function) έχω αρχικοποιήσει το i που είναι ο counter για το left και το j που ειναι ο counter για το j, το result=[] είναι μια άδεια λίστα όπου θα αποθηκεύσω το merge αποτέλεσμα και το aggregation_dictionary = {} είναι ένα λεξικό για να αποθηκεύσω aggregation values που αντιστοιχούν σε κάθε clustering attribute. Τα κλειδιά του λεξικού αντιπροσωπεύουν μοναδικά clustering attribute values. Οι τιμές του λεξικού είναι λίστες που περιέχουν τις αντίστοιχες τιμές των aggregation attribute. Οι τιμές αυτές συμπληρώνονται με βάση τις τιμές των clusterring attribute που γίνονται count.

Στη συνέχεια επαναληπτικά περνάει και απο τα δυό μισά και συγκρίνει τα clustering attributes για να κάνει κατάλληλα merge. Ανακτά τις τιμές των clustering attributes απο το αριστερό και το δεξί μισό για σύγκριση και ανάλογα με τη σύγκριση αποφασίζει ποιά στοιχεία θα γίνουν merge. Αφού ολοκληρωθεί η διαδικασία του merge , οι τιμές συνάθροισης υπολογίζονται απο τις λίστες του λεξικού και εγγράφονται στο ζητούμενο αρχείο εξόδου (O1.csv).

Μέρος 2(merge join)

Αρχικά ορίζω τη συνάρτηση my_merge_join που λαμβάνει τρεις παραμέτρους: R_file, S_file και output_file. Αυτές οι παράμετροι αντιπροσωπεύουν τις διαδρομές προς τα δύο αρχεία CSV εισόδου R_file και S_file και το αρχείο CSV εξόδου output.file, όπου θα γραφτεί το αποτέλεσμα της λειτουργίας merge join. Στη συνέχεια ανοίγω τα αρχεία εισόδου για ανάγνωση και το αρχείο εξόδου για εγγραφή. Τα R_file, S_file και output ειναι αντικείμενα αρχείων. Δημιουργώ τα R_reader και S_reader που είναι αντικείμενα αναγνωσης csv για την αναάγνωση δεδομένων απο τα αρχεία εισόδου και ένα αντικείμενο εγγραφης csv writer για την εγγραφή δεδομένων στο αρχείο εξόδου. Η csv.reader() χρησιμοποιείται για την ανάγνωση δεδομένων CSV από τα αρχεία εισόδου και η csv.writer() χρησιμοποιείται για την εγγραφή δεδομένων CSV στο αρχείο εξόδου. Έπειτα χρησιμοποιώ τη συνάρτηση next() για να διαβάσω τη πρώτη γραμμή δεδομένων απο κάθε αρχείο εισόδου. Για όσο η R_row και S_row υπάρχουν(δηλαδη για όσο υπάρχουν δεδομένα προς ανάγνωση απο ένα απο τα αρχεία εισόδου), εάν οι τιμες των κλειδιών απο τα R_file και S_file είναι ίσες γράφει μια νέα γραμμή στο αρχείο εξόδου με τη συνένωση των τρέχων γραμμών των R_file και S_file παραλείποντας την δευτερη στήλη του S_file. Αφού γράψει τη γραμμή , προχωρά στην επόμενη γραμμή του S_file χρησιμοποιώντας next(S_reader,None). Εάν η τιμή του κλειδιού απο το R_file είναι μικρότερη απο την τιμή του κλειδιού απο το S_file τότε προχωράει στην επόμενη γραμμή του αρχείου R_file χρησιμοποιώντας next(R_reader,None). Αλλιώς αν η τιμή του κλειδιού απο το R_file ειναι μεγαλύτερη απο την τιμή του κλειδιού απο το S_file τότε προχωράει στην επόμενη γραμμή του αρχείου S_file χρησιμοποιώντας next(S_reader,None). Παρακάτω αφού βγεί απο το while , εάν έχουν μείνει γραμμές στο R_file τις γράφει στο αρχειο εξοδου με κενά πεδία για τις στήλες απο το S_file. Αντίστοιχα η επόμενη while εάν έχουν μείνει γραμμές στο S_file τις γράφει στο αρχειο εξοδου με κενά πεδία για τις στήλες απο το R_file. Ουσιαστικά εκτελώ ενα merge join σε δυο αρχεια csv με βάση τα ακέραια κλειδιά που βρίσκονται στην πρώτη στήλη κάθε αρχείου και γράφω τα αποτελέσματα σε ένα άλλο αρχείο εξόδου.

Μέρος 3(composite query)

Αρχικά ορίζω τη συνάρτηση my_composite_query που λαμβάνει τρεις παραμέτρους: R_file, S_file και output_file. Αυτές οι παράμετροι αντιπροσωπεύουν τις διαδρομές προς τα δύο αρχεία CSV εισόδου R_file και S_file και το αρχείο CSV εξόδου output.file, όπου θα γραφτεί το αποτέλεσμα. Αρχικοποιώ ενα κενό λεξικό sum_e_dictionary για να αποθηκευσω τα αθροισματα των τιμων E για καθε ξεχωριστη τιμη A που βρίσκεται στο αρχείο S.csv. Στη συνέχεια ανοίγω τα αρχεία εισόδου για ανάγνωση και το αρχείο εξόδου για εγγραφή. Αρχικοποιώ ενα csv αντικείμενο writer για εγγραφη στο αρχείο εξοδου. Οι γραμμες R_line και line_s αρχικοποιούνται διαβάζοντας την πρώτη γραμμή απο το R_file και το S_file αντιστοιχα χρησιμοποιωντας την readline().Αρχικοποιώ και μια κενή λίστα toadd για να αποθηκεύσω την τιμές απο το R_file βάση της συνθήκης. Για όσο η R_line και S_line υπάρχουν(δηλαδη για όσο υπάρχουν δεδομένα προς ανάγνωση απο ένα απο τα αρχεία εισόδου) θα είναι μέσα στο βρόχο. Με τα R_items = R_line.split(',') και S_items =

S_line.split(',') χωρίζω τις γραμμες που διαβάζονται απο τα αρχεία εισόδου σε λίστες στοιχείων χρησιμοποιώντας την split().Έπειτα εξάγω συγκεκριμένες τιμές απο τις split lines. Η μεταβλητή a_r αντιπροσωπευει την τιμη του πρώτου στοιχείου απο το R_file ως int. Η μεταβλητή c αντιπροσωπευει την τιμη του τρίτου στοιχείου απο το R_file ως int. Η μεταβλητή a_s αντιπροσωπευει την τιμη του δεύτερου στοιχείου απο το S_file ως int. Η μεταβλητή e αντιπροσωπευει την τιμη του τρίτου στοιχείου απο το S_file ως int. Παρακάτω ενημερώνω το sum_e_dictionary. Ελέγχω αν το κλειδί a_s υπάρχει ήδη στο λεξικό και αν όχι αρχικοποιώ την τιμή σε 0. Στη συνέχεια προσθέτω την τιμή του e στην υπάρχουσα τιμή που ειναι associated με το κλειδι a_s στο λεξικό. Εάν η τιμή του c απο το R_file ειναι ίση με 7 προσθέτω την τιμή της a_r στη λιστα toadd. Διαβάζω τις επόμενες γραμμές απο τα αρχεία εισόδου προχωρώντας έτσι στην επόμενη επανάληψη του while.Με τη for για κάθε στοιχείο της toadd ελέγχω αν το x υπάρχει ως κλειδί στο λεξικό. Εαν το x υπάρχει στο λεξικό γραφει μια νέα γραμμη στο αρχείο εξόδου που περιέχει την τιμή του x και το αθροισμα που σχετίζεται με αυτό στο λεξικό.Ουσιαστικα δηλαδη η συνάρτηση διαβάζει τιμές και απο τα δυο αρχεια, υπολογίζει αθροισματα με βάση τις συνθήκες και γράφει τα αποτελέσματα σε ένα αρχείο εξόδου.