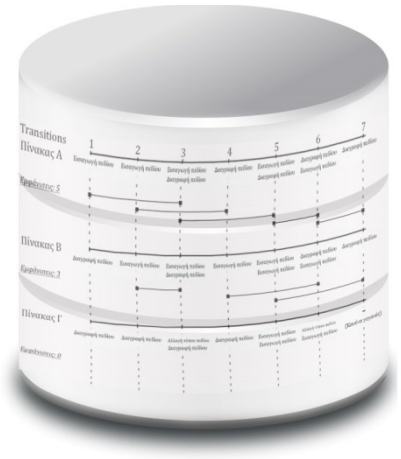
ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΜΗΧ. Η/Υ & ΠΛΗΡΟΦΟΡΙΚΗΣ
ΠΑΝΕΠΙΣΤΗΜΙΟΥΠΟΛΗ ΙΩΑΝΝΙΝΩΝ, ΙΩΑΝΝΙΝΑ

Δ.Ε.-2015-01 21 Σεπτεμβρίου 2015

ΕΞΑΓΩΓΗ ΠΡΟΤΥΠΩΝ ΕΞΕΛΙΞΗΣ ΤΟΥ ΣΧΗΜΑΤΟΣ ΒΑΣΕΩΝ ΔΕΔΟΜΕΝΩΝ

ΠΑΠΠΑΣ ΑΘΑΝΑΣΙΟΣ

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ: Π. Βασιλειάδης



ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

Πρόλογος

Μια βάση δεδομένων, από την στιγμή που θα δημιουργηθεί, αλλάζει εσωτερική δομή με το πέρασμα του χρόνου: νέοι πίνακες δημιουργούνται, παλαιοί καταστρέφονται, πεδία διαγράφονται, μετονομάζονται κλπ. Η διαδικασία αυτή ονομάζεται «εξέλιξη του σχήματος της βάσης δεδομένων» (schema evolution). Μπορούν όμως να ανευρεθούν επαναλαμβανόμενες ακολουθίες γεγονότων από την εξέλιξη των σχημάτων της βάσης δεδομένων; Η έρευνα στην περιοχή αυτή είναι θεμελιώδους φύσεως και αφορά στο να κατανοήσουμε την ύπαρξη προτύπων (ή ακόμα καλύτερα νόμων) για το πώς εξελίσσονται οι βάσεις δεδομένων με την πάροδο του χρόνου. Η συγκεκριμένη εργασία έχει ως αντικείμενο την υλοποίηση αλγορίθμων εξόρυξης προτύπων τύπου συχνών στοιχειοσυνόλων για την εύρεση των προτύπων συμπεριφοράς στον παλμό των γεγονότων των σχημάτων και την οπτικοποίηση τους με τρόπο ο οποίος είναι διαδραστικός.

Ολοκληρώνοντας την διπλωματική μου εργασία θα ήθελα να ευχαριστήσω τον επιβλέπον καθηγητή μου κ. Π. Βασιλειάδη, ο οποίος με αρκετή υπομονή με καθοδήγησε και με υποστήριξε σε όλη την διάρκεια εκπόνησης της εργασίας. Επίσης, θα ήθελα να ευχαριστήσω την οικογένεια μου για την στήριξή της.

Σεπτέμβριος 2015

Παππάς Αθανάσιος

Περίληψη στα ελληνικά

Η μελέτη των επαναλαμβανόμενων προτύπων από τον παλμό των γεγονότων της εξέλιξης των σχημάτων βάσεων δεδομένων είναι καίριας σημασίας καθώς μπορεί να διευκολύνει και να μειώσει το κόστος της συντήρησης της βάσης δεδομένων. Στην συγκεκριμένη εργασία κατασκευάστηκε ένα εργαλείο το οποίο δέχεται ως είσοδο τις αλλαγές που έχει υποστεί το σχήμα μιας βάσης δεδομένων στο χρόνο και παράγει ως έξοδο συχνές ακολουθίες από τις συγκεκριμένες αλλαγές. Για τον σκοπό αυτό, εφαρμόστηκαν αλγόριθμοι εξόρυξης συχνών ακολουθιών οι οποίοι υποστηρίζουν δύο διαφορετικούς τρόπους μέτρησης της υποστήριξης. Επιπλέον, υλοποιήθηκε ένα γραφικό περιβάλλον στο οποίο ο χρήστης μπορεί να δει με ένα διαδραστικό τρόπο όλες τις συχνές ακολουθίες που προκύπτουν. Τέλος, μέσω των πειραμάτων που πραγματοποιήθηκαν παρατηρήθηκε ότι υπάρχουν χρήσιμες ακολουθίες από αλλαγές που έχει υποστεί το σχήμα μιας βάσης και ότι η μελέτη τους μπορεί να βοηθήσει στην κατανόηση της εξέλιξης του σχήματος της βάσης δεδομένων.

Λέξεις Κλειδιά: εξέλιξη βάσης δεδομένων, εξόρυξη προτύπων

Περίληψη στα αγγλικά

Studying the patterns derived from the evolution of database schemata is of great importance as it can reduce the maintenance cost of the database. In this work, a tool was built which receives as input all the changes that the schema of a database has undergone and produces as output frequent patterns from these changes. In order to achieve this goal, sequence mining algorithms that support two different support counting methods were used. In addition, a graphical user interface providing an interactive way to show the frequent patterns to the user was built. Finally, the experiments show that there are useful patterns of changes that are derived from the evolution of database schemata and studying them will be helpful to understand the schema evolution of a database.

Keywords: schema evolution, sequence mining

Πίνακας περιεχομένων

Κεφάλαιο 1. Εισαγωγή

Μια βάση δεδομένων είναι μια οργανωμένη συλλογή δεδομένων τα οποία είναι συνήθως οργανωμένα έτσι ώστε να μοντελοποιήσουν διάφορες πτυχές της πραγματικότητας και να διευκολύνουν την επερώτηση τους. Σήμερα, το μέγεθος των δεδομένων το οποίο αποθηκεύεται στις βάσεις δεδομένων είναι τεράστιο και έτσι η έρευνα πάνω στην εξέλιξη των σχημάτων των βάσεων δεδομένων είναι ουσιώδης για την ανακάλυψη γνώσης που κρύβεται σε αυτήν. Από την αλλαγή της δομής της βάσης στο χρόνο μπορούν να προκύψουν χρήσιμα συμπεράσματα τα οποία μπορούν να χρησιμοποιηθούν για την διευκόλυνση της συντήρησής της.

1.1 Αντικείμενο της διπλωματικής

Όπως τα περισσότερα συστήματα σήμερα, έτσι και οι βάσεις δεδομένων έχουν ανάγκη να εξελίσσονται έτσι ώστε να μπορούν να ανταποκριθούν στις ανάγκες που προκύπτουν. Η εξέλιξη του σχήματος μιας βάσης δεδομένων μπορεί να προκαλέσει αρκετά προβλήματα στις εφαρμογές που την χρησιμοποιούν. Για παράδειγμα η αφαίρεση ενός πεδίου που στηρίζει τη συσχέτιση μεταξύ δύο πινάκων της βάσης μπορεί να οδηγήσει σε κατάρρευση μιας εφαρμογής που εκτελεί ένα ερώτημα που χρησιμοποιεί την συγκεκριμένη συσχέτιση. Οι αλλαγές στο σχήμα μιας βάσης επηρεάζουν επίσης και τα δεδομένα τα οποία είναι αποθηκευμένα στη βάση. Για παράδειγμα, με την εισαγωγή ενός πεδίου σε έναν πίνακα πρέπει να οριστούν οι κατάλληλες τιμές του συγκεκριμένου πεδίου για όλα τα δεδομένα που είναι ήδη αποθηκευμένα στην βάση. Το πρόβλημα της εξέλιξης του σχήματος βάσεων δεδομένων απασχολεί για αρκετά χρόνια την ερευνητική κοινότητα, όμως λόγω της έλλειψης δημόσια διαθέσιμων ιστοριών από την εξέλιξη βάσεων δεδομένων, και τα αποτελέσματα και τα εργαλεία που έχουν υλοποιηθεί για την λύση του προβλήματος είναι περιορισμένα.

Αντικείμενο της συγκεκριμένης διπλωματικής εργασίας είναι η εξαγωγή προτύπων εξέλιξης του σχήματος βάσεων δεδομένων. Υπάρχουν αρκετές συλλογές από εκδόσεις του σχήματος της ίδιας βάσης όπου σε κάθε μετάβαση από μία εκδοχή της βάσης σε μια άλλη μπορούμε να εξάγουμε ένα διάνυσμα από αλλαγές. Οι αλλαγές αυτές είναι πολύπλοκες και υπάρχει η ανάγκη εύρεσης ενός συστηματικού τρόπου για την κατανόηση του τρόπου εμφάνισης των αλλαγών στον χρόνο καθώς και την συσχέτιση μεταξύ τους.

Αν θεωρηθεί ότι όλες οι αλλαγές που υπόκειται κάθε πίνακας μιας βάσης αποτελούν μία ταξινομημένη λίστα από γεγονότα που σχετίζονται με τον συγκεκριμένο πίνακα τότε αυτή η λίστα αποτελεί μία ακολουθία από γεγονότα. Η εξέλιξη του σχήματος μιας βάσης δεδομένων αποτελείται από αρκετές τέτοιες ακολουθίες γεγονότων, μία για κάθε πίνακα. Εφαρμόζοντας αλγορίθμους ανακάλυψης συχνών ακολουθιών στην ιστορία των πινάκων μιας βάσης δεδομένων μπορούν να εξαχθούν όλες εκείνες οι ακολουθίες οι οποίες έχουν υποστήριξη μεγαλύτερη ή ίση από ένα ελάχιστο κατώφλι υποστήριξης.

Μία συχνή ακολουθία είναι μία λίστα από γεγονότα που αναφέρονται σε έναν πίνακα, ταξινομημένη κατά αύξουσα σειρά με βάση τον χρόνο, η οποία ξεπερνάει ένα ελάχιστο κατώφλι υποστήριξης. Για παράδειγμα, αν μία ακολουθία εμφανίζεται σε ένα μεγάλο ποσοστό της ιστορίας των πινάκων μιας βάσης δεδομένων τότε θεωρείται συχνή.

Η αναλυτική περιγραφή της υποστήριξης καθώς επίσης και η λύση στο πρόβλημα εξόρυξης συχνών ακολουθιών περιγράφονται στο κεφάλαιο 3.

Η συγκεκριμένη εργασία έχει ως στόχο την υλοποίηση ενός εργαλείου το οποίο θα επεξεργάζεται τα γεγονότα των αλλαγών που έχει υποστεί μια βάση δεδομένων στον χρόνο και θα εξάγει επαναλαμβανόμενα πρότυπα από τις συγκεκριμένες αλλαγές χρησιμοποιώντας βασικούς αλγόριθμους εξόρυξης προτύπων.

1.2 Οργάνωση του τόμου

Η συγκεκριμένη διπλωματική εργασία αποτελείται από 7 κεφάλαια τα οποία αναλύονται στις παρακάτω παραγράφους.

Στο κεφάλαιο 2 περιγράφονται οι σχετικές εργασίες πάνω στην εξέλιξη του σχήματος βάσεων δεδομένων καθώς επίσης και στην ανακάλυψη ακολουθιακών

υποδειγμάτων. Επιπλέον, δίνεται μία πιο σαφής περιγραφή του στόχου της συγκεκριμένης εργασίας περιγράφοντας πιο αναλυτικά το θέμα.

Στο πρώτο μισό του κεφαλαίου 3 δίνονται κάποιοι χρήσιμοι ορισμοί και στη συνέχεια δίνεται ο τυπικός ορισμός του προβλήματος. Επιπλέον, περιγράφεται ο αναλυτικός τρόπος επίλυσης του προβλήματος μέσω του αλγόριθμου του σχήματος 3 καθώς επίσης και διάφορες προσθήκες και παραλλαγές που μπορούν να εφαρμοστούν. Στο δεύτερο κομμάτι του τρίτου κεφαλαίου παρουσιάζεται ο σχεδιασμός και η αρχιτεκτονική του λογισμικού με την χρήση UML διαγραμμάτων.

Στο κεφάλαιο 4 περιγράφονται οι λεπτομέρειες υλοποίησης του λογισμικού. Παρουσιάζονται τα βασικότερα κομμάτια κώδικα του εργαλείου και επίσης οι κλάσεις με την μεγαλύτερη χρησιμότητα. Επίσης, περιγράφεται η μορφή των αρχείων εισόδου και εξόδου του εργαλείου. Τέλος, παρουσιάζεται η μεθοδολογία ελέγχου του λογισμικού και δίνεται ένα παράδειγμα αρχείου εισόδου καθώς επίσης και το αναμενόμενο αποτέλεσμα του αλγορίθμου.

Στο κεφάλαιο 5 παρουσιάζονται όλα τα πειράματα που πραγματοποιήθηκαν με τα αποτελέσματά τους. Πιο συγκεκριμένα, περιγράφεται η πειραματική διαδικασία εξαγωγής των προτύπων εξέλιξης της βάσης δεδομένων και αναλύονται διεξοδικά τα αποτελέσματα.

Τέλος, στο κεφάλαιο 6 παραθέτονται τα συμπεράσματα των πειραμάτων και περιγράφονται οι μελλοντικές επεκτάσεις του λογισμικού.

Κεφάλαιο 2. Περιγραφή Θέματος

2.1 Σχετικές εργασίες

Ένας από τους πρώτους που ασχολήθηκε με την εξέλιξη σχημάτων βάσεων δεδομένων ήταν ο D. Sjoberg ο οποίος στο άρθρο του [Sjob91] κατασκεύασε ένα εργαλείο με το όνομα "thesaurus". Το συγκεκριμένο εργαλείο υλοποιήθηκε πάνω από ένα υπάρχον σύστημα διαχείρισης ιατρικών δεδομένων (health management system) και παρατηρούσε την εξέλιξή του για ένα διάστημα 18 μηνών και τα αποτελέσματα της έρευνας αναφέρονται στο συγκεκριμένο άρθρο. Πιο αναλυτικά, κατά την διάρκεια της μελέτης διαπιστώθηκε ότι ο αριθμός των σχέσεων αυξήθηκε κατά 139% όπως και ο αριθμός των πεδίων κατά 274%. Επίσης, σημειώθηκαν 35% περισσότερες προσθήκες σχέσεων και πεδίων από ότι διαγραφές. Από την μεριά των επιπτώσεων της εξέλιξης του σχήματος ο συγγραφέας αναφέρει ότι υπάρχει ανάγκη χρήσης εργαλείων διαχείρισης και μέτρησης των αλλαγών.

Αρκετά χρόνια αργότερα, εκδόθηκε μια έρευνα [CMTZ08] πάνω στη MediaWiki, ένα λογισμικό ανοικτού κώδικα (open-source) που δημιουργήθηκε για να υποστηρίζει την γνωστή σε όλους Wikipedia. Οι συγγραφείς του συγκεκριμένου άρθρου συγκεντρώνοντας 171 διαφορετικές versions, που αντιστοιχούν σε διάρκεια 4 χρόνων και 7 μηνών, παραθέτουν τα αποτελέσματα της έρευνας τα οποία εξήχθησαν από ένα σύνολο εργαλείων που δημιουργήθηκαν για αυτό τον σκοπό. Συγκεκριμένα, παρατηρήθηκε αύξηση στον αριθμό των πινάκων κατά 100% και στον αριθμό των πεδίων κατά 142%. Όσον αφορά το χρόνο ζωής των πινάκων και των πεδίων της βάσης, διαπιστώθηκε ότι κατά μέσο όρο ο χρόνος ζωής κάθε πίνακα και κάθε πεδίου είναι 60,4% και 56,8% του συνολικού χρόνου ζωής της βάσης αντίστοιχα. Τέλος, οι συγγραφείς του άρθρου καταλήγουν στο συμπέρασμα ότι υπάρχουν σοβαρές ενδείξεις ότι η εξέλιξη του σχήματος βάσεων δεδομένων έχει μεγάλο αντίκτυπο στις εφαρμογές που χρησιμοποιούν την βάση

και στηρίζουν τον ισχυρισμό ότι υπάρχει ανάγκη καλύτερης υποστήριξης στην εξέλιξη του σχήματος.

Μία επιπλέον μελέτη, μεγαλύτερης κλίμακας αυτή την φορά με τίτλο "Open-Source Database: Within, Outside, or Beyond Lehman's Laws of Software Evolution?" δημοσιεύθηκε το 2014. Στην συγκεκριμένη μελέτη οι συγγραφείς ελέγχουν αν οι νόμοι του Lehman[LMR+97] για την εξέλιξη του λογισμικού ισχύουν στην εξέλιξη σχημάτων βάσεων δεδομένων. Πιο συγκεκριμένα, επεξεργάζονται τα σχήματα 8 βάσεων δεδομένων μέσω του εργαλείου Εκάτη (Hecate), που έχει δημιουργηθεί για τον συγκεκριμένο σκοπό. Οι συγγραφείς του άρθρου διαπίστωσαν ότι όσον αφορά το μέγεθος του σχήματος υπάρχουν κάποιες περίοδοι στις οποίες υπάρχει αύξηση, κυρίως στην αρχή ή μετά από μεγάλες πτώσεις του μεγέθους αλλά υπάρχουν και περίοδοι σταθερότητας. Επιπλέον παρατηρήθηκε ότι η ύπαρξη της συντήρησης της βάσης υπάρχει σε όλες τις συλλογές δεδομένων με την έννοια των αφαιρέσεων πεδίων και σχέσεων. Τέλος, οι συγγραφείς καταλήγουν στο συμπέρασμα ότι οι νόμοι του Lehman ισχύουν στις βάσεις ανοικτού κώδικα. Τα αποτελέσματα της μελέτης παραθέτονται στο [SkVZ14].

Παρ' όλα αυτά σε καμία περίπτωση δεν επιχειρήθηκε η ανεύρεση επαναλαμβανόμενων προτύπων τύπου συχνών στοιχειοσυνόλων πάνω στην εξέλιξη σχημάτων βάσεων δεδομένων, πράγμα το οποίο έχει ως στόχο η συγκεκριμένη διπλωματική εργασία. Ο πρώτος αλγόριθμος παραγωγής συχνών στοιχειοσυνόλων είναι ο αλγόριθμος Apriori [AgSr94] ο οποίος εφαρμόζεται σε πολλά εμπορικά προϊόντα αυτός καθαυτός ή με διάφορες παραλλαγές.

Ένα χρόνο αργότερα, αναπτύχθηκε μια σειρά αλγορίθμων για την ανακάλυψη ακολουθιακών υποδειγμάτων οι οποίοι βασίζονται στον αλγόριθμο Apriori. Οι συγγραφείς του άρθρου [AgSr95] παρουσιάζουν τρείς αλγορίθμους για την επίλυση του συγκεκριμένου προβλήματος και παραθέτουν τις επιδόσεις τους.

2.2 Συνεισφορά της εργασίας

2.2.1 Υπόβαθρο: Το εργαλείο «Εκάτη» και οι συλλογές δεδομένων

Αρχικά, αξίζει να αναφερθούμε στην εργασία του Ι. Σκουλή [Skou13] η οποία έχει άμεση σχέση με την παρούσα διπλωματική εργασία. Η «Εκάτη» είναι ένα ανοικτό (open-source) λογισμικό το οποίο δέχεται ως είσοδο αρχεία DLL τα οποία

αποτελούν όλες τις εκδόσεις της βάσης από την στιγμή της δημιουργίας της. Συγκρίνοντας δύο οποιεσδήποτε εκδόσεις μιας βάσης, η Εκάτη παράγει ως έξοδο όλες τις αλλαγές που υπέστη το σχήμα της βάσης δεδομένων ανάμεσα στις 2 εκδόσεις της βάσης καθώς και διάφορες επιπλέον μετρικές όπως είναι για παράδειγμα ο αριθμός των πεδίων που προστέθηκαν ή διαγράφτηκαν. Πιο συγκεκριμένα, οι αλλαγές οι οποίες μπορεί να αναγνωριστούν από την «Εκάτη» είναι:

- Εισαγωγές πεδίων
- Διαγραφές πεδίων
- Αλλαγές τύπων πεδίων
- Αλλαγές πρωτευόντων κλειδιών
- Αλλαγές στο πλήθος των πεδίων ενός πίνακα
- Δημιουργία πινάκων
- Διαγραφή πινάκων

Η λίστα με τα datasets που έχει συγκεντρωθεί και επεξεργαστεί από την «Εκάτη» αποτελείται από 8 Open-Source συστήματα λογισμικού τα οποία προέρχονται από ένα ευρύ φάσμα εφαρμογών όπως Συστήματα Διαχείρισης Περιεχομένου (CMS's), συστήματα διαχείρισης εικόνων, online καταστήματα, καθώς και επιστημονικές αποθήκες δεδομένων. Τόσο η «Εκάτη» όσο και τα datasets μπορούν να βρεθούν στο δημόσιο αποθετήριο: `https://github.com/DAINTINESS-Group`.

2.2.2 Στόχος

Η συγκεκριμένη εργασία έχει ως στόχο την υλοποίηση ενός εργαλείου το οποίο λαμβάνοντας ως είσοδο τις αλλαγές που έχει υποστεί μια βάση δεδομένων από την στιγμή που δημιουργήθηκε να μπορεί να εξάγει ως έξοδο επαναλαμβανόμενα πρότυπα από τις συγκεκριμένες αλλαγές. Για την εξαγωγή επαναλαμβανόμενων προτύπων θα χρησιμοποιηθούν βασικοί αλγόριθμοι εξόρυξης προτύπων. Αν για κάθε πίνακα της βάσης δεδομένων όλα τα γεγονότα που σχετίζονται με τον συγκεκριμένο πίνακα ταξινομηθούν σε αύξουσα σειρά με βάση τον χρόνο, τότε λαμβάνεται μια ακολουθία για τον πίνακα αυτόν. Αυτά τα γεγονότα μπορεί να είναι: εισαγωγές πεδίων, διαγραφές πεδίων, αλλαγές στον τύπο των πεδίων, αλλαγές στα πρωτεύοντα κλειδιά, δημιουργία πινάκων, διαγραφή πινάκων κ.α. Επομένως, κάθε πίνακας της βάσης δεδομένων περιέχει μια ακολουθία δεδομένων, όπου ο όρος ακολουθία δεδομένων αναφέρεται σε ταξινομημένη λίστα από

γεγονότα που σχετίζονται με τον συγκεκριμένο πίνακα. Μία ακολουθία χαρακτηρίζεται από το μήκος της και από το πλήθος των γεγονότων που λαμβάνουν χώρα.

Στόχος λοιπόν της συγκεκριμένης διπλωματικής είναι η υλοποίηση ενός εργαλείου το οποίο θα λαμβάνει ως είσοδο τις αλλαγές που έχει υποστεί το σχήμα μιας βάσης στη διάρκεια της ζωής της, θα επεξεργάζεται τα συγκεκριμένα δεδομένα και θα ανακαλύπτει ακολουθιακά πρότυπα. Ένα ακολουθιακό πρότυπο είναι μία ακολουθία από γεγονότα η οποία εμφανίζεται συχνά στις αλλαγές που έχουν υποστεί οι πίνακες της βάσης δεδομένων. Το εργαλείο θα παρέχει επίσης ένα γραφικό περιβάλλον στο οποίο ο χρήστης θα μπορεί να δει τα συγκεκριμένα ακολουθιακά πρότυπα και να ελέγξει την ισχύ τους.

Κεφάλαιο 3. Ανάλυση και Σχεδίαση

Στο συγκεκριμένο κεφάλαιο περιγράφονται οι αλγόριθμοι επίλυσης του προβλήματος της εξαγωγής προτύπων εξέλιξης του σχήματος βάσεων δεδομένων όπως αυτό προσδιορίστηκε στα προηγούμενα κεφάλαια και επίσης δίνονται κάποιοι βασικοί ορισμοί για την καλύτερη κατανόηση τους. Επιπλέον περιγράφεται η ανάλυση και ο σχεδιασμός του συστήματος με την χρήση UML διαγραμμάτων.

3.1 Ορισμός προβλήματος και αλγόριθμοι επίλυσης

3.1.1 Τυπικός ορισμός του προβλήματος.

Προτού ξεκινήσουμε καλό θα ήταν να δοθούν κάποιοι σημαντικοί ορισμοί [PaMV10]:

Γεγονός (Atomic Change Event): Οποιαδήποτε μεμονωμένη αλλαγή η οποία επηρεάζει το σχήμα μιας βάσης. Ένα γεγονός αναφέρεται σε μία αλλαγή που συμβαίνει σε έναν πίνακα της βάσης δεδομένων. Πιο συγκεκριμένα τα είδη των γεγονότων που μπορεί να έχουν συμβεί για έναν πίνακα είναι τα εξής:

- Εισαγωγή πεδίου σε υπάρχοντα πίνακα
- Εισαγωγή πεδίου σε πίνακα που δημιουργήθηκε την ίδια χρονική στιγμή με την εισαγωγή του πεδίου
- Διαγραφή πεδίου από πίνακα που παραμένει ζωντανός
- Διαγραφή πεδίου από πίνακα που πεθαίνει την ίδια χρονική στιγμή που διαγράφεται το πεδίο.
- Εισαγωγή νέου πίνακα
- Διαγραφή πίνακα
- Αλλαγή στο τύπο κάποιου πεδίου
- Αλλαγή στο πρωτεύον κλειδί ενός πίνακα

Εκδοχή (Version): Ένα στιγμιότυπο του σχήματος της βάσης δεδομένων κάποια χρονική στιγμή. Μία εκδοχή μπορεί να αναφέρεται σε μία συγκεκριμένη ημερομηνία αλλά χάριν απλότητας αναθέτουμε σε κάθε εκδοχή έναν θετικό ακέραιο που αυξάνει κατά ένα από μία εκδοχή στην επόμενη χρονικά.

Μετάβαση (Transition): Η αλλαγή του σχήματος μιας βάσης δεδομένων από μία εκδοχή v_i σε μία εκδοχή v_j, με i < j. Κάθε μετάβαση αποτελείται από μια λίστα γεγονότων τα οποία συμβαίνουν σε έναν ή περισσότερους πίνακες της βάσης δεδομένων κατά την διάρκεια μιας μετάβασης.

Ακολουθία (Sequence): Μία λίστα από γεγονότα που αναφέρονται σε έναν πίνακα, ταξινομημένη κατά αύξουσα σειρά με βάση τον χρόνο. Μία σημαντική ιδιότητα στις ακολουθίες είναι το ότι αν ένα γεγονός που συνέβη την χρονική στιγμή t_1 και προηγείται ενός γεγονότος e τότε το γεγονός e συνέβη την χρονική στιγμή $t_2 > t_1$. Επίσης, μία ακολουθία μπορεί να χαρακτηριστεί από το μήκος της και από το πλήθος των γεγονότων που συμβαίνουν. Για παράδειγμα η ακολουθία { Εισαγωγή_πεδίου, Εισαγωγή_πεδίου}{Διαγραφή_πεδίου} έχει μήκος 3.

Υποακολουθία(Subsequence): Μια ακολουθία $t = \{t_1 t_2 \ldots t_m\}$ είναι υποακολουθία μιας ακολουθίας $s = \{s_1 s_2 \ldots s_n\}$, αν υπάρχουν ακέραιοι $1 \leq j_1 \leq j_2 \leq \ldots j_m \leq n$ τέτοιοι ώστε $t_1 \subseteq s_{j1}$, $t_2 \subseteq s_{j2}$ ... $t_m \subseteq s_{jm}$.

Στοιχείο Πίνακα(Table History Element): Περιέχει μία λίστα από γεγονότα τα οποία συμβαίνουν σε έναν πίνακα σε μία συγκεκριμένη μετάβαση. Ένα στοιχείο πίνακα μπορεί να προσδιοριστεί μοναδικά από τον πίνακα στον οποίον βρίσκεται και από την μετάβαση στην οποία αναφέρεται.

Ιστορία ενός πίνακα (Table History Sequence): Μία λίστα από στοιχεία(Table History Elements) που αναφέρονται σε έναν πίνακα και είναι της μορφής {($Event_1$, $Transition_1$), ($Event_2$, $Transition_2$) ... ($Event_N$, $Transition_N$)}.

Ιστορία βάσης δεδομένων (Database History): Ένα σύνολο από ιστορίες πινάκων.

3.1.2 Συχνές ακολουθίες.

Τι είναι όμως μία συχνή ακολουθία; Μία ακολουθία μπορεί να χαρακτηριστεί ως συχνή αν εμφανίζεται αρκετές φορές και πληροί κάποιους περιορισμούς. Αυτοί οι περιορισμοί μπορεί να είναι είτε χρονικοί περιορισμοί είτε περιορισμοί που προσδιορίζουν το ελάχιστο πλήθος εμφανίσεων της ακολουθίας στα δεδομένα εισόδου έτσι ώστε να χαρακτηριστεί ως συχνή. Στις περισσότερες περιπτώσεις οι δύο κατηγορίες περιορισμών συνδυάζονται με στόχο την εξόρυξη συχνών

ακολουθιών οι οποίες ικανοποιούν τους χρονικούς περιορισμούς και οι εμφανίσεις τους ξεπερνούν έναν ελάχιστο αριθμό εμφανίσεων.

Χρονικοί περιορισμοί: Όσον αφορά την πρώτη κατηγορία περιορισμών, τους χρονικούς περιορισμούς, αφού κάθε γεγονός χαρακτηρίζεται από ένα transition μπορούμε να περιορίσουμε τις ακολουθίες των γεγονότων να συμβαίνουν μέσα σε ένα χρονικό παράθυρο. Για παράδειγμα μπορούμε να ορίσουμε τον περιορισμό μέγιστης διάρκειας(ms – Maximum Span) ο οποίος περιορίζει τον μέγιστο χρόνο που επιτρέπεται ανάμεσα στην πρώτη και την τελευταία εμφάνιση γεγονότων μέσα σε ολόκληρη την ακολουθία. Άλλοι παράμετροι χρονικών περιορισμών είναι:

- ws : Event-set Window Size
- xg – Maximum Gap
- ng – Minimum Gap

Η πλήρης περιγραφή των παραμέτρων αναφέρεται στο [MaKK99]. Στην συγκεκριμένη εργασία δεν θα ασχοληθούμε με τους χρονικούς περιορισμούς.

Περιορισμοί συχνότητας: Η δεύτερη κατηγορία περιορισμών αναφέρεται στον τρόπο μέτρησης των εμφανίσεων μιας ακολουθίας στα δεδομένα εισόδου και στον προσδιορισμό του πλήθους των εμφανίσεων οι οποίες είναι αρκετές για τον χαρακτηρισμό μιας ακολουθίας ως συχνής. Αυτός ο προσδιορισμός καθορίζεται από την υποστήριξη (support) η οποία προσδιορίζει πόσο συχνά είναι εφαρμόσιμος ένας κανόνας σε ένα σύνολο δεδομένων. Επιπλέον η εμπιστοσύνη (confidence) καθορίζει πόσο συχνά οι ακολουθίες στο σύνολο Υ εμφανίζονται στην ιστορία πινάκων που περιέχουν τις ακολουθίες από το σύνολο Χ.

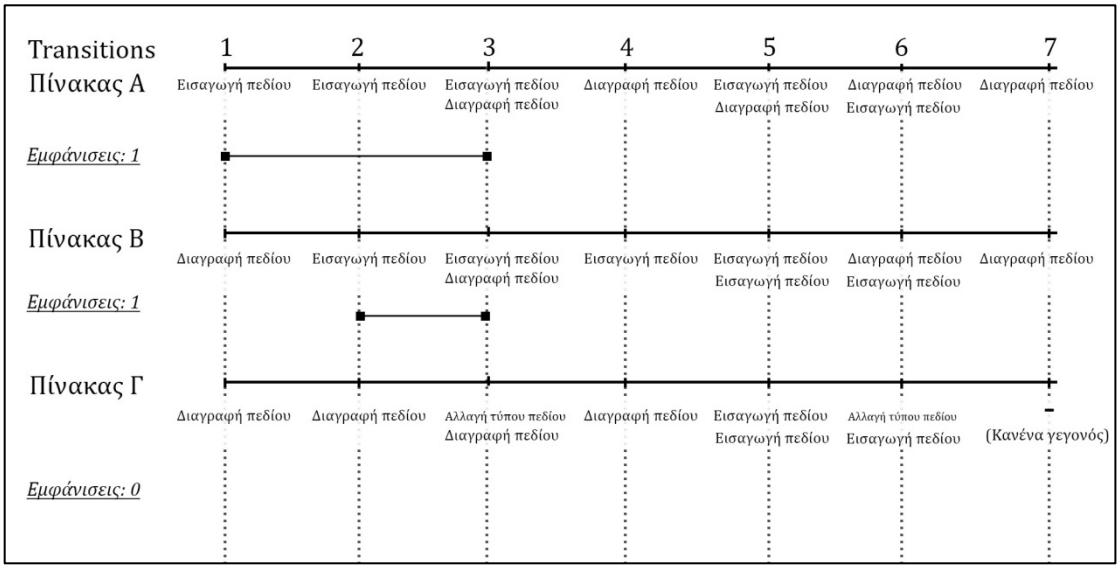
Υπάρχουν 5 διαφορετικοί τρόποι μέτρησης των εμφανίσεων μιας ακολουθίας στην ιστορία μιας βάσης δεδομένων, σύμφωνα με το [MaKK99], οι οποίοι χωρίζονται σε τρεις ομάδες. Η πρώτη ομάδα, στην οποία ανήκει η μέτρηση COBJ, μετράει μία εμφάνιση της υποψήφιας ακολουθίας στην ιστορία ενός πίνακα. Η δεύτερη ομάδα βασίζεται στην μέτρηση των παραθύρων στα οποία η υποψήφια ακολουθία εμφανίζεται (CWIN, CMINWIN). Τέλος, η τρίτη ομάδα βασίζεται στις διακριτές εμφανίσεις μιας υποψήφιας ακολουθίας στην ιστορία κάθε πίνακα (CDIST, CDIST_O). Στα πλαίσια της συγκεκριμένης διπλωματικής εργασίας χρησιμοποιούνται δύο διαφορετικοί τρόποι μέτρησης οι οποίοι αναλύονται παρακάτω:

- Μία εμφάνιση της ακολουθίας ανά ιστορία πίνακα – COBJ

  Ο συγκεκριμένος τρόπος μέτρησης βρίσκει το πολύ μία εμφάνιση της υποψήφιας ακολουθίας στην ιστορία κάθε πίνακα. Οπότε, η μέτρηση COBJ

αναφέρεται στον συνολικό αριθμό πινάκων στους οποίους εμφανίζεται η υποψήφια ακολουθία. Στο σχήμα 1 φαίνεται ένα παράδειγμα μέτρησης για την υποψήφια ακολουθία s={ Εισαγωγή πεδίου}{Διαγραφή πεδίου}[1].

Σχήμα 1. Μέτρηση COBJ για την ακολουθία { Εισαγωγή πεδίου}{Διαγραφή πεδίου}.



Όπως φαίνεται και στο σχήμα 1 στον πίνακα Α υπάρχει μια εισαγωγή πεδίου στην μετάβαση 1 και μία διαγραφή πεδίου στην μετάβαση 3. Υπάρχουν κι άλλες ακολουθίες της μορφής <Εισαγωγή πεδίου} {Διαγραφή πεδίου> αλλά στον συγκεκριμένο τρόπο μέτρησης μία οποιαδήποτε εμφάνιση αρκεί. Αξίζει να σημειωθεί ότι στις ακολουθίες η σειρά των γεγονότων είναι σημαντική και για αυτόν τον λόγο η ακολουθία <{Εισαγωγή πεδίου}{Διαγραφή πεδίου}> στον πίνακα Γ δεν έχει καμία εμφάνιση. Ο πίνακας Γ έχει υποστεί διαγραφή πεδίων και εισαγωγές πεδίων αλλά όχι με την σειρά της υποψήφιας ακολουθίας.

Για τον υπολογισμό της υποστήριξης, ο αριθμός των πινάκων στους οποίους υπάρχει η υποψήφια ακολουθία διαιρείται με τον συνολικό αριθμό των πινάκων. Οπότε σύμφωνα με τα παραπάνω ο τύπος που δίνει την υποστήριξη είναι:

$$\sigma_{COBJ}(r) = \frac{\#\text{πινάκων στους οποίους εμφανίζεται η ακολουθία } r}{\text{Συνολικός αριθμός πινάκων}}$$

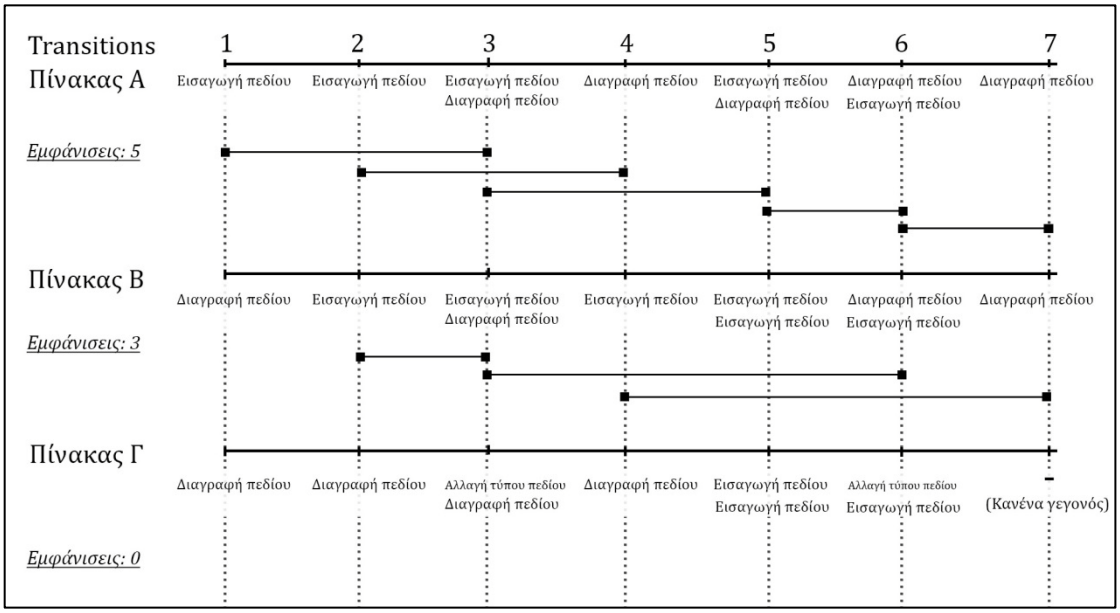
- Διακριτές εμφανίσεις χωρίς επικαλύψεις γεγονότος – μετάβασης – CDIST

[1] Η s = {X}{Y} περιγράφει μία ακολουθία της οποίας τα γεγονότα X και Y έχουν συμβεί σε διαφορετικές μεταβάσεις και η ακολουθία r = {X, Y} απεικονίζει μία ακολουθία της οποίας τα γεγονότα X και Y έχουν συμβεί στην ίδια μετάβαση.

Όπως αναφέρθηκε παραπάνω, ένα γεγονός προσδιορίζεται από ένα version ID, έναν πίνακα και ένα πεδίο στο οποίο αναφέρεται. Οπότε, όταν ένα ζεύγος γεγονότος - version ID για έναν πίνακα χρησιμοποιηθεί μία φορά για την μέτρηση, τότε δεν μπορεί να χρησιμοποιηθεί ξανά για την μέτρηση της συγκεκριμένης υποψήφιας ακολουθίας για τον συγκεκριμένο πίνακα. Ένα παράδειγμα της συγκεκριμένης μέτρησης για την υποψήφια ακολουθία s={Εισαγωγή πεδίου}{Διαγραφή πεδίου}.φαίνεται στο σχήμα 2.

Για τον υπολογισμό της υποστήριξης, ο αριθμός των ακολουθιών που εμφανίζονται σε ολόκληρη την ιστορία της βάσης (χωρίς επικαλύψεις) διαιρείται με τον μέγιστο πιθανό αριθμό εμφανίσεων της ακολουθίας σε όλη την ιστορία της βάσης.

Σχήμα 2. Μέτρηση CDIST για την ακολουθία { Εισαγωγή πεδίου}{Διαγραφή πεδίου}.



Στο σχήμα 2 φαίνεται ότι στον πίνακα Α η υποψήφια ακολουθία έχει 5 εμφανίσεις. Αυτό συμβαίνει διότι μόλις βρεθεί η πρώτη εμφάνιση (στις μεταβάσεις 1, 3), τότε τα ζεύγη <Εισαγωγή_πεδίου, Μετάβαση 1> και <Διαγραφή πεδίου, Μετάβαση 3> δεν μπορούν να ξαναχρησιμοποιηθούν για την μέτρηση. Για αυτό το λόγο η ακολουθία <Εισαγωγή πεδίου-Μετάβαση 2, Διαγραφή πεδίου-Μετάβαση 3> δεν λαμβάνεται υπόψιν. Αντίθετα, η ακολουθία <Εισαγωγή πεδίου-Μετάβαση 3, Διαγραφή πεδίου-Μετάβαση 5> λαμβάνεται υπόψιν διότι το ζεύγος <Εισαγωγή πεδίου, Μετάβαση 3> δεν έχει χρησιμοποιηθεί για την μέτρηση.

Οπότε σύμφωνα με τα παραπάνω ο τύπος που δίνει την υποστήριξη είναι:
#occurrences(r) = *Αριθμός εμφανίσεων της ακολουθίας r σε ολόκληρη την ιστορία της βάσης (χωρίς επικαλύψεις γεγονότος – μετάβασης για μία συγκεκριμένη ακολουθία και έναν συγκεκριμένο πίνακα)*
tcount = *Αριθμός μεταβάσεων στις οποίες συνέβησαν μία ή περισσότερες αλλαγές για όλους τις πίνακες της βάσης.*

$$\sigma_{CDIST}(r) = \frac{\#occurrences(r)}{tcount}$$

Η υποστήριξη και η εμπιστοσύνη ορίζονται ως εξής:

$$\text{Support} = \sigma(r) \qquad \text{Confidence} = \frac{\sigma(r1 \cup r2)}{\sigma(r1)}$$

Όπου το σ(r) εξαρτάται από το είδος της μέτρησης που εφαρμόζεται, δηλαδή σ(r) = σ_{COBJ} ή σ(r) = σ_{CDIST} . Επιπλέον r1 και r2 είναι δύο ακολουθίες γεγονότων.

Τυπικός ορισμός προβλήματος:

Το πρόβλημα εξόρυξης συχνών ακολουθιών μπορεί να διατυπωθεί ως εξής: Δοθείσης της ιστορίας μιας βάσης δεδομένων $\mathcal{H}$ στόχος είναι η παραγωγή συχνών ακολουθιών $\mathcal{R}$ των οποίων η υποστήριξη ξεπερνάει το ελάχιστο κατώφλι υποστήριξης (minsup).

3.1.3 Αλγόριθμος εξαγωγής συχνών ακολουθιών με τη μέθοδο Apriori

Συγκεντρώνοντας όλες τις ακολουθίες δεδομένων μιας βάσης, στόχος είναι η ανακάλυψη ακολουθιών που εμφανίζονται συχνά εφαρμόζοντας διάφορους αλγόριθμους εξόρυξης προτύπων. Το πλήθος των ακολουθιών είναι εκθετικά μεγάλο γι' αυτό και το κόστος των υπολογισμών για την ανακάλυψη των ακολουθιακών υποδειγμάτων αποτελεί πρόκληση. Πιο συγκεκριμένα αποδεικνύεται ότι το πλήθος των ακολουθιών, οι οποίες περιέχουν k γεγονότα και βρίσκονται σε μία ακολουθία δεδομένων με n γεγονότα είναι $\binom{n}{k}$. Το πλήθος των υποψήφιων ακολουθιών είναι σημαντικά μεγάλο διότι ένα γεγονός μπορεί να εμφανιστεί αρκετές φορές σε μία ακολουθία και επίσης η ταξινόμηση παίζει ρόλο στις ακολουθίες. Για παράδειγμα, τα γεγονότα <{attr_inserted} {attr_deleted}> και <{attr_deleted}{attr_inserted}> αντιστοιχούν σε διαφορετικές ακολουθίες. Το συγκεκριμένο πρόβλημα μπορεί να λυθεί με την εκ των πρότερων αρχή (apriori

principle) η οποία ισχύει στα ακολουθιακά δεδομένα και περιγράφεται με το παρακάτω θεώρημα.

Θεώρημα: Αν μία ακολουθία είναι συχνή, τότε όλες οι υποακολουθίες της πρέπει να είναι συχνές.

Σχήμα 3. Αλγόριθμος εξαγωγής συχνών ακολουθιών με τη μέθοδο Apriori

1. **Είσοδος**: Ιστορία βάσης δεδομένων (Database History), ελάχιστο κατώφλι υποστήριξης
2. **Έξοδος**: Συχνές ακολουθίες(Patterns)
3. **Begin**
4. k = 1
5. F(k) = Εύρεση όλων των συχνών υποακολουθιών μήκους 1
6. **Όσο** η λίστα F_k με τους υποψήφιους δεν είναι κενή επανάλαβε{
    a. k = k+1
    b. C_k = Παραγωγή υποψήφιων υποακολουθιών μεγέθους k
    c. **Για κάθε** ακολουθία δεδομένων από την βάση δεδομένων ακολουθιών **{**
        i. C_t = Προσδιορισμός όλων των υποψηφίων που περιέχονται στην βάση
        ii. **Για κάθε** υποψήφια υποακολουθία μήκους k που ανήκει στο Ct **{**
        iii. σ(c) = σ(c) + 1
        }
    }
    d. F(k) = Εξαγωγή των συχνών υποακολουθιών μήκους k
7. **}**
8. **Αποτέλεσμα** = F_k
9. **End**

Παραγωγή υποψηφίων

Για την παραγωγή μιας ακολουθίας μήκους k συγχωνεύονται δύο συχνές ακολουθίες μήκους k-1 η κάθε μια. Για την αποφυγή διπλότυπων υποψηφίων μήκους k ακολουθείται η παρακάτω διαδικασία συγχώνευσης:

Μία ακολουθία s_1 μπορεί να συγχωνευτεί με μια ακολουθία s_2 αν η ακολουθία που προκύπτει από την αφαίρεση του πρώτου γεγονότος από την s_1 είναι ίδια με την ακολουθία που προκύπτει από την αφαίρεση του τελευταίου γεγονότος της s_2. Αν διαπιστωθεί ότι οι δύο ακολουθίες μήκους k-1 μπορούν να συγχωνευτούν τότε η νέα ακολουθία που προκύπτει από την συνένωση είναι η ακολουθία s_1 συνενωμένη με το τελευταίο γεγονός της s_2. Το τελευταίο γεγονός της s_2 μπορεί να συγχωνευτεί με δύο διαφορετικούς τρόπους:

1. Αν τα δύο τελευταία γεγονότα στην ακολουθία s2 ανήκουν στην ίδια μετάβαση τότε το τελευταίο γεγονός στην s_2 ενσωματώνεται στο τελευταίο στοιχείο της s_1 . Για παράδειγμα, οι ακολουθίες s_1={1}{2 3}{4} και s_2= {2 3}{4 5} παράγουν την υποψήφια ακολουθία {1}{2 3}{4

5} διότι τα δύο τελευταία γεγονότα στην ακολουθία s2 (γεγονότα 4 και 5) ανήκουν στην ίδια μετάβαση.

2. Αν τα δύο τελευταία γεγονότα στην ακολουθία s_2 ανήκουν σε διαφορετικές μεταβάσεις τότε το τελευταίο γεγονός στην s_2 τοποθετείται σαν ξεχωριστό στοιχείο στο τέλος της s_1. Για παράδειγμα, οι ακολουθίες s_1={1}{2 3}{4} και s_2={2 3}{4}{5} παράγουν την υποψήφια ακολουθία {1}{2 3}{4}{5} διότι τα δύο τελευταία γεγονότα(γεγονότα 4 και 5) δεν ανήκουν στην ίδια μετάβαση.

3.1.4 Διαστασιολόγηση του προβλήματος

Η μελέτη του προβλήματος της εξόρυξης συχνών ακολουθιών από τις αλλαγές που έχει υποστεί το σχήμα μιας βάσης δεδομένων απαιτεί τον καθορισμό μιας τιμής για κάθε μία από 5 παραμέτρους. Οι εναλλακτικές τιμές κάθε παραμέτρου μας επιτρέπουν να μελετήσουμε το πρόβλημα από διαφορετικές σκοπιές. Στο σχήμα 4 φαίνονται όλοι οι διαφορετικοί τρόποι μελέτης του προβλήματος. Για κάθε μία παράμετρο (στήλη), ορίζουμε το επίπεδο της αφαίρεσης και σε κάθε στήλη του πίνακα στο σχήμα 4 σημειώνονται με έντονα γράμματα οι κατηγορίες οι οποίες είναι οι πιο αναλυτικές. Πιο συγκεκριμένα, το πρόβλημα αποτελείται από 5 επιμέρους συστατικά τα οποία αναλύονται παρακάτω:

1. <u>Αναπαράσταση γεγονότων:</u> Στην συγκεκριμένη κατηγορία αναφέρεται ο τρόπος με τον οποίο μπορούμε να ερμηνεύσουμε τα γεγονότα που λαμβάνουν χώρα στην ιστορία της βάσης. Διακρίνονται 3 περιπτώσεις:
   - Γεγονότα που αναφέρονται σε συγκεκριμένα πεδία: Τα γεγονότα αυτής της κατηγορίας αναφέρονται στο είδος του γεγονότος ακολουθούμενο από το πεδίο του πίνακα που επηρεάζεται από αυτή την αλλαγή. Τα γεγονότα αυτής της κατηγορίας είναι της μορφής <Είδος_γεγονότος(όνομα_πεδίου)>.
   - Είδος γεγονότων: Λαμβάνουμε υπόψιν μόνο το είδος του γεγονότος που συμβαίνει στην ιστορία ενός πίνακα για την εξαγωγή των συχνών ακολουθιών. Τα είδη των γεγονότων είναι: Εισαγωγή πεδίου σε υπάρχον πίνακα, εισαγωγή πεδίου σε πίνακα που δημιουργήθηκε την ίδια χρονική στιγμή με την εισαγωγή του πεδίου, διαγραφή πεδίου από πίνακα που παραμένει ζωντανός, διαγραφή πεδίου από πίνακα που διαγράφεται την ίδια χρονική στιγμή που διαγράφεται το πεδίο, εισαγωγή νέου πίνακα, διαγραφή πίνακα, αλλαγή στο τύπο κάποιου πεδίου, αλλαγή στο πρωτεύον κλειδί ενός πίνακα. Για παράδειγμα μία ακολουθία από γεγονότα αυτής της κατηγορίας είναι <Εισαγωγή_πεδίου, Διαγραφή_πεδίου, Εισαγωγή_πεδίου>

- Γεγονότα με αριθμό: Στην συγκεκριμένη περίπτωση τα γεγονότα αναφέρονται στο είδος του γεγονότος ακολουθούμενο από έναν ακέραιο αριθμό ο οποίος συμβολίζει το συνολικό πλήθος των γεγονότων του συγκεκριμένου είδους που συνέβησαν σε μια χρονική περίοδο. Τα γεγονότα αυτής την κατηγορίας είναι της μορφής <Είδος_γεγονότος(2)>.

2. Είδος αλλαγών:
   - Λίστα από αλλαγές: Η ιστορία κάθε πίνακα αποτελείται από μία λίστα από αλλαγές οι οποίες επηρεάζουν τον συγκεκριμένο πίνακα. Παραδείγματα τέτοιων αλλαγών αναφέρθηκαν στο κεφάλαιο 3.1.1.
   - Σχεσιακές αλλαγές: Η συγκεκριμένη κατηγορία αναφέρεται σε αλλαγές σχέσεων των πινάκων της βάσης. Παράδειγμα τέτοιων αλλαγών είναι <Εισαγωγή_Σχέσης, Διαγραφή_Σχέσης, Ανανέωση_Σχέσης>.
   - Αλλαγή ασχέτως σχέσης ή πεδίου: Η συγκεκριμένη κατηγορία είναι μια απλοποίηση της κατηγορίας «Λίστα από αλλαγές» στην οποία περιέχονται 3 ειδών αλλαγές οι οποίες είναι η εισαγωγή, η διαγραφή και η ενημέρωση.
   - Απλή αλλαγή: Σε αυτήν την κατηγορία θεωρείται ότι υπάρχει μόνο ένα είδους αλλαγής. Η συγκεκριμένη κατηγορία είναι μία γενική περίπτωση
3. Ομαδοποίηση πινάκων: Υπάρχουν τρεις διαφορετικές κατηγορίες ομαδοποίησης σε ότι αφορά τους πίνακες μιας βάσης:
   - Κατά πίνακα: Κάθε πίνακας μπορεί να θεωρηθεί από μόνος του μία ομάδα και έτσι να εξεταστεί ατομικά.
   - Κατά είδος πίνακα: Οι πίνακες μπορούν να κατηγοριοποιηθούν με αρκετούς τρόπους όπως για παράδειγμα ανάλογα το πλήθος των πεδίων που έχουν (μικροί, μεσαίοι, μεγάλοι) ή το είδος των εξαρτήσεων των πρωτευόντων κλειδιών τους.
   - Κατά ομάδες πινάκων: Οι πίνακες μπορούν να ομαδοποιηθούν με βάση τις σχέσεις μεταξύ τους. Πίνακες που σχετίζονται μεταξύ τους τοποθετούνται στην ίδια ομάδα.
4. Διαίρεση χρόνου: Αυτή η κατηγορία περιγράφει τους τρόπους με τους οποίους μπορεί να διαιρεθεί ο χρόνος για την μελέτη και την μέτρηση των γεγονότων που συμβαίνουν σε κάθε πίνακα. Υπάρχουν 4 διαφορετικοί τρόποι διαίρεσης του χρόνου:

- Version ID: Στην συγκεκριμένη ομαδοποίηση κάθε εκδοχή (version) λαμβάνεται και ελέγχεται ξεχωριστά για την εμφάνιση ενός γεγονότος της υποψήφιας ακολουθίας.
- Χρονικά σημεία: Η συγκεκριμένη ομαδοποίηση επιτρέπει την επιλογή χρονικών σημείων στα οποία λαμβάνονται οι εμφανίσεις των γεγονότων των υποψήφιων ακολουθιών. Ο συγκεκριμένος τρόπος είναι παρόμοιος την διαίρεση χρόνου Version ID με την μόνη διαφορά ότι η ομαδοποίηση γίνεται με βάση τον χρόνο και όχι με βάση την εκδοχή της βάσης.
- Χρονικό διάστημα: Μπορεί να χρησιμοποιηθούν χρονικοί περιορισμοί ανάμεσα στην πρώτη και την τελευταία εμφάνιση των γεγονότων μιας ακολουθίας. Σε αυτή την κατηγορία υπάρχουν παράμετροι όπως οι Maximum Span, Event-set Window Size, Maximum Gap, Minimum Gap οι οποίες περιγράφονται στο προηγούμενο κεφάλαιο και ορίζουν ένα χρονικό παράθυρο στο οποίο επιτρέπουν εμφανίσεις των γεγονότων.
- Φάσεις: Σε αυτήν την περίπτωση ο χρόνος μπορεί να οργανωθεί σε φάσεις με βάση την κατανομή των γεγονότων στο χρόνο.

5. Είδος μέτρησης: Το είδος μέτρησης αναφέρεται στον τρόπο με τον οποίο μπορεί να διαπιστωθεί αν μία ακολουθία είναι συχνή. Υπάρχουν αρκετοί τρόποι μέτρησης και κάποιοι από αυτούς είναι: COBJ, CDIST, CDIST_O, CWIN, CMINWIN. Οι συγκεκριμένοι τρόποι μέτρησης αναφέρονται παραπάνω.

Στην συγκεκριμένη εργασία χρησιμοποιούνται:

- Αναπαράσταση γεγονότων: Είδη αλλαγών
- Είδος αλλαγών: λίστα αλλαγών
- Ομαδοποίηση πινάκων: κατά πίνακα
- Ομαδοποίηση χρόνου: version ID
- Είδος μέτρησης: COBJ, CDIST

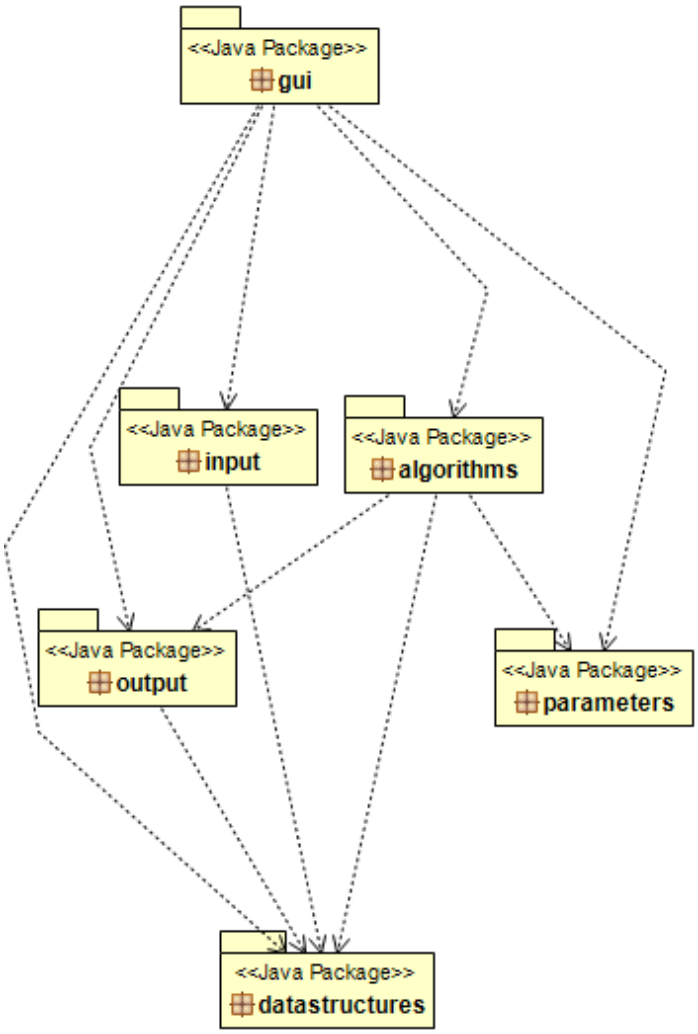
Σχήμα 4. Πίνακας Διαστασιολόγησης

Αναπαράσταση Γεγονότων	Είδος Αλλαγών	Ομαδοποίηση Πινάκων	Ομαδοποίηση Χρόνου	Είδος Μέτρησης
• **Είδη αλλαγών** • Είδος γεγονότος ακολουθούμενο από το όνομα του πεδίου • Είδος γεγονότος ακολουθούμενο από πλήθος	• **Λίστα αλλαγών** • Σχεσιακές αλλαγές • Αλλαγές ανεξάρτητες από σχέση και πεδία • Αλλαγή οποιουδήποτε είδους	• **Κατά πίνακα** • Κατά είδος πίνακα • Κατά ομάδες πινάκων	• **Version ID** • Χρονικά σημεία • Χρονικό διάστημα • Φάσεις	• **COBJ** • CWIN • CMINWIN • CDIST • **CDIST_O**

3.2 Σχεδίαση και αρχιτεκτονική λογισμικού

Όπως αναφέρθηκε στα προηγούμενα κεφάλαια, ο στόχος του συγκεκριμένου λογισμικού είναι η ανακάλυψη συχνών ακολουθιών και η εμφάνιση τους στον χρήστη με έναν διαδραστικό τρόπο. Για την υλοποίηση του λογισμικού σχεδιάστηκαν και υλοποιήθηκαν οι κατάλληλες κλάσεις οι οποίες έχουν χωριστεί σε 6 πακέτα συνολικά: input, datastructures, parameters, algorithm, output και gui τα οποία αναλύονται παρακάτω.

Σχήμα 5. Διάγραμμα πακέτων



3.2.1 Πακέτο input

Το συγκεκριμένο πακέτο περιέχει τις κλάσεις οι οποίες χρησιμοποιούνται για την ανάγνωση των αρχείων εισόδου και την αποθήκευση τους σε κατάλληλες δομές. Το πακέτο αποτελείται από την κλάση TransitionsParser η οποία είναι υπεύθυνη για την ανάγνωση του επιλεγμένου αρχείου εισόδου και την αποθήκευση του σε μία κατάλληλη δομή, την TableSequenceHistory η οποία ανήκει στο πακέτο datastructures. Το αρχείο εισόδου περιέχει όλα τα γεγονότα που έχουν συμβεί στην ιστορία μιας βάσης για κάθε πίνακα και για κάθε μετάβαση. Από τα παραπάνω προκύπτει ότι η κλάση TransitionsParser εξαρτάται από την κλάση TableSequenceHistory και η εξάρτηση φαίνεται στο σχήμα 7.

3.2.2 Πακέτο datastructures

Το πακέτο αυτό περιέχει όλες τις κλάσεις οι οποίες είναι απαραίτητες για την αποθήκευση των δεδομένων που χρειάζεται το εργαλείο. Πιο συγκεκριμένα οι βασικές κλάσεις από τις οποίες αποτελείται το πακέτο είναι οι κλάσεις TableHistorySequence, TableHistoryElements, AtomicChangeEvent, Details, TableInfo και TransitionInfo.

Η κλάση TableHistorySequence κρατάει όλες τις πληροφορίες για την ιστορία ενός πίνακα όπως ορίστηκε στο κεφάλαιο 3.1.1, δηλαδή μία λίστα από στοιχεία(Table HistoryElements) που αναφέρονται σε έναν πίνακα. Η συγκεκριμένη κλάση εξαρτάται από την κλάση TableHistorySequence και από την κλάση TableInfo οι οποίες περιγράφονται παρακάτω.

Η κλάση TableHistoryElements κρατάει τις πληροφορίες οι οποίες αναφέρονται σε ένα στοιχείο πίνακα όπως αυτό ορίστηκε στο κεφάλαιο 3.1.1. Η κλάση αυτή εξαρτάται από τις κλάσεις TransitionInfo και AtomicChangeEvent οι οποίες αποτελούν τα ζευγάρια γεγονότος – transition.

Η κλάση AtomicChangeEvent είναι μία αφηρημένη κλάση η οποία περιγράφει την γενική οντότητα του γεγονότος όπως αυτό ορίστηκε στο προηγούμενο κεφάλαιο. Την συγκεκριμένη κλάση επεκτείνουν οι κλάσεις: AttributeAdditionAtExistingTable, AttributeAdditionAtTableCreation, Attribute DeletionAtExistingTable, AttributeDeletion AtTableDeletion, TableCreation, TableDeletion, PrimaryKeyUpdate, AttributeDataTypeUpdate οι οποίες προσδιορίζουν τα διαφορετικά είδη γεγονότων.

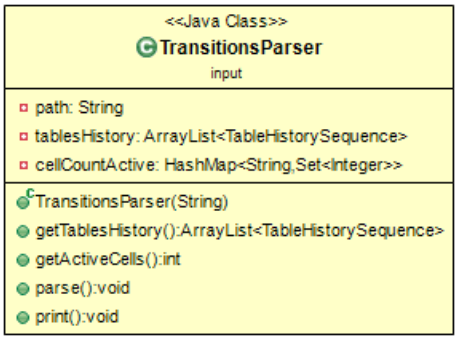
Η κλάση TableInfo είναι μία αφηρημένη κλάση η οποία είναι σχεδιασμένη για να επιτρέπει την επέκταση για την αποθήκευση των πληροφοριών σχετικά με την ομαδοποίηση πινάκων, όπως αναφέρθηκε στο κεφάλαιο 3.1.4. Στην συγκεκριμένη έκδοση του λογισμικού υποστηρίζεται μόνο η κατηγορία «Κατά πίνακα» (κλάση Table που επεκτείνει την TableInfo).

Η κλάση TransitionInfo είναι μία αφηρημένη κλάση η οποία είναι σχεδιασμένη για να επιτρέπει την επέκταση για την αποθήκευση των πληροφοριών σχετικά με την ομαδοποίηση του χρόνου όπως περιγράφηκε στο κεφάλαιο 3.1.4. Στην συγκεκριμένη έκδοση του εργαλείου υποστηρίζεται μόνο η κατηγορία «Version ID» (κλάση Transition που επεκτείνει την TransitionInfo).

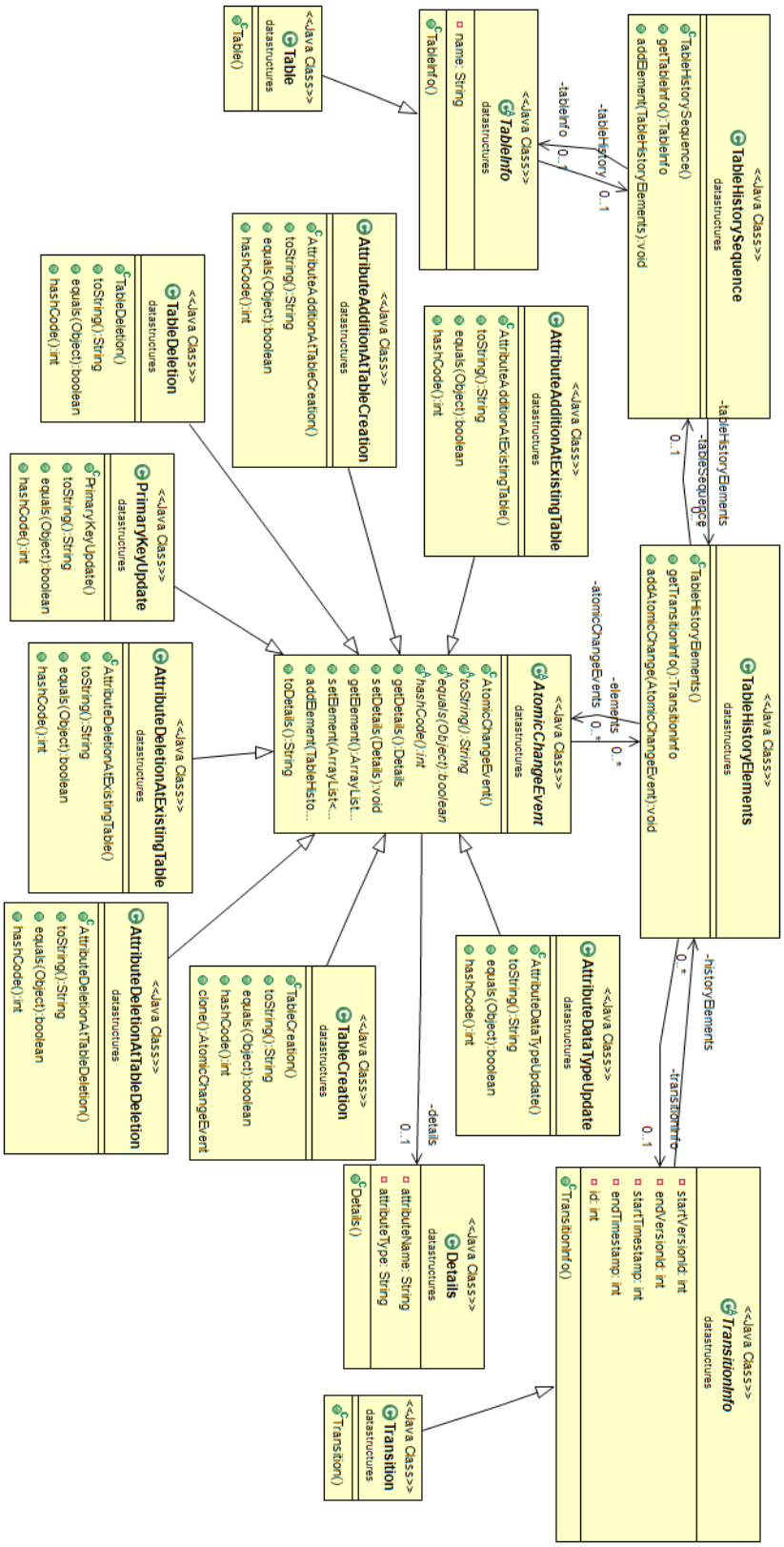
3.2.3 Πακέτο parameters

Το συγκεκριμένο πακέτο περιέχει τις κλάσεις οι οποίες είναι απαραίτητες για την αποθήκευση των παραμέτρων που χρειάζονται οι αλγόριθμοι του εργαλείου. Πιο συγκεκριμένα, περιέχει τις κλάσεις Parameters και AprioriSequenceParameters. Η κλάση Parameters είναι μία αφηρημένη κλάση η οποία έχει δημιουργηθεί για την καλύτερη επεκτασιμότητα του εργαλείου. Η κλάση AprioriSequenceParameters διατηρεί όλες τις πληροφορίες τις οποίες χρειάζεται ο αλγόριθμος εξόρυξης προτύπων τύπου Apriori για να εκτελεστεί. Αυτές οι πληροφορίες είναι για παράδειγμα το ελάχιστο κατώφλι υποστήριξης και η μέθοδος μέτρησης (CDIST ή COBJ).

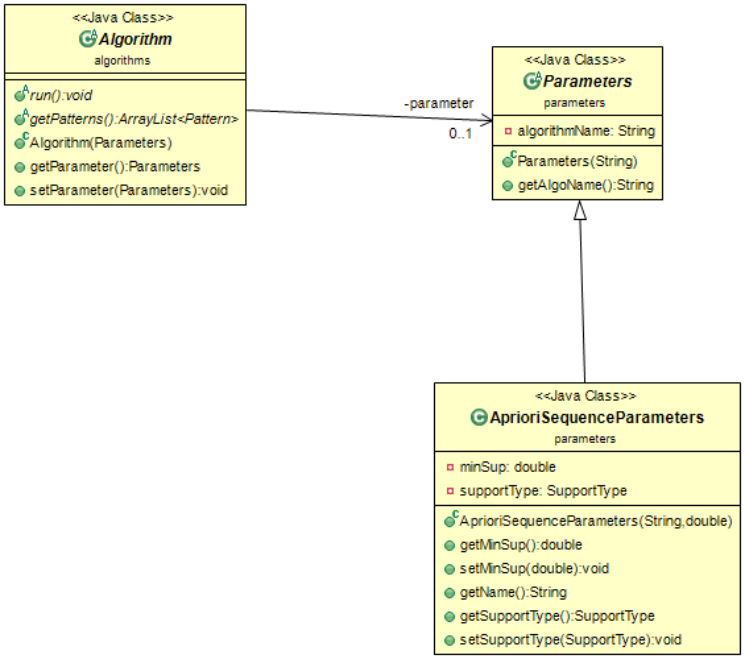
Σχήμα 6. UML διάγραμμα για την κλάση του πακέτου input



Σχήμα 7. UML Διάγραμμα για τις κλάσεις του πακέτου datastructures



Σχήμα 8. UML διάγραμμα για τις κλάσεις του πακέτου parameters



3.2.4 Πακέτο algorithm

Το συγκεκριμένο πακέτο περιέχει τις κλάσεις οι οποίες είναι υπεύθυνες για την υλοποίηση των αλγορίθμων εξόρυξης προτύπων. Πιο συγκεκριμένα, το πακέτο περιέχει τις κλάσεις Algorithm, AprioriSequenceAlgo, CandidateGenarator, SupportCounter, CDISTCounter, COBJCounter.

Η κλάση Algorithm είναι μία αφηρημένη κλάση που έχει αφηρημένες μεθόδους για την διευκόλυνση της επεκτασιμότητας.

Η κλάση AprioriSequenceAlgo υλοποιεί τον αλγόριθμο εξαγωγής συχνών ακολουθιών με τη μέθοδο Apriori που περιγράφηκε στο κεφάλαιο 3.1.3. Η συγκεκριμένη κλάση εξαρτάται από τις κλάσεις CandidateGenerator, SupportCounter, TableHistorySequence και Pattern όπως φαίνεται από το UML διάγραμμα του σχήματος 9.

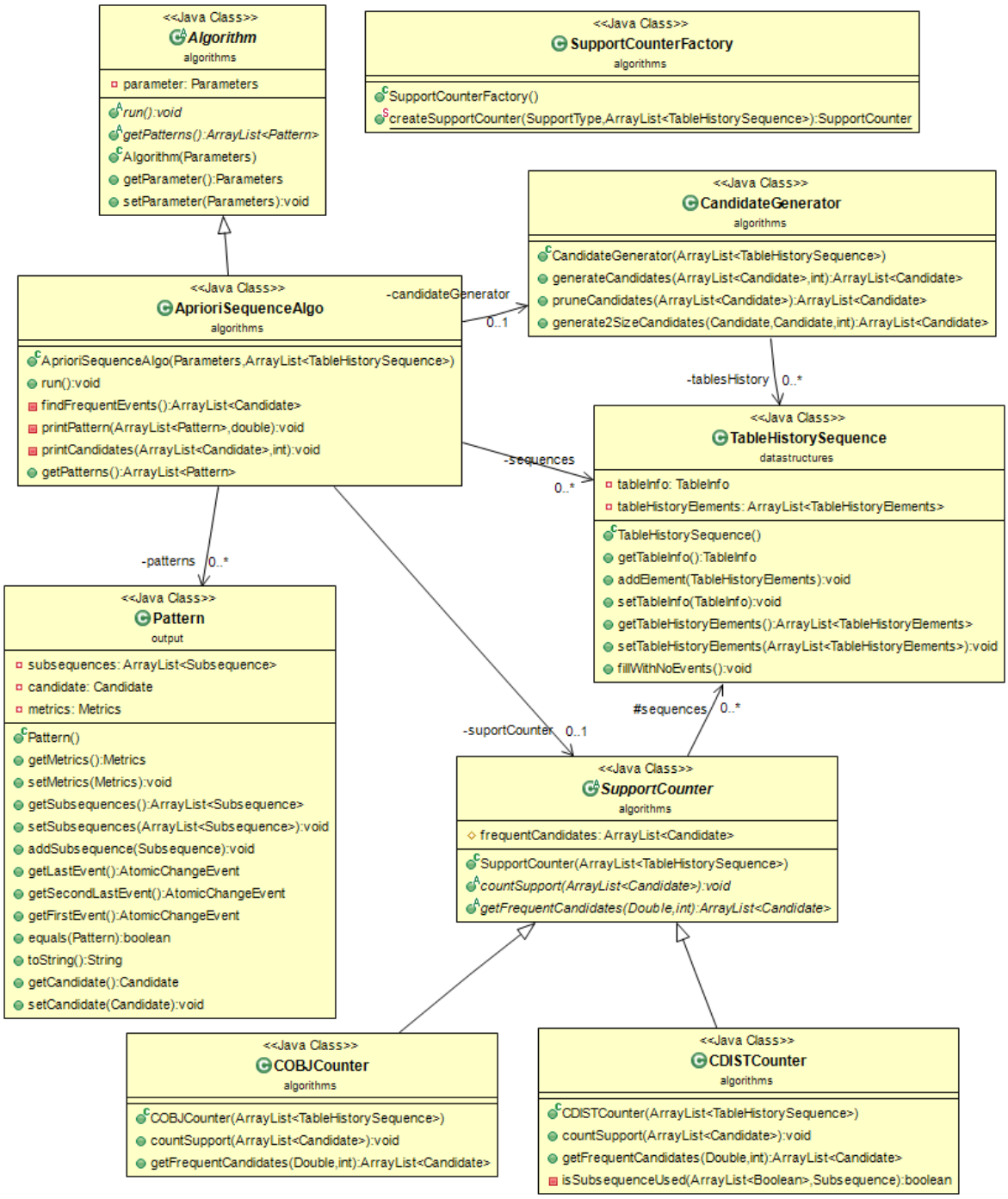
Η κλάση CandidateGenerator είναι υπεύθυνη για την παραγωγή των υποψήφιων ακολουθιών με την μέθοδο που αναφέρθηκε στο κεφάλαιο 3.1.3. Επίσης κλαδεύει τους υποψήφιους οι οποίοι δεν βρίσκονται στα δεδομένα εισόδου. Η κλάση CandidateGenerator εξαρτάται από την κλάση TableHistorySequence.

Η κλάση SupportCounter είναι μία αφηρημένη κλάση η οποία περιέχει αφηρημένες μεθόδους που υλοποιούν διαφορετικούς τρόπους μέτρησης της υποστήριξης των υποψήφιων ακολουθιών.

Η κλάση COBJCounter επεκτείνει την κλάση SupportCounter υλοποιώντας την μέτρηση υποστήριξης των υποψήφιων ακολουθιών COBJ όπως αυτή αναφέρθηκε στο κεφάλαιο 3.1.2.

Η κλάση CDISTCounter επεκτείνει την κλάση SupportCounter υλοποιώντας την μέτρηση υποστήριξης των υποψήφιων ακολουθιών CDIST όπως διατυπώθηκε στο κεφάλαιο 3.1.2.

Σχήμα 9. UML διάγραμμα για τις κλάσεις του πακέτου algorithm



3.2.5 Πακέτο output

Το συγκεκριμένο πακέτο αποτελείται από τις κλάσεις οι οποίες είναι υπεύθυνες για την αποθήκευση των συχνών ακολουθιών, των υποψήφιων ακολουθιών καθώς επίσης και κάποιων χρήσιμων μετρικών. Πιο συγκεκριμένα το πακέτο περιέχει τις κλάσεις Pattern, Candidate, Subsequence και Metrics.

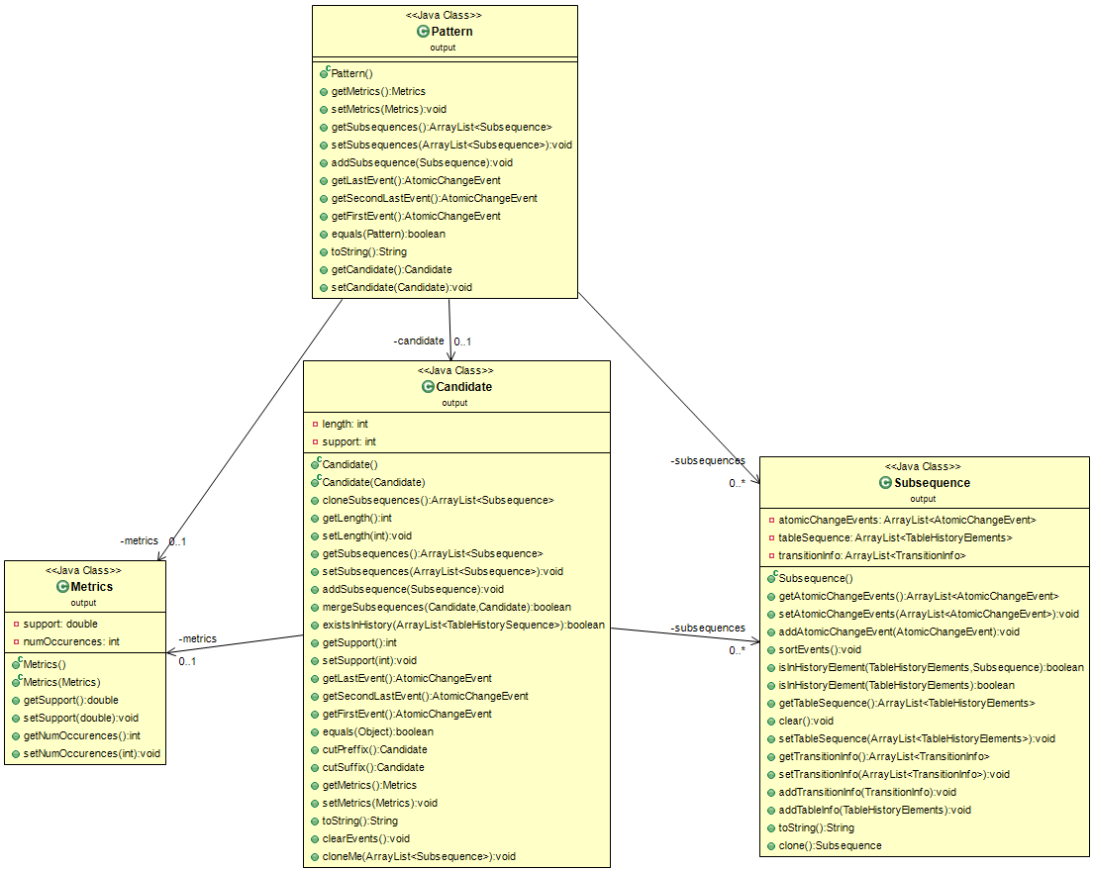
Η κλάση Pattern κρατάει την πληροφορία για μία συχνή ακολουθία όπως αυτή υπολογίστηκε από τον αλγόριθμο εξόρυξης συχνών ακολουθιών. Επίσης η συγκεκριμένη κλάση έχει μια αναφορά στην υποψήφια ακολουθία από την οποία προήλθε (κλάση Candidate). Τέλος, έχει σαν πεδίο την κλάση Metrics η οποία αναλύεται παρακάτω.

Η κλάση Candidate χρησιμοποιείται για την αποθήκευση των υποψήφιων ακολουθιών για τις οποίες πρέπει να διαπιστωθεί αν είναι συχνές. Επιπλέον, υλοποιεί κάποιες λειτουργίες οι οποίες βοηθούν στην συγχώνευση ακολουθιών και τον έλεγχο της ύπαρξης της υποψήφιας ακολουθίας στην ιστορία του πίνακα όπως αυτά περιγράφηκαν στο προηγούμενο κεφάλαιο. Η κλάση Candidate εξαρτάται από τις κλάσεις Metrics και Subsequences.

Η κλάση Subsequence είναι υπεύθυνη για την αποθήκευση των γεγονότων που συμβαίνουν σε κάποια χρονική περίοδο σε μία κατηγορία πινάκων. Για την επίτευξη αυτού του σκοπού εξαρτάται από τις κλάσεις AtomicChangeEvent, TableHistoryElements και TransitionInfo οι οποίες περιγράφηκαν παραπάνω.

Η κλάση Metrics αποθηκεύει χρήσιμες μετρικές για μία συχνή ακολουθία. Οι μετρικές αυτές είναι η υποστήριξη και ο αριθμός εμφάνισης της συχνής ακολουθίας.

Σχήμα 10. UML Διάγραμμα για τις κλάσεις του πακέτου output



3.2.6 Πακέτο gui

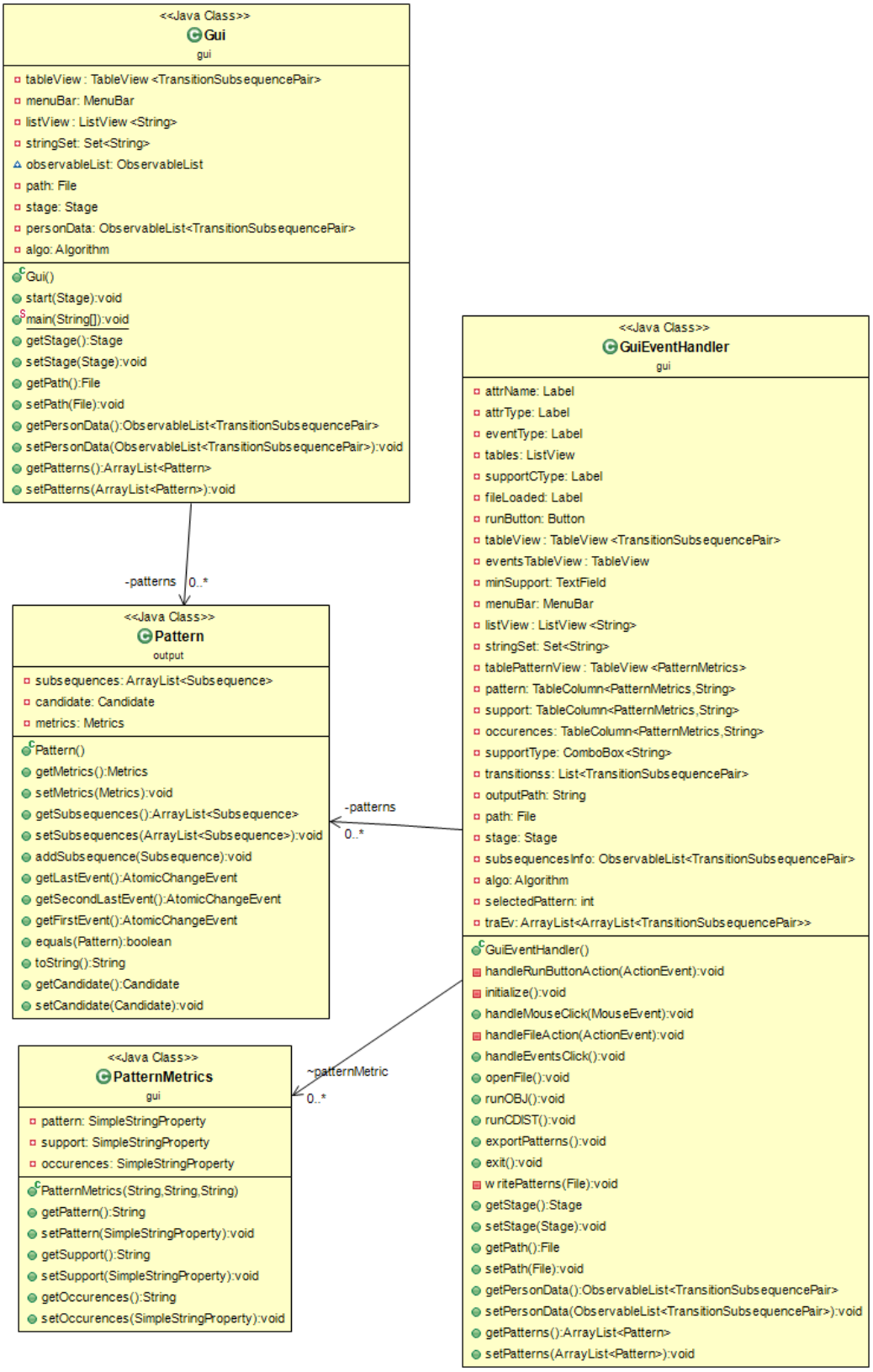
Το συγκεκριμένο πακέτο περιέχει τις κλάσεις οι οποίες προσφέρουν την γραφική διεπαφή του χρήστη με το εργαλείο μέσω της οποίας μπορούν να επικοινωνούν καθώς και κάποιες βοηθητικές δομές για την απεικόνιση των δεδομένων στον χρήστη. Πιο συγκεκριμένα, το πακέτο αποτελείται από τις κλάσεις Gui, GuiEventHandler και PatternMetrics.

Η κλάση Gui αρχικοποιεί και εμφανίζει στον χρήστη το βασικό παράθυρο του λογισμικού.

Η κλάση GuiEventHandler αλληλεπιδρά με το χρήστη αναγνωρίζοντας τις ενέργειες του. Η συγκεκριμένη κλάση, ανάλογα τις επιλογές του χρήστη, διαβάζει ένα έγκυρο αρχείο εισόδου και εκτελεί τον κατάλληλο αλγόριθμο εμφανίζοντας τα αποτελέσματα του στην οθόνη. Επιπλέον είναι υπεύθυνη για την εγγραφή των αρχείων εξόδου τα οποία περιέχουν τα αποτελέσματα του αλγόριθμου εξόρυξης προτύπων.

Η κλάση PatternMetrics βοηθητική κλάση για την προσωρινή αποθήκευση των δεδομένων εξόδου του αλγόριθμου για την αναπαράσταση τους στον πίνακα με τις συχνές ακολουθίες.

Σχήμα 11. UML Διάγραμμα για τις κλάσεις του πακέτου gui



Κεφάλαιο 4. Υλοποίηση

4.1 Πλατφόρμες και προγραμματιστικά εργαλεία

Το συγκεκριμένο λογισμικό αναπτύχθηκε στην προγραμματιστική γλώσσα Java (http://www.java.com) η οποία είναι μία αντικειμενοστρεφής γλώσσα προγραμματισμού με πολλά πλεονεκτήματα και ευελιξία. Με την συγκεκριμένη γλώσσα προγραμματισμού είναι εύκολο να δημιουργηθεί επαναχρησιμοποιήσιμος κώδικας ο οποίος είναι εύκολο να μεταφερθεί από ένα υπολογιστικό σύστημα σε κάποιο άλλο. Επίσης η Java είναι μία γλώσσα που προσφέρει μεγάλη ασφάλεια και αξιοπιστία και σε αυτό οφείλεται η μεγάλη της απήχηση.

Το περιβάλλον ανάπτυξης που χρησιμοποιήθηκε για την συγγραφή του κώδικα είναι το eclipse (https://eclipse.org/) το οποίο είναι ένα πολύ διαδεδομένο εργαλείο. Το γεγονός ότι το eclipse είναι ένα δωρεάν και ανοιχτό λογισμικό σε συνδυασμό με τις μεγάλες δυνατότητες επεκτασιμότητας και παραμετροποίησης του συνέβαλε καθοριστικά στην επιλογή του ως περιβάλλον ανάπτυξης. Επίσης, η προσθήκη πρόσθετων(Plug-ins) κάνει το eclipse κάτι παραπάνω από ένα απλό περιβάλλον ανάπτυξης καθώς με τα πρόσθετα μπορούν να υποστηριχθούν επιπλέον δυνατότητας όπως είναι για παράδειγμα το ObjectAid (http://www.objectaid.com) το οποίο δίνει έναν εύκολο τρόπο απεικόνισης των UML διαγραμμάτων λαμβάνοντας τις απαραίτητες πληροφορίες από τις κλάσεις που έχουν οριστεί στο eclipse. Τέλος η υποστήριξη του eclipse καθώς και οι αναβαθμίσεις που δέχεται είναι συχνές και έτσι συμβαδίζει με την ανάπτυξη της τεχνολογίας.

Για τον σχεδιασμό και την υλοποίηση του γραφικού περιβάλλοντος της εφαρμογής χρησιμοποιήθηκε η πλατφόρμα λογισμικού JavaFX (http://www.oracle.com/technetwork/java/javafx/overview/index.html) η οποία είναι μία πλατφόρμα για την δημιουργία εφαρμογών για υπολογιστές οι οποίες

μπορούν να τρέξουν σε μια πληθώρα συσκευών. Η πλατφόρμα JavaFX είναι αρκετά διαδεδομένη τα τελευταία χρόνια και έρχεται να αντικαταστήσει την Swing λόγω των επιπλέον δυνατοτήτων που προσφέρει. Κάποια από τα πλεονεκτήματα της είναι η ύπαρξη διαγραμμάτων, τρισδιάστατη υποστήριξη περιεχομένου, υψηλή απόδοση κα.

4.2 Λεπτομέρειες υλοποίησης

Αρχικά, για την εκτέλεση του αλγορίθμου ο χρήστης πρέπει να επιλέξει κάποιες παραμέτρους από το γραφικό περιβάλλον που εμφανίζεται. Πιο συγκεκριμένα, πρέπει να επιλέξει ένα αρχείο εισόδου που περιέχει όλες τις αλλαγές που έχει υποστεί η βάση, ένα ελάχιστο κατώφλι υποστήριξης και ένα είδος μέτρησης της υποστήριξης. Το αρχείο εισόδου περιέχει την ιστορία μιας βάσης δεδομένων, όπως αυτή ορίστηκε στο κεφάλαιο 3.1.1, και έχει την εξής μορφή: αποτελείται από 10 στήλες οι οποίες είναι χωρισμένες με τον χαρακτήρα semicolon (“;”).

- Πρώτη στήλη (trID): αναφέρεται στον αριθμό της μετάβασης
- Δεύτερη στήλη (oldVer): αναφέρεται στην παλιά εκδοχή της βάσης.
- Τρίτη στήλη (newVer): αναφέρεται στη νέα εκδοχή της βάσης ύστερα από την μετάβαση t_i η οποία αναφέρεται στην πρώτη κολώνα.
- Τέταρτη στήλη (Table): αναφέρεται στο όνομα του πίνακα στον οποία συμβαίνει κάποια αλλαγή.
- Πέμπτη στήλη (eventType): περιγράφει το είδος του γεγονότος που συμβαίνει. Τα είδη των γεγονότων που μπορούν να συμβούν είναι τα εξής:
    - Deletion:DeleteTable – Διαγραφή πεδίου όταν ο πίνακας διαγράφεται.
    - Insertion:NewTable – Εισαγωγή πεδίου όταν ο πίνακας δημιουργείται.
    - Update:TypeChange – Αλλαγή στον τύπο ενός πεδίου.
    - Insertion:UpdateTable – Εισαγωγή ενός πεδίου σε υπάρχοντα πίνακα.
    - Deletion:UpdateTable – Διαγραφή ενός πεδίου σε πίνακα ο οποίος δεν διαγράφεται.
    - Update:KeyChange – Αλλαγή στο πρωτεύον κλειδί.
- Έκτη στήλη (attrName): αναφέρεται στο όνομα του πεδίου το οποίο επηρεάζεται από την αλλαγή.

- Έβδομη στήλη (attrType): αναφέρεται στον τύπο του πεδίου που δέχεται την αλλαγή.
- Οι στήλες 8, 9 και 10 αναφέρονται σε πληροφορίες σχετικά με τα κλειδιά αλλά δεν λαμβάνονται υπόψιν στην παρούσα εργασία.

Ένα παράδειγμα αρχείου εισόδου φαίνεται στο σχήμα 12. Αξίζει να σημειωθεί ότι αν σε κάποια μετάβαση δεν υπάρχει κανένα γεγονός τότε οι στήλες 4-10 συμπληρώνονται με «-».

Σχήμα 12. Παράδειγμα αρχείου εισόδου.

trID	oldVer	newVer	Table	EventType	attrName	attrType	iskey	pkey	fkey
1	1012181431.sql	1014631726.sql	location_qualifier_value	Insertion:UpdateTable	qualifier_int_value	INT(10)	FALSE	0	-
1	1012181431.sql	1014631726.sql	location_qualifier_value	Deletion:UpdateTable	slot_value	INT(10)	FALSE	0	-
2	1014631726.sql	1014707807.sql	-	-	-	-	-	-	-
3	1014707807.sql	1014889243.sql	bioentry_date	Deletion:DeleteTable	bioentry_id	INT(10)	TRUE	0	bioentry_id@bioentry
3	1014707807.sql	1014889243.sql	bioentry_date	Deletion:DeleteTable	date	VARCHAR(200)	TRUE	0	-
3	1014707807.sql	1014889243.sql	bioentry_description	Deletion:DeleteTable	bioentry_id	INT(10)	FALSE	0	bioentry_id@bioentry
3	1014707807.sql	1014889243.sql	bioentry_description	Deletion:DeleteTable	description	VARCHAR(255)	FALSE	0	-
3	1014707807.sql	1014889243.sql	bioentry_direct_links	Deletion:UpdateTable	accession	VARCHAR(40)	FALSE	0	-
3	1014707807.sql	1014889243.sql	bioentry_direct_links	Deletion:UpdateTable	dbname	VARCHAR(40)	FALSE	0	-
3	1014707807.sql	1014889243.sql	bioentry_direct_links	Insertion:UpdateTable	dbxref_id	INT(10)	FALSE	0	dbxref_id@dbxref
3	1014707807.sql	1014889243.sql	bioentry_keywords	Deletion:DeleteTable	bioentry_id	INT(10)	TRUE	0	bioentry_id@bioentry
3	1014707807.sql	1014889243.sql	bioentry_keywords	Deletion:DeleteTable	keywords	VARCHAR(255)	FALSE	0	-
3	1014707807.sql	1014889243.sql	bioentry_qualifier_value	Insertion:NewTable	bioentry_id	INT(10)	FALSE	0	bioentry_id@bioentry
3	1014707807.sql	1014889243.sql	bioentry_qualifier_value	Insertion:NewTable	ontology_term_id	INT(10)	FALSE	0	ontology_term_id@ont
3	1014707807.sql	1014889243.sql	bioentry_qualifier_value	Insertion:NewTable	qualifier_value	MEDIUMTEXT	FALSE	0	-
3	1014707807.sql	1014889243.sql	biosequence	Insertion:UpdateTable	seq_length	INT(10)	FALSE	0	-

Στα σχήματα 13 και 14 εμφανίζονται δύο μέθοδοι οι οποίες διαβάζουν αυτές τις παραμέτρους και εκτελούν είτε την ανάγνωση του αρχείου(σχήμα 13) είτε την εκτέλεση του αλγορίθμου(σχήμα 14).

Σχήμα 13. Κλήση για διάβασμα του αρχείου εισόδου.

```
@FXML
private void handleFileAction(final ActionEvent event){
        FileChooser fileChooser = new FileChooser();
       fileChooser.setTitle("Open Resource File");
       File file = fileChooser.showOpenDialog(stage);
       if(file != null){
              setPath(file);
              fileLoaded.setText("File loaded: " + file.getAbsolutePath());
              Alert alert = new Alert(AlertType.INFORMATION);
              alert.setTitle("Successfull load");
              alert.setHeaderText("Success!");
              alert.setContentText("The file was loaded successfully!!");
              alert.showAndWait();
       }
       else{
              Alert alert = new Alert(AlertType.INFORMATION);
              alert.setTitle("Unsuccessfull load");
              alert.setHeaderText("Error!");
              alert.setContentText("The file was not loaded successfully!!");
              alert.showAndWait();
       }

}
```

Σχήμα 14. Διάβασμα παραμέτρων και εκτέλεση αλγορίθμου

```
@FXML
private void handleRunButtonAction(ActionEvent event) {
        String path = this.path.getAbsolutePath();
       TransitionsParser trParser = new TransitionsParser(path);
       trParser.parse();
       ArrayList<TableHistorySequence> tablesHistory =
             trParser.getTablesHistory();
       patternMetric.clear();
       tablePatternView.getItems().clear();
       Double minSup;
       if(minSupport.getText().isEmpty()){
              minSup = 0.01;
       }
       else{
           minSup = Double.parseDouble(minSupport.getText());
       }

       AprioriSequenceParameters param = new AprioriSequenceParameters
                                                 ("Apriori", minSup);
       if(supportType.getValue().equals("COBJ")){
           param.setSupportType(SupportType.COBJ);
       }
       else{
           param.setSupportType(SupportType.CDIST);
       }
       algo = new AprioriSequenceAlgo(param, tablesHistory);
       long startTime = System.nanoTime();
       algo.run();
       long stopTime = System.nanoTime();
       //Print algotithm execution time in sec
       double seconds =  (double)(stopTime - startTime) / 1000000000.0;
       System.out.println("Algorithm execution took: " + seconds);
       setPatterns(algo.getPatterns());
       for(Pattern pattern:this.getPatterns()){
              String patter ="<";
              for(Subsequence subSeq: pattern.getSubsequences()){
```

```
                        patter += "{";
                        for(int i=0; i < subSeq.getAtomicChangeEvents().size();
                          i++){
                              if(i == subSeq.getAtomicChangeEvents().size()-1)
                                    patter +=
                            subSeq.getAtomicChangeEvents().get(i).toString();
                              else
                                    patter +=
                        subSeq.getAtomicChangeEvents().get(i).toString()+",";
                        }
                        patter += "}";
                        }
                        patter += ">";
                        PatternMetrics pm = new PatternMetrics(patter,
Double.toString(pattern.getMetrics().getSupport()),
Integer.toString(pattern.getMetrics().getNumOccurences()));
                patternMetric.add(pm);
        }
        pattern.setCellValueFactory(new PropertyValueFactory<PatternMetrics,
String>("pattern"));
        support.setCellValueFactory(new PropertyValueFactory<PatternMetrics,
String>("support"));
        occurences.setCellValueFactory(new PropertyValueFactory<PatternMetrics,
String>("occurences"));
        tablePatternView.getItems().setAll(patternMetric);
        supportCType.setText("Support type: " + supportType.getValue());

}
```

Η βασική δομή του αλγόριθμου υλοποιείται στην μέθοδο run() της κλάσης AprioriSequenceAlgo. Αρχικά, αρχικοποιούνται οι τιμές των παραμέτρων και στη συνέχεια ανακαλύπτονται τα συχνά γεγονότα(υποψήφιες ακολουθίες μήκους 1) με την κλήση της findFrequentEvents(). Έπειτα, επαναληπτικά παράγονται νέες υποψήφιες ακολουθίες μήκους k με βάση τις συχνές ακολουθίες μήκους k-1. Αμέσως μετά την παραγωγή των υποψηφίων εκτελείται η μέτρηση υποστήριξης των συγκεκριμένων υποψήφιων και κλαδεύονται οι ακολουθίες οι οποίες έχουν υποστήριξη μικρότερη από το ελάχιστο κατώφλι. Τέλος, οι ακολουθίες που πέρασαν το στάδιο του κλαδέματος μπορούν πλέον να θεωρηθούν ως συχνές και έτσι αποθηκεύονται σε μία λίστα από συχνές ακολουθίες(`ArrayList<Candidate> patterns`) μαζί με τις μετρικές οι οποίες αφορούν τον αριθμό των εμφανίσεων και την υποστήριξη της συχνής ακολουθίας. Η μέθοδος run() φαίνεται στο σχήμα 15.

Σχήμα 15. Βασική δομή αλγορίθμου τύπου συχνών στοιχειοσυνόλων.

```
public void run() {
        int k = 1;
        Parameters parameter = super.getParameter();
        double minSup = ((AprioriSequenceParameters) parameter).getMinSup();
        ArrayList<Candidate> candidates = findFrequentEvents();
        //used for last time frequent patterns has patterns
        ArrayList<Candidate> previousFrequentCandidates = candidates;
        while(candidates != null && !candidates.isEmpty()){
                k++;
                previousFrequentCandidates = candidates;
                candidates = candidateGenerator.generateCandidates(candidates, k);
                //Count support for each candidate and keep the frequent only
                suportCounter.countSupport(candidates);
                //FilleroutAneparkeis
                candidates = suportCounter.getFrequentCandidates(minSup,k);
                for(Candidate candidate:candidates){
                        Pattern pattern = new Pattern();
                        Metrics metrics = new Metrics();
                        metrics.setNumOccurences(candidate.getMetrics().getNumOccurences());
                        metrics.setSupport(candidate.getMetrics().getSupport());
                        pattern.setMetrics(metrics);
                        for(int i=0;i<candidate.getSubsequences().size();i++){
                                Subsequence sub = new Subsequence();
                                ArrayList<AtomicChangeEvent> ev =candidate.getSubsequences(
                                                ).get(i).getAtomicChangeEvents();
                                for(int j=0;j<ev.size();j++){
                                        sub.addAtomicChangeEvent(ev.get(j));
                                }
                                ArrayList<TableHistoryElements> th = new ArrayList
                                          <TableHistoryElements>();
                                for(TableHistoryElements the: candidate.getSubsequences(
                                      ).get(i).getTableSequence()){
                                        th.add(the);
                                }
                                sub.setTableSequence(th);
                                pattern.addSubsequence(sub);
                                //patterns.add(pattern);
                        }
                        patterns.add(pattern);
                }
                for(int i=0;i < previousFrequentCandidates.size(); i++){
                    Pattern pattern = new Pattern();
                    pattern.setMetrics(previousFrequentCandidates.get(i).getMetrics());
                    pattern.setSubsequences(
                                  previousFrequentCandidates.get(i).getSubsequences());
                    pattern.setCandidate(previousFrequentCandidates.get(i));
                }
        }
}
```

Η παραγωγή των υποψήφιων ακολουθιών πραγματοποιείται στην μέθοδο generateCandidates() της κλάσης CandidateGenerator. Στην συγκεκριμένη μέθοδο καλούνται δύο διαφορετικές μέθοδοι ανάλογα με το μήκος των συχνών ακολουθιών που δέχεται ως παράμετρο. Αν το μήκος των ακολουθιών είναι 2 τότε η παραγωγή των υποψηφίων είναι απλή και καλείται η μέθοδος `generate2SizeCandidates()` για τον σκοπό αυτό, η υλοποίηση της οποίας φαίνεται στο σχήμα 17. Αν το μήκος των ακολουθιών είναι μεγαλύτερο από 2 τότε εφαρμόζεται η τεχνική συγχώνευσης ακολουθιών η οποία διατυπώθηκε στον κεφάλαιο 3.1.3.

Σχήμα 16. Μέθοδος generateCandidates – Παραγωγή υποψηφίων.

```
public ArrayList<Candidate> generateCandidates(ArrayList<Candidate>
frequentSequences,int k){
     ArrayList<Candidate> prunedCandidates = new ArrayList<Candidate>();
     ArrayList<Candidate> candidates = new ArrayList<Candidate>();
     if(k == 2){
          for(int i=0;i < frequentSequences.size(); i++){
               for(int j = 0 ; j < frequentSequences.size() ; j++){
                    ArrayList<Candidate> tempCandidates = generate2SizeCandidates(
                         frequentSequences.get(i),frequentSequences.get(j),k);
                    for(Candidate candidate: tempCandidates){
                         if(!prunedCandidates.contains(candidate)){
                              prunedCandidates.add(candidate);
                         }
                    }
                }
            }
     }
     else{
          System.gc();
          for(int i=0;i < frequentSequences.size(); i++){
               for(int j = i+1 ; j < frequentSequences.size() ; j++){
                    //e.g checks for (1)(2)(3) and (1,2)(2)
                    Candidate pattern1 = frequentSequences.get(i).cutPreffix();
                    Candidate pattern2 = frequentSequences.get(j).cutSuffix();
                    //Check if mergable
                    if(pattern1.equals(pattern2)){
                         //merge into new
                         Candidate candidate = new Candidate();
                         candidate.setLength(k);
                         candidate.mergeSubsequences(frequentSequences.get(i),
                              frequentSequences.get(j));
                         candidates.add(candidate);
                     }
                     //e.g checks for (1,2)(2) and (1)(2)(3) reverse than above
                     pattern2 = frequentSequences.get(i).cutSuffix();
                     pattern1 = frequentSequences.get(j).cutPreffix();
                     //Check if mergable
                     if(pattern1.equals(pattern2)){
                          //merge into new
                          Candidate candidate = new Candidate();
                          candidate.setLength(k);
                          candidate.mergeSubsequences(frequentSequences.get(j),
                               frequentSequences.get(i));
                          candidates.add(candidate);
                      }
               }
          }
          prunedCandidates = pruneCandidates(candidates);
     }
     return prunedCandidates;
}
```

Για την καλύτερη απόδοση του αλγόριθμου, χρησιμοποιείται η μέθοδος `pruneCandidates()` στην κλάση CandidateGenerator. Η συγκεκριμένη μέθοδος ελέγχει την ύπαρξη της υποψήφιας ακολουθίας στην ιστορία των πινάκων(με την βοήθεια της μεθόδου `existsInHistory()` της κλάσης Candidate) και αν δεν εμφανίζεται τότε δεν χρειάζεται να συνυπολογιστεί στις υποψήφιες ακολουθίες.

Σχήμα 17. Μέθοδοι pruneCandidate και generate2SizeCandidates.

```
public ArrayList<Candidate> pruneCandidates(ArrayList<Candidate> candidates){
        ArrayList<Candidate> pruned = new ArrayList<Candidate>();
        for(Candidate candidate:candidates){
                if(candidate.existsInHistory(tablesHistory)){
                        Candidate cand = new Candidate();
                        cand.setMetrics(candidate.getMetrics());
                        cand.setLength(candidate.getLength());
                        cand.setSubsequences(candidate.getSubsequences());
                        cand.setSupport(candidate.getSupport());
                        pruned.add(cand);
                }
        }
        return pruned;
}


public ArrayList<Candidate> generate2SizeCandidates(Candidate pattern1,Candidate
pattern2,int k){
        //Base case: each pattern has 1 subsequence and 1 event
        AtomicChangeEvent event1 = pattern1.getSubsequences().get(0).
             getAtomicChangeEvents().get(0);
        AtomicChangeEvent event2 = pattern2.getSubsequences().get(0).
             getAtomicChangeEvents().get(0);
        ArrayList<Candidate> ret = new ArrayList<Candidate>();
        Candidate candidate = new Candidate();
        candidate.setLength(k);
        //Add {x,x}
        Subsequence subSequence = new Subsequence();
        subSequence.addAtomicChangeEvent(event1);
        subSequence.addAtomicChangeEvent(event2);
        candidate.addSubsequence(subSequence);
        ret.add(candidate);
        candidate = new Candidate();
        candidate.setLength(k);
        //Add {x}{x}
        subSequence = new Subsequence();
        subSequence.addAtomicChangeEvent(event1);
        candidate.addSubsequence(subSequence);
        subSequence = new Subsequence();
        subSequence.addAtomicChangeEvent(event2);
        candidate.addSubsequence(subSequence);
        ret.add(candidate);
        return ret;
}
```

Αρκετό ενδιαφέρον παρουσιάζουν οι κλάσεις CDISTCounter και COBJCounter οι οποίες είναι υπεύθυνες για τους δύο διαφορετικούς τρόπους μέτρησης.

Σχήμα 18. Μέθοδος countSupport της κλάσης COBJCounter

```
public void countSupport(ArrayList<Candidate> candidates){
     frequentCandidates = candidates;
     for(int l=0; l < candidates.size(); l++){
          candidates.get(l).getMetrics().setNumOccurences(0);
     }
     for(int l=0; l < candidates.size(); l++){
          ArrayList<Subsequence> subsequences1 =
          candidates.get(l).getSubsequences();
          for(int i = 0; i < subsequences1.size(); i++){
              subsequences1.get(i).getTableSequence().clear();
          }
          for(TableHistorySequence tableHistory:sequences){
               ArrayList<TableHistoryElements> elements =
                                  tableHistory.getTableHistoryElements();
               ArrayList<Subsequence> subsequences =
                                  candidates.get(l).getSubsequences();
              int currentPosition = 0;
              int itemsFoundInHistory = 0;
              TableHistoryElements selected[] = new
               TableHistoryElements[subsequences.size()];
              for(int i = 0; i < subsequences.size(); i++){
                   for(int j = currentPosition; j <elements.size(); j++){
                        if(subsequences.get(i).isInHistoryElement(
                                    elements.get(j))){
                             currentPosition = j+1;
                             itemsFoundInHistory++;
                            //keep here elements.get(j) and add them in if down
                             selected[i] = elements.get(j);
                             break;
                        }
                   }
               }
               if(itemsFoundInHistory == subsequences.size()){
                    for(int i = 0; i < subsequences.size(); i++){
                         subsequences.get(i).addTransitionInfo(
                              selected[i].getTransitionInfo());
                         subsequences.get(i).addTableInfo(selected[i]);
                    }
                    Candidate c = candidates.get(l);
                    c.setSubsequences(subsequences);
                    Metrics metrics = new Metrics();
                    metrics.setNumOccurences(candidates.get(l).
                         getMetrics().getNumOccurences() + 1);
                    c.setMetrics(metrics);
                    candidates.set(l, c);
                    this.frequentCandidates.set(l,c);
              }
          }
     }
}
```

Σχήμα 19. Μέθοδος countSupport της κλάσης CDISTCounter

```
public void countSupport(ArrayList<Candidate> candidates) {
     frequentCandidates = candidates;
     for(int l=0; l < candidates.size(); l++){
          ArrayList<Subsequence> subsequences1 = candidates.get(l).getSubsequences();
          for(int i = 0; i < subsequences1.size(); i++){
               subsequences1.get(i).getTableSequence().clear();
          }
          for(TableHistorySequence tableHistory:sequences){
               HashMap<Integer, ArrayList<AtomicChangeEvent>> eventTimestampFlag =
                      new HashMap<Integer,ArrayList<AtomicChangeEvent>>();
               HashMap<Integer, ArrayList<Boolean>> atomicChangeUsedinTransition =
                      new HashMap<Integer,ArrayList<Boolean>>();
               ArrayList<TableHistoryElements> elements =
                      tableHistory.getTableHistoryElements();
               ArrayList<Subsequence> subsequences =
                      candidates.get(l).getSubsequences();
               TableHistoryElements selected[] = new
                         TableHistoryElements[subsequences.size()];
               //Initialize data structures

               for(int j = 0; j < elements.size(); j++){
                    ArrayList<Boolean> tmpBool = new ArrayList<Boolean>();
                    ArrayList<AtomicChangeEvent> events = new
                              ArrayList<AtomicChangeEvent>();
                    for(AtomicChangeEvent event:
elements.get(j).getAtomicChangeEvents()){
                         tmpBool.add(false);
                         events.add(event);
                    }
                    atomicChangeUsedinTransition.put(elements.get(j).
                         getTransitionInfo().getId(), tmpBool);
                    eventTimestampFlag.put(elements.get(j).
                         getTransitionInfo().getId(),events);
               }
               int startingPos = 0;
               boolean isFinished = false;
               int firstPosition = 0;
               while(!isFinished){
                    int currentPosition = firstPosition +1;
                    int itemsFoundInHistory = 0;
                    for(int i = startingPos; i < subsequences.size(); i++){
                         int j;
                         boolean firstFound = false;
                         for(j = currentPosition; j < elements.size(); j++){
                              ArrayList<Boolean> boolEvents =
                                   atomicChangeUsedinTransition.get(
                                      elements.get(j).getTransitionInfo().getId());
                              if(subsequences.get(i).isInHistoryElement(
                                   elements.get(j))){
                                   if(!isSubsequenceUsed(boolEvents,
                                        subsequences.get(i))){
                                        if(!firstFound){
                                             firstPosition = j;
                                             firstFound = true;
                                        }

                                        currentPosition = j+1;
                                        itemsFoundInHistory++;
                    //keep here elements.get(j) and add them in if down

                                        selected[i] = elements.get(j);
```

```
                                        atomicChangeUsedinTransition.put(
                                              elements.get(j).
                                         getTransitionInfo().getId(), boolEvents);

                                         break;
                                    }
                               }
                          }
                          if(j >= elements.size()){
                               isFinished = true;
                          }
                     }
                     if(itemsFoundInHistory == subsequences.size()){
                          for(int i = 0; i < subsequences.size(); i++){
                               subsequences.get(i).addTransitionInfo(
                                    selected[i].getTransitionInfo());
                               subsequences.get(i).addTableInfo(selected[i]);
                          }
                          Candidate c = candidates.get(l);
                          c.setSubsequences(subsequences);
                          Metrics metrics = new Metrics();
                          metrics.setNumOccurences(candidates.get(l).getMetrics().
                               getNumOccurences()+1);

                          c.setMetrics(metrics);
                          candidates.set(l, c);
                          this.frequentCandidates.set(l,c);
                     }
                }
           }
      }
}
```

Σχήμα 20. Παράδειγμα αρχείου εξόδου

Pattern	Support	Occurences
<{AttrTypeUpd},{AttrTypeUpd}>	61%	54
<{AttrAdd@ExistTable},{AttrTypeUpd}>	60%	53
<{AttrAdd@ExistTable},{AttrDel@ExistTable}>	60%	53
<{AttrTypeUpd},{AttrAdd@ExistTable}>	59%	52
<{AttrTypeUpd},{AttrDel@ExistTable}>	58%	51
<{AttrDel@ExistTable},{AttrTypeUpd}>	58%	51
<{AttrTypeUpd},{AttrAdd@ExistTable},{AttrDel@ExistTable}>	57%	50
<{AttrTypeUpd},{AttrDel@ExistTable},{AttrTypeUpd}>	55%	48
<{AttrAdd@TableCreation,AttrAdd@TableCreation}>	36%	32
<{AttrTypeUpd,AttrTypeUpd}>	36%	32
<{AttrTypeUpd},{AttrTypeUpd},{AttrDel@ExistTable}>	36%	32
<{AttrTypeUpd},{AttrTypeUpd},{AttrDel@ExistTable},{AttrTypeUpd}>	33%	29
<{AttrAdd@ExistTable},{AttrAdd@ExistTable}>	31%	27
<{AttrDel@ExistTable},{AttrAdd@ExistTable}>	27%	24
<{AttrAdd@ExistTable},{AttrDel@ExistTable},{AttrAdd@ExistTable}>	24%	21

Τέλος, το εργαλείο παράγει ως έξοδο ένα αρχείο CSV το οποίο περιέχει τις συχνές ακολουθίες για την επιλεγμένη συλλογή δεδομένων. Πιο συγκεκριμένα, κάθε γραμμή του αρχείου περιέχει τρεις στήλες από τις οποίες η πρώτη περιέχει την συχνή ακολουθία (Pattern), η δεύτερη περιέχει την υποστήριξη της συγκεκριμένης ακολουθίας (Support) και η τρίτη τον αριθμό των πινάκων στους οποίους εμφανίζεται η ακολουθία (Occurences). Ένα παράδειγμα αρχείου εξόδου φαίνεται στο σχήμα 20.

4.3 Μεθοδολογία ελέγχου του λογισμικού

Για τον έλεγχο του εργαλείου χρησιμοποιήθηκε η μέθοδος μαύρου κουτιού στην οποία κατασκευάζονται τα δεδομένα εισόδου (μελέτες περιπτώσεων) τα οποία δίνονται σε ένα μαύρο κουτί (το εργαλείο της διπλωματικής στη συγκεκριμένη περίπτωση) και παράγεται μία έξοδος. Αν η έξοδος είναι ίδια με την αναμενόμενη έξοδο τότε η λειτουργία του μαύρου κουτιού είναι σωστή, ενώ σε αντίθετη περίπτωση είναι λάθος.

Σχήμα 21. Παράδειγμα αρχείου εισόδου.

	1	**2**	**3**
A	AttrAdd@ExistTable AttrDel@ExistTable PrimaryKeyUpd	AttrDel@ExistTable AttrTypeUpd	AttrDel@TableDel
B	AttrAdd@ExistTable AttrDel@ExistTable	AttrDel@ExistTable AttrTypeUpd PrimaryKeyUpd	-
C	AttrAdd@ExistTable AttrDel@ExistTable	AttrDel@ExistTable AttrTypeUpd PrimaryKeyUpd	AttrDel@ExistTable PrimaryKeyUpd AttrDel@TableDel
D	AttrDel@ExistTable	AttrTypeUpd PrimaryKeyUpd	PrimaryKeyUpd AttrDel@TableDel
E	AttrAdd@ExistTable AttrTypeUpd	AttrDel@ExistTable PrimaryKeyUpd AttrDel@TableDel	-

Στο σχήμα 21 φαίνεται ένα παράδειγμα αρχείου εισόδου το οποίο δόθηκε ως είσοδος στο εργαλείο στην φάση του ελέγχου. Το αρχείο περιέχει 5 πίνακες και 3 μεταβάσεις στις οποίες λαμβάνουν χώρα διάφορα γεγονότα.

Επιπλέον δημιουργήθηκε ένα αρχείο με την αναμενόμενη έξοδο, το οποίο περιέχει όλες τις συχνές ακολουθίες και φαίνεται στο σχήμα 22. Συγκρίνοντας τα αποτελέσματα εξόδου του αλγορίθμου, με είσοδο τα δεδομένα του σχήματος 21 και κατώφλι υποστήριξης 0.5, με τα αναμενόμενα αποτελέσματα του σχήματος 21 παρατηρείται ότι είναι ακριβώς ίδια. Συνεπώς, το εργαλείο παράγει την αναμενόμενη έξοδο.

Σχήμα 22. Αναμενόμενο αποτέλεσμα με την είσοδο του σχήματος 21

Pattern	Support	Occurences
<{AttrAdd@ExistTable},{AttrDel@ExistTable}>	0,8	4
<{AttrTypeUpd},{AttrDel@TableDel}>	0,8	4
<{AttrDel@ExistTable},{AttrTypeUpd}>	0,8	4
<{AttrDel@ExistTable,PrimaryKeyUpd}>	0,8	4
<{AttrAdd@ExistTable},{AttrTypeUpd}>	0,6	3
<{AttrAdd@ExistTable,AttrDel@ExistTable}>	0,6	3
<{AttrAdd@ExistTable},{AttrDel@TableDel}>	0,6	3
<{AttrAdd@ExistTable},{PrimaryKeyUpd}>	0,6	3
<{AttrDel@ExistTable,AttrTypeUpd}>	0,6	3
<{AttrTypeUpd,PrimaryKeyUpd}>	0,6	3
<{AttrTypeUpd},{PrimaryKeyUpd}>	0,6	3
<{AttrDel@ExistTable},{AttrDel@ExistTable}>	0,6	3
<{AttrDel@ExistTable},{AttrDel@TableDel}>	0,6	3
<{AttrDel@ExistTable},{PrimaryKeyUpd}>	0,6	3
<{AttrDel@TableDel,PrimaryKeyUpd}>	0,6	3
<{PrimaryKeyUpd},{AttrDel@TableDel}>	0,6	3
<{AttrAdd@ExistTable,AttrDel@ExistTable},{AttrTypeUpd}>	0,6	3
<{AttrAdd@ExistTable,AttrDel@ExistTable},{AttrDel@ExistTable}>	0,6	3
<{AttrAdd@ExistTable},{AttrDel@ExistTable,AttrTypeUpd}>	0,6	3
<{AttrAdd@ExistTable},{AttrDel@ExistTable,PrimaryKeyUpd}>	0,6	3
<{AttrDel@ExistTable},{AttrDel@ExistTable,AttrTypeUpd}>	0,6	3
<{AttrDel@ExistTable},{AttrTypeUpd},{AttrDel@TableDel}>	0,6	3
<{AttrTypeUpd},{AttrDel@TableDel,PrimaryKeyUpd}>	0,6	3
<{AttrDel@ExistTable},{AttrTypeUpd,PrimaryKeyUpd}>	0,6	3
<{AttrAdd@ExistTable,AttrDel@ExistTable},{AttrDel@ExistTable,AttrTypeUpd}>	0,6	3

Κεφάλαιο 5. Πειραματική Αξιολόγηση

5.1 Μεθοδολογία πειραματισμού

Για τα πειράματα χρησιμοποιήθηκαν δύο διαφορετικές τιμές κατωφλίου υποστήριξης (minsup) για τους δύο διαφορετικούς τρόπους μέτρησης υποστήριξης και για κάθε συλλογή δεδομένων. Πιο συγκεκριμένα χρησιμοποιήθηκαν οι τιμές κατωφλίου: 0.05, 0.1 για τον τρόπο μέτρησης της υποστήριξης COBJ και οι τιμές κατωφλίου 0.05 και 0.01 για την μέτρηση CDIST για τις διαφορετικές συλλογές δεδομένων. Για τα πειράματα έχουν συγκεντρωθεί 8 συλλογές δεδομένων οι οποίες αναφέρονται σε ανοιχτά συστήματα λογισμικού τα οποία προέρχονται από ένα ευρύ φάσμα εφαρμογών όπως Συστήματα Διαχείρισης Περιεχομένου (CMS's), συστήματα διαχείρισης εικόνων, online καταστήματα, καθώς και επιστημονικές αποθήκες δεδομένων.

Atlas Triger: είναι ένας από τους επτά ανιχνευτές σωματιδίων ο οποίος κατασκευάστηκε στο LHC (Large Hadron Collider), στον επιταχυντή σωματιδίων στο CERN. Στόχος του Atlas είναι η εύρεση γνώσης σχετικά με τις βασικές δυνάμεις οι οποίες σχημάτισαν το σύμπαν από την αρχή του και πως αυτές θα καθορίσουν το μέλλον του.

Biosql: είναι ένα γενικό σχεσιακό μοντέλο το οποίο καλύπτει αλληλουχίες, χαρακτηριστικά, μία ταξινόμηση αναφοράς και οντολογίες από διάφορες πηγές όπως το GenBank[2] και το Swissport[3]. Ενώ στην αρχική ενσάρκωση του (το 2011), στην οποία συνέβαλε ο Ewan Birney, ήταν ένα τοπικό σχεσιακό σύστημα για το GenBank, το έργο έχει γίνει από τότε μία συνεργασία μεταξύ των BioPerl, BioPython, BioJava και BioRuby. Στόχος είναι η κατασκευή ενός αρκετά γενικού

[2] http://www.ncbi.nlm.nih.gov/genbank/

[3] http://www.uniprot.org/

σχήματος για την αποθήκευση των ακολουθιών, των χαρακτηριστικών και τον σχολιασμό τους με ένα λειτουργικό τρόπο μεταξύ των Βιο* έργων.

Coppermine: είναι ένα λογισμικό για διαχείριση εικόνων με δυνατότητες πολυμέσων. Οι δυνατότητες που παρέχει είναι η οργάνωση των φωτογραφιών σε κατηγορίες, υποστήριξη πολυμέσων, επιλογή για ιδιωτικά άλμπουμ φωτογραφιών και πολυγλωσσικό περιβάλλον. Η εγκατάσταση απαιτεί PHP, MySQL και το ImageMagick ή την βιβλιοθήκη GD Graphics, και δουλεύει με τους περισσότερους διαδικτυακούς διακομιστές (web servers) όπως είναι ο Apache. Το Coppermine είναι ένα δωρεάν λογισμικό το οποίο παρέχεται με την άδεια GNU GPL.

Ensembl: είναι ένα επιστημονικό έργο του Ευρωπαϊκού Ινστιτούτου Βιοπληροφορικής (European Bioinformatics Institute) και του Wellcome Trust Sanger Institute το οποίο ξεκίνησε το 1999 για την ολοκλήρωση του προγράμματος του ανθρώπινου γονιδιώματος. Ο στόχος του Ensembl ήταν ο αυτόματος σχολιασμός του γονιδιώματος, η ενσωμάτωση του σχολιασμού με άλλα βιολογικά δεδομένα και η δημοσιοποίησή τους στον παγκόσμιο ιστό.

MediaWiki: παρουσιάστηκε στις αρχές του 2002 από την Wikipedia Foundation μαζί με την Wikipedia, και φιλοξενεί το περιεχόμενο της Wikipedia από τότε. Το λογισμικό είναι βελτιστοποιημένο να χειρίζεται μεγάλα έργα τα οποία έχουν αρκετά terabyte περιεχομένου.

OpenCart: είναι ένα ανοιχτού κώδικα λογισμικό για διαδικτυακά καταστήματα. Μπορεί να χρησιμοποιηθεί σε οποιονδήποτε διακομιστή που τρέχει PHP και MySQL και χρησιμοποιείται από πάρα πολλά διαδικτυακά καταστήματα.

Phpbb: είναι ένα πακέτο διαχείρισης φόρουμ το οποίο είναι γραμμένο σε PHP. Είναι διαθέσιμο δωρεάν κάτω από την άδεια GNU General Public License και υποστηρίζει αρκετά συστήματα διαχείρισης βάσεων δεδομένων όπως είναι οι PostgreSQL, SQLite, MySQL, Oracle Database και Microsoft SQL Server.

Typo3: είναι ένα ανοικτού κώδικα σύστημα διαχείρισης περιεχομένου που βασίζεται σε PHP. Διατίθεται κάτω από την άδεια GNU General Public License και μπορεί να τρέξει σε μία πληθώρα από web servers όπως είναι ο Apache καθώς επίσης και σε πολλά λειτουργικά συστήματα όπως είναι τα Linux, Windows, Mac OS X κ.α.

Απόδοση: Αρχικά, χρονομετρήθηκαν οι χρόνοι εκτέλεσης του αλγορίθμου για κάθε διαφορετική τιμή του κατωφλίου υποστήριξης και του είδους μέτρησης. Οι χρόνοι εκτέλεσης φαίνονται στα σχήματα 23 και 24 και προέκυψαν από τον μέσο όρο τεσσάρων εκτελέσεων για κάθε κατώφλι υποστήριξης για ένα συγκεκριμένο είδος

μέτρησης και σύνολο δεδομένων. Στο συγκεκριμένο πείραμα ελέγχονται δύο πράγματα: (α) τα διαφορετικά είδη υποστήριξης για την ίδια συλλογή δεδομένων και (β) διαφορετικές συλλογές δεδομένων μεταξύ τους. Όσον αφορά τα διαφορετικά είδη υποστήριξης για την ίδια συλλογή δεδομένων φαίνεται ξεκάθαρα ότι η επιλογή του κατωφλίου υποστήριξης επηρεάζει άμεσα την απόδοση του αλγορίθμου.

Σχήμα 23. Χρόνοι εκτέλεσης σε sec για κάθε κατώφλι υποστήριξης και συλλογή δεδομένων με μέτρηση CDIST

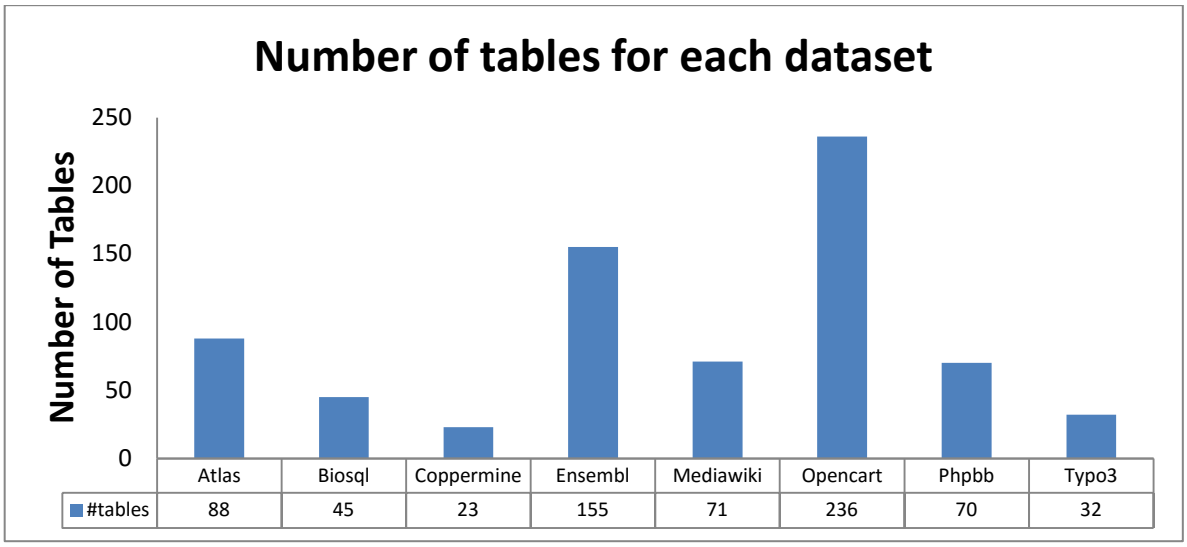
Support Threshold	0,01	0,05
Atlas	0,78	0,40
Biosql	3,43	0,43
Coppermine	0,66	0,09
Ensembl	43,88	1,31
Mediawiki	1,50	0,15
Opencart	0,58	0,25
Phpbb	114,96	2,88
Typo3	3,62	0,13

Για την εύρεση του παράγοντα ο οποίος επηρεάζει την απόδοση του αλγορίθμου δημιουργήθηκαν τα διαγράμματα των σχημάτων 24 και 25 όπου φαίνεται το πλήθος των πινάκων και το πλήθος των μεταβάσεων για κάθε συλλογή δεδομένων. Αν συγκρίνει κανείς τα σχήματα 25 και 26 με τα σχήματα 23 και 24 εύκολα παρατηρεί ότι ο παράγοντας ο οποίος επηρεάζει την αποδοτικότητα του αλγορίθμου, στις διαφορετικές συλλογές δεδομένων, είναι το πλήθος των πινάκων και ο αριθμός των μεταβάσεων που υπάρχουν στην συγκεκριμένη συλλογή δεδομένων.

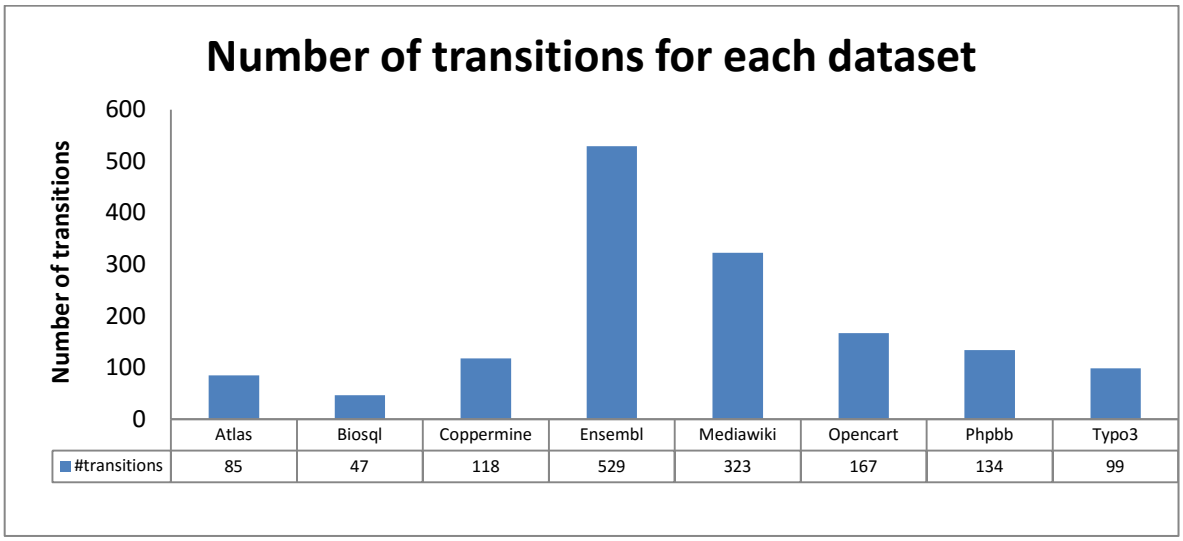
Σχήμα 24. Χρόνοι εκτέλεσης για κάθε κατώφλι υποστήριξης και συλλογή δεδομένων με μέτρηση COBJ

Support Threshold	0,05	0,1
Atlas	0,48	0,29
Biosql	1,05	0,36
Coppermine	0,22	0,20
Ensembl	113,44	1,11
Mediawiki	0,63	0,30
Opencart	0,36	0,26
Phpbb	96,45	0,38
Typo3	0,56	0,18

Σχήμα 25. Αριθμός πινάκων για κάθε συλλογή δεδομένων.



Σχήμα 26. Πλήθος μεταβάσεων για κάθε συλλογή δεδομένων



Επιπλέον, συγκρίνοντας τις ίδιες συλλογές δεδομένων για τους δύο διαφορετικούς τρόπους μέτρησης της υποστήριξης με ελάχιστο κατώφλι υποστήριξης 0.05 παρατηρείται ότι οι συχνές ακολουθίες της μέτρησης CDIST είναι καθαρό υποσύνολο της μέτρησης COBJ. Έτσι, από εδώ και στο εξής θα μελετηθούν μόνο τα αποτελέσματα της μέτρησης COBJ.

Σχήμα 27. Συνολικός αριθμός συχνών ακολουθιών για όλες τις συλλογές δεδομένων με την μέτρηση CDIST

Support Threshold	0,01	0,05	0,1
Atlas	521	40	12
Biosql	5223	214	42
Coppermine	963	13	4
Ensembl	49914	126	16
Mediawiki	1104	39	9
Opencart	580	39	12
Phpbb	43864	183	15
Typo3	1146	30	7

Ο σκοπός του συγκεκριμένου πειράματος είναι η εύρεση μιας αντιπροσωπευτικής τιμής κατωφλίου για όλες τις συλλογές δεδομένων η οποία θα χρησιμοποιηθεί για την μελέτη των αποτελεσμάτων του αλγόριθμου εξόρυξης προτύπων. Αυτό που παρατηρείται είναι ότι μικρές αλλαγές στην τιμή του κατωφλίου υποστήριξης επηρεάζουν σε μεγάλο βαθμό τον αριθμό των συχνών ακολουθιών που παράγονται.

Σχήμα 28. Συνολικός αριθμός συχνών ακολουθιών για όλες τις συλλογές δεδομένων με την μέτρηση COBJ

Support Threshold	0,05	0,1
Atlas	483	135
Biosql	1725	407
Coppermine	52	26
Ensembl	43175	1467
Mediawiki	1067	164
Opencart	209	74
Phpbb	37130	327
Typo3	1034	67

5.2 Αναλυτική παρουσίαση αποτελεσμάτων

Ύστερα από την εκτέλεση του αλγορίθμου με ελάχιστο κατώφλι υποστήριξης 0.05 συγκεντρώθηκαν οι 20 πιο συχνές ακολουθίες από κάθε συλλογή δεδομένων. Για την καλύτερη κατανόηση των παρακάτω πινάκων ορίζονται οι εξής συντομογραφίες:

- **AttrAdd@ExistTable:** Εισαγωγή πεδίου σε υπάρχοντα πίνακα
- **AttrAdd@TableCreation**: Εισαγωγή πεδίου σε πίνακα που δημιουργήθηκε την ίδια χρονική στιγμή με την εισαγωγή του πεδίου

- **AttrDel@ExistTable:** Διαγραφή πεδίου από πίνακα που παραμένει ζωντανός
- **AttrDel@TableDeletion:** Διαγραφή πεδίου από πίνακα που πεθαίνει την ίδια χρονική στιγμή που διαγράφεται το πεδίο.
- **AttrTypeUpd:** Αλλαγή στο τύπο κάποιου πεδίου
- **PrimaryKeyUpd:** Αλλαγή στο πρωτεύον κλειδί ενός πίνακα

5.2.1 Atlas

Η συγκεκριμένη συλλογή δεδομένων χαρακτηρίζεται από μαζικές αλλαγές στους πίνακες της βάσης στην ίδια μετάβαση. Πιο συγκεκριμένα, στην μετάβαση 19 όλοι οι ζωντανοί πίνακες της βάσης δέχονται μία ή περισσότερες αλλαγές στον τύπο των πεδίων τους και στην μετάβαση 32 δέχονται προσθήκη ενός νέου πεδίου. Επιπλέον, μια μετάβαση αργότερα (μετάβαση 33), σχεδόν σε όλους τις πίνακες της βάσης διαγράφονται ένα ή περισσότερα πεδία και τέλος, στην μετάβαση 49, υπάρχει ακόμα μία μαζική αλλαγή στο τύπο όλων των ζωντανών πινάκων. Αυτές οι 4 μαζικές αλλαγές στο μεγαλύτερο ποσοστό των πινάκων δικαιολογούν τις κορυφαίες 20 συχνές ακολουθίες του σχήματος 29.

Σχήμα 29. Atlas: 20 πιο συχνές ακολουθίες

Pattern	Support	Occurences
<{AttrTypeUpd},{AttrTypeUpd}>	0,6136	54
<{AttrAdd@ExistTable},{AttrTypeUpd}>	0,6023	53
<{AttrAdd@ExistTable},{AttrDel@ExistTable}>	0,6023	53
<{AttrTypeUpd},{AttrAdd@ExistTable}>	0,5909	52
<{AttrTypeUpd},{AttrDel@ExistTable}>	0,5795	51
<{AttrDel@ExistTable},{AttrTypeUpd}>	0,5795	51
<{AttrTypeUpd},{AttrAdd@ExistTable},{AttrTypeUpd}>	0,5682	50
<{AttrTypeUpd},{AttrAdd@ExistTable},{AttrDel@ExistTable}>	0,5682	50
<{AttrAdd@ExistTable},{AttrDel@ExistTable},{AttrTypeUpd}>	0,5682	50
<{AttrTypeUpd},{AttrDel@ExistTable},{AttrTypeUpd}>	0,5455	48
<{AttrTypeUpd},{AttrAdd@ExistTable},{AttrDel@ExistTable},{AttrTypeUpd}>	0,5341	47
<{AttrTypeUpd},{AttrTypeUpd},{AttrAdd@ExistTable}>	0,375	33
<{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,3636	32
<{AttrTypeUpd,AttrTypeUpd}>	0,3636	32
<{AttrTypeUpd},{AttrTypeUpd},{AttrDel@ExistTable}>	0,3636	32
<{AttrTypeUpd},{AttrTypeUpd},{AttrAdd@ExistTable},{AttrTypeUpd}>	0,3523	31
<{AttrTypeUpd},{AttrTypeUpd},{AttrAdd@ExistTable},{AttrDel@ExistTable}>	0,3409	30
<{AttrTypeUpd},{AttrTypeUpd},{AttrDel@ExistTable},{AttrTypeUpd}>	0,3295	29
<{AttrAdd@ExistTable},{AttrAdd@ExistTable}>	0,3068	27

5.2.2 Biosql

Η συγκεκριμένη συλλογή χαρακτηρίζεται από πολλές προσθαφαιρέσεις πεδίων, κυρίως την ίδια χρονική στιγμή. Αυτό συμβαίνει διότι στην μετάβαση 21 έχουμε πολλές εισαγωγές και διαγραφές πεδίων σε ένα πολύ μεγάλο ποσοστό των πινάκων. Κάποιες από αυτές τις προσθαφαιρέσεις οφείλονται σε μετονομασίες πεδίων, δηλαδή πεδία που διαγράφονται επιστρέφουν στον πίνακα, την ίδια χρονική στιγμή με την μέθοδο της εισαγωγής νέων πεδίων τα οποία έχουν τον ίδιο τύπο με τα διαγραμμένα και μια μικρή παραλλαγή στο όνομα τους

Σχήμα 30. Biosql: 20 πιο συχνές ακολουθίες

Pattern	Support	Occurences
<{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,5333	24
<{AttrAdd@ExistTable,AttrDel@ExistTable}>	0,4	18
<{AttrDel@TableDel,AttrDel@TableDel}>	0,3778	17
<{AttrAdd@ExistTable},{AttrAdd@ExistTable}>	0,3556	16
<{AttrAdd@ExistTable,AttrAdd@ExistTable}>	0,3333	15
<{AttrAdd@ExistTable},{AttrDel@ExistTable}>	0,3333	15
<{AttrDel@ExistTable,AttrDel@ExistTable}>	0,3333	15
<{AttrAdd@ExistTable,AttrAdd@ExistTable,AttrDel@ExistTable}>	0,3333	15
<{AttrAdd@ExistTable},{AttrAdd@ExistTable,AttrDel@ExistTable}>	0,3333	15
<{AttrAdd@ExistTable,AttrDel@ExistTable,AttrDel@ExistTable}>	0,3333	15
<{AttrAdd@TableCreation},{AttrAdd@ExistTable}>	0,3111	14
<{AttrAdd@ExistTable},{AttrAdd@ExistTable,AttrAdd@ExistTable}>	0,3111	14
<{AttrAdd@ExistTable},{AttrAdd@ExistTable},{AttrDel@ExistTable}>	0,3111	14
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrAdd@ExistTable}>	0,3111	14
<{AttrAdd@ExistTable},{AttrAdd@ExistTable,AttrAdd@ExistTable,AttrDel@ExistTable}>	0,3111	14
<{AttrAdd@ExistTable,AttrAdd@ExistTable,AttrDel@ExistTable,AttrDel@ExistTable}>	0,3111	14
<{AttrDel@ExistTable},{AttrAdd@ExistTable}>	0,2889	13
<{AttrDel@ExistTable},{AttrDel@ExistTable}>	0,2889	13
<{AttrAdd@ExistTable,AttrDel@ExistTable},{AttrAdd@ExistTable}>	0,2889	13

5.2.3 Coppermine

Η συγκεκριμένη συλλογή δεδομένων δεν έχει πολλές αλλαγές στους πίνακες με αποτέλεσμα το πλήθος των συχνών ακολουθιών να είναι σχετικά μικρό. Αυτό συμβαίνει διότι η υποστήριξη των υποψήφιων ακολουθιών δεν ξεπερνά το ελάχιστο κατώφλι υποστήριξης.

Σχήμα 31. Coppermine: 20 πιο συχνές ακολουθίες

Pattern	Support	Occurences
<{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,6522	15
<{AttrAdd@ExistTable},{AttrAdd@ExistTable}>	0,3043	7
<{AttrAdd@ExistTable,AttrAdd@ExistTable}>	0,2609	6
<{AttrAdd@TableCreation},{AttrAdd@ExistTable}>	0,2609	6
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrAdd@ExistTable}>	0,2609	6
<{AttrAdd@ExistTable},{AttrDel@ExistTable}>	0,2174	5
<{AttrDel@ExistTable},{AttrAdd@ExistTable}>	0,1739	4
<{AttrDel@ExistTable,AttrDel@ExistTable}>	0,1739	4
<{AttrAdd@ExistTable,AttrAdd@ExistTable},{AttrAdd@ExistTable}>	0,1739	4
<{AttrAdd@ExistTable},{AttrAdd@ExistTable},{AttrDel@ExistTable}>	0,1739	4
<{AttrAdd@ExistTable},{AttrDel@ExistTable},{AttrAdd@ExistTable}>	0,1739	4
<{AttrAdd@ExistTable},{AttrDel@ExistTable,AttrDel@ExistTable}>	0,1739	4
<{AttrAdd@ExistTable},{AttrTypeUpd}>	0,1304	3
<{AttrAdd@TableCreation},{AttrTypeUpd}>	0,1304	3
<{AttrDel@ExistTable},{AttrDel@ExistTable}>	0,1304	3
<{AttrAdd@ExistTable},{AttrAdd@ExistTable,AttrAdd@ExistTable}>	0,1304	3
<{AttrAdd@ExistTable},{AttrDel@ExistTable},{AttrDel@ExistTable}>	0,1304	3
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrTypeUpd}>	0,1304	3
<{AttrDel@ExistTable,AttrDel@ExistTable},{AttrAdd@ExistTable}>	0,1304	3

5.2.4 Ensembl

Η συγκεκριμένη συλλογή δεδομένων χαρακτηρίζεται από αρκετές δημιουργίες πινάκων στην ιστορία της βάσης όπως επίσης και διαγραφές πινάκων. Επιπλέον, παρατηρείται ότι ένα μεγάλο ποσοστό των πινάκων που δημιουργούνται σε κάποια ενδιάμεση χρονική στιγμή στην ιστορία της βάσης δέχονται αλλαγές στα πεδία τους, είτε αυτά αφορούν εισαγωγές και διαγραφές πεδίων είτε αλλαγές στον τύπο των πεδίων.

Σχήμα 32. Ensembl: 20 πιο συχνές ακολουθίες

Pattern	Support	Occurences
<{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,9032	140
<{AttrDel@TableDel,AttrDel@TableDel}>	0,671	104
<{AttrAdd@TableCreation},{AttrDel@TableDel}>	0,6194	96
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrDel@TableDel}>	0,6194	96
<{AttrAdd@TableCreation},{AttrDel@TableDel,AttrDel@TableDel}>	0,6194	96
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrDel@TableDel,AttrDel@TableDel}>	0,6194	96
<{AttrAdd@TableCreation},{AttrTypeUpd}>	0,5161	80
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrTypeUpd}>	0,5161	80
<{AttrTypeUpd},{AttrTypeUpd}>	0,4516	70
<{AttrAdd@TableCreation},{AttrAdd@ExistTable}>	0,4	62
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrAdd@ExistTable}>	0,4	62
<{AttrAdd@TableCreation},{AttrTypeUpd},{AttrTypeUpd}>	0,3806	59
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrTypeUpd},{AttrTypeUpd}>	0,3806	59
<{AttrAdd@ExistTable},{AttrTypeUpd}>	0,3677	57
<{AttrAdd@TableCreation},{AttrAdd@ExistTable},{AttrTypeUpd}>	0,3097	48
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrAdd@ExistTable},{AttrTypeUpd}>	0,3097	48
<{AttrAdd@ExistTable},{AttrTypeUpd},{AttrTypeUpd}>	0,3032	47
<{AttrAdd@ExistTable,AttrAdd@ExistTable}>	0,2968	46
<{AttrAdd@TableCreation},{AttrDel@ExistTable}>	0,2839	44

5.2.5 MediaWiki

Στην συγκεκριμένη συλλογή δεδομένων ένα μεγάλο ποσοστό των πινάκων της βάσης δημιουργούνται σε κάποια ενδιάμεση χρονική στιγμή στην ιστορία της βάσης. Επίσης, παρατηρούνται συχνές αλλαγές στον τύπο των πεδίων των πινάκων οι οποίες κυρίως ακολουθούν εισαγωγές πεδίων.

Σχήμα 33. MediaWiki: 20 πιο συχνές ακολουθίες

Pattern	Support	Occurences
<{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,7324	52
<{AttrAdd@TableCreation},{AttrTypeUpd}>	0,5775	41
<{AttrTypeUpd},{AttrTypeUpd}>	0,5493	39
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrTypeUpd}>	0,5352	38
<{AttrAdd@TableCreation},{AttrTypeUpd},{AttrTypeUpd}>	0,4225	30
<{AttrTypeUpd,AttrTypeUpd}>	0,4085	29
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrTypeUpd},{AttrTypeUpd}>	0,3944	28
<{AttrAdd@ExistTable},{AttrTypeUpd}>	0,3239	23
<{AttrTypeUpd,AttrTypeUpd},{AttrTypeUpd}>	0,3239	23
<{AttrDel@TableDel,AttrDel@TableDel}>	0,3099	22
<{AttrAdd@TableCreation},{AttrTypeUpd,AttrTypeUpd}>	0,2817	20
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrTypeUpd,AttrTypeUpd}>	0,2817	20
<{AttrAdd@TableCreation},{AttrAdd@ExistTable}>	0,2676	19
<{AttrAdd@ExistTable},{AttrTypeUpd},{AttrTypeUpd}>	0,2676	19
<{AttrAdd@ExistTable},{AttrAdd@ExistTable}>	0,2535	18
<{AttrTypeUpd},{AttrAdd@ExistTable}>	0,2535	18
<{AttrAdd@ExistTable},{AttrTypeUpd,AttrTypeUpd}>	0,2535	18
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrAdd@ExistTable}>	0,2535	18
<{AttrTypeUpd},{AttrTypeUpd,AttrTypeUpd}>	0,2535	18

5.2.6 OpenCart

Η συγκεκριμένη συλλογή δεδομένων χαρακτηρίζεται από πολλές δημιουργίες και διαγραφές πινάκων. Πιο συγκεκριμένα, από την μετάβαση 0 μέχρι και την μετάβαση 23 οι υπάρχοντες πίνακες διαγράφονται, δημιουργούνται νέοι οι οποίοι στην πορεία διαγράφονται και κάποιοι από αυτούς ξαναδημιουργούνται ύστερα από έναν μικρό αριθμό μεταβάσεων. Για τον παραπάνω λόγο, οι κορυφαίες 20 συχνές ακολουθίες έχουν μεγάλη υποστήριξη και αναφέρονται σε δημιουργία και διαγραφή πινάκων.

Σχήμα 34. OpenCart: 20 πιο συχνές ακολουθίες

Pattern	Support	Occurences
<{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,9619	227
<{AttrDel@TableDel,AttrDel@TableDel}>	0,6992	165
<{AttrAdd@TableCreation},{AttrDel@TableDel}>	0,6695	158
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrDel@TableDel}>	0,6653	157
<{AttrAdd@TableCreation},{AttrDel@TableDel,AttrDel@TableDel}>	0,6653	157
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrDel@TableDel,AttrDel@TableDel}>	0,6653	157
<{AttrDel@TableDel},{AttrAdd@TableCreation}>	0,4025	95
<{AttrDel@TableDel},{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,4025	95
<{AttrDel@TableDel,AttrDel@TableDel},{AttrAdd@TableCreation}>	0,4025	95
<{AttrDel@TableDel,AttrDel@TableDel},{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,4025	95
<{AttrAdd@TableCreation},{AttrTypeUpd}>	0,2542	60
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrTypeUpd}>	0,2542	60
<{AttrAdd@TableCreation},{AttrAdd@TableCreation}>	0,2331	55
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrAdd@TableCreation}>	0,2331	55
<{AttrAdd@TableCreation},{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,2331	55
<{AttrAdd@TableCreation},{AttrDel@TableDel},{AttrAdd@TableCreation}>	0,2331	55
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,2331	55
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrDel@TableDel},{AttrAdd@TableCreation}>	0,2331	55
<{AttrAdd@TableCreation},{AttrDel@TableDel},{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,2331	55

5.2.7 Phpbb

Η συγκεκριμένη συλλογή δεδομένων χαρακτηρίζεται από πολλές δημιουργίες και διαγραφές πινάκων κυρίως στο δεύτερο μισό της ιστορίας της βάσης. Αντίθετα, στο πρώτο μισό της ιστορίας της βάσης υπάρχουν αρκετές αλλαγές στον τύπο των πεδίων των πινάκων, όπως φαίνεται και στο σχήμα 35.

Σχήμα 35. Phpbb: 20 πιο συχνές ακολουθίες

Pattern	Support	Occurences
<{AttrTypeUpd},{AttrTypeUpd}>	0,2429	17
<{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,1857	13
<{AttrDel@ExistTable},{AttrAdd@ExistTable}>	0,1857	13
<{AttrAdd@ExistTable},{AttrAdd@ExistTable}>	0,1714	12
<{AttrAdd@TableCreation},{AttrAdd@TableCreation}>	0,1714	12
<{AttrAdd@TableCreation},{AttrDel@TableDel}>	0,1714	12
<{AttrDel@TableDel},{AttrAdd@TableCreation}>	0,1714	12
<{AttrDel@TableDel,AttrDel@TableDel}>	0,1714	12
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrAdd@TableCreation}>	0,1714	12
<{AttrAdd@TableCreation},{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,1714	12
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrDel@TableDel}>	0,1714	12
<{AttrDel@TableDel},{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,1714	12
<{AttrAdd@TableCreation},{AttrDel@TableDel},{AttrAdd@TableCreation}>	0,1714	12
<{AttrAdd@TableCreation},{AttrDel@TableDel,AttrDel@TableDel}>	0,1714	12
<{AttrDel@TableDel,AttrDel@TableDel},{AttrAdd@TableCreation}>	0,1714	12
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,1714	12
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrDel@TableDel},{AttrAdd@TableCreation}>	0,1714	12
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrDel@TableDel,AttrDel@TableDel}>	0,1714	12
<{AttrAdd@TableCreation},{AttrDel@TableDel},{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,1714	12

5.2.8 Typo3

Η συγκεκριμένη συλλογή δεδομένων δεν χαρακτηρίζεται από πολλές αλλαγές στην ιστορία της βάσης και αυτό φαίνεται από το σχήμα 36 στο οποίο οι κορυφαίες συχνές ακολουθίες έχουν μικρή τιμή υποστήριξης. Επίσης, παρατηρείται ότι ένα ποσοστό των πινάκων δέχεται μία αλλαγή στον τύπο των πεδίων του η οποία ακολουθεί την εισαγωγή ενός πεδίου στον ίδιο πίνακα.

Σχήμα 36. Typo3: 20 πιο συχνές ακολουθίες

Pattern	Support	Occurences
<{AttrAdd@TableCreation,AttrAdd@TableCreation}>	0,6875	22
<{AttrTypeUpd},{AttrTypeUpd}>	0,3438	11
<{AttrAdd@ExistTable},{AttrTypeUpd}>	0,3125	10
<{AttrDel@TableDel,AttrDel@TableDel}>	0,2812	9
<{AttrAdd@ExistTable},{AttrTypeUpd},{AttrTypeUpd}>	0,2812	9
<{AttrAdd@ExistTable,AttrAdd@ExistTable}>	0,25	8
<{AttrAdd@ExistTable},{AttrAdd@ExistTable}>	0,25	8
<{AttrAdd@TableCreation},{AttrDel@TableDel}>	0,25	8
<{AttrTypeUpd},{AttrAdd@ExistTable}>	0,25	8
<{AttrTypeUpd,AttrTypeUpd}>	0,25	8
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrDel@TableDel}>	0,25	8
<{AttrAdd@TableCreation},{AttrDel@TableDel,AttrDel@TableDel}>	0,25	8
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrDel@TableDel,AttrDel@TableDel}>	0,25	8
<{AttrAdd@TableCreation},{AttrTypeUpd}>	0,2188	7
<{AttrAdd@ExistTable},{AttrAdd@ExistTable},{AttrTypeUpd}>	0,2188	7
<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrTypeUpd}>	0,2188	7
<{AttrAdd@ExistTable,AttrAdd@ExistTable},{AttrAdd@ExistTable}>	0,1875	6
<{AttrAdd@ExistTable},{AttrAdd@ExistTable,AttrAdd@ExistTable}>	0,1875	6
<{AttrAdd@ExistTable},{AttrTypeUpd},{AttrAdd@ExistTable}>	0,1875	6

5.2.9 Αναλυτική περιγραφή αποτελεσμάτων

Στη συνέχεια εμφανίζεται ο συγκεντρωτικός πίνακας των αποτελεσμάτων ο οποίος αποτελείται από 12 στήλες. Η στήλη Group αναφέρεται στον αριθμό της ομάδας από συχνές ακολουθίες. Κάθε ομάδα αποτελείται από συχνές ακολουθίες οι οποίες έχουν ίδια ή παρόμοια ερμηνεία. Η στήλη Patterns περιγράφει τις συχνές

ακολουθίες. Οι στήλες Atlas, Biosql, Ensembl, Opencart, Phpbb και Typo αναφέρονται στις 8 διαφορετικές συλλογές δεδομένων και περιέχουν το ποσοστό της υποστήριξης μίας συγκεκριμένης ακολουθίας σε κάθε συλλογή (αν μία συλλογή δεδομένων δεν περιέχει τη συχνή ακολουθία τότε συμπληρώνεται με παύλα «-»). Τα ποσοστά με έντονη γραμματοσειρά συμβολίζουν ότι η συγκεκριμένη ακολουθία βρίσκεται στις κορυφαίες 20 ακολουθίες για την συγκεκριμένη συλλογή δεδομένων. Τα υπόλοιπα ποσοστά συμβολίζουν το ποσοστό της υποστήριξης για μία συγκεκριμένη ακολουθία η οποία όμως δεν βρίσκεται στις κορυφαίες 20 ακολουθίες τις συγκεκριμένης συλλογής δεδομένων. Η στήλη Count περιέχει το πλήθος των συλλογών δεδομένων στις οποίες εμφανίζεται η ακολουθία. Τέλος, η στήλη ION περιγράφει το πόσο ενδιαφέρουσα είναι μία συχνή ακολουθία συμβολίζοντας με 1 τις αδιάφορες ακολουθίες και με 5 τις πολύ ενδιαφέρουσες ακολουθίες.

Σχήμα 37. Συγκεντρωτικός πίνακας κορυφαίων 20 ακολουθιών από κάθε συλλογή δεδομένων

Group	Patterns	Atlas	Biosql	Coppermine	Ensembl	Mwiki	Opencart	Phpbb	Typo	Count	ION
1	<{AttrAdd@ExistTable},{AttrAdd@ExistTable}>	**31%**	**36%**	**30%**	26%	**25%**	8%	**17%**	**25%**	**8**	5
1	<{AttrAdd@ExistTable,AttrAdd@ExistTable}>	17%	**33%**	**26%**	**30%**	20%	7%	9%	**25%**	**8**	3
1	<{AttrAdd@ExistTable},{AttrAdd@ExistTable},{AttrTypeUpd}>	18%	7%	9%	22%	20%	5%	6%	**22%**	**8**	4
1	<{AttrAdd@ExistTable},{AttrAdd@ExistTable,AttrAdd@ExistTable}>	13%	**31%**	**13%**	17%	15%	-	6%	**19%**	**7**	4
1	<{AttrAdd@ExistTable,AttrAdd@ExistTable},{AttrAdd@ExistTable}>	7%	24%	**17%**	21%	14%	-	7%	**19%**	**7**	3
1	<{AttrAdd@ExistTable},{AttrAdd@ExistTable},{AttrDel@ExistTable}>	16%	**31%**	**17%**	19%	13%	-	14%	9%	**7**	3
2	<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrTypeUpd}>	16%	9%	**13%**	**52%**	**54%**	**25%**	6%	**22%**	**8**	5
2	<{AttrAdd@TableCreation},{AttrTypeUpd}>	16%	9%	**13%**	**52%**	**58%**	**25%**	6%	**22%**	**8**	5
3	<{AttrAdd@ExistTable},{AttrTypeUpd}>	**60%**	13%	**13%**	**37%**	**32%**	11%	10%	**31%**	**8**	5
3	<{AttrAdd@ExistTable},{AttrTypeUpd},{AttrTypeUpd}>	26%	7%	9%	**30%**	**27%**	5%	9%	**28%**	**8**	4
3	<{AttrTypeUpd,AttrTypeUpd}>	**36%**	13%	-	27%	**41%**	6%	16%	**25%**	**7**	4
3	<{AttrAdd@ExistTable},{AttrTypeUpd,AttrTypeUpd}>	**30%**	7%	-	17%	**25%**	-	6%	19%	**6**	3
3	<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrAdd@ExistTable},{AttrTypeUpd}>	8%	-	-	**31%**	20%	11%	-	6%	**5**	2
3	<{AttrAdd@TableCreation},{AttrAdd@ExistTable},{AttrTypeUpd}>	8%	-	-	**31%**	21%	11%	-	6%	**5**	2
4	<{AttrAdd@TableCreation},{AttrAdd@ExistTable}>	16%	**31%**	**26%**	**40%**	**27%**	27%	11%	16%	**8**	5
4	<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrAdd@ExistTable}>	16%	**31%**	**26%**	**40%**	**25%**	25%	11%	16%	**8**	4
5	<{AttrAdd@TableCreation},{AttrAdd@TableCreation}>	-	-	-	25%	6%	**23%**	**17%**	-	**4**	1
5	<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrAdd@TableCreation,AttrAdd@TableCreation}>	-	-	-	25%	6%	**23%**	**17%**	-	**4**	1
5	<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrAdd@TableCreation}>	-	-	-	25%	6%	**23%**	**17%**	-	**4**	1
5	<{AttrAdd@TableCreation},{AttrAdd@TableCreation,AttrAdd@TableCreation}>	-	-	-	25%	6%	**23%**	**17%**	-	**4**	1
6	<{AttrAdd@TableCreation},{AttrDel@TableDel,AttrDel@TableDel},{AttrAdd@TableCreation}>	-	-	-	25%	6%	**23%**	17%	-	**4**	3
6	<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrDel@TableDel},{AttrAdd@TableCreation}>	-	-	-	25%	6%	**23%**	**17%**	-	**4**	3
6	<{AttrAdd@TableCreation},{AttrDel@TableDel},{AttrAdd@TableCreation,AttrAdd@TableCreation}>	-	-	-	25%	6%	**23%**	**17%**	-	**4**	3

6	<{AttrAdd@TableCreation},{AttrDel@TableDel},{AttrAdd@TableCreation}>	-	-	-	25%	6%	**23%**	**17%**	-	**4**	3
7	<{AttrDel@TableDel,AttrDel@TableDel},{AttrAdd@TableCreation,AttrAdd@TableCreation}>	-	-	-	27%	7%	**40%**	**17%**	-	**4**	2
7	<{AttrDel@TableDel,AttrDel@TableDel},{AttrAdd@TableCreation}>	-	-	-	27%	7%	**40%**	**17%**	-	**4**	2
7	<{AttrDel@TableDel},{AttrAdd@TableCreation,AttrAdd@TableCreation}>	-	-	-	27%	7%	**40%**	**17%**	-	**4**	2
7	<{AttrDel@TableDel},{AttrAdd@TableCreation}>	-	-	-	27%	7%	**40%**	**17%**	-	**4**	2
8	<{AttrAdd@TableCreation},{AttrDel@TableDel,AttrDel@TableDel}>	6%	11%	-	**62%**	21%	**67%**	**17%**	**25%**	**7**	4
8	<{AttrAdd@TableCreation},{AttrDel@TableDel}>	6%	11%	-	**62%**	21%	**67%**	**17%**	**25%**	**7**	4
8	<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrDel@TableDel,AttrDel@TableDel}>	6%	11%	-	**62%**	21%	**67%**	**17%**	**25%**	**7**	3
8	<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrDel@TableDel}>	6%	11%	-	**62%**	21%	**67%**	**17%**	**25%**	**7**	3
9	<{AttrAdd@ExistTable,AttrDel@ExistTable}>	17%	**40%**	9%	28%	11%	10%	9%	9%	**8**	2
9	<{AttrAdd@ExistTable,AttrAdd@ExistTable,AttrDel@ExistTable}>	10%	**33%**	-	21%	7%	-	-	9%	**5**	2
9	<{AttrAdd@ExistTable,AttrDel@ExistTable,AttrDel@ExistTable}>	10%	**33%**	-	21%	6%	-	-	9%	**5**	2
9	<{AttrAdd@ExistTable,AttrAdd@ExistTable,AttrDel@ExistTable,AttrDel@ExistTable}>	8%	**31%**	-	17%	-	-	-	9%	**4**	3
10	<{AttrAdd@ExistTable},{AttrAdd@ExistTable,AttrDel@ExistTable}>	13%	**33%**	-	17%	8%	-	7%	9%	**6**	2
10	<{AttrAdd@ExistTable},{AttrAdd@ExistTable,AttrAdd@ExistTable,AttrDel@ExistTable}>	8%	**31%**	-	14%	6%	-	-	9%	**5**	2
11	<{AttrAdd@ExistTable},{AttrDel@ExistTable}>	**60%**	**33%**	**22%**	23%	21%	7%	16%	13%	**8**	4
11	<{AttrAdd@ExistTable},{AttrDel@ExistTable,AttrDel@ExistTable}>	11%	29%	**17%**	17%	6%	-	-	13%	**6**	2
12	<{AttrTypeUpd},{AttrDel@ExistTable}>	**58%**	24%	-	19%	14%	-	13%	13%	**6**	2
12	<{AttrTypeUpd},{AttrTypeUpd},{AttrDel@ExistTable}>	**36%**	9%	-	13%	10%	-	9%	9%	**6**	2
12	<{AttrTypeUpd},{AttrDel@ExistTable},{AttrTypeUpd}>	**55%**	-	-	16%	13%	-	7%	6%	**5**	2
12	<{AttrTypeUpd},{AttrTypeUpd},{AttrDel@ExistTable},{AttrTypeUpd}>	**33%**	-	-	10%	6%	-	-	6%	**4**	2
13	<{AttrTypeUpd},{AttrTypeUpd}>	**61%**	11%	9%	**45%**	**55%**	12%	**24%**	**34%**	**8**	5
13	<{AttrTypeUpd,AttrTypeUpd},{AttrTypeUpd}>	13%	-	-	24%	**32%**	-	11%	19%	**5**	3
13	<{AttrTypeUpd},{AttrTypeUpd,AttrTypeUpd},{AttrTypeUpd}>	9%	-	-	23%	**24%**	-	6%	13%	**5**	2
13	<{AttrTypeUpd},{AttrTypeUpd,AttrTypeUpd}>	26%	-	-	24%	**25%**	-	9%	19%	**5**	2
14	<{AttrTypeUpd},{AttrAdd@ExistTable}>	**59%**	24%	9%	24%	**25%**	6%	16%	**25%**	**8**	4
14	<{AttrTypeUpd},{AttrTypeUpd},{AttrAdd@ExistTable}>	**38%**	9%	-	19%	13%	-	11%	19%	**6**	2
15	<{AttrTypeUpd},{AttrAdd@ExistTable},{AttrDel@ExistTable}>	**57%**	20%	-	16%	13%	-	11%	9%	**6**	2
15	<{AttrTypeUpd},{AttrAdd@ExistTable},{AttrDel@ExistTable},{AttrTypeUpd}>	**53%**	-	-	12%	11%	-	6%	6%	**5**	2
16	<{AttrTypeUpd},{AttrAdd@ExistTable},{AttrTypeUpd}>	**57%**	-	-	21%	-	10%	19%	**19%**	**5**	4
16	<{AttrTypeUpd},{AttrTypeUpd},{AttrAdd@ExistTable},{AttrTypeUpd}>	**35%**	-	-	16%	8%	-	7%	16%	**5**	2
17	<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrTypeUpd},{AttrTypeUpd}>	8%	-	-	**38%**	**39%**	11%	-	9%	**5**	3
17	<{AttrAdd@TableCreation},{AttrTypeUpd},{AttrTypeUpd}>	8%	-	-	**38%**	**42%**	11%	-	9%	**5**	3
17	<{AttrAdd@TableCreation,AttrAdd@TableCreation},{AttrTypeUpd,AttrTypeUpd}>	13%	-	-	23%	**28%**	5%	-	6%	**5**	2
17	<{AttrAdd@TableCreation},{AttrTypeUpd,AttrTypeUpd}>	13%	-	-	23%	**28%**	5%	-	6%	**5**	2
	<{AttrAdd@TableCreation},{AttrDel@ExistTable}>	11%	22%	9%	**28%**	17%	11%	10%	9%	**8**	2
	<{AttrAdd@TableCreation,AttrAdd@TableCreation}>	**36%**	**53%**	**65%**	**90%**	**73%**	**96%**	**19%**	**69%**	**8**	1
	<{AttrDel@ExistTable},{AttrAdd@ExistTable}>	27%	**29%**	**17%**	25%	21%	-	**19%**	13%	**7**	4
	<{AttrDel@ExistTable,AttrDel@ExistTable}>	15%	**33%**	**17%**	24%	8%	-	6%	13%	**7**	3
	<{AttrDel@ExistTable},{AttrDel@ExistTable}>	18%	**29%**	**13%**	21%	11%	-	14%	9%	**7**	3
	<{AttrDel@ExistTable},{AttrTypeUpd}>	**58%**	9%	-	**28%**	18%	6%	10%	9%	**7**	3
	<{AttrAdd@ExistTable},{AttrDel@ExistTable},{AttrAdd@ExistTable}>	24%	22%	**17%**	20%	17%	-	14%	9%	**7**	2

<{AttrAdd@ExistTable},{AttrDel@ExistTable},{AttrDel@ExistTable}>	15%	24%	**13%**	17%	7%	-	14%	6%	**7**	2
<{AttrAdd@ExistTable},{AttrTypeUpd},{AttrAdd@ExistTable}>	22%	9%	9%	15%	17%	-	7%	**19%**	**7**	2
<{AttrDel@TableDel,AttrDel@TableDel}>	19%	**38%**	-	**67%**	**31%**	**70%**	**17%**	**28%**	**7**	1
<{AttrTypeUpd},{AttrTypeUpd},{AttrAdd@ExistTable},{AttrDel@ExistTable}>	**34%**	9%	-	11%	8%	-	9%	9%	**6**	3
<{AttrAdd@ExistTable,AttrDel@ExistTable},{AttrAdd@ExistTable}>	8%	**29%**	-	20%	8%	-	7%	9%	**6**	2
<{AttrAdd@ExistTable},{AttrDel@ExistTable},{AttrTypeUpd}>	**57%**	7%	-	19%	15%	-	6%	6%	**6**	2
<{AttrDel@ExistTable,AttrDel@ExistTable},{AttrAdd@ExistTable}>	6%	16%	**13%**	17%	8%	-	-	13%	**6**	2
<{AttrDel@ExistTable},{AttrDel@ExistTable},{AttrAdd@ExistTable}>	9%	20%	**13%**	19%	6%	-	11%	-	**6**	2
<{AttrDel@ExistTable},{AttrAdd@ExistTable,AttrDel@ExistTable}>	8%	**29%**	-	18%	-	-	7%	6%	**5**	2

Στη συνέχεια περιγράφονται αναλυτικά όλες οι ομάδες συχνών ακολουθιών.

Ομάδα 1

Οι ακολουθίες της συγκεκριμένης ομάδας περιγράφουν το εξής: με την προσθήκη ενός πεδίου σε έναν πίνακα, είναι συχνό το φαινόμενο της προσθήκης ενός ακόμα σε μεταγενέστερη χρονική στιγμή το οποίο ακολουθείται από την διαγραφή ενός πεδίου στον ίδιο πίνακα. Η συγκεκριμένη ακολουθία εμφανίζεται σε όλες τις συλλογές δεδομένων. Στην συγκεκριμένη ομάδα υπάρχει επίσης και η ακολουθία <{AttrAdd@ExistTable}, {AttrAdd@ExistTable, AttrAdd@ExistTable}> η οποία περιγράφει την εισαγωγή 2 πεδίων σε έναν πίνακα, την ίδια χρονική, αφού έχει προηγηθεί μία εισαγωγή ενός πεδίου στον πίνακα σε κάποια προγενέστερη χρονική στιγμή.

Ομάδες 2,3 και 4

Πίνακες που δημιουργούνται, δέχονται αργότερα μία εισαγωγή ενός πεδίου και σε κάποια μεταγενέστερη χρονική στιγμή αλλαγή στον τύπο κάποιου πεδίου τους.

Επίσης σε μεγάλο ποσοστό των πινάκων είναι συχνό το φαινόμενο της αλλαγής στον τύπο ενός πεδίου ύστερα από εισαγωγή κάποιου πεδίου, σε μια προγενέστερη χρονική στιγμή, χωρίς βέβαια να προηγείται η δημιουργία του πίνακα. Χαρακτηριστικό παράδειγμα είναι η περίπτωση του Atlas, του οποίου στη μετάβαση 32 όλοι οι ζωντανοί πίνακες δέχονται εισαγωγή πεδίων και στη μετάβαση 49 όλοι οι ζωντανοί πίνακες δέχονται αλλαγές στον τύπο των πεδίων τους (για αυτόν το λόγο έχουμε υποστήριξη 60%).

Ομάδες 5,6,7 και 8

Στις συγκεκριμένες ομάδες παρατηρείται το φαινόμενο της δημιουργίας πίνακα, διαγραφή του ύστερα από ένα χρονικό διάστημα, και τέλος ξαναδημιουργία του

σε μια μεταγενέστερη χρονική στιγμή. Η συγκεκριμένη συχνή ακολουθία εμφανίζεται σε 4 συλλογές δεδομένων. Επιπλέον, το γεγονός ότι πίνακες που δημιουργούνται σε κάποιο ενδιάμεσο χρονικά σημείο, στη ζωή μιας βάσης, συχνά καταστρέφονται κάποια χρονική στιγμή μετά την δημιουργία τους είναι πιο συχνό και εμφανίζεται σε 7 συλλογές δεδομένων, πράγμα το οποίο είναι λογικό.

Ομάδες 9,10 και 11

Στις ομάδες 9, 10 και 11 είναι συχνό το φαινόμενο της προσθαφαίρεσης πεδίων σε κάποιον πίνακα την ίδια χρονική στιγμή. Αξίζει να σημειωθεί ότι ένα ποσοστό αυτών των προσθαφαιρέσεων περιγράφουν τις μετονομασίες πεδίων που λαμβάνουν χώρα σε έναν πίνακα. Αυτό μπορεί να παρατηρηθεί εύκολα διότι τα πεδία που διαγράφονται, επιστρέφουν στον πίνακα, την ίδια χρονική στιγμή, με την μέθοδο της εισαγωγής νέων πεδίων τα οποία έχουν τον ίδιο τύπο με τα διαγραμμένα και μια μικρή παραλλαγή στο όνομα τους. Χαρακτηριστικό παράδειγμα είναι το Biosql στο οποίο ένα μεγάλο ποσοστό των πινάκων δέχονται εισαγωγές και διαγραφές στη μετάβαση 21. Επιπλέον είναι συχνό το φαινόμενο της προσθαφαίρεσης πεδίων σε κάποιον πίνακα την ίδια χρονική στιγμή ύστερα από την εισαγωγή ενός πεδίου.

Ομάδες 12 και 13

Η ομάδα 13 περιγράφει το εξής: ένα μεγάλο ποσοστό των πινάκων, στις περισσότερες συλλογές δεδομένων, αν υποστεί τουλάχιστον μία αλλαγή στον τύπο ενός πεδίου, τότε συχνά θα γίνει ακόμα μία αλλαγή στον τύπο ενός πεδίου του πίνακα στο μέλλον. Στην συγκεκριμένη ομάδα εμφανίζονται και διάφορες παραλλαγές της παραπάνω ακολουθίας όπως για παράδειγμα είναι η ακολουθία <{AttrTypeUpd},{AttrTypeUpd,AttrTypeUpd},{AttrTypeUpd}> η οποία περιγράφει 4 αλλαγές στον τύπο των πεδίων ενός πίνακα. Η ομάδα 12 περιγράφει τις ακολουθίες οι οποίες περιέχουν 2 αλλαγές στον τύπο των πεδίων των πινάκων σε διαφορετικές χρονικές στιγμές και συνδυάζονται με μία διαγραφή πεδίου. Δηλαδή, συχνά η διαγραφή ενός πεδίου του πίνακα ακολουθεί 2 αλλαγές στον τύπο των πεδίων του ή επίσης μια διαγραφή ενός πεδίου ακολουθεί 1 αλλαγή στον τύπο κάποιου πεδίου και ακολουθείται από ακόμα μία αλλαγή τύπου πεδίου.

Ομάδα 14, 15 και 16

Δύο αλλαγές στον τύπο των πεδίων ενός πίνακα σε διαφορετικές χρονικές στιγμές συχνά ακολουθούνται από μία εισαγωγή και μία αλλαγή στον τύπο ενός πεδίου σε

δύο διαφορετικές χρονικές στιγμές. Επίσης, ανάμεσα σε δύο αλλαγές τύπων πεδίων κάποιου πίνακα μπορεί να παρεμβάλλεται αρχικά μία εισαγωγή ενός πεδίου και σε κάποια μεταγενέστερη στιγμή μία διαγραφή ενός πεδίου.

Ομάδα 17

Οι συχνές ακολουθίες της συγκεκριμένης ομάδας περιγράφουν το εξής: πίνακες που δημιουργούνται σε κάποια ενδιάμεση χρονικά στιγμή στη ζωή μιας βάσης, δέχονται τουλάχιστον δύο αλλαγές στον τύπο των πεδίων τους. Οι αλλαγές στον τύπο των πεδίων μπορούν να συμβαίνουν είτε την ίδια χρονική στιγμή, είτε σε δύο διαφορετικές χρονικές στιγμές.

Από τις συχνές ακολουθίες οι οποίες δεν οργανώθηκαν σε κάποια ομάδα παρατηρείται ότι η εισαγωγή ενός πεδίου ή η διαγραφή ενός πεδίου συχνά ακολουθείται από επιπλέον αλλαγές στα πεδία του πίνακα είτε αυτά αφορούν εισαγωγές και διαγραφές είτε αλλαγές στον τύπο των πεδίων. Επιπλέον, εμφανίζονται κάποιες συχνές ακολουθίες οι οποίες δεν έχουν κάποια χρήσιμη πληροφορία όπως είναι για παράδειγμα η ακολουθία <{AttrDel@TableDel, AttrDel@TableDel}> η οποία περιγράφει ότι πίνακες του διαγράφονται έχουν τουλάχιστον δύο πεδία.

Κεφάλαιο 6. Επίλογος

6.1 Σύνοψη και συμπεράσματα

Στόχος της συγκεκριμένης διπλωματικής εργασίας ήταν η υλοποίηση ενός εργαλείου το οποίο λαμβάνοντας ως είσοδο τις αλλαγές που έχει υποστεί μια βάση δεδομένων από την στιγμή που δημιουργήθηκε να μπορεί να εξάγει ως έξοδο συχνές ακολουθίες γεγονότων που προκύπτουν από τις συγκεκριμένες αλλαγές. Για την εξαγωγή συχνών ακολουθιών χρησιμοποιήθηκε ο βασικός αλγόριθμος εξόρυξης συχνών ακολουθιών με την μέθοδο Apriori υλοποιώντας δύο διαφορετικούς τρόπους μέτρησης της υποστήριξης. Το εργαλείο που υλοποιήθηκε παρέχει στον χρήστη ένα γραφικό περιβάλλον στο οποίο μπορεί να επιλέξει ως είσοδο την ιστορία της βάσης που επιθυμεί και να εξάγει τις συχνές ακολουθίες γεγονότων οι οποίες ξεπερνούν το ελάχιστο κατώφλι υποστήριξης που έχει δοθεί ως είσοδος στο εργαλείο.

Από τα αποτελέσματα των πειραμάτων που πραγματοποιήθηκαν στη συγκεκριμένη εργασία προκύπτουν τα παρακάτω:

- Ένα μεγάλο ποσοστό των αλλαγών αφορούν δημιουργία και διαγραφή πινάκων.
- Πίνακες που δημιουργούνται σε κάποια ενδιάμεση χρονική στιγμή στη ζωή μιας βάσης συχνά δέχονται επιπλέον αλλαγές στα πεδία τους, είτε αυτά αφορούν εισαγωγές και διαγραφές είτε αλλαγές τύπου.

6.2 Μελλοντικές επεκτάσεις

Υπάρχει μία μεγάλη λίστα από πράγματα που έχει ενδιαφέρον να υλοποιηθούν στο μέλλον τα οποία περιγράφονται στις επόμενες παραγράφους.

Βελτίωση απόδοσης αλγορίθμου. Παρατηρήθηκε ότι με την χρήση ακόμα και μικρών κατωφλίων υποστήριξης της τάξης του 1% και μικρότερα η χρονική απόδοση του αλγορίθμου είναι αρκετά χαμηλή. Έτσι, αξίζει να ανευρεθεί ένας εναλλακτικός και πιο αποδοτικός αλγόριθμος που να επιλύει το συγκεκριμένο πρόβλημα πιο αποδοτικά.

Εύρεση νέου τρόπου μέτρησης υποστήριξης. Ο τρόπος μέτρησης COBJ μετράει συχνές ακολουθίες οι οποίες συμβαίνουν σε ένα μεγάλο ποσοστό των πινάκων της βάσης. Έχει παρατηρηθεί ότι κάποιοι από τους πίνακες της βάσης δεν δέχονται καμία αλλαγή ή δέχονται έναν μικρό αριθμό αλλαγών. Σε αυτήν την περίπτωση αν υπάρχει ένα μικρό ποσοστό των πινάκων που δέχονται μεγάλο αριθμό από αλλαγές τότε η υποστήριξή τους θα μειωθεί αρκετά. Οπότε, σύμφωνα με τα παραπάνω υπάρχει η ανάγκη εύρεσης ενός διαφορετικού τρόπου μέτρησης της υποστήριξης που επιλύει το παραπάνω πρόβλημα.

Εισαγωγή τιμής κατωφλίου μεγέθους παραθύρου. Στην παρούσα εργασία οι υποψήφιες ακολουθίες που αναζητούνται δεν έχουν κάποιον περιορισμό όσον αφορά τη μέγιστη επιτρεπτή χρονική διαφορά ανάμεσα σε δύο ή περισσότερα γεγονότα (μέγεθος παραθύρου = άπειρο). Με την χρήση ενός κυλιόμενου παραθύρου μπορούμε να αναζητήσουμε ακολουθίες που το πρώτο με το τελευταίο γεγονός δεν ξεπερνούν μία χρονική διαφορά ή επίσης μπορούμε να ορίσουμε ένα χρονικό κενό ανάμεσα σε δύο διαδοχικά γεγονότα.

Αναζήτηση αλλαγών που συνέβησαν χρονικά κοντά με ένα δοθέν γεγονός. Μία εύλογη απορία που δημιουργείται είναι για παράδειγμα να εξετάσουμε τι συμβαίνει λίγο μετά την δημιουργία ενός πίνακα, ή λίγο πριν την διαγραφή του. Μία επέκταση του εργαλείου θα μπορούσε να είναι η εύρεση συχνών ακολουθιών που αρχίζουν με ένα δοθέν γεγονός ή που το τελευταίο τους γεγονός είναι ίδιο με το δοθέν. Αυτή η επέκταση μπορεί να συνδυαστεί με την χρήση χρονικού παραθύρου ώστε να περιορίζεται χρονικά η αναζήτηση γύρω από την χρονική στιγμή που συνέβη το δοθέν γεγονός.

Ομαδοποίηση πινάκων. Υπάρχουν τρεις διαφορετικές κατηγορίες ομαδοποίησης σε ότι αφορά τους πίνακες μιας βάσης όπως διατυπώθηκε στο κεφάλαιο 3. Μπορούμε να ομαδοποιήσουμε τους πίνακες μιας βάσης: κατά πίνακα, κατά το είδος πίνακα και κατά ομάδες πινάκων. Η συγκεκριμένη εργασία επικεντρώθηκε στην μελέτη κάθε πίνακα ξεχωριστά, οπότε μία μελλοντική επέκταση μπορεί να περιλαμβάνει τα υπόλοιπα είδη ομαδοποίησης.

Διαφορετική διαίρεση χρόνου. Ο χρόνος μπορεί να διαιρεθεί με βάση το version ID, με βάση κάποια χρονικά σημεία, κάποιο χρονικό διάστημα ή να χωριστεί σε φάσεις. Η αναλυτική περιγραφή των παραπάνω επεκτάσεων υπάρχει στο κεφάλαιο 3.

Διαφορετικός τρόπος αναπαράστασης των γεγονότων. Τα γεγονότα μπορούν να αναπαρασταθούν με διαφορετικούς τρόπους: Είδος γεγονότος ακολουθούμενο από το όνομα του πεδίου, είδος γεγονότος ακολουθούμενο από πλήθος που αναφέρεται στον αριθμό ίδιων γεγονότων που συνέβησαν την ίδια χρονική στιγμή. Τα διαφορετικά είδη αναπαράστασης των γεγονότων περιγράφονται επίσης στο κεφάλαιο 3.

Κεφάλαιο 7. Βιβλιογραφία

[AgSr94] Rakesh Agrawal, Ramakrishnan Srikant. Fast Algorithms for Mining Association Rules, Proceedings of the 20th International Conference on Very Large Data Bases (VLDB '94), pp. 487-499, 1994

[AgSr95] Rakesh Agrawal, Ramakrishnan Srikant. Mining Sequential Patterns. Proceedings of the 5th International Conference on Extending Database Technology. (EDBT '96) Advances in Database Technology, pp. 3-17, 1996

[CMTZ08] Carlo A. Curino, Hyun J. Moon, Letizia Tanca, Carlo Zaniolo, Schema Evolution In Wikipedia toward a Web Information System Benchmark, International Conference on Enterprise Information Systems (ICEIS '08), pp. 323-332, 2008

[LMRW97] Lehman, M.M., Ramil, J.F, Wernick, P., Perry, D.E., Turski, W.M. Metrics and laws of software evolution - the nineties view. Fourth International, Software Metrics Symposium. pp. 20 - 32 November 1997

[MaKK99] Mahesh Joshi, George Karypis, Vipin Kumar. A Universal Formulation of Sequential Patterns. Technical Report, University of Minnesota, Minneapolis, pp. 7-12, May 1999

[PaMV10] Pang-Ning Tan, Michael Steinbach, Vipin Kumar, Εισαγωγή στην εξόρυξη δεδομένων. pp. 363-541, 2010

[Sjøb91] Dag Sjøberg. Quantifying Schema Evolution. Information and Software Technology, Vol. 35, No. 1, pp. 35-44, January 1993

[Skou13] Ιωάννης Σκουλής. Ανάλυση της εξέλιξης σχήματος για βάσεις δεδομένων σε λογισμικό ανοιχτού κώδικα. Μεταπτυχιακή εργασία εξειδίκευσης. Πανεπιστήμιο Ιωαννίνων, Τμήμα Μηχανικών Η/Υ και

Πληροφορικής, pp. 1-124, Σεπτέμβριος 2013

[SkVZ14] Ioannis Skoulis, Panos Vassiliadis, Apostolos Zarras. Open-Source Databases: Within, Outside, or Beyond Lehman's Laws of Software Evolution?. 26th International Conference on Advanced Information Systems Engineering (CAiSE 2014). 16-20 June 2014, Thessaloniki, Hellas.