



**ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ
ΠΟΛΥΤΕΧΝΙΚΗ ΣΧΟΛΗ**

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΑΚΩΝ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

**Ανάλυση της Απόδοσης Χωρικών-Κειμενικών Επερωτήσεων στην
MySQL**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της/του

Κιντής Νικόλαος

Επιβλέπων : Ακριβή Βλάχου

Μέλη εξεταστικής επιτροπής: κα. Κωνσταντίνου , κ. Κωστούλας

Σάμος, [02/2024]

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πρόλογος και ευχαριστίες

Θα ήθελα να ευχαριστήσω την επιβλέπουσα καθηγήτρια κυρία Βλάχου Ακριβή για την βοήθεια και την καθοδήγηση που μου παρείχε κατά την περίοδο της διπλωματικής μου. Επίσης θα ήθελα επίσης να ευχαριστήσω την οικογένεια και τους φίλους μου παρείχαν στήριξη σε όλη την διάρκεια των σπουδών μου.

2024

του/της

Κιντής Νικόλαος

Τμήμα Μηχανικών Πληροφοριακών και Επικοινωνιακών Συστημάτων

ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΙΓΑΙΟΥ

Η σελίδα αυτή είναι σκόπιμα λευκή.

Πίνακας Περιεχομένων

1	Εισαγωγή	1
1.1	Περιγραφή του προβλήματος.....	1
1.2	Αντικείμενο διπλωματικής.....	1
1.3	Δομή της διπλωματικής	1
2	Εισαγωγικές έννοιες.....	2
2.1	Χωρικά δεδομένα	2
2.1.1	Τι είναι χωρικά δεδομένα και γιατί είναι σημαντικά.....	2
2.2	Χωρικά-κειμενικά δεδομένα.....	2
2.2.1	Τι είναι χωρικά δεδομένα και γιατί είναι σημαντικά.....	2
2.2.2	Βασικές χωρικές ερωτήσεις.....	2
2.2.3	Τεχνικές ή ευρετήρια για χωρικά δεδομένα	2
2.3	Σχετική βιβλιογραφία	2
3	MySQL και Spatial MySQL	3
3.1	Explain και πλάνα εκτέλεσης	3
3.2	MySQL για χωρικά δεδομένα.....	3
3.2.1	Τύποι Χωρικών Δεδομένων.....	4
3.2.2	Ερωτήσεις Σημείου.....	4
3.2.3	Ερωτήσεις Πλησιέστερου Γείτονα	4
3.2.4	Ερωτήσεις Ακτίνας.....	4
3.2.5	Ερωτήσεις Παραθύρου	4
4	Πειραματική αποτίμηση	4
Κύριο μέρος διπλωματικής		
4.1	Πειραματική αποτίμηση: Πρώτο Πείραμα	4
4.1.1	Επεξεργασία στη MySQL.....	4
4.1.1.1	Απλή MySQL χωρίς ευρετήρια.....	4
4.1.1.2	Απλή MySQL με ευρετήρια.....	4
4.1.2	Επεξεργασία στη Spatial MySQL.....	4
4.1.3	Αποτελέσματα	4
4.2	Πειραματική αποτίμηση: Δεύτερο Πείραμα.....	4
4.2.1	Επεξεργασία στη MySQL.....	4
4.2.2	Αποτελέσματα	4
5	Συμπεράσματα	6
Βιβλιογραφία		7

Λίστα Εικόνων

Εικόνα 1. R-tree κάτω από την λέξη expensive.....	6
Εικόνα 2. Grid κάτω από την λέξη Italian.....	6
Εικόνα 3. Καμπύλες Hilbert	6
Εικόνα 4. Καμπύλες Z-order – Hilbert.....	6
Εικόνα 5. Διάγραμμα μέσων τιμών αποτελεσμάτων πρώτου πειράματος.....	19
Εικόνα 6. Διάγραμμα μέσων τιμών αποτελεσμάτων δεύτερου πειράματος.....	24

Λίστα Πινάκων

Πίνακας 1. Αποτελέσματα εντολής Explain.....	10
Πίνακας 2. Αποτελέσματα πρώτου ερωτήματος, πρώτου πειράματος.....	17
Πίνακας 3. Αποτελέσματα δεύτερου ερωτήματος, πρώτου πειράματος.....	17
Πίνακας 4. Αποτελέσματα τρίτου ερωτήματος, πρώτου πειράματος.....	18
Πίνακας 5 Μέσες τιμές αποτελεσμάτων πρώτου πειράματος.....	19
Πίνακας 6. Αποτελέσματα πρώτου ερωτήματος, δεύτερου πειράματος.....	21
Πίνακας 7. Αποτελέσματα δεύτερου ερωτήματος, δεύτερου πειράματος.....	22
Πίνακας 8. Αποτελέσματα τρίτου ερωτήματος, δεύτερου πειράματος.....	22
Πίνακας 9. Αποτελέσματα τέταρτου ερωτήματος, δεύτερου πειράματος.....	22
Πίνακας 10. Μέσες τιμές αποτελεσμάτων δεύτερου πειράματος.....	24

Περίληψη

Με αφορμή την αυξανόμενη χρήση χωρικών – κειμενικών δεδομένων στο διαδίκτυο και τον μεγάλο όγκο αυτών των δεδομένων, αυτή η διπλωματική εργασία επιχειρεί να βρει λύσεις στο πρόβλημα διαχείρισης των μεγάλων δεδομένων και επικεντρώνεται στη μελέτη των χρόνων ανταπόκρισης των βάσεων δεδομένων της MySQL στα χωρο-κειμενικά δεδομένα. Αναλύονται παράμετροι και μελετώνται διαφορετικές μέθοδοι ευρετηρίασης με σκοπό την εύρεση της βέλτιστης απόδοσης. Με βάση τα αποτελέσματα, προτείνονται πρακτικές λύσεις και μέθοδοι για την βελτιστοποίηση της απόδοσης των χωρο-κειμενικών δεδομένων στη MySQL. Επιπλέον η εργασία αυτή συμβάλει στη κατανόηση των παραγόντων που επηρεάζουν την απόδοση σε περιβάλλοντα με χωρο-κειμενικά δεδομένα.

Για να πραγματοποιηθεί αυτό χρησιμοποιήθηκαν δυο ξεχωριστά αρχεία με δεδομένα εστιατορίων ενώ για την αποθήκευση, ευρετηρίαση και διαχείριση τους χρησιμοποιήθηκαν τρεις διαφορετικοί μέθοδοι: α) απλή MySQL χωρίς ευρετήρια, β) απλή MySQL με ευρετήρια, γ) Spatial MySQL με ευρετήρια. Για την εκτέλεση των ερωτημάτων της MySQL και την συλλογή των αποτελεσμάτων χρησιμοποιήθηκε Java.

Λέξεις κλειδιά : Χωρο – κειμενικά δεδομένα , MySQL , Spatial MySQL , ευρετήρια(index), λέξεις – κλειδιά, χωρικές ερωτήσεις.

Abstract

In light of the increasing use of spatial and textual data on the internet and the large volume of such data, this thesis attempts to find solutions to the challenge of managing large datasets. It focuses on studying the response times of MySQL databases for spatial-textual data. Parameters are analyzed, and different indexing methods are studied with the aim of achieving optimal performance. Based on the results, practical solutions and methods are proposed to optimize the performance of spatial-textual data in MySQL. Furthermore, this work contributes to understanding the factors that affect performance in environments with spatial-textual data.

To achieve this, two separate datasets of restaurant data were used, and three different methods were employed for storing, indexing, and managing them: a) plain MySQL without indexes, b) plain MySQL with indexes, c) Spatial MySQL with indexes. For executing MySQL queries and collecting results, Java was utilized.

Keywords: *Spatial-textual data, MySQL, Spatial MySQL, index, keywords, spatial queries.*

1

Εισαγωγή

1.1 Καθώς ο όγκος των δεδομένων που παράγονται στον διαδικτυακό χώρο συνεχίζει να αυξάνεται εκθετικά, η κατανόηση και η αξιοποίηση των χωρικών κειμενικών δεδομένων έχουν γίνει ουσιαστικές για διάφορες εφαρμογές και βιομηχανίες. Το διαδίκτυο λειτουργεί ως ένας μεγάλος αποθηκευτικός χώρος πληροφοριών, περιλαμβάνοντας ποικίλες πηγές, όπως κοινωνικά μέσα, ειδήσεις, ιστολόγια και αξιολογήσεις. Εντός αυτής της τεράστιας συλλογής κειμενικού περιεχομένου, οι χωρικές πληροφορίες εμφανίζονται συχνά υπό τη μορφή ετικετών τοποθεσίας, ονομάτων τόπων, διευθύνσεων ή γεωγραφικών αναφορών. Συνδυάζοντας αυτά τα κειμενικά δεδομένα με τις γεωχωρικές πληροφορίες, μπορούμε να αποκτήσουμε μια πιο ολοκληρωμένη κατανόηση του χωρικού περιβάλλοντος στο οποίο βρίσκεται το κείμενο. Οι μεγάλες ποσότητες δεδομένων που δημιουργούνται στον διαδικτυακό χώρο, γνωστές και ως " big data ", αποτελούν μια πολύτιμη πηγή για την ανάλυση του χωρικού περιεχομένου. Η ανάλυση των μεγάλων αυτών συνόλων δεδομένων μας επιτρέπει να ανακαλύψουμε πρότυπα, να προβλέψουμε τάσεις, να προσδιορίσουμε συμπεράσματα και να λάβουμε αποφάσεις βασιζόμενες σε δεδομένα, προωθώντας την επιχειρηματική ευφυΐα και την καινοτομία. Όμως ο μεγάλος όγκος δεδομένων δημιουργεί την ανάγκη για χρήση προηγμένων τεχνικών και εργαλείων διαχείρισης δεδομένων ώστε να μπορεί να αξιοποιηθεί αποδοτικά και αποτελεσματικά.

1.2 *Αντικείμενο διπλωματικής*

Σε αυτήν την εργασία πραγματοποιείται μια πειραματική μελέτη με σκοπό να εξετάσει την απόδοση ερωτήσεων στη MySQL σε χωρικά-κειμενικά δεδομένα χρησιμοποιώντας διαφορετικές τεχνικές αποθήκευσης και ευρετηρίασης των δεδομένων. Τα δεδομένα περιέχουν πληροφορίες της τοποθεσίας (γεωγραφικό πλάτος και γεωγραφικό μήκος) και λέξεις κλειδιά. Οι ερωτήσεις αναζητούν αντικείμενα σύμφωνα με την τοποθεσία τους σε συνδυασμό με τις λέξεις κλειδιά. Η εκτέλεση των πειραμάτων θα πραγματοποιηθεί σε δύο διαφορετικά αρχεία. Για κάθε ένα από τα αρχεία, τα δεδομένα του θα είναι αποθηκευμένα σε τρεις διαφορετικές βάσεις δεδομένων οι οποίες θα διαφέρουν μεταξύ τους στον τρόπο αποθήκευσης και ευρετηρίασης των. Η πρώτη βάση θα αποτελείται από απλή MySQL χωρίς ευρετήρια. Η δεύτερη θα αποτελείται από απλή MySQL με ευρετήρια ενώ η τρίτη από Χωρική (Spatial) MySQL. Για τις βάσεις δεδομένων θα χρησιμοποιηθεί το MySQL Workbench. Για μεταφορά των δεδομένων από τα αρχεία CSV στις βάσεις αλλά και για την εκτέλεση ερωτημάτων, την συλλογή και την επεξεργασία των αποτελεσμάτων θα χρησιμοποιηθεί η γλώσσα προγραμματισμού Java.

1.3 *Δομή της διπλωματικής*

Στο κεφάλαιο 2 της εργασίας αναλύονται βασικές εισαγωγικές έννοιες. Στη συνέχεια στο κεφάλαιο 3 ακολουθεί μια περεταίρω ανάλυση της MySQL και της Spatial MySQL. Στο κεφάλαιο 4 αναλύονται τα πειράματα και τα αποτελέσματα τους.

2

Χωρικά Δεδομένα

2.1 Χωρικά Δεδομένα

Τα χωρικά δεδομένα αποτελούν ένα θεμελιώδες στοιχείο του σύγχρονου κόσμου μας, παρέχοντας πληροφορίες σχετικά με τα φυσικά και γεωγραφικά χαρακτηριστικά του περιβάλλοντός μας. Αναφέρονται σε οποιαδήποτε δεδομένα συνδέονται με μια συγκεκριμένη τοποθεσία στη Γη ή στο χώρο.

Τα χωρικά δεδομένα καλύπτουν μια ευρεία γκάμα πληροφοριών, συμπεριλαμβανομένων γεωγραφικών συντεταγμένων, ορίων, τοπογραφίας, κάλυψης γης, κατανομής πληθυσμού, δικτύων μεταφοράς και πολλά άλλα. Μπορούν να συλλεχθούν από διάφορες πηγές, όπως εικόνες από δορυφόρους, αεροφωτογραφίες, συσκευές Παγκόσμιου Συστήματος Εντοπισμού (GPS), τεχνολογίες απομακρυσμένης ανίχνευσης και γεωγραφικά συστήματα πληροφοριών (GIS)

Τα χωρικά δεδομένα έχουν ζωτικό ρόλο σε πολλές βιομηχανίες και επιστήμες. Οι επιστήμονες του περιβάλλοντος τα χρησιμοποιούν για να παρακολουθούν τα οικοσυστήματα. Οι επιχειρήσεις τα αξιοποιούν για να εντοπίσουν τις τάσεις της αγοράς, να βελτιστοποιήσουν τις αλυσίδες εφοδιασμού και να στοχεύσουν τους πελάτες με βάση τη γεωγραφική τους τοποθεσία.

Για να αξιοποιηθεί η δύναμη των χωρικών δεδομένων, χρησιμοποιούνται εξειδικευμένα εργαλεία και τεχνικές. Το λογισμικό GIS επιτρέπει την καταγραφή, αποθήκευση, επεξεργασία, ανάλυση και οπτικοποίηση των χωρικών δεδομένων. Προηγμένοι αλγόριθμοι και χωρική στατιστική βοηθούν στον εντοπισμό μοτίβων και σχέσεων εντός των συνόλων δεδομένων, ενώ οι τεχνικές οπτικοποίησης δεδομένων παρέχουν ευανάγνωστες και αλληλεπιδραστικές αναπαραστάσεις των γεωχωρικών πληροφοριών.

2.1.1 Χωρικά Δεδομένα Σημείου

Τα σημειακά χωρικά δεδομένα αναφέρονται σε έναν τύπο χωρικών δεδομένων που αναπαριστούν συγκεκριμένες τοποθεσίες ή σημεία στην επιφάνεια της Γης. Κάθε σημείο καθορίζεται από τις γεωγραφικές του συντεταγμένες, συνήθως εκφραζόμενες ως γεωγραφικό πλάτος και μήκος. Τα σημειακά χωρικά δεδομένα χρησιμοποιούνται συνήθως για να αναπαριστούν αντικείμενα στο χώρο, αξιοθέατα ή γεγονότα με συγκεκριμένη τοποθεσία.

Τα σημειακά χωρικά δεδομένα είναι θεμελιώδους σημασίας για τη χωρική ανάλυση, καθώς παρέχουν τη βάση για την κατανόηση της κατανομής, της απόστασης και των σχέσεων μεταξύ διαφορετικών γεωγραφικών στοιχείων.

2.2 Χωρικά Κειμενικά Δεδομένα

2.2.1 Τι είναι τα χωρικά κειμενικά δεδομένα

Τα χωρικά δεδομένα συχνά περιέχουν επιπλέον πληροφορίες από τις συντεταγμένες τοποθεσίας. Τα δεδομένα που συνδυάζουν πληροφορίες που βασίζονται στην τοποθεσία και σε κειμενικό περιεχόμενο ονομάζονται χωρικά-κειμενικά δεδομένα. Ο συνδυασμός των χωρικών δεδομένων με κειμενική περιγραφή μας δίνει μια πιο ολοκληρωμένη εικόνα της γεωγραφικής περιοχής την οποία περιγράφουν. Η χρήση των δεδομένων αυτών έχει ευρύ φάσμα εφαρμογών, από τον τομέα του μάρκετινγκ μέχρι την περιβαλλοντική ανάπτυξη και τον τομέα της έρευνας.

2.2.2 Χωρικές Κειμενικές Ερωτήσεις

Μια χωρική κειμενική ερώτηση αποτελείται από δύο κομμάτια, το χωρικό και το κειμενικό. Το χωρικό ερώτημα περιγράφει ένα συγκεκριμένο γεωγραφικό σημείο που αναζητούμε ή μια γεωγραφική περιοχή από την οποία θέλουμε να κρατήσουμε μόνο τα σημεία που βρίσκονται μέσα σε αυτή. Το κειμενικό ερώτημα αποτελείται από τις λέξεις κλειδιά τις οποίες δίνει ο χρήστης προκειμένου να χρησιμοποιηθούν σαν φίλτρο ώστε να βρεθούν τα στοιχεία τα οποία περιέχουν τα ίδια κειμενικά ορίσματα.

Βασικές Χωρικές Κειμενικές Ερωτήσεις

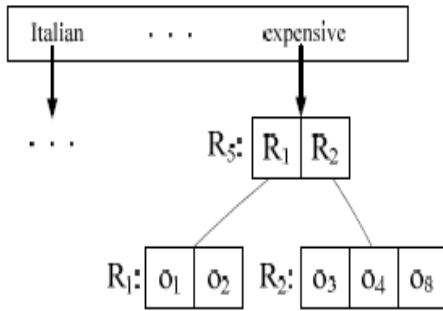
1. **Boolean-Boolean (BB) Query:** Δεδομένου ενός ερωτήματος $q=(r,w)$ όπου το r συμβολίζει τον χώρο αναζήτησης (συνήθως είναι ένας κυκλικός ή ορθογώνιος χώρος) και το w συμβολίζει ένα σύνολο από λέξεις κλειδιά, επιστρέφονται όλα τα δεδομένα τα οποία βρίσκονται μέσα στο χώρο του r και περιέχουν όλες τις λέξεις κλειδιά από το w .
2. **Boolean-Ranking (BR) Query:** Το Boolean-Ranking Query $q=(p, w, k)$, αποτελεί ένα ερώτημα με τρία στοιχεία όπου το p είναι ένα χωρικό σημείο, το w είναι ένα σύνολο από λέξεις κλειδιά και το k συμβολίζει τον αριθμό αντικειμένων που θα επιστρέψει το ερώτημα και επιστρέφεται ένα k -υποσύνολο, το οποίο αποτελείται από στοιχεία τα οποία βρίσκονται στον χώρο p , περιέχουν όλες τις λέξεις κλειδιά του w και το κάθε στοιχείο είναι ο k κοντινότερος γείτονας του q .
3. **Full-Ranking (FR) Query:** Το Full-Ranking Query $q=(p, w, k, \alpha)$, αποτελεί ένα ερώτημα με τέσσερα στοιχεία όπου το p είναι ένα χωρικό σημείο, το w είναι ένα σύνολο από λέξεις κλειδιά, το k συμβολίζει τον αριθμό αντικειμένων που θα επιστρέψει το ερώτημα και το α είναι μια παράμετρος η οποία εξισορροπεί το βάρος μεταξύ της κειμενικής συνάφειας και της χωρικής εγγύτητας. Το ερώτημα q επιστρέφει τα k στοιχεία τα οποία ταξινομήθηκαν σύμφωνα με τα κριτήρια του α .

Πιο συγκεκριμένα η συνάφεια μεταξύ ενός στοιχείου i και ενός ερωτήματος FR q ορίζεται από τη σχέση: $\mathbf{Rel}(i, q) = F(\mathbf{SRel}(i, q), \mathbf{TRel}(i, q))$ όπου το $\mathbf{SRel}(i, q)$ υποδηλώνει την χωρική συνάφεια μεταξύ i και q και υπολογίζεται ως: $\mathbf{SRel}(i, q) = G(\text{dist}(i.\text{loc}, q.p))$ Όπου το G είναι μια μονότονη φθίνουσα συνάρτηση που κυμαίνεται μεταξύ 0 και 1 και το $\text{dist}(i.\text{loc}, q.p)$ είναι η ευκλείδεια απόσταση μεταξύ i και q . Το $\mathbf{TRel}(i, q)$ υποδηλώνει την κειμενική συνάφεια μεταξύ του στοιχείου i και του ερωτήματος w . Η συνάρτηση F είναι μια μονότονη αύξουσα συνάρτηση και στο x και στο y και κυμαίνεται μεταξύ του 0 και του 1.

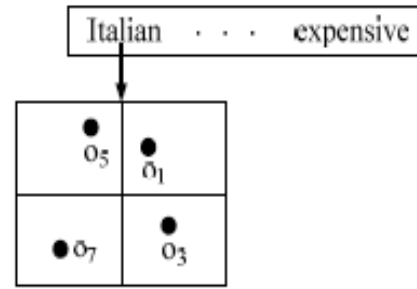
2.2.3 Τεχνικές και ευρετήρια για χωρικά δεδομένα

Ο συνεχής πολλαπλασιασμός των διαδικτυακών αντικειμένων τα οποία σχετίζονται με γεωγραφική τοποθεσία αλλά και με κειμενική περιγραφή οδηγεί σε μια ραγδαία αύξηση του όγκου των δεδομένων. Αυτή η αύξηση έχει φέρει ως αποτέλεσμα να απαιτούνται τεχνικές για την ευρετηρίαση των γεωκειμενικών δεδομένων. Από τα ευρετήρια τα οποία έχουν προταθεί μπορούμε να ξεχωρίσουμε τρεις βασικές κατηγορίες: (1) R-tree based indices, (2) grid bases indices, (3) space filling curve indices.

1. **R-tree based:** Αυτή η κατηγορία ευρετηρίων χρησιμοποιεί το R-tree ή κάποια παραλλαγή του. Σε αυτή την κατηγορία ανήκει ο μεγαλύτερος αριθμός χωρικών ευρετηρίων και για την κειμενική ευρετηρίαση κάνουν χρήση των ανεστραμμένων αρχείων. Στην αρχή τα τα R-tree based ευρετήρια χρησιμοποιούσαν τα ανεστραμμένα αρχεία και το R-tree για να ξεχωρίσουν τα χωρικά από τα κειμενικά δεδομένα. Αντίθετος, πλέον τα σύγχρονα ευρετήρια οργανώνουν μαζί τα χωρο-κειμενικά δεδομένα συνδιάζοντας το R-tree με κειμενικά ευρετήρια.
2. **Grid based:** Αυτή η κατηγορία ευρετηρίων χρησιμοποιεί συνδυαστικά ευρετήρια τύπου Grid με ευρετήρια κειμένου. Το ευρετήριο Grid διαμερίζει τον χώρο σε ένα προκαθορισμένο αριθμό τετραγώνων ή ορθογώνιων κελιών, το καθένα ίσου μεγέθους. Το ευρετήριο Grid μπορεί να χρησιμοποιηθεί είτε ξεχωριστά είτε συνδυαστικά με το ευρετήριο κειμένου.
3. **Space filling curve based:** Αυτά τα ευρετήρια συνδυάζουν ανεστραμμένα αρχεία με μια καμπύλη πλήρωσης χώρου, και περιλαμβάνουν ένα ευρετήριο βασισμένο στην καμπύλη Hilbert και ένα βασισμένο στην καμπύλη Z. Τα ευρετήρια αυτά βασίζονται στον χώρο που δείχνει κοντά το ένα στο άλλο στον εγγενή χώρο (native space) και είναι επίσης κοντά το ένα στο άλλο στην καμπύλη πλήρωσης χώρου.

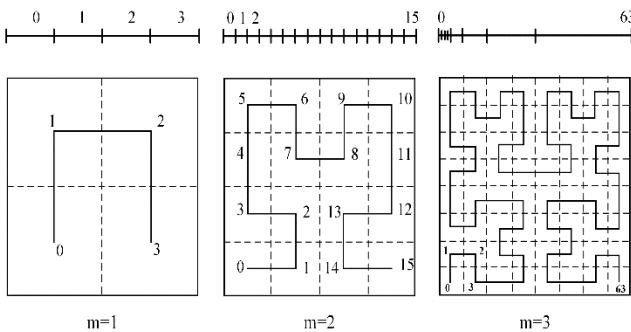


Εικόνα 1. R-tree κάτω από την λέξη expensive.

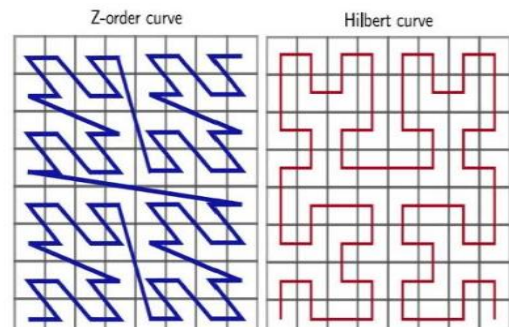


Εικόνα 2. Grid κάτω από την λέξη Italian.

Πηγή: Spatial Keyword Query Processing: An Experimental Evaluation



Εικόνα 3. Καμπύλες Hilbert



Εικόνα 4. Καμπύλες Z-order - Hilbert

Πηγή: A Parallel N-Dimensional Space-Filling Curve Library and Its Application in Massive Point Cloud Management

***Ανεστραμμένο ευρετήριο:** Το ανεστραμμένο ευρετήριο είναι μια δομή δεδομένων που χρησιμοποιείται σε συστήματα ανάκτησης πληροφοριών για την αποτελεσματική ανάκτηση εγγράφων που περιέχουν κάποιον συγκεκριμένο όρο ή ένα σύνολο όρων. Σε ένα ανεστραμμένο ευρετήριο, το ευρετήριο οργανώνεται με βάση τους όρους, δηλαδή τις λέξεις που υπάρχουν ως κείμενα δεδομένα. Κάθε όρος σχετίζεται με μια ανεστραμμένη λίστα εγγράφων και κάθε έγγραφο περιλαμβάνει τις τοποθεσίες όπου εμφανίζεται ο συγκεκριμένος όρος. Κάθε ανεστραμμένη λίστα περιλαμβάνει μια ακολουθία αναρτήσεων, η κάθε μια από αυτές περιέχει το αναγνωριστικό ενός αντικειμένου και η περιγραφή του περιέχει τον όρο και την συχνότητα του. Οι αναρτήσεις κάθε ανεστραμμένης λίστας οργανώνονται σύμφωνα με το αναγνωριστικό του αντικειμένου. Ωστόσο, ορισμένα γεωκειμενικά ευρετήρια ταξινομούν τις αναρτήσεις με διαφορετικό τρόπο, όπως να τις ταξινομούν με τη σειρά τους σε κελιά **Grid** ή να τις ταξινομούν σύμφωνα με μια καμπύλη πλήρωσης χώρου (Hilbert curve).

3

MySQL και Spatial MySQL

3.1 Explain και πλάνα εκτέλεσης

Το explain χρησιμοποιείται για τη λήψη του σχεδίου εκτέλεσης ενός ερωτήματος παρέχοντας πληροφορίες σχετικά με τον τρόπο με τον οποίο η MySQL εκτελεί τις δηλώσεις. Το explain χρησιμοποιείται μαζί με τις εντολές select, delete, update, insert, replace. Χρησιμοποιώντας τον βελτιστοποιητή της, η MySQL παρέχει πληροφορίες σχετικά με το πλάνο εκτέλεσης των εντολών. Από τη στιγμή που εκτελείται μια δήλωση, η MySQL μας δείχνει το πώς προτίθεται να επεξεργαστεί τη δήλωση αυτή, συμπεριλαμβανομένων λεπτομερειών όπως τον τρόπο ένωσης των πινάκων και τη σειρά με την οποία γίνεται αυτό. Χρησιμοποιώντας το explain μπορούμε να καταλάβουμε σε πια tables χρειάζεται να προστεθούν ευρετήρια ώστε η εκτέλεση να γίνεται ταχύτερα. Επιπλέον με την χρήση του explain μπορούμε να καταλάβουμε αν το εργαλείο του optimizer έχει επιλέξει την βέλτιστη λύση.

Το explain επιστρέφει τις παρακάτω στήλες:

- **id:** ένα αναγνωριστικό για κάθε SELECT μέσα στο ερώτημα
- **select_type:** ο τύπος του ερωτήματος SELECT. Πιθανά αποτελέσματα είναι:
 1. SIMPLE: το ερώτημα αποτελείται από ένα ερώτημα SELECT χωρίς άλλα υποερωτήματα ή UNION.
 2. PRIMARY: το SELECT βρίσκεται στο εξωτερικό ερώτημα ενός JOIN.
 3. DERIVED: το SELECT βρίσκεται μέσα σε ένα δευτερεύον ερώτημα το οποίο είναι μέσα σε μια FROM.
 4. SUBQUERY: το πρώτο SELECT ενός υποερωτήματος.
 5. UNCACHEABLE SUBQUERY: ένα δευτερεύον ερώτημα που δεν μπορεί να αποθηκευτεί προσωρινά (υπάρχουν ορισμένες προϋποθέσεις για να έχει ένα ερώτημα τη δυνατότητα αποθήκευσης στην προσωρινή μνήμη).
 6. DEPENDENT SUBQUERY: πρόκειται για ένα υποερώτημα το οποίο εξαρτάται από το εξωτερικό ερώτημα.

7. UNION: το SELECT δεν είναι η πρώτη αλλά μια μεταγενέστερη δήλωση μιας ΕΝΩΣΗΣ(UNION).
8. DEPENDENT UNION: δεν είναι η πρώτη αλλά μια μεταγενέστερη SELECT μιας UNION η οποία εξαρτάται από ένα εξωτερικό ερώτημα.
9. UNION RESULT: το SELECT είναι αποτέλεσμα ΕΝΩΣΗΣ

- **table:** ο πίνακας που αναφέρεται από τη σειρά
- **type:** πώς η MySQL ενώνει τους πίνακες που χρησιμοποιούνται. Αυτό το πεδίο είναι από τα πιο σημαντικά πεδία, καθώς μπορεί να μας βοηθήσει να αντιληφθούμε ότι λείπουν ευρετήρια ή ότι ο τρόπος με τον οποίο το ερώτημα έχει συνταχθεί πρέπει να επανεξεταστεί.

Πιθανά αποτελέσματα είναι:

1. **system:** ο πίνακας έχει μόνο μηδέν ή μία σειρά.
2. **const:** στο πίνακα αντιστοιχεί μόνο μία σειρά η οποία είναι ευρετηριασμένη. Αυτός είναι ο χρονικά βέλτιστος τύπος join, καθώς ο πίνακας πρέπει να διαβαστεί μόνο μία φορά και η τιμή της στήλης μπορεί να αντιμετωπιστεί ως σταθερά κατά την ένωση άλλων πινάκων.
3. **ref:** όλες οι αντίστοιχες σειρές μιας ευρετηριασμένης στήλης διαβάζονται για κάθε συνδυασμό γραμμών από τον προηγούμενο πίνακα. Αυτός ο τύπος join εμφανίζεται για ευρετηριασμένες στήλες συγκριτικά με τη χρήση τελεστών = ή <=>.
4. **eq_ref:** όλα τα μέρη ενός ευρετηρίου χρησιμοποιούνται από την ένωση και το ευρετήριο είναι PRIMARY KEY ή UNIQUE NOT NULL. Αυτός είναι ο επόμενος καλύτερος δυνατός τύπος join.
5. **ref_or_null:** είναι το ίδιο με το ref, αλλά επιπλέον περιέχει γραμμές με μηδενική τιμή για τη στήλη.
6. **fulltext:** η ένωση χρησιμοποιεί το ευρετήριο FULLTEXT του πίνακα.
7. **index_merge:** Το JOIN χρησιμοποιεί μια λίστα από δείκτες για τη δημιουργία του συνόλου αποτελεσμάτων. Η στήλη κλειδιού στην έξοδο της εντολής EXPLAIN θα περιέχει τα κλειδιά που χρησιμοποιούνται για την εκτέλεση της σύζευξης.
8. **unique_subquery:** ένα υπό ερώτημα IN επιστρέφει μόνο ένα αποτέλεσμα από τον πίνακα και χρησιμοποιεί το πρωτεύον κλειδί.
9. **index_subquery:** το ίδιο με το unique_subquery αλλά επιστρέφει περισσότερες από μία σειρές αποτελεσμάτων.
10. **range:** Ένα ευρετήριο χρησιμοποιείται για την αναζήτηση σειρών που εμπίπτουν σε ένα συγκεκριμένο εύρος. Συνήθως, αυτό συμβαίνει όταν η τιμή στη στήλη του κλειδιού συγκρίνεται με μια σταθερά χρησιμοποιώντας τελεστές όπως BETWEEN, IN, >, >= κ.λπ.
11. **index:** Ολόκληρο το δέντρο των ευρετηρίων σαρώνεται προσεκτικά προκειμένου να εντοπιστούν οι αντίστοιχες σειρές. Αυτή η διαδικασία είναι χρονοβόρα και συνήθως υποδηλώνει την έλλειψη κατάλληλων ευρετηρίων στη δομή των δεδομένων.
12. **all :** Ο πίνακας σαρώνεται εκτενώς προκειμένου να εντοπιστούν οι κατάλληλες σειρές για το join. Αυτό το join θεωρείται ανεπιθύμητο καθώς υπονοεί την ανεπάρκεια κατάλληλων ευρετηρίων στον πίνακα.

- **possible_keys:** Αυτή η στήλη δείχνει τα κλειδιά που μπορούν να χρησιμοποιηθούν από τη MySQL για την αναζήτηση σειρών στον πίνακα. Είναι σημαντικό να σημειωθεί ότι αυτά τα κλειδιά ενδέχεται να μην χρησιμοποιούνται πάντα στην πράξη. Παρόλα αυτά, η πληροφορία αυτή μπορεί συχνά να βοηθήσει στη βελτιστοποίηση των ερωτημάτων. Εάν η στήλη είναι NULL, υποδηλώνει ότι δεν υπάρχουν σχετικά ευρετήρια που μπορούν να χρησιμοποιηθούν για τη συγκεκριμένη αναζήτηση.
- **key:** Αυτή η στήλη υποδεικνύει το πραγματικό ευρετήριο που χρησιμοποιείται από τη MySQL. Μπορεί να περιέχει ένα ευρετήριο που δεν αναγράφεται στη στήλη possible_key. Ο βελτιστοποιητής της MySQL πάντα αναζητά το βέλτιστο ευρετήριο που μπορεί να χρησιμοποιηθεί για το συγκεκριμένο ερώτημα. Κατά τη σύνδεση πολλαπλών πινάκων, μπορεί να εντοπίσει κλειδιά που δεν αναφέρονται στη στήλη possible_key, αλλά είναι καλύτερα για την εκτέλεση του ερωτήματος.
- **key_len:** Αυτή η στήλη υποδεικνύει το μήκος του ευρετηρίου που επέλεξε να χρησιμοποιήσει ο Query Optimizer. Παρέχει πληροφορίες σχετικά με το πόσα bytes του ευρετηρίου χρησιμοποιούνται για την αναζήτηση στο ευρετήριο. Για παράδειγμα, μια τιμή key_len 4 σημαίνει ότι απαιτεί μνήμη για την αποθήκευση τεσσάρων χαρακτήρων.
- **ref:** Εμφανίζει τις στήλες ή τις σταθερές που συγκρίνονται με το ευρετήριο που αναφέρεται στη στήλη κλειδιού. Η MySQL μπορεί να επιλέξει είτε μια σταθερή τιμή για σύγκριση είτε μια στήλη, ανάλογα με το σχέδιο εκτέλεσης που επιλέγει για το συγκεκριμένο ερώτημα. Ο βελτιστοποιητής της MySQL προσπαθεί να επιλέξει το πιο αποδοτικό σχέδιο εκτέλεσης, λαμβάνοντας υπόψη τα στατιστικά του πίνακα, τα ευρετήρια και άλλους παράγοντες
- **rows:** Αυτή η στήλη μας δίνει τον αριθμό των εγγραφών που ελέγχθηκαν για την παραγωγή της εξόδου.
- **Filtered:** αντικατοπτρίζει ένα εκτιμώμενο ποσοστό σειρών του πίνακα που υποβάλλονται σε φιλτράρισμα μέσω των συνθηκών του πίνακα. Η μέγιστη τιμή είναι 100, δείχνοντας ότι δεν υπάρχει κανένα φιλτράρισμα των σειρών. Αντίθετα, οι τιμές που μειώνονται κάτω από το 100 υποδεικνύουν το εκτιμώμενο ποσοστό των σειρών που υποβάλλονται σε αυξανόμενα επίπεδα φιλτραρίσματος, με το 0 να υποδηλώνει ότι όλες οι σειρές πληρούν τις συνθήκες.
- **Extra:** περιλαμβάνει επιπλέον πληροφορίες που σχετίζονται με το σχέδιο εκτέλεσης του ερωτήματος. Τιμές όπως " Using filesort ", " Using temporary " κ.λπ. ενδέχεται να εμφανίζονται σε αυτήν τη στήλη και μπορεί να υποδεικνύουν προβλήματα που πιθανόν να υπάρχουν στο ερώτημα, όπως η χρήση προσωρινών πινάκων ή η ανάγκη για ταξινόμηση αρχείων.

Παράδειγμα.

Εκτελούμε την παρακάτω ερώτηση

Keyword: Italian, Dinner, Greek

Point: Longitude=30, Latitude=-70, Ακτίνα=20

```
EXPLAIN SELECT
  kl.id, p.onoma
from kleidia kl
INNER JOIN point AS p
  ON kl.id = p.id
INNER JOIN keyword AS r
  ON r.idrest=kl.idrest
WHERE r.keyword in('Italian','Diner','Greek') AND p.longitude<=30+20 AND p.longitude>=30-20
AND p.latitude<=-70+20 AND p.latitude>=-70-20
group by kl.id
HAVING COUNT(*) > 2;
```

id	Select_type	table	partitions	type	Possible_keys	key	Key_len	ref	rows	Filtered	extra
1	Simple	P	NULL	ALL	PRIMARY,longitudeIndex,latitudeIndex	NULL	NULL	NULL	75427	25.00	Using where; Using temporary
1	Simple	kl	NULL	Ref	idindex,idrestindex	idindex	4	rest.p.id	3	100.00	NULL
1	Simple	r	NULL	eq_ref	Primary	Primary	4	rest.kl.id	1	30.00	Using Where

Πίνακας 1. Αποτελέσματα εντολής Explain

Αποτελέσματα

Select_type : παρατηρούμε ότι το ερώτημα είναι ένα απλό ερώτημα χωρίς υποερωτήματα ή Union.

Type : Για το table p παρατηρούμε ότι ο τύπος είναι ALL κάτι που σημαίνει ότι σαρώνεται ολόκληρος ο πίνακας για να βρεθούν οι αντίστοιχες σειρές. Αυτός ο τύπος join θεωρείται ο χειρότερος και μας υποδηλώνει ότι στον πίνακα δεν υπάρχουν τα κατάλληλα ευρετήρια. Για το table kl ο τύπος είναι Ref που σημαίνει ότι όλες οι αντίστοιχες σειρές μιας ευρετηριασμένης στήλης διαβάζονται για κάθε συνδυασμό γραμμών από τον προηγούμενο πίνακα. Για το table r ο τύπος είναι eq_ref, αυτό σημαίνει ότι όλα τα μέρη ενός ευρετηρίου χρησιμοποιούνται από την ένωση και το ευρετήριο είναι PRIMARY KEY ή UNIQUE NOT NULL. Αυτός είναι ο επόμενος καλύτερος δυνατός τύπος join μετά τον const

Possible_keys: Σε αυτή την στήλη μπορούμε να δούμε κάθε ευρετήριο που μπορεί να χρησιμοποιηθεί σε κάθε table.

Key: Σε αυτή την στήλη μπορούμε να δούμε κάθε ευρετήριο που χρησιμοποιήθηκε σε κάθε table.

Key_len: Από αυτή τη στήλη μπορούμε να δούμε το μήκος του ευρετηρίου που επιλέχθηκε να χρησιμοποιηθεί από το Query Optimizer. Παρατηρείται πως για τα table kl και r απαιτείτε μνήμη τεσσάρων χαρακτήρων για να την αποθήκευση ενώ για το table p η τιμή είναι NULL καθώς δεν χρησιμοποιήθηκε κάποιο ευρετήριο.

Rows : Αυτή η στήλη δείχνει πόσες σειρές εξετάστηκαν από κάθε table. Παρατηρούμε ότι εξετάστηκαν όλες οι στήλες του talbe p ενώ από το table kl μόνο τρεις, όσα δηλαδή είναι και τα keyword του ερωτήματος.

Filtered: Παρατηρούμε ότι για το table kl δεν έχει γίνει κανένα φιλτράρισμα.

3.2 MySQL για χωρικά δεδομένα.

Με σκοπό να επιτευχθεί καλύτερη διαχείριση των χωρικών δεδομένων η MySQL εφαρμόζει ως ένα υποσύνολο της SQL επεκτάσεις για χωρικά δεδομένα με ένα περιβάλλον γεωμετρικών τύπων. Έτσι υπάρχει ένα σύνολο από γεωμετρικούς τύπους της SQL και λειτουργίες αυτών των τύπων για την δημιουργία και την ανάλυση γεωμετρικών τιμών.

Μέσω της χρήσης των χωρικών επεκτάσεων της SQL γίνεται δυνατή η δημιουργία, η αποθήκευση και η ανάλυση κάθε είδους γεωγραφικού χαρακτήρα.

Πιο συγκεκριμένα οι επεκτάσεις αυτές προσφέρουν:

1. Ξεχωριστούς τύπους για τις χωρικές τιμές
2. Ειδικές λειτουργίες για τον χειρισμό χωρικών τιμών
3. Χωρικά ευρετήρια (spatial index) για την βελτίωση της ταχύτητας πρόσβασης σε χωρικές στήλες

3.2.1 Τύποι Χωρικών Δεδομένων

Στη spatial MySQL υπάρχουν τύποι δεδομένων όπου μπορούν να δεχθούν μόνο μονές τιμές και τύποι όπου μπορούν να δεχθούν πολλαπλές τιμές. Ο τύπος χωρικών δεδομένων που θα μας απασχολήσει είναι το Point και πρόκειται για τύπο που δέχεται μόνο μονές τιμές.

3.2.2 Ερωτήσεις Σημείου

Σημείο είναι ένα γεωμετρικό αντικείμενο που αποτελεί ένα σημείο στον χώρο. Κάθε σημείο έχει τις παρακάτω ιδιότητες :

- Τιμή συντεταγμένης πλάτους.
- Τιμή συντεταγμένης μήκους.
- Το όριο του σημείου είναι το κενό.
- Το σημείο ορίζεται ως μια μηδενική γεωμετρία, δηλαδή δεν έχει μήκος, ύψος και βάθος.

Στα σημειακά αντικείμενα το γεωγραφικό μήκος μπορεί να ληφθεί χρησιμοποιώντας τη συνάρτηση ST_Longitude() ενώ το πλάτος μπορεί να ληφθεί χρησιμοποιώντας τη συνάρτηση ST_Latitude(). Επιπλέον αυτές οι συναρτήσεις επιτρέπουν ένα προαιρετικό δεύτερο όρισμα το

οποίο ορίζει μια τιμή γεωγραφικού μήκους ή πλάτους αντίστοιχα, οπότε το αποτέλεσμα της συνάρτησης είναι το αντικείμενο Point από το πρώτο όρισμα με το γεωγραφικό μήκος ή πλάτος τροποποιημένο ώστε να είναι ίσο με το δεύτερο όρισμα. Ο τύπος Point για τον τύπο γεωγραφικών δεδομένων αντιπροσωπεύει μια μεμονωμένη θέση όπου το Lat αντιπροσωπεύει το γεωγραφικό πλάτος και το Long το γεωγραφικό μήκος. Οι τιμές για το γεωγραφικό πλάτος και μήκος μετρούνται σε μοίρες. Οι τιμές για το γεωγραφικό πλάτος βρίσκονται στο διάστημα $[-90, 90]$ και οι τιμές που εισάγονται εκτός αυτού του εύρους δημιουργούν exception. Οι τιμές για το γεωγραφικό μήκος βρίσκονται πάντα στο διάστημα $(-180, 180]$, και κάθε τιμή που εκτός αυτού του εύρους προσαρμόζεται για να χωρέσει σε αυτό το εύρος.

Οι λειτουργίες που χρησιμοποιούνται για την λήψη των ιδιοτήτων ενός σημείου είναι:

- **ST_Latitude(p [, new_latitude_val]):** Χρησιμοποιείται ένα μόνο όρισμα που αντιπροσωπεύει ένα αντικείμενο Point p και η λειτουργία ST_Latitude() μας επιστρέφει την τιμή του γεωγραφικού πλάτους του p σε μορφή double (αριθμό διπλής ακρίβειας). Το δεύτερο όρισμα είναι προαιρετικό και αντιπροσωπεύει μια τιμή γεωγραφικού πλάτους, η ST_Latitude() επιστρέφει ένα αντικείμενο Point όπως το πρώτο όρισμα με το γεωγραφικό πλάτος του ίσο με το δεύτερο όρισμα.
- **ST_Longitude(p [, new_longitude_val]):** Χρησιμοποιείται ένα μόνο όρισμα που αντιπροσωπεύει ένα αντικείμενο Point p και η λειτουργία ST_Longitude() μας επιστρέφει την τιμή του γεωγραφικού μήκους του p σε μορφή double (αριθμό διπλής ακρίβειας). Το δεύτερο όρισμα είναι προαιρετικό και αντιπροσωπεύει μια τιμή γεωγραφικού μήκους, η ST_Longitude() επιστρέφει ένα αντικείμενο Point όπως το πρώτο όρισμα με το γεωγραφικό μήκος του ίσο με το δεύτερο όρισμα.

3.2.3 Ερωτήσεις Πλησιέστερου Γείτονα

το ερώτημα του πλησιέστερου γείτονα είναι εάν ερώτημα που συναντάτε συχνά σε ερωτήσεις με χωρικά δεδομένα. Σκοπός αυτού του ερωτήματος είναι η εύρεση του πλησιέστερου χωρικού αντικειμένου μέσα σε ένα συγκεκριμένο χώρο. Για παράδειγμα, ένας εντοπιστής εστιατορίων σε μια τοποθεσία Web καλείτε να βρει τα εστιατόρια τα οποία βρίσκονται πλησιέστερα στη τοποθεσία ενός πελάτη.

Για την εκτέλεση των ερωτημάτων πλησιέστερου γείτονα σε στήλες χωρικών δεδομένων στη MySQL χρησιμοποιούνται οι όροι ORDER BY και TOP. Ο όρος TOP χρησιμοποιείται για να δηλώσουμε τον αριθμό αντικειμένων που πρέπει να επιστραφούν για το ερώτημα. Ο όρος ORDER BY καλεί την μέθοδο ST_Distance() ώστε να επιστραφούν τα χωρικά δεδομένα με κριτήριο την απόσταση τους από το επιλεγμένο σημείο.

Για να εκτελεστεί το ερώτημα πλησιέστερου γείτονα χρησιμοποιώντας ένα χωρικό ευρετήριο θα πρέπει να πληρούνται οι παρακάτω απαιτήσεις:

- Σε μια από τις στήλες που περιέχουν τα χωρικά δεδομένα θα πρέπει να περιέχεται και ένα χωρικό ευρετήριο. Επιπλέον αυτή η στήλη θα πρέπει να χρησιμοποιείται από τη μέθοδο ST_Distance() στους όρους WHERE και ORDER BY.
- Στον όρο WHERE θα πρέπει να περιέχεται μια μέθοδος ST_Distance().
- Εάν υπάρχουν πολλαπλοί όροι στη δήλωση WHERE, τότε ο όρος που περιέχει τη μέθοδο ST_Distance() πρέπει να συνδεθεί με έναν σύνδεσμο AND με τους άλλους όρους. Η

μέθοδος ST_Distance() δεν μπορεί να βρίσκεται σε κάποιο προαιρετικό μέρος της δήλωσης WHERE.

- Ο όρος TOP δεν μπορεί να περιέχει ποσοστό στη δήλωση του.
- Η μέθοδος ST_Distance() πρέπει να χρησιμοποιείται στον πρώτο όρο της έκφρασης ORDER BY.
- Κάθε σειρές την οποία επιστρέφει ως NULL η μέθοδος ST_Distance() πρέπει να φιλτραρισθεί και να αφαιρεθεί.
- Η ταξινόμηση της πρώτης έκφραση ST_Distance() στον όρο ORDER BY πρέπει να είναι ASC.

3.2.4 Ερωτήσεις Ακτίνας

Η ερώτηση ακτίνας αναφέρεται σε μια επερώτηση που επιστρέφει όλες τις εγγραφές ενός πίνακα μιας βάσης δεδομένων που βρίσκονται εντός μιας συγκεκριμένης απόστασης σε ακτίνα από ένα κεντρικό σημείο.

Για παράδειγμα, σε μια βάση δεδομένων με χωρικά δεδομένα τα οποία περιέχουν γεωγραφικές συντεταγμένες (πλάτος, μήκος) για γεωγραφικά σημεία, μια ερώτηση ακτίνας θα μπορούσε να ήταν “Βρείτε όλες τις τοποθεσίες που βρίσκονται εντός δυο χιλιομέτρων από ένα συγκεκριμένο γεωγραφικό σημείο”

Για την εκτέλεση αυτών των ερωτήσεων συνήθως χρησιμοποιούμε λ

λειτουργίες γεωχωρικών δεδομένων της Spatial MySQL, όπως λειτουργίες που παρέχονται για τον τύπο δεδομένων ‘GEOMETRY’ και μεθόδους όπως η ST_Distance().

3.2.5 Ερωτήσεις Παραθύρου

Μια συνάρτηση παραθύρου εκτελεί μια λειτουργία που μοιάζει αθροιστική, σε ένα σύνολο σειρών ερωτημάτων. Ωστόσο, μια αθροιστική λειτουργία θα συγκέντρωνέ τις σειρές ερωτημάτων σε μια ενιαία σειρά αποτελεσμάτων, αντίθετος μια ερώτηση παραθύρου ομαδοποιεί για κάθε μια σειρά ερωτήματος από ένα αποτέλεσμα. Η λειτουργία παραθύρου σε ένα ερώτημα δηλώνεται με τη συμπερίληψη μιας OVER() που καθορίζει τον τρόπο διαχωρισμού των σειρών ερωτήματος σε ομάδες για επεξεργασία από τη συνάρτηση παραθύρου. Οι χρήση των ερωτήσεων παραθύρου επιτρέπεται μόνο στο Select και στην συνθήκη ORDER BY. Οι σειρές αποτελεσμάτων του ερωτήματος καθορίζονται από τον όρο FROM, μετά το WHERE, GROUP BY και HAVING και η εκτέλεση παραθύρου πραγματοποιείται πριν από τα ORDER BY, LIMIT και SELECT DISTINCT.

Για παράδειγμα, Υποθέτοντας ότι έχουμε ένα πίνακα με το όνομα sales που περιλαμβάνει τα πεδία date και amount, και θέλουμε να υπολογίσουμε το σύνολο του πεδίου amount ταξινομημένο βάσει του πεδίου date:

```
SELECT
  date, amount,
  SUM(amount) OVER (ORDER BY date) AS running_total
FROM
  sales;
```

Το SUM(amount) είναι η παραθυρική συνάρτηση που υπολογίζει το άθροισμα του πεδίου amount.

Το OVER (ORDER BY date) καθορίζει το παράθυρο, υποδεικνύοντας ότι το άθροισμα θα πρέπει να υπολογιστεί με βάση τη σειρά του πεδίου date.

Το αποτέλεσμα θα περιλαμβάνει τα αρχικά πεδία (date και amount) μαζί με ένα επιπλέον πεδίο running_total, που αντιπροσωπεύει το σύνολο του πεδίου amount όσο οι εγγραφές επεξεργάζονται με βάση την καθορισμένη σειρά.

4

Πειραματική αποτίμηση.

4.1 Πειραματική αποτίμηση: Πρώτο Πείραμα

4.1.1 Επεξεργασία στη MySQL

Αρχικά στην παρούσα διπλωματική εργασία για το πρώτο πείραμα έγινε χρήση ενός αρχείου από το factual.com, μέσα στο οποίο εμπεριέχονται οι εγγραφές των εστιατορίων. Το συγκεκριμένο αρχείο είναι της μορφής txt, ενώ κάθε εγγραφή καταλαμβάνει μία γραμμή . Στο αρχείο αναπαρίστανται συνολικά 78.981 εστιατόρια. Οι εγγραφές του αρχείου αναπαριστούν πραγματικά δεδομένα και αφορούν επιχειρήσεις που βρίσκονται στις ΗΠΑ.

- 1) Ονομασία..
- 2) τοποθεσία σε γεωγραφικό μήκος και πλάτος.
- 3) Κάποια επιπλέον χαρακτηριστικά σχετικά με τις παροχές του καταστήματος. Τα χαρακτηριστικά αυτά θα χρησιμοποιηθούν ως λέξεις κλειδιά στα ερωτήματα της MySQL.

Παράδειγμα γραμμής: 78965,Scotty's Bar and Grill,5,47.677876,-117.28277,American Burgers Grill Pub Food

Από την παραπάνω στήλη καταλαβαίνουμε ότι

- 1) Όνομα του καταστήματος είναι : Scotty's Bar and Grill.
- 2) Τοποθεσία : Γεωγραφικό πλάτος = 47.677876 , Γεωγραφικό μήκος = ,-117.28277.
- 3) Λέξεις κλειδιά : American / Burgers / Grill / Pub / Food

Στη MySQL έχουμε δημιουργήσει τρεις διαφορετικές βάσεις δεδομένων. Η πρώτη βάση χρησιμοποιεί απλή MySQL χωρίς να κάνει χρήση ευρετηρίων (index), η δεύτερη βάση είναι όμοια της πρώτης αλλά χρησιμοποιεί ευρετήρια ενώ η τρίτη βάση χρησιμοποιεί χωρική MySQL και χωρικό τύπο δεδομένων για την αποθήκευση των συντεταγμένων. Επιπλέον χρησιμοποιούνται τα ίδια ευρετήρια όπου χρησιμοποιήθηκαν και στην δεύτερη βάση αλλά για την ευρετηρίαση των συντεταγμένων χρησιμοποιείται χωρικό ευρετήριο. Και οι τρεις βάσεις περιέχουν από τρία tables.

Το πρώτο table περιέχει το όνομα και τις συντεταγμένες του κάθε εστιατορίου. Το δεύτερο table περιέχει τις λέξεις κλειδιά. Το τρίτο table περιέχει τα ξένα κλειδιά των άλλων δυο tables και χρησιμοποιείται για την αντιστοιχία των εστιατορίων με τις λέξεις κλειδιά.

Δομή βάσεων:

4.1.1.1 Απλή MySQL χωρίς ευρετήρια

Η βάση με απλή MySQL περιέχει τα παρακάτω tables:

- Table 1: Point
Στήλες : id (primary key) / name / longitude / latitude .
Ευρετήρια: -
- Table 2: Keyword
Στήλες: idkey (primary key) / keyword .
Ευρετήρια: -
Περιέχει τις **αντιστοιχίες των ξένων κλειδίων** των λέξεων κλειδίων του table keyword με τα αντίστοιχα εστιατόρια από το table point.
- Table 3: keyjoin
Στήλες: idjoin (primary key) / id (foreign key) / idkey (foreign key) .
Ευρετήρια: -

4.1.1.2 Απλή MySQL με ευρετήρια.

Αντίστοιχα η δεύτερη βάση που εξετάσαμε περιέχει τα ίδια tables αλλά χρησιμοποιεί ευρετήρια:

Σε αυτή την βάση χρησιμοποιούνται **ευρετήρια** στις στήλες: longitude , latitude , keyword.

- Table 1: Point
Στήλες: id (primary key) / name / longitude / latitude.
Ευρετήρια : longitude / latitude. Type: BTREE
- Table 2: Keyword
Στήλες: idkey (primary key) / keyword.
Ευρετήρια: keyword. Type: BTREE
- Table 3: keyjoin
Στήλες: idjoin (primary key) / id (foreign key) / idkey (foreign key) .
Ευρετήρια: id / idkey. Type : BTREE

Παράδειγμα εντολής Select για απλή MySQL:

```
SELECT kl.id, p.onoma FROM kleidia kl INNER JOIN point AS p ON kl.id = p.id INNER JOIN keyword AS r ON r.idrest = kl.idrest WHERE r.keyword IN ('American', 'Mexican') AND latitude BETWEEN latitude - radius AND latitude + radius AND longitude BETWEEN longitude - radius AND longitude + radius GROUP BY kl.id HAVING COUNT(*) > 1;
```

4.1.2 Επεξεργασία στην Spatial MySQL

Εκτός των προσεγγίσεων της απλής MySQL στην διπλωματική εργασία εξετάσαμε και την Χωρική (Spatial) MySQL ως εναλλακτική για την επεξεργασία χωρό-κειμενικών ερωτήσεων.

Βάση 3. Χωρική(Spatial) MySQL με ευρετήρια.

- Table 1: Point

Στήλες: id (primary key) / name / coordinates (geometry type point SRID 4326).

Ευρετήρια: coordinates. Type: Spatial

- Table 2: Keyword

Στήλες: idkey (primary key) / keyword .

Ευρετήρια: keyword. Type: BTREE

- Table 3: keyjoin

Στήλες: idjoin (primary key) / id (foreign key) / idkey (foreign key) .

Ευρετήρια: id / idkey. Type : BTREE

Σε αυτή τη βάση χρησιμοποιούμε **ευρετήρια** στη στήλη : keyword. Επιπλέον χρησιμοποιούμε **χωρικά ευρετήρια** (spatial index) στη στήλη coordinates. Η κύρια διαφορά στη Spatial MySQL είναι ότι χρησιμοποιείται ο τύπος δεδομένων GEOMETRY (Point) με SRID 4326 για την αποθήκευση των χωρικών δεδομένων (πλάτος, μήκος) και στη στήλη αυτή χρησιμοποιείται χωρικό ευρετήριο.

Παράδειγμα Select για Χωρική MySQL:

```
SELECT kl.id = p.id
```

```
INNER JOIN keyword AS r ON r.idrest=kl.idrest
```

```
WHERE r.keyword in('American','Mexican') AND
```

```
ST_Distance_Sphere(p.coordinates,ST_GeomFromTextPOINT( longitude latitude), 4326)) <= radiuskm group by kl.id HAVING COUNT(*) > 1;
```

4.1.3 Αποτελέσματα

Στη συνέχεια του πειράματος με σκοπό την εύρεση της πιο αποτελεσματικής προσέγγισης εκτελέσαμε τις ίδιες ερωτήσεις σε τρεις διαφορετικές βάσεις οι οποίες χρησιμοποιούν τις

παραπάνω προσεγγίσεις. Κάθε ερώτηση εκτελέστηκε 50 φορές σε κάθε βάση με το μόνο παράγοντα που αλλάζει να είναι οι συντεταγμένες του κεντρικού σημείου. Η ακτίνα καθώς και οι λέξεις κλειδιά παρέμειναν οι ίδιες.

Αναλυτικά αποτελέσματα πρώτου πειράματος.

Με πράσινο χρώμα αναγράφονται οι καλύτεροι χρόνοι ενώ με κόκκινο οι χειρότεροι χρόνοι κάθε ερωτήματος.

Ακτίνα = - Keywords: -	Απλή MySQL χωρίς ευρετήρια	Απλή MySQL με ευρετήρια	Spatial MySQL με ευρετήρια
Minimum Rows	1.097	249	681
Maximum Rows	185.716	185.716	185.180
Minimum Time (ms)	103	20	3.174
Maximum Time (ms)	2.648	2.509	3.839
Average Rows	138.429	129.685	105.776
Average Time (ms)	624	559	3.367

Πίνακας 2. Αποτελέσματα πρώτου ερωτήματος πρώτου πειράματος

Παράμετροι : Ακτίνα -, Keywords: -

Ακτίνα = 20 Keywords: American, Mexican	Απλή MySQL χωρίς ευρετήρια	Απλή MySQL με ευρετήρια	Spatial MySQL με ευρετήρια
Minimum Rows	112	103	108
Maximum Rows	2.192	2.201	2040
Minimum Time (ms)	102	74	607
Maximum Time (ms)	1.018	1.056	1.349
Average Rows	1.377	1396	1.290
Average Time (ms)	593	532	607

Πίνακας 3. Αποτελέσματα δεύτερου ερωτήματος πρώτου πειράματος

Παράμετροι : Ακτίνα 20, Keywords: American, Mexican

Ακτίνα = 30 Keywords: American, fast, food	Απλή MySQL χωρίς ευρετήρια	Απλή MySQL με ευρετήρια	Spatial MySQL με ευρετήρια
--	-------------------------------	----------------------------	-------------------------------

Minimum Rows	206	206	204
Maximum Rows	7763	7763	7763
Minimum Time (ms)	91	67	781
Maximum Time (ms)	1.617	1.680	1.587
Average Rows	4.840	4.970	4.692
Average Time (ms)	702	684	911

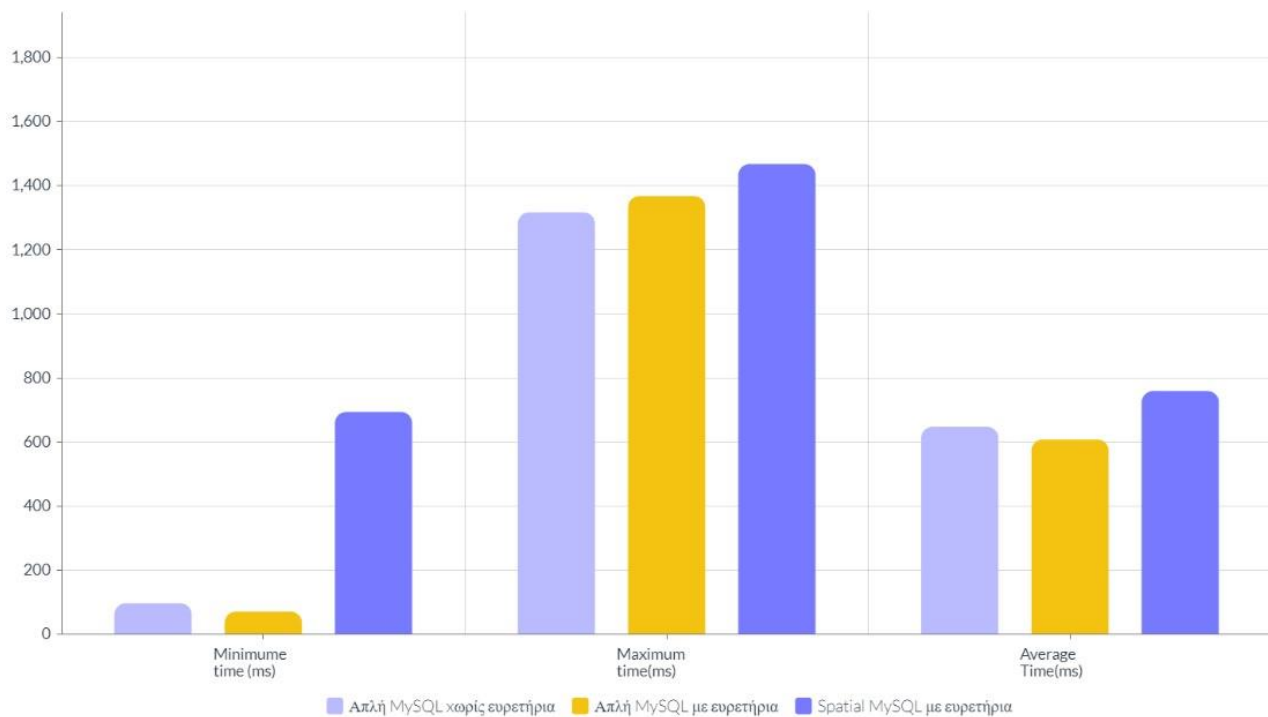
Πίνακας 4. Αποτελέσματα τρίτου ερωτήματος πρώτου πειράματος

Παράμετροι : Ακτίνα 30, Keywords: American, fast, food.

	Minimume time (ms)	Maximum time(ms)	Average Time(ms)
Απλή MySQL χωρίς ευρετήρια	96.5	1,317	647.5
Απλή MySQL με ευρετήρια	70.5	1,368	608
Spatial MySQL με ευρετήρια	694	1,468	759

Πίνακας 5. Μέσες τιμές αποτελεσμάτων (Πίνακας 2-3).

Δεν χρησιμοποιήθηκαν τα δεδομένα των ερωτήσεων όπου δεν περιείχαν λέξεις κλειδιά (Πίνακας 1).



Εικόνα 5. Διάγραμμα μέσων τιμών αποτελεσμάτων πρώτου πειράματος.

Μετά τα πρώτα αποτελέσματα παρατηρούμε πως η βάση που χρησιμοποιεί χωρική MySQL έχει τους πιο αργούς χρόνους, ενώ ο μέσος μικρότερος χρόνος είναι σημαντικά μεγαλύτερος συγκριτικά με τους υπόλοιπους. Η βάση με απλή MySQL που χρησιμοποιεί ευρετήρια έχει τη βέλτιστη απόδοση.

4.2 Πειραματική αποτίμηση: Δεύτερο Πείραμα

4.2.1 Επεξεργασία στη MySQL

Το δεύτερο αρχείο που χρησιμοποιήθηκε για την πειραματική μελέτη είναι ένα CSV αρχείο από το kaggle.com. Το συγκεκριμένο αρχείο είναι της μορφής CSV και κάθε εγγραφή καταλαμβάνει μία γραμμή. Στο αρχείο αναπαρίστανται συνολικά 224.814 εστιατόρια. Οι εγγραφές του αρχείου αναπαριστούν πραγματικά δεδομένα επιχειρήσεων που βρίσκονται στην Ινδία.

Παράδειγμα:

Mount View Restaurant,"Burger, Cafe, Chinese, Continental, Modern Indian, North Indian",771894969,322450634

Όνομα εστιατορίου: Mount View Restaurant

Κουζίνες : Burger, Cafe, Chinese, Continental, Modern Indian, North Indian

Γεωγραφικό πλάτος και μήκος : 77.1894969

Γεωγραφικό μήκος :32.2450634

Όπως και στο πρώτο πείραμα στη MySQL έχουμε δημιουργήσει τρεις διαφορετικές βάσεις δεδομένων. Η πρώτη βάση χρησιμοποιεί απλή MySQL χωρίς να κάνει χρήση ευρετηρίων (index).

Η δεύτερη βάση είναι ίδια με την πρώτη αλλά χρησιμοποιεί ευρετήρια. Η τρίτη βάση χρησιμοποιεί χωρική MySQL και χωρικό τύπο δεδομένων για την αποθήκευση των συντεταγμένων. Επιπλέον σε αυτή την βάση χρησιμοποιούνται τα ίδια ευρετήρια όπου χρησιμοποιήθηκαν και στην δεύτερη βάση και επιπλέον για την ευρετηρίαση των συντεταγμένων χρησιμοποιείται χωρικό ευρετήριο. Η κάθε μια από τις τρεις βάσεις αποτελείται από τρία tables. Το πρώτο table περιέχει το όνομα και τις συντεταγμένες του κάθε εστιατορίου. Το δεύτερο table περιέχει τις λέξεις κλειδιά και το τρίτο table περιέχει τα ξένα κλειδιά των άλλων δυο table που χρησιμοποιούνται για την αντιστοιχία των εστιατορίων με τις λέξεις κλειδιά.

- Βάση 1. Απλή MySQL.

Table 1: Point

Στήλες : id (primary key) / name / longitude / latitude .

Ευρετήρια: -

Table 2: Keyword

Στήλες: idkey (primary key) / keyword .

Ευρετήρια: -

Table 3: keyjoin

Στήλες: idjoin (primary key) / id (foreign key) / idkey (foreign key) .

Ευρετήρια: -

- Βάση 2. Απλή MySQL με ευρετήρια.

Table 1: Point

Στήλες: id (primary key) / name / longitude / latitude.

Ευρετήρια: longitude / latitude. Type: BTREE

Table 2: Keyword

Στήλες: idkey (primary key) / keyword.

Ευρετήρια: keyword. Type: BTREE

Table 3: keyjoin

Στήλες: idjoin (primary key) / id (foreign key) / idkey (foreign key) .

Ευρετήρια: id / idkey. Type : BTREE

- Βάση 3. Χωρική(Spatial) MySQL με ευρετήρια.

Table 1: Point

Στήλες: id (primary key) / name / coordinates (geometry type point SRID 4326).

Ευρετήρια: coordinates. Type: Spatial

Table 2: Keyword

Στήλες: idkey (primary key) / keyword .

Ευρετήρια: keyword. Type: BTREE

Table 3: keyjoin

Στήλες: idjoin (primary key) / id (foreign key) / idkey (foreign key) .

Ευρετήρια: id / idkey. Type : BTREE

- Παράδειγμα ερωτήματος για απλή MySQL :

```
SELECT k.id, p.name FROM keyjoin k INNER JOIN point AS p ON k.id = p.id
INNER JOIN keyword AS r ON r.idkey = k.idkey WHERE      r.keyword IN ('Street
food') AND      latitude BETWEEN (latitude - radius) AND (latitude + radius)
AND      longitude BETWEEN (longitude - radius) AND (longitude + radius)
GROUP BY k.id HAVING COUNT(*) > 0;
```

- Παράδειγμα ερωτήματος για Spatial MySQL :

```
SELECT k.id, p.name FROM keyjoin k " +
    INNER JOIN point_table AS p ON k.id = p.id " +
    INNER JOIN keyword AS r ON r.idkey = k.idkey " +
    WHERE r.keyword IN ('Chinese','North Indian','Fast Food') AND " +
    ST_Distance_Sphere(p.coordinates, ST_GeomFromText('POINT("+ longitude
    " + latitude + ")", 4326)) <= " + radiuskm + "
    GROUP BY k.id HAVING COUNT(*) > 2";
```

4.2.2 Αποτελέσματα.

Στη συνέχεια του πειράματος όπως και στο πρώτο πείραμα με σκοπό την εύρεση της πιο αποτελεσματικής προσέγγισης εκτελέσαμε τις ίδιες ερωτήσεις σε τρεις διαφορετικές βάσεις οι οποίες χρησιμοποιούν τις παραπάνω προσεγγίσεις. Κάθε ερώτηση εκτελέστηκε 50 φορές σε κάθε βάση με το μόνο παράγοντα που αλλάζει να είναι οι συντεταγμένες. Οι ακτίνα καθώς και οι λέξεις κλειδιά παρέμειναν οι ίδιες.

Αναλυτικά αποτελέσματα δεύτερου πειράματος.

Ακτίνα = 10 Keywords : -	Απλή MySQL χωρίς ευρετήρια	Απλή MySQL με ευρετήρια	Spatial MySQL με ευρετήρια
Minimum Rows	26.613	24.669	24.330
Maximum Rows	197.336	198.936	205.306
Minimum Time (ms)	261	335	11.364
Maximum Time (ms)	1.636	2.673	12.308
Average Rows	124.249	129.194	147.456
Average Time (ms)	980	1.502	11.808

Πίνακας 6. Αποτελέσματα πρώτου ερωτήματος δεύτερου πειράματος

Παράμετροι : Ακτίνα 10, Keywords: -

Ακτίνα = 10 Keywords : Street Food	Απλή MySQL χωρίς ευρετήρια	Απλή MySQL με ευρετήρια	Spatial MySQL με ευρετήρια
Minimum Rows	1.124	1.297	3.375
Maximum Rows	13.092	13.050	13.301
Minimum Time (ms)	219	32	368
Maximum Time (ms)	1.306	133	443
Average Rows	8.516	7.760	11.607
Average Time (ms)	816	57	385

Πίνακας 7. Αποτελέσματα δεύτερου ερωτήματος δεύτερου πειράματος
Παράμετροι : Ακτίνα 10, Keywords: Street Food

Ακτίνα = 10 Keywords: Chinese, North Indian	Απλή MySQL χωρίς ευρετήρια	Απλή MySQL με ευρετήρια	Spatial MySQL με ευρετήρια
Minimum Rows	5.991	5.239	5.505
Maximum Rows	38.956	38.779	39.595
Minimum Time (ms)	261	450	3.624
Maximum Time (ms)	1.727	794	4.277
Average Rows	22.727	24.816	29.310
Average Time (ms)	951	621	3.839

Πίνακας 8. Αποτελέσματα τρίτου ερωτήματος δεύτερου πειράματος
Παράμετροι : Ακτίνα 10, Keywords: Chinese, North Indian

Radius = 10 Keywords: Chinese, North Indian, Fast Food	Απλή MySQL χωρίς ευρετήρια	Απλή MySQL με ευρετήρια	Spatial MySQL με ευρετήρια
Minimum Rows	711	789	668
Maximum Rows	5.350	5.379	5.441
Minimum Time (ms)	269	620	4.981
Maximum Time (ms)	1.749	870	6.555
Average Rows	3.422	3.590	3.775
Average Time (ms)	977	738	5.372

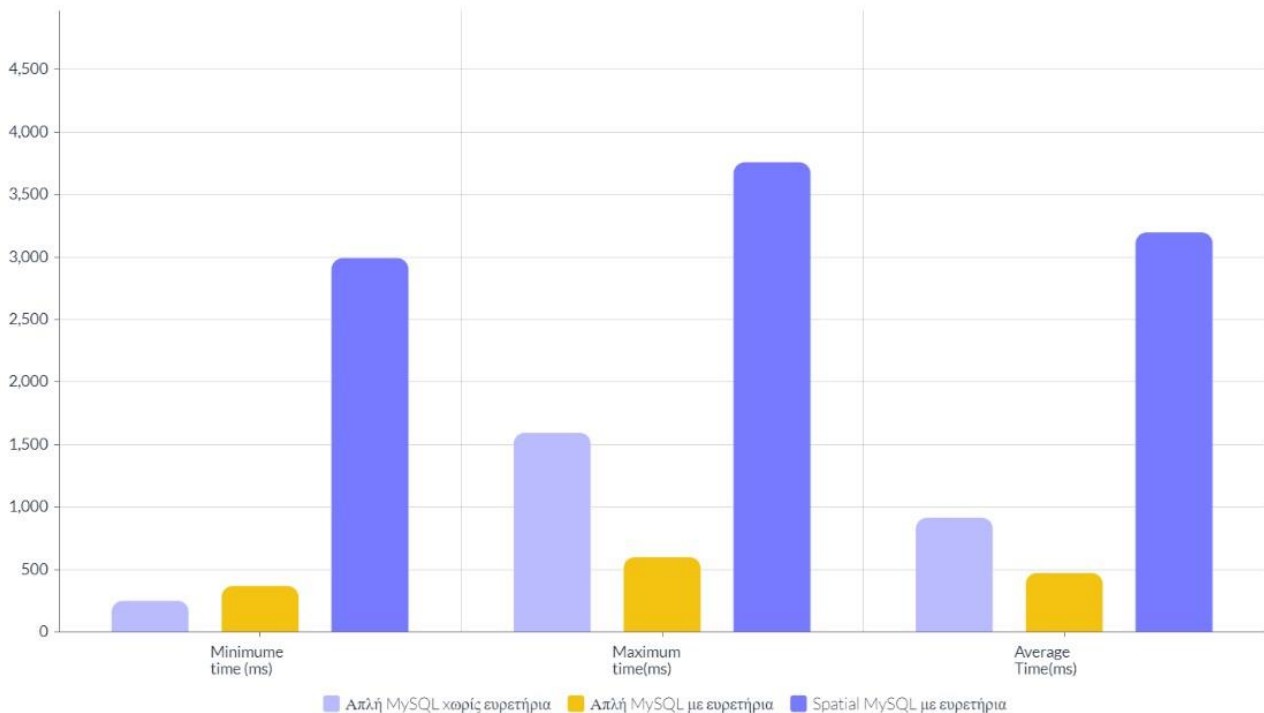
Πίνακας 9. Αποτελέσματα τέταρτου ερωτήματος δεύτερου πειράματος
Παράμετροι : Ακτίνα 10, Keywords: Chinese, North Indian, Fast Food

	Minimum time (ms)	Maximum time(ms)	Average Time(ms)
--	-------------------	------------------	------------------

Απλή MySQL χωρίς ευρετήρια	249.6	1,594	914
Απλή MySQL με ευρετήρια	367.3	599	472
Spatial MySQL με ευρετήρια	2,991	3,758	3,198

Πίνακας 10. Μέσες τιμές αποτελεσμάτων (Πίνακας 7-8-9).

Δεν χρησιμοποιήθηκαν τα δεδομένα των ερωτήσεων όπου δεν περιείχαν λέξεις κλειδιά (Πίνακας 6).



Εικόνα 6. Διάγραμμα μέσων τιμών αποτελεσμάτων δεύτερου πειράματος.

Από τα αποτελέσματα του δεύτερου πειράματος παρατηρείται όπως και στο πρώτο πείραμα πως, η απλή MySQL που χρησιμοποιεί ευρετήρια είχε τις καλύτερες χρονικές αποδόσεις ενώ η βάση που χρησιμοποιεί Spatial MySQL είχε τις χειρότερες χρονικές αποδόσεις παρά την βελτίωση στο χρόνο που προκαλεί η χρήση των χωρικών ευρετηρίων.

5

Συμπεράσματα.

Η παρούσα διπλωματική εργασία είχε ως σκοπό την μελέτη των χρόνων ανταπόκρισης της Mysql σε ερωτήσεις που αφορούν χωρο-κειμενικά δεδομένα. Για να γίνει αυτό πειραματιστήκαμε με διαφορετικές βάσεις δεδομένων και διαφορετικά ευρετήρια σε δύο ξεχωριστά αρχεία δεδομένων με τα αποτελέσματα και στα δύο πειράματα να συμβαδίζουν. Τα δεδομένα που χρησιμοποιήθηκαν ήταν στοιχεία εστιατορίων με συντεταγμένες και λέξεις κλειδιά ενώ αντίστοιχα οι ερωτήσεις που χρησιμοποιήθηκαν έψαχναν για εστιατόρια σε ακτίνα από ένα γεωγραφικό σημείο και με μια η παραπάνω λέξεις κλειδιά. Τα αποτελέσματα έδειξαν ότι η χρονικά βέλτιστη λύση είναι η απλή MySQL με χρήση BTREE ευρετηρίων σε λέξεις κλειδιά και συντεταγμένες ενώ η χειρίστη λύση είναι η Spatial MySQL όπου σε μεγάλο όγκο δεδομένων παρατηρήσαμε να είναι η πιο αργή με μεγάλη διαφορά από τις υπόλοιπες λύσεις.

Βιβλιογραφία

- [1] Ian De Felipe, Vagelis Hristidis, Naphtali Rishe. Keyword Search on Spatial Databases.
- [2] Gao Cong, Christian S. Jensen. Querying Geo-Textual Data: Spatial Keyword Queries and Beyond.
- [3] Maria Christoforaki, Jinru He, Constantinos Dimopoulos , Alexander Markowetz, Torsten Suel. Text vs. Space: Efficient Geo-Search Query Processing.
- [4] Anders Skovsgaard, Christian S. Jensen. Top-k Point of Interest Retrieval Using Standard Indexes.
- [5] Jin, J., An, N., & Sivasubramaniam, A. (n.d.). Analyzing range queries on spatial data.
- [6] Suprio Ray, Bogdan Simion, Angela Demke Brown. Jackpine: A Benchmark to Evaluate Spatial Database Performance.
- [7] Zhang, Dongxiang, et al. "Keyword search in spatial databases: Towards searching by document.
- [8] Xin Cao¹, Gao Cong¹, Christian S. Jensen², Beng Chin Ooi. Collective Spatial Keyword Querying.
- [9] A. Guttman. R-trees: A dynamic index structure for spatial searching.
- [10] Jo˜ao B. RochaJunior, Akrivi Vlachou, Christos Doulkeridis, and Kjetil Nørvag. Efficient Processing of Topk Spatial Preference Queries.
- [11] Panagiotis Tampakis, Dimitris Spyrellis, Christos Doulkeridis, Nikos Pelekis, Christos Kalyvas, Akrivi Vlachou. A Novel Indexing Method for Spatial-Keyword Range Queries.

- [12] Gao Cong, Christian S. Jensen, Dingming Wu. Efficient Retrieval of the Topk Most Relevant Spatial Web Objects.
- [13] Lisi Chen, Gao Cong, Christian S. Jensen, Dingming W. Spatial Keyword Query Processing: An Experimental Evaluation.
- [14] Lisi Chen, Shuo Shang, Chengcheng Yang, Jing Li. Spatial keyword search: a survey.