

## Column Store ΑΡΧΕΙΟ ΣΩΡΟΥ

ΥΛΟΠΟΙΗΘΗΚΕ ΑΠΟ ΤΟΥΣ:

ΓΡΗΓΟΡΙΟΣ ΔΗΜΟΠΟΥΛΟΣ	1115201100198
ΔΗΜΗΤΡΗΣ ΑΝΑΣΤΑΣΟΠΟΥΛΟΣ	1115201100030

Στο πρώτο block του κάθε αρχείου κρατάμε πληροφορίες σχετικά με το αν το αρχείο είναι τύπου column store heap (αν οι 4 πρώτες θέσεις byte=0) στις επόμενες 4 θέσεις κρατάμε πληροφορία με την μορφή ακέραιου αριθμού από το 1 έως το 9 για ποιο αρχείο column μιλάμε (π.χ. για 1 μιλάμε για id για 2 για name κ.ο.κ.). Στις επόμενες 4 θέσεις φυλάμε πληροφορία για το μέγιστο πλήθος εγγραφών που χωράνε σε ένα block αρχείου τύπου column που ορίζεται πιο πίσω. Στις επόμενες 4 θέσεις βάζουμε το μέγεθος του πεδίου που αποθηκεύεται σε αυτό το αρχείο. Και στις επόμενες θέσεις του block αποθηκεύεται το row id της τελευταίας εγγραφής του αρχείου. Παραπάνω περιγράφηκε η δομή του πρώτου block κάθε αρχείου τύπου CSFile\_\*columnhere\*. Όσο αναφορά για τα block τα οποία αποθηκεύονται εγγραφές σε αυτά, στις 4 πρώτες θέσεις byte αποθηκεύεται ένας μετρητής που μας δείχνει κάθε φορά πόσες εγγραφές έχει μέσα το block. Στην συνέχεια οι εισαγωγές των στοιχείων γίνονται αποθηκεύοντας το row id της εγγραφής ακολουθούμενο από το περιεχόμενο του πεδίου που προσδιορίζεται από το αντίστοιχο αρχείο στο οποίο θα γίνει η εισαγωγή. Η ίδια λογική ισχύει για όλα τα blocks με εγγραφές των αρχείων τύπου CSFile\_\*columnhere\*. Τώρα, όσο αναφορά το αναγνωριστικό αρχείο CSFile\_info, σε αυτό αποθηκεύονται εγγραφές της μορφής Header\_Info (struct με 2 μέλη: όνομα αρχείου και filedesc αρχείου). Συνολικά στο αρχείο αυτό θα μπουν 9 τέτοιες εγγραφές.

Οι συναρτήσεις και ο φόρτος εργασίας διαμοιράστηκε όπως περιγράφεται πιο κάτω:

**void print\_record(Record record).....ΔΗΜΗΤΡΗΣ ΑΝΑΣΤΑΣΟΠΟΥΛΟΣ**

Η συνάρτηση αυτή εκτυπώνει ένα record.

**int access(int fileDesc, Record \*record, int c\_id, int use).....ΔΗΜΗΤΡΗΣ ΑΝΑΣΤΑΣΟΠΟΥΛΟΣ**

Η συνάρτηση αυτή χρησιμοποιείται από την insert για εισαγωγή περιεχομένων column στο κατάλληλο αρχείο column μιας εγγραφής. Και επίσης χρησιμοποιείται και από την get all entries για να “χτίσει” ένα record με συγκεκριμένο column id από τα επιμέρους αρχεία.

**void CS\_GetAllEntries(int fileDesc, char\* fieldName, void\* value, char\*\* fieldNames, int n).....ΔΗΜΗΤΡΗΣ ΑΝΑΣΤΑΣΟΠΟΥΛΟΣ**

Περιγράφεται στην εκφώνηση.

**int CS\_InsertEntry (int fileDesc, Record record).....ΓΡΗΓΟΡΙΟΣ ΔΗΜΟΠΟΥΛΟΣ**

Περιγράφεται στην εκφώνηση.

**int CS\_CloseFile(int fileDesc).....ΓΡΗΓΟΡΙΟΣ ΔΗΜΟΠΟΥΛΟΣ**

Περιγράφεται στην εκφώνηση.

**int CS\_OpenFile(char\* header\_info).....ΓΡΗΓΟΡΙΟΣ ΔΗΜΟΠΟΥΛΟΣ**

Περιγράφεται στην εκφώνηση.

**int CS\_CreateFiles(char\*\* fileNames).....ΔΗΜΗΤΡΗΣ ΑΝΑΣΤΑΣΟΠΟΥΛΟΣ**

Περιγράφεται στην εκφώνηση.

**Read\_me.....ΓΡΗΓΟΡΙΟΣ ΔΗΜΟΠΟΥΛΟΣ**

Βέβαια, υπήρξε υποστήριξη και από τις δυο πλευρές πάνω σε θέματα που κάποιος χρειάστηκε βοήθεια. Το πρόγραμμα γράφτηκε και δοκιμάστηκε σε περιβάλλον Ubuntu και στους υπολογιστές του εργαστηρίου Sun της σχολής μέσω Putty.

## ΣΥΓΚΡΙΣΗ ΥΛΟΠΟΙΗΣΕΩΝ (HP, HT, CS)

- **HP:** Για εκτύπωση όλων των εγγραφών χρειάστηκαν 44 block. Για αναζήτηση κάποιων συγκεκριμένων εγγραφών, πάλι 44. Αρκετά αργός τρόπος αναζήτησης, με γρήγορη εκτύπωση όλων των εγγραφών.
- **HT:** Για εκτύπωση όλων των εγγραφών χρειάστηκαν 61 block, ευρετηριάζοντας με βάση το πεδίο "surname". Για αναζήτηση κάποιων συγκεκριμένων εγγραφών, 8 block ("surname" = "Paradopoulos"). Αρκετά γρήγορος τρόπος αναζήτησης με βασικό πλεονέκτημα την ταχύτατη εύρεση κάποιας εγγραφής με συγκεκριμένα χαρακτηριστικά. Βασικό μειονέκτημα η πιο αργή εκτύπωση όλων των εγγραφών από την HP.
- **CS:** Για εκτύπωση όλων των εγγραφών χρειάστηκε η προσπέλαση 5402 block. Για αναζήτηση κάποιων συγκεκριμένων εγγραφών ("dateOfBirth" = "15-04-1984"), 173 block, μιλώντας και για τα 9 πεδία. Αρκετά αργός τρόπος εκτύπωσης όλων των εγγραφών και αρκετά γρήγορος τρόπος αναζήτησης. Βασικό πλεονέκτημα ότι μπορούμε να επιλέξουμε ποια πεδία της εγγραφής θέλουμε να ψάξουμε και ποια θέλουμε να τυπώσουμε και έτσι η αναζήτηση αυξάνει ταχύτητα στην περίπτωση που μας ενδιαφέρει π.χ. μόνο το id, για id = 17 τα blocks που διαβάστηκαν για την εκτύπωση του id γίνονται 8. Γενικότερα, η εκτύπωση ενός πεδίου με συγκεκριμένο row id κοστίζει 2 block, ενώ η αναζήτηση κάποιας τιμής ενός πεδίου κοστίζει x block, όπου x το πλήθος των block του αντίστοιχου αρχείου.

Ακολουθούν και μερικά ενδεικτικά διαγράμματα:

