

Τμήμα Πληροφορικής & Τηλεπικοινωνιών

Κ23α - Ανάπτυξη Λογισμικού Για Πληροφοριακά Συστήματα

Χειμερινό Εξάμηνο 2019 – 2020

Καθηγητής Ι. Ιωαννίδης

Άσκηση 2 – Παράδοση: Κυριακή 15 Δεκεμβρίου 2019

Στο δεύτερο μέρος της εργασίας θα ασχοληθούμε με τη σύζευξη (join) και την εφαρμογή φίλτρου σε επερωτήσεις (queries) στις οποίες συμμετέχουν περισσότεροι πίνακες με αρκετές στήλες. Συγκεκριμένα θα υλοποιηθεί η ανάλυση επερωτήσεων που χρησιμοποιήθηκε στον προγραμματιστικό διαγωνισμό SIGMOD 2018. Στη συνέχεια αναλύονται τα αρχεία εισόδου και η γλώσσα επερωτήσεων που χρησιμοποιείται.

Διαδικασία

Το πρόγραμμα στην αρχή θα τροφοδοτείται με το σύνολο των σχέσεων (πινάκων) στο standard input. Θα λαμβάνει δηλαδή πολλές γραμμές (διαχωριζόμενες με το χαρακτήρα newline '\n'), όπου κάθε γραμμή θα περιέχει ένα αλφαριθμητικό string που θα αναπαριστά το όνομα αρχείου της συγκεκριμένης σχέσης. Τα αρχεία σχέσεων θα βρίσκονται ήδη σε δυαδική μορφή και δεν απαιτούν parsing.

Η δυαδική μορφή μιας σχέσης αποτελείται από την επικεφαλίδα και τα δεδομένα. Η επικεφαλίδα περιέχει τον αριθμό των γραμμών (tuples) και τον αριθμό των στηλών (columns). Το τμήμα των δεδομένων ακολουθεί την επικεφαλίδα και αποθηκεύει όλες τις γραμμές χρησιμοποιώντας αποθήκευση κατά στήλες. Με τον τρόπο αυτό, όλες οι τιμές μιας στήλης αποθηκεύονται στη σειρά, στη συνέχεια ακολουθούν οι τιμές της επόμενης στήλης, κ.ο.κ. Η γενική δυαδική μορφή είναι η εξής, όπου το σύμβολο T0C0 σημαίνει γραμμή (tuple) 0, στήλη (column) 0' και το σύμβολο της σωλήνωσης δεν είναι τμήμα της δυαδικής μορφής):

```
uint64_t numTuples|uint64_t numColumns|uint64_t T0C0|uint64_t T1C0|..|uint64_t  
TnC0|uint64_t T0C1|..|uint64_t TnC1|..|uint64_t TnCm
```

Μετά την αποστολή του συνόλου των σχέσεων, το πρόγραμμα θα περιμένει να λάβει μία γραμμή που θα περιέχει το αλφαριθμητικό "Done".

Οι επερωτήσεις θα έρχονται σε ομάδες (batches). Μία ομάδα επερωτήσεων περιέχει ένα σύνολο από επερωτήσεις σύζευξης (join query) (κάθε γραμμή περιέχει μία επερώτηση). Μια επερώτηση σύζευξης αποτελείται από τρία διαδοχικά μέρη, τα οποία διαχωρίζονται από το σύμβολο σωλήνωσης '|':

- **Σχέσεις.** Μία λίστα από τις σχέσεις που θα συζευχθούν. Θα δίνονται οι ταυτότητες των σχέσεων, διαχωρισμένες με κενά διαστήματα (' '). Οι ταυτότητες των σχέσεων

δίνονται από τη σειρά με την οποία περάστηκαν οι σχέσεις κατά την πρώτη φάση. Για παράδειγμα, η ταυτότητα 0 αναφέρεται στην πρώτη σχέση.

- **Κατηγορήματα (predicates).** Κάθε κατηγορήμα διαχωρίζεται με το σύμβολο '&'. Υπάρχουν δύο τύποι κατηγορήματος: κατηγορήμα φίλτρο και κατηγορήμα σύζευξης. Τα κατηγορήματα φίλτρου είναι της μορφής: στήλη φίλτρου + τύπος σύγκρισης (δηλαδή μεγαλύτερο '>', μικρότερο '<', ίσο '=') + ακέραιος αριθμός. Τα κατηγορήματα σύζευξης καθορίζουν τις στήλες στις οποίες οι σχέσεις θα συζευχθούν. Ένα κατηγορήμα σύζευξης αποτελείται από τις στήλες στις συζευγόμενες σχέσεις, συνδεόμενες με το σύμβολο ισότητας ('='). Στα κατηγορήματα η ταυτότητα της σχέσης είναι η σειρά με την οποία ταυτοποιείται στη λίστα των σχέσεων που πρόκειται να συζευχθούν (εμμέσως ορίζεται ως σχέση 0 στα κατηγορήματα η πρώτη σχέση σε μία επερώτηση σύζευξης, σχέση 1 η επόμενη κ.ο.κ.)
- **Προβολές.** Μία λίστα από στήλες που απαιτούνται για να υπολογιστούν τα αθροίσματα ελέγχου της επερώτησης. Παρομοίως με τα κατηγορήματα σύζευξης, οι στήλες αναφέρονται ως ζεύγη σχέσης-στήλης. Οι προβολές διαχωρίζονται με το χαρακτήρα κενού διαστήματος (' ').

Παράδειγμα: "0 2 4|0.1=1.2&1.0=2.1&0.1>3000|0.0 1.1"

Μετάφραση σε SQL:

SELECT SUM("0".c0), SUM("1".c1)

FROM r0 "0", r2 "1", r4 "2"

WHERE 0.c1=1.c2 and 1.c0=2.c1 and 0.c1>3000

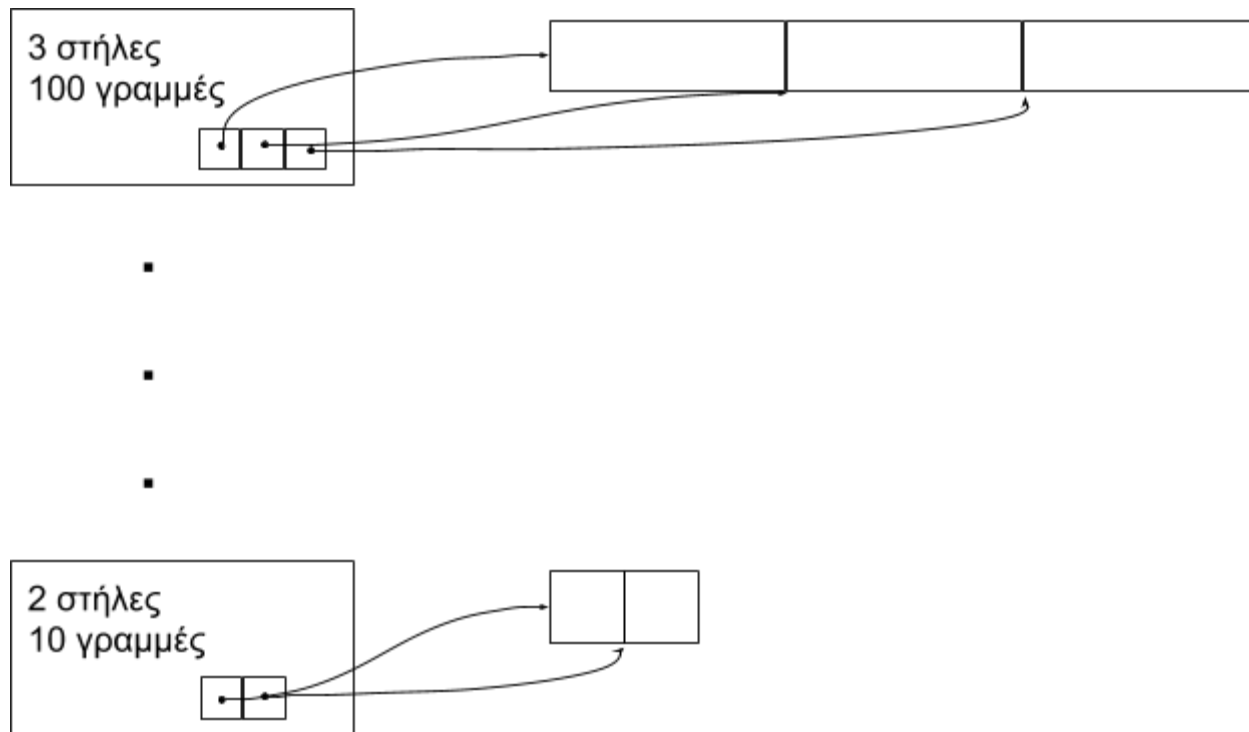
Το τέλος μιας ομάδας επερωτήσεων καθορίζεται με μία γραμμή που περιέχει τον χαρακτήρα 'F'. Το πρόγραμμα στη συνέχεια θα τυπώνει τα αποτελέσματα στο standard output. Για κάθε επερώτηση, το πρόγραμμα θα πρέπει να τυπώνει μία γραμμή που να περιέχει τα αθροίσματα ελέγχου των προβολών διαχωριζόμενα με κενά διαστήματα (π.χ. "42 4711"). Αν δεν υπάρχει γραμμή που να ικανοποιεί τα κατηγορήματα, κάθε άθροισμα ελέγχου θα πρέπει να τυπώνει το αλφαριθμητικό "NULL", όπως στην SQL. Όταν τυπωθούν τα αποτελέσματα, θα δοθεί η επόμενη ομάδα επερωτήσεων.

Τα αθροίσματα ελέγχου δεν θα υπερχειλίζουν αν χρησιμοποιηθούν 64 bit unsigned integers.

Αποθήκευση Σχέσεων στην Μνήμη

Το πρώτο πράγμα που καλείστε να διαχειριστείτε είναι η αποθήκευση των δεδομένων στην μνήμη. Όπως περιγράφηκε, στην αρχή σας έρχεται ένα πλήθος από file paths, το κάθε ένα από τα οποία έχει τα δεδομένα κάθε σχέσης. Τα δεδομένα κάθε σχέσης θα πρέπει να διαβαστούν και να αποθηκευτούν στη μνήμη. Για τον λόγο αυτό θα χρειαστεί να κρατάτε ένα πίνακα με τόσα κελιά όσες και οι σχέσεις που σας έρχονται αρχικά. Κάθε κελί αυτού του πίνακα θα κρατάει μια μεταπληροφορία όπως ο αριθμός των γραμμών αυτού του πίνακα και ο αριθμός των στηλών. Επίσης θα κρατάει και έναν πίνακα που θα έχει δείκτες στην μνήμη προς της

θέσης που έχει αποθηκευτεί η κάθε στήλη του. Το πρώτο κελί του πίνακα θα πρέπει να κρατάει την πληροφορία για την πρώτη σχέση το δεύτερο για την δεύτερη κοκ. Όταν αναφερόμαστε στην σχέση 0 θα αναφερόμαστε στην σχέση που έχει αποθηκευτή στο κελί 0 κοκ. Η αναπαράσταση αυτής της δομής φαίνεται στο πιο κάτω σχήμα.



Εκτέλεση Κατηγορημάτων

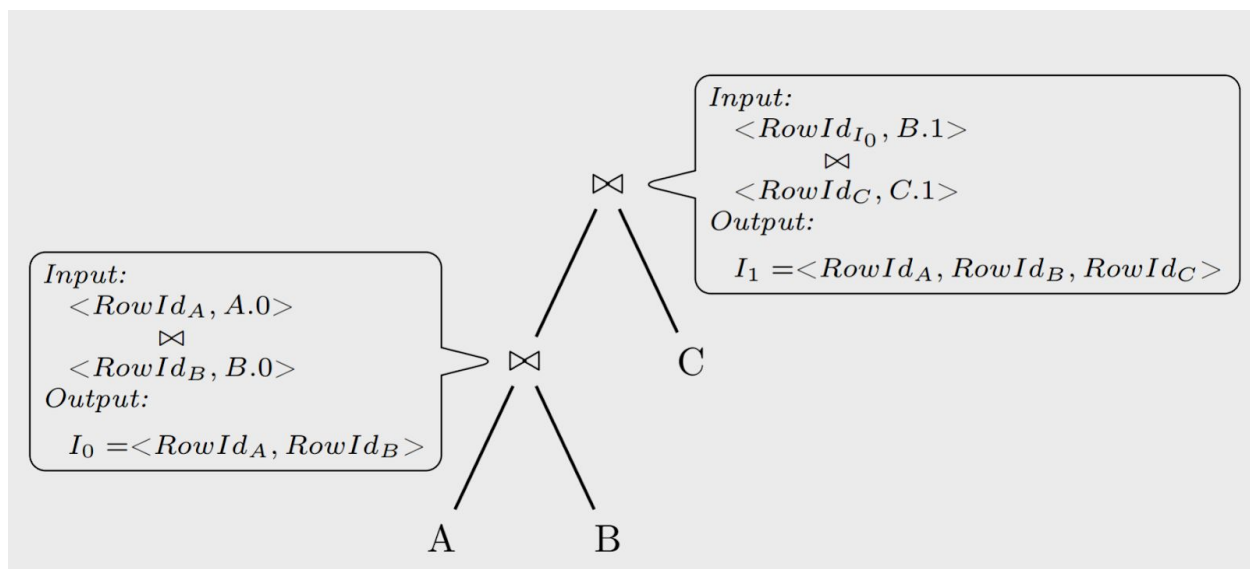
Εφόσον έχουμε αποθηκεύσει τις σχέσεις μας στην μνήμη, το επόμενο στάδιο είναι να εκτελέσουμε τα ερωτήματα. Σε κάθε βήμα θα πρέπει να επιλέξουμε και να εκτελέσουμε ένα κατηγορημα (predicate) και το αποτέλεσμα του κατηγορήματος θα πρέπει να ενημερώνει μια οντότητα/δομή η οποία θα κρατάει τα ενδιάμεσα αποτελέσματα που παράγονται σε κάθε βήμα. Πιο αναλυτικά έστω ότι θέλουμε να εκτελέσουμε το πιο κάτω ερώτημα:

0 2 4|0.1=1.2&1.0=2.1&0.1>3000|0.0 1.1

Έστω ότι στην αρχή επιλέγουμε να εκτελέσουμε το κατηγορημα $0.1 > 3000$. Σε αυτήν την περίπτωση θα πρέπει να σαρώσουμε τη στήλη 1 από την σχέση 0 και να επιστρέψουμε σαν αποτέλεσμα τα RowIds των γραμμών που ικανοποιούν το κατηγορημα. Το αποτέλεσμα αυτό θα αποθηκευτεί στην οντότητα που κρατάει τα ενδιάμεσα αποτελέσματα. Η οντότητα αυτή μετά την εκτέλεση του φίλτρου θα κρατάει ένα πίνακα με μια στήλη με τα RowId0 που ικανοποιούν το φίλτρο. Έστω ότι το επόμενο κατηγορημα που θα εκτελέσουμε είναι το $0.1 = 1.2$, το οποίο υπονοεί μια ζεύξη ισότητας μεταξύ των σχέσεων 0 και 2. Σε αυτήν την περίπτωση παρατηρούμε

ότι στα ενδιάμεσα αποτελέσματά μας υπάρχει είδη η σχέση 0. Για τον λόγο αυτό η αναπαράσταση RowId, value που θα λάβει μέρος στο join για την σχέση 0 θα πρέπει να αναφέρεται σε row ids του πίνακα που είναι αποθηκευμένος στην οντότητα αναπαράστασης των ενδιάμεσων αποτελεσμάτων. Αντίθετα, η αναπαράσταση RowId, value που θα λάβει μέρος στο join για την σχέση 2 θα κατασκευαστεί κατευθείαν από τη στήλη 2.2. Το αποτέλεσμα του Join που θα προκύψει θα έχει rowids που αντιστοιχούν στον πίνακα των ενδιάμεσων αποτελεσμάτων και rowids που θα αντιστοιχούν στην σχέση 2. Με αυτά τα αποτελέσματα θα ενημερώσουμε των πίνακα των ενδιάμεσων αποτελεσμάτων έτσι ώστε να κρατάνε πια ένα πίνακα με 2 στήλες με RowId0|RowId2. Τέλος θα πρέπει να εκτελέσουμε τη ζεύξη 1.0=2.1. Πάλι βλέπουμε ότι στα ενδιάμεσα αποτελέσματα υπάρχει η σχέση 2 ενώ δεν υπάρχει η σχέση 4. Οπότε κάνοντας την ίδια διαδικασία με πριν, η ενημερωμένη οντότητα που προκύπτει θα έχει την μορφή RowId0|RowId2|RowId4 και θα αναφέρεται σε όλες τις γραμμές που ικανοποιούν όλα τα κατηγορήματα.

Στην περίπτωση που ένα κατηγορήμα κάνει ζεύξη ισότητας μεταξύ 2 στηλών του ίδιου πίνακα τότε **δεν** θα πρέπει να χρησιμοποιήσουμε τον τελεστή ζεύξης αλλά να σαρώσουμε την σχέση και να επιστρέψουμε τα rowids των γραμμών που οι τιμές στα δύο αυτά πεδία είναι ίδιες. Το ίδιο ισχύει αν η ζεύξη αναφέρετε σε στήλες δυο διαφορετικών σχέσεων οι οποίες όμως υπάρχουν και οι δύο στα ενδιάμεσα αποτελέσματα. Τέλος αν η ζεύξη που έχουμε επιλέξει δεν χρησιμοποιεί κανένα από τους πίνακες που βρίσκονται στην οντότητα των ενδιάμεσων αποτελεσμάτων, θα πρέπει να εισάγουμε μια νέα τέτοια οντότητα. Στην περίπτωση που μία επόμενη ζεύξη χρησιμοποιεί σχέσης που βρίσκονται και στην μία και στην άλλη οντότητα, τότε η αναπαράσταση RowId, value που θα λάβει μέρος στο join θα αναφέρεται σε rowids της μίας και την άλλης οντότητας αντίστοιχα.



Ακόμη, κατά τον υπολογισμό ενός ενδιάμεσου αποτελέσματος που προκύπτει από κάποιο κατηγορήμα ζεύξης, τα δεδομένα εξαιτίας της μεθόδου που ακολουθούμε κατά τη ζεύξη θα είναι ταξινομημένα ως προς το κλειδί της αριστερής σχέσης. Επομένως, αν κάποιο επόμενο

κατηγορήμα κάνει join στην ίδια σχέση και στήλη με κάποιο προηγούμενο ενδιάμεσο αποτέλεσμα που είναι ταξινομημένο ήδη, τότε αρκεί να αξιοποιήσουμε την υπάρχουσα ταξινόμηση του ενδιάμεσου αποτελέσματος κατά την εκτέλεση του merge join.

Για παράδειγμα στο παρακάτω ερώτημα:

8 0 11|1.0=0.2&1.0=2.2&0.3=9477|0.2

Παρατηρούμε ότι τα αποτελέσματα του 1ου κατηγορήματος θα είναι ταξινομημένα ως προς την 1η στήλη της σχέσης 0. Έπειτα, το 2ο κατηγορήμα επεκτείνει το προηγούμενο αποτέλεσμα κάνοντας join τη 3η στήλη της σχέσης 11. Αφότου ταξινομήσουμε τα δεδομένα της σχέσης 11 για τη ζεύξη, μετά αρκεί να κάνουμε το merge join το ενδιάμεσο αποτέλεσμα του πρώτου κατηγορήματος με τα δεδομένα της 2ης στήλης της σχέσης 11.

Εκτέλεση Αθροισμάτων

Για την εκτέλεση των αθροισμάτων ελέγχου, θα πρέπει να σαρώσουμε τις στήλες που θέλουμε παίρνοντας υπ όψιν μόνο τις γραμμές των οποίων τα rowids είναι αποθηκευμένα στον πίνακα των τελικών ενδιάμεσων αποτελεσμάτων.

Γενικά

Ο κώδικάς σας αναμένεται να δουλεύει με όλους τους διαφορετικούς συνδυασμούς ερωτημάτων και κατηγορημάτων, ακόμη και αν δεν αναφέρονται (οι συνδυασμοί) στην περιγραφή της εργασίας.

Προθεσμία παράδοσης: 15/12/2019 23:59

Γλώσσα υλοποίησης: C / C++ χωρίς χρήση stl.

Περιβάλλον υλοποίησης: Linux (gcc 5.4+).

Παραδοτέα: Η παράδοση της εργασίας θα γίνει με βάση το τελευταίο commit πριν την προθεσμία υποβολής στο git repository σας. **Η χρήση git είναι υποχρεωτική.**