Θέματα μεταπτυχιακών, διπλωματικών και πτυχιακών εργασιών για το ακ. έτος 2021- 2022

Π. ΒΑΣΙΛΕΙΑΔΗΣ – 2021-09-21

Όλες οι εργασίες που θα λάβουν χώρα θα πρέπει να χρησιμοποιήσουν υποχρεωτικά

- Apache Spark για τη διαχείριση ερωτημάτων και αλγορίθμων αναλυτικής δεδομένων
- Java Spring [& Spring Boot] για την ανάπτυξη του front-end

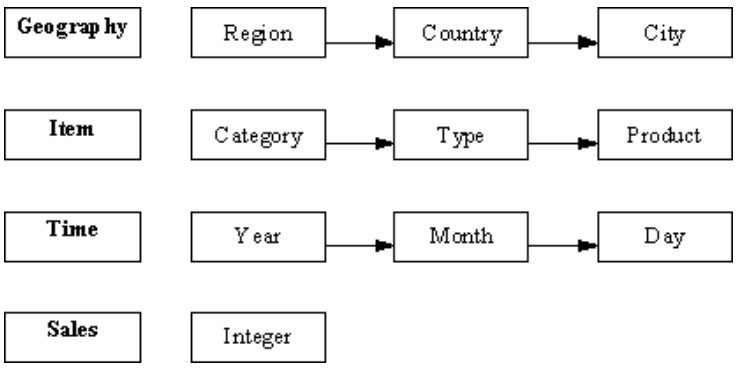
/* Δυστυχώς, είναι τελικά ο μόνος τρόπος να μάθω και γω => θα μάθουμε μαζί.*/

1. Analytics For OLAP & Business Intelligence

Οι εφαρμογές On-Line Analytical Processing (OLAP) αντιμετωπίζουν τα δεδομένα σαν *κύβους* (*data cubes*), οι οποίοι αποτελούνται από διαστάσεις και μετρήσιμες τιμές. Οι κύβοι αποτελούνται από *διαστάσεις* (χρόνος, γεωγραφική περιοχή κ.λπ.) και *μετρήσιμα μεγέθη* (π.χ. σύνολο πωλήσεων). Οι διαστάσεις οργανώνονται σε λογικές *ιεραρχίες*. Για παράδειγμα, ο χρόνος μπορεί να οργανωθεί στην ιεραρχία χρόνος, μήνας, μέρα. Για παράδειγμα, οι πωλήσεις ενός οργανισμού μπορούν να μοντελοποιηθούν σαν ένας κύβος της μορφής

Day (Year-month-day)	Product	City	Sales
1997-01-01	“Report to El Greco”	Rhodes	15
1997-01-01	“Ace of Spades”	Paris	8
1997-01-01	“Report to El Greco”	Athens	11

με διαστάσεις οργανωμένες σε *ιεραρχίες*, όπως:



Ο χρήστης μπορεί να κάνει ερωτήσεις όπως μετασχηματισμός του κύβου σε διαφορετικά επίπεδα ιεραρχίας, επιλογή κάποιων υποσυνόλων του κύβου κ.λ.π.

Για το υπό συζήτηση σύστημα, η παρούσα πτυχιακή εντάσσεται στο μεγάλο στόχο να αναθεωρηθεί εκ βάθρων το μοντέλο κύβων και η απάντηση κάθε ερώτησης να επαυξηθεί με αποτελέσματα μηχανικής μάθησης, εκτός από τα δεδομένα αυτά καθαυτά.

Ο χρήστης υποβάλλει ένα OLAP ερώτημα (πρακτικά ένα ερώτημα συνάθροισης που περιλαμβάνει group-by και where clause). Το ερώτημα εκτελείται σε μία back-end OLAP engine, η οποία είναι ήδη σε μια πρώτη εκδοχή. Τα αποτελέσματα πρέπει να παρουσιαστούν στον χρήστη σε μια εκδοχή "ευέλικτου Excel" που επιτρέπει την οπτικοποίηση των αποτελεσμάτων και τη διαδραστική τους επερώτηση.

Σε οποιοδήποτε textbook βάσεων δεδομένων, θα βρείτε ένα κεφάλαιο για αποθήκες δεδομένων και OLAP για τις βασικές έννοιες.

1.1 Υπόβαθρο για όσες εργασίες απαιτούν Delian Cubes

Για την πρότασή μας στο Παν. Ιωαννίνων, δείτε στο

http://www.cs.uoi.gr/~pvassil/projects/olap_III/index.html

- το σχετικό άρθρο στο DOLAP 2018 [http://ceur-ws.org/Vol-2062/paper07.pdf] καθώς και
- τον τελεστή compare ως τον "Assess" operator στο EDBT 2021 [https://doi.org/10.5441/002/edbt.2021.12]

Έχουμε ήδη κατασκευάσει ένα σύστημα επερώτησης data cubes, με μια απλή διαπροσωπεία σε JavaFX. Το σχετικό project έχει το δικό του repository στο Github:

https://github.com/pvassil/DelianCubeEngine

Στο σύστημα Delian Cubes, εξετάζουμε την απάντηση σε ερωτήματα μέσω *intentional queries & result mining*. Ο χρήστης υποβάλλει ένα OLAP ερώτημα (πρακτικά ένα ερώτημα συνάθροισης που περιλαμβάνει group-by και where clause). Το σύστημα παράγει ως έξοδο (α) τα αποτελέσματα της ερώτησης, (β) μια λίστα από μοντέλα που είναι το αποτέλεσμα του τρεξίματος αλγορίθμων εξόρυξης γνώσης και στατιστικών (correlations, clusters, …) και (γ) ένα mapping μεταξύ των (α) και (β).

1.2 Master Theses for Business Intelligence

Για θέματα Master μπορούμε να δούμε πιθανές υλοποιήσεις κάποιων τελεστών στο σύστημα Delian Cubes.

Η γενική ιδέα περιλαμβάνει: (α) τον ορισμό του τελεστή επακριβώς, (β) τον ορισμό των αλγορίθμων και μοντέλων που θα τρέξουμε (θα χρειαστούμε κάτι σαν ένα απλοϊκό αλγόριθμο και μια βελτίωσή του) και (γ) την πειραματική αξιολόγηση σε (α) πραγματικά δεδομένα (που είναι μικρά) και (β) τεχνητά δεδομένα σε μεγέθη διαφορετικής κλίμακας (για να μελετήσουμε την κλιμακοσιμότητα).

1.3 Πες μου όλα όσα μπορείς για ένα KPI

ΠΕΡΙΛΗΨΗ: Αυτόματη εκτέλεση ερωτημάτων αποτίμησης ενός Key Performance Indicator και παραγωγή report

ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ: Ένας Key Performance Indicator (βασικός δείκτης απόδοσης) πρακτικά αφορά ένα μετρήσιμο μέγεθος που αποτιμά (συχνά σε συνεχή τρόπο) το επίπεδο / ποσοστό στο οποίο ένας οργανισμός καταφέρνει να ανταποκριθεί σε στόχους «υψηλού επιπέδου» που έχει θέσει. Οι KPI's μπορούν να ορισθούν σε πολλά επίπεδα λεπτομέρειας, π.χ., από πολύ αφαιρετικοί KPI σε πιο υψηλό επίπεδο («πώς πηγαίνουν οι πωλήσεις?») έως πολύ τεχνικοί KPIs σε πολύ συγκεκριμένο επίπεδο («το Τμήμα ΧΧ πρέπει να αυξήσει την απορρόφηση πόρων κατά 25% μέχρι το Νοέμβριο»). Σε τελικό επίπεδο, όμως, οι KPI's καταλήγουν να είναι πολύ συγκεκριμένοι και μετρήσιμοι. Εμείς θα ασχοληθούμε με KPI's χαμηλού επιπέδου που μπορούν να αποτιμηθούν με ερωτήσεις σε μια βάση δεδομένων.

Π.χ., έστω ένα super market που παίρνει την απόφαση: «τα έσοδα (σε Ευρώ) του Τμήματος Πώλησης Γαλακτοκομικών θα πρέπει να αυξηθούν κατά 20% φέτος σε σχέση με πέρυσι» όπου

- κάθε μήνα θα αποτιμάται το εν λόγω μέγεθος του μήνα αυτού,

- θα συγκρίνεται με τον αντίστοιχο μήνα πέρυσι και θα αποτιμάται η σχετική αύξηση, και,

- στο τέλος του χρόνου θα αποτιμηθεί η συνολική ετήσια αύξηση.

Εμείς θα αγνοήσουμε όλα τα μη τεχνικά στοιχεία των KPIs, π.χ., ποιος είναι υπεύθυνος, τι ενέργειες θα κάνουμε για να τον πετύχουμε κλπ , και θα θεωρήσουμε ότι ένας δείκτης απόδοσης έχει τις εξής ιδιότητες:

- την τρέχουσα τιμή του αυτή τη στιγμή (που προκύπτει συνήθως από μια ερώτηση στη βάση δεδομένων)
- την τάση των τελευταίων *k* χρονικών στιγμών για την τιμή του δείκτη
- μια τιμή στόχο που θα έπρεπε να έφτανε κατ' ελάχιστον/μέγιστον ο στόχος (ανάλογα με το χαρακτήρα του -- π.χ., για τις πωλήσεις όσο πιο ψηλά τόσο πιο καλά, ενώ για τα έξοδα το αντίθετο)
- μια συνάρτηση που απεικονίζει την τρέχουσα τιμή σε ένα εύρος από [-1 .. 1], σαν ένδειξη του πόσο καλά πηγαίνουν τα πράματα, καθώς και μία συνάρτηση που περιγράφει την τιμή αυτή με κάποιους λεκτικούς χαρακτηρισμούς

Ένα σύστημα θα πρέπει να υποστηρίζει σε offline mode

1. την εγγραφή μιας OLAP βάσης κύβων, με τα μέτρα και τις ιεραρχίες της (ιδεατά σε Spark Dataset / DataFrame – αν όχι, σε κάποια σχεσιακή βάση, όπως ξέρουμε να το κάνουμε στο Delian)
2. την εγγραφή ενός νέου KPI πάνω στη βάση δεδομένων αυτή (θα έχουμε πολλούς τέτοιους)

Ένα σύστημα θα πρέπει να υποστηρίζει σε online mode, όταν υπάρχει το σχετικό αίτημα

1. Την αποτίμηση όλων των παραπάνω χαρακτηριστικών ενός KPI (με τις σωστές ερωτήσεις)

2. Την εφαρμογή πράξεων roll-up & drill down στις διαστάσεις που εμπλέκονται στον ορισμό του KPI για να δούμε αν η κατάσταση είναι συγχρονισμένη με τις γενικότερες τάσεις (roll-up) και ποια είναι τα στοιχεία που επηρεάζουν την κατάσταση (drill-down) /* π.χ., στο παράδειγμά μας, θα ψάξουμε για όλα τα προϊόντα διατροφής με ένα roll-up, αλλά και για την επί μέρους επίδοση γάλακτος, τυριών, … με drill-down για να δούμε ποιος επηρεάζει την κατάσταση των γαλακτοκομικών */
3. Τη σύγκριση με παρόμοιες καταστάσεις στη βάση των σχετικών φίλτρων (π.χ., αντί για γαλακτοκομικά, να δούμε πώς πάει το τμήμα απορρυπαντικών, το τμήμα κρεάτων, κ.ο.κ)
4. Την πρόβλεψη των μελλοντικών τιμών του δείκτη με βάση κάποιο μοντέλο πρόβλεψης της χρονοσειράς τιμών του δείκτη
5. Την παραγωγή ενός report με τα ευρήματα των επί μέρους, αυτομάτως παραχθέντων, αποτελεσμάτων

ΕΠΙΠΕΔΟ: *Διπλωματική για Μηχανικούς ή MSc*

Προφανώς η έκταση της εργασίας θα εξαρτηθεί από το επίπεδο στο οποίο θα διεξαχθεί.

Αν και δε το συνιστώ, μπορούν και να το αναλάβουν από κοινού 2 φοιτητές ως Διπλωματική.

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java για το back-end , Apache Spark, if front-end: Java Spring/Spring Boot

ΠΡΟΚΛΗΣΕΙΣ και ΟΦΕΛΗ:

Η δυσκολία βρίσκεται κυρίως στη σωστή σχεδίαση του λογισμικού που δημιουργεί τα ερωτήματα αυτομάτως. Τα οφέλη για ένα φοιτητή είναι: (α) τεχνογνωσία σε θέματα αναλυτικής δεδομένων και (β) hands-on σε ένα ευμέγεθες κομμάτι λογισμικού.

Η εργασία είναι πλέον κατάλληλη για φοιτητές με ταλέντο σε προγραμματιστικά θέματα και σε θέματα αναλυτικής δεδομένων. Πρέπει να μπορείτε να ανταπεξέλθετε και στη σχεδίαση λογισμικού και στην ανάπτυξη του σχετικού συστήματος.

Απαιτούμενα προσόντα είναι η πολύ καλή γνώση Java και η δεδηλωμένη δέσμευση να ολοκληρωθεί η εργασία αυστηρά εντός ενός έτους από την ανάληψή της

1.4 Πες μου όλα όσα μπορείς για ένα data set

ΠΕΡΙΛΗΨΗ: Αυτόματο profiling ενός συνόλου δεδομένων και παραγωγή του σχετικού report

ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ: Ένα τυπικό data set δεν είναι παρά ένα αρχείο κειμένου, με γραμμές, όπου κάθε γραμμή είναι μια εγγραφή, τα πεδία της οποίας χωρίζονται με κάποιο διαχωριστικό (p.x., tabs, comma, pipe, …). Θέλουμε, αφού εγγράψουμε ένα data set, την 100% αυτόματη παραγωγή ενός στατιστικού προφίλ για το data set.

Συγκεκριμένα, ο χρήστης θα πρέπει να εγγράψει ένα data set και με τη βοήθεια του συστήματος να δηλώσει τα πεδία του, και τον τύπο τους (int, double, dateTime, Boolean, enum of class labels, …).

Θα ονομάζουμε labeled πεδία, αυτά για τα οποία υπάρχει ένα πολύ συγκεκριμένο, πεπερασμένο σύνολο τιμών (όπως π.χ., οι μήνες ή οι μέρες της εβδομάδας – σαν enum δλδ). Επίσης, αν έχει κάποιο ειδικό ενδιαφέρον για κάποια πεδία (βλ. παρακάτω) θα πρέπει να το δηλώσει.

Στη συνέχεια, το σύστημα, απολύτως αυτόματα, θα πρέπει να κάνει τα παρακάτω (τα οποία παρέχονται έτοιμα από το Spark)

1. Προφίλ για τις κολώνες (δλδ., ιστόγραμμα τιμών, καθώς και min, max, median, mean, stdev, …)
2. Υπολογισμός της συσχέτισης στηλών (correlation)
3. Αν ο χρήστης έχει δηλώσει κανόνες labeling για μια στήλη, χαρακτηρισμός των τιμών αυτής της στήλης και παραγωγή της νέας labeled στήλης

Για κάθε labeled πεδίο (ή/και για κάθε πεδίο με λιγότερες από 10 τιμές στο πεδίο τιμών του):

4. Αυτόματο τρέξιμο decision trees για κάθε labeled field στη βάση όλων των πεδίων του data set. Αν ο χρήστης έχει δηλώσει ότι για θέλει μόνο κάποια πεδία να εμπλακούν στην παραγωγή ενός συγκεκριμένου decision tree, τότε εμπλέκονται μόνο αυτά.
5. Clustering των εγγραφών του data set και αποτίμηση της ποιότητας του clustering
6. Hypothesis testing & chi-square test

Για όλα τα αυτομάτως παραχθέντα tasks το σύστημα θα πρέπει να κάνει

- Αποτίμηση της σημαντικότητας (π.χ., silhouette evaluation of clustering, value of correlation for correlation, …)
- Καταγραφή ενός αναλυτικού report με όλα τα ευρήματα
- Καταγραφή ενός συνοπτικού report με τα ευρήματα που έχουν τουλάχιστον ένα ελάχιστο threshold σημαντικότητας
- /* Ιδεατά, και παραγωγή bar charts, lines, scatter-plots με τα σημαντικά αποτελέσματα */

Μπορείτε να παίξετε λίγο με ένα απλό εργαλείο data analytics, όπως π.χ., το Orange για να δείτε μια ιδέα για το πώς ο χρήστης διαδραστικά μπορεί να κάνει κάποια από αυτά τα tasks. Εμείς θέλουμε να γίνονται αυτόματα.

Διευκρίνιση: αν και το θέμα ΔΕΝ σχετίζεται με ιεραρχίες, OLAP, cube queries, αλλά μόνο με απλά analytics over simple text data sets, είναι καθαρά στο χώρο του data analytics

ΕΠΙΠΕΔΟ: *Διπλωματική για Μηχανικούς*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java για το back-end , Apache Spark, if front-end: Java Spring/Spring Boot

ΠΡΟΚΛΗΣΕΙΣ και ΟΦΕΛΗ:

Η δυσκολία βρίσκεται κυρίως στη σωστή και επεκτάσιμη σχεδίαση του λογισμικού. Τα οφέλη για ένα φοιτητή είναι: (α) τεχνογνωσία σε θέματα αναλυτικής δεδομένων και (β) hands-on σε ένα ευμέγεθες κομμάτι λογισμικού.

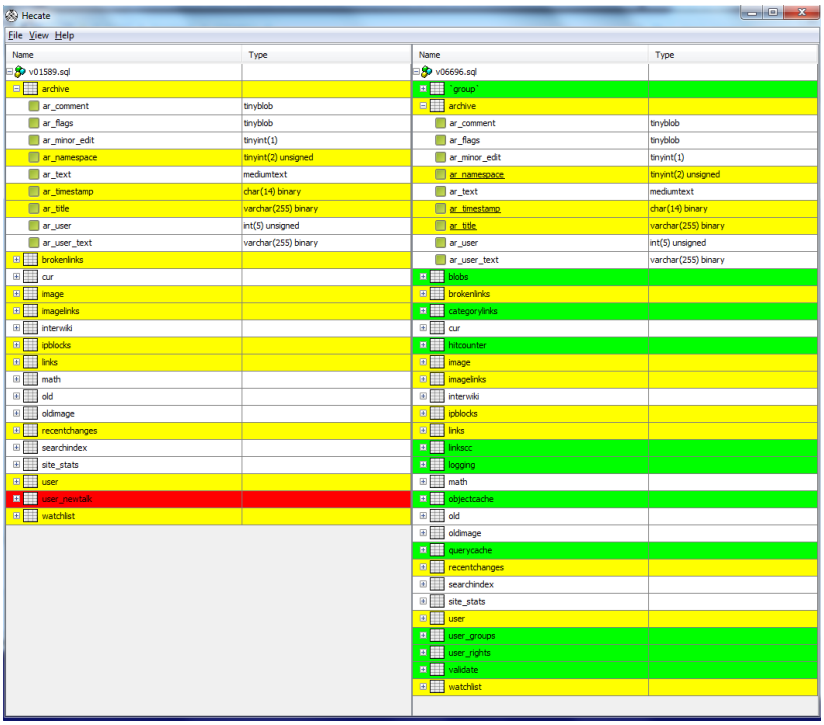
Η εργασία είναι πλέον κατάλληλη για φοιτητές με ταλέντο σε προγραμματιστικά θέματα και σε θέματα αναλυτικής δεδομένων. Πρέπει να μπορείτε να ανταπεξέλθετε και στη σχεδίαση λογισμικού και στην ανάπτυξη του σχετικού συστήματος.

Απαιτούμενα προσόντα είναι η πολύ καλή γνώση Java και η δεδηλωμένη δέσμευση να ολοκληρωθεί η εργασία αυστηρά εντός ενός έτους από την ανάληψή της

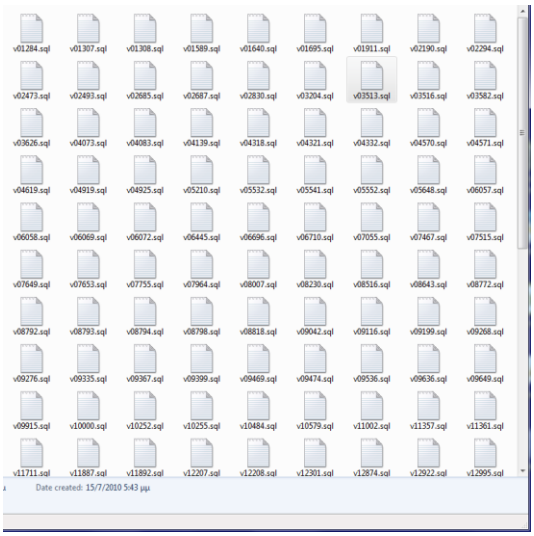
2 Μελέτη της Εξέλιξης Βάσεων Δεδομένων

Μια βάση δεδομένων, από τη 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).

Το εργαλείο **Hecate** [https://github.com/DAINTINESS-Group/Hecate] μπορεί να συγκρίνει δύο σχήματα και να βρει τις διαφορές τους (κίτρινο: updated attributes, red: deletions, green: insertions).

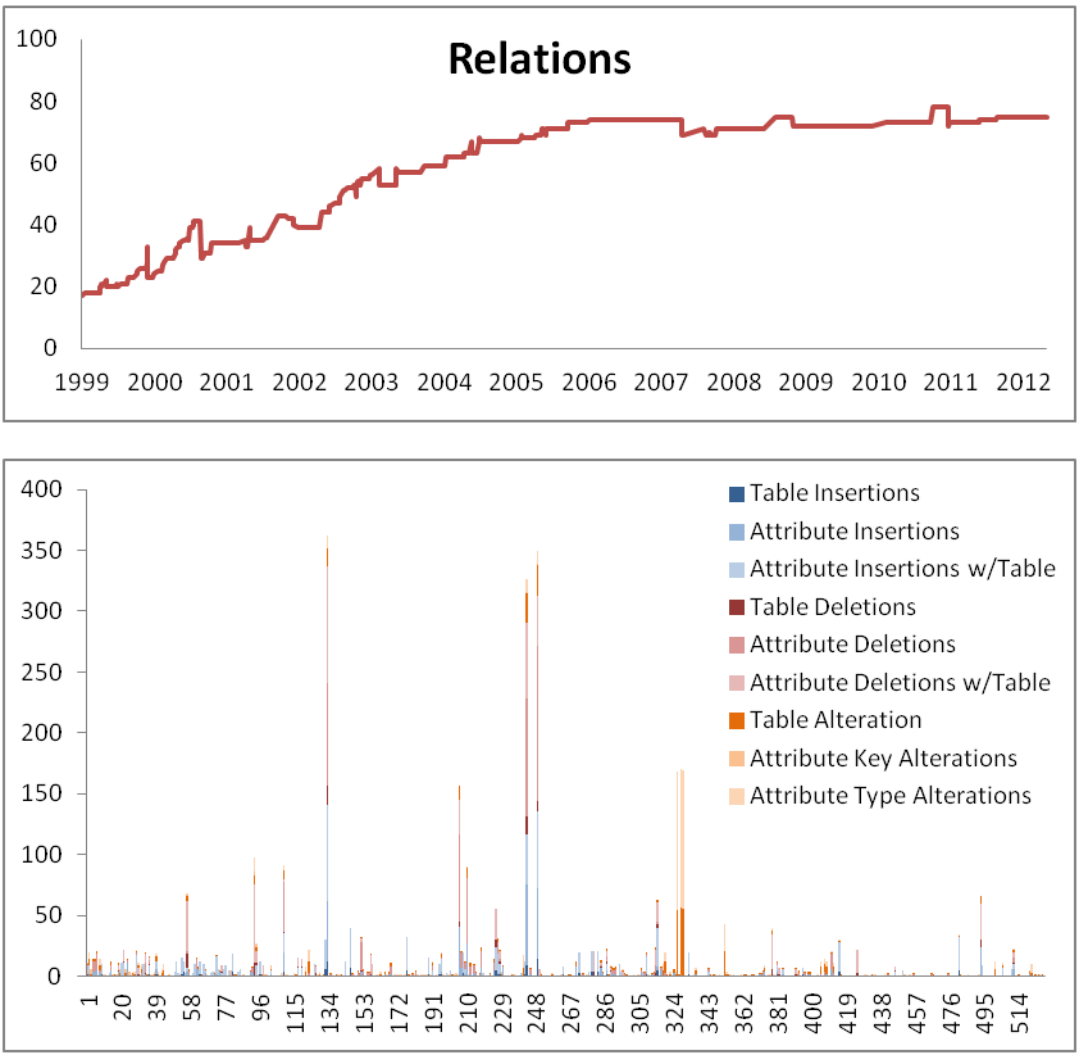


Επιπλέον, υπάρχουν αρκετές συλλογές από εκδόσεις του σχήματος της ιδίας βάσης (παρακάτω ένα screenshot από τη βάση της Wikimedia).



Η Εκάτη μπορεί να ταξινομήσει τις επί μέρους εκδοχές του σχήματος και να τις συγκρίνει διαδοχικά.

Έχουμε ήδη χρησιμοποιήσει την Εκάτη για να επεξεργαστούμε την εξέλιξη σχήματος διαφόρων βάσεων δεδομένων ανοιχτού λογισμικού, όπως για παράδειγμα, της βάσης της Wikimedia (της βάσης δεδομένων πίσω από τη Wikipedia), της βάσης του Atlas Trigger (του εργαλείου που διαχειρίζεται τα δεδομένα από το πείραμα Atlas για την ανεύρεση του μποζονίου του Χιγκς), της Ensembl (του εργαλείου για τη διαχείριση των δεδομένων του ανθρώπινου γονιδιώματος) και πολλών CMS's (opencart, coppermine, phpBB, typo3, …). Έχουμε επίσης συλλέξει την ιστορία από πολλά συστήματα ανοιχτού κώδικα που περιλαμβάνουν βάσεις δεδομένων και καταγράφουν και τις εκδοχές τους σε δημόσια αποθετήρια (κυρίως github, αλλά και svn) αλλά δεν την έχουμε επεξεργαστεί ακόμα.

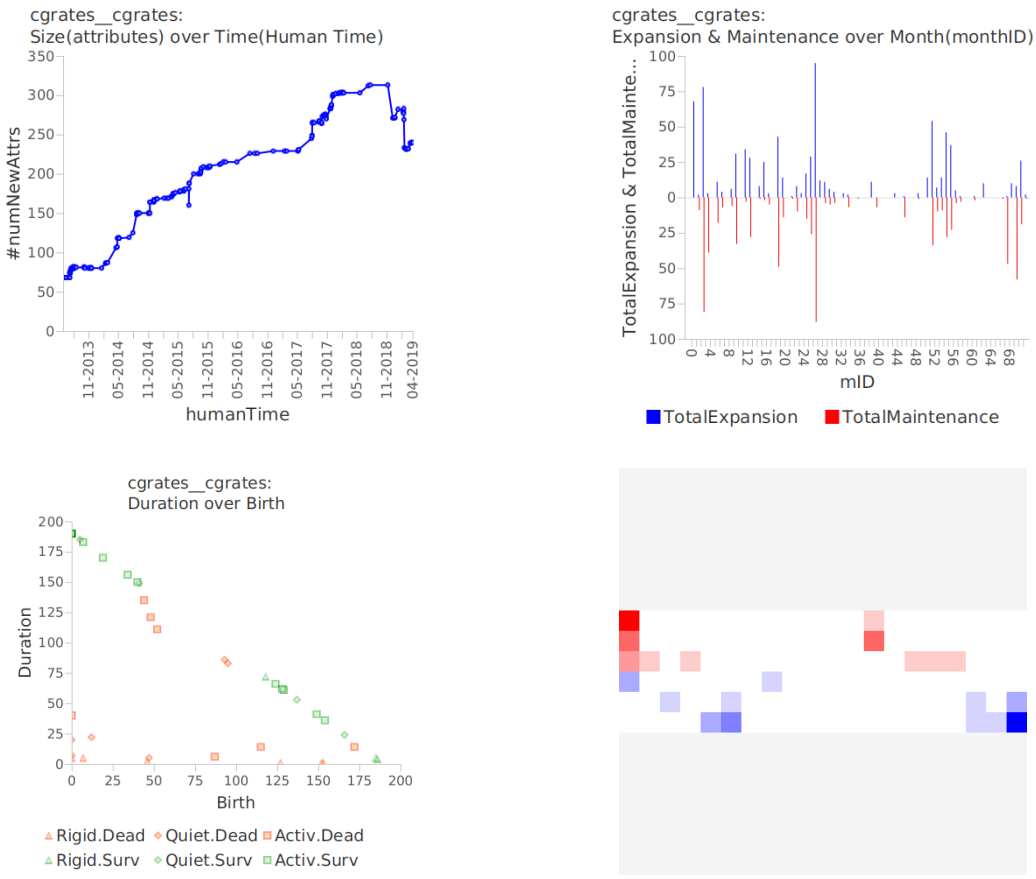


Στο παραπάνω σχήμα βλέπετε (α) το πώς εξελίχθηκε το μέγεθος του σχήματος της βάσης στο χρόνο και (β) τον παλμό των αλλαγών (το πώς διαρθρώθηκαν οι αλλαγές σε κάθε monitored version) για τη βάση Ensembl.

Το εργαλείο ROSES από τη Μ. Ζέρβα είναι ένα εργαλείο βασισμένο σε μια βάση δεδομένων, όπου έχουμε περάσει την εξαχθείσα πληροφορία, για να μπορούμε να απομονώνουμε εύκολα υποσύνολα πινάκων που μας ενδιαφέρουν και να οπτικοποιούμε γραφικές παραστάσεις. Το εργαλείο MUSES από τον Α. Παππά μας επιτρέπει να εξάγουμε πρότυπα συχνών υποακολουθιών από τα δεδομένα μας.

Το εργαλείο Heraclitus Fire [`https://github.com/pvassil/HeraclitusFire`] χρησιμοποιείται για να συμπληρώσει τα αρχικώς εξαχθέντα αποτελέσματα της Εκάτης με επιπλέον στατιστικά -- ενδεικτικά:

- Ανάλυση των αλλαγών σαν (α) επέκταση (προσθέσεις πινάκων και πεδίων) και (β) συντήρηση (διαγραφές πινάκων και πεδίων, αλλαγές τύπων, κλπ) και περαιτέρω ανάλυση σε άλλες υποκατηγορίες και μετρικές
- Συγκεντρωτικά στατιστικά για τις αλλαγές ανά commit και αλλαγές μήνα
- Έλεγχο κάποιων πρωτόλειων προτύπων
- Γραφικές παραστάσεις

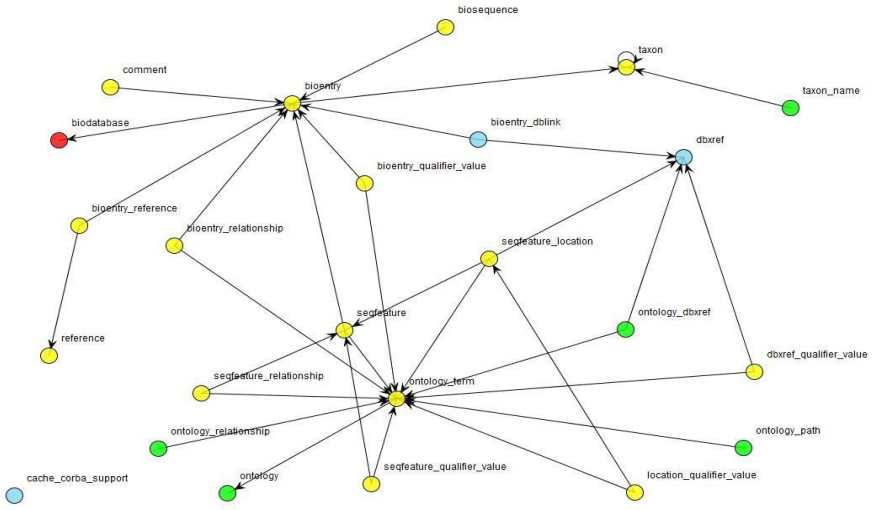


Αυτή τη στιγμή, έχουμε συλλέξει τις ζωές για 195 σχήματα από Free Open Source Software systems. Δείτε το άρθρο στο ICDE'21 στο βοηθητικό υλικό – και στο [`https://doi.org/10.1109/ICDE51399.2021.00008`]

Ο Ηράκλειτος είναι το κεντρικό εργαλείο με το οποίο μελετάμε το πώς εξελίσσονται τα σχήματα που έχουμε συλλέξει.

Το εργαλείο «Πλουτάρχου Βίοι Παράλληλοι» [https://github.com/DAINTINESS-Group/PlutarchParallelLives] είναι ένα εργαλείο από τον Θ. Γιάχο και μετέπειτα το Ν. Παντελίδη το οποίο απεικονίζει την εξέλιξη των πινάκων μιας βάσης δεδομένων σε παράλληλες γραμμές. Κάθε version αναπαριστάται από 3 κολώνες για εισαγωγές, διαγραφές και ενημερώσεις πινάκων. Οι γεννήσεις πινάκων και πεδίων φαίνονται με πράσινο και οι διαγραφές με κόκκινο χρώμα.

Το εργαλείο «Παρμενίδεια Αλήθεια» [https://github.com/DAINTINESS-Group/ParmenidianTruth] είναι ένα εργαλείο από τον Μ. Κολοζώφ που αναπαριστά το σχήμα μιας βάσης δεδομένων με ένα διαχρονικό γράφημα και φροντίζει να οπτικοποιεί κάθε version και τις εκδοχές της σε ένα slide μιας Powerpoint παρουσίασης (πρακτικά φτιάχνει μια ταινία για το πώς αλλάζει το σχήμα της βάση 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.

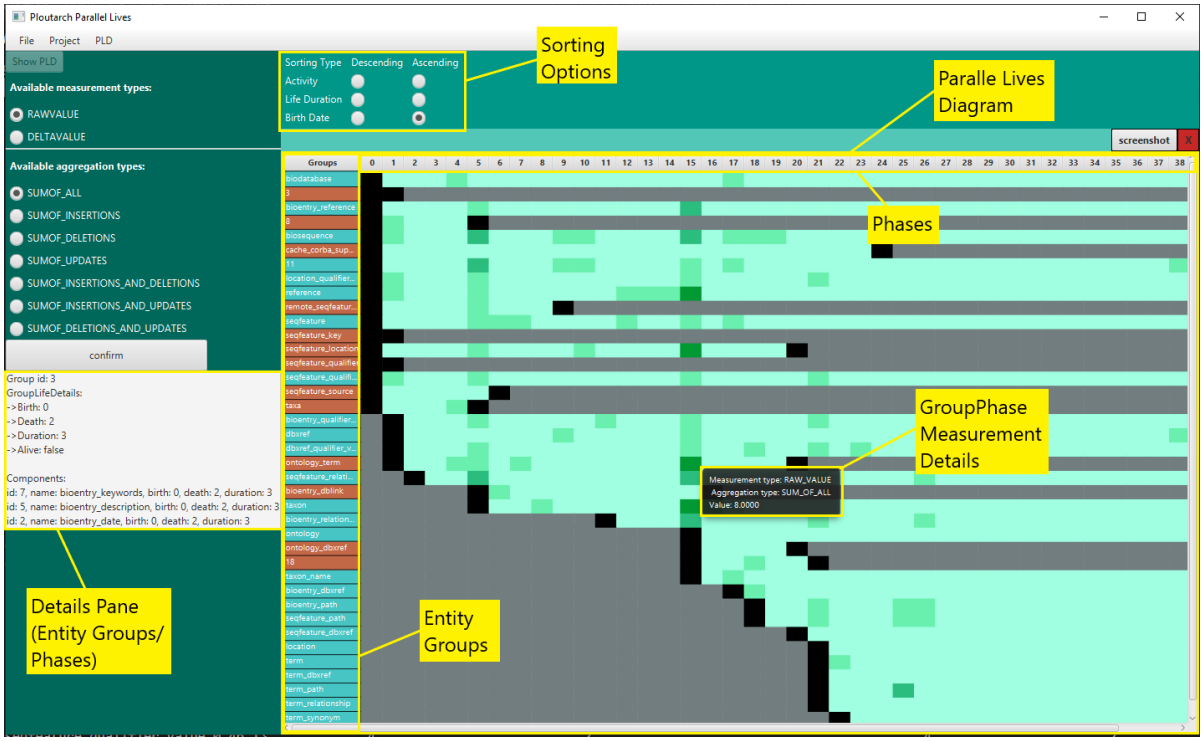
http://www.cs.uoi.gr/~pvassil/projects/schemaBiographies/

2.1 Πλούταρχου Βίοι Παράλληλοι

ΠΕΡΙΛΗΨΗ: Επέκταση και αναμόρφωση του εργαλείου Πλουτάρχου Βίοι Παράλληλοι

ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ: Το εργαλείο Πλουτάρχου Βίοι Παράλληλοι απεικονίζει οπτικά την εξέλιξη ενός σχεσιακού σχήματος. https://github.com/DAINTINESS-Group/PlutarchParallelLives

Στον Πλούταρχο, το «Διάγραμμα Παράλληλων Ζωών» χρησιμοποιείται για να αναπαραστήσουμε οπτικά την εξέλιξη μίας ομάδας οντοτήτων ως έναν 2Δ πίνακα, όπου κάθε οντότητα αντιστοιχεί σε μία γραμμή, κάθε χρονική στιγμή αντιστοιχεί σε μία στήλη και η εντάσεις των χρωμάτων κάθε κελιού αναπαριστούν το μέγεθος της δραστηριότητας για τον συνδυασμό οντότητας και χρονικής στιγμής.



Τα ζητούμενα από την Διπλωματική είναι:

- Αναμόρφωση του front-end ώστε να μπορεί -με ένα απλό 2Δ πίνακα- να χειριστεί μεγάλο αριθμό γραμμών
- Να μπορούμε να ορίσουμε «μοτίβα» (π.χ., στην παραπάνω φωτό βλέπουμε: «σκάλα προοδευτικών γεννήσεων πινάκων» στο κάτω και κεντρικό μέρος της οθόνης, «κολώνα με αλλαγές σε όλους στους πίνακες» στη μέση της οθόνης, και «μεγάλο τετράγωνο μη αλλαγών» στο κάτω δεξιά μέρος της οθόνης, με μια καταπράσινη ομοιόμορφη απουσία συμβάντων». Ένα μοτίβο είναι ένα διδιάστατο παράθυρο για το οποίο έχουμε προδιαγράψει τη μορφή των κελιών του. Το ζητούμενο εδώ είναι πόσο εκφραστικά μπορούμε να προδιαγράψουμε «μοτίβα».
- Να τρέχουμε αλγορίθμους που μπορούν να «σκοράρουν» τετράγωνα σε σχέση με το πόσο καλά καλύπτουν ένα μοτίβο, και να χαρακτηρίζουν με μια label το αν κάτι είναι αξιόλογο προς αναφορά ή όχι.
- Να μπορούμε να περιγράψουμε τα ευρήματα από τα μοτίβα που ανακαλύψαμε σε ένα τελικό report που να λέει την ιστορία του πληθυσμού των παράλληλων ζωών.

Το αν θα χρησιμοποιηθεί Spark / Spring είναι κάτι που θα προκύψει στην πορεία σε σχέση με τους αλγορίθμους και το front-end.

ΕΠΙΠΕΔΟ: *Διπλωματική για Μηχανικούς*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java

ΠΡΟΚΛΗΣΕΙΣ και ΟΦΕΛΗ:

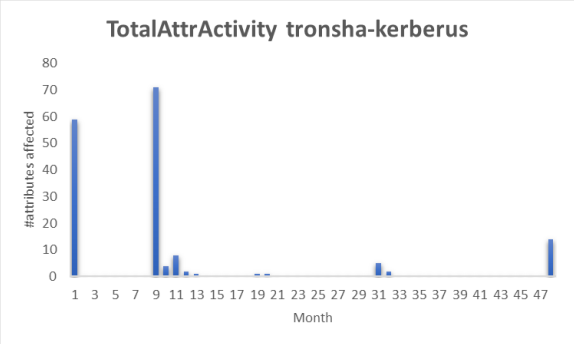
Η δυσκολία βρίσκεται κυρίως στη σωστή και επεκτάσιμη σχεδίαση του pattern definition mechanism. Τα οφέλη για ένα φοιτητή είναι: (α) τεχνογνωσία σε ζητήματα σχεδίασης, ελέγχου, και hands-on σε ένα ευμέγεθες κομμάτι λογισμικού, και (β), εμπλοκή στο χώρο του προγραμματισμού διαδραστικών γραφικών διαπροσωπειών.

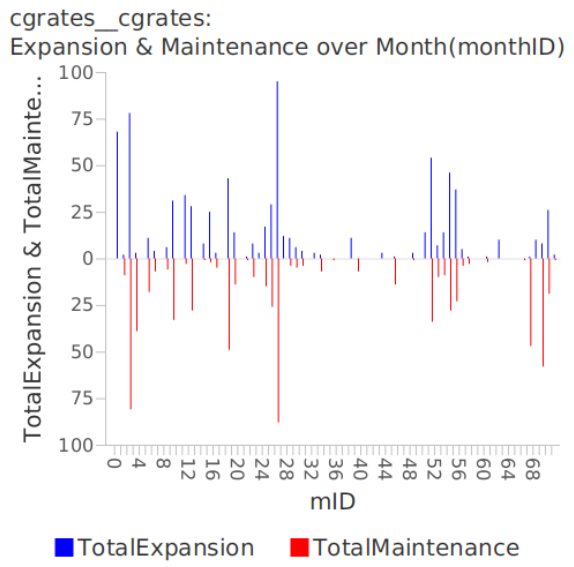
Η εργασία είναι πλέον κατάλληλη για φοιτητές με ταλέντο σε προγραμματιστικά θέματα και σε θέματα διαχείρισης δεδομένων. Πρέπει να μπορείτε να ανταπεξέλθετε και στη σχεδίαση λογισμικού και στην ανάπτυξη του σχετικού συστήματος.

Απαιτούμενα προσόντα είναι η πολύ καλή γνώση Java και η δεδηλωμένη δέσμευση να ολοκληρωθεί η εργασία αυστηρά εντός ενός έτους από την ανάληψή της.

2.2 Εξαγωγή state diagrams για την εξέλιξη βάσεων δεδομένων

ΠΕΡΙΛΗΨΗ: Derivation of state diagrams on how databases evolve





ΣΥΝΤΟΜΗ ΠΕΡΙΓΡΑΦΗ: Από τον Ηράκλειτο, μπορούμε να λαμβάνουμε στατιστικά για το πώς εξελίσσεται η βάση σε κάθε commit ή σε κάθε μήνα. Τα διαγράμματα που έχουμε εδώ δείχνουν δύο πολύ διαφορετικές συμπεριφορές: (α) ένα σύστημα με ένα και μόνο μεγάλο σημείο συντήρησης και κάποια (λίγα) σημεία μικρής αλλαγής, και (β) ένα σύστημα με συστηματική αλλαγή του σχήματος σε βάθος 6 ετών.

Το ζητούμενο από την εργασία είναι:

- Για κάθε μήνα, να χαρακτηρίσουμε το μέγεθος των αλλαγών με ένα label
- Να εξάγουμε τη ζωή του σχήματος ως ένα διάγραμμα καταστάσεων (π.χ., στο πάνω σχήμα: αρχική γέννηση -> έρημος -> μεγάλη αλλαγή -> σειρά μικρών αλλαγών -> σχεδόν έρημος -> μεσαία αλλαγή) και να κρατήσουμε τα σχετικά στατιστικά για τις καταστάσεις και τις μεταβάσεις (θα χρειαστεί να κάνουμε fine tune την εξομάλυνση κάποιων ενδιάμεσων καταστάσεων, όπως ίσως παρατηρήσατε)
- Να κάνουμε cluster τα διαγράμματα καταστάσεων (θα χρησιμοποιήσουμε την ταξινομία του άρθρου στο ICDE '21), ώστε να βγάλουμε και κέντρα του κάθε cluster, αλλά και να κρατήσουμε τα σχετικά στατιστικά για τις καταστάσεις και τις μεταβάσεις
- Να οπτικοποιήσουμε τα διαγράμματα καταστάσεων και τα στατιστικά τους

Τελικά, το αποτέλεσμα θα είναι χαρακτηριστικά state diagrams on how databases evolve

ΕΠΙΠΕΔΟ: *Διπλωματική για Μηχανικούς [ή ίσως και MSc]*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java

ΠΡΟΚΛΗΣΕΙΣ και ΟΦΕΛΗ: Η εργασία είναι πλέον κατάλληλη για φοιτητές με ταλέντο σε θέματα data mining και σε θέματα αναλυτικής δεδομένων. Πρέπει να μπορείτε να ανταπεξέλθετε και στη σχεδίαση λογισμικού και στην ανάπτυξη του σχετικού συστήματος.

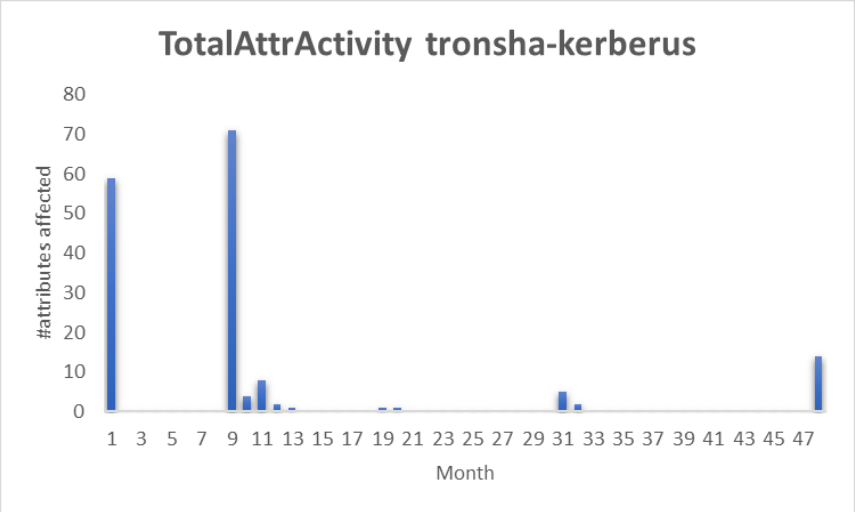
Απαιτούμενα προσόντα είναι η πολύ καλή γνώση Java και η δεδηλωμένη δέσμευση να ολοκληρωθεί η εργασία αυστηρά εντός ενός έτους από την ανάληψή της

2.3 Στατιστικό μοντέλο πρόβλεψης σημαντικών αλλαγών

Οι σεισμολόγοι προβλέπουν τους σεισμούς με ένα στατιστικό μοντέλο που στηρίζεται στην ιδέα «μετράω τους μικρούς σεισμούς σε μια περιοχή, και όσο πιο πολλοί έχουν γίνει, τόσο μεγαλύτερη η πιθανότητα να συμβεί ένας μεγάλος».

Δείτε την εικόνα 7, στο άρθρο του 2018 από το περιοδικό Nature στο φάκελο με το βοηθητικό υλικό.

Στις βάσεις δεδομένων, όμως, δε φαίνεται να ισχύει το παραπάνω. Η παρακάτω εικόνα δείχνει το μέγεθος των αλλαγών ανά μήνα για ένα σχήμα ενός project.



Τα ζητούμενα από την εργασία είναι:

1. Να κατασκευαστεί ένα σύστημα που φορτώνουμε χρονοσειρές και μπορούμε να αποτιμήσουμε τα στατιστικά μοντέλα των σεισμολόγων. Αυτό θα πρέπει να είναι ένα σύστημα γενικής φύσεως, π.χ., να το δοκιμάσουμε με τιμές μετοχών, σεισμών, … και φυσικά, εξελισσόμενων σχημάτων (έχουμε ένα σύνολο από 195 σχήματα).
2. Να μπορούμε να οπτικοποιήσουμε τις γραφικές παραστάσεις.
3. Να διερευνήσουμε αν μπορούμε να εξάγουμε κάποιο μοντέλο που να περιγράφει τον τρόπο εξέλιξης των σχημάτων.

ΕΠΙΠΕΔΟ: *MSc ή ίσως (στην απλή της μορφή) και Διπλωματική για Μηχανικούς*

ΠΛΑΤΦΟΡΜΑ ΕΡΓΑΣΙΑΣ: Java

ΠΡΟΚΛΗΣΕΙΣ και ΟΦΕΛΗ: Η δυσκολία βρίσκεται κυρίως στη σωστή κατανόηση του θεωρητικού υποβάθρου. Τα οφέλη για ένα φοιτητή είναι: (α) τεχνογνωσία σε θέματα αναλυτικής δεδομένων και (β) πιθανότητα για ένα ερευνητικά σημαντικό αποτέλεσμα.

Η εργασία είναι πλέον κατάλληλη για φοιτητές με ταλέντο σε θέματα στατιστικής και σε θέματα αναλυτικής δεδομένων. Πρέπει να μπορείτε να ανταπεξέλθετε και στη σχεδίαση λογισμικού και στην ανάπτυξη του σχετικού συστήματος.

Απαιτούμενα προσόντα είναι η πολύ καλή γνώση Java και η δεδηλωμένη δέσμευση να ολοκληρωθεί η εργασία αυστηρά εντός ενός έτους από την ανάληψή της