



Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής

Πρόγραμμα Μεταπτυχιακών Σπουδών

«Προηγμένα Συστήματα Πληροφορικής»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	Εφαρμογή της μηχανής διαχείρισης κινούμενων αντικειμένων SECONDO
Όνοματεπώνυμο Φοιτητή	Πολλάτος Βίκτωρ
Πατρώνυμο	Διονύσιος
Αριθμός Μητρώου	ΜΠΣΠ/ 09003
Επιβλέπων	Πελέκης Νίκος, Λέκτορας

Τριμελής Εξεταστική Επιτροπή

(υπογραφή)

(υπογραφή)

(υπογραφή)

Νίκος Πελέκης
Λέκτορας

Γιάννης Θεοδορίδης
Καθηγητής

Γιάννης Σίσκος
Καθηγητής

ABSTRACT

This thesis presents Secondo, an extensible DataBase Management System (DBMS). We analyze Secondo's parts and architecture together with the way these parts operate and communicate. We also describe and use the BerlinMOD benchmark which is suitable for MOD (Mobility Object Databases) such as Secondo. BerlinMOD comprises of a set of queries giving us the opportunity to benchmark Secondo's overall performance.

Περίληψη

Η διατριβή αυτή παρουσιάζει το Secondo, ένα επεκτάσιμο Σύστημα Διαχείρισης Βάσεων Δεδομένων. Παρουσιάζεται η αρχιτεκτονική του συστήματος για να δούμε τον τρόπο που επικοινωνούν τα διαφορετικά μέρη του Secondo μεταξύ τους και ποιες λειτουργίες εκτελούν. Χρησιμοποιούμε το σύστημα αναφοράς BerlinMOD το οποίο έχει δημιουργηθεί για να τεστάρει συστήματα σαν το Secondo. Διαθέτει μια σειρά από επερωτήσεις κατάλληλες για να μπορούμε να συγκρίνουμε Συστήματα Διαχείρισης Βάσεων Δεδομένων(DBMS) και κυρίως αυτά που αφορούν σε Βάσεις Κινούμενων Αντικειμένων(MOD).

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ.....	8
ΚΕΦΑΛΑΙΟ 1: Αρχιτεκτονική και Χαρακτηριστικά Secondo DBMS.....	9
1.1 SECONDO KERNEL (πυρήνας).....	10
1.3 ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΧΡΗΣΤΗ (GUI).....	12
1.4 Διεπαφές SECONDO.....	13
ΚΕΦΑΛΑΙΟ 2 : BerlinMOD.....	13
2.1 Μοντέλα βάσεων δεδομένων.....	14
2.2 Περιγραφή των αντικειμένων που χρησιμοποιούνται στην βάση.....	15
2.3 Επιλογή/Παρουσίαση των Επερωτήσεων.....	16
2.3.1 BerlinMOD/R Επερωτήσεις.....	16
2.3.2 Παρουσίαση των Επερωτήσεων.....	16
ΚΕΦΑΛΑΙΟ 3 : Εφαρμογή του BerlinMOD στο SECONDO – Αποτελέσματα/Χρόνοι.....	24
ΚΕΦΑΛΑΙΟ 4 : Οδηγίες Εγκατάστασης Secondo.....	29
ΑΝΑΦΟΡΕΣ.....	39

ΕΙΣΑΓΩΓΗ

Ο σύγχρονος τρόπος ζωής έχει υποχρεώσει ουσιαστικά τον κάθε ένα από εμάς να έχει στην κατοχή του τουλάχιστον μία έξυπνη φορητή συσκευή η οποία τις περισσότερες φορές είναι και το προσωπικό μας τηλέφωνο. Η εξέλιξη της τεχνολογίας μας έχει δώσει την δυνατότητα μέσα από αυτές τις έξυπνες συσκευές να δηλώνουμε αλλά και να λαμβάνουμε κάθε στιγμή γεωγραφικές πληροφορίες σχετικά με το που είμαστε, που βρισκόμαστε ή που σκοπεύουμε να πάμε. Όλη αυτή η διαδικασία έχει σαν αποτέλεσμα να δημιουργείται ένας τεράστιος όγκος γεωγραφικών δεδομένων διαθέσιμος για περαιτέρω επεξεργασία και αυτό το γεγονός σε συνδυασμό με την ραγδαία ανάπτυξη των Γεωγραφικών Πληροφοριακών Συστημάτων (GIS) έχει δυναμώσει την έρευνα γύρω από τις Βάσεις Κινούμενων Αντικειμένων (MOD).

Ένα κινούμενο αντικείμενο μπορεί να είναι μια οποιαδήποτε γεωμετρία (σημείο, γραμμή, περιοχή, κλπ.) που αλλάζει θέσεις ή ακόμη και το σχήμα του (περιοχή) με την πάροδο του χρόνου. Μέσω συσκευών καταγραφής θέσης έχουμε την θέση ενός αντικειμένου μία χρονική στιγμή T1 και έπειτα την επόμενη θέση την στιγμή T2 ανάλογα πάντα με το δείγμα που παίρνει η συσκευή καταγραφής. Εμείς όμως είμαστε αυτοί που καθορίζουμε τι συμβαίνει ανάμεσα στις δύο χρονικές στιγμές, T1-T2. Η τροχιά ενός αντικειμένου καθορίζεται από το είδος της παρεμβολής όπου τελικά θα έχουμε καθορίσει εμείς.

Στην παρούσα μελέτη δοκιμάζουμε το Σύστημα Διαχείρισης Βάσεων Δεδομένων – Secondo DBMS (<http://dna.fernuni-hagen.de/secondo/>), το οποίο είναι ένα επεκτάσιμο σύστημα σχεδιασμένο στο πανεπιστήμιο FernUniversität της Χάγης στην Γερμανία με σκοπό την υλοποίηση και τον πειραματισμό με διάφορα μοντέλα δεδομένων. Μετά από μια σύντομη παρουσίαση της αρχιτεκτονικής και των χαρακτηριστικών του Secondo(κεφάλαιο 1) περνάμε στο σύστημα αναφοράς BerlinMOD(κεφάλαιο 2), αναλύουμε τον τρόπο με τον οποίο έχουν κατασκευαστεί τα δεδομένα και οι επερωτήσεις τις οποίες και παρουσιάζουμε καταλήγοντας στα αποτελέσματα/χρόνους που παίρνουμε από το συγκεκριμένο σύστημα αναφοράς.

ΣΧΕΤΙΚΕΣ ΕΡΓΑΣΙΕΣ

Τα κινούμενα αντικείμενα αποτελούν ουσιαστικά γεωμετρίες οι οποίες εξαρτώνται από τον χρόνο. Εκφράζουμε δηλαδή τις γεωμετρίες αυτές σαν συνάρτηση του χρόνου. Τα τελευταία χρόνια υπάρχουν δύο βασικές προσεγγίσεις στον τομέα των κινούμενων αντικειμένων. Η μία εξετάζει την τωρινή κίνηση των αντικειμένων και τις κινήσεις που προβλέπουμε ότι μπορεί να κάνουν στο μέλλον. Για να μοντελοποιήσουμε κινούμενα αντικείμενα αυτού του είδους έχουν προταθεί τα μοντέλα MOST(Moving Objects Spatio-Temporal) και FTL(Future Temporal Logic)[13,14,15,16]. Η δεύτερη προσέγγιση ασχολείται με το ιστορικό των κινήσεων των αντικειμένων αυτών. Το σύστημα αναφοράς BerlinMOD που θα δούμε αναλυτικά στην διατριβή αυτή εστιάζει στην δεύτερη προσέγγιση.

Επειδή η συλλογή πραγματικών δεδομένων για κινούμενα αντικείμενα είναι πρακτικά πολύ δύσκολη και απαιτητική από άποψη αποθήκευσης δεδομένων χρησιμοποιούνται αντί αυτού γεννήτριες που παράγουν αντίστοιχα δεδομένα. Μια από τις πρώτες γεννήτριες δεδομένων που φτιάχτηκαν, η *GSTD*[17], κατασκευάζει κινούμενα σημεία και κινούμενα ορθογώνια παραλληλόγραμμα χωρίς περιορισμούς. Η γεννήτρια *G-TERD*[18] είναι μια γεννήτρια που παράγει μεγάλα σύνολα αντικειμένων που συνεχώς κινούνται και αλλάζουν μεγέθη σε διαστάσεις περιοχές. Τα αντικείμενα αυτά δεν είναι απλώς σημεία, αλλά ορίζουν περιοχές που έχουν

σχήμα, μέγεθος και χρώμα για την απεικόνιση των διαφορετικών ιδιοτήτων που έχουν μεταξύ τους. Η γεννήτρια αυτή θα μπορούσε εύκολα να χρησιμοποιηθεί για την παρακολούθηση αντικειμένων. Όμως κι εδώ η κίνηση είναι χωρίς περιορισμούς, οπότε δεν εξυπηρετεί στην παρατήρηση αντικειμένων κινούμενων πάνω σε δίκτυο. Μια ακόμη γεννήτρια που έχει κατασκευαστεί ονομάζεται *Oporto*[19] και προσομοιώνει το σενάριο ψαρέματος με βάρκα. Τα κοπάδια των ψαριών κινούνται με κατεύθυνση κυμαινόμενα σημεία του πλαγκτόν. Στα πλαίσια της παρούσας διατριβής θα ασχοληθούμε με την γεννήτρια παραγωγής χώρο-χρονικών δεδομένων και δεδομένων τροχιάς και κίνησης του BerlinMOD.

Συζητώντας για Χωρικά Συστήματα Διαχείρισης Βάσεων Δεδομένων – SDBMS το πλέον διακεκριμένο σύστημα αναφοράς είναι το *Sequoia 2000 storage benchmark*. [20] Αποτελείται από πραγματικά χωρικά δεδομένα και από μια σειρά από ερωτήματα για τον έλεγχο της απόδοσης ενός SDBMS. Παρόλο που κάποιες από τις επερωτήσεις περιλαμβάνουν χρονική επιλογή και διαλογή το σύστημα Sequoia δεν χειρίζεται πραγματικά χώρο-χρονικά δεδομένα. Στο [21] ο Θεοδωρίδης προτείνει ένα σχήμα βάσης και ένα σύνολο από 10 επερωτήσεις σχετικά με την υποστήριξη των LBS (Location Based Services). Υπάρχουν επιπλέον συστήματα αναφοράς όπως το DynaMark και το COST Benchmark που παρουσιάζονται στα [22] ,[23] και χρησιμοποιούν διαφορετικές προσεγγίσεις στον τύπο των επερωτήσεων που προτείνουν.

ΚΕΦΑΛΑΙΟ 1: Αρχιτεκτονική και Χαρακτηριστικά Secondo DBMS

Το SECONDO είναι ένα επεκτάσιμο σύστημα διαχείρισης βάσεων δεδομένων το οποίο υποστηρίζει κυρίως μη τυποποιημένες εφαρμογές. Έχει σχεδιαστεί στο πανεπιστήμιο του HAGEN στη Γερμανία. Αποτελεί μια πλατφόρμα που μας δίνει τη δυνατότητα να υλοποιούμε και να πειραματιζόμαστε με διάφορα μοντέλα δεδομένων, δεν περιλαμβάνει κάποιο σταθερό μοντέλο και είναι ανοιχτό στην δημιουργία νέων.

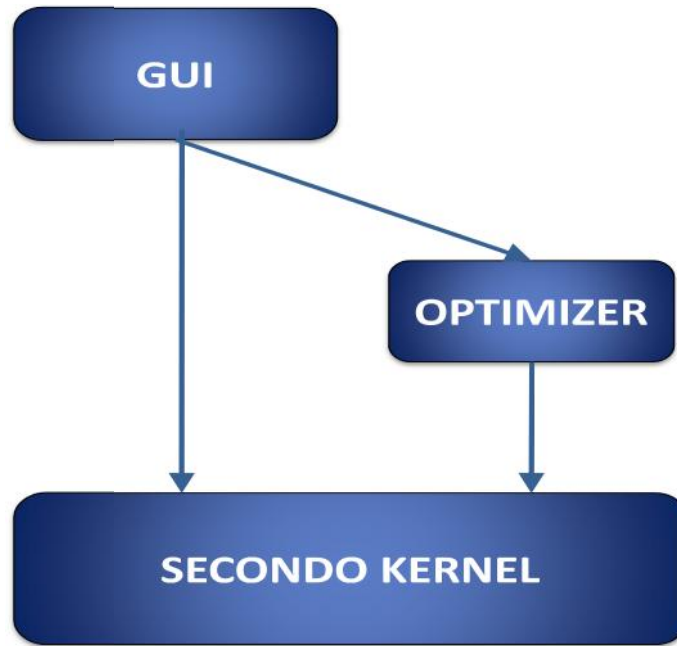
Σκοπός του είναι να μας παρέχει ένα γενικό πλαίσιο συστήματος διαχείρισης βάσεων δεδομένων στο οποίο μπορούμε να εισάγουμε μοντέλα δεδομένων για συστήματα κινητών αντικειμένων. Θα μπορούσαμε δηλαδή να υλοποιήσουμε σχεσιακά, αντικειμενοστραφή, χρονικά ή και XML μοντέλα ώστε να ενσωματώσουμε τύπους δεδομένων για χωρικά δεδομένα, κινούμενα αντικείμενα κ.α.

Για να πετύχουμε τα παραπάνω, διαχωρίζουμε τα μέρη και τους μηχανισμούς που είναι ανεξάρτητα του μοντέλου δεδομένων του συστήματος βάσεων δεδομένων από εκείνα που εξαρτώνται από το μοντέλο. Δομούμε την υλοποίηση του μοντέλου δεδομένων σε ένα σύνολο αλγεβρικών ενοτήτων (οντοτήτων) κάθε μια από τις οποίες παρέχει συγκεκριμένες δομές δεδομένων και λειτουργίες.

Η κατασκευή του συστήματος αυτού έχει περάσει από διάφορα στάδια τα οποία έχουν οδηγήσει στην τωρινή έκδοση η οποία χρησιμοποιεί την βάση BerkeleyDB, τρέχει στα λειτουργικά συστήματα Windows, Linux, Solaris και αποτελείται από τρία κύρια μέρη τα οποία είναι γραμμένα σε διαφορετικές γλώσσες προγραμματισμού. Το SECONDO KERNEL, τον OPTIMIZER και γραφικό προβάλλον του χρήστη (GUI) .

Γενικά κατά την λειτουργία του SECONDO απαιτείται μια συνεχής επικοινωνία και συνεργασία των τριών αυτών μερών. (σχήμα 1.1)

- Το GUI στέλνει εκτελέσιμες επερωτήσεις (πλάνο επερωτήσεων) στον πυρήνα και εμφανίζει τα αποτελέσματα.
- Το GUI στέλνει επερωτήσεις στον OPTIMIZER, λαμβάνει το πλάνο, το στέλνει στον πυρήνα και εμφανίζει το αποτέλεσμα.
- Ο OPTIMIZER στέλνει εντολές και εκτελέσιμες επερωτήσεις στον πυρήνα για να πάρει πληροφορίες για τα αντικείμενα της βάσης.

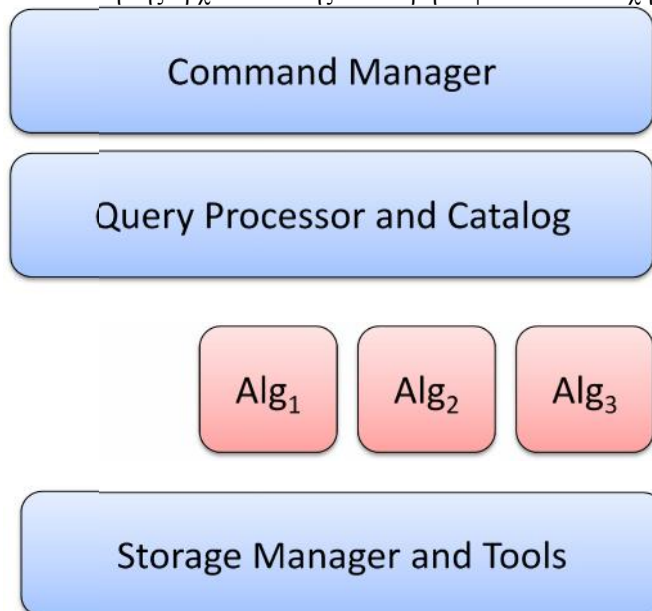


Εικόνα 1: Η διεπαφή χρήστη του SECONDO

1.1 SECONDO KERNEL (πυρήνας)

Ο πυρήνας υλοποιεί συγκεκριμένα μοντέλα δεδομένων, είναι επεκτάσιμος με βάση αλγεβρικές ενότητες, μπορεί να παρέχει επεξεργασία ερωτημάτων πάνω στις ήδη υλοποιημένες άλγεβρες, είναι γραμμένος σε C++ και έχει υλοποιηθεί ένα επίπεδο πάνω από την BerkeleyDB.

Μια απλή απεικόνιση της αρχιτεκτονικής του πυρήνα φαίνεται στο σχήμα 1.2



Εικόνα 2: Απεικόνιση της αρχιτεκτονικής του πυρήνα

Ένα μοντέλο δεδομένων υλοποιείται ως ένα σύνολο τύπων δεδομένων και λειτουργιών οι οποίες είναι ομαδοποιημένες σε άλγεβρες. Μια αλγεβρική ενότητα παρέχει μια συλλογή τύπων δεδομένων και κατασκευάζει μια δομή για καθένα από αυτούς χρησιμοποιώντας ένα σύνολο κατάλληλων συναρτήσεων. Ο επεξεργαστής επερωτήσεων αξιολογεί τις επερωτήσεις δημιουργώντας ένα δέντρο λειτουργιών και διασχίζοντας το από κάτω προς τα πάνω, εκτελεί τους κατάλληλους τελεστές της άλγεβρας.

Ο πυρήνας διαχειρίζεται τις βάσεις, οι οποίες είναι ένα σύνολο από αντικείμενα του SECONDO. Ένα αντικείμενο SECONDO είναι μια τριάδα της μορφής (όνομα, τύπος, τιμή), όπου ο τύπος αντιστοιχεί σε ένα τύπο της υλοποιημένης άλγεβρας και τιμή η αντίστοιχη τιμή του.

Όσον αφορά τις βάσεις υπάρχει ένα σύνολο λειτουργιών για τη διαχείρισή τους : δημιουργία, διαγραφή, άνοιγμα, κλείσιμο, εξαγωγή σε, εισαγωγή από αρχείο. Αυτή τη στιγμή υπάρχουν περίπου 20 άλγεβρες που χρησιμοποιεί το SECONDO. Καθεμία από αυτές περιλαμβάνει συγκεκριμένες λειτουργίες. Για παράδειγμα:

- StandardAlgebra: παρέχει τύπους δεδομένων int, Bool, string, real.
- RelationAlgebra: σχετίζεται με όλες τις λειτουργίες που απαιτούνται για την υλοποίηση μιας σχεσιακής γλώσσας όπως η SQL.
- BTreeAlgebra: B-δέντρα.
- RTreeAlgebra: R-δέντρα.
- SpatialAlgebra: χωρικοί τύποι δεδομένων, σημείο, σημεία, γραμμή, περιοχή.
- DateAlgebra: μικρή άλγεβρα που παρέχει ένα τύπο για τα δεδομένα ημερομηνίας.

1.2 OPTIMIZER

Καθώς το SECONDO είναι ένα επεκτάσιμο σύστημα διαχείρισης βάσεων δεδομένων, κατάλληλο για διδασκαλία και έρευνα, θα πρέπει να μπορούμε να προσθέσουμε σε αυτό άλγεβρες , όπως χωρικές άλγεβρες ή άλγεβρα για κινούμενα αντικείμενα, χωρίς σημαντική προσπάθεια. Επομένως προκύπτουν οι ακόλουθες απαιτήσεις από τον OPTIMIZER τις οποίες και εκπληρώνει:

- Να βρίσκει τα βέλτιστα πλάνα για την εκτέλεση των επερωτήσεων γρήγορα
- Να δουλεύει καλά για μη προτύπους τύπους δεδομένων
- Η υλοποίηση των μη προτύπων τύπων δεδομένων να είναι μικρή και να μην απαιτεί πολύ προσπάθεια
- Οι μηχανισμοί επέκτασης να είναι απλοί και κατανοητοί
- Ο αλγόριθμος βελτιστοποίησης να είναι απλός και ξεκάθαρος και ο κώδικας του OPTIMIZER προσπελάσιμος και τεκμηριωμένος, για την χρήση του ως διδακτικό εργαλείο

Ο OPTIMIZER είναι γραμμένος σε γλώσσα προγραμματισμού PROLOG σε περιβάλλον προγραμματισμού SWI-PROLOG το οποίο υποστηρίζει και τα λειτουργικά συστήματα Windows, Linux, Solaris.

Ο OPTIMIZER διασπά το γενικό πρόβλημα σε τρία κομμάτια, παίρνει ένα σύνολο από σχέσεις και επιλογές και ενώσεις κατηγορημάτων παράγοντας ένα εκτελέσιμο πλάνο για το SECONDO.

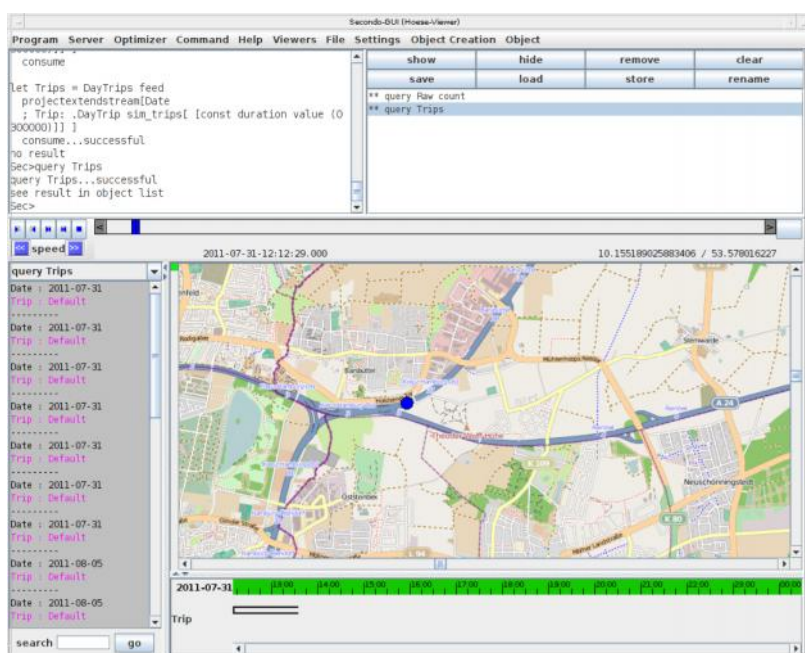
Ο OPTIMIZER δουλεύει αυστηρά ως μια ανεξάρτητη ενότητα. Έχει πρόσβαση στον πυρήνα μόνο μέσω της πρότυπης διεπαφής, όπως και οποιαδήποτε άλλη εφαρμογή. Έτσι η ενότητα του OPTIMIZER είναι δυνατόν να μεταφερθεί και να χρησιμοποιηθεί και σε άλλα περιβάλλοντα στα οποία απαιτείται η επεξεργασία επερωτήσεων.

1.3 ΓΡΑΦΙΚΟ ΠΕΡΙΒΑΛΛΟΝ ΧΡΗΣΤΗ (GUI)

Εκτός από τη διεπαφή εκτέλεσης εντολών το SECONDO περιλαμβάνει και τη γραφική διεπαφή χρήστη. Η διεπαφή είναι εξολοκλήρου γραμμένη σε JAVA για αυτό και μπορεί να χρησιμοποιηθεί σε πολλές διαφορετικές πλατφόρμες. Επικοινωνεί με το SECONDO μέσω του TCP/IP , χρησιμοποιώντας εμφωλευμένες λίστες για τη μεταφορά αντικειμένων.

Το GUI μπορεί να χρησιμοποιηθεί εκτός σύνδεσης, ανεξάρτητα από το SECONDO, για να απεικονίσει τους τύπους δεδομένων που μπορούν να παρουσιαστούν από τις διαθέσιμες απεικονίσεις (**viewers**).

Η διεπαφή χρησιμοποιεί τους τύπους των διαφόρων απεικονίσεων για να εμφανίσει αντικείμενα. Το GUI περιλαμβάνει μια περιοχή εντολών (πάνω αριστερά), τον διαχειριστή αντικειμένων (πάνω αριστερά) και την περιοχή των απεικονίσεων (κάτω)(εικόνα 1.3). Από την περιοχή εντολών εισάγουμε επερωτήσεις για το SECONDO. Επίσης μπορούμε να χρησιμοποιήσουμε την περιοχή εντολών για να δίνουμε εντολές στο ίδιο το GUI, χρησιμοποιώντας τη λέξη GUI πριν από κάθε εντολή.



Εικόνα 3: Γραφικό Περιβάλλον Χρήστη - GUI

Το GUI προαιρετικά μπορεί να επικοινωνήσει με έναν optimizer server. Αν μια επερώτηση ξεκινάει με την λέξη select ή sql ο optimizer προσπαθεί να βρει το βέλτιστο πλάνο για αυτήν την επερώτηση. Έπειτα, στέλνει το πλάνο στο SECONDO για εκτέλεση.

Ο διαχειριστής αντικειμένων περιέχει όλα τα αποτελέσματα των επερωτήσεων που έχουν εκτελεστεί με επιτυχία, καθώς και τα αντικείμενα που έχουν φορτωθεί από αρχεία. Τα αντικείμενα αυτά μπορούν να αποσυρθούν από την απεικόνιση προσωρινά ή μόνιμα και να τα αποθηκεύσουμε σε αρχείο ή στη βάση που έχουμε ήδη ανοίξει. Η διεπαφή μπορεί να εισάγει αντικείμενα από πολλούς τύπους αρχείων. Με αυτό τον τρόπο καταφέρνουμε να δημιουργήσουμε καινούρια αντικείμενα για τη βάση.

Η τωρινή έκδοση του προγράμματος έχει διαθέσιμες πάνω από τέσσερις απεικονίσεις, οι οποίες μας επιτρέπουν να μπορούμε να εμφανίσουμε μια μεγάλη γκάμα διαφορετικών τύπων δεδομένων.

Όπως βλέπουμε και στο σχήμα 1.2 φαίνεται η απεικόνιση Hoese Viewer (ονομάστηκε έτσι από τον δημιουργό της). Η απεικόνιση αυτή μπορεί να αναπαριστά σχέσεις μεταξύ χωρικών ή χώρο-χρονικών αντικειμένων, κινούμενων αντικειμένων και έχει εξελιγμένη λειτουργικότητα. Μερικά από τα χαρακτηριστικά της απεικόνισης είναι :

- Εστίαση, κύλιση, αναζήτηση κειμένου
- Ρύθμιση γνωρισμάτων αποδοτικότητας, πχ χρώμα, διαφάνεια
- Επιλογή: όταν ένα γραφικό αντικείμενο επιλέγεται με κλικ του ποντικιού, το αντίστοιχο κείμενο μαρκάρεται και αντιστρόφως.

1.4 Διεπαφές SECONDO

Το second έχει πέντε διαφορετικές διεπαφές χρήστη,

1. SecondoTTYBDB – είναι μία απλή διεπαφή κειμένου για έναν χρήστη υλοποιημένη σε C++. Συνδέεται απευθείας με το σύστημα και χρησιμοποιείται κυρίως για εκσφαλμάτωση και δοκιμές του συστήματος χωρίς να απαιτείται επικοινωνία πελάτη-εξυπηρετητή.
2. SecondoTTYCS – διεπαφή πολλαπλών χρηστών με επικοινωνία πελάτη-εξυπηρετητή
3. SecondoPL – είναι το αντίστοιχο του Optimizer αλλά για έναν χρήστη
4. SecondoPLCS - διεπαφή πολλαπλών χρηστών με επικοινωνία πελάτη-εξυπηρετητή
5. Javagui - διεπαφή πολλαπλών χρηστών με επικοινωνία πελάτη-εξυπηρετητή

Τα SecondoTTYCS, SecondoPLCS και Javagui ανταλλάσσουν μηνύματα με το κεντρικό σύστημα καθώς τρέχουν μια διεργασία εξυπηρετητή μέσω του πρωτοκόλλου TCP/IP. Εφόσον λοιπόν η διεργασία του εξυπηρετητή της βάσης έχει ξεκινήσει πολλοί πελάτες διεπαφών χρηστών μπορούν να συνδεθούν στην βάση του SECONDO ταυτόχρονα.

ΚΕΦΑΛΑΙΟ 2 : BerlinMOD

Τα περισσότερα συστήματα βάσης δεδομένων μπορούν εν γένει να αποθηκεύουν μεγάλες ποσότητες δεδομένων. Εκτός από τα συνηθισμένα δεδομένα μπορούν επίσης να αποθηκεύουν και χωρικά δεδομένα, χώρο-χρονικά καθώς και πολυμέσων. Ενώ όμως ο χειρισμός και η πρόσβαση στα τυπικά δεδομένα είναι ευρέως γνωστή, η αποθήκευση και η αποτελεσματική επεξεργασία στα μη τυπικά δεδομένα αποτελεί πρόκληση. Για να μπορούμε, λοιπόν, να συγκρίνουμε διαφορετικά συστήματα διαχείρισης βάσεων δεδομένων και τις μεθόδους αποθήκευσης και πρόσβασης σε αυτά χρησιμοποιούμε συστήματα αναφοράς (benchmark).

Γενικά ένα σύστημα αναφοράς αποτελείται από καλά προσδιορισμένα (διαβαθμισμένα) σύνολα δεδομένων και προβλημάτων. Σε ότι αφορά τα συστήματα βάσεων δεδομένων έχουν κυρίως τη μορφή επερωτήσεων – SQL. Τα συστήματα αναφοράς έχουν αποδειχθεί ως ένα αποτελεσματικό εργαλείο για την μελέτη της απόδοσης συστημάτων διαχείρισης βάσεων δεδομένων. Τα συστήματα αναφοράς απλοποιούν τη δημιουργία και την περιγραφή διαφόρων προγραμμάτων επειδή καθένας μπορεί να κάνει αναφορά σε ένα καλά ορισμένο σύνολο δεδομένων οι ιδιότητες του οποίου έχουν περιγραφεί με λεπτομέρεια σε κάποιο άλλο σύστημα. Ακόμα, με αυτό τον τρόπο μειώνουμε την εμφάνιση προκαταλήψεων στα πειράματά

μας. Το σύστημα αναφοράς BerlinMOD αναφέρεται σε βάσεις κινητών αντικειμένων MOD οι οποίες αναπαριστούν ένα πλήρες ιστορικό κινήσεων επιτρέποντάς μας να κάνουμε ανάλυση αυτών των κινήσεων στο παρελθόν. Το BerlinMOD καλύπτει ένα μεγάλο εύρος επερωτήσεων το οποίο μπορεί να μας φανερώσει όχι μόνο τα δυνατά αλλά και τα αδύναμα σημεία σε υπάρχουσες λύσεις υποδεικνύοντας σε πιο τομέα πρέπει να δουλέψουμε περισσότερο.

Είναι προφανές, ότι το καλύτερο σύνολο δεδομένων είναι τα πραγματικά δεδομένα. Η προσπάθεια για να συλλέξουμε πραγματικά δεδομένα τις περισσότερες φορές καθίσταται δύσκολη και μπορεί να προκαλέσει προβλήματα. Για παράδειγμα, στην περίπτωση που θέλουμε να παρακολουθήσουμε 2000 οχήματα την ίδια στιγμή καθένα από αυτά θα πρέπει να έχει μια συσκευή απεικόνισης θέσης (GPS) και θα πρέπει όλα τα δεδομένα να καταγραφούν. Ένα ακόμα πρόβλημα των πραγματικών δεδομένων είναι ότι δεν είναι επεκτάσιμα. Μια καλή λύση για την αντιμετώπιση αυτών των προβλημάτων είναι οι γεννήτριες δεδομένων. Μπορούμε να προσαρμόσουμε εμείς το μέγεθος της παραγωγής δεδομένων απλά αλλάζοντας κάποιες παραμέτρους.

Το σύστημα αναφοράς BerlinMOD διαχειρίζεται δεδομένα κινητών σημείων και εκτός από ένα σύνολο επερωτήσεων παρέχει επίσης ένα εργαλείο που δημιουργεί αυτά τα δεδομένα. Για να παραχθούν τα συγκεκριμένα δεδομένα προσομοιώνουμε την κίνηση ενός αριθμού αμαζιών στο οδικό δίκτυο του Βερολίνου μια συγκεκριμένη χρονική περίοδο καταγράφοντας τη θέση τους τουλάχιστον κάθε δύο δευτερόλεπτα. Η πρωτοτυπία σε αυτή την περίπτωση αφορά στο ότι δεν χρησιμοποιείται ένα ανεξάρτητο εργαλείο παραγωγής δεδομένων αλλά το SECONDO. Τα δεδομένα του συστήματος αναφοράς δημιουργούνται από ένα σύνολο εντολών του SECONDO.

Το σενάριο βάση του οποίου έχει δημιουργηθεί το σύστημα αναφοράς BerlinMOD χρησιμοποιεί τις θέσεις κινούμενων αντικειμένων (οχήματα) οι οποίες κατασκευάζονται με βάση τη ν υπόθεση ότι έχουμε μια αναπαράσταση σε γράφημα του οδικού δικτύου. Οι κόμβοι αναπαριστούν διασταυρώσεις και αδιέξοδα του δρόμου, ενώ οι ακμές που ενώνουν τους κόμβους είναι τα κομμάτια των δρόμων αυτών. Ένας τύπος αντικείμενου γραμμή περιγράφει την δισδιάστατη γεωμετρία της κάθε ακμής, κάθε ακμή ξεχωρίζει από μία τιμή η οποία είναι ανάλογη με τον χρόνο που χρειάζεται για να τη διασχίσουμε με την μέγιστη επιτρεπόμενη ταχύτητα και με ένα προσδιορισμό για τον δρόμο. Οι κόμβοι ξεχωρίζουν με βάση έναν ατομικό αναγνωριστικό και με την χωρική δισδιάστατη θέση τους.

Κάθε ένα αντικείμενο έχει κόμβο σπιτιού, κόμβο εργασίας και την γειτονιά του. Ο κόμβος σπιτιού αναπαριστά την οικία του ιδιοκτήτη του αντικείμενου, ο κόμβος εργασίας το μέρος όπου δουλεύει και η γειτονιά είναι το σύνολο των κόμβων που απέχουν απόσταση μικρότερη από 3 χλμ. Του κόμβου σπιτιού. Ο σκοπός αυτής της υπόθεσης είναι να αναπαρασταθεί η φυσιολογική συμπεριφορά κάποιου ατόμου, σαν να είχαμε πραγματικά δεδομένα.

2.1 Μοντέλα βάσεων δεδομένων

Το BerlinMOD χρησιμοποιεί δύο προσεγγίσεις σχετικά με τους τύπους δεδομένων κινητών αντικειμένων: 1) βασισμένους στα αντικείμενα , 2) βασισμένους στις διαδρομές.

Στην πρώτη περίπτωση κρατάμε ολόκληρο το ιστορικό ενωμένο, στην δεύτερη περίπτωση όμως καταγράφεται η κίνηση των αντικειμένων και αποθηκεύεται ως μία ακολουθία ξεχωριστών διαδρομών τα οποία αποθηκεύονται σε μια πρόσθετη σχέση. Η πινακίδα κυκλοφορίας κάθε οχήματος χρησιμοποιείται σαν αναφορά από την σχέση της βάσης στην σχέση η οποία περιέχει τις διαδρομές και το αντίστροφο.

Το μέγεθος των χώρο-χρονικών δεδομένων που δημιουργούνται καθορίζεται από την παράμετρο SCALEFACTOR. Η παράμετρος αυτή είναι ουσιαστικά μια κλίμακα των δεδομένων που θα παραχθούν. Ανάλογα με την τιμή που θα της δώσουμε θα πάρουμε αντίστοιχα συγκεκριμένο αριθμό οχημάτων και ανάλογο αριθμό ημερών που θα γίνεται καταγραφή των κινήσεων των οχημάτων. Αν η τιμή της παραμέτρου αυτής είναι 1.0 τότε θα έχουμε δεδομένα για 2000 οχήματα σε διάστημα 28 ημερών ξεκινώντας από μια Δευτέρα.

2.2 Περιγραφή των αντικειμένων που χρησιμοποιούνται στη βάση

Τα αντικείμενα που χρησιμοποιούνται στη βάση του BerlinMOD διαχωρίζονται σε αυτά που χρησιμοποιούνται και για τις δύο προσεγγίσεις (OBA, TBA, βασισμένη στα αντικείμενα και βασισμένη στις διαδρομές) και σε αυτά που χρησιμοποιούνται σε κάποια από τις δύο ξεχωριστά.

Αντικείμενα που χρησιμοποιούνται και στις δύο προσεγγίσεις:

- ✓ Κόμβοι - Nodes: relation{NodeId: int , Pos: point} — μία σχέση για όλους τους κόμβους
- ✓ Αντικείμενο - QueryPoints: relation{Id: int , Pos: point} — σχέση για το αντικείμενο QueryPoints με κλειδί Id
- ✓ Αντικείμενο - QueryRegions: relation{Id: int , Region: region} — το αντικείμενο QueryRegions με κλειδί Id. Οι περιοχές regions είναι κανονικά πολύγωνα με κέντρο p και ύψος h.
- ✓ Αντικείμενο - QueryInstants: relation{Id: int , Instant : instant} — το αντικείμενο QueryInstants με κλειδί Id. Οι χρονικές που αναπαριστά το αντικείμενο είναι ίσα καταναμημένες καθόλη την διάρκεια της περιόδου καταγραφής.
- ✓ Αντικείμενο - QueryPeriods: relation{Id: int , Period: periods}— το αντικείμενο QueryPeriods με κλειδί Id.
- ✓ Αντικείμενο - QueryLicences: relation{Id: int , Licence: string} — το αντικείμενο QueryLicences με κλειδί Id, είναι οι άδειες κυκλοφορίας των οχημάτων (πινακίδες).

Προσέγγιση με βάση τα αντικείμενα:

- ✓ Σχέση - dataScar: relation{Licence: string, Model : string, Type: string, Trip: mpoint} — μία σχέση που περιγράφει τις ιδιότητες κάθε οχήματος (τύπος, μοντέλο και αριθμός κυκλοφορίας), καθώς και το πλήρες ιστορικό κινήσεων του οχήματος σαν μια τιμή mpoint με κλειδί τον αριθμό κυκλοφορίας.

Προσέγγιση με βάση τις διαδρομές:

- ✓ Σχέση - dataMcar: relation{Licence: string, Model : string, Type: string} — μία σχέση που περιγράφει τις ιδιότητες κάθε οχήματος χωρίς όμως να περιέχει το ιστορικό των κινήσεων του.
- ✓ Σχέση - dataMtrip: relation{Licence: string, Trip: mpoint} — μία σχέση που περιλαμβάνει το ιστορικό των κινήσεων του κάθε οχήματος
- ✓ Εδώ το γνώρισμα, {Licence} είναι ξένο κλειδί για το αντικείμενο dataMcar και το γνώρισμα {Licence, Trip} είναι κλειδί για το αντικείμενο dataMtrip.

2.3 Επιλογή/Παρουσίαση των Επερωτήσεων

Τις επερωτήσεις τις οποίες θα χρησιμοποιήσει το σύστημα αναφοράς BerlinMOD τις χωρίζουμε σε δύο σετ. Στις BerlinMOD/R(range type) οι οποίες είναι επερωτήσεις που έχουν να κάνουν με την απόσταση και στις BerlinMOD/NN(nearest neighbor) οι οποίες αφορούν σε επερωτήσεις ‘πλησιέστερου γείτονα’ και έχουν να κάνουν με βάσεις που ασχολούνται με το ιστορικό κινούμενων αντικειμένων. Αυτός ο διαχωρισμός πρέπει να γίνει καθώς για τις επερωτήσεις BerlinMOD/NN δεν υπάρχει κάποιος κοινά αποδεκτός τρόπος να τις μετατρέψουμε σε SQL.

2.3.1 BerlinMOD/R Επερωτήσεις

Για να μπορέσει να γίνει η επιλογή και ο διαχωρισμός των κατάλληλων επερωτήσεων για το BerlinMOD ξεχωρίζουμε πέντε πτυχές/ιδιότητες που μπορούν να έχουν οι επερωτήσεις:

1. Ταυτότητα αντικείμενου(γνωστό, άγνωστο) – εδώ έχουμε επερωτήσεις όπου είτε αναφερόμαστε σε κάποιο γνωστό αντικείμενο ή δεν γνωρίζουμε από πριν για ποιο αντικείμενο κάνουμε την επερωτηση.
2. Διάσταση(τυπική, χωρική, χρονική, χώρο-χρονική)
3. Query interval
4. Τύπος κατάστασης(μονό αντικείμενο, σχέσεις μεταξύ αντικειμένων)
5. Σύνολο

2.3.2 Παρουσίαση των Επερωτήσεων

Παρακάτω ακολουθεί περιγραφή των επερωτήσεων που έχουν κατασκευαστεί, τα αποτελέσματα των οποίων ουσιαστικά αποτελούν το κριτήριο αξιολόγησης του εκάστοτε συστήματος όπου θα εφαρμόσουμε το σύστημα αναφοράς BerlinMOD.

Η παρουσίαση των επερωτήσεων γίνεται πρώτα σε κοινή γλώσσα και έπειτα ακολουθεί μία μορφοποίηση τύπου SQL.

Επερώτηση 1

Ποια είναι τα μοντέλα των οχημάτων με αριθμούς κυκλοφορίας από το αντικείμενο QueryLicences?

```
SELECT DISTINCT LL. Licence AS Licence , C.Model AS  
Model
```

```
FROM dataScar C, QueryLicences LL
```

```
WHERE C. Licence = LL. Licence ;
```

Η επερώτηση αυτή τεστάρει την επίδοση σε καθορισμένους τύπους και ευρετήρια. Ένα ευρετήριο με βάση τον αριθμό κυκλοφορίας μπορεί να είναι χρήσιμο. Κάποια ΣΔΒΔ μπορεί ζητήσουν πρόσβαση σε δεδομένα διαδρομών ενώ δεν χρειάζεται και αυτό μπορεί να οδηγήσει σε απώλειες απόδοσης. Κάτι τέτοιο μπορεί να αποφευχθεί με την προσέγγιση βασισμένη στις διαδρομές.

Επερώτηση 2

Πόσα οχήματα υπάρχουν τα οποία είναι τύπου “επιβατικά”?

```
SELECT COUNT ( License )
```

```
FROM dataScar
```

```
WHERE Type =“passenger”;
```

Εδώ τεστάρουμε πάλι καθορισμένους τύπους έχοντας όμως ένα πιο επιλεκτικό κατηγορήμα και χρησιμοποιώντας και το τελικό άθροισμα.

Επερώτηση 3

Πού βρίσκονταν τα οχήματα με αριθμό κυκλοφορίας που ανήκει στο αντικείμενο QueryLicences1 τις χρονικές στιγμές που ανήκουν στο αντικείμενο QueryInstants1?

```
SELECT LL. Licence AS Licence , II. Instant AS Instant
,
val (C. Trip atinstant II. Instant ) AS Pos
FROM dataScar C, QueryLicences1 LL, Query Instants1 II
WHERE C. Licence = LL. Licence AND C. Trip present II.
Instant ;
```

Επερώτηση 4

Ποιοι αριθμοί κυκλοφορίας ανήκουν σε οχήματα τα οποία έχουν περάσει από τα σημεία που ανήκουν στο αντικείμενο QueryPoints?

```
SELECT PP. Pos AS Pos, C. Licence AS Licence
FROM dataScar C, QueryPoints PP
WHERE C. Trip passes PP. Pos ;
```

Στην προσέγγιση με βάση τις διαδρομές ο αριθμός κυκλοφορίας δεν είναι πρωτεύων κλειδί, για αυτό τον λόγο χρησιμοποιούμε την παρακάτω εντολή

```
SELECT DISTINCT
```

Επερώτηση 5

Ποια είναι η ελάχιστη απόσταση μεταξύ τοποθεσιών, όπου ένα όχημα με αριθμό κυκλοφορίας από το αντικείμενο QueryLicences1 και ένα όχημα με αριθμό κυκλοφορίας από το αντικείμενο QueryLicences2 έχουν περάσει?

```
SELECT LL1 . Licence AS Licence1 , LL2 . Licence AS
Licence2 ,
distance ( trajectory (V1 . Trip ) , trajectory (V2 .
Trip ) )
AS Dist
FROM dataScar V1 , dataScar V2 , QueryLicences1 LL1 ,
QueryLicences2 LL2
WHERE V1 . Licence = LL1 . Licence AND V2 . Licence =
LL2 . Licence
AND V1 . Licence <> V2. Licence ;
```

Στην προσέγγιση με βάση τα αντικείμενα δεν χρειάζεται να κάνουμε κάποιο άθροισμα αφού η καθολική απόσταση υπολογίζεται χρησιμοποιώντας έναν συνδυασμό τροχιάς και απόστασης.

Για την προσέγγιση που βασίζεται στις διαδρομές όμως χρειαζόμαστε ένα καθολικό άθροισμα:

```
SELECT LL1 . Licence AS Licence1 , LL2 . Licence AS
Licence2 ,
MIN ( distance ( trajectory (T1 . Trip ) , trajectory
(T2 . Trip ) ) )
AS Dist ,
FROM dataMtrip T1 , dataMtrip T2 , QueryLicences1 LL1 ,
QueryLicences2 LL2
WHERE T1 . Licence = LL1 . Licence AND T2 . Licence =
LL2 . Licence
```

```

AND T1 . Licence <> T2 . Licence
GROUP BY LL1 . Licence , LL2 . Licence ;

```

Είναι προφανές εδώ ότι τα ευρητήρια στον αριθμό κυκλοφορίας είναι πολύ χρήσιμα. Σε αυτό το σημείο θα εμφανιστούν σημαντικές διαφορές όταν θα εφαρμόσουμε την χωρική απόσταση και την τροχιά σε δεδομένα διαφορετικού μεγέθους.

Επερώτηση 6

Ποια είναι τα ζευγάρια αριθμών κυκλοφορίας και οχημάτων "φορτηγών" τα οποία έχουν βρεθεί σε απόσταση 10 μέτρων μεταξύ τους?

```

SELECT V1 . Licence AS Licence1 , V2 . Licence AS
Licence2
FROM dataScar V1 , dataScar V2
WHERE V1 . Licence < V2 . Licence AND V1 . Type = "
truck "
AND V2 . Type = " truck "
AND sometimes ( distance ( V1 . Trip , V2 . Trip ) <= 1 0
. 0 ) ;

```

Για την προσέγγιση βασισμένη στις διαδρομές:

```

SELECT DISTINCT T1 . Licence AS Licence1 , T2 . Licence
AS Licence2
FROM dataMtrip T1 , dataMcar C1 , dataMtrip T2 ,
dataMcar C2
WHERE T1 . Licence < T2 . Licence AND T1 . Licence = C1
. Licence
AND T2 . Licence = C2 . Licence AND C1 . Type = " truck
"
AND C2 . Type = " truck "
AND sometimes ( distance ( T1 . Trip , T2 . Trip ) <= 1
0 . 0 ) ;

```

Ένα χωρικό ή χώρο-χρονικό ευρητήριο στις διαδρομές μπορεί να χρησιμοποιηθεί για να ξεχώσει κάποια από τα οχήματα. Αυτή η επερώτηση μπορεί να βοηθήσει στο να συγκρίνουμε την απόδοση διαφορετικών δομών ευρητηρίων.

Επερώτηση 7

Ποιοι είναι οι αριθμοί κυκλοφορίας οχημάτων τύπου "επιβατικά" τα οποία έχουν φτάσει στο σημεία από το αντικείμενο QueryPoints πρώτα από όλα τα "επιβατικά" οχήματα καθόλη την διάρκεια του πειράματος?

```

SELECT PP. Pos AS Pos , V1 . Licence AS Licence
FROM dataScar V1 , QueryPoints PP
WHERE V1 . Trip passes PP. Pos AND V1 . Type = "
passenger "
AND inst ( initial ( V1 . Trip at PP. Pos ) ) <= ALL
( SELECT inst ( initial ( V2 . Trip at PP2 . Pos ) ) AS
FirstTime
FROM dataScar V2

```

```

WHERE V2 . Trip passes PP. Pos AND V2 . Type = "
passenger "
) ;

```

Για την προσέγγιση βασισμένη στις διαδρομές:

```

SELECT DISTINCT V1 . Licence AS Licence , PP. Pos AS
Pos
FROM dataMtrip T1 , dataMcar C1 , QueryPoints1 PP
WHERE T1 . Trip passes PP. Pos
AND C1 . Type = " passenger " AND C1 . Licence = T1 .
Licence
AND inst ( initial ( T1 . Trip at PP. Pos ) ) <= ALL
( SELECT inst ( initial ( T2 . Trip at PP. Pos ) ) AS
FirstTime
FROM dataMtrip T2 , dataMcar C2
WHERE V2 . Trip passes PP. Pos
AND T2 . Licence = C2 . Licence
AND C2 . Type = " passenger "
) ;

```

Αυτή η επερώτηση είναι τυπικό παράδειγμα χρήσης χωρικού ευρετηρίου.

Επερώτηση 8

Ποιες είναι οι συνολικές αποστάσεις που έχουν διανύσει τα οχήματα με αριθμό κυκλοφορίας από το αντικείμενο QueryLicences1 κατά την διάρκεια των περιόδων που ανήκουν στο αντικείμενο QueryPeriods1?

```

SELECT V1 . Licence AS Licence , PP. Per iod AS Per iod
length ( V1 . Trip at period s PP. Peiod ) AS Dist
FROM dataScar V1 , QueryPeriods1 PP, QueryLicences1 LL
WHERE V1 . Licence = LL. Licence AND V1 . Trip present
PP. Period ;

```

Within the TBA, we need an aggregation to sum up the travelled distances.

```

SELECT V1 . Licence AS Licence , PP. Period AS Period ,
SUM ( length ( V1 . Trip at period s PP. Period ) ) AS
Dist
FROM dataMtrip V1 , QueryPeriods1 PP, QueryLicences1 LL
WHERE V1 . Licence = LL. Licence AND V1 . Trip present
PP. Period
GROUP BY V1 . Licence , PP. Period ;

```

Ένα χρονικό ευρετήριο στις διαδρομές θα ήταν πολύ χρήσιμο στην συγκεκριμένη προσέγγιση ενώ δεν θα είχε ιδιαίτερο νόημα στην προσέγγιση με βάση τα αντικείμενα.

Επερώτηση 9

Ποια είναι η μεγαλύτερη απόσταση που έχει διανύσει ένα όχημα κατά την διάρκεια μιας περιόδου η οποία ανήκει στο αντικείμενο QueryPeriods?

```

SELECT PP. Period AS Period ,
MAX ( length ( V1 . Trip at periods PP) ) AS Dist

```

```

FROM dataScar V1 , QueryPeriods PP
WHERE V1 . Trip present PP. Period
GROUP BY PP. Period ;

```

Και πάλι στην προσέγγιση με βάση τις διαδρομές θα πρέπει να υπολογίσουμε το άθροισμα όλων των επιμέρους διαδρομών, δημιουργώντας πρώτα μία όψη:

```

CREATE VIEW Distances AS
SELECT SUM ( length (V1 . Trip at periods PP) ) AS
Length ,
PP. Period AS Period , V1 . Li ence AS Licence
FROM dataMtrip V1 , QueryPeriods PP
WHERE V1 . Trip present PP. Period
GROUP BY PP. Period , V1 . Licence ;
SELECT Period , MAX ( Length ) AS Dist
FROM Distances
GROUP BY Period ;

```

Ένα χρονικό ευρετήριο στις διαδρομές μπορεί να χρησιμεύσει στην προσέγγιση με βάση τις διαδρομές για να διαλέξουμε κατάλληλες διαδρομές για το άθροισμα.

Επερώτηση 10

Πότε και πού τα οχήματα με αριθμό κυκλοφορίας που ανήκουν στο αντικείμενο QueryLicences1 συναντήθηκαν με άλλα οχήματα (απόσταση < 3 μ) και ποιοι είναι οι αριθμοί κυκλοφορίας των οχημάτων αυτών?

```

SELECT V1 . Licence AS QueryLicence , V2 . Licence AS
OtherLicence ,
(V1 . Trip at periods ( deftime ( ( distance (V1 . Trip
, V2 . Trip )
<= 3 . 0 ) at TRUE ) ) ) AS Pos
FROM dataScar V1 , dataScar V2 , QueryLicences1 LL
WHERE V1 . Licenc e = LL. Licence AND V2 . Licence <>
V1 . Licence
AND sometimes ( distance (V1 . Trip , V2. Trip ) <= 3 .
0 ) ;

```

Επειδή πρόκειται για μία περίπλοκη επερώτηση περιορίζουμε το σύνολο των αριθμών κυκλοφορίας στις 10, αφού πρέπει να υπολογίσουμε τις χρονικές αποστάσεις σε σχέση με τα υπόλοιπα οχήματα

Για την προσέγγιση βασισμένη στις διαδρομές πρέπει να κάνουμε ομαδοποίηση με βάση τους αριθμούς κυκλοφορίας και να υπολογίσουμε το άθροισμα των ενδιάμεσων αποτελεσμάτων σε μία τιμή mpoint ανά αντικείμενο QueryLicence:

```

SELECT V1 . Licence AS Licence , V2 . Licence AS
OtherLicence ,
AGGR ( concat , V1 . Trip at periods ( deftime ( (
distance (
V1. Trip , V2 . Trip ) <= 3 . 0 ) at TRUE ) ) ,
emptympoint )
AS Pos

```

```

FROM dataMtrip V1 , dataMtrip V2 , QueryLicences1 LL
WHERE V1 . Licence = LL. Licence AND V2 . Licence <> V1
. Licence
AND sometimes ( distance (V1 . Trip , V2. Trip ) <= 3 .
0 )
GROUP BY V1 . Licence , V2 . Licence ;

```

Επερώτηση 11

Ποια οχήματα πέρασαν από σημείο που ανήκει το αντικείμενο QueryPoints1 κάποια χρονική στιγμή από το αντικείμενο QueryInstants1?

```

SELECT C. Licence AS Licence , PP. Pos AS Pos ,
II. Instant AS Instant
FROM dataScar C, QueryPoints1 PP, QueryInstant s1 II
WHERE val (C. Trip at instant II. Instant ) = PP. Pos ;

```

Ενώ εκ πρώτης όψεως φαίνεται ότι πρόκειται για μία χωρική απόσταση, η χώρο-χρονική σημασιολογία απαιτεί για το χαρακτηριστικό διαδρομή η θέση του να είναι συνάρτηση του χρόνου, έτσι αυτή η έκφραση είναι ουσιαστικά μία χώρο-χρονική έκφραση.

Στην προσέγγιση βασισμένη στις διαδρομές αυτή η επερώτηση θα δώσει καλύτερα αποτελέσματα με την χρήση ενός χώρο-χρονικού ευρετηρίου, χωρίς όμως να αποκλείουμε και την χρήση μόνο χρονικού ή χωρικού ευρετηρίου.

Στην προσέγγιση με βάση τα αντικείμενα όμως η χρήση ενός χώρο-χρονικού ευρετηρίου θα είναι κατώτερη από ένα χωρικό ευρετήριο ενώ το χρονικό δεν θα βοηθήσει καθόλου.

Επερώτηση 12

Ποια οχήματα συναντήθηκαν σε ένα σημείο από το αντικείμενο QueryPoints1 μία χρονική στιγμή η οποία ανήκει στο αντικείμενο QueryInstants1?

```

SELECT PP. Pos AS Pos , II . Instant AS Instant ,
C1 . Licence AS Licence1 , C2 . Licence AS Licence2
FROM dataScar C1 , dataScar C2 ,
QueryPoints1 PP, QueryInstant s1 II
WHERE val (C1 . Trip at instant II. Instant ) = PP. Pos
AND val (C2 . Trip at instant II. Instant ) = PP. Pos ;

```

Στην προσέγγιση βασισμένη στις διαδρομές θα χρειαστεί να χρησιμοποιήσουμε την εντολή **SELECT DISTINCT** αντί της **SELECT**.

Επερώτηση 13

Ποια οχήματα πέρασαν από την περιοχή η οποία ανήκει στο αντικείμενο QueryRegions1 κατά την διάρκεια μιας περιόδου από το αντικείμενο QueryPeriods1?

```

SELECT RR. Region AS Region, PP. Period AS Period ,
C. Licence AS Licence
FROM dataScar C, QueryRegions1 RR, QueryPeriods1 PP
WHERE NOT ( isempty ( ( C. Trip at periods PP. Period
)
at RR. Region ) ) ) ;

```

Στην προσέγγιση βασισμένη στις διαδρομές πρέπει να χρησιμοποιήσουμε την εντολή **SELECT DISTINCT**. Εδώ τεστάρουμε την απόδοση των χώρο-χρονικών μεθόδων πρόσβασης για μία επερώτηση απόστασης.

Επερώτηση 14

Ποια οχήματα έχουν περάσει από περιοχή που ανήκει στο αντικείμενο QueryRegions1 μία χρονική στιγμή η οποία ανήκει στο αντικείμενο QueryInstants1?

```
SELECT RR. Region AS Region, II. Instant AS Instant ,
C. Licence AS Licence
FROM dataScar C, QueryRegions1 RR, QueryInstant s1 II
WHERE val (C. Trip at instant II. Instant ) inside RR.
Region ;
```

Στην προσέγγιση βασισμένη στις διαδρομές χρησιμοποιούμε την εντολή **SELECT DISTINCT**. Πρόκειται για μία χώρο-χρονική επερώτηση απόστασης, η οποία μαζί με τις επερωτήσεις 13 και 15 θα μας βοηθήσει να τεστάρουμε την απόδοση των χώρο-χρονικών ευρετηρίων.

Επερώτηση 15

Ποια οχήματα έχουν περάσει από ένα σημείο που ανήκει στο αντικείμενο QueryPoints1 κατά την διάρκεια μιας περιόδου από το αντικείμενο QueryPeriods1?

```
SELECT PO. Pos AS Pos, PR. Period AS Period ,
C. Licence AS Licence
FROM dataScar C, QueryPoints1 PO, QueryPeriods1 PR
WHERE NOT ( isempty ( ( C. Trip at periods PR. Period
) at PO. Pos ) ) ;
```

Όπως και προηγούμενα θα χρησιμοποιήσουμε την εντολή **SELECT DISTINCT** για την προσέγγιση η οποία βασίζεται στις διαδρομές.

Επερώτηση 16

Να καταγραφούν οι αριθμοί κυκλοφορίας των οχημάτων από τα ζευγάρια που ανήκουν στα αντικείμενα QueryLicences1 και QueryLicences2 αντίστοιχα, όπου τα συγκεκριμένα οχήματα βρίσκονται και τα δύο μέσα σε μία περιοχή η οποία ανήκει στο αντικείμενο Query-Regions1 κατά την διάρκεια μιας περιόδου από το αντικείμενο QueryPeriod1, αλλά δεν συναντά το ένα το άλλο την περίοδο αυτή και στο στην συγκεκριμένη περιοχή.

```
SELECT PP. Period AS Period , RR. Region AS Region,
C1 . Licence AS Licence1, C2. Licence AS Licence2
FROM dataScar C1 , dataScar C2 , QueryRegions1 RR,
QueryPeriods1 PP, QueryLicences1 LL1 , QueryLicences2
LL2
WHERE C1 . Licence = LL1 . Licence AND C2 . Licence =
LL2 . Licence
AND LL1 . Licence < LL2 . Licence
AND (C1 . Trip at PP. Period ) passes RR. Region
AND (C2 . Trip at PP. Period ) passes RR. Region
AND isempty ( ( intersection (C1 . Trip , C2 . Trip )
at periods PP. Period ) at RR. Region ) ;
```

Στην προσέγγιση βασισμένη στις διαδρομές έχουμε:

```
SELECT RR. Region AS Region , PP. Period AS Period ,
C1 . Licence AS Licence1 , C2 . Licence AS Licence2
FROM dataMtrip C1 , dataMtrip C2 , QueryRegions1 RR,
QueryPeriods1 PP, QueryLicences1 LL1 , QueryLicences2
LL2
WHERE C1 . Licence < C2 . Licence
AND C1 . Licence = LL1 . Licence
AND (C1 . Trip at PP. Period ) passes RR. Region
AND C2 . Licence = LL2 . Licence
AND (C2 . Trip at PP. Period ) passes RR. Region
GROUP BY RR. Region , PP. Period , C1 . Licence , C2 .
Licence )
EXCEPT
( SELECT RR. Region AS Region, PP. Period AS Period ,
C1 . Licence AS Licence1 , C2 . Licence AS Licence2
FROM dataMtrip C1 , dataMtrip C2 , QueryRegions1 RR,
QueryPeriods1 PP, QueryLicences1 LL1 , QueryLicences2
LL2
WHERE C1 . Licence < C2 . Licence
AND C1 . Licence = LL1 . Licence
AND (C1 . Trip at PP. Period ) passes RR. Region
AND C2 . Licence = LL2 . Licence
AND (C2 . Trip at PP. Period ) passes RR. Region
AND NOT ( isempty ( de f t ime ( ( intersection (C1 .
Trip , C2 . Trip )
at periods PP. Period ) at RR. Region ) ) )
GROUP BY RR. Region , PP. Period , C1 . Licence , C2 .
Licence )
```

Αυτή η επερώτηση τεστάρει την απόδοση των χώρο-χρονικών τελεστών. Στην προσέγγιση με βάση τις διαδρομές χωρικά, χρονικά ή χώρο-χρονικά ευρετήρια ενδεχομένως να αυξήσουν την απόδοση .

Επερώτηση 17

Ποια σημεία από το αντικείμενο QueryPoints έχουν επισκεφτεί με τον μέγιστο αριθμό διαφορετικών οχημάτων?

```
CREATE VIEW PosCount AS
SELECT PP. Pos AS Pos , COUNT (C. Licence ) AS Hits
FROM QueryPoints PP, dataScar C
WHERE C. Trip passes PP. Pos
GROUP BY PP. Pos ;
SELECT Pos
FROM PosCount AS N
WHERE N. Hits = ( SELECT MAX ( Hits ) FROM PosCount ) ;
```

Για την προσέγγιση βασισμένη στις διαδρομές έχουμε την μορφοποίηση:

```

CREATE VIEW PosCount AS
SELECT PP. Pos AS Pos , COUNT DISTINCT(C. Licence ) AS
Hits
FROM QueryPoints PP, dataMtrip C
WHERE C. Trip passes PP. Pos
GROUP BY PP. Pos ;
SELECT Pos
FROM PosCount AS N
WHERE N. Hits = ( SELECT MAX ( Hits ) FROM PosCount ) ;

```

ΚΕΦΑΛΑΙΟ 3 : Εφαρμογή του BerlinMOD στο SECONDO – Αποτελέσματα/Χρόνοι

Γεννήτρια δεδομένων

Για να δημιουργήσουμε τα δεδομένα του συστήματος αναφοράς BerlinMOD χρησιμοποιούμε τις λειτουργίες του SECONDO. Τρέχουμε ένα αρχείο το οποίο περιέχει ένα σύνολο εντολών του SECONDO μορφοποιημένη κατάλληλα για το εκτελέσιμο επίπεδο του SECONDO. Τα δεδομένα αυτά δημιουργούνται χρησιμοποιώντας γεωγραφικά στοιχεία από την Πρωτεύουσα της Γερμανίας, το Βερολίνο. Τα δεδομένα αυτά τα έχουμε πάρει από το πρόγραμμα bbbike και από διάφορες στατιστικές μελέτες από την αντίστοιχη υπηρεσία του δήμου του Βερολίνου.

Τα δεδομένα βρίσκονται μέσα σε τρία αρχεία στην μορφή nested list του SECONDO.

Τα αρχεία αυτά είναι:

- streets.data, παρέχει την σχέση streets: relation {Vmax: real , geoData: line}: Περιλαμβάνει όλους τους δρόμους και την μέγιστη επιτρεπόμενη ταχύτητα σε χλμ/ώρα – Vmax καθώς και την γεωμετρία του δρόμου – geoData
- homeRegions.data, παρέχει την σχέση homeRegions: relation {Priority: int , Weight : real ,geoData: region}: είναι μία σχέση που περιγράφει την κατανομή των κόμβων σπιτιών, όπου το geoData είναι μια περιοχή η οποία ορίζει μια έκταση η οποία θα παρέχει τους κόμβους σπιτιών ανάλογα με το γνώρισμα Weight σε σχέση με το σύνολο όλων των τιμών Weight. Το γνώρισμα Priority καθορίζει την προτεραιότητα μιας περιοχής σε σχέση με τις άλλες όσο αφορά στην επικάλυψη περιοχών.
- workRegions.data, παρέχει την σχέση workRegions: relation {Priority: int , Weight : real ,geoData: region} , η οποία είναι μια σχέση που περιγράφει την κατανομή των κόμβων δουλειάς, με τα βασικά της γνωρίσματα να είναι όπως και στο παραπάνω αρχείο homeRegions.data.

Το σύστημα στο οποίο έγινε η εκτέλεση της γεννήτριας δεδομένων καθώς και η εκτέλεση των επερωτήσεων χρησιμοποιεί Intel I5-2410M (quad core) 2,3 GHz CPU , 8 GB RAM ,640 GB HDD . Όμως λόγω της χρήση Virtual Machine για την λειτουργία του Secondo (Secondo VM Appliance) η μνήμη RAM περιορίστηκε σε 2GB , και ο σκληρός δίσκος σε 20 GB , και λειτουργικό σύστημα (μέσω VM) UBUNTU 10.04 .

Στο αρχικό τμήμα του αρχείου της γεννήτριας δεδομένων έχουμε την δυνατότητα να θέσουμε εμείς τις τιμές που θέλουμε σε μια σειρά από παραμέτρους. Η πιο σημαντική καθολική παράμετρος η οποία επηρεάζει και τις υπόλοιπες είναι η SCALEFACTOR, αλλαγές σε αυτή την παράμετρο καθορίζουν και το μέγεθος των παραγόμενων δεδομένων. Στην τιμή 1.0 παράγονται 2000 οχήματα των οποίων τις κινήσεις καταγράφουμε για χρονικό διάστημα 28 ημερών, για τιμή

SCALEFACTOR=0.2 παράγονται 894 οχήματα για διάστημα καταγραφής 13 ημερών. Στον παρακάτω πίνακα παρουσιάζουμε μεγέθη των δεδομένων που δημιουργούνται για διαφορετικές τιμές της παραμέτρου SCALEFACTOR.

Scalefactor	Network Based			Region Based
	0.05	0.2	1.0	2.0
Πλήθος Παρατηρούμενων	447	894	2000	894
Πλήθος Παρατηρούμενων	6	13	28	13
Χρόνος Δημιουργίας Βάσης & Αντικειμένων	19:42 min	78:20 min	343:18 min	79:33 min
Μέγεθος στον Δίσκο	1.91 GB	4.97GB	19.45GB	5.18GB

Εικόνα 4: Μεγέθη Δεδομένων ανάλογα με το SCALEFACTOR

Πριν προβούμε στην παρουσίαση των αποτελεσμάτων που εξάγαμε από την εκτέλεση των επερωτήσεων του BerlinMOD στο δικό μας μηχάνημα, πρέπει σε αυτό το σημείο να σταθούμε στην διαφορετικότητα που παρουσιάζουν οι χρόνοι απόκρισης των επερωτήσεων, με βάση και όσων έχουμε αναπτύξει στις προηγούμενες παραγράφους, ανάλογα την προσέγγιση που χρησιμοποιούμε (OBA – TBA –Network Based – Region Based).

Πιο συγκεκριμένα όπως φαίνονται και στον πίνακα που ακολουθεί, οι επερωτήσεις 1 & 2 εκτελούνται πάρα πολύ γρήγορα. Αυτό συμβαίνει διότι αυτές είναι απλές επερωτήσεις μη χώρο-χρονικές.

ΕΠΕΡΩΤΗΣΕΙΣ	Network Based						Region Based	
	Scalefactor = 0,05		Scalefactor = 0,2		Scalefactor = 1,0		Scalefactor = 0,2	
	ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ (sec)		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ (sec)		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ (sec)		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ (sec)	
	OBA	TBA	OBA	TBA	OBA	TBA	OBA	TBA
ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΗΜΙΟΥΡΓΩΝ BERLIN MOD								
Q1	0,406	0,335	0,476	0,451	0,46	0,407	0,362	0,341
Q2	0,099	0,055	0,05	0,14	0,113	0,099	0,028	0,021
ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΙΚΗΣ ΜΑΣ ΔΟΚΙΜΗΣ								
Q1	0,052	0,049	0,058	0,06	0,053	0,049	0,063	0,039
Q2	0,047	0,01	0,014	0,017	0,019	0,008	0,019	0,019

Εικόνα 5: Επερωτήσεις 1,2

Όμως όσον αφορά στις επόμενες επερωτήσεις παρατηρούμε διαφορές στους χρόνους απόκρισης, μεταξύ των διαφορετικών προσεγγίσεων (OBA-TBA). Στις επερωτήσεις 6 και 9 η Αντικειμενοστραφής Προσέγγιση (OBA) εμφανίζει γρηγορότερους χρόνους καθώς εκμεταλλεύεται την πρόσβαση που έχει στο ιστορικό διαδρομών, ολόκληρο ή μεγαλύτερες διαδρομές, ενός αντικειμένου. Στις υπόλοιπες η Προσέγγιση Βασισμένη στις Διαδρομές (TBA) εκμεταλλεύεται τους χώρο-χρονικούς δείκτες σε οριοθετημένες χώρο-χρονικές επερωτήσεις. Τέτοιες περιπτώσεις επερωτήσεων είναι οι 10 έως 16 όπου έχουμε όρια που αφορούν είτε σε απόσταση, είτε σε χώρο, είτε ακόμα και σε χρονικές περιόδους. Σε αυτό συμβάλλει η σχέση dataMtrips που υπάρχει στην TBA προσέγγιση και περιέχει όλες τις κινήσεις και στάσεις ενός οχήματος ως μεμονωμένα «ταξίδια».

Ακολουθούν οι αντίστοιχοι πίνακες :

ΕΠΕΡΩΤΗΣΕΙΣ	Network Based						Region Based	
	Scalefactor = 0,05		Scalefactor = 0,2		Scalefactor = 1,0		Scalefactor = 0,2,	
	ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ	
	ΟΒΑ	ΤΒΑ	ΟΒΑ	ΤΒΑ	ΟΒΑ	ΤΒΑ	ΟΒΑ	ΤΒΑ
ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΗΜΙΟΥΡΓΩΝ BERLIN MOD								
Q6	71,396	189,435	333,341	1942,071	7032	53910,502	193,308	430,185
Q9	392,336	784,329	1102,58	3241,18	4791,73	21730,8	1078,71	2020,96
ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΙΚΗΣ ΜΑΣ ΔΟΚΙΜΗΣ								
Q6	23,634	14,791	202,127	269,094	7810,641	13256,042	267,13	240,348
Q9	311,592	476,322	664,613	1271,92	2192,207	5510,445	756,826	1346,06

Εικόνα 6: Επερωτήσεις 6,9

ΕΠΕΡΩΤΗΣΕΙΣ	Network Based						Region Based	
	Scalefactor = 0,05		Scalefactor = 0,2		Scalefactor = 1,0		Scalefactor = 0,2,	
	ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ	
	ΟΒΑ	ΤΒΑ	ΟΒΑ	ΤΒΑ	ΟΒΑ	ΤΒΑ	ΟΒΑ	ΤΒΑ
ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΗΜΙΟΥΡΓΩΝ BERLIN MOD								
Q10	681,937	221,276	3170,48	1189,13	23951,8	16410,2	3011,84	627,194
Q11	0,956	0,58	1,862	0,849	11,602	1,411	0,82	0,669
Q12	2,188	0,553	144,466	0,51	964,456	0,625	97,807	0,564
Q13	50,376	45,189	426,079	128,682	2015,68	261,572	139,375	90,711
Q14	2,129	2,02	6,444	3,083	138,305	13,075	4,909	2,873
Q15	3,662	4,53	121,011	30,562	322,635	36,343	51,973	40,674
Q16	144,565	58,967	102,139	49,206	132,165	74,842	118,154	51,944
ΑΠΟΤΕΛΕΣΜΑΤΑ ΔΙΚΗΣ ΜΑΣ ΔΟΚΙΜΗΣ								
Q10	379,585	102,191	1352	374,851	7824,863	3530,972	1699,39	553,356
Q11	0,077	0,074	0,275	0,395	3,142	2,394	0,246	0,219
Q12	0,283	0,108	8,432	0,216	25,402	0,574	8,855	0,199
Q13	7,283	4,391	39,168	32,712	117,820	173,956	32,273	31,882
Q14	0,212	0,216	1,169	1,135	45,708	16,572	1,27	1,307
Q15	0,599	0,404	12,983	11,648	22,704	59,194	16,446	16,491
Q16	11,707	9,125	13,365	36,694	27,944	268,952	21,675	39,644

Εικόνα 7: Καλύτεροι Χρόνοι ΤΒΑ(Q10-Q16)

Στο [24] οι συγγραφείς περιέχουν επιπλέον μια λεπτομερή περιγραφή του τρόπου λειτουργίας των επερωτήσεων 7,9,10 και 17 σε σχέση με την συμπεριφορά τους και τους αποκρίσεις ,ανάλογα με την προσέγγιση.

Αξίζει όμως να σταθούμε στην εμφανή διαφορά των χρόνων απόκρισης για την επερώτηση 17 ,και την περίπτωση προσέγγισης Network based – Region based όπου η διαφορετική κατανομή των σημείων αποτελεί σημαντικό παράγοντα για την συγκεκριμένη επερώτηση. Αυτή η διαφορά είναι αρκετά μεγάλη όπως φαίνεται παρακάτω:

Network Based		Region Based	
Scalefactor = 0,2		Scalefactor = 0,2,	
ΧΡΟΝΟΙ		ΧΡΟΝΟΙ	
ΑΠΟΚΡΙΣΗΣ (sec)		ΑΠΟΚΡΙΣΗΣ (sec)	
OBA	TBA	OBA	TBA
467,125	242,219	69,109	194,322

Εικόνα 8: Επερώτηση 17

Επίσης αρκετές επερωτήσεις φαίνονται να επηρεάζονται περισσότερο στις αλλαγές των δεδομένων και πιο συγκεκριμένα οι 6,7,13,15,17 (OBA) και 6,9,10 (TBA) ,όπως φαίνονται στους ακόλουθους πίνακες.

ΕΠΕΡΩΤΗΣΕΙΣ	Network Based			Region Based
	Scalefactor =0,05 Χρονος Απόκρισης (sec)	Scalefactor = 0,2 Χρονος Απόκρισης (sec)	Scalefactor = 1,0 Χρονος Απόκρισης (sec)	Sclaefactor = 0,2 Χρονος Απόκρισης (sec)
OBA ΕΠΕΡΩΤΗΣΕΙΣ ΔΗΜΙΟΥΡΓΩΝ BERLINMOD				
Q6	71,396	333,341	7032	193,308
Q7	92,666	2325,34	23324,7	564,038
Q13	50,376	426,079	2015,68	139,375
Q15	3,662	121,011	322,635	51,973
Q17	5,129	467,125	5374,08	69,109
OBA ΕΠΕΡΩΤΗΣΕΙΣ ΔΙΚΗΣ ΜΑΣ ΔΟΚΙΜΗΣ				
Q6	23,634	202,127	7810,641	267,13
Q7	7,241	47,368	123,861	47,134
Q13	7,283	39,168	117,820	32,273
Q15	0,599	12,983	22,704	16,446
Q17	1,782	17,418	21,506	16,635

Εικόνα 9: Επερωτήσεις OBA

ΕΠΕΡΩΤΗΣΕΙΣ	Network Based			Region Based
	Scalefactor =0,05 Χρονος Απόκρισης (sec)	Scalefactor = 0,2 Χρονος Απόκρισης (sec)	Scalefactor = 1,0 Χρονος Απόκρισης (sec)	Sclaefactor = 0,2 Χρονος Απόκρισης (sec)
TBA ΕΠΕΡΩΤΗΣΕΙΣ ΔΗΜΙΟΥΡΓΩΝ BERLINMOD				
Q6	189,435	1942,071	53910,502	430,185
Q9	784,329	3241,18	21730,8	2020,96
Q10	221,276	1189,13	16410,2	627,194
TBA ΕΠΕΡΩΤΗΣΕΙΣ ΔΙΚΗΣ ΜΑΣ ΔΟΚΙΜΗΣ				
Q6	14,791	269,094	13256,042	240,348
Q9	476,322	1271,92	5510,445	1346,06
Q10	102,191	374,851	3530,972	553,356

Εικόνα 10: Επερωτήσεις TBA

Ενόσω ο αριθμός μονάδων σε σχέση με τα οχήματα είναι μεγαλύτερος για την Region Based προσέγγιση ,υπάρχουν μεγάλες διαφορές που αφορούν στην κατανομή των κόμβων εργασίας και οικίας (Γκαουσιανή κατανομή με βάση το κέντρο του δικτύου για την Network Based και χωρική κατανομή των κόμβων οικίας και εργασίας για την Region Based) . Αυτές οι διαφορές οδηγούν στην δημιουργία διαφορετικών συστάδων και με μικρότερη «ποικιλία» την Region Based , και σε αυτές ακριβώς αποδίδεται η διαφορετικότητα στον χρόνο απόκρισης.

Ακολουθούν οι συγκεντρωτικοί πίνακες με τα αποτελέσματα που έχουν εξάγει οι δημιουργοί του BerlinMOD και τα αντίστοιχα αποτελέσματα από τις δικές μας δοκιμές :

ΕΠΕΡΩΤΗΣΕΙΣ	Network Based						Region Based	
	Scalefactor = 0,05		Scalefactor = 0,2		Scalefactor = 1,0		Scalefactor = 0,2,	
	ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ	
	(sec)		(sec)		(sec)		(sec)	
	OBA	TBA	OBA	TBA	OBA	TBA	OBA	TBA
Q1	0,406	0,335	0,476	0,451	0,46	0,407	0,362	0,341
Q2	0,099	0,055	0,05	0,14	0,113	0,099	0,028	0,021
Q3	2,29	0,616	6,303	0,993	12,08	1,092	7,407	0,963
Q4	76,664	49,516	625,431	273,426	6232,56	966,393	640,642	284,292
Q5	16,737	20,059	45,535	34,71	121,885	61,015	38,16	26,666
Q6	71,396	189,435	333,341	1942,071	7032	53910,502	193,308	430,185
Q7	92,666	35,654	2325,34	241,182	23324,7	135,724	564,038	194,097
Q8	1,209	1,214	5,854	3,521	13,989	4,308	2,11	1,403
Q9	392,336	784,329	1102,58	3241,18	4791,73	21730,8	1078,71	2020,96
Q10	681,937	221,276	3170,48	1189,13	23951,8	16410,2	3011,84	627,194
Q11	0,956	0,58	1,862	0,849	11,602	1,411	0,82	0,669
Q12	2,188	0,553	144,466	0,51	964,456	0,625	97,807	0,564
Q13	50,376	45,189	426,079	128,682	2015,68	261,572	139,375	90,711
Q14	2,129	2,02	6,444	3,083	138,305	13,075	4,909	2,873
Q15	3,662	4,53	121,011	30,562	322,635	36,343	51,973	40,674
Q16	144,565	58,967	102,139	49,206	132,165	74,842	118,154	51,944
Q17	5,129	27,393	467,125	242,219	5374,08	1097,145	69,109	194,322

Εικόνα 11: Συγκεντρωτικά Αποτελέσματα I

ΕΠΕΡΩΤΗΣΕΙΣ	Network Based						Region Based	
	Scalefactor = 0,05		Scalefactor = 0,2		Scalefactor = 1,0		Scalefactor = 0,2,	
	ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ		ΧΡΟΝΟΙ ΑΠΟΚΡΙΣΗΣ	
	(sec)		(sec)		(sec)		(sec)	
	OBA	TBA	OBA	TBA	OBA	TBA	OBA	TBA
Q1	0,052	0,049	0,058	0,06	0,053	0,049	0,063	0,039
Q2	0,047	0,01	0,014	0,017	0,019	0,008	0,019	0,019
Q3	0,159	0,087	0,423	0,179	0,784	0,251	0,611	0,309
Q4	5,811	2,274	48,015	56,068	484,621	884,829	50,31	53,335
Q5	4,273	6,842	4,729	9,63	5,149	13,769	7,427	12,752
Q6	23,634	14,791	202,127	269,094	7810,641	13256,042	267,13	240,348
Q7	7,241	3,597	47,368	40,418	123,861	37,782	47,134	37,92
Q8	0,72	0,59	0,559	0,418	0,214	0,125	0,957	0,849
Q9	311,592	476,322	664,613	1271,92	2192,207	5510,445	756,826	1346,06
Q10	379,585	102,191	1352	374,851	7824,863	3530,972	1699,39	553,356
Q11	0,077	0,074	0,275	0,395	3,142	2,394	0,246	0,219
Q12	0,283	0,108	8,432	0,216	25,402	0,574	8,855	0,199
Q13	7,283	4,391	39,168	32,712	117,820	173,956	32,273	31,882
Q14	0,212	0,216	1,169	1,135	45,708	16,572	1,27	1,307
Q15	0,599	0,404	12,983	11,648	22,704	59,194	16,446	16,491
Q16	11,707	9,125	13,365	36,694	27,944	268,952	21,675	39,644
Q17	1,782	1,159	17,418	42,686	21,506	805,332	16,635	40,422

Εικόνα 12: Συγκεντρωτικά Αποτελέσματα II

ΚΕΦΑΛΑΙΟ 4 : Οδηγίες Εγκατάστασης Secondo

Το SECONDO μπορεί να εγκατασταθεί σε διάφορα λειτουργικά συστήματα όπως έχουμε αναφέρει και παραπάνω. Στην ιστοσελίδα του συστήματος υπάρχουν διαθέσιμα όλα τα πακέτα ανάλογα με το λειτουργικό που θα γίνει η εγκατάσταση όπως επίσης και ο πηγαίος κώδικας. Μπορούμε λοιπόν να εγκαταστήσουμε την εφαρμογή στα ακόλουθα λειτουργικά συστήματα:

- SuSe Linux: 8.1, 8.2, 9.0, 9.2, 10.x
- Mac-OSX: Tiger
- MS-Windows: Win98, Win2000, WinXP, Windows 7

Επίσης έχουμε την δυνατότητα να χρησιμοποιήσουμε το Secondo-Live CD μέσω του οποίου δεν θα χρειαστεί να κάνουμε οποιαδήποτε αλλαγή στον σκληρό δίσκο του υπολογιστή μας. Το Live CD βασίζεται σε Xubuntu 10.04 και στην έκδοση Secondo 3.0. Βέβαια, όπως αποδεικνύεται και στην πράξη, το να χρησιμοποιήσουμε το Live CD δεν είναι η καλύτερη μέθοδος για να τεστάρουμε το Secondo, ειδικά με πολύπλοκες επερωτήσεις. Πιθανώς να είναι μια καλή μέθοδος μόνο αν θέλουμε να δούμε βασικές λειτουργίες του προγράμματος. Αυτό συμβαίνει κυρίως γιατί κατά την διάρκεια εκτέλεσης του το Secondo χρησιμοποιεί τον σκληρό δίσκο για να αποθηκεύει δεδομένα τα οποία σε ορισμένες περιπτώσεις ενδέχεται να είναι αρκετά μεγάλα σε όγκο ενώ όταν τρέχει από το Live CD ο σκληρός δίσκος αντιστοιχεί στην κύρια μνήμη του υπολογιστή, η οποία χρησιμοποιείται και για άλλες λειτουργίες.

Το Secondo μπορούμε βεβαίως να το εγκαταστήσουμε και σε υπολογιστή με λειτουργικό σύστημα Linux. Επειδή όμως το Secondo εξαρτάται από κάποια πακέτα λογισμικού πρέπει πρώτα να εγκαταστήσουμε αυτά για να μπορέσουμε έπειτα να προχωρήσουμε με την εγκατάσταση του ίδιου του Secondo. Κάποια από αυτά τα πακέτα είναι τα περιβάλλοντα Java(απαραίτητο για το GUI) και Prolog (Optimizer). Τα πακέτα αυτά είναι διαθέσιμα στην ιστοσελίδα του προγράμματος μαζεμένα σε ένα ISO αρχείο σαν ένα ολοκληρωμένο πακέτο λογισμικού. Η καλύτερη προσέγγιση είναι να κατεβάσουμε το αρχείο με το σενάριο εγκατάστασης έτσι ώστε όλα τα απαραίτητα εργαλεία να εγκατασταθούν με τη σειρά που πρέπει. Την ίδια μέθοδο ακολουθούμε και για εγκατάσταση σε περιβάλλον Windows με την βασική διαφορά ότι πρέπει πρώτα να εγκαταστήσουμε το πρόγραμμα MinGW installer και έπειτα την κονσόλα εντολών MSYS όπου από εκεί θα προχωρήσει όλη η εγκατάσταση και μετά την ολοκλήρωση από το MSYS θα τρέχουμε τις διεπαφές ανάλογα με την εργασία που έχουμε να κάνουμε στο Secondo.

Η καλύτερη μέθοδος σύμφωνα με τους δημιουργούς του Secondo και με τις δοκιμές που πραγματοποιήσαμε στα πλαίσια της παρούσας έρευνας είναι το SecondoVM Appliance. Πρόκειται για ένα εικονικό λειτουργικό σύστημα που μπορεί να τρέξει μέσω του VM-Ware Player και έχει εγκατεστημένο το Secondo. Υπάρχουν διαθέσιμες αρκετές εκδόσεις του Secondo οι οποίες τρέχουν μέσα σε διαφορετικές εκδόσεις Ubuntu Linux.

Για να τρέξουμε το SecondoVM Appliance κατεβάζουμε την έκδοση που θέλουμε από τις διαθέσιμες στην ιστοσελίδα του Secondo, κάνουμε unzip το αρχείο και αφού τρέξουμε το VM-Ware Player φορτώνουμε το εικονικό λειτουργικό σύστημα από το αρχείο που κατεβάσαμε. Σε αυτό το σημείο μπορούμε να κάνουμε κάποια παραμετροποίηση στο σύστημα (σκληρός δίσκος, μνήμη, κλπ.). Αφού ανοίξει το λειτουργικό και συνδεθούμε μπορούμε να ξεκινήσουμε να δουλεύουμε σε όποια διεπαφή του Secondo επιθυμούμε. Μπορούμε να ξεκινήσουμε είτε το SecondoTTYBDB που είναι μια απλή διεπαφή εισαγωγής εντολών χωρίς γραφικό περιβάλλον είτε το GUI σε συνδυασμό με τον Optimizer και τον πυρήνα – kernel. Για να συνδεθούμε στο GUI ανοίγουμε ένα terminal και μεταφερόμαστε στο φάκελο Secondo\bin. Στον φάκελο αυτό υπάρχουν οι εκτελέσιμες διεπαφές του SECONDO.

Για να τρέξουμε το Secondo GUI πρέπει πρώτα να ξεκινήσουμε τον Secondo Server. Αυτό πραγματοποιείται εκτελώντας το Secondo Monitor με την εντολή `SecondoMonitor -s`. Έπειτα, αν θέλουμε να χρησιμοποιήσουμε και τον βελτιστοποιητή επερωτήσεων (optimizer), ανοίγουμε νέο terminal και μεταφερόμαστε στο φάκελο `Secondo\Optimizer`. Ξεκινάμε τον βελτιστοποιητή εκτελώντας την εντολή `StartOptServer`. Μπορούμε να ξεκινήσουμε το γραφικό περιβάλλον χρήστη GUI ανοίγοντας ένα νέο terminal. Μεταφερόμαστε στο φάκελο `Secondo\JavaGui` και εκτελούμε την εντολή `sgui`. Με την εκτέλεση της εντολής αυτής εμφανίζεται το γραφικό περιβάλλον του Secondo και μπορούμε να αρχίσουμε να εργαζόμαστε σε αυτό.

Στη παρούσα εργασία για να ελεγχτεί το σύστημα αναφοράς BerlinMOD χρησιμοποιείται η απλή διεπαφή εισαγωγής εντολών `SecondoTTYBDB`. Για να ενεργοποιηθεί, ανοίγουμε ένα terminal και μεταφερόμαστε στον φάκελο `Secondo\bin` και τρέχουμε την εντολή `SecondoTTYBDB`. Απαραίτητη προϋπόθεση είναι να έχουν μεταφερθεί τα γεωγραφικά δεδομένα από τα αρχεία `streets.data`, `homeRegions.data` και `workRegions.data` στο φάκελο `Secondo\bin`. Τα αρχεία αυτά αποτελούν τα βασικά συστατικά για να δημιουργηθούν τα δεδομένα της βάσης καθώς και τη ίδια η βάση. Στον ίδιο φάκελο έχουμε μεταφέρει επίσης τα αρχεία που περιέχουν τις εντολές για να δημιουργήσουμε τα δεδομένα, τη βάση και να εκτελέσουμε τις επερωτήσεις (`BerlinMOD_DataGenerator.SEC`, `BerlinMOD_CreateObjects.SEC`, `BerlinMOD_OBA-Queries.SEC`, `BerlinMOD_TBA-Queries.SEC`). Έπειτα, τρέχουμε, με την παραπάνω αναγραφόμενη σειρά, τα αρχεία εκτελώντας την εντολή `@BerlinMOD_DataGenerator.SEC` για καθένα από τα αρχεία.

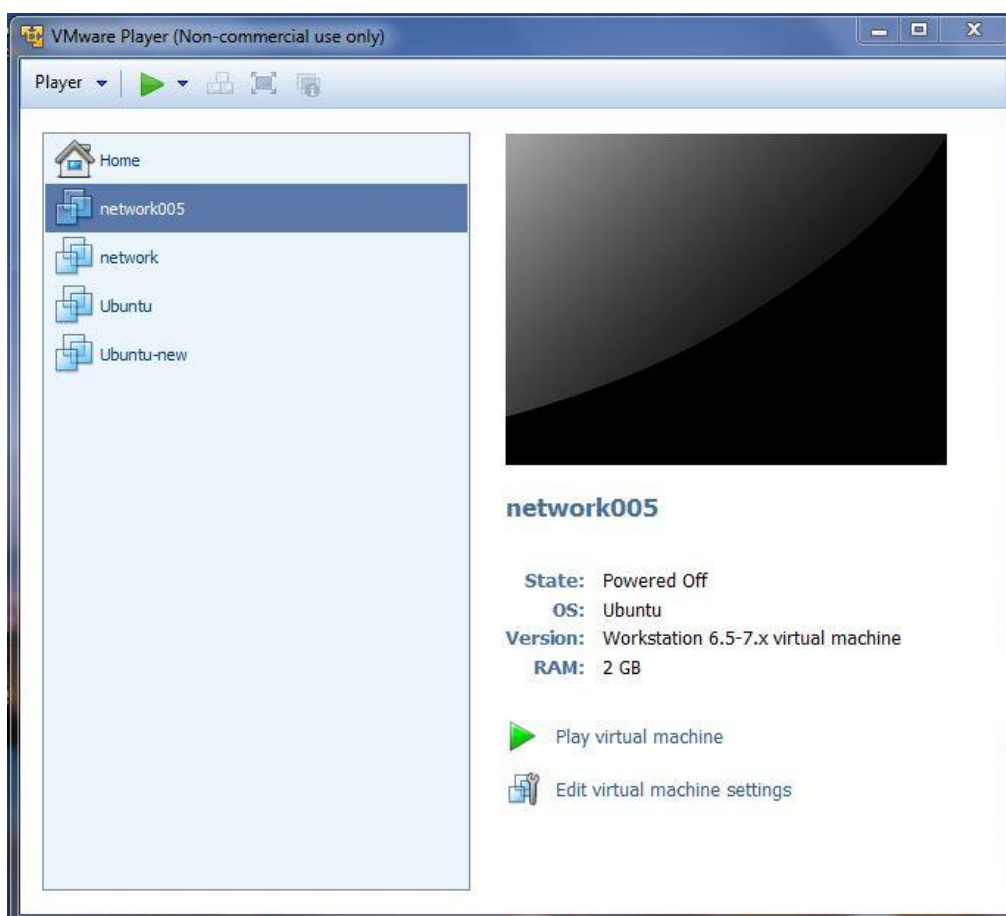
Πριν τρέξουμε το αρχείο της γεννήτριας δεδομένων έχουμε την δυνατότητα να αλλάξουμε κάποιες παραμέτρους της βάσης. Η πιο βασική παράμετρος η οποία θα επηρεάσει τον αριθμό των οχημάτων που θα δημιουργηθούν καθώς και τον αριθμό των ημερών που θα διαρκέσει το πείραμα είναι η `SCALEFACTOR`. Αν θέσουμε στην παράμετρο `SCALEFACTOR` την τιμή 1.0 το πείραμα διαρκεί 28 ημέρες καταγράφοντας τις κινήσεις 2000 οχημάτων. Εάν θέσουμε στην παράμετρο την τιμή 0.2 το πείραμα διαρκεί 13 ημέρες καταγράφοντας τις κινήσεις 894 οχημάτων. Το μέγεθος των χωρικών δεδομένων δεν αλλάζει. Οι τιμές που θα θέσουμε στην παράμετρο για να δημιουργήσουμε δεδομένα είναι 0.05, 0.2, 1.0. Επίσης έχουμε την επιλογή να εξάγουμε σε αρχεία τα δεδομένα που θα δημιουργηθούν. Μπορούμε να επιλέξουμε ανάμεσα στους εξής τύπους αρχείων στα οποία θα εξάγουμε τα δεδομένα μας: `.csv`, `shape files (.shp)` και `secondo nested lists`.

Στο σημείο αυτό είμαστε έτοιμοι να τρέξουμε το αρχείο `BerlinMOD_DataGenerator.SEC`. Η διάρκεια της εκτέλεσης των εντολών του `BerlinMOD_DataGenerator.SEC` κυμαίνεται από μερικά λεπτά για `SCALEFACTOR` 0.05 έως και αρκετές ώρες για `SCALEFACTOR` 1.0. Επόμενο βήμα είναι να τρέξουμε με τον ίδιο τρόπο το αρχείο `BerlinMOD_CreateObjects.SEC` που ουσιαστικά στήνει τη βάση και η εκτέλεσή του διαρκεί αντίστοιχο χρόνο με τη γεννήτρια δεδομένων. Τέλος, τρέχουμε τις επερωτήσεις, οι οποίες βρίσκονται μέσα στα αρχεία `BerlinMOD_OBA-Queries.SEC` και `BerlinMOD_TBA-Queries.SEC`. Τα αποτελέσματα που εξάγονται από την εκτέλεση των επερωτήσεων (χρόνος) αποτελούν και το σημείο αναφοράς του συστήματός μας.

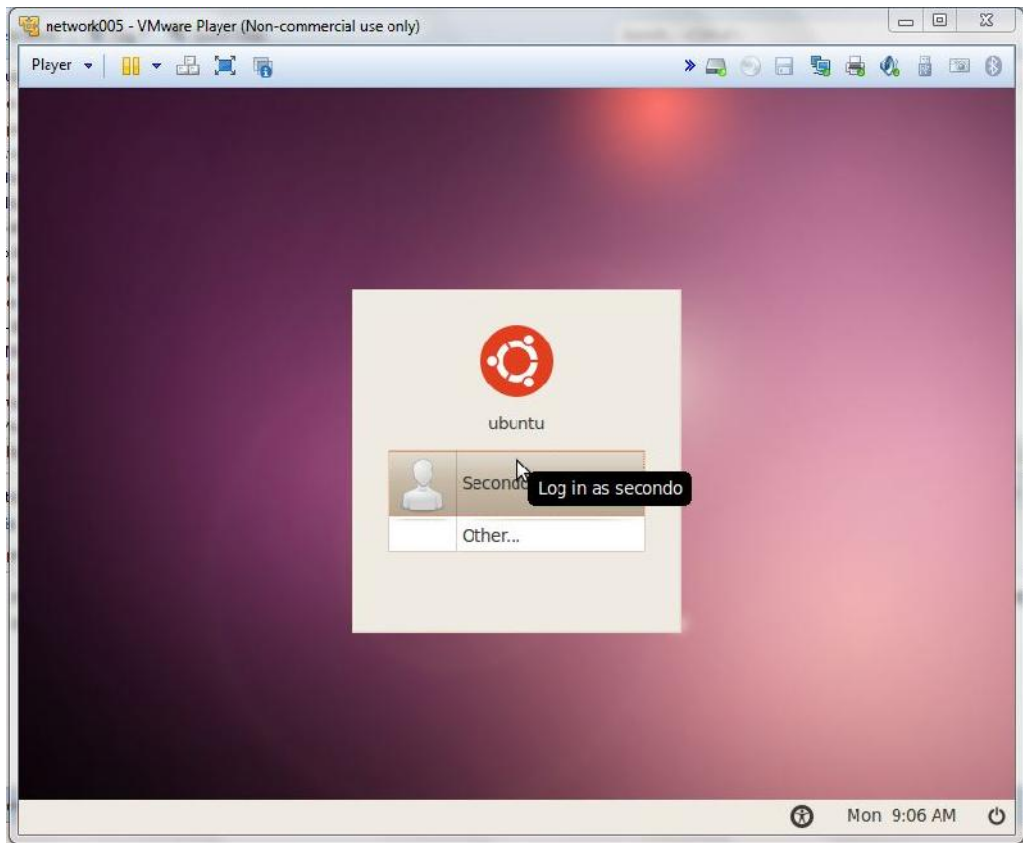
Παρακάτω ακολουθεί η διαδικασία αναλυτικά:

1. Ανοίγουμε το VMware Player
2. Επιλέγουμε το σύστημα που θα τρέξουμε (το έχουμε κατεβάσει από την ιστοσελίδα του Secondo)
3. Αν επιθυμούμε μπορούμε να κάνουμε και αλλαγές στο λειτουργικό σύστημα (αύξηση μνήμης, σκληρού δίσκου, επιλογή αριθμού επεξεργαστών)

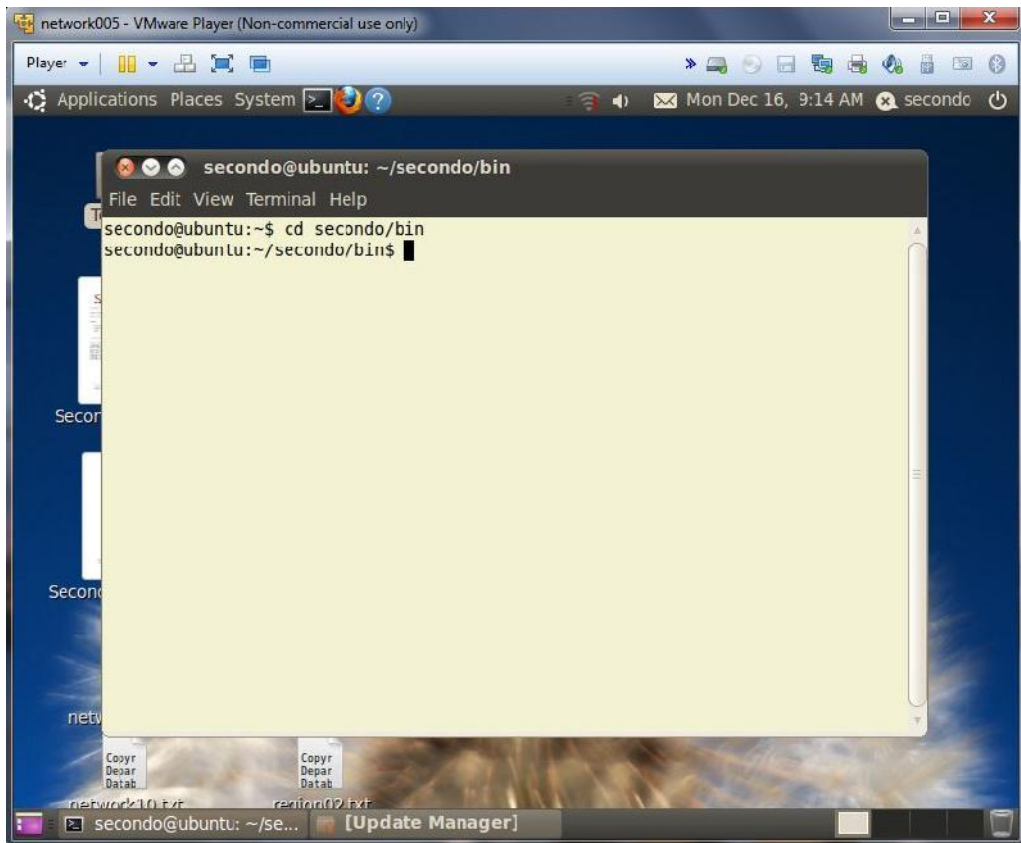
4. Αφού ξεκινήσει το λειτουργικό σύστημα συνδεόμαστε ως χρήστης Secondo με κωδικό secondo
5. Ανοίγουμε ένα terminal και μεταφερόμαστε στον φάκελο Secondo/bin
6. Τρέχουμε την απλή διεπαφή SecondoTTYBDB
7. Με το συγκεκριμένο κομμάτι του Secondo τρέξαμε το σύστημα αναφοράς BerlinMOD όπως φαίνεται και στην εικόνα
8. Αν θέλουμε να ξεκινήσουμε το γραφικό περιβάλλον χρήστη πρέπει πρώτα να ξεκινήσουμε τον πυρήνα(Kernel) με την εντολή SecondoMonitor -s. Ο πυρήνας λειτουργεί σαν εξυπηρετητής πολλαπλών χρηστών. Αν έχει σηκωθεί ο εξυπηρετητής θα έχουμε την εικόνα ...
9. Ανοίγουμε ένα καινούριο τερματικό και μεταφερόμαστε στον φάκελο Secondo/Optimizer, είναι ο φάκελος του βελτιστοποιητή, τον ξεκινάμε με την εντολή StartOptServer
10. Είμαστε έτοιμοι να ξεκινήσουμε το GUI, σε ένα καινούριο terminal πάμε στον φάκελο Secondo/Javagui και τρέχουμε την εντολή sgui
11. Μπορούμε τώρα να ξεκινήσουμε να εργαζόμαστε στο γραφικό περιβάλλον χρήστη.



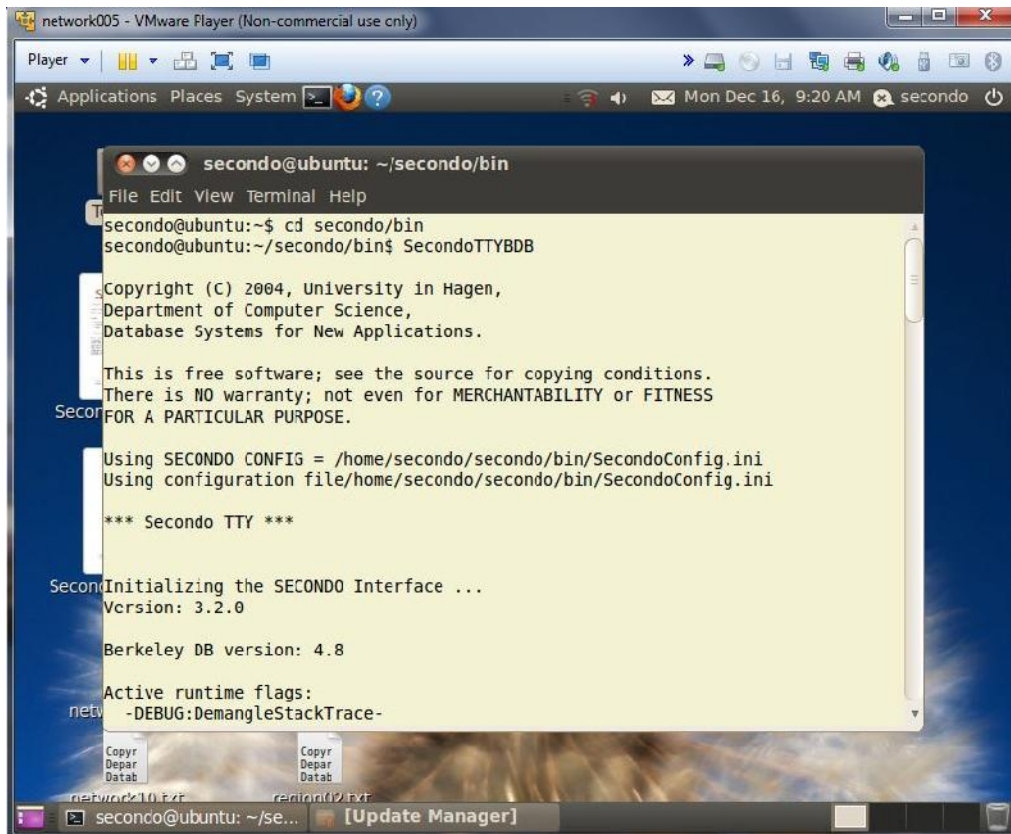
Βήμα 1



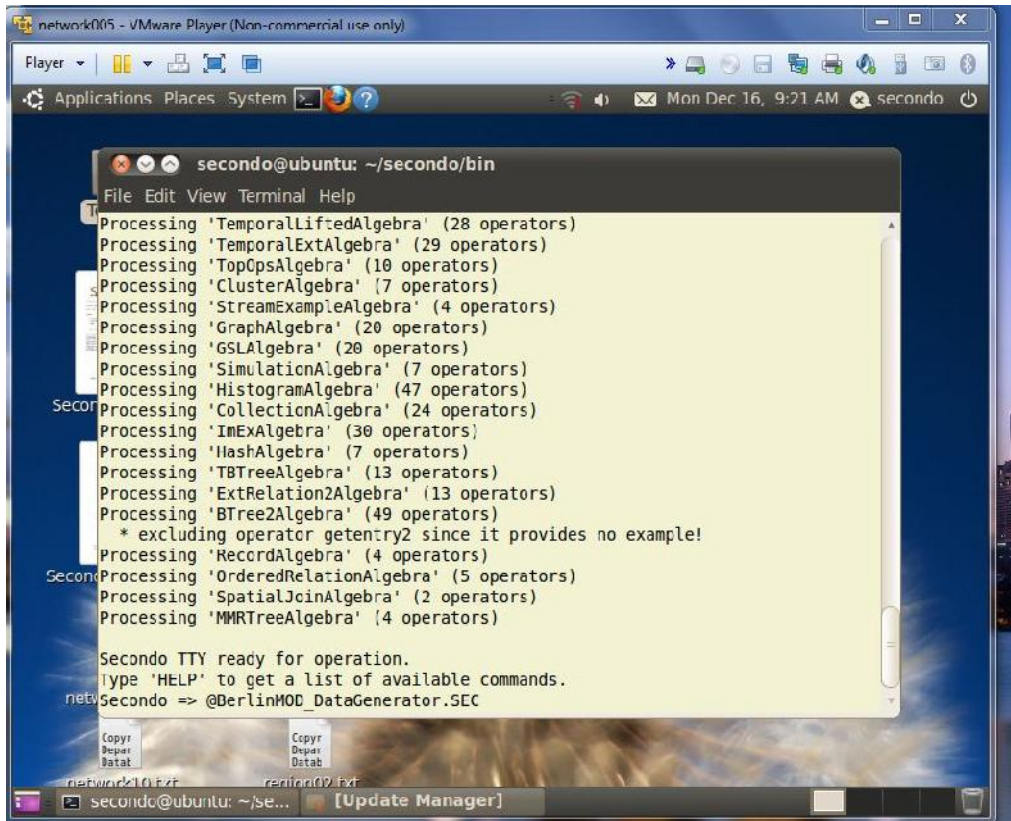
Βήμα 2



Βήμα 3



Βήμα 4



Βήμα 5

```
Applications Places System > ?
secondo@ubuntu: ~/secondo/bin
File Edit View Terminal Help
Searching environment for configuration file ...
Configuration file '/home/secondo/secondo/bin/SecondoConfig.ini':
Configuration seems to be ok.

Initializing process management ... completed.
Initializing storage management interface ... Startup of the Storage Management
Interface (SMI) ...
Cachesize: 65536 kb.
Lock timeout: 0 microseconds
TXN timeout: 0 microseconds
Warning: The folder SecondoHome='/home/databases1' does not exist!
Using default directory ...
Database directory: SecondoHome='/home/secondo/secondo-databases'.
SMI-Mode: MultiUserMaster
Lock timeout: 0 microseconds
TXN timeout: 0 microseconds
completed.
Launching Checkpoint service ... Starting Process:
Program: SecondoCheckpoint
Args: "/home/secondo/secondo/bin/SecondoConfig.ini"
completed.
Launching Secondo Registrar ... Starting Process:
Program: SecondoRegistrar
Args: "/home/secondo/secondo/bin/SecondoConfig.ini"
completed.

Secondo Monitor ready for operation.
Type 'HELP' to get a list of available commands.
Startup in progress ... Starting Process:
Program: SecondoListener
Args: "/home/secondo/secondo/bin/SecondoConfig.ini"
completed.
SEC_MON>
```

Βήμα 6


```

secondo@ubuntu: ~/secondo/Optimizer
File Edit View Terminal Help

[ ] eagerObjectCreation: Create all samples and small objects at 'open data
bases'.
[ ] rewriteMacros: Allow for macros in queries.
[ ] rewriteInference: Add inferred predicates to where clause.
  ( ) rtreeIndexRules: Infer predicates to exploit R-tree indices.
[ ] rewriteCSE: Extend with attributes for CSE values.
  ( ) rewriteCSEall: Extend with attributes for ALL_CSEs.
  ( ) rewriteRemove: Remove attributes as early as possible.
[x] debug: Execute debugging code. Also use 'toggleDebug.'.
[x] autosave: Autosave option settings on 'halt.'.
[ ] noprogess: Do not send predicate data for progress estimation

[ ] subqueries: Process subqueries.
  ( ) subqueryUnnesting: Apply unnesting algorithms to subqueries.

Type 'loadOptions.' to load the saved option configuration.
Type 'saveOptions.' to save current option configuration to disk.
Type 'defaultOptions.' to restore the default options.
Type 'setOption(X).' to select option X.
Type 'delOption(X).' to unselect option X.
Type 'showOptions.' to view this option list.

Type 'helpMe.' to get an overview on user level predicates.

% calloptimizer compiled 0.02 sec, 1,261,828 bytes
optserver >
waiting for requests

opt-server >

```

Βήμα 7

```

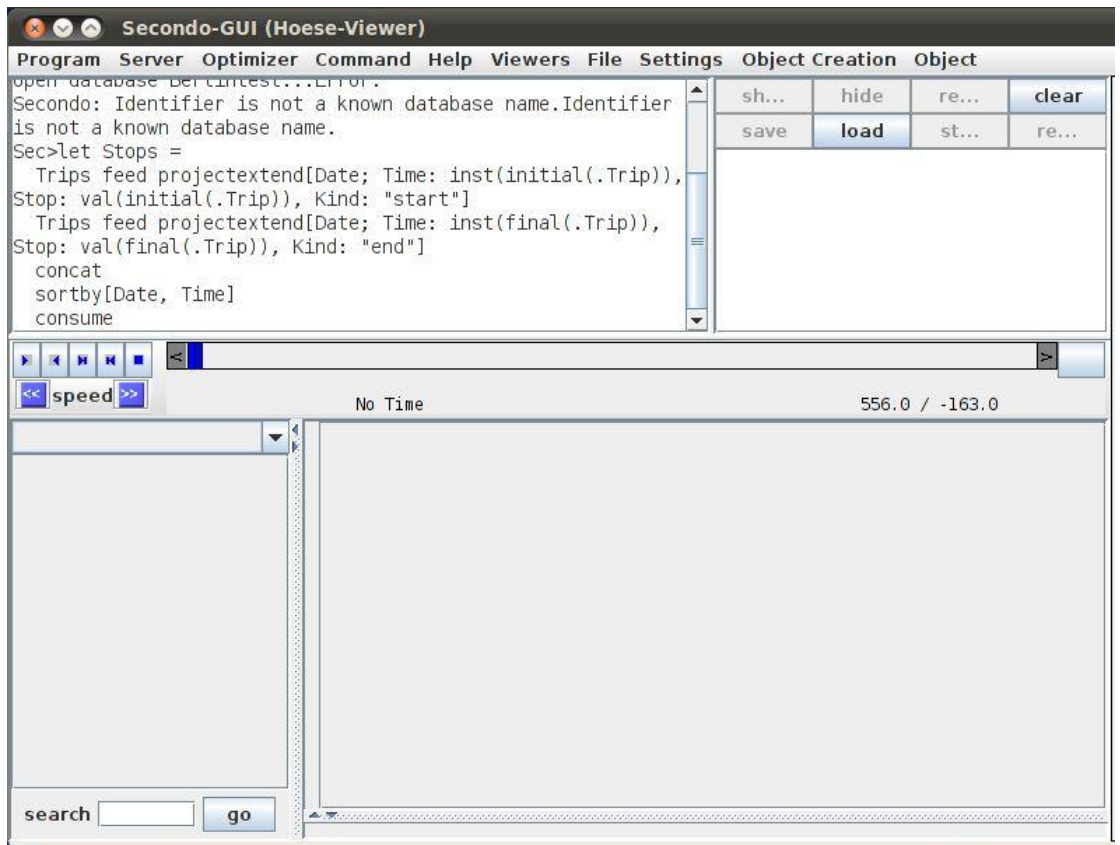
secondo@ubuntu: ~/secondo/Javagui
File Edit View Terminal Help

secondo@ubuntu:~$ cd ~/secondo/Javagui
secondo@ubuntu:~/secondo/Javagui$ sgui
classpath= lib/java_cup_v10k_runtime.jar;lib/jl0.4.jar;.:lib/14_os_jpedal.jar;li
bye.jar;lib/grappal.2.jar;lib/batik.jar;secondointerface/SecondoInterface.jar
Info: start Javagui without any argument
Info: load configuration data from: /home/secondo/secondo/Javagui/gui.cfg
Info: set ServerName to 127.0.0.1
Info: set port to 1234
Info: initialize NLCache : 500000
Info: addviewer StandardViewer
Info: addviewer RelViewer
Info: addviewer FormattedViewer
Info: set TexturePath to /home/secondo/secondo/Data/GuiData/hoesc/textures/
Info: Building nested List from File /home/secondo/secondo/Data/GuiData/hoesc/re
ferences/0snabrueck.ref has taken 17 milliseconds
Info: set bounding box to java.awt.geom.Rectangle2D.Double[x=-50000.0,y=-50000.0
,w=100000.0,h=100000.0]
Info: Building nested List from file /home/secondo/secondo/Data/GuiData/hoesc/ca
tegories/0snabrueck.cat has taken 13 milliseconds
Info: addviewer HoeseViewer
Info: ShowCommand false
Info: use binary lists
Info: Swap texts with length greater than 1024 to file
Info: set objectdirectory to /home/secondo/secondo/Data/GuiData/gui/objects
Info: set historydirectory to /home/secondo/secondo/Data/GuiData/gui/histories
Info: set querydirectory to /home/secondo/secondo/Data/GuiData/gui/queries
Info: set databasedirectory to /home/secondo/secondo/Data/Databases
Info: set snapshotdirectory to /home/secondo/secondo/Data/GuiData/gui/snapshots
Info: optimizer enabled

```



Βήμα 8



Βήμα 9

ΑΝΑΦΟΡΕΣ

1. Ιστοσελίδα Secondo . <http://dna.fernuni-hagen.de/secondo/>
2. Ιστοσελίδα BerlinMOD. <http://dna.fernuni-hagen.de/secondo/BerlinMOD/BerlinMOD.html>
3. Ιστοσελίδα Hermes. <https://hermes-mod.java.net/>
4. Ralf Hartmut Güting, How to Build Your Own Moving Objects Database System, Fernuniversität Hagen
5. Güting, R.H., V.T. de Almeida, and Z. Ding, Modeling and Querying Moving Objects in Networks. Fernuniversität Hagen, Informatik-Report 308, 2004.
6. Güting, R.H., A Query Optimizer for SECONDO. Source Code and Documentation for the SECONDO Optimizer, from December 2002 on, file optimizer.pl within SECONDO.
7. Güting, R.H., Second-Order Signature: A Tool for Specifying Data Models, Query Processing, and Optimization. In: Proc. ACM SIGMOD Conference. Washington, USA, 1993, 277-286.
8. S. Dieker and R. H. Güting. Plug and play with query algebras: Secondo-a generic dbms development environment. In *Proceedings of the 2000 International Symposium on Database Engineering & Applications*, IDEAS '00, pages 380-392, Washington, DC, USA, 2000. IEEE Computer Society.
9. N. Pelekis, E. Frenzos, N. Giatrakos, and Y. Theodoridis. Hermes: aggregative lbs via a trajectory db engine. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, New York, NY, USA, 2008. ACM.
10. N. Pelekis, E. Stefanakis, I. Kopanakis, C. Zotali, M. Voudas, and Y. Theodoridis. Chorochronos.org: A geoportal for movement data and processes. In *Proceedings of the 10th International Conference on Spatial Information Theory*, COSIT '11, 2011.
11. N. Pelekis and Y. Theodoridis. Boosting location-based services with a moving object database engine. In *Proceedings of the 5th ACM international workshop on Data engineering for wireless and mobile access*, MobiDE '06, New York, NY, USA, 2006. ACM.
12. N. Pelekis, Y. Theodoridis, S. Vosinakis, and T. Panayiotopoulos. Hermes - a framework for location-based data management. In *Proceedings of the 10th international conference on Advances in Database Technology*, EDBT'06, pages, Berlin, Heidelberg, 2006. Springer-Verlag.
13. O. Wolfson, S. Chamberlain, S. Dao, L. Jiang, and G. Mendez. Cost and imprecision in modeling the position of moving objects. In ICDE, pages 588-596, 1998.
14. O. Wolfson, A. P. Sistla, S. Chamberlain, and Y. Yesha. Updating and querying databases That track mobile units. *Distributed and Parallel Databases*, 7(3):257-387, 1999.
15. O. Wolfson, B. Xu, S. Chamberlain, and L. Jiang. Moving objects databases: Issues and solutions In *Statistical and Scientific Database Management*, pages 111-122, 1998.
16. A. P. Sistla, O. Wolfson, S. Chamberlain, and S. Dao. Querying the uncertain position of moving objects. *Lecture Notes in Computer Science*, 1399:310-337, 1998.
17. Y. Theodoridis, J. R. O. Silva, and M. A. Nascimento. On the generation of spatiotemporal datasets In *SSD*, pages 147-164, 1999.
18. T. Tzouramanis, M. Vassilakopoulos, and Y. Manolopoulos. the generation of time-evolving regional data, 2002.

19. J.-M. Saglio and J. Moreira. Oporto: A Realistic Scenario Generator for Moving Objects. *Geo-informatica*, 5(1):71–93, 2001.
20. M. Stonebraker, J. Frew, K. Gardels, and J. Meredith. The Sequoia 2000 Benchmark. In *SIGMOD Conference*, 1993.
21. Y. Theodoridis. Ten benchmark database queries for location-based services. *Comput. J.*, 2003.
22. J. Myllymaki and J. Kaufman. Dynamark: A benchmark for dynamic spatial indexing. In *MDM '03: Proceedings of the 4th International Conference on Mobile Data Management*, pages 92–105, London, UK, 2003. Springer-Verlag.
23. C. S. Jensen, D. Tiesyte, and N. Tradisaukas. The COST benchmark - Comparison and evaluation of spatio-temporal indexes. In *DASFAA*, pages 125–140, 2006.
24. C. Duntgen, T. Behr, and R. H. Guting, “BerlinMOD – A Benchmark for Moving Object Databases”, Faculty of Mathematics and Computer Science, Fernuniversität in Hagen, 2007
25. C. Duntgen, T. Behr, and R. H. Guting, “Assessing Representations for moving object histories”, Faculty of Mathematics and Computer Science, Fernuniversität in Hagen, 2010