ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΥΡΕΤΗΡΙΟΠΟΙΗΣΗ ΣΕ ΥΨΗΛΕΣ ΔΙΑΣΤΑΣΕΙΣ - ΠΕΙΡΑΜΑΤΙΚΗ ΜΕΛΕΤΗ ΕΥΡΕΤΗΡΙΩΝ GIST, BRIN ΚΑΙ R*-TREE ΣΤΙΣ PostgreSQL ΚΑΙ SQLite

Διπλωματική εργασία

της

Τριανταφύλλου Αναστασία

Θεσσαλονίκη, Ιούνιος 2020

ΕΥΡΕΤΗΡΙΟΠΟΙΗΣΗ ΣΕ ΥΨΗΛΕΣ ΔΙΑΣΤΑΣΕΙΣ - ΠΕΙΡΑΜΑΤΙΚΗ ΜΕΛΕΤΗ ΕΥΡΕΤΗΡΙΩΝ GIST, BRIN ΚΑΙ R*-TREE ΣΤΙΣ PostgreSQL ΚΑΙ SQLite.

Τριανταφύλλου Αναστασία

Πτυχίο Γεωγραφίας, Τμήμα Γεωγραφίας του Πανεπιστημίου Αιγαίου

Διπλωματική Εργασία

υποβαλλόμενη για τη μερική εκπλήρωση των απαιτήσεων του

ΜΕΤΑΠΤΥΧΙΑΚΟΥ ΤΙΤΛΟΥ ΣΠΟΥΔΩΝ ΣΤΗΝ ΕΦΑΡΜΟΣΜΕΝΗ ΠΛΗΡΟΦΟΡΙΚΗ

Επιβλέπων Καθηγητής Ευαγγελίδης Γεώργιος

Ονοματεπώνυμο Καθηγητή Ευαγγελίδης Γεώργιος

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 25/06/2020

Ονοματεπώνυμο 1	Ονοματεπώνυμο 2	Ονοματεπώνυμο 3
ΕΥΑΓΓΕΛΙΔΗΣ ΓΕΩΡΓΙΟΣ	ΚΟΛΩΝΙΑΡΗ ΓΕΩΡΓΙΑ	ΣΑΜΑΡΑΣ ΝΙΚΟΛΑΟΣ

Τριανταφύλλου Αναστασία

Στον πατέρα μου, τη μητέρα και τις αδελφές μου, καθώς και στους φίλους μου όπου η υποστήριξη και αγάπη τους με βοήθησαν καθ 'όλη τη διάρκεια.

Ευχαριστίες

Αρχικά, θα ήθελα να ευχαριστήσω τον επιβλέποντα και σύμβουλό μου τον Δρ. Ευαγγελίδη Γεώργιο χωρίς την πολύτιμη βοήθεια η διατριβή δεν θα είχε επιτευχτεί. Είναι σημαντικό να τονίσουμε ότι η σωστή καθοδήγησή του και η συνεχής υποστήριξή από την αρχή με βοήθησαν να κατανοήσω σε βάθος τις βάσεις δεδομένων και τις διάφορες τεχνικές ευρετηρίασης. Έπειτα, θα ήθελα να ευχαριστήσω τους Δρ. Κολωνιάρη Γεωργία και τον Δρ. Σαμαρά Νικόλαο για το ενδιαφέρον τους για την έρευνά μου, καθώς αποτελούν την επιτροπή της διατριβής μου. Στη συνέχεια να ευχαριστήσω όλους τους καθηγητές του Πανεπιστημίου Αιγαίου στη Λέσβο του τμήματος Γεωγραφίας και πιο συγκεκριμένα τους Δρ. Βαΐτη Μιχάλη, Δρ. Σουλακέλλη Νικόλαο και Δρ. Καβρουδάκη Δημήτριο, επειδή πίστεψαν σε μένα και με ενθάρρυναν να συνεχίσω. Θα ήθελα επίσης να εκφράσω την εκτίμησή μου προς τους καθηγητές του Τμήματος της Εφαρμοσμένης Πληροφορικής της Σχολής Επιστημών Πληροφορίας του Πανεπιστημίου Μακεδονίας.

Τέλος, θα ήθελα να τονίσω την ευγνωμοσύνη στον πατέρα μου και τη μητέρα μου, που με ενέπνευσαν και με ενθάρρυναν όλες τις στιγμές για να πετύχω τους στόχους μου. Εκτείνω την αγάπη και την ευγνωμοσύνη μου στους φίλους μου που πάντα πιστεύουνε σε μένα και με κάνουν να συνειδητοποιώ πόσο δυνατή είμαι.

29 Ιουνίου 2020

Περίληψη

ΕΥΡΕΤΗΡΙΟΠΟΙΗΣΗ ΣΕ ΥΨΗΛΕΣ ΔΙΑΣΤΑΣΕΙΣ – ΠΕΙΡΑΜΑΤΙΚΗ ΜΕΛΕΤΗ ΕΥΡΕΤΗΡΙΩΝ GIST ΚΑΙ BRIN ΣΤΙΣ PostgreSQL και SQLite.

Τριανταφύλλου Αναστασία

Πανεπιστήμιο Μακεδονίας, Θεσσαλονίκη – Ιούνιος 2020

Επιβλέπων καθηγητής: Ευαγγελίδης Γεώργιος

Αντικείμενο της διατριβής, είναι η μελέτη των ευρετηρίων GiST, BRIN και R*-Tree σε δύο τύπους δεδομένων. Ο πρώτος τύπος είναι τα χωρικά δεδομένα και ο δεύτερος τα πολυδιάστατα.

Ο κύριος λόγος που χρησιμοποιούμε ευρετήρια είναι η μείωση του χρόνου που χρειάζεται για να απαντηθεί ένα SQL ερώτημα. Κάθε σύστημα διαχείρισης βάσεων δεδομένων (DBMS) υποστηρίζει διαφορετικά ευρετήρια, όπου η επιλογή του καθενός εξαρτάται από τα δεδομένα που χρησιμοποιούνται και από την χρήση του κάθε ερευνητή. Στη συγκεκριμένη έρευνα θα χρησιμοποιήσουμε δύο ευρέως γνωστές βάσεις δεδομένων, την PostgreSQL και την SQLite.

Το Generalized Search Tree, ή αλλιώς GiST αποτελεί ένα προηγμένο σύστημα το οποίο συνδυάζει ένα μεγάλο εύρος από διαφορετικούς αλγόριθμους ευρετηριοποίησης όπως B-Tree, B+-Tree, R-Tree, partial sum trees, ranked B+-Trees και αρκετούς ακόμα. Το BRIN είναι μια νέα τεχνική ευρετηρίου βάσης δεδομένων, το οποίο έχει σχεδιαστεί για μεγάλους πίνακες δεδομένων. Το GiST και το BRIN αποτελούν ευρετήρια της PostgreSQL.

Το R*-Tree είναι μια παραλλαγή των R-Trees που χρησιμοποιούνται για την ευρετηρίαση χωρικών και πολυδιάστατων δεδομένων. Το R*-Tree είναι ενσωματωμένο στο λογισμικό ανοιχτού κώδικα SQLite.

Έχουν προταθεί διάφορες τεχνικές για τη βελτίωση της απόδοσης των ερωτημάτων σε μία βάση δεδομένων. Στη διατριβή μας θα συγκρίνουμε τις αποδόσεις ξεχωριστά για κάθε βάση δεδομένων με και χωρίς ευρετήρια.

Είναι επίσης σημαντικό να δούμε σε τι τύπους ερωτημάτων χρησιμοποιούνται τα ευρετήρια. Για αυτό το λόγο θα χωρίσουμε τα χωρικά ερωτήματα σε πέντε κατηγορίες: απλή αναζήτηση SQL (Simple SQL), γεωμετρία(Geometry), χωρική σχέση (Spatial Relationship), χωρική σύνδεση (Spatial Join) και πλησιέστερο γείτονα (Nearest Neighbor). Εκτελέσαμε εκτενή πειράματα σε όλες αυτές τις πέντε κατηγορίες και καταγράψαμε τον χρόνο εκτέλεσης. Φυσικά ο ίδιος τύπος ερωτημάτων δεν ισχύει για τα πολυδιάστατα δεδομένα. Σε αυτή την περίπτωση θα έχουμε ερωτήματα με διαστήματα τιμών. Τα αποτελέσματα της έρευνας θα δώσουν στον αναγνώστη να κατανοήσει ποια είναι η καταλληλότερη δομή ευρετηρίου ανάλογα με τους τύπους των δεδομένων.

Τα χωρικά δεδομένα αποτελούν ένα σύνολο δεδομένων αναφοράς της πόλης του Λονδίνου στη Μεγάλη Βρετανία και τα πολυδιάστατα δεδομένα είναι το αρχείο Letter Recognition.

Λέξεις Κλειδιά: DBMS, SQLite, PostgreSQL, GiST, BRIN

Abstract

HIGH LEVEL INDEXING – EXPERIMENTAL STUDY OF GIST, BRIN AND R*-TREE INDEXES IN PostgreSQL AND SQLite.

Triantafillou Anastasia

University of Macedonia, Thessaloniki - June 2020

Supervising Professor: Evangelides Georgios

The purpose of this thesis is to study the GiST, BRIN and R*-Tree indexes in two types of data. The first type is spatial data and the second one is multidimensional data.

The main reason we use indexes is to reduce the time needed to answer an SQL query. Each database management system (DBMS) supports different indexes, wherein the selection of each is dependent on the data used and the use of each researcher. In this thesis, we will use two well known databases, PostgreSQL and SQLite.

The Generalized Search Tree, or otherwise GiST is an advanced system that combines a wide range of different search algorithms such as B-Tree, B+-Tree, R-Tree, partial sum trees, ranked B+-Trees and many others. BRIN is a new database index technique designed for larger amounts of data. The GiST and BRIN are indexes of PostgreSQL.

R*-Tree is also used for the same purpose, ie indexing spatial and multidimensional data. R*-Tree is embedded in SQLite.

Various techniques have been proposed to improve the performance of queries. In our thesis we will compare the performance separately for each database with and without indexes.

It is also important to see what queries types are used in the indexes. For this reason, we will divide the spatial questions into five categories: simple SQL (Simple SQL) search, Geometry,

Spatial Relationship, Spatial Join, and Nearest Neighbor. We conducted extensive experiments in all five of these categories and recorded the execution time. Of course, the same type of questions do not apply to multidimensional data. In this case we will have questions with values intervals. The results of the research will allow the reader to understand which is the most appropriate index structure according to the queries types.

The spatial data we have used is a set of reference data of the city of London in Great Britain and the multidimensional data is the Letter Recognition file.

Keywords: DBMS, SQLite, PostgreSQL, GiST, BRIN

Περιεχόμενα

Κατάλογος Πινάκων

Κατάλογος Εικόνων

ΚΕΦΑΛΑΙΟ 1- ΕΙΣΑΓΩΓΗ

1.1 Εισαγωγικά στοιχεία της έρευνας

Είναι γεγονός ότι στη σημερινή εποχή κατακλυζόμαστε από σύνολο δεδομένων, τα οποία κάθε δευτερόλεπτο αυξάνονται κάνοντας την χρήση τους και τη διαχείριση τους δύσκολη. Τα δεδομένα αυτά μπορεί να προέρχονται από εσωτερικές πηγές δεδομένων π.χ. μιας επιχείρησης, αλλά και από εξωτερικές πηγές, όπως το διαδίκτυο.

Πολλές σύγχρονες εφαρμογές βάσεων δεδομένων διαχειρίζονται μεγάλες ποσότητες πολυδιάστατων δεδομένων. Οι εφαρμογές αυτές περιλαμβάνουν:

1. *Multimedia Content -based Retrieval*
2. *Spatial/Spatio-temporal Databases*
3. *Time Series/Scientific/Medical Databases*
4. *Data Mining/OLAP*

Η παρούσα διατριβή εστιάζει στις δύο κατηγορίες από τις τέσσερεις, στην χωρικές βάσεις δεδομένων και στην εξόρυξη χρήσιμων πληροφοριών από μεγάλα σύνολα δεδομένων.

Είναι γεγονός τα τελευταία 15 χρόνια, με διαφόρους τρόπους μέσα στην καθημερινότητά μας χρησιμοποιούμε χωρικά δεδομένα. Αυτό οφείλεται στη συνεχή ανάπτυξη της τεχνολογίας σε συνδυασμό με τα μέσα που χρησιμοποιούμε, όπως τα Smartphones κ.α. Πολλές εφαρμογές σήμερα(apps) κάνουν χρήση των συστημάτων GNSS. Τα αρχικά προέρχονται από το (satellite navigation or satnav system), με το οποίο ο κάθε χρήστης έχει γρήγορη πρόσβαση σε χωρικά δεδομένα όπως πόλεις, δρόμους ή φαρμακεία, βενζινάδικα και άλλα σημεία ενδιαφέροντος.

Οι χωρικές βάσεις δεδομένων αναπαριστούν τις θέσεις των αντικειμένων με (x;y) (2-Διαστάσεις) ή (x; y; z) (3- Διαστάσεις) και τις αποθηκεύουν μαζί με άλλα χαρακτηριστικά των αντικειμένων[15].

Τέτοιου είδους βάσεων δεδομένων λειτουργούν ως επέκταση σε υπηρεσίες βασισμένες σε γεωγραφικές τοποθεσίες, όπως τα γεωγραφικά συστήματα πληροφοριών(GIS),

περιβαλλοντική μοντελοποίηση και αξιολόγηση επιπτώσεων, διαχείριση πόρων και τέλος υποστήριξη αποφάσεων.

Σε μια βάση δεδομένων, κάθε εγγραφή δεδομένων περιέχει τιμές για πολλά χαρακτηριστικά που ορίζουν μαζί έναν πολυδιάστατο χώρο. Για παράδειγμα, στη βάση δεδομένων Απογραφής Πληθυσμού, κάθε αρχείο προσώπων περιέχει πληροφορίες για την ηλικία, το εισόδημα, το εκπαιδευτικό επίτευγμα, την εργασία κ.α.

Όσον αφορά στον όρο "πολυδιάστατο" τείνει να εφαρμόζεται μόνο σε σύνολα δεδομένων με τρεις ή περισσότερες διαστάσεις. Εννοιολογικά, μια πολυδιάστατη βάση δεδομένων χρησιμοποιεί την ιδέα ενός ‘υπερκύβου’(cube) δεδομένων για την αναπαράσταση των διαστάσεων των διαθέσιμων δεδομένων σε ένα χρήστη.

Τα Συστήματα Διαχείρισης Βάσεων Δεδομένων πολλών διαστάσεων χρησιμοποιούν δομές ευρετηρίων για την γρήγορη και αποτελεσματική ανάκτηση δεδομένων. Χωρίς τα ευρετήρια, οποιαδήποτε αναζήτηση δεδομένων θα απαιτούσε μια "διαδοχική σάρωση" κάθε εγγραφής στη βάση δεδομένων, με αποτέλεσμα πολύ μεγαλύτερο χρόνο επεξεργασίας. Στη παρούσα έρευνα θα παρουσιάσουμε τρεις δομές ευρετηρίων, οι οποίες συγκρίνονται βάση της απόδοσής τους σε διαφορετικές συνθήκες.

1.2 Σκοπός – Στόχοι

Θα μελετήσουμε τα ευρετήρια GiST, BRIN και R*-Tree σε δύο διαφορετικά συστήματα διαχείρισης δεδομένων, PostgreSQL και SQLite, που τους επιτρέπουν να πραγματοποιούν ερωτήματα πάνω σε γεωγραφικά δεδομένα, τα οποία εκτείνονται με PostGIS και SpatiaLite αντίστοιχα, αλλά και σε πολύ-μεταβλητά αριθμητικά δεδομένα. Χρησιμοποιούμε γεωγραφικά δεδομένα με μορφή shapefile. Τα shapefiles είναι στοιχεία αναφοράς του Λονδίνου της Μεγάλη Βρετανία, εγγραφές που αποτελούνται από σημεία: stations, γραμμές: roads, railways και πολύγωνα: borough, buildings, wards. Το καθένα αντιπροσωπεύεται από τη γεωμετρία x-y, αλλά και πολλά μη χωρικά δεδομένα όπως ο πληθυσμός.

Τα πολύ-μεταβλητά δεδομένα είναι οι εγγραφές Αναγνώρισης Γραμμάτων και αποτελείται από 17 στήλες (κατηγορία γραμμάτων και 16 αριθμητικά χαρακτηριστικά).

Ο κύριος στόχος αυτής της εργασίας είναι να συγκρίνει την απόδοση των ευρετηρίων βάση διαφορετικών κατηγοριών ερωτημάτων στους δύο τύπους δεδομένων.

1.3 Δομή της Εργασίας

Η διατριβή οργανώνεται ως εξής: στο Κεφάλαιο 2 αναλύονται τα ευρετήρια που θα χρησιμοποιήσουμε, το GiST, BRIN και R*-Tree. Το Κεφάλαιο 3 παρουσιάζει κάποια γενικά στοιχεία για τα Γεωγραφικά Πληροφοριακά Συστήματα(GIS) και τα χωρικά δεδομένα. Στο Κεφάλαιο 4 αναφέρονται σημαντικά στοιχεία για τα πολυδιάστατα αριθμητικά δεδομένα. Στο Κεφάλαιο 5 παρέχεται μια επισκόπηση των DBMS όπως η PostgreSQL και η SQLite. Το Κεφάλαιο 6 εξηγεί τα ερωτήματα σε χωρικά και πολυδιάστατα δεδομένα. Το κεφάλαιο 7 παρουσιάζει τα πειραματικά αποτελέσματα και τέλος το Κεφάλαιο 8 ολοκληρώνει την έρευνα μας.

ΚΕΦΑΛΑΙΟ 2 – ΜΕΘΟΔΟΙ ΕΥΡΕΤΗΡΙΑΣΗΣ

2.1 Ευρετήρια σε Πολλές Διαστάσεις

Η χρήση ευρετηρίων σε χωρικά και πολυδιάστατα δεδομένα είναι μια τεχνική για τη βελτιστοποίηση της επεξεργασίας ερωτημάτων σε βάσεις δεδομένων. Η απόδοση των ερωτημάτων αυξάνεται ιδιαίτερα όταν έχουμε πολλές εγγραφές σε έναν πίνακα. Ο αριθμός των εγγραφών θα μειωθεί μόλις δημιουργηθούν τα κατάλληλα ευρετήρια. Στο παραδοσιακό σύστημα βάσης δεδομένων, τα δεδομένα ταξινομούνται για αποτελεσματική αναζήτηση χρησιμοποιώντας την προσέγγιση ταξινόμησης, όπως το B-Tree. Αλλά αυτή η προσέγγιση περιορίζεται σε δεδομένα μιας διαστάσεως όπως αριθμούς, συμβολοσειρές κλπ.

Η δημιουργία ενός ευρετηρίου περιλαμβάνει τη δήλωση CREATE INDEX, η οποία μας επιτρέπει να ονομάσουμε το ευρετήριο, να καθορίσουμε τον πίνακα και ποια στήλη ή στήλες να χρησιμοποιήσουμε τα ευρετήρια και να υποδείξουμε αν ο δείκτης είναι σε αύξουσα ή φθίνουσα σειρά[20].

2.1.1 Τύποι Ευρετηρίων

Η PostgreSQL παρέχει μια ποικιλία ευρετηρίων και επίσης αρκετούς τρόπους δημιουργίας αυτών των ευρετηρίων. Τα ευρετήρια αυτά είναι τα: B-tree, Hash, GiST, SP-GiST, GIN και BRIN. Το καθένα χρησιμοποιεί έναν διαφορετικό αλγόριθμο που ταιριάζει καλύτερα σε διαφορετικούς τύπους ερωτημάτων.

B-Tree είναι ο προεπιλεγμένος τύπος ευρετηρίου στη PostgreSQL που δημιουργείται με την εντολή ‘CREATE INDEX’ χωρίς να αναφέρετε το όνομα ευρετηρίου. Αυτό το ευρετήριο είναι κατάλληλο για δεδομένα που μπορούν να ταξινομηθούν και μπορούν να χειριστούν την ισότητα και ερωτήματα εύρους.

Η ακόλουθη εντολή χρησιμοποιείται για τη δημιουργία ευρετηρίου **B-Tree**:

CREATE INDEX name ON table (column); or

CREATE INDEX name ON table USING BTREE (column);

Το **hash index** μερικές φορές αποδίδει καλύτερα από το ευρετήριο B-Tree. Το ευρετήριο κατακερματισμού λειτουργεί μόνο με τελεστές ισότητας που σημαίνει ότι μπορούμε να αναζητήσουμε μόνο δεδομένα που ταιριάζουν ακριβώς. Αυτό σημαίνει ότι ευρετήριο είναι πιο εξειδικευμένο σε συγκρίσεις που απαιτείτε η ισότητα.

Η ακόλουθη εντολή χρησιμοποιείται για τη δημιουργία ευρετηρίου κατακερματισμού:

CREATE INDEX name ON table USING HASH (column);

Το **GiST** ή Generalized Search Tree είναι χρήσιμο όταν τα δεδομένα που πρόκειται να ευρετηριαστούν είναι πιο περίπλοκα από το να κάνουμε μια απλή σύγκριση εξίσωσης ή εύρους όπως η εύρεση πλησιέστερου γείτονα. Το παράδειγμα τέτοιων δεδομένων περιλαμβάνει γεωμετρικά δεδομένα. Τα ευρετήρια GiST μπορούν να χρησιμοποιηθούν ως R-Tree και είναι ακόμη δυνατό να λειτουργήσουν και ως B-Tree.

CREATE INDEX name ON table USING gist (column);

Τα ευρετήρια **SP-GiST** ή Space Partitioned Gist είναι χρήσιμα όταν τα δεδομένα μπορούν να ομαδοποιηθούν σε μη επικαλυπτόμενες ομάδες. Το SP-GiST, όπως και το GiST επιτρέπει την υλοποίηση ενός ευρέως φάσματος διαφορετικών μη ισοζυγισμένων δομών δεδομένων στο δίσκο, όπως: quadtrees, k-d trees, and radix trees[29].

Τα ευρετήρια **GIN** ή Generalized Inverted indexes είναι χρήσιμα για την ευρετηρίαση δεδομένων που αποτελούνται από πολλά στοιχεία σε μία μόνο στήλη, όπως πίνακες, έγγραφα json (jsonb) ή έγγραφα αναζήτησης κειμένου (tsvector).

CREATE INDEX name ON table USING gin (column);

Τα ευρετήρια **BRIN** ή Block Range Indexes είναι χρήσιμα για πίνακες μεγάλου μεγέθους που έχουν στήλες με κάποια φυσική σειρά ταξινόμησης. Το ευρετήριο BRIN χωρίζει τον πίνακα σε κομμάτια εύρους και διατηρεί μια σύνοψη πληροφοριών αυτών των κομματιών. Αυτή η σύνοψη περιλαμβάνει τις ελάχιστες και μέγιστες τιμές του εύρους.

CREATE INDEX name ON table USING brin (column);

Η παραπάνω λίστα περιγράφει τους διαθέσιμους αλγόριθμους ευρετηρίων που υπάρχουν στη βάση δεδομένων PostgreSQL, τώρα ας δούμε τα ευρετήρια σε SQLite.

Το **R-Tree** είναι ένα ευρετήριο ειδικά σχεδιασμένο για ερωτήματα εύρους. Τα R-Trees χρησιμοποιούνται συχνότερα σε χωρικά συστήματα όπου κάθε είσοδος είναι ορθογώνιο με ελάχιστες και μέγιστες συντεταγμένες X και Y. Τα R-Trees βρίσκουν επίσης χρήση σε ερωτήματα που αφορούν χρονικά εύροι.

Η ιδέα του R-Tree ξεκίνησε με τον Toni Guttman: R-Trees: A Dynamic Index Structure for Spatial Searching, Proc. 1984 Διεθνές συνέδριο ACM SIGMOD για τη διαχείριση δεδομένων, Η υλοποίηση που βρέθηκε στο SQLite είναι μια βελτίωση της αρχικής ιδέας του Guttman, που ονομάζεται **R*-Trees**, που περιγράφεται από τους Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider, Bernhard Seeger: The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles. SIGMOD Conference 1990: 322-331. [24][27][28]

Τα ευρετήρια που θα αναλύσουμε και θα χρησιμοποιήσουμε για τη διεκπεραίωση της διατριβής μας θα είναι τρία:

- Το GiST μια δομή δεδομένων και ένα API που χρησιμοποιείται για την κατασκευή ποικιλιών δομών ευρετηρίασης.
- Το BRIN είναι σχεδιασμένο για χειρισμό πολύ μεγάλων πινάκων.
- Το R*-Tree είναι μια παραλλαγή του R-Tree, και χρησιμοποιείται για την επεξεργασία δεδομένων στην SQLite.

2.2 GiST (Generalized Search Tree)

Οι Hellerstein et al. [HNP95] εισήγαγαν μια δομή ευρετηρίου, που ονομάζεται Generalized Search Tree (GiST), η οποία είναι μια γενικευμένη μορφή ενός R-tree [Gut84]. Τα (Generalized Search Trees) είναι πιο αποτελεσματικά σε σύνθετους (λ.χ. με πολλές διαστάσεις) ή νέους τύπους δεδομένων (π.χ. σύνολα τιμών).

Το δέντρο γενικευμένης αναζήτησης (GiST), είναι μια δομή ευρετηρίου που υποστηρίζει ένα επεκτάσιμο σύνολο ερωτημάτων και τύπων δεδομένων. Σε μια δομή δεδομένων, το ευρετήριο GiST παρέχει όλη τη λογική των δέντρων αναζήτησης που απαιτείται από ένα σύστημα βάσης δεδομένων, με αυτό τον τρόπο ενοποιεί διαφορετικές δομές όπως τα B+-Trees και R-Trees σε ένα κομμάτι κώδικα[21].

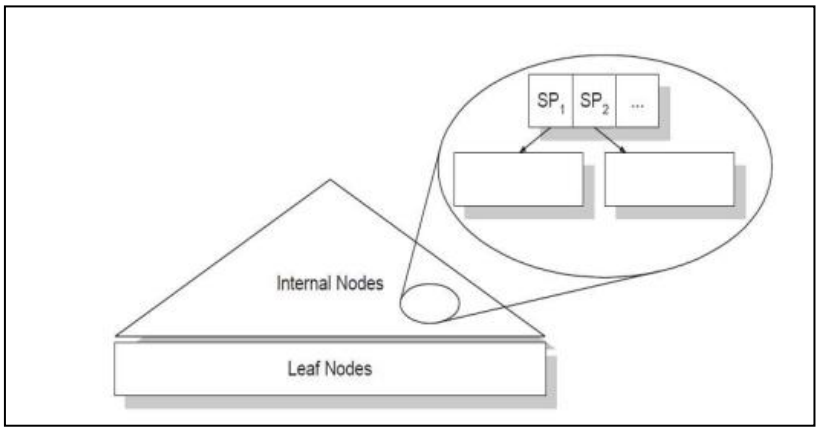
Σε αυτό το κεφάλαιο θα περιγράψουμε την δομή, την λειτουργίες, τις μεθόδους επέκτασης και τους ενσωματωμένους αλγόριθμους.

2.2.1 Δομή GiST

Ως προς τη δομή, το GiST είναι ένα ισορροπημένο δέντρο σε μεταβλητό εύρος μεταξύ kM και M, 2/M <= k <= ½. Εξαίρεση αποτελεί η ρίζα το δέντρου, η οποία μπορεί να κυμαίνεται στο εύρος τιμών 2 και M.

Η σταθερά k ονομάζεται ο ελάχιστος συντελεστής πληρότητας του δέντρου[26]. Οι κόμβοι φύλλων του δέντρου περιέχουν ζεύγη E = (p, ptr), όπου το p είναι ένα κατηγόρημα που χρησιμοποιείται ως κλειδί αναζήτησης και το ptr είναι το αναγνωριστικό κάποιας πλειάδας(γραμμή στο πίνακα) στη βάση δεδομένων.

Οι κόμβοι που δεν περιέχουν φύλλα περιέχουν ζεύγη E = (p, ptr), όπου το p είναι ένα κατηγόρημα που χρησιμοποιείται ως κλειδί αναζήτησης και το ptr είναι δείκτης σε έναν άλλο κόμβο δέντρου(δείτε Εικόνα 1).



Εικόνα 1: Δομή GiST

2.2.2 Ιδιότητες GiST

Ένα γενικευμένο δένδρο αναζήτησης χαρακτηρίζεται από τις παρακάτω ιδιότητες:

1. Ο αριθμός των στοιχείων κάθε φύλλου ή εσωτερικού κόμβου κυμαίνεται μεταξύ m και M, εκτός αν είναι η ρίζα.

2. Για κάθε καταχώριση ευρετηρίου (p, ptr) σε έναν κόμβο φύλλων, το p είναι αληθές όταν τεκμηριώνεται με τις τιμές από την υποδεικνυόμενη πλειάδα (δηλαδή το p περιέχει την πλειάδα.)

3. Για κάθε καταχώριση ευρετηρίου (p, ptr) σε έναν κόμβο χωρίς φύλλα, το p είναι αληθές όταν τεκμηριώνεται με τις τιμές οποιασδήποτε πλειάδας προσβάσιμη από το ptr.

4. Η ρίζα έχει τουλάχιστον δύο παιδιά, εκτός αν είναι φύλλο.

5. Όλα τα φύλλα εμφανίζονται στο ίδιο επίπεδο

2.2.3 Μέθοδοι χειρισμού κλειδιών στο GiST

Τα κλειδιά καθορίζονται από το χρήστη και μπορεί να περιλαμβάνουν διαφόρων μορφών κατηγορήματα, ανάλογα με το είδος των δεδομένων που χρειάζεται να δεικτοδοτηθούν. Σε περίπτωση που χρησιμοποιούνται ως R-δένδρα, τα κατηγορήματα εκφράζουν ελάχιστα bounding box (Εικόνα 1). Παρέχονται εφτά μμέθοδοι που υλοποιούν την λειτουργία των κλειδιών του δένδρου

- (Consistent (entry, predicate)), δίνεται μια καταχώριση E = (p, ptr) και ένα κατηγόρημα του στοιχείου entry q, επιστρέφει ψευδές εάν το p ^ q τότε η αναζήτηση δεν συνεχίζεται στους απογόνους του entry (αφού είναι βέβαιο ότι κανένα φύλλο του δεν θα ικανοποιεί το predicate)

- Union (P): δεδομένο ένα σύνολο P εγγραφών (p1; ptr1); ::: (pn; ptrn), επιστρέφει κάποιο κατηγόρημα r που ισχύει για όλες τις πλειάδες που είναι αποθηκευμένες κάτω από το ptr1 έως το ptrn. Αυτό μπορεί να γίνει με την εύρεση ενός r έτσι ώστε (p1 _ ::: _ pn)! r.

- (Compress (entry E)): δεδομένη μια καταχώριση E = (p, ptr) επιστρέφει μια καταχώριση (π, ptr) όπου π είναι μια συμπιεσμένη αναπαράσταση του p.

- (Decompress (entry E)). δίνεται μια συμπιεσμένη παράσταση E = (π; ptr), όπου π = Compress (p), επιστρέφει μια καταχώριση (r, ptr) έτσι ώστε p! r. Είναι το ανάποδο της προηγούμενης μμεθόδου, μόνο που η συμπίεση πιθανόν να συνεπάγεται απώλειες (lossy), καθώς δεν απαιτούμε το p ◄► r.

- (Penalty (entry E1, entry E2)), δίνονται δύο καταχωρήσεις E1 = (p1, ptr1); E2 = (p2, ptr2), που πρακτικά μεταφράζεται στο κόστος εισαγωγής του στοιχείου E1 στο υποδένδρο με ρίζα το E2. Διαισθητικά, η λειτουργία αυτή δημιουργεί τις «συστάδες» των περιεχομένων του δείκτη, κατευθύνοντας τα στοιχεία προς το κατάλληλο υποδένδρο.

- (PickSplit (P)) δεδομένου ενός συνόλου P καταχωρήσεων M + 1 (p, ptr), χωρίζει το P σε δύο ομάδες καταχωρίσεων P1. P2, κάθε μέγεθος τουλάχιστον kM. Αν κάποιος κόμβος υπερβεί τη μέγιστη πληρότητά του (το πολύ M στοιχεία), τα M+1 στοιχεία E[ ] διαμοιράζονται μεταξύ δύο κόμβων, με τουλάχιστον kM στοιχεία στον καθένα.

- (Distance). Υπολογίζει την απόσταση (έναν αριθμό) μεταξύ ενός κλειδιού και της τιμής ερωτήματος. Η συνάρτηση απόστασης είναι προαιρετική και απαιτείται σε περίπτωση που υποστηρίζεται η αναζήτηση KNN[9]

2.2.4 Λειτουργίες στο GiST

Θα περιγράψουμε τις διάφορες λειτουργίες της μεθόδου GiST, όπως αναζήτησης, εισαγωγής και διαγραφής:

1. Αναζήτηση (Search)

Μπορεί να χρησιμοποιηθεί για την αναζήτηση οποιουδήποτε συνόλου δεδομένων με οποιοδήποτε πρόθεμα ερωτήματος, διασχίζοντας το μεγαλύτερο μέρος του δέντρου όπως είναι απαραίτητο για την ικανοποίηση του ερωτήματος.

Ο αλγόριθμος αναζήτησης, μοιάζει πολύ με τον αντίστοιχο των R-δένδρων, με την εξής γενική μορφή:

- Αφετηρία της αναζήτησης είναι πάντοτε η ρίζα.
- Διασχίζονται αναδρομικά όλα τα μονοπάτια των οποίων τα κατηγορήματα εμφανίζουν συνάφεια με το ζητούμενο αντικείμενο.
- Στα φύλλα γίνεται ο τελικός έλεγχος της συνάφειας με τα στοιχεία που υπάρχουν αποθηκευμένα στις αντίστοιχες πλειάδες[21][9][19].

Algorithm Search(*R*, *q*)

Input: GiST rooted at *R*, predicate *q*

Output: all tuples that satisfy *q*

Sketch: Recursively descend all paths in tree whose keys are consistent with *q*.

S1: [Search subtrees] If *R* is not a leaf, check each entry *E* on *R* to determine whether Consistent(*E, q*). For all entries that are Consistent, invoke Search on the subtree whose root node is referenced by *E* .ptr.

S2: [Search leaf node] If *R* is a leaf, check each entry *E* on *R* to determine whether Consistent(*E, q*). If *E* is Consistent, it is a qualifying entry. At this point *E* .ptr could be fetched to check *q* accurately, or this check could be left to the calling process[17].

2. Εισαγωγή (Insert)

Βασικό μέλημα κατά την εισαγωγή στοιχείων είναι τα γενικευμένα δένδρα να διατηρούνται ισοζυγισμένα, οπότε ακολουθείται αλγόριθμος παρόμοιος με τα R-δένδρα. Η βασική διαφορά της τεχνικής είναι ότι επιτρέπει προσδιορισμό του επιπέδου όπου θα γίνει η εισαγωγή[21][9][19].

- Ο αλγόριθμος ξεκινά από κάποιο επίπεδο (όχι απαραίτητα από τη ρίζα), εντοπίζοντας τον κόμβο που επαληθεύει το ζητούμενο κατηγόρημα.
- Σε κάθε επίπεδο του δένδρου, μπορεί να χρειαστεί να επιλεγεί το καταλληλότερο υποδένδρο: πρόκειται γι’ αυτό με το μικρότερο κόστος (Penalty). Επομένως, διασχίζεται ένα μονοπάτι από τη ρίζα προς τα φύλλα.
- Εντοπίζεται το φύλλο όπου πρέπει να γίνει η εισαγωγή και το στοιχείο τοποθετείται εκεί.
- Αν χρειαστεί, το φύλλο διασπάται και το δένδρο αναδιοργανώνεται, διαδίδοντας τις αλλαγές προς τους προγόνους.
- Τα κλειδιά των προγόνων προσαρμόζονται κατάλληλα, ώστε να αντιπροσωπεύουν επακριβώς τα σύνολα τιμών που περιέχονται στο υποδένδρο τους.

Algorithm Insert(*R, E, l*)

Input: GiST rooted at *R*, entry *E* = (*p, ptr*), and level *l*, where *p* is a predicate such that *p* holds for all tuples reachable from *ptr*.

Output: new GiST resulting from insert of *E* at level *l*.

Sketch: find where *E* should go, and add it there, splitting if necessary to make room.

I1. [invoke ChooseSubtree to find where *E* should go] Let *L* = ChooseSubtree(*R, E, l*)

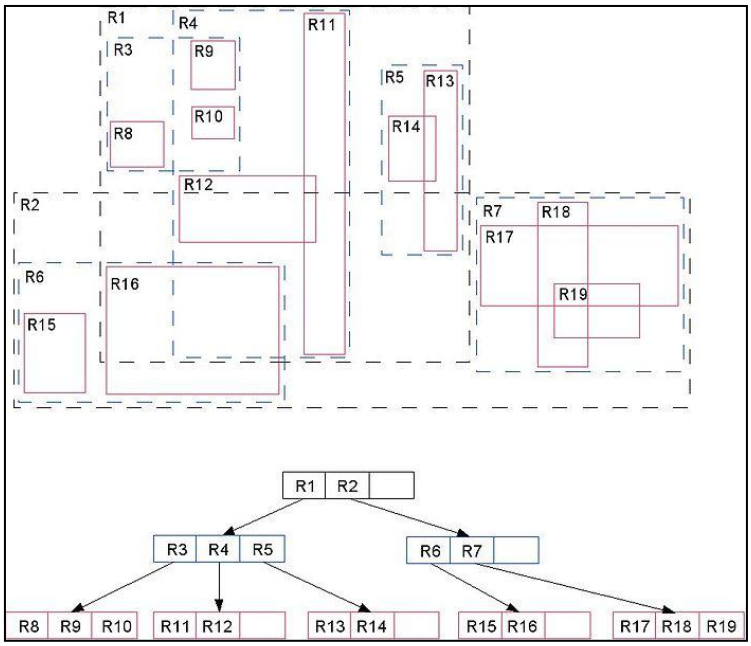
I2. If there is room for *E* on *L*, install *E* on *L* (in order according to Compare, if IsOrdered.) Otherwise invoke Split(*R, L, E*).

I3. [propagate changes upward] AdjustKeys(*R, L*) [17].

3. Διαγραφή (Delete)

Για να εντοπιστεί το στοιχείο που θα διαγραφεί, προηγείται η διαδικασία αναζήτησης. Επίσης ο αλγόριθμος Διαγραφής διατηρεί την ισορροπία του δέντρου. Αφού το εντοπίσει το διαγράφει και στη συνέχεια προσαρμόζει το δένδρο αν το διαγραμμένο στοιχείο έχει προκαλέσει αλλαγές στη δομή του δένδρου[21][9][19].

Παράδειγμα GiST:



Εικόνα 2: Παράδειγμα δέντρου GiST[25]

Όπως θα παρατηρήσουμε(δείτε Εικόνα 2) ότι τα R1 και R2 αποτελούν την κορυφή του δέντρου. Τα R1 και R2 είναι τα bounding boxes που περιέχουν οτιδήποτε άλλο. Τα R3, R4 και R5 περιέχονται στο R1. Τα R8, R9 και R10 περιέχονται στο R3 και ούτω καθ 'εξής. Κατά συνέπεια, ένας δείκτης GiST είναι ιεραρχικά οργανωμένος. Αυτό που μπορούμε να παρατηρήσουμε στο διάγραμμα είναι ότι υποστηρίζονται ορισμένες λειτουργίες που δεν είναι διαθέσιμες στα b-tree. Ορισμένες από αυτές τις λειτουργίες είναι επικαλύψεις, αριστερά, δεξιά. Η διάταξη ενός δέντρου GiST είναι ιδανική για γεωμετρική ευρετηρίαση.

Δηλαδή, επιτρέπει την αναζήτηση για αυθαίρετες περιοχές, σημεία και την εκτίμηση του αριθμού των κουκίδων σε μια περιοχή χωρίς πλήρη σάρωση δεδομένων[9].

2.3 BRIN (Block Range Index)

Η PostgreSQL 9.5 εισήγαγε μια δυνατότητα που ονομάζεται ευρετήρια εύρους μπλοκ, "Block Range Index" (γνωστό και ως BRIN) που προορίζονται να βελτιώσουν την απόδοση ερωτημάτων σε εξαιρετικά μεγάλους πίνακες. Το BRIN Index θεωρείται μια επαναστατική ιδέα στην ευρετηρίαση που προτάθηκε για πρώτη φορά από τον Alvaro Herrera και έχει το

πλεονέκτημα να καταλαμβάνει σημαντικά λιγότερο χώρο στο δίσκο από ένα τυπικό ευρετήριο B-Tree.

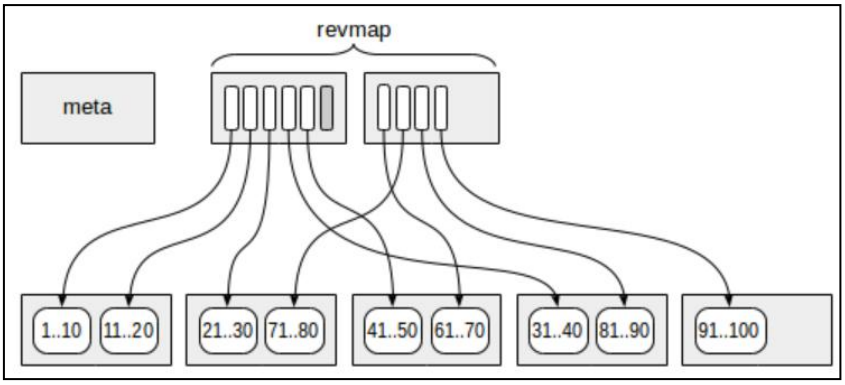
Ένα ευρετήριο BRIN εφαρμόζεται σε ένα πίνακα, όπου η τιμή κλειδιού ευρετηρίου ταξινομείται και υπολογίζεται εύκολα με μια συνάρτηση MinMax. Επίσης, είναι γεγονός όταν χρησιμοποιείται ένας δείκτης BRIN όχι μόνο θα ξεπεράσει σε ταχύτητα το B-Tree αλλά και θα εξοικονομήσει πάνω από το 99% του χώρου στο δίσκο.

2.3.1 Δομή BRIN

Το ευρετήριο BRIN "συνοψίζει" μεγάλα τμήματα δεδομένων σε μια συμπαγή μορφή, το οποίο μπορεί αποτελεσματικά να αποκλείσει πολλά από αυτά τα τμήματα από νωρίς, σε ένα ερώτημα βάσης δεδομένων. Μειώνοντας τον όγκο δεδομένων τόσο νωρίς, τόσο αντιπροσωπεύοντας μεγάλα τμήματα ως μικρές πλειάδες, όσο και εξαλείφοντας πολλά από αυτά, το BRIN μειώνει ουσιαστικά την ποσότητα λεπτομερών δεδομένων που πρέπει να εξεταστούν από τον κόμβο της βάσης δεδομένων ανά σειρά.

Σε αντίθεση με ένα παραδοσιακό ευρετήριο που εντοπίζει τις περιοχές του πίνακα που περιέχουν τιμές ενδιαφέροντος, το BRIN ενεργεί ως "αρνητικοί δείκτης", δείχνοντας τα τμήματα που σίγουρα δεν ενδιαφέρουν και επομένως δεν χρειάζεται να υποστούν περαιτέρω επεξεργασία.

- Η πρώτη φύλλο του δέντρου περιέχει τα μεταδεδομένα.
- Τα φύλλα με τις ‘συνοπτικές’ πληροφορίες των δεδομένων βρίσκονται σε μια ορισμένη μετατόπιση από τα μεταδεδομένα. Κάθε ευρετήριο σε αυτές τις σελίδες περιέχει συνοπτικές πληροφορίες για ένα εύρος.
- Μεταξύ των μεταδεδομένων και των περιληπτικών δεδομένων, βρίσκονται τα φύλλα αντίστροφης σειράς ή «revmap». Στην πραγματικότητα, αυτή είναι μια σειρά από δείκτες (TID) στις αντίστοιχες σειρές ευρετηρίου (δείτε Εικόνα 3).



Εικόνα 3: Δομή BRIN, για αποθήκευση BRIN ακλουθούμε την διαδρομή meta page, revmap pages και regular pages

Για ορισμένες σειρές, το ευρετήριο στο «revmap» δεν μπορεί να οδηγήσει σε καμία σειρά ευρετηρίου (με γκρι χρώμα στην εικόνα). Σε μια τέτοια περίπτωση, η σειρά θεωρείται ότι δεν έχει 'συνοπτικές' πληροφορίες ακόμη.

2.3.2 Πρόσθετα χαρακτηριστικά BRIN

Το μόνο που χρειάζεται για να λειτουργήσει η μέθοδος πρόσβασης BRIN είναι να εφαρμόσει μεθόδους που καθορίζονται από τον χρήστη, οι οποίες καθορίζουν τη συμπεριφορά των τιμών που είναι αποθηκευμένες στο ευρετήριο και τον τρόπο που αλληλεπιδρούν με τα κλειδιά σάρωσης.

Οι λειτουργίες υποστήριξης που απαιτούνται από το BRIN είναι οι εξής:

- opcInfo. Παρέχονται εσωτερικές πληροφορίες σχετικά με τις εγγραφές, που χρησιμοποιούνται για ευρετηρίαση.
- add_value. Δεδομένης μιας πλειάδας ευρετηρίου και μιας τιμής ευρετηρίου, τροποποιεί το υποδεικνυόμενο χαρακτηριστικό της πλειάδας έτσι ώστε να αντιπροσωπεύει τη νέα τιμή. Εάν πραγματοποιήθηκε οποιαδήποτε τροποποίηση στη πλειάδα (εγγραφή στον Πίνακα), επιστρέφει TRUE.
- Consistent. Ελέγχει εάν μια τιμή ταιριάζει με μια συνθήκη.
- Ένωση (Union). Υπολογίζει την ένωση δύο συνοπτικών καταχωρίσεων (ελάχιστες/μέγιστες τιμές)[28]

❖ Το BRIN index δεν υποστηρίζει ερωτήματα K-NN. Αυτό θα απαιτούσε μία νέα υποδομή στη PostgreSQL[10].

2.3.3 Λειτουργίες στο BRIN

Θα περιγράψουμε τις διάφορες λειτουργίες της μεθόδου BRIN.

1. Αναζήτηση και σάρωση ευρετηρίου (Search and index scan)

Πώς χρησιμοποιείται το ευρετήριο εάν δεν περιέχει αναφορές σε σειρές πινάκων;

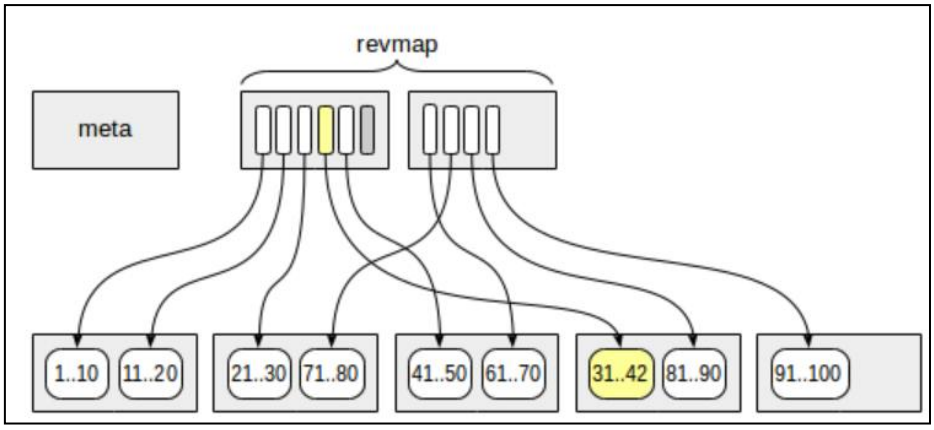
Αυτή η μέθοδος πρόσβασης, μπορεί να δημιουργήσει ένα bitmap. Μπορεί να υπάρχουν δύο είδη φύλλων bitmap: ακριβής, στη σειρά και ανακριβής, στο φύλλο.

- Το revmap σαρώνεται διαδοχικά.
- Οι δείκτες χρησιμοποιούνται για τον προσδιορισμό ευρετηρίων σε σειρές του πίνακα με περιληπτικές πληροφορίες για κάθε εύρος.
- Εάν ένα εύρος δεν περιέχει την ζητούμενη τιμή, παραλείπεται και εάν μπορεί να περιέχει την τιμή (ή οι περιληπτικές πληροφορίες δεν είναι διαθέσιμες), όλες οι σελίδες του εύρους προστίθενται στο bitmap.
- Το προκύπτον bitmap χρησιμοποιείται στη συνέχεια ως καθιερωμένο.

2. Εισαγωγή (Insert)

Κατά την προσθήκη μιας νέας καταχώρησης στο πίνακα, προσδιορίζουμε το εύρος στο οποίο περιέχεται και χρησιμοποιούμε το καθορισμένο εύρος(revmap) για να βρούμε τη σειρά ευρετηρίου με τις συνοπτικές πληροφορίες(ελ/μεγ).

- Για παράδειγμα, το μέγεθος ενός εύρους είναι τέσσερα και στο φύλλο 13, εμφανίζεται μια σειρά με την τιμή 42.
- Ο αριθμός του εύρους (ξεκινώντας από το μηδέν) είναι 13/4 = 3, επομένως, στο «revmap» παίρνουμε τον δείκτη με την μετατόπιση του 3 (Είναι ο αριθμός 4, σε κανονική σειρά).
- Η ελάχιστη τιμή για αυτό το εύρος είναι 31 και η μέγιστη τιμή είναι 40 (Εικόνα 4). Δεδομένου ότι η νέα τιμή 42 είναι εκτός του διαστήματος.
- Ενημερώνουμε τη μέγιστη τιμή. Αλλά εάν η νέα τιμή παραμένει εντός των αποθηκευμένων ορίων, το ευρετήριο δεν χρειάζεται να ενημερωθεί.



Εικόνα 4: Εισαγωγή της τιμής 42

3. Διαγραφή (Delete)

Όταν διαγράφεται μια σειρά, δεν συμβαίνει τίποτα. Μπορούμε να παρατηρήσουμε ότι μερικές φορές η ελάχιστη ή η μέγιστη τιμή θα διαγραφεί, οπότε το διάστημα θα μπορούσε να μειωθεί. Αλλά για να το ανιχνεύσουμε, θα πρέπει να διαβάσουμε όλες τις τιμές στο εύρος, και αυτό είναι δαπανηρό[25].

2.4 R*-Tree

Το R*-Tree[BKSS90] προτάθηκε από τους Norbert Beckmann, Hnas-Peter Kriegal, Ralf Schneider και Bernard Seeger το 1990. Στα R*-Tree εξετάζονται 4 παράμετροι.

1. Η ελαχιστοποίηση της περιμέτρου
2. Η κατά το δυνατόν βέλτιστη αξιοποίηση της χωρητικότητας των κόμβων του δένδρου
3. Η ελαχιστοποίηση της έκτασης κάθε παραλληλογράμμου
4. Η ελαχιστοποίηση της επικάλυψης μεταξύ των παραλληλογράμμων που τίθενται στα R-Tree

Σαφώς, και οι τέσσερις παράμετροι είναι συνυφασμένες μεταξύ τους, αφού λ.χ. η ελαχιστοποίηση των επικαλύψεων και του μεγέθους των παραλληλογράμμων ελαττώνει τη μέση πληρότητα των κόμβων[9].

Τα R*-Tree έχουν ελαφρώς υψηλότερο κόστος κατασκευής από τα τυπικά R-Tree, καθώς τα δεδομένα ενδέχεται να πρέπει να εισαχθούν ξανά (reinserted), αλλά το δέντρο που προκύπτει συνήθως θα έχει καλύτερη απόδοση ερωτημάτων[11].

2.4.1 Ανάλυση Παραμέτρων στα R*-Tree

Το R-Tree βασίζεται στη ελαχιστοποίηση του εμβαδού του κάθε MBR (minimum bounding rectangle).

Ανάλυση Παραμέτρων στα R*-Tree:

1. **Ελαχιστοποιεί το εμβαδό που καλύπτεται από το MBR (Minimizes of the area covered by each MBR).** Ο στόχος είναι να ελαχιστοποιηθεί το εμβαδό του χώρου που δεν χρησιμοποιείται, δηλαδή το εμβαδό που καλύπτεται από το MBR, αλλά όχι τα τμήματα ορθογωνίων που περιέχουν τα δεδομένων. Δηλαδή, προσπαθεί να μειώσει τον αριθμό των διαδρομών που πρέπει να διανύσει το ευρετήριο κατά τη διάρκεια της επεξεργασίας ερωτημάτων.
2. **Ελαχιστοποιεί την αλληλοεπικάλυψη των MBR (Minimizes the overlap of the MBR.).** Προσπαθεί να μειώσει την αλληλοεπικάλυψη μεταξύ MBR, καθώς μια μεγάλη επικάλυψη θα προκαλέσει μεγάλο αριθμό διαδρομών που ακολουθούνται για ένα ερώτημα.
3. **Ελαχιστοποιεί τα περιθώρια του MBR (Minimizes the margins of the MBR).** Το περιθώριο ορίζεται ως το άθροισμα όλων των πλευρών ενός MBR που είναι επίσης γνωστό και ως περίμετρος. Στοχεύει στη διαμόρφωση περισσότερων τετραγωνικών ορθογωνίων για τη βελτίωση της μεγαλύτερης απόδοσης των ερωτημάτων.
4. **Μεγιστοποίηση της χρήσης αποθήκευσης (Maximization the storage utilization).** Υπάρχει μια αύξηση στον αριθμό των κόμβων που γίνονται ερωτήματα όταν η χρήση είναι χαμηλή, ειδικά για ένα μεγαλύτερο ερώτημα. Επίσης, όποτε μειώνεται η χρήση του κόμβου, αυξάνεται το ύψος του δέντρου[22] [17].

2.4.2 Λειτουργίες στο R*-Tree

Το R*-Tree διαφέρει από το R-Tree στην τεχνική εισαγωγής και δεν χρησιμοποιεί εξειδικευμένο αλγόριθμο διαγραφής. Η διαγραφή στο R*-Tree είναι παρόμοια με τον αρχικό αλγόριθμο διαγραφής R-Tree.

1. Εισαγωγή (Insert)

Κατά την εισαγωγή, το R*-tree χρησιμοποιεί μια συνδυαστική στρατηγική.

- ChooseSubtree [Root Node]

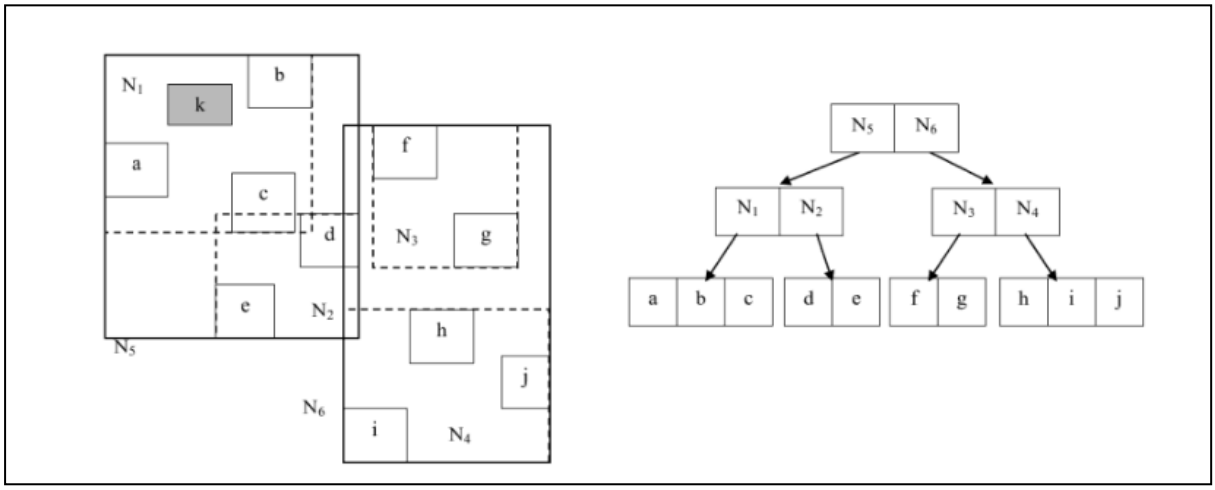
Επιλέγουμε την καταχώριση στη ρίζα της οποίας το MBR χρειάζεται τη λιγότερη μεγέθυνση εμβαδού για να εισαγάγουμε τη νέα καταχώριση.

- ChooseSubtree [leaf node]

Εδώ ακολουθεί το κριτήριο MBR (minimizing the overlap) όπου επιλέγει την καταχώριση της οποίας η διεύρυνση MBR απαιτεί μικρότερη αύξηση επικάλυψης από όλες τις καταχωρήσεις στον κόμβο.

2. Reinsert

Σε περίπτωση που το ChooseSubtree δεν μπορεί να βρει τον κόμβο με αρκετό χώρο για να εισαγάγει τη νέα καταχώρηση, αντί να διασπαστεί, του αφαιρείται ένα ποσοστό στοιχείων του (γύρω στο 30%, όπως έδειξαν πειραματικές μετρήσεις), τα οποία θα επανεισαχθούν στη συνέχεια.



Εικόνα 5: Παράδειγμα R-Tree[17]*

Αν κοιτάξουμε το παραπάνω παράδειγμα(δείτε Εικόνα 5), υποθέτουμε ότι ο κόμβος N1 είναι γεμάτος και το κεντροειδές του b είναι το πιο μακρινό κεντροειδές του N1 και το b θεωρείται για επανεισαγωγή. Η επανεισαγωγή βελτιώνει την απόδοση κατά την επεξεργασία ερωτημάτων καθώς προσπαθεί να εξισορροπήσει το δέντρο.

Δεδομένου ότι η επανεισαγωγή είναι μια πολύ δαπανηρή διαδικασία, περιορίζεται σε μία μόνο εφαρμογή επανεισόδου ανά επίπεδο.

Εάν η παραπάνω διαδικασία δεν είναι δυνατή, τότε γίνεται η διαδικασία διαχωρισμού (split).

3. Splitting

Το R*-Tree χρησιμοποιεί τοπολογική διάσπαση όταν πρέπει να γίνει διαχωρισμός. Αυτή η μέθοδος επιλέγει έναν άξονα διαχωρισμού με βάση την περίμετρο και στη συνέχεια ελαχιστοποιεί την επικάλυψη[9][22].

ΚΕΦΑΛΑΙΟ 3

ΓΕΩΓΡΑΦΙΚΑ ΠΛΗΡΟΦΟΡΙΑΚΑ ΣΥΣΤΗΜΑΤΑ & ΧΩΡΙΚΑ ΔΕΔΟΜΕΝΑ

3.1 Γεωγραφικά Πληροφοριακά Συστήματα (GIS)

Ο όρος GIS (Σύστημα Γεωγραφικών Πληροφοριών - Geographic Information System) περιγράφει οποιοδήποτε σύστημα πληροφοριών που ενσωματώνει, αποθηκεύει, επεξεργάζεται, αναλύει, διαμοιράζεται και απεικονίζει γεωγραφικές πληροφορίες. Η μοντελοποίησης και χαρτογραφική απόδοσης εκτεταμένων περιοχών με τη χρήση μεγάλου όγκου χωρικών δεδομένων, έχει οδηγήσει σε βαθμιαία εξάπλωση των εφαρμογών τους.

Με μία γενικότερη προσέγγιση, οι GIS εφαρμογές είναι εργαλεία τα οποία επιτρέπουν στους χρήστες να δημιουργούν αλληλεπιδραστικά ερωτήματα για την ανάλυση χωρικών πληροφοριών, επεξεργασία δεδομένων, χαρτών και να προβάλουν τα αποτελέσματα αυτών των ενεργειών τους[14][19].

Τέλος, είναι σημαντικό να επισημάνουμε ότι τα ΓΠΣ, εκτός από την απεικόνιση των γεωγραφικών οντοτήτων, συνέδεσαν την χωρική πληροφορία με την αντίστοιχη περιγραφική. Τα ΓΠΣ, δηλαδή, είναι μια βάση δεδομένων με δύο όψεις, την γεωγραφική και την περιγραφική. Κάθε γεωγραφικό στοιχείο συνδέεται με ένα μοναδικό κωδικό με μία εγγραφή στην περιγραφική βάση.

3.2 Απεικόνιση (Χωρικών Δεδομένων)

Τα GIS δεδομένα αναπαριστούν αντικείμενα του πραγματικού κόσμου (δρόμους, δάση, υψόμετρα) με ψηφιακά δεδομένα. Διακρίνονται σε 2 κατηγορίες

- Χωρικά δεδομένα (spatial data): προσδιορίζουν τα γεωμετρικά χαρακτηριστικά του στοιχείου και έχουν άμεση σχέση με τον εντοπισμό του.

- Περιγραφικά ή μη χωρικά δεδομένα (attributes): αναφέρονται σε χαρακτηριστικά ή ιδιότητες που αποδίδονται στο συγκεκριμένο στοιχείο του χώρου.

και αναπαρίστανται με δύο βασικές μορφές: τα δεδομένα διανυσματικής μορφής (vector) και τα δεδομένα κανονικοποιημένης ψηφιδωτής μορφής (raster). Η έρευνα μας έχει βασιστεί μόνο σε δεδομένα διανυσματικής μορφής 2 διαστάσεων, αλλά είναι εξίσου σημαντικό να αναφερθούμε και στα δύο.

3.2.1 Raster Δεδομένα

Ο τύπος Raster είναι παρόμοιος με αυτό της ψηφιακής εικόνας. Η μονάδα πληροφορίας σε μια ψηφιακή εικόνα είναι το pixel. Ο συνδυασμός των pixels δημιουργεί την εικόνα. Τα Raster δεδομένα αποθηκεύονται σε διάφορες μορφές όπως αρχεία TIF, JPEG, PNG κ.λ.π., ή με τη μορφή BLOBs σε σχεσιακές βάσεις δεδομένων. Επιπλέον Raster μπορεί να είναι ψηφιακές αεροφωτογραφίες, εικόνες από δορυφόρους, ψηφιακές εικόνες ή ακόμα και σαρωμένοι χάρτες.

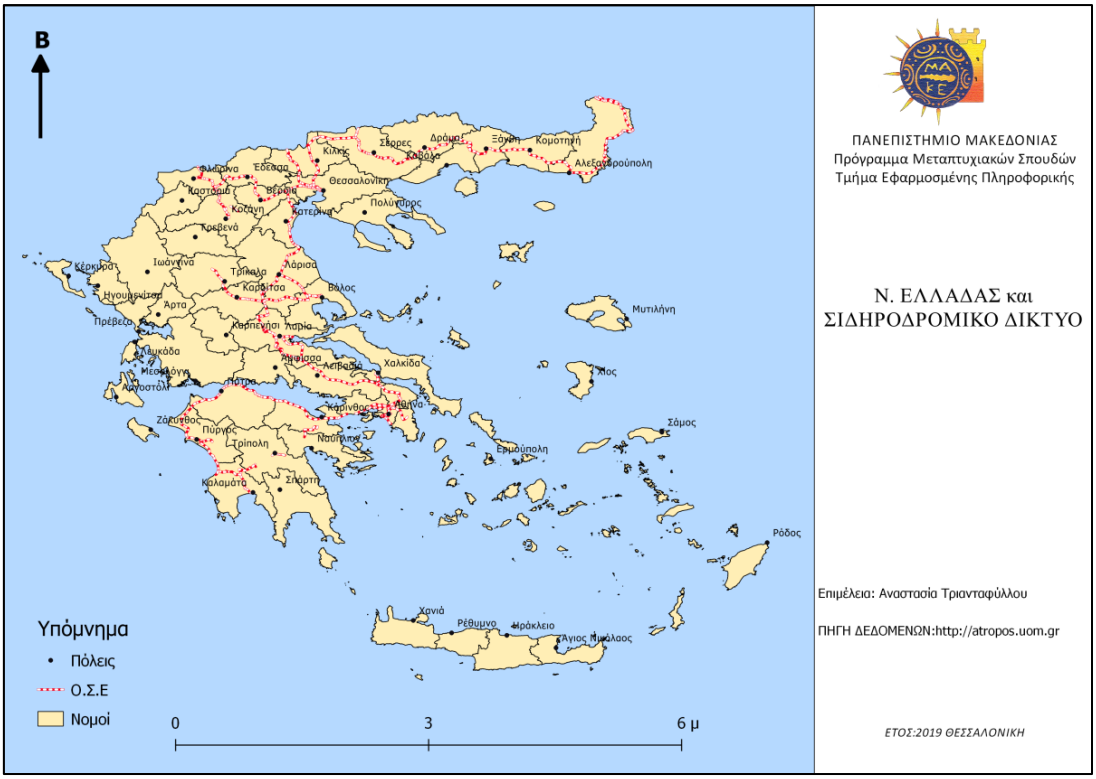
3.2.2 Vector δεδομένα

Τα δεδομένα διανυσματικής μορφής κατά κάποιο τρόπο αναπαριστούν χαρακτηριστικά του πραγματικού κόσμου μέσω του gis. Ένα χαρακτηριστικό είναι οτιδήποτε μπορούμε να δούμε στο τοπίο.

Διαφορετικά γεωγραφικά χαρακτηριστικά μπορούν να εκφραστούν από διαφορετικούς τύπους γεωμετρίας(δείτε Εικόνα 6):

- Σημεία
- Γραμμές
- Πολύγωνα

Κάθε ένα vector στοιχείο μπορεί να σχετίζεται με μία εγγραφή σε μια βάση δεδομένων, όπου εκτός από τη γεωμετρία του μπορεί να περιγράφονται τα χαρακτηριστικά του.



Εικόνα 6: Vector χάρτης με σημεία, γραμμές και πολύγωνα

3.3 Shapefile (.shp)

Η μορφή Shapefile της ESRI, είναι από τις συνηθέστερες για την αποθήκευση GIS πληροφοριών. Τα Shapefiles αποθηκεύουν μη-τοπολογικά διανυσματικά δεδομένα μαζί με τα σχετικά χαρακτηριστικά τους. Αναπτύχθηκε από το Esri και αποτελεί μία ανοικτή μορφή και επιπλέον είναι μια δημοφιλής επιλογή μεταφοράς δεδομένων. Για παράδειγμα, τα .shp αρχεία μπορούν να διαβαστούν απευθείας από μια σειρά προγραμμάτων λογισμικού GIS όπως το ArcGIS και το QGIS. Ένα shapefile είναι στην πραγματικότητα μια συλλογή από τουλάχιστον τρία βασικά αρχεία: .shp, .shx και .dbf. Και τα τρία αρχεία πρέπει να υπάρχουν στον ίδιο κατάλογο για να είναι ορατά. Μπορεί να υπάρχουν πρόσθετα αρχεία, όπως ένα αρχείο .prj με τις πληροφορίες προβολής των shapefiles[13].

3.4 Προεργασία δεδομένων

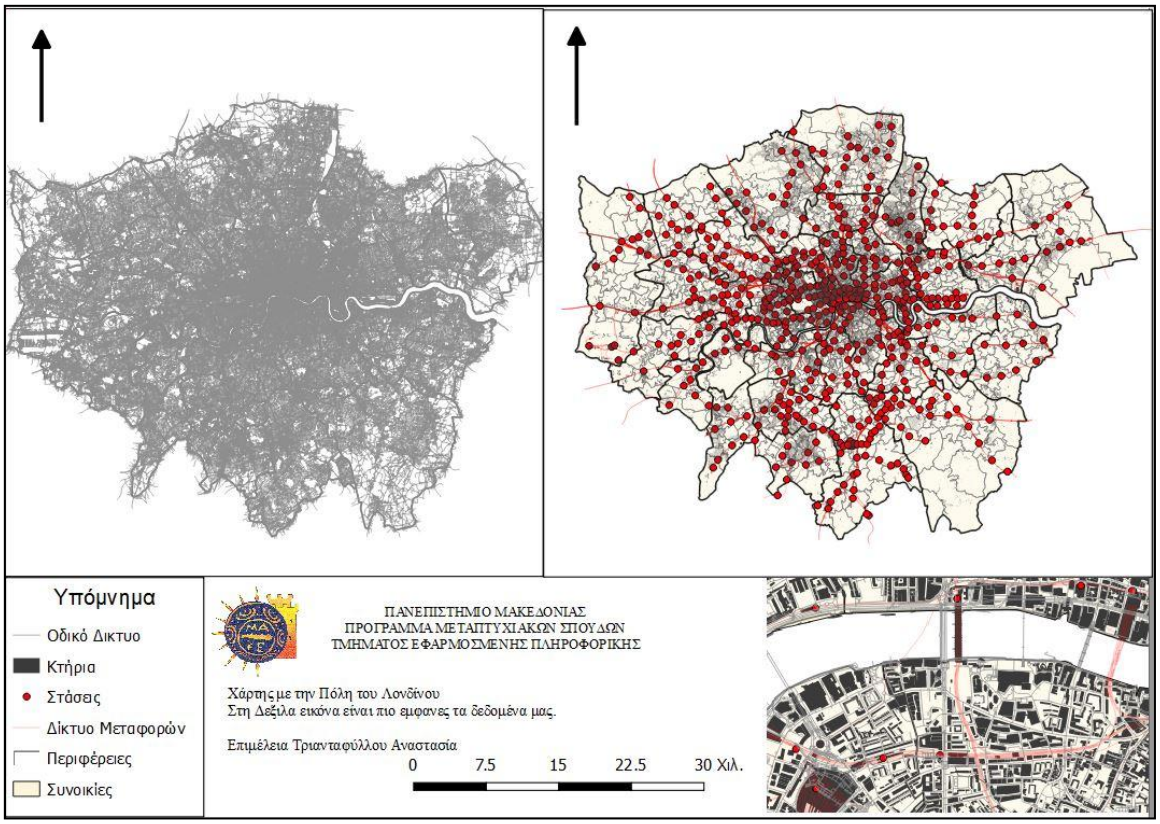
Τα shapefiles που χρησιμοποιήσαμε όπως αναφέρθηκε και στο προηγούμενο κεφάλαιο είναι στοιχεία αναφοράς της πόλης του Λονδίνου στη Μεγάλη Βρετανία(δείτε Εικόνα 7). Οι

εγγραφές αποτελούνται από σημεία: stations, γραμμές: roads, railways, πολύγωνα: borough, buildings, wards.

Τα αρχεία αυτά διαφέρουν από την αρχική τους δομή. Είναι αρχεία από 4 διαφορετικές πηγές οπότε χρειαζόταν επεξεργασία πριν τα τοποθετήσουμε στις βάσεις δεδομένων. Σημαντικό εργαλείο που μας βοήθησε στην καλύτερη κατανόηση των δεδομένων είναι το QGIS. Το QGIS διανέμεται δωρεάν και είναι ανοιχτού κώδικα λογισμικό, που επεξεργάζεται γεωγραφικές πληροφορίες. Υποστηρίζει λειτουργίες απεικόνισης, επεξεργασίας, και ανάλυσης.

Αυτό που κάναμε στην αρχή είναι να ελέγξουμε αν τα δεδομένα βρίσκονται στο ίδιο προβολικό σύστημα. Μεταφέραμε και αποθηκεύσαμε εκ νέου τα δεδομένα σε EPSG:27700 OSGB 1936/British National Grid.

Τέλος, επεξεργαστήκαμε τις στήλες του κάθε .shp αρχείου. Κάναμε Delete αυτές που δεν χρειαζόμαστε στην ανάλυση και προσθέσαμε μη-χωρικά δεδομένα όπως ο πληθυσμός από .csv αρχείο με την διαδικασία join.



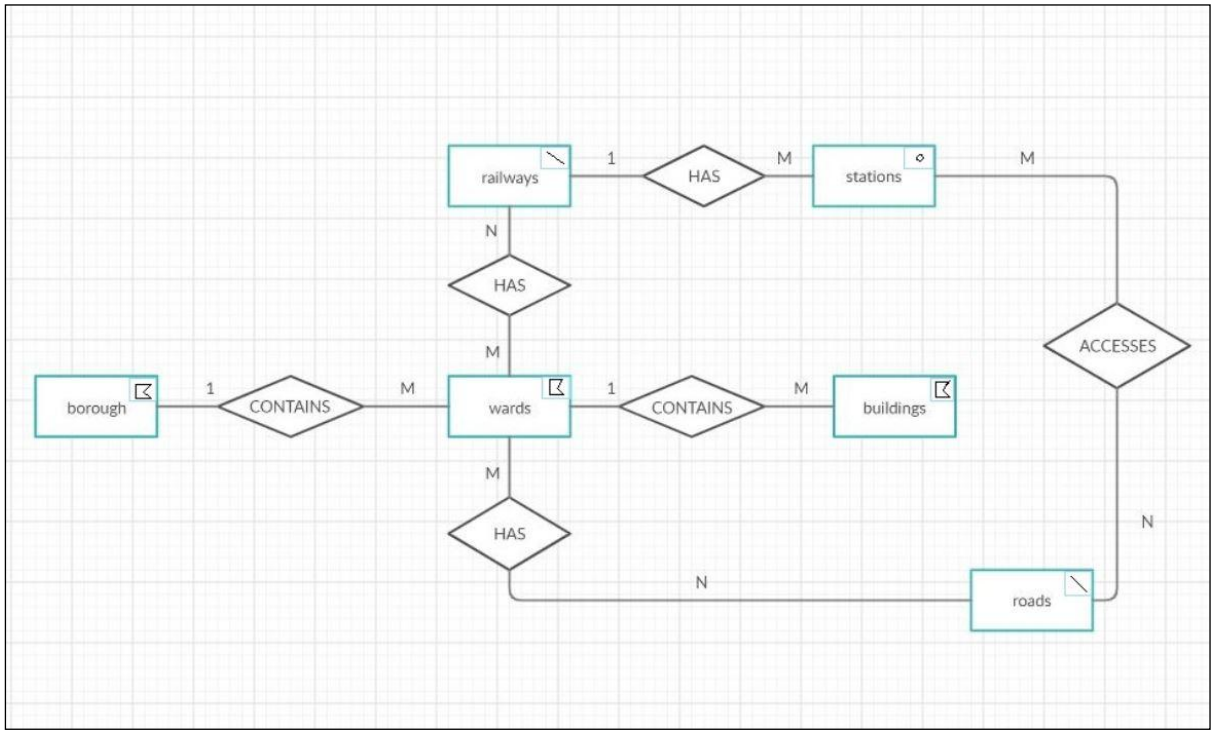
Εικόνα 7: Χάρτης με τα δεδομένα της έρευνας

Πηγές Δεδομένων:

- https://data.london.gov.uk/, OpenStreetMap, Greater London Authority (GLA) (Τελευταία Πρόσβαση: 07/05/2020) [34]
- https://data.london.gov.uk/, Statistical GIS Boundary Files for London, Greater London Authority (GLA) (Τελευταία Πρόσβαση: 07/05/2020) [35]
- https://www.arcgis.com/, London rail network (Τελευταία Πρόσβαση: 07/05/2020) [36]
- https://www.ordnancesurvey.co.uk/, Τελευταία Πρόσβαση: 07/05/2020) [37]

3.5 Διάγραμμα σχέσης οντοτήτων

Ένα διάγραμμα σχέσης οντότητας (ERD), όπως φαίνεται στην Εικόνα 8, δείχνει τις σχέσεις των συνόλων οντοτήτων που είναι αποθηκευμένες σε μια βάση δεδομένων..



Εικόνα 8: Διάγραμμα ER για τα δεδομένα του Λονδίνου, με εικονογράμματα

ΚΕΦΑΛΑΙΟ 4
ΠΟΛΥΔΙΑΣΤΑΤΑ ΔΕΔΟΜΕΝΑ

4.1 Εισαγωγή στα Πολυδιάστατα Δεδομένα

Ο όρος "πολυδιάστατο" τείνει να εφαρμόζεται μόνο σε σύνολα δεδομένων με τρεις ή περισσότερες διαστάσεις. Οι διαστάσεις είναι κατηγορίες που χρησιμοποιούνται για την ταξινόμηση δεδομένων όπως ο χρόνος, η γεωγραφία, τα τμήματα μιας εταιρείας, οι σειρές προϊόντων και ούτω καθεξής. Τα αποτελέσματα που σχετίζονται με ένα συγκεκριμένο σύνολο διαστάσεων ονομάζονται γεγονότα. Τα γεγονότα είναι συνήθως αριθμοί που σχετίζονται με πωλήσεις προϊόντων, κέρδη, όγκους, μετρήσεις κ.λπ.

Όταν αναλύουμε και κατηγοριοποιούμε δεδομένα βάση πολλαπλών διαστάσεων και μετρήσεων, η διαδικασία ονομάζεται πολυδιάστατη ανάλυση δεδομένων. Βοηθά στη μαθηματική μοντελοποίηση των επιχειρηματικών διαδικασιών, όπου εμπλέκονται μεγάλα και πολύπλοκα δεδομένα σε πολλές περιοχές και προϊόντα. Η ανάλυση που προκύπτει από αυτά τα μοντέλα μπορεί να βοηθήσει στη λήψη αποφάσεων και στο σχεδιασμό δραστηριοτήτων επιχειρηματικών δραστηριοτήτων.

4.2 Χαρακτηριστικά Διαστάσεων

Τα δεδομένα μπορεί να προέρχονται από πολλές πηγές και να περιλαμβάνουν πολλές διαστάσεις. Οι δυνατότητες της πολυδιάστατης ανάλυσης περιλαμβάνουν διαστάσεις(dimensions), ιεραρχίες(hierarchies), μέλη(members), τίτλους(titles), τιμές(values), στιγμιότυπα (instances) και σημείακά δεδομένων(data points).

- Η διάσταση αποτελεί δομικό στοιχείο του κύβου, και συντίθεται από σχετιζόμενα δεδομένα, και ιεραρχούμενα μέλη.
- Το μέλος μιας διάστασης αποτελεί ένα από τα ομοειδή στοιχεία που την απαρτίζουν.

- Η ιεράρχηση μιας διάστασης ταξινομεί τα μέλη της με βάση τη σχέση parent/child.
- Ο τίτλος της διάστασης είναι το όνομα με το οποίο είναι γνωστή.
- Η τιμή ενός μέλους μιας διάστασης είναι ένα στιγμιότυπο του μέλους.

Ένα data point είναι η τομή πολλαπλών διαστάσεων[16].

Τα συστήματα υπολογιστών για το multidimensional analysis(MDA) περιλαμβάνουν Online αναλυτική επεξεργασία (OLAP) για δεδομένα σε σχεσιακές βάσεις δεδομένων και πίνακες pivot, για δεδομένα σε υπολογιστικά φύλλα[6].

4.3 Πολυδιάστατα Δεδομένων

Τα πολυδιάστατα δεδομένα που θα χρησιμοποιήσουμε στην έρευνα μας είναι τα Δεδομένα αναγνώρισης γραμμάτων(Letter Image Recognition Data). Τα δεδομένα αυτά δημιουργήθηκαν από τον David J. Slate, Odesta Corporation; 1890 Maple Ave; Suite 115; Evanston, IL 60201, τον Ιανουάριο το 1991. Περιέχει 20.000 εγγραφές και ο στόχος είναι να προσδιοριστεί κάθε ένας από τους μεγάλους αριθμούς ασπρόμαυρων ορθογώνιων pixel ως ένα από τα 26 κεφαλαία γράμματα του αγγλικού αλφαβήτου. Οι εικόνες χαρακτήρων βασίστηκαν σε 20 διαφορετικές γραμματοσειρές και κάθε γράμμα σε αυτές τις 20 γραμματοσειρές παραμορφώθηκε τυχαία για να παράγει ένα αρχείο 20.000 μοναδικών ερεθισμάτων.

Κάθε ερέθισμα μετατράπηκε σε 16 αριθμητικά χαρακτηριστικά, τα οποία στη συνέχεια κλιμακώθηκαν ώστε να χωρέσουν σε μια σειρά ακέραιων τιμών από 0 έως 15.

Αριθμός χαρακτηριστικών: 17 (κατηγορία γραμμάτων και 16 αριθμητικά χαρακτηριστικά)

Attribute Information:

1. letter capital letter (26 values from A to Z)
2. x-box horizontal position of box (integer)
3. y-box vertical position of box (integer)
4. width width of box (integer)
5. high height of box (integer)
6. onpix total # on pixels (integer)
7. x-bar mean x of on pixels in box (integer)
8. y-bar mean y of on pixels in box (integer)
9. x2bar mean x variance (integer)

10. y2bar mean y variance (integer)
11. xybar mean x y correlation (integer)
12. x2ybr mean of x * x * y (integer)
13. xy2br mean of x * y * y (integer)
14. x-ege mean edge count left to right (integer)
15. xegvy correlation of x-ege with y (integer)
16. y-ege mean edge count bottom to top (integer)
17. yegvx correlation of y-ege with x (integer)

4.4 Προεργασία δεδομένων

Τα δεδομένα μας αποτελούνται από αριθμούς τύπου integer. Για να κατασκευάσουμε ερωτήματα, θα πρέπει πρώτα να βγάλουμε κάποια στατιστικά στοιχεία, ώστε να δούμε που κυμαίνονται αυτοί οι αριθμοί. Για να βγάλουμε στατιστικά συμπεράσματα θα χρησιμοποιήσουμε το R-Studio. Το R-Studio είναι ένα ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) για την R, μια γλώσσα προγραμματισμού για στατιστικούς υπολογισμούς και γραφικά.

Αρχικά θα εισάγουμε το .csv αρχείο στη R με την παρακάτω εντολή.

letter= read.csv(file = 'C:\\Users\\DATA\\letter-recognition.csv')***

summary(letter)

Πίνακας 1: Στατιστικά δεδομένα letter-recognition

	Min.	**1st Qu.**	**Median**	**Mean.**	**3rd Qu.**	**Max.**
xbox	0.000	3.000	4.000	4.024	5.000	15.000
ybox	0.000	5.000	7.000	7.035	9.000	15.000
width	0.000	4.000	5.000	5.122	6.000	15.000
height	0.000	4.000	6.000	5.372	7.000	15.000
onpix	0.000	2.000	3.000	3.506	5.000	15.000
xbar	0.000	6.000	7.000	6.898	8.000	15.000
ybar	0.000	6.000	7.000	7.500	9.000	15.000
x2bar	0.000	3.000	4.000	4.629	6.000	15.000
y2bar	0.000	4.000	5.000	5.179	7.000	15.000
xybar	0.000	7.000	8.000	8.282	10.000	15.000
x2ybar	0.000	5.000	6.000	6.454	8.000	15.000
xy2bar	0.000	7.000	8.000	7.929	9.000	15.000
xedge	0.000	1.000	3.000	3.046	4.000	15.000
xedgey	0.000	8.000	8.000	8.339	9.000	15.000
yedge	0.000	2.000	3.000	3.692	5.000	15.000
yedgex	0.000	7.000	8.000	7.801	9.000	15.000

Πίνακας 2: Πίνακας Συχνοτήτων για κάθε αριθμό στις 16 στήλες των δεδομένων letter-recognition (κόκκινο-μεγαλύτερη συχνότητα)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
xbox	132	1261	2909	**4157**	4477	3169	1894	1006	513	284	121	48	20	4	3	2
ybox	709	778	530	1330	1342	1566	1705	2302	2180	**2702**	2211	1625	321	271	198	230
width	195	385	1285	1994	3816	**4262**	3641	1946	1418	679	237	91	39	6	4	2
height	365	883	1304	1559	2718	2675	**3656**	2695	3559	347	103	76	31	10	15	4
onpix	830	2437	**4153**	3939	3157	2153	1379	857	519	283	142	85	40	13	7	6
xbar	148	179	167	680	1069	1780	2717	**6024**	4019	1752	802	338	201	67	43	14
ybar	47	134	393	488	714	877	2506	**6010**	3753	1599	1252	1131	663	192	165	76
x2bar	422	1084	2693	**3424**	3199	2982	2340	1422	1013	436	235	150	158	120	184	138
y2bar	269	822	1852	1914	3011	**3243**	3099	2623	1688	874	318	128	48	31	46	34
xybar	145	62	97	90	267	819	2751	**5578**	1668	1971	2563	1799	1043	772	279	96
x2ybar	188	430	887	726	1495	2457	**5666**	2598	1495	1408	888	935	472	200	135	20
xy2bar	1	9	67	269	701	1335	1874	2807	**6548**	3000	1437	926	409	313	2019	85
xedge	2461	2568	4213	**4779**	1500	1363	1264	722	587	246	154	81	29	17	12	4
xedgey	1	13	17	34	79	416	1592	2516	**7624**	3437	2394	1437	348	72	16	4
yedge	2472	2040	2475	3078	**3091**	2048	1723	1227	973	613	154	64	22	13	3	4
yedgex	2	17	30	130	478	992	1827	3472	**8047**	2358	1578	868	137	49	13	2

Ο παραπάνω πίνακας παράγεται με την εξής εντολή στην R. Την ίδια εντολή την επαναλαμβάνουμε και για τις 16 στήλες.

table(letter$xedgey)

Όλα αυτά τα στατιστικά δεδομένα θα μας βοηθήσουνε στη κατασκευή ερωτημάτων εύρους που θα δούμε σε ένα από τα επόμενα κεφάλαια.

ΚΕΦΑΛΑΙΟ 5 – ΕΠΙΣΚΟΠΗΣΗ ΤΩΝ DBMS

5.1 Εισαγωγή

Μία βάση δεδομένων είναι μια συλλογή δεδομένων ή εγγραφών. Ένα σύστημα διαχείρισης (DBMS) είναι ένα σύστημα λογισμικού που χρησιμοποιεί μια τυπική μέθοδο αποθήκευσης και οργάνωσης δεδομένων. Τα δεδομένα μπορούν να προστεθούν (added), να ενημερωθούν (update), να διαγραφούν (deleted) ή να διασταυρωθούν (traversed) χρησιμοποιώντας διάφορους αλγόριθμους και ερωτήματα.

5.2 Ταξινόμηση Βάσεων Δεδομένων

Ένας τρόπος ταξινόμησης βάσεων δεδομένων περιλαμβάνει τον τύπο του περιεχομένου τους, για παράδειγμα: βιβλιογραφικά, κείμενα εγγράφων, στατιστικά ή αντικείμενα πολυμέσων. Ένας άλλος τρόπος είναι από την περιοχή εφαρμογής τους, για παράδειγμα: λογιστική, μουσικές συνθέσεις, ταινίες, τραπεζικές εργασίες, κατασκευή ή ασφάλιση. Ένας τρίτος τρόπος είναι από κάποια τεχνική άποψη, όπως η δομή της βάσης δεδομένων ή ο τύπος διεπαφής.

5.2.1 Spatial Databases

Ο όρος "σύστημα χωρικών βάσεων δεδομένων" έχει γίνει δημοφιλής τα τελευταία χρόνια, σε κάποιο βαθμό μέσω μιας σειράς συνεδρίων με τίτλο “Symposium on Large Spatial Databases (SSD)” που διοργανώνονται δύο φορές το χρόνο από το 1989 [Buch89, GünS91, AbO93] και συνδέεται με μια άποψη μιας βάσης δεδομένων που περιέχει σύνολα αντικειμένων στο χώρο και όχι εικόνες ή φωτογραφίες ενός χώρου[12].

Τα χωρικά συστήματα βάσεων δεδομένων παρέχουν την υποκείμενη τεχνολογία βάσεων δεδομένων για γεωγραφικά συστήματα πληροφοριών και άλλες εφαρμογές. Στην πιο βασική τους μορφή, το σύστημα χωρικής βάσης δεδομένων χρησιμοποιείται για την αποθήκευση,

τον υπολογισμό και την ανάκτηση χωρικών αντικειμένων όπως σημεία, γραμμές και πολύγωνα.

Σε πολλές εφαρμογές αυτές τις μέρες, διαχειριζόμαστε 2D γεωγραφικά, γεωμετρικά ή χωρικά δεδομένα. Για τη διαχείριση υψηλού επιπέδου δεδομένων 3D, όπως ο ανθρώπινος εγκέφαλος ή τη διάταξη μοριακών πρωτεϊνών σε ανθρώπινο σώμα, απαιτούνται οι διαδικασίες εγκατάστασης, αποθήκευσης και ανάκτησης. Μετά την εμφάνιση συστημάτων σχεσιακής βάσης δεδομένων υπήρξαν προσπάθειες για τη διαχείριση τέτοιων δεδομένων σε συστήματα βάσεων δεδομένων. Χαρακτηριστικό για την τεχνολογία που αναδύεται για την αντιμετώπιση αυτών των αναγκών είναι η ικανότητα αντιμετώπισης μεγάλων συλλογών σχετικά απλών γεωμετρικών αντικειμένων, για παράδειγμα, ενός συνόλου 100.000 πολυγώνων. Αυτό είναι κάπως διαφορετικό από τις βάσεις δεδομένων CAD (solid modeling κ.λπ.), όπου οι γεωμετρικές οντότητες συντίθενται ιεραρχικά σε πολύπλοκες δομές[2][12].

Σύμφωνα με τον Ralf Hartmut, υπάρχουν τρία προαπαιτούμενα για ένα σύστημα χωρικών βάσεων δεδομένων.

- Πρώτον, τονίζει το γεγονός ότι οι χωρικές ή γεωμετρικές πληροφορίες στην πράξη συνδέονται πάντοτε με "μη χωρικά" (π.χ. αλφαριθμητικά) δεδομένα. Δηλαδή, ένα σύστημα χωρικών βάσεων δεδομένων είναι ένα πλήρες σύστημα βάσης δεδομένων με πρόσθετες δυνατότητες χειρισμού χωρικών δεδομένων.
- Δεύτερον, οι τύποι χωρικών δεδομένων (σημεία, γραμμές και πολύγωνα) παρέχουν μια θεμελιώδη αρχή, για τη μοντελοποίηση της δομής των γεωμετρικών οντοτήτων στο χώρο, καθώς και των σχέσεών τους και των λειτουργιών τους.
- Τρίτον, ένα σύστημα πρέπει να είναι ικανό να ανακτήσει δεδομένα από μια μεγάλη συλλογή αντικειμένων χωρίς να σαρώνει όλα τα αντικείμενα. Για αυτό το λόγο χρειαζόμαστε τα χωρικά ευρετήρια[12].

5.2.2 Data Warehouse

Όσο αφορά το πολυδιάστατο σύστημα διαχείρισης βάσεων δεδομένων (MDBMS) είναι ένα σύστημα διαχείρισης βάσεων δεδομένων που χρησιμοποιεί έναν υπερ-κύβο δεδομένων ως ιδέα για την αναπαράσταση πολλαπλών διαστάσεων των δεδομένων που διατίθενται στους

χρήστες. Αυτή η βάση δεδομένων χρησιμοποιείται για εφαρμογές σε αποθήκες δεδομένων και διαδικτυακής αναλυτικής επεξεργασίας[5].

Στη σημερινή εποχή κατακλυζόμαστε από ένα μεγάλο όγκο δεδομένων, τα οποία προέρχονται από εσωτερικές και εξωτερικές πηγές. Τα δεδομένα αυτά, αν και πολύτιμα για την εξαγωγή πληροφορίας και τη λήψη αποφάσεων, είναι υπερβολικά λεπτομερή, διασκορπισμένα σε διάφορες πηγές, ανομοιογενή και πάσχουν από σφάλματα, ελλείψεις, κλπ. Μια Αποθήκη Δεδομένων (ΑΔ) (Data Warehouse (DW)) είναι μια βάση δεδομένων διαφορετική από τις βάσεις δεδομένων που τηρούν τα λειτουργικά δεδομένα του οργανισμού. Τα δεδομένα απαλλάσσονται από προβλήματα, ομογενοποιούνται, αποθηκεύονται σε συγκεντρωτική μορφή και χρησιμοποιούνται για ανάλυση, εξαγωγή συμπερασμάτων και λήψη αποφάσεων.

Ειδικότερα, τα τέσσερα βασικά χαρακτηριστικά της ΑΔ έχουν ως εξής:

- Θεματικός Προσανατολισμός: Η πληροφορία στις ΑΔ είναι οργανωμένη με βάση το περιεχόμενο των δεδομένων, όπως οι πωλήσεις σε μία επιχείρηση. Στόχος των ΑΔ είναι να συγκεντρώσει και να οργανώσει την πληροφορία αυτή αποκλείοντας τις μη χρήσιμες πληροφορίες, έτσι ώστε ο επιχειρηματίας να διευκολυνθεί στη λήψη αποφάσεων.
- Ολοκλήρωση: Όπως έχουμε ήδη αναφέρει προηγουμένως κάθε δευτερόλεπτο έχουμε καινούρια δεδομένα, τα οποία συγκεντρώνονται από διάφορες πηγές και αυτό τα κάνουν να πάσχουν από προβλήματα. Για παράδειγμα διαφορετικές μονάδες μέτρησης ή ονομασίες. Για να εισέλθουν τα δεδομένα σε μια ΑΔ θα πρέπει να ομογενοποιηθούν και στη συνέχεια αποθηκεύονται χωρίς προβλήματα.
- Χρονική Διαφοροποίηση: Οι ΑΔ διατηρούν ιστορική πληροφορία. Αυτό σημαίνει ότι δίνει τη δυνατότατα διεξαγωγής συγκρίσεων και η αναγνώριση τάσεων. Πολλά συστήματα, όπως αυτό των Συναλλαγών διατηρούν τρέχουσες πληροφορίες αντιθέτως οι ΑΔ διατηρούν πληροφορίες, που μπορεί να αναφέρονται σε βάθος χρόνου μέχρι και δεκαετίας.
- Μη ευμετάβλητα δεδομένα: Τα δεδομένα των Συστημάτων Επεξεργασίας Συναλλαγών τελούν υπό συνεχή ανανέωση, καθώς οι χρήστες διαρκώς εισάγουν, τροποποιούν και διαγράφουν δεδομένα. Αντιθέτως, στις ΑΔ τα δεδομένα μεταφέρονται μαζικά σε συγκεκριμένες χρονικές στιγμές, και στη συνέχεια προσπελαύνονται με σκοπό την ανάλυση τους, αλλά δεν τροποποιούνται[8].

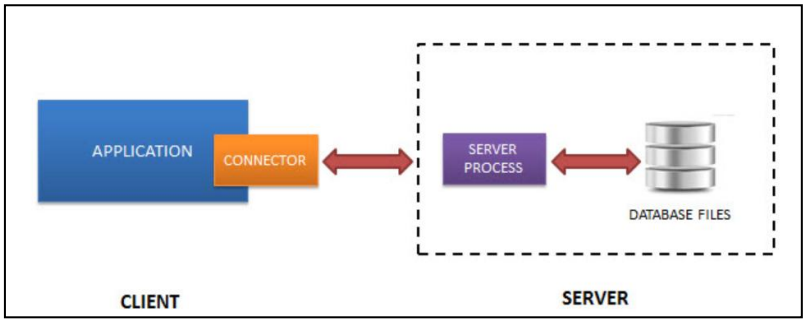
Στο κεφάλαιο 5, θα μελετήσουμε δύο διαφορετικά είδη βάσεων δεδομένων που έχουν χρησιμοποιηθεί ως εργαλεία για τη διεξαγωγή ερωτημάτων. Οι βάσεις δεδομένων που έχουμε χρησιμοποιήσει είναι:

1. PostgreSQL Database Management System
2. SQLite Database Management System

5.3 PostgreSQL

Η PostgreSQL αποτελεί μια ανοικτού κώδικα σχεσιακή βάση δεδομένων (σχέση-αντικειμένου) με πολλές δυνατότητες (ORDBMS). Χρησιμοποιείται για την ασφαλή αποθήκευση δεδομένων, υποστηρίζοντας βέλτιστες πρακτικές και επιτρέποντας την ανάκτηση τους κατά την επεξεργασία ενός αιτήματος. Η PostgreSQL (που επίσης προφέρεται Post-gress-Q-L) αναπτύχθηκε από την PostgreSQL Global Development Group (μια παγκόσμια ομάδα εθελοντών). Δεν ελέγχεται από καμία εταιρεία ή άλλο ιδιωτικό φορέα. Είναι ανοικτού κώδικα και ο πηγαίος κώδικας του είναι διαθέσιμος δωρεάν[31].

Η PostgreSQL, απαιτεί ξεχωριστή διαδικασία διακομιστή για να λειτουργήσει. Οι εφαρμογές που επιθυμούν να έχουν πρόσβαση στον διακομιστή βάσης δεδομένων χρησιμοποιούν πρωτόκολλο TCP/IP για την αποστολή και λήψη αιτημάτων. Αυτό ονομάζεται αρχιτεκτονική πελάτη/διακομιστή (client/server)(δείτε Εικόνα 9).



Εικόνα 9: RDBMS client/server architecture, https://www.sqlitetutorial.net/what-is-sqlite/, SQLite Tutorial

5.3.1 Χαρακτηριστικά της PostgreSQL

Υποστηρίζει ένα μεγάλο μέρος του προτύπου SQL και προσφέρει πολλά σύγχρονα χαρακτηριστικά:

- complex queries
- foreign keys
- triggers
- updatable views
- transactional integrity
- multiversion concurrency control

Επίσης, η PostgreSQL μπορεί να επεκταθεί από τον χρήστη με πολλούς τρόπους, για παράδειγμα προσθέτοντας νέα:

- data types
- functions
- operators
- aggregate functions
- index methods
- procedural languages

Επειδή είναι ανοιχτού κώδικα, η PostgreSQL μπορεί να χρησιμοποιηθεί, να τροποποιηθεί και να διανεμηθεί από οποιονδήποτε δωρεάν για οποιονδήποτε σκοπό, είτε ιδιωτικό είτε εμπορικό ή ακαδημαϊκό. Τέλος, η PostgreSQL είναι πολλαπλή πλατφόρμα και λειτουργεί σε πολλά λειτουργικά συστήματα όπως το Linux, το FreeBSD, το OS X, το Solaris και τα Microsoft Windows κ.λπ.

5.3.2 PostgreSQL και PostGIS

Το PostGIS είναι μία επέκταση (extension) ανοιχτού λογισμικού για τη σχεσιακή βάση δεδομένων PostgreSQL που δίνει την δυνατότητα υποστήριξης γεωγραφικών αντικειμένων στη βάση και δημιουργήθηκε από την Refractions Research Inc ως ερευνητικό έργο τεχνολογίας χωρικών δεδομένων. Στην πραγματικότητα, το PostGIS "ενεργοποιεί χωρικά" τον διακομιστή της PostgreSQL, επιτρέποντάς του να χρησιμοποιηθεί ως χωρική βάση

δεδομένων για γεωγραφικά συστήματα πληροφοριών (GIS), όπως το SDE της ESRI ή ως χωρική επέκταση της Oracle.

Επιπλέον οι χωρικοί τύποι όπως γεωμετρία, raster, vector και ευρετήρια κ.α, προστίθενται από το PostGIS στη βάση δεδομένων PostgreSQL με την βοήθεια των μεθόδων functions, operators and indexing που καθιστούν το σύστημα διαχείρισης βάσεων δεδομένων PostgreSQL πιο γρήγορο[32].

Κύριες λειτουργίες:

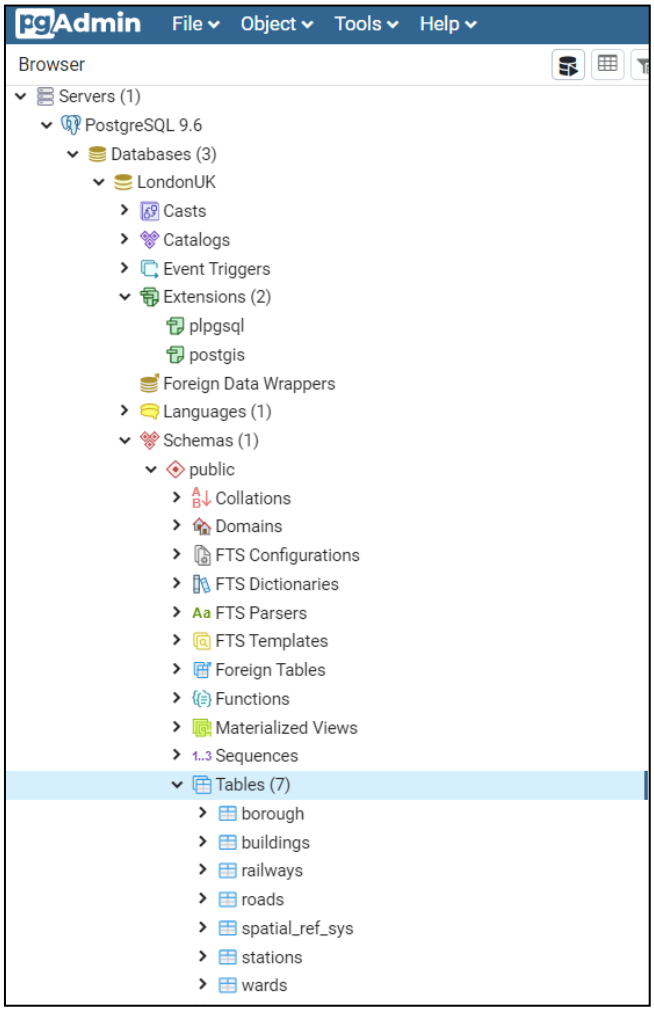
- Προσθέτει ένα τύπο δεδομένων «γεωμετρίας» στους συνηθισμένους τύπους των βάσεων δεδομένων(π.χ. varchar, integer, date).
- Προσθέτει νέες συναρτήσεις που εισάγουν τον τύπο «γεωμετρίας» και επιστρέφουν πίσω χρήσιμες πληροφορίες(π.χ. ST_Distance(geometry, geometry).
- Προσθέτει έναν μηχανισμό ευρετηρίασης (indexing) για να επιτρέψει στα ερωτήματα με χωρικούς περιορισμούς να επιστρέφουν εγγραφές πολύ γρήγορα από μεγάλους πίνακες δεδομένων[18].

5.3.3 Εισαγωγή των .shp Αρχείων στην PostgreSQL

Γενικά είναι πιο ασφαλές να επιλέγετε σταθερές εκδόσεις των λογισμικών που χρησιμοποιείτε. Στην περίπτωσή μας θα εγκαταστήσουμε την PostgreSQL 9.6 και το PostGIS 2.5.

1. Κατεβάσαμε από το https://www.enterprisedb.com/downloads/postgres-postgresqldownloads#windows τον installer της PostgreSQL
2. Εκτελέσαμε το αρχείο που κατέβηκε στο προηγούμενο βήμα και ονομάζεται postgresql-9.6.13-windows.exe. Όταν μας ζητηθηκε, δώσαμε password για τον superuser postgres (π.χ., 123). Επιλέξαμε ελληνικό locale.
3. Όταν ολοκληρώθηκε η εγκατάσταση του server, επιλέξαμε Finish και συνεχίσαμε με τον Stack Builder. Στον Stack Builder επιλέξαμε τον τοπικό server (PostgreSQL 9.6 on port 5432) και Next.
4. Στην επόμενη οθόνη επιλέξαμε για εγκατάσταση από τα Spatial Extensions το "PostGIS 2.5 Bundle for PostgreSQL 9.6 (32 bit) v2.3.3". Σε επόμενη οθόνη, δεν χρειάζεται να τσεκάρετε το "Create spatial database".

5. Αφού δώσαμε το password του superuser postgres, επιλέξτε YES και OK μέχρι να ολοκληρωθεί η εγκατάσταση.
6. Εκτελέσαμε από το μενού "PostgreSQL 9.6" το "pgAdmin4" και συνδεθείτε ως χρήστης postgres στην PostgreSQL. Δημιουργήσαμε μια νέα βάση (δεξί κλικ στο Databases) με όνομα LondonUK. Επιλέξαμε την LondonUK και από το μενού Tools εκτελέστε το Query Tool.
7. Εκτελέσαμε την sql εντολή "create extension postgis".
8. Εκτελέσαμε από το μενού "PostGIS Bundle 2.5 for PostgreSQL x32 9.6" το "PostGIS 2.0 Shapefile and DBF Loader". Δώσαμε τα στοιχεία της σύνδεσης (Username: postgres, Password: 123, Database: LondonUK) και φορτώσαμε τα shapefiles.
9. Μεταβήκαμε ξανά πίσω στο pgAdmin4 και αφού κάναμε Refresh (δεξί κλικ πάνω στο Databases) παρατηρήσαμε ότι τα shapefiles είναι πίνακες στο σχήμα public της βάσης LondonUK..



Εικόνα 10: Αποτέλεσμα των παραπάνω Βημάτων

5.3.4 Δεδομένα .shp

Όπως αναφέραμε σε προηγούμενο κεφάλαιο τα shapefile είναι στοιχεία αναφοράς της πόλης του Λονδίνου στη Μεγάλη Βρετανία εγγραφών που αποτελείται από σημεία: stations, γραμμές: roads, railways, πολύγωνα: borough, buildings, wards που αντιπροσωπεύονται από τη γεωμετρία x-y, αλλά και πολλά μη χωρικά δεδομένα όπως ο πληθυσμός.

Πιο αναλυτικά:

1. borough, περιέχει 23 εγγραφές

gid	Μοναδικό id για κάθε borough
name	Όνομα κάθε borough
gss_code	Εννέα χαρακτήρων Κωδικός στατιστικής υπηρεσίας του Ην. Βασιλείου
hectares	Εκτάρια
nonld_area	----------
ons_inner	----------
year	Χρονολογία καταμέτρησης πληθυσμού (2019)
population	Πληθυσμός
popperhect	Πληθυσμός ανά Εκτάριο
squre_kil	Έκταση ανά τετραγωνικό Χιλ.
poppersquk	Πληθυσμός ανά τετραγωνικό Χιλ.
geom	Όριο πολυγώνου για κάθε borough

2. buildings, περιέχει 490033 εγγραφές

gid	Μοναδικό id για κάθε building
osm_id	Open Street Map id
code	Κωδικός Open Street Map
fclass	building
name	Όνομα του κάθε κτηρίου

type	Όνομα του κάθε κτηρίου
geom	Όριο πολυγώνου για κάθε κτήριο

3. railways, περιέχει 12768 εγγραφές

gid	Μοναδικό id για κάθε railway
osm_id	Open Street Map id
code	Κωδικός Open Street Map
fclass	Κατηγορία γραμμής (rail, subway, light_rail)
name	Όνομα κάθε γραμμής
layer	Επίπεδο (-1,0,1)
bringe	Αν είναι γέφυρα (T,F)
tunnel	Αν είναι τούνελ (T,F)
geom	Γεωμετρία γραμμών για κάθε railway

4. roads, περιέχει 291215 εγγραφές

gid	Μοναδικό id για κάθε road
osm_id	Open Street Map id
code	Κωδικός Open Street Map
fclass	Κατηγορία Δρόμου (footway, motorway κ.α)
name	Όνομα κάθε δρόμου
ref	Τύπος κάθε Δρόμου με αρθμό
oneway	Αν είναι μονόδρομος
maxspeed	Μέγιστη Ταχύτητα
layer	Επίπεδο (-1,0,1)
bringe	Αν είναι γέφυρα (T,F)
tunnel	Αν είναι τούνελ (T,F)

geom Γεωμετρία γραμμών για κάθε Δρόμο

5. stations

gid Μοναδικό id για κάθε station

name Όνομα κάθε στάσης

geom Γεωμετρία σημείου για κάθε στάση

6. wards, περιέχει 983 εγγραφές

gid Μοναδικό id για κάθε ward

msoa11cd Κωδικός περιοχής κάθε ward

msoa11nm Ονομασία κάθε ward

lad11cd Εννέα χαρακτήρων Κωδικός στατιστικής υπηρεσίας του Ην. Βασιλείου

lad11nm Όνομα κάθε ward

rgn11cd Εννέα χαρακτήρων Κωδικός στατιστικής υπηρεσίας του Ην. Βασιλείου

rgn11nm London

usualres Τακτική ταξινόμηση κατοικίας

hholdres ----------

comestres ----------

popden Πυκνότητα πληθυσμού

hhollds ----------

avhholdssz ----------

geom Όριο πολυγώνου για κάθε ward

5.3.5 Εισαγωγή των Πολυδιάστατων Αρχείων στην PostgreSQL

Η εισαγωγή του αρχείου letter-recognition γίνεται με σχεδόν παρόμοιο τρόπο μόνο που εδώ δεν θα χρησιμοποιήσουμε το spatial extension. Οι διαδικασίες είναι ίδιες μέχρι το βήμα 5 του υποκεφαλαίου 5.3.3.

Στο βήμα 6:

6. Εκτελέσαμε από το μενού "PostgreSQL 9.6" το "pgAdmin4" και συνδεθείτε ως χρήστης postgres στην PostgreSQL. Δημιουργήσαμε μια νέα βάση (δεξί κλικ στο Databases) με όνομα letter-recognition. Επιλέξαμε την letter-recognition και από το μενού Tools εκτελέστε το Query Tool.
7. Σε αυτό το βήμα θα πρέπει να δημιουργήσουμε ένα table με όνομα letters και τις παρακάτω στήλες.

CREATE TABLE letters

(

letter character varying(2),

xbox int,

ybox int,

width int,

height int,

onpix int,

xbar int,

ybar int,

x2bar int,

y2bar int,

xybar int,

x2ybar int,

xy2bar int,

xedge int,

xedgey int,

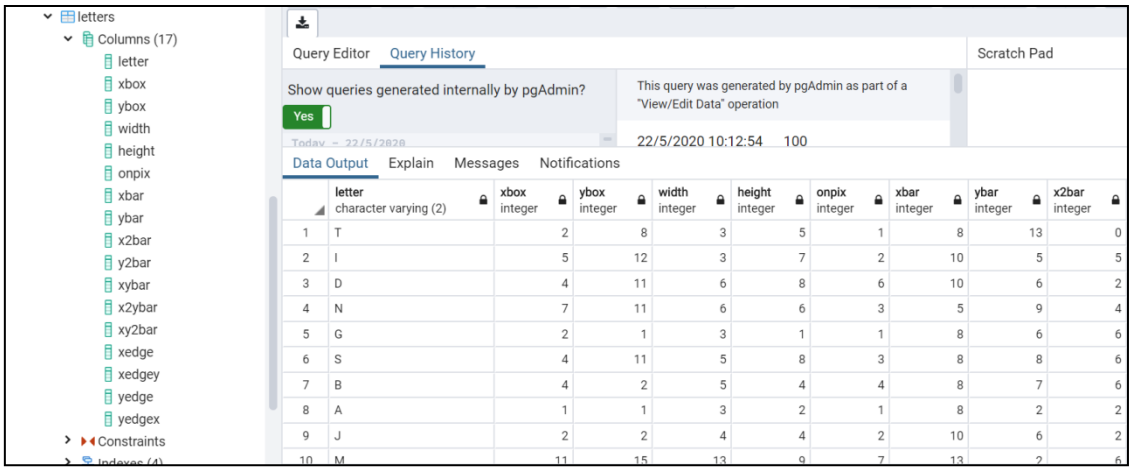
yedge int,

yedgex int

);

8. Στην συνεχεια θα πρέπει να κάνουμε εισαγωγή το αρχείο .data που κατεβάσαμε από το https://archive.ics.uci.edu/, USI, Machine Learning Repository, Letter Recognition Data Set.Εκτελούμε την παρακάτω εντολή στο Query Tool:

COPY letters FROM 'C:*\letter.data' DELIMITER ',';**



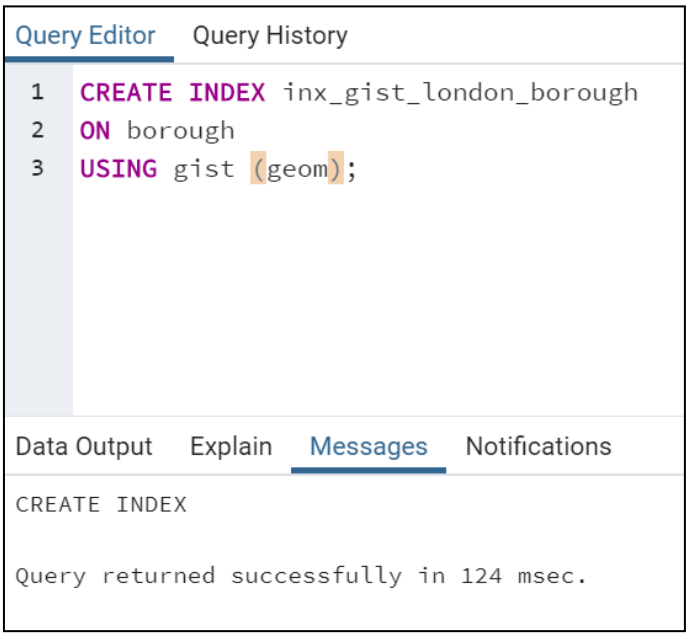
Εικόνα 11: Πίνακας εγγραφών του αρχείου Letter Recognition στην PostgreSQL

5.3.6 Δημιουργία Ευρετηρίων σε χωρικά δεδομένα σε PostgreSQL

Όπως αναφέραμε και σε προηγούμενα κεφάλαια θα πρέπει να εισάγουμε τα ευρετήρια σε κάθε πίνακα δεδομένων αφού έχουμε τρέξει τα ερωτήματα χωρίς αυτά. Η PostgreSQL όταν εισάγουμε τα .shp αρχεία δημιουργεί σε κάθε table από μόνη της ευρετήριο που παίρνει την ονομασία για το table borough π.χ borough_geom_ind. Για να τρέξουμε ερωτήματα χωρίς ευρετήρια θα πρέπει να γίνουνε DROP. Στη συνέχεια αφού τρέξουμε ερωτήματα στη βάση μας χωρίς ευρετήρια θα εισάγουμε αρχικά το GiST και έπειτα το BRIN.

Πρώτα θα εισάγουμε το ευρετήριο GiST. Η σύνταξη PostGIS για τη δημιουργία του ευρετηρίου είναι:

CREATE INDEX (“index_name”)
ON (“table_name”)
USING gist (“geometry_column”);



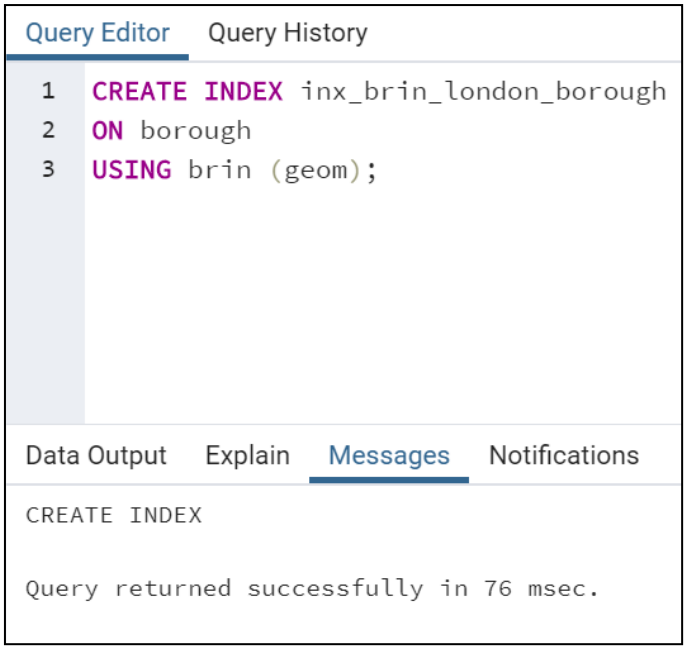
Εικόνα 12: Δημιουργία του ευρετηρίου GiST στο πίνακα borough και γεωμετρική στήλη geom.

Με τον παραπάνω τρόπο δημιουργούμε ένα ευρετήριο GiST στη στήλη γεωμετρίας (geom) και στους υπόλοιπους πίνακες.

Επιπλέον, αφού πραγματοποιήσαμε τα πειράματα εκτελώντας διάφορα χωρικά ερωτήματα στους πίνακες με ευρετήριο τη δομή GiST, καταγράψαμε τον χρόνο εκτέλεσης και διαγράφουμε τα ευρετήρια. Τα χωρικά ερωτήματα που εκτελέσαμε εξηγούνται λεπτομερώς σε επόμενο κεφάλαιο. Στη συνέχεια ευρετηριάσαμε τις γεωμετρίες με τη δομή BRIN και εκτελέσαμε το ίδιο σύνολο διαφορετικών χωρικών ερωτημάτων και καταγράψαμε τον χρόνο εκτέλεσης.

Η σύνταξη για τη δημιουργία του ευρετηρίου BRIN στις στήλες γεωμετρίας είναι(δείτε Εικόνα 13):

CREATE INDEX (“index_name”)
ON (“table_name”)
USING brin (“geometry_column”);



Εικόνα 13: Δημιουργία του ευρετηρίου BRIN στο πίνακα borough και γεωμετρική στήλη geom.

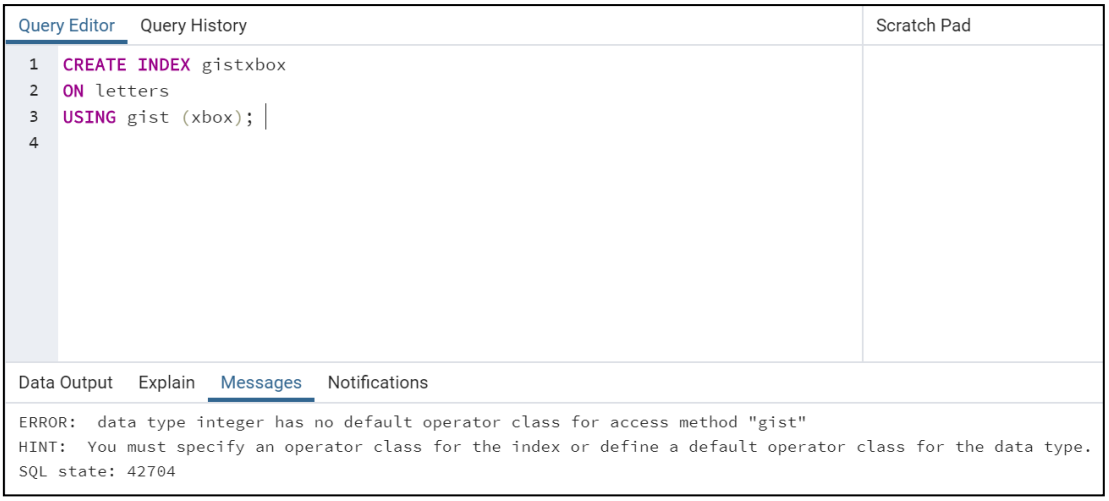
5.3.7 Δημιουργία Ευρετηρίων σε πολυδιάστατα δεδομένα σε PostgreSQL

Η διαδικασία ευρετηρίασης σε πολυδιάστατα δεδομένα διαφέρει κατά πολύ από τα χωρικά δεδομένα. Στα χωρικά δεδομένα όπως είδαμε δημιουργούνται ευρετήρια σε μία στήλη που ονομάζεται geom, ενώ στα πολυδιάστατα τοποθετούμε ευρετήρια σε όσες στήλες επιθυμούμε, εκτός από την στήλη που περιγράφει τα δεδομένα. Τα GiST και BRIN είναι ευρετήρια πολλαπλών στηλών. Μπορούν να καθοριστούν έως και 32 στήλες.

Πρώτα θα εισάγουμε το ευρετήριο GiST:

Όταν εκτελέσουμε την παρακάτω εντολή στην αρχή η PostgreSQL θα μας βγάλει error. Όπως φαίνεται στην παρακάτω εικόνα.

CREATE INDEX (“index_name”)
ON (“table_name”)
USING gist (“column”);



Εικόνα 14: Error Gist Index

Μπορούμε να χρησιμοποιήσουμε την επέκταση btree_gist για να ορίσουμε περιορισμούς αποκλεισμού σε τύπους απλών κλιματικών δεδομένων, οι οποίοι στη συνέχεια μπορούν να συνδυαστούν με εξαιρέσεις εύρους για μέγιστη ευελιξία.

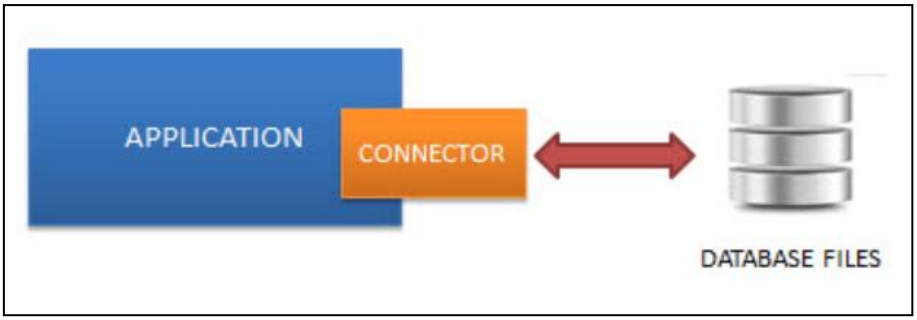
CREATE EXTENSION btree_gist;

Με το ίδιο τρόπο θα εισάγουμε και το BRIN.

5.4 SQLite

Το SQLite είναι δωρεάν λογισμικό ανοικτού κώδικα και δεν απαιτείται ειδική άδεια χρήσης του. Επίσης, είναι γνωστό για τη φορητότητα, την αξιοπιστία και την ισχυρή απόδοση του. Το lite στη ονομασία, σημαίνει ότι είναι ένα λογισμικό εύκολο στην εγκατάσταση και τη διαχείριση της βάσης δεδομένων. Ο συγγραφέας της, D. Richard Hipp, εργάστηκε για το General Dynamics στο U.S.Navy και την άνοιξη του 2000 σχεδίαζε την SQLite - λογισμικό για ενσωματωμένη χρήση σε καθοδηγούμενους καταστροφείς πυραύλων.

Η επίσημη ιστοσελίδα της SQLite το περιγράφει ως βάση δεδομένων χωρίς διακομιστές(δείτε Εικόνα 15). Αυτό την κάνει να διαφέρει από την PostgreSQL που αναφέραμε προηγουμένως. Η SQLite το μόνο που χρειάζονται είναι η πρόσβαση στο δίσκο[4].



Εικόνα 15: SQLite server-less architecture, https://www.sqlitetutorial.net/what-is-sqlite/, SQLite Tutorial

5.4.1 Χρήστες της SQLite

Η SQLite χρησιμοποιείται από εκατομμύρια εφαρμογές και θεωρείται ο πιο ευρέως αναπτυγμένος μηχανισμός βάσεων δεδομένων στον κόσμο σήμερα. Μερικοί από τους πιο γνωστούς χρήστες του SQLite είναι:

1. Η Apple χρησιμοποιεί τη SQLite σε πολλές εφαρμογες που εκτελούνται σε σταθερούς υπολογιστές και διακομιστές Mac OS-X και σε συσκευές iOS, όπως iPhones και iPods. SQLite χρησιμοποιείται επίσης και στο iTunes.
2. Η Google χρησιμοποιεί το SQLite στο δικό της λειτουργικό σύστημα κινητού τηλεφώνου Android και στο Chrome Web Browser.
3. Όλες οι διανομές της Python από την Python 2.5 περιλαμβάνουν SQLite.
4. Η Microsoft χρησιμοποιεί το SQLite ως βασικό στοιχείο των Windows 10
5. Η δημοφιλής γλώσσα προγραμματισμού PHP έρχεται με built-in τόσο SQLite2 όσο και SQLite3
6. Η Adobe χρησιμοποιεί το SQLite ως μορφή αρχείου εφαρμογών για το προϊόν Photoshop Lightroom. Έχει αναφερθεί ότι το Acrobat Reader χρησιμοποιεί επίσης SQLite[27].

5.4.2 SQLite και SpatiaLite

Για την έρευνα μας, είναι σημαντική η χρήση της SpatialLite, όπου αποτελεί μια μηχανή βάσης δεδομένων SQLite με πρόσθετες χωρικές λειτουργίες. Μπορεί κανείς να σκεφτεί το SpatialLite ως πρόσθετη χωρική τεχνολογία για το SQLite παρόμοιο με αυτό που κάνει το PostGIS για την PostgreSQL. Στην πραγματικότητα, η λειτουργικότητα της SpatialLite είναι λίγο πολύ οι ίδιες με αυτές του PostGIS[24].

Λογισμικά που υποστηρίζουν το SpatialLite:

Desktop:

- ESRI's ArcGIS
- QGIS
- Autocad MAP
- Global Mapper

(Web)Server:

- GeoServer via SpatiaLite extension
- GeoDjango via the GeoDjango module
- Web2py (web framework) native

Tools and libraries:

- OGR Simple Feature Library reads and writes SpatiaLite since version 1.7
- GeoTools supports SpatiaLite using JDBC module
- Mapnik, a renderer[34]

5.4.3 Εισαγωγή των .shp Αρχείων στην SQLite

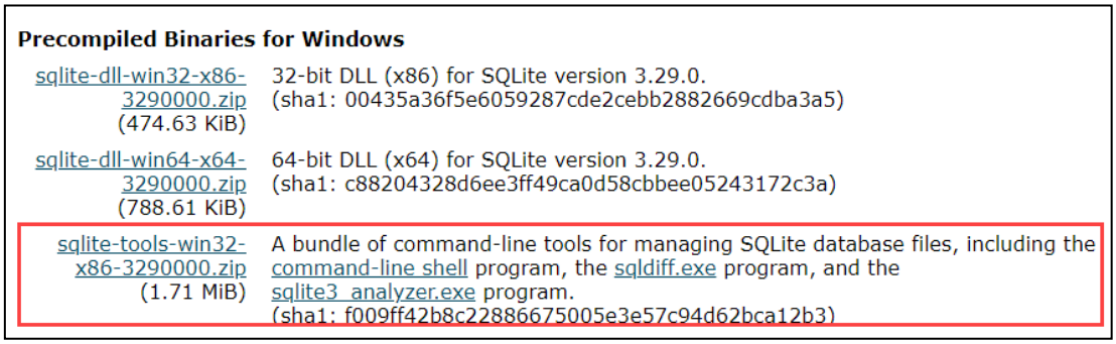
Αρχικά θα πρέπει να εγκαταστήσουμε το λογισμικό της SQLite στο υπολογιστή.

Για συστήματα Windows:

1. Επισκεπτόμαστε την επίσημη σελίδα της SQLite, https://www.sqlite.org.

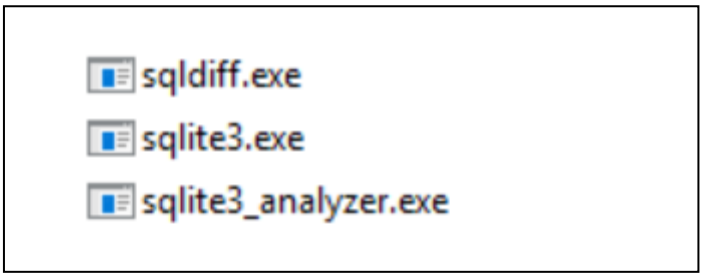
2. Υπάρχουν δύο τρόποι για να εργαστούμε με την SQLite. Ο πρώτος είναι με το sqlite3 shell και ο δεύτερος με το SQLite GUI tool
3. SQLite tools shell

   i. Ανοίγουμε την σελίδα με τα download links https://www.sqlite.org/download.html. Το SQLite παρέχει διάφορα εργαλεία για εργασία σε πλατφόρμες, π.χ. Windows, Linux και Mac. Εμείς θα επιλέξουμε Windows.



Εικόνα 16:Download links για Windows

   ii. Η εγκατάσταση του SQLite είναι απλή.
   iii. Δημιουργούμε έναν νέο φάκελο, π.χ. C: \ sqlite.
   iv. Εξάγουμε το περιεχόμενο του αρχείου που κατεβάσατε στην προηγούμενη ενότητα στο φάκελο C: \ sqlite. Θα πρέπει να δείτε τρία προγράμματα στο φάκελο C: \ sqlite όπως φαίνεται παρακάτω:



Εικόνα 17: Αρχεία φακέλου της sqlite

   v. Ανοίγουμε το παράθυρο της γραμμής εντολών και μεταβαίνουμε στο φάκελο της sqlite3

C: \ sqlite.

C: \ cd c: \ sqlite

C: \ sqlite>

C:\sqlite>sqlite3

4. Εγκαθιστούμε το εργαλείο SQLite GUI

Το sqlite3 shell είναι εξαιρετικό εργαλείο, αλλά μερικές φορές πιο δύσκολο στην διαχείριση βάσεων δεδομένων από τον χρήστη. Για αυτό το λόγο υπάρχουν και τα εργαλεία GUI. Υπάρχουν πολλά εργαλεία GUI για τη διαχείριση βάσεων δεδομένων SQLite, που κυμαίνονται από δωρεάν λογισμικό έως εμπορικές άδειες.

- SQLiteStudio

Το εργαλείο SQLiteStudio είναι ένα δωρεάν εργαλείο GUI για τη διαχείριση βάσεων δεδομένων SQLite. Είναι δωρεάν, διαισθητικό και πολλαπλών πλατφορμών. Το εργαλείο SQLite παρέχει επίσης μερικές από τις πιο σημαντικές δυνατότητες για εργασία με βάσεις δεδομένων SQLite, όπως εισαγωγή, εξαγωγή δεδομένων σε διάφορες μορφές, συμπεριλαμβανομένων των CSV, XML και JSON.

Μπορείτε να κατεβάσετε την έκδοση του SQLiteStudio και να την εγκαταστήσετε μεταβαίνοντας στη σελίδα λήψης https://sqlitestudio.pl/. Στη συνέχεια, μπορείτε να εξαγάγετε (ή να εγκαταστήσετε) το αρχείο λήψης σε ένα φάκελο, π.χ., C: \ sqlite \ gui \ και να το εκκινήσετε.

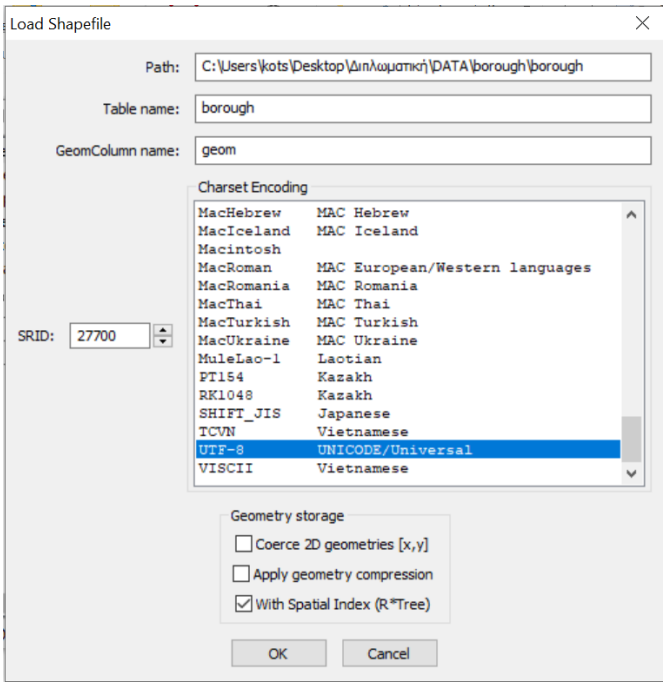
- DB Browser

Το πρόγραμμα περιήγησης DB Browser για SQLite είναι ένα υψηλής ποιότητας, οπτικό εργαλείο ανοιχτού κώδικα για τη δημιουργία, το σχεδιασμό και την επεξεργασία αρχείων βάσης δεδομένων συμβατών με το SQLite.

5. Για τα χωρικά δεδομένα μπορούμε να εγκαταστήσουμε το εργαλείο **spatialite-gui.exe** https://www.gaia-gis.it/fossil/spatialite_gui/index

Το spatialite-gui είναι ένα γραφικό περιβάλλον εργασίας χρήστη (GUI) για το SpatiaLite και φιλικό προς το χρήστη. Το Spatialite περιλαμβάνει επίσης το Virtualshape και το Virtualtext για να επιτρέπεται η πρόσβαση σε αρχεία shapefiles και csv/text ως εικονικοί πίνακες.

6. Αφού έχουμε εγκαταστήσει το spatialite-gui, το ανοίγουμε και δημιουργούμε μία βάση δεδομένων με όνομα LondonUK, όπως και στην PostgreSQL.
7. Φορτώνουμε τα .shp αρχεία όπως φαίνεται στην παρακάτω εικόνα. Επιλέγουμε SRID: 27700 και Charset Encoding UTF-8. Την διαδικασία την επαναλαμβάνουμε για όλα τα .shp αρχεία.

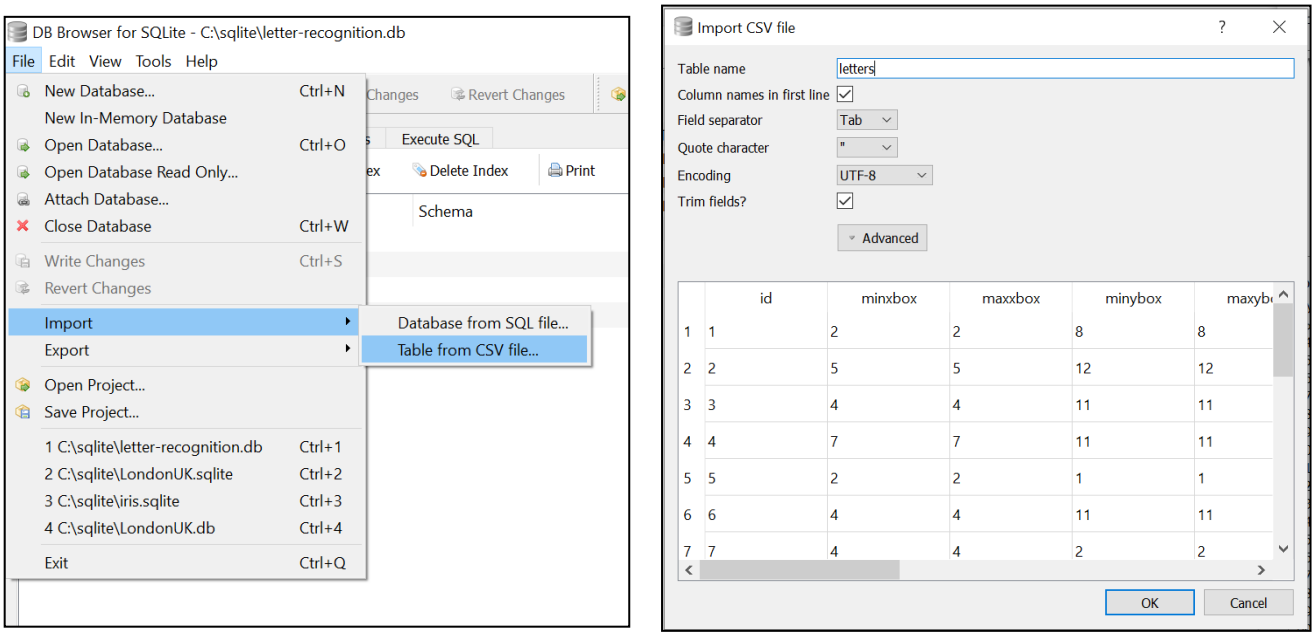


Εικόνα 18:Εισαγωγή .shp αρχείων στο spatialite-gui.

5.4.4 Εισαγωγή των Πολυδιάστατων Αρχείων στην SQLite

Η εισαγωγή του αρχείου letter-recognition θα γίνει με την μορφή .csv αρχείου στο DB Browser.

1. Δημιουργούμε μία βάση δεδομένων με όνομα letter-recognition.
2. Με την εντολή import θα γίνει η παραπάνω διαδικασία



Εικόνα 19:Εισαγωγή αρχείου letters στην SQLite

5.4.5 Δημιουργία Ευρετηρίων σε χωρικά δεδομένα σε SQLite

Το SQLite υποστηρίζει το ευρετήριο R*Tree. Οι λειτουργίες και το πώς εφαρμόζεται διαφέρουν από τη προηγούμενη DBMS που αναλύσαμε.

Κάθε R*Tree στη SQLite απαιτεί τέσσερις αυστηρά συσχετισμένους πίνακες:

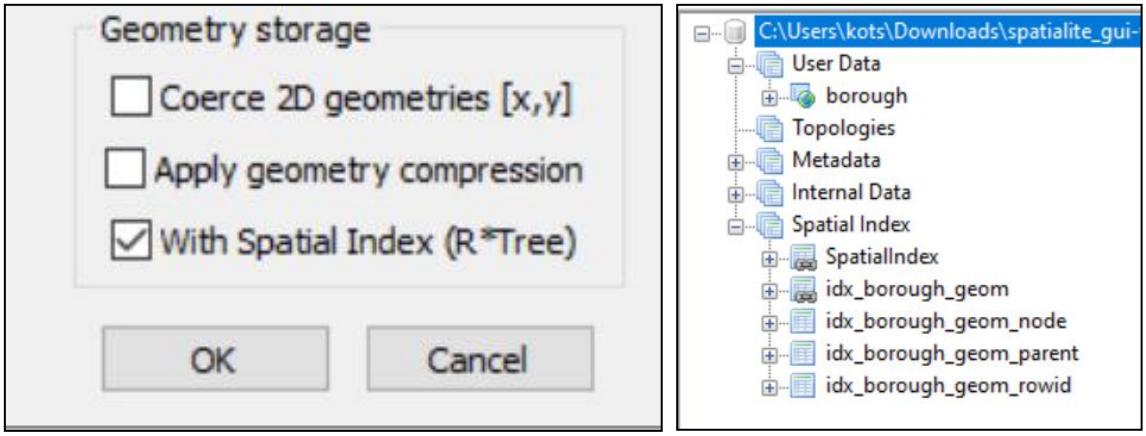
- rtreebasename_node αποθηκεύει (δυαδική μορφή) τους στοιχειώδεις κόμβους R*-Tree.
- rtreebasename_parent αποθηκεύει σχέσεις που συνδέουν parent and child κόμβους.
- rtreebasename_rowid αποθηκεύει τιμές ROWID που συνδέουν έναν κόμβο R*-Tree και μια αντίστοιχη σειρά στον πίνακα ευρετηρίου.

Κανένας από αυτούς τους τρεις πίνακες δεν προορίζεται για άμεση πρόσβαση: προορίζονται για εσωτερική διαχείριση.

- Το rtreebasename είναι στην πραγματικότητα ένας εικονικός πίνακας και δίνει στο χρήστη τη δυνατότητα για εξωτερική διαχείριση του ευρετηρίου.

Η δημιουργία ευρετηρίων σε χωρικά δεδομένα γίνεται μέσο του spatialite-gui. Όταν εισάγουμε ένα .shp αρχείο μας δίνεται η δυνατότητα να δημιουργήσουμε το ευρετήριο με την επιλογή (with Spatial Index R*Tree).

Το ίδιο το επαναλαμβάνουμε και για τους υπόλοιπους πίνακες.



Εικόνα 20: Δημιουργία Ευρετηρίων σε χωρικά δεδομένα σε SQLite

Η geometry_column στη βάση δεδομένων αποθηκεύει τις χωρικές πληροφορίες για όλους τους πίνακες της βάσης. Αν ρίξουμε μια πιο προσεκτική ματιά στη στήλη γεωμετρίας, θα

SELECT * FROM geometry_columns;

Πινάκας 3: Χωρικές πληροφορίες για όλους τους πίνακες της βάσης

f_table_name	f_geometry_column	type	coord_dimension	srid	spatial_index_enabled
borough	geom	MULTIPOLYGON	XY	27700	1
buildings	geom	MULTIPOLYGON	XY	27700	1
railways	geom	LINESTRING	XY	27700	1
roads	geom	LINESTRING	XY	27700	1
stations	geom	POINT	XYZM	27700	1
wards	geom	MULTIPOLYGON	XY	27700	1

πρέπει να παρατηρήσουμε την στήλη Spatial_index_enabled, όπου επιτρέπει στο χρήστη να γνωρίζει εάν το ευρετήριο είναι ενεργοποιημένο (1) ή όχι (0).

5.4.6 Δημιουργία Ευρετηρίων σε πολυδιάστατα δεδομένα σε SQLite

Το ευρετήριο R*-Tree στην SQLite υλοποιείται ως εικονικός πίνακας. Κάθε ευρετήριο R*-Tree είναι ένας εικονικός πίνακας με αριθμό στηλών μεταξύ 3 και 11. Η πρώτη στήλη αποτελεί το κλειδί και είναι πάντα ένα ακέραιος αριθμός 64-bit. Οι άλλες στήλες είναι ζεύγη, ένα ζεύγος αποτελεί μία διάσταση, που περιέχει τις ελάχιστες και μέγιστες τιμές για αυτήν. Ένα μονοδιάστατο R*-Tree έχει 3 στήλες. Ένα δισδιάστατο R*-Tree έχει 5 στήλες. Ένα τρισδιάστατο R*-Tree έχει 7 στήλες. Ένα τετραδιάστατο R*-Tree έχει 9 στήλες. Και ένα 5-διαστατικό R*-Tree έχει 11 στήλες. Η εφαρμογή SQLite R*-Tree δεν υποστηρίζει R*-Tree δέντρα μεγαλύτερα από 5 διαστάσεις.

Οι στήλες ζεύγους ελάχιστης/μέγιστης τιμής αποθηκεύεται ως τιμές κινητής υποδιαστολής 32-bit για «rtree» εικονικούς πίνακες ή ως 32-bit ακέραιοι σε «rtree_i32» εικονικούς πίνακες. Σε αντίθεση με τους κανονικούς πίνακες SQLite που μπορούν να αποθηκεύσουν δεδομένα σε μια ποικιλία τύπων δεδομένων και μορφών, το R*-Tree επιβάλλει αυστηρά αυτούς τους τύπους αποθήκευσης. Εάν εισαχθεί οποιοσδήποτε άλλος τύπος τιμής σε μια τέτοια στήλη, το r-tree module το μετατρέπει αυτόματα στον απαιτούμενο τύπο πριν εισάγθεί νέα εγγραφή στη βάση δεδομένων.

Ένα ευρετήριο R*-Tree δημιουργείται με την παρακάτω εντολή:

CREATE VIRTUAL TABLE <name> USING rtree(<column-names>);

Στην ανάλυση μας, θα χρησιμοποιήσουμε τις πρώτες 5 στήλες του αρχείου letter-recognition. Κάθε στήλη θα αποτελεί κα μία διάσταση. Αυτό σημαίνει ότι θα έχουμε συνολικά 11 στήλες.

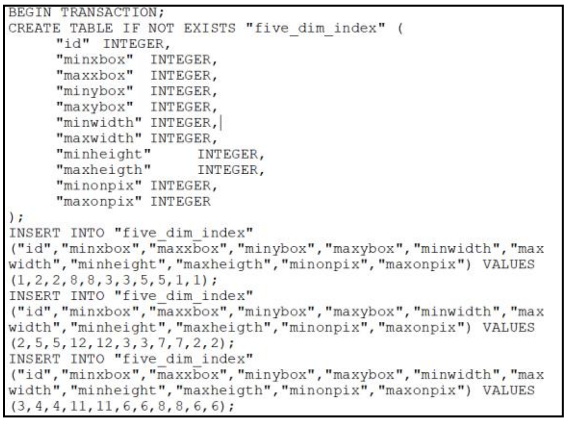
```
CREATE VIRTUAL TABLE five_dim_index USING rtree(
  id,                    -- Integer primary key
  minxbox, maxxbox,      -- Minimum and maximum 1 coordinate
  minybox, maxybox,      -- Minimum and maximum 2 coordinate
  minwidth, maxwidth,    -- Minimum and maximum 3 coordinate
  minheight, maxheigth,  -- Minimum and maximum 4 coordinate
```

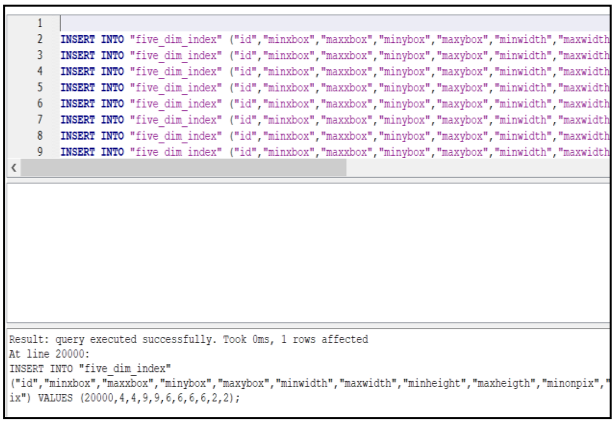
minonpix, maxonpix   -- Minimum and maximum 5 coordinate

);

Στη συνέχεια θα πρέπει αν εισάγουμε τις 20.000 εγγραφές του .csv αρχείου μέσα στο 'Κενό' Πίνακα.

Η διαδικασία που ακολούθησαμε ήταν ίδια με το υπο-κεφάλαιο 5.4.4 μόνο που ονομάσαμε τον πίνακα five_dim_index και όχι letters και επίσης τροποποιήσαμε το .csv αρχείο εισάγοντας μία νέα στήλη id. Έπειτα κάναμε export Database to SQL file και copy-paste τις εντολές INSERT στο ***VIRTUAL TABLE five_dim_index USING rtree.***





Εικόνα 21: Δημιουργία Ευρετηρίων σε πολυδιάστατα δεδομένα σε SQLite

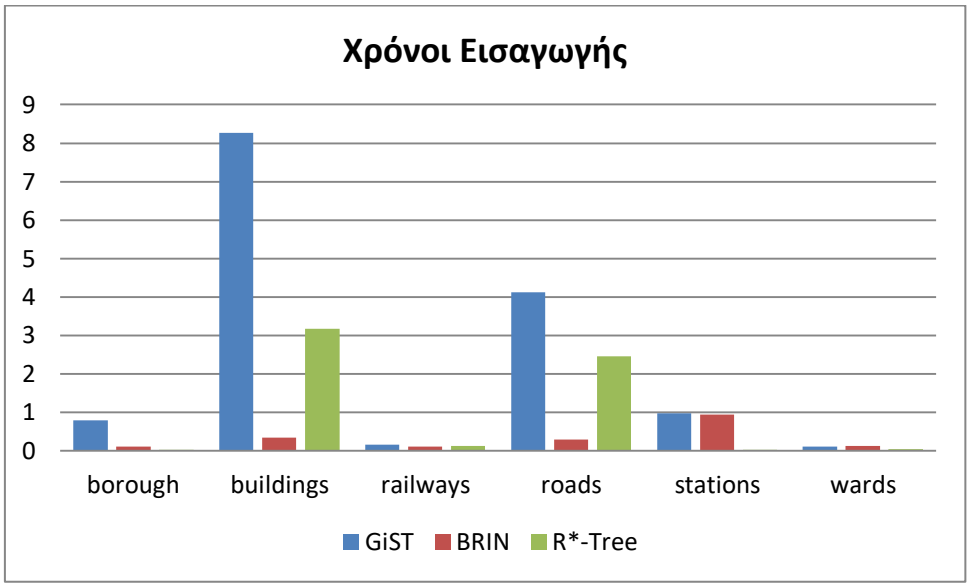
5.5 Χρόνοι εκτέλεσης ευρετηρίων στα χωρικά δεδομένα

Η ενεργοποίηση του ευρετηρίου στις στήλες γεωμετρίας απαιτεί χρόνο ανάλογα με τον τύπο του ευρετηρίου και το μέγεθος των εγγραφών στον κάθε πίνακα. Ορισμένα ευρετήρια διαρκούν λίγο χρόνο, ενώ άλλα χρειάζονται περισσότερο από το διπλάσιο των δεδομένων χρόνων.

Αυτό που πρακτικά κάναμε ήταν να καταγράψουμε το χρόνο που απαιτείται για την εισαγωγή ευρετηρίων, δηλαδή, GiST, BRIN και R*-Tree για εφαρμογή στη γεωμετρία του συνόλου δεδομένων αναφοράς. Τα shapefiles που χρησιμοποιήσαμε διατηρήθηκαν τα ίδια και για τις τρεις δομές ευρετηρίου, επομένως η σύγκριση δίνει μια σαφή εικόνα στον αναγνώστη.

Πίνακας 4: Καταγραφή χρόνων για την εισαγωγή του κάθε index

Ευρετήρια/.shp	borough	buildings	railways	roads	stations	wards	*Average*
GiST	0.79	8.26	0.154	4.119	0.97	0.104	**2,3995**
BRIN	0.102	0.332	0.103	0.290	0.94	0.119	**0,314333**
R*-Tree	0.006	3.671	0.121	2.453	0.004	0.043	**1,049667**



Εικόνα 22: Γράφημα χρόνων για την εισαγωγή του κάθε index σε (sec).

❖ Για τα πολυδιάστατα δεδομένα δεν έχει κάποιο νόημα αυτή η διαδικασία.

Είναι ολοφάνερο ότι απαιτείται περισσότερος χρόνος για την εισαγωγή του ευρετηρίου GiST.

5.6. Χρόνοι εκτέλεσης ερωτημάτων στις (DBMS) - Explain

Ο Χρόνος εκτέλεσης των ερωτημάτων αποτελεί ένα σημαντικό κεφάλαιο για την ανάλυση μας. Υπάρχουνε αναφορές, οι οποίες υποστηρίζουν ότι όταν τρέχουμε ένα ερώτημα, η δεύτερη εκτέλεση του ίδιου ερωτήματος είναι συχνά πιο γρήγορη. Αυτό βασίζεται στην μνήμη cache και το φαινόμενο ονομάζεται "results cache".

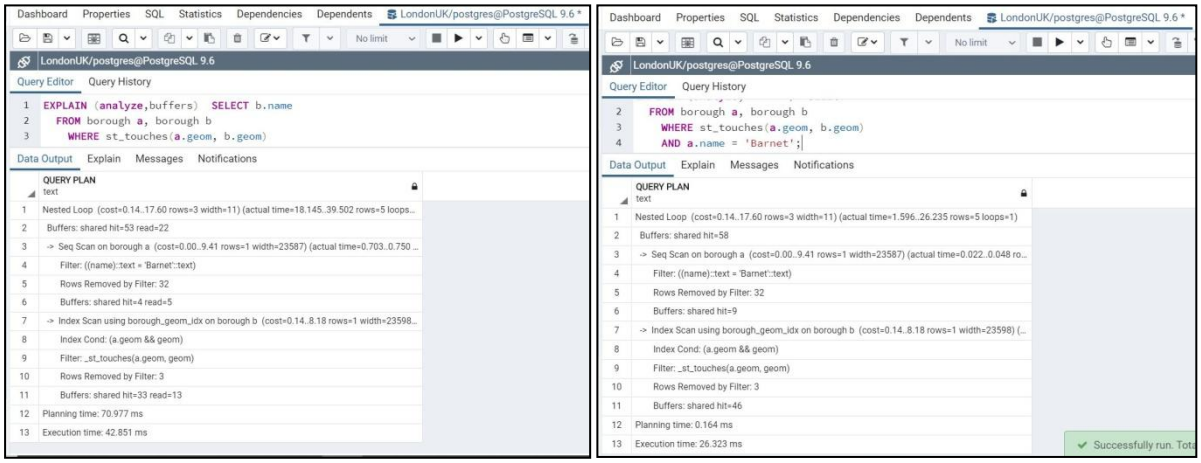
Η Εντολή explain θα μας δώσει την απάντηση σε αυτό το φαινόμενο και αν ισχύει για τα ερωτήματα μας. Μπορεί ακόμη και να μας δώσει πόσα data blocks δεδομένων προήλθαν από το δίσκο και πόσα προήλθαν από shared_buffers, δηλαδή τη μνήμη. Επίσης, καταγράφεται το

execution time, το οποίο θα χρησιμοποιήσουμε στα αποτελέσματα μας. Τέλος, είναι πολύ σημαντική εντολή, γιατί μας δείχνει ποτέ ένα ερώτημα χρησιμοποιεί τα ευρετήρια μας.

Shared read, σημαίνει ότι προέρχεται από το δίσκο και δεν αποθηκεύτηκε στην μνήμη. Εάν το ερώτημα εκτελεστεί ξανά και εάν όντως χρησιμοποιεί τη μνήμη, θα εμφανιστεί ως shared hit. Τα ερωτήματα μας, χρησιμοποιούν την μνήμη, οπότε επηρεάζεται ο χρόνος εκτέλεσης. Υπάρχουνε δύο λύσεις για αυτό:

1. Πριν εκτελέσουμε ένα ερώτημα να 'Καθαρίζουμε' την μνήμη ή να κάνουμε reboot τον Υπολογιστή μας.
2. Να τρέχουμε τρεις ή περισσότερες φορές το κάθε ερώτημα και να παίρνουμε τον μέσο όρο του χρόνου.

Εμείς ακολουθήσαμε το 1 βήμα. Όντως τα ερωτήματα μας χρησιμοποιούν την μνήμη στη $2^{η}$ Εκτέλεση και απαιτείται σχεδόν ο μισός χρόνος από 42.851ms σε 26.323ms.



Εικόνα 23: 1η Εκτέλεση (Αριστερή Εικόνα) με Shared read, 2η Εκτέλεση (Δεξιά Εικόνα) με Shared hit[33]

ΚΕΦΑΛΑΙΟ 6 – ΕΡΩΤΗΜΑΤΑ (QUERIES)

6.1 Γλώσσα χωρικών ερωτημάτων(Spatial Query Language)

Η μακροχρόνια έρευνα που βασίζεται στο SDBMS είναι να έχουμε μια βαθύτερη ανάλυση του χώρου και των αντικειμένων του οποίου κατασκευάζεται. Τα χωρικά δεδομένα όπως αναφέραμε σε προηγούμενο κεφάλαιο είναι ένας όρος που χρησιμοποιείται για την περιγραφή δεδομένων που σχετίζονται με το χώρο που καταλαμβάνουν τα αντικείμενα σε μια βάση δεδομένων. Είναι τα γεωμετρικά δεδομένα όπως σημεία, γραμμές, πολύγωνα.

Η Spatial Query Language είναι μια γλώσσα βάσης δεδομένων που έχει αναπτυχθεί για την αναζήτηση χωρικών χαρακτηριστικών χρησιμοποιώντας την παραδοσιακή δομημένη γλώσσα ερωτήσεων (SQL).

Το χωρικό ερώτημα είναι ένας ειδικός τύπος ερωτήματος βάσης δεδομένων που υποστηρίζεται από γεωγραφικές βάσεις δεδομένων και χωρικές βάσεις δεδομένων. Τα ερωτήματα διαφέρουν από τα μη χωρικά ερωτήματα SQL με διάφορους σημαντικούς τρόπους. Δύο από τα πιο σημαντικά είναι ότι επιτρέπουν τη χρήση τύπων δεδομένων γεωμετρίας όπως σημεία, γραμμές και πολύγωνα και ότι αυτά τα ερωτήματα λαμβάνουν υπόψη τη χωρική σχέση μεταξύ αυτών των γεωμετριών[1].

Οι διαφορετικές λειτουργίες χωρικού ερωτήματος μπορούν να ταξινομηθούν σε ακόλουθες μεγάλες ομάδες:

1. Λειτουργία ενημέρωσης (Update operation): Περιλαμβάνει τυπικές λειτουργίες βάσης δεδομένων όπως δημιουργία (create), τροποποίηση (modify) και ενημέρωση (update).
2. Υπάρχουν επιπλέον συναρτήσεις για τη διατύπωση – υπολογισμό χωρικών παραμέτρων.
    - Point query: Παρουσιάζει όλα τα ορθογώνια που περιέχουν το δεδομένο σημείο.
    - Range query: Παρουσιάζει όλα τα σημεία που βρίσκονται μέσα σε ένα ορθογώνιο ερωτήματος.
    - Nearest neighbor: Παρουσιάζει όλες τις γραμμές που τέμνουν ένα ορθογώνιο ερωτήματος.

- Distance scan: Γίνεται καταμέτρηση σημείων σε αυξανόμενη απόσταση από ένα σημείο ερωτήματος.
- Intersection query: Παρουσιάζει όλα τα ορθογώνια και πολύγωνα που τέμνουν ένα ορθογώνιο ερωτήματος.
- Containment query: Παρουσιάζει όλα τα ορθογώνια ή πολύγωνα που βρίσκονται μέσα σε ένα ορθογώνιο ερωτήματος[5].

6.1.1 Χωρική συσχέτιση ερωτημάτων

Μια χωρική σχέση πραγματοποιείται μεταξύ 2 ή περισσότερων χαρακτηριστικών και περιλαμβάνει κατεύθυνση(direction), απόσταση(distance) ή τοπολογία(topology).

1. κατεύθυνση(direction), Οι ερωτήσεις κατεύθυνσης ρωτούν σχετικά με τον προσανατολισμό των χαρακτηριστικών σε έναν χάρτη.
2. απόσταση(distance), Οι ερωτήσεις απόστασης ρωτούν σχετικά χαρακτηριστικά εντός μίας δεδομένης απόστασης από άλλα χαρακτηριστικά. Οι ερωτήσεις απόστασης μπορούν επίσης να θεωρηθούν το πόσο κοντά είναι το ένα αντικείμενο στο άλλο.
3. τοπολογία(topology), Οι ερωτήσεις τοπολογίας ρωτούν για το πώς τα γεωμετρικά χαρακτηριστικά, δηλαδή σημεία, γραμμές και πολύγωνα σχετίζονται χωρικά.

Υπάρχουν διάφορα ερωτήματα που πραγματοποιήσαμε στο πείραμά μας πάνω από το σύνολο δεδομένων αναφοράς χρησιμοποιώντας PostGIS και Spatialite. Και οι δύο βάσεις δεδομένων πραγματοποίησαν το ίδιο σύνολο ερωτημάτων, αλλά με μια μικρή διαφορά όσον αφορά τη χρήση των συναρτήσεων.

Υπάρχουν διάφορες χωρικές λειτουργίες που υποστηρίζει ένα SDBMS. Κάθε συνάρτηση εμπίπτει σε μια συγκεκριμένη κατηγορία που έχει έναν καθορισμένο ρόλο ενώ θέτουμε ερωτήματα σε μία βάση δεδομένων[3].

Πίνακας 5: Λίστα με τις λειτουργίες στο OGIS για την SQL. The Handbook of Geographic Information Science [3]

Basic Functions	SpatialReference()	Returns the underlying coordinate geometry
	Envelope()	Returns the minimum orthogonal bounding rectangle of the geometry
	Export()	Returns the geometry in a different representation
	IsEmpty()	Returns true if the geometry is an empty set
	IsSimple()	Returns true if the geometry is simple (no self-intersection
	Boundary()	Returns the boundary of the geometry
Topological/Set Operators	Equal	Returns true if the interior and the boundary of the two geometries are spatially equal
	Disjoint	Returns true if the boundaries and interior do not intersect
	Intersect	Returns true if the interiors of the geometries intersect
	Touch	Returns true if the boundaries intersect but the interiors do no
	Cross	Returns true if the interiors of the geometries intersect with a curve
	Within	Returns true if the interior of the given geometry does not intersect with the exterior of another geometry
	Contains	Tests if the given geometry contains another geometry
	Overlaps	Returns true if the interiors of two geometries have non-empty intersection
Spatial Analysis	Distance	Returns the shortest distance between two geometries
	Buffer	Returns a geometry that consists of all points whose distance from the given geometry is less than or equal to the distance
	ConvexHull	Returns the smallest convex set enclosing the geometry
	Intersection	Returns the geometric intersection of two geometries
	Union	Returns the geometric union of two geometries
	Difference	Returns the portion of a geometry which does not intersect with another given geometry
	SymmDiff	Returns the portion of two geometries which do not intersect with each other

6.2 Χωρικά Ερωτήματα σε PostgreSQL

Όπως ήδη συζητήθηκε στην ενότητα 6.1.1, προκειμένου να εκτελέσουμε χωρικά ερωτήματα στη βάση δεδομένων μας χρησιμοποιήσαμε πολλές χωρικές συναρτήσεις. Σε αυτήν την ενότητα θα εξετάσουμε τα διάφορα ερωτήματα και θα συζητήσουμε τη χωρική συνάρτηση που χρησιμοποιήσαμε για να τα εκτελέσουμε[2][30].

Αρχικά, θα εκτελέσουμε ερωτήματα χωρίς ευρετήρια και στη συνέχεια με ευρετήρια και θα συγκρίνουμε την απόδοση τους.

Simple SQL:

Q1: Find number of letters in all the roads names in residential.

SELECT char_length(name)

FROM roads

WHERE fclass = 'residential';

Q2: Find the percentage of population per Square kil. For each borough.

SELECT name

,100*Sum(population)/Sum(hectares) AS popPerSqurKil

FROM borough

GROUP BY name;

Λειτουργίες που χρησιμοποιούνται στην ***Simple SQL:***

F1: Average ()

Η συνάρτηση Average () στο PostGIS επιστρέφει τη μέση τιμή της αριθμητικής στήλης.

F2: Char_length ()

Η συνάρτηση char_length () στο PostGIS μετρά το μήκος των χαρακτήρων στη στήλη.

F3: sum ()

Η συνάρτηση sum () στο PostGIS επιστρέφει το άθροισμα των εγγραφών σε ένα σύνολο εγγραφών.

Geometry:

Q3: Compute the area of ward name ‘'Bexley' in acres. (The unit given to us in the data is in meters)

SELECT Sum(ST_Area(geom)) / 4047

FROM wards

WHERE lad11nm = 'Bexley';

Q4: What is the JSON representation of the ward of 'Bexley'?

SELECT ST_AsGeoJSON(geom)

FROM wards

WHERE lad11nm = 'Bexley';

Q5: Find the length of the subway 'Sutton and Mole Valley Line' in Kilometers.

SELECT sum (ST_Length(geom))/1000

FROM railways

WHERE name = 'Sutton and Mole Valley Line';

Λειτουργίες που χρησιμοποιούνται στην άσκηση ***Geometry:***

F4: ST_Area()

Η συνάρτηση ST_Area στο PostGIS επιστρέφει την επιφάνεια εάν είναι POLYGON ή MULTIPOLYGON.

F5: ST_AsGeoJSON ()

Η συνάρτηση ST_AsGeoJSON στο PostGIS επιστρέφει τη γεωμετρία ως στοιχείο GeoJSON.

F6: ST_GeometryN

Η συνάρτηση ST_GeometryN στο PostGIS επιστρέφει τη γεωμετρία Nth που βασίζεται στο 1, εάν η γεωμετρία είναι GEOMETRYCOLLECTION, MULTIPOINT, MULTILINESTRING, MULTICURVE ή MULTIPOLYGON. Διαφορετικά, επιστρέφει NULL.

Spatial relationship:

Q6: Find the roads which join road 'Fairlie Gardens'.

SELECT ST_AsText(geom)

FROM roads

WHERE name = 'Fairlie Gardens';

EXPLAIN (FORMAT JSON)SELECT name

FROM roads

WHERE ST_DWithin(

geom,

ST_GeomFromText('MULTILINESTRING((535464.655410508 173720.677114196,535470.799838103 173792.938794016,535486.457997386 173855.727675173))',

27700),

0.1

); AND a.name = 'Barnet';

(Εδώ, το 0.1 στο τέλος είναι η απόσταση σε μέτρα, το οποίο σημαίνει να βρεθεί ο δρόμος που η μεταξύ απόσταση είναι 0.1 μέτρα από το ***'Zulu Mews'***).

Q7: Find the total number of people who live within 300 meters of 'Big Ben building'.

SELECT name, ST_AsText(geom)

FROM buildings

WHERE name = 'Big Ben';

EXPLAIN (FORMAT JSON) SELECT Sum(population)

FROM borough

WHERE ST_DWithin(

geom,

ST_GeomFromText('MULTIPOLYGON(((530263.550278131 179633.565778108,530263.582804329 179633.922695463,530264.669214672 179646.547002348,530264.70202556 179646.892799396,530265.057132224 179646.857387052,530278.191523924 179645.725073007,530278.546345859 179645.700781928,530278.513820838 179645.343864455,530277.420788654 179632.708255917,530277.388263409 179632.351338459,530277.026500006 179632.37545179,530276.775785403 179632.402410568,530265.359948989 179633.41184594,530263.905100798 179633.541485999,530263.550278131 179633.565778108)))',

27700),

300

);

Λειτουργίες που χρησιμοποιούνται στις ερωτήσεις ***Spatial relationship:***

F7: ST_AsText ()

Η συνάρτηση ST_AsText() στο PostGIS επιστρέφει την αναπαράσταση του κειμένου (WKT) της γεωμετρίας/γεωγραφίας χωρίς μεταδεδομένα SRID.

F8: ST_GeomFromText ()

Η συνάρτηση ST_GeomFromText () στο PostGIS επιστρέφει μια καθορισμένη τιμή γεωμετρίας από την γνωστή αναπαράσταση κειμένου (WKT).

F9: ST_DWithin (γεωμετρία A, γεωμετρία B, ακτίνα)

Η συνάρτηση ST_DWithin (γεωμετρία A, γεωμετρία B, ακτίνα) στο PostGIS επιστρέφει TRUE εάν οι γεωμετρίες βρίσκονται εντός της καθορισμένης απόστασης μεταξύ τους.

F10: ST_Intersects (γεωμετρία A, γεωμετρία B)

Επιστρέφει TRUE εάν οι γεωμετρίες/γεωγραφία "τέμνονται χωρικά") και FALSE εάν δεν.

Spatial Joins:

Q8: Find the ward name of 'Forest Gate' subway station.

SELECT

stations.name,

wards.lad11nm,

wards.lad11cd

FROM wards

JOIN stations

ON ST_Contains(wards.geom,

stations.geom)

WHERE stations.name = 'Forest Gate';

Λειτουργίες που χρησιμοποιούνται στις ερωτήσεις ***Spatial Join:***

F11: ST_Contains (γεωμετρία A, γεωμετρία B)

Η συνάρτηση ST_Contains() στο PostGIS επιστρέφει TRUE εάν και μόνο αν δεν υπάρχουν σημεία B που βρίσκονται στο εξωτερικό του A και τουλάχιστον ένα σημείο του εσωτερικού του B βρίσκεται στο εσωτερικό του A.

F12: ST_Distance (γεωμετρία A, γεωμετρία B)

Η συνάρτηση ST_Distance (γεωμετρία A, γεωμετρία B) στο PostGIS επιστρέφει 2-διαστατική καρτεσιανή ελάχιστη απόσταση μεταξύ δύο γεωμετριών.

Nearest Neighborhoods:

Q9: Show a list of all the names of borough adjoining the Barnet borough;

EXPLAIN (FORMAT JSON) SELECT b.name

FROM borough a, borough b

WHERE st_touches(a.geom, b.geom)

AND a.name = 'Barnet';

Q10: What is the ward of the 'Mottingham' station;

SELECT name, ST_AsText(geom)

FROM stations

WHERE name = 'Mottingham';

EXPLAIN (FORMAT JSON)SELECT lad11nm

FROM wards

WHERE ST_Intersects(geom, ST_GeomFromText ('POINT ZM (542567.29834444 173238.949433276 0 0)',27700));

Q11: Find the ward name and code area of 'Zenoria Street'.

EXPLAIN (FORMAT JSON) SELECT lad11nm, lad11cd

FROM wards

WHERE ST_Intersects(

geom,

ST_GeomFromText('MULTILINESTRING((533811.085354646 175242.226502009,533802.865784929 175240.753036595,533720.274189245 175225.997597686))',

27700));

Q12: Find the ward name and code area of 'Subway Northern Line'.

EXPLAIN (FORMAT JSON) SELECT lad11nm, lad11cd

FROM wards

WHERE ST_Intersects(

geom,

ST_GeomFromText('MULTILINESTRING((525668.597162791 168779.72725613,525670.395365942 168708.414074064,525670.79495106 168696.429007071,525671.153387802 168688.092583552,525671.749836641 168680.830214566,525672.468737735 168675.095260976))',

27700));

Q13: Closest street to Battersea Park station

EXPLAIN (FORMAT JSON) SELECT roads.gid, roads.name

FROM

roads,

stations

WHERE stations.name = 'Battersea Park'

AND roads.geom && ST_Expand(stations.geom, 1000) -- Magic number: 1000m

ORDER BY ST_Distance(roads.geom, stations.geom) ASC

LIMIT 1;

Q14: Closest train station building to Battersea Park station

EXPLAIN (FORMAT JSON) SELECT buildings.gid, buildings.name, buildings.type

FROM

buildings,

stations

WHERE stations.name = 'Battersea Park' and buildings.type = 'train_station'

AND buildings.geom && ST_Expand(stations.geom, 200) -- Magic number: 200m

ORDER BY ST_Distance(buildings.geom, stations.geom) ASC;

Λίστα μερικών πιο σημαντικών λειτουργιών στο σύστημα διαχείρισης χωρικών βάσεων δεδομένων:

F13: ST_Length

Η συνάρτηση ST_Length στο PostGIS επιστρέφει το 2D μήκος της γεωμετρίας, εάν πρόκειται για inestring or multilinestring.

F8: ST_Perimeter

Η συνάρτηση ST_Perimeter επιστρέφει το συνολικό μήκος του ορίου του polygon or multipolygon.

F9: ST_X

Επιστρέφει τη συντεταγμένη X του σημείου.

F10: ST_Y

Επιστρέφει τη συντεταγμένη Y του σημείου.

F11: ST_Crosses (γεωμετρία A, γεωμετρία B)

Η συνάρτηση ST_Crosses (γεωμετρία A, γεωμετρία B) επιστρέφει TRUE εάν η γεωμετρία A και η γεωμετρία B έχουν κάποια κοινά σημεία εσωτερικού.

F12: ST_Disjoint (γεωμετρία A, γεωμετρία B)

Η συνάρτηση ST_Disjoint (γεωμετρία A, γεωμετρία B) επιστρέφει TRUE εάν οι γεωμετρίες δεν τέμνονται χωρικά.

F13: ST_Equals (γεωμετρία A, γεωμετρία B)

Αν και η γεωμετρία A και B παρουζιάζουν την ίδια γεωμετρία ανεξάρτητα από την κατεύθυνση τους, τότε η συνάρτηση ST_Equals (γεωμετρία A, γεωμετρία B) επιστρέφει TRUE

F14: ST_Overlaps (Γεωμετρία A, Γεωμετρία B)

Η συνάρτηση ST_Overlaps(Γεωμετρία A, Γεωμετρία B) επιστρέφει TRUE εάν και οι δύο γεωμετρίες έχουν την ίδια διάσταση και μοιράζονται χώρο αλλά δεν περιλαμβάνονται εντελώς μεταξύ τους

F15: ST_Touches (Γεωμετρία A, Γεωμετρία B)

Η συνάρτηση ST_Touches (Γεωμετρία A, Γεωμετρία B) επιστρέφει TRUE εάν οι εσωτερικές γεωμετριες δεν τέμνονται αλλά έχουν τουλάχιστον ένα κοινό σημείο.

F16: ST_Within (Γεωμετρία A, Γεωμετρία B)

Η συνάρτηση ST_Within (Γεωμετρία A, Γεωμετρία B) επιστρέφει TRUE εάν η γεωμετρία A είναι στο εσωτερικό στη γεωμετρία B.

6.3 Χωρικά Ερωτήματα σε SQLite

Τα ερωτήματα μεταξύ των δυο βάσεων δεδομένων διαφέρουν. Στην SQLite, όπως έχουμε ήδη αναφέρει δεν υπάρχει σχέση μεταξύ των πινάκων με ευρετήριο και χωρίς ευρετήριο.

Διακρίναμε τα παρακάτω ερωτήματα:

1. **Χωρίς ευρετήριο:**

```
SELECT b1.name AS "Borough",
b2.name AS "Neighour"
FROM borough AS b1,
  borough  AS b2
WHERE ST_Touches(b1.geom, b2.geom);
```

Με Ευρετήριο:

```
SELECT b1.name AS "Borough ",
  b2.name AS "Neighour"
FROM borough AS b1,
  borough  AS b2
WHERE b2.ROWID IN (
  SELECT pkid
    FROM idx_borough_geom
    WHERE pkid MATCH RTreeIntersects(
      MbrMinX(b1.geom),
      MbrMinY(b1.geom),
      MbrMaxX(b1.geom),
      MbrMaxY(b1.geom)));
```

2. **Χωρίς ευρετήριο:**

```
SELECT name, fclass FROM buildings
          WHERE X(geom) > 510000
 AND X(geom) < 517271
           AND Y(geom) > 159951
AND Y(geom) < 161823;
```

Με Ευρετήριο:

```
SELECT name,type,fclass FROM buildings WHERE ROWID IN
    (SELECT pkid FROM idx_buildings_geom WHERE
    xmin > 510000 AND xmax < 517271 AND ymin > 159951 AND ymax < 161823);
```

Τα πρώτα ερωτήματα δεν έχουν τίποτα ενδιαφέρον. Είναι 'Ασήμαντα' ερωτήματα και δεν περιλαμβάνει το Χωρικό Ευρετήριο. Τα δεύτερα ερωτήματα εκμεταλλεύονται πλήρως τη χρήση του χωρικού ευρετηρίου.

6.4 Ερωτήματα σε Πολυδιάστατα Δεδομένα

Τα ερωτήματα εύρους για τα πολυδιάστατα δεδομένα είναι ήδη ερωτημάτων που δυσκολεύουν μία βάση δεδομένων σε πολλές εφαρμογές. Η εκτέλεσή τους μπορεί να επιταχυνθεί χρησιμοποιώντας πολυδιάστατες δομές ευρετηρίου (MDIS), όπως kd-trees ή R-Trees.

Ένα ερώτημα εύρους είναι μια κοινή λειτουργία βάσης δεδομένων που ανακτά όλες τις εγγραφές όπου κάποια τιμή βρίσκεται μεταξύ ενός άνω και κάτω ορίου. Τα ερωτήματα εύρους θεωρούνται ασυνήθιστα, επειδή δεν είναι γενικά γνωστό εκ των προτέρων πόσες καταχωρήσεις θα επιστρέψει ένα ερώτημα εύρους ή εάν θα επιστρέψει καθόλου. Πολλά άλλα ήδη ερωτημάτων, όπως να επιστρέψει τους δέκα μεγαλύτερους αριθμού, μπορούν να γίνουν πιο αποτελεσματικά επειδή υπάρχει ένα ανώτατο όριο στον αριθμό των αποτελεσμάτων που θα επιστρέψουν. Ένα ερώτημα που επιστρέφει ακριβώς ένα αποτέλεσμα ονομάζεται μερικές μοναδικό.

Όπως έχουμε αναφέρει και σε προηγούμενο κεφάλαιο τα πολυδιάστατα δεδομένα αποτελούνται από 16 στήλες.

6.4.1 Πολυδιάστατα ερωτήματα σε PostgreSQL

Θα κατασκευάσουμε ευρετήριο στις πρώτες 5 στήλες του πίνακα letters και θα πραγματοποιήσουμε ερωτήματα εύρους.

CREATE INDEX gist5dim ON letters

USING gist

(

xbox ,

ybox ,

width ,

height,

onpix

);

CREATE INDEX gist5dim ON letters

USING brin

(

xbox ,

ybox ,

width ,

height,

onpix

);

Τα ερωτήματα μας θα βασίζονται σε 3 κατηγορίες

1. Βάση του μέσου όρου (παίρνουμε τις πρώτες 5 στήλες που έχουμε δημιουργήσει και το ευρετήριο)
2. Βάση του αριθμού που εμφανίζεται περισσότερο σε κάθε στήλη
3. Σε σταθερό διάστημα τιμών

Βάση του μέσου όρο:

Q1

explain (analyze, buffers)SELECT *

FROM letters

WHERE xbox BETWEEN 4 AND 5 AND

ybox BETWEEN 7 AND 8 AND

width BETWEEN 5 AND 6 AND

height BETWEEN 5 AND 6 AND

onpix BETWEEN 3 AND 4

Βάση του αριθμού που εμφανίζεται περισσότερο σε κάθε στήλη:

Q2

explain (analyze, buffers) SELECT *

FROM letters

WHERE xbox BETWEEN 2 AND 4 AND

ybox BETWEEN 8 AND 10 AND

width BETWEEN 4 AND 6 AND

height BETWEEN 5 AND 7 and onpix BETWEEN 1 AND 3

Σε σταθερό διάστημα τιμών:

Q3

explain (analyze, buffers) SELECT *

FROM letters

WHERE

xbox BETWEEN 6 AND 8 AND

ybox BETWEEN 3 AND 5 AND

width BETWEEN 0 AND 2 AND

height BETWEEN 1 AND 3 AND

onpix BETWEEN 7 AND 9 AND

xbar BETWEEN 12 AND 14 AND

ybar BETWEEN 1 AND 3 AND

x2bar BETWEEN 10 AND 12 AND

y2bar BETWEEN 13 AND 15 AND

xybar BETWEEN 11 AND 13 AND

x2ybar BETWEEN 4 AND 6 AND

xy2bar BETWEEN 6 AND 8 AND

xedge BETWEEN 4 AND 6 AND

xedgey BETWEEN 8 AND 10 AND

yedge BETWEEN 3 AND 5 AND

yedgex BETWEEN 8 AND 10

6.4.2 Πολυδιάστατα ερωτήματα σε SQLite

SELECT * FROM five_dim_index WHERE id=1;

Q1

1. **Χωρίς Ευρετήριο (Για id=1):**
   explain query plan SELECT id FROM letters
   WHERE minxbox>=2.0 AND maxxbox<=2.0
   AND minybox>=8.0 AND maxybox<=8.0
   AND minwidth>=3.0 AND maxwidth<=3.0
   AND minheight>=5.0 AND maxheigth<=5.0
   AND minonpix>=1.0 AND maxonpix<=1.0 ;

   Με ευρετήριο:
   explain query plan SELECT id FROM five_dim_index
   WHERE minxbox>=2.0 AND maxxbox<=2.0
   AND minybox>=8.0 AND maxybox<=8.0
   AND minwidth>=3.0 AND maxwidth<=3.0
   AND minheight>=5.0 AND maxheigth<=5.0

AND minonpix>=1.0 AND maxonpix<=1.0 ;

Q2

2. **Χωρίς Ευρετήριο(Διάστημα 2 τιμών):**

explain query plan SELECT id FROM letters
WHERE minxbox>=1.0 AND maxxbox<=3.0
AND minybox>=8.0 AND maxybox<=10.0
AND minwidth>=3.0 AND maxwidth<=5.0
AND minheight>=4.0 AND maxheigth<=6.0
AND minonpix>=6.0 AND maxonpix<=8.0 ;

Με ευρετήριο:

explain query plan SELECT id FROM five_dim_index
WHERE minxbox>=1.0 AND maxxbox<=3.0
AND minybox>=8.0 AND maxybox<=10.0
AND minwidth>=3.0 AND maxwidth<=5.0
AND minheight>=4.0 AND maxheigth<=6.0
AND minonpix>=6.0 AND maxonpix<=8.0 ;

ΚΕΦΑΛΑΙΟ 7 – ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΡΕΥΝΑΣ

7.1 Πλατφόρμα

Στην πειραματική μας έρευνα, παρουσιάζουμε τις μεθοδολογίες για τα πειράματά μας. Χρησιμοποιήσαμε επεξεργαστή Intel core i7-8550U, 1.80GHz - 2.00GHz, με μνήμη 8 GB στο λειτουργικό σύστημα Windows 10 64 bit. Πραγματοποιήσαμε πειράματα σε δυσδιάστατα και πολυδιάστατα δεδομένα στη PostgreSQL με ευρετηρίαση BRIN και GiST και στη SQLite που χρησιμοποιήσαμε το ευρετήριο R*-Trees. Όπως ήδη συζητήθηκε στο Κεφάλαιο 3 και 4, τα χωρικά δεδομένα που χρησιμοποιήσαμε ήταν ένα σύνολο δεδομένων αναφοράς του Λονδίνου, ενώ τα πολυδιάστατα δεδομένα είναι Δεδομένα αναγνώρισης γραμμάτων.

7.2. Μεθοδολογία

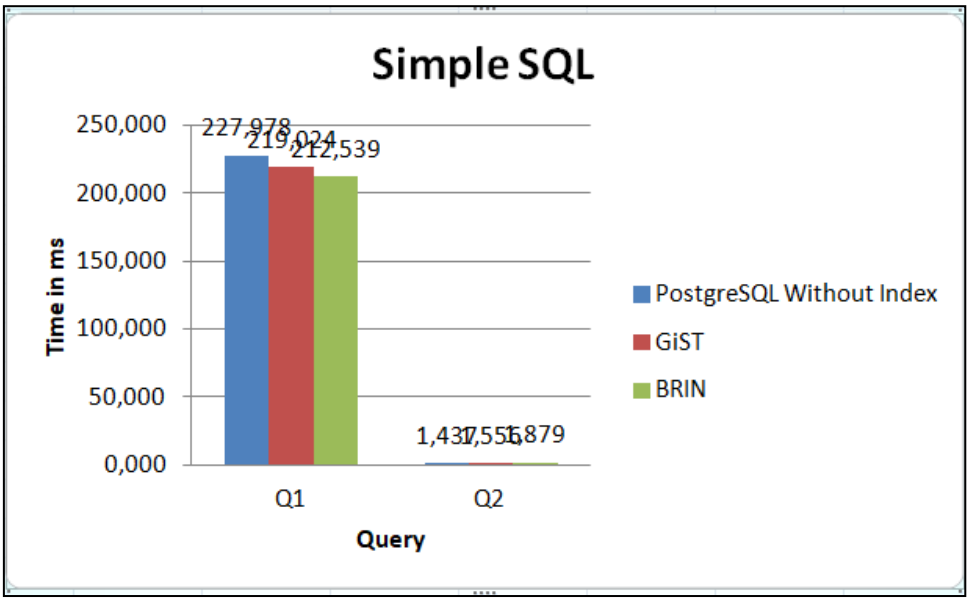
Εκτελέσαμε συνολικά τέσσερεις κατηγορίες ερωτημάτων, δύο για την PostgreSQL και δύο για τη SQLite με χωρικά και πολυδιάστατα δεδομένα. Στο τελευταίο κεφάλαιο, διαιρέσαμε τα ερωτήματα σύμφωνα με τις κατηγορίες τους. Στη έρευνα μας, εκτελέσαμε κάθε σύνολο ερωτημάτων στη βάση δεδομένων χωρίς και με ευρετήρια, GiST, BRIN και R*-tree και τελικά καταγράψαμε τον χρόνο εκτέλεσης. Εδώ θα περιγράψουμε τον χρόνο εκτέλεσης για κάθε ερώτημα από ιστογράμματα που θα μας βοηθήσουν να αξιολογήσουμε την απόδοση κάθε ευρετηρίου σε διαφορετικές κατηγορίες.

Στην επόμενη ενότητα θα δούμε τον χρόνο εκτέλεσης σε (σε ms) για κάθε κατηγορία χωρίς και με ευρετήρια.

7.3 Χρόνοι Χωρικών ερωτημάτων

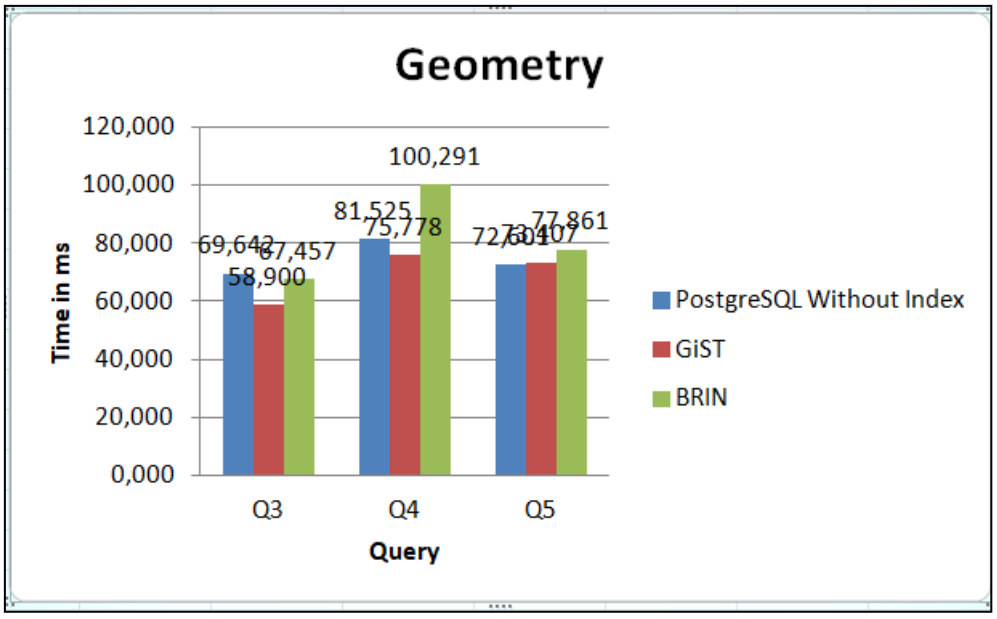
Σε αυτήν την ενότητα θα περιγράψουμε τον χρόνο εκτέλεσης κάθε κατηγορίας των χωρικών ερωτημάτων με τη βοήθεια ιστογραμμάτων. Κάθε γραμμή του ιστογράμματος θα αντιπροσωπεύει το χρόνο που απαιτείται (σε ms) από κάθε ερώτημα.

7.3.1 Χρόνοι χωρικών ερωτημάτων σε PostgreSQL



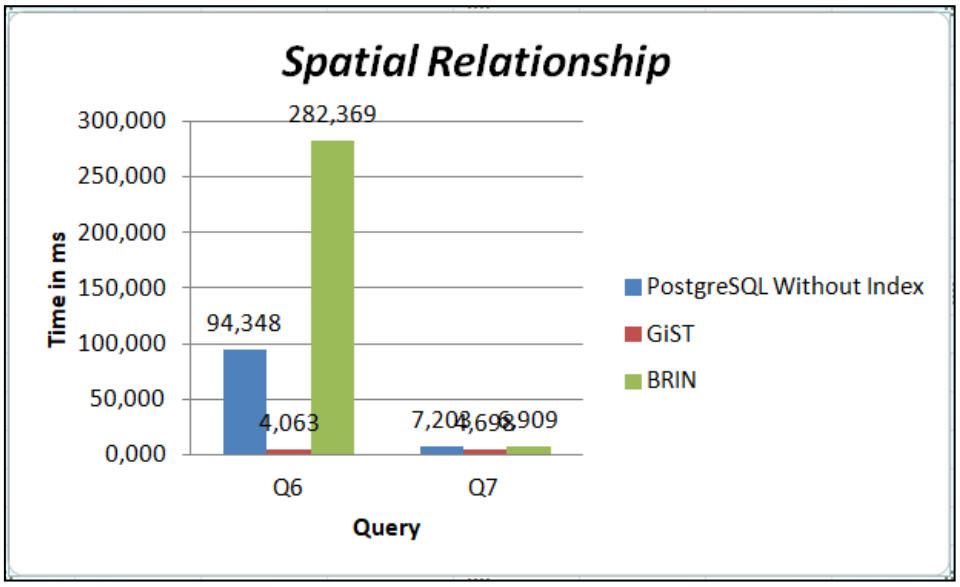
Εικόνα 24: Χρόνος που απαιτείται Simple SQL

Αυτό που θα παρατηρήσουμε στα ερωτήματα της Simple SQL(δείτε Εικόνα 24), είναι ότι τα ερωτήματα δεν χρησιμοποιούν κάποιο ευρετήριο, γι' αυτό το λόγο δεν υπάρχει κάποια διαφορά στους χρόνους με ή χωρίς ευρετήριο.



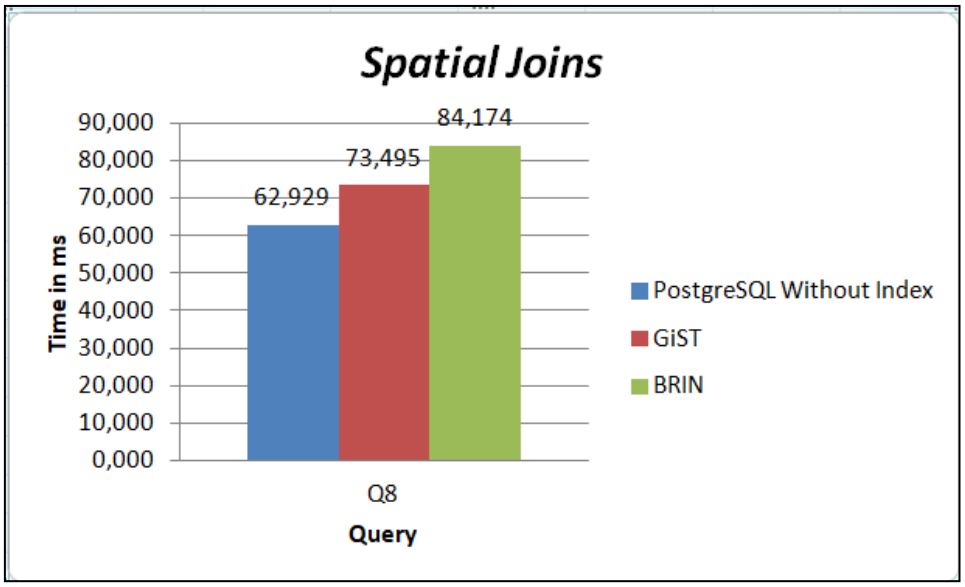
Εικόνα 25: Χρόνος που απαιτείται Geometry

Το ίδιο ακριβώς παρατηρείται και στα ερωτήματα Γεωμετρία(δείτε Εικόνα 25). Δεν γίνεται κάποια χρήση ευρετηρίου ακολουθούν seq. ανάλυση.



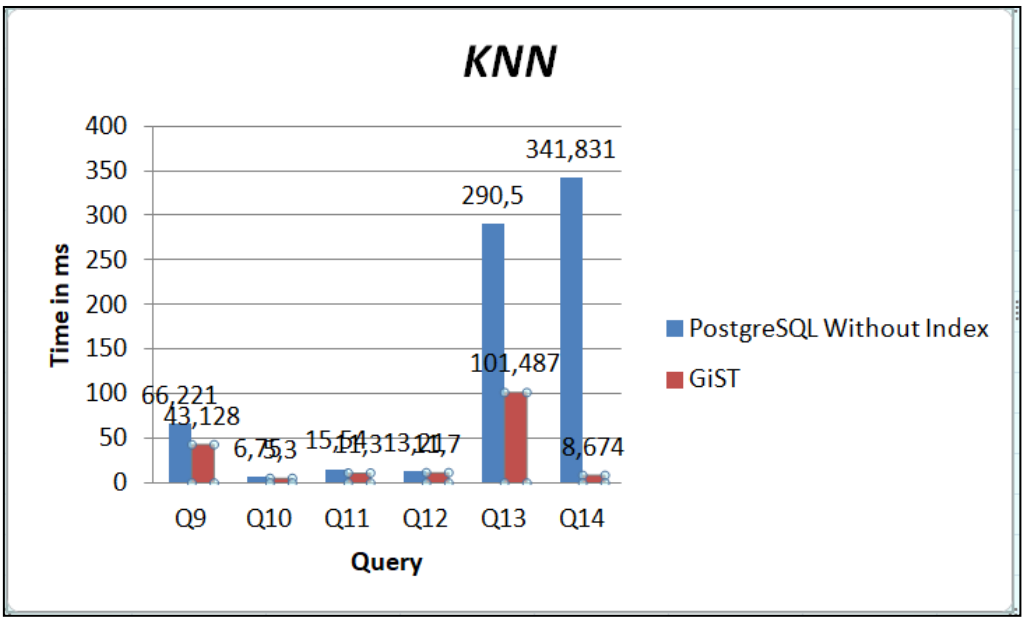
Εικόνα 26: Χρόνος που απαιτείται Spatial Relationship

Στα ερωτήματα χωρικών σχέσεων(δείτε Εικόνα 26) γίνεται χρήση των ευρετηρίων. Αυτό που θα παρατηρήσουμε είναι τη τεράστια διαφορά που σχηματίζεται μεταξύ των ερωτημάτων με ευρετήρια Brin και GiST.Το ευρετήριο Brin πραγματοποιεί τον τριπλάσιο χρόνο από την seq ανάλυση. Αυτό συμβαίνει γιατί ευρετηριοποιεί τους δρόμους δύο φορές ενώ το Gist μια.



Εικόνα 27: Χρόνος που απαιτείται Spatial Joins

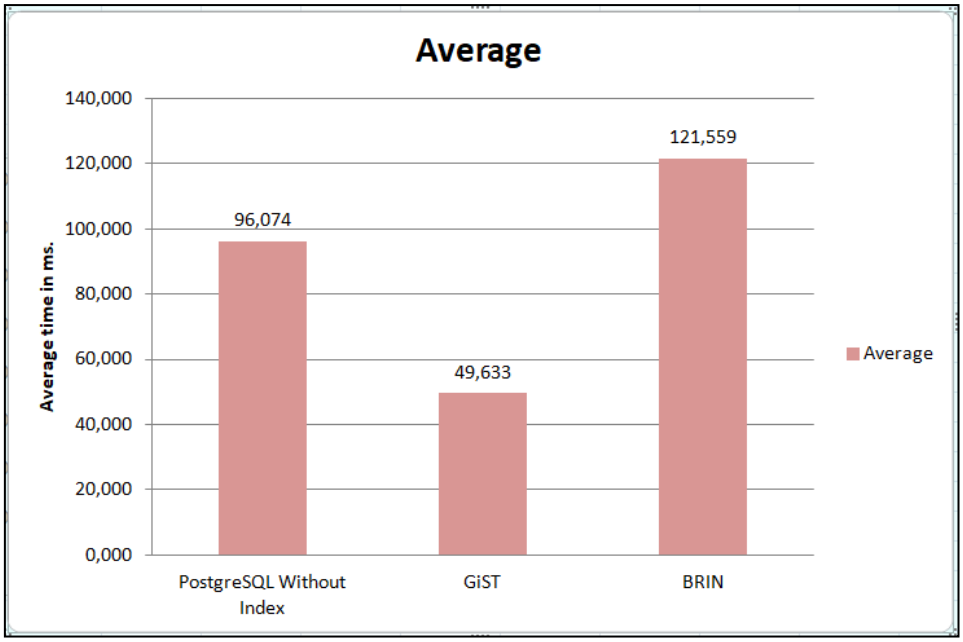
Στο ερώτημα που αφορά τα spatial joins(δείτε Εικόνα 27) παρατηρούμε ότι κανένα ευρετήριο δεν είναι αποτελεσματικό.



Εικόνα 28: Χρόνος που απαιτείται KNN

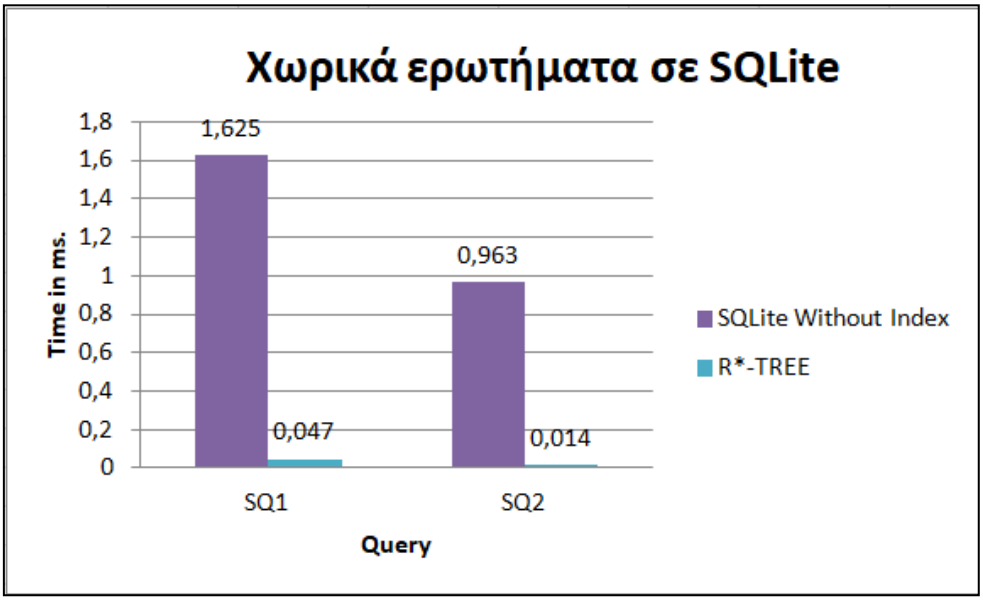
Θα πρέπει αρχικά να αναφέρουμε ότι το BRIN Index δεν χρησιμοποιείται σε ερωτήματα πλησιέστερου γείτονα(δείτε Εικόνα 28). Στο ερώτημα Q9 όπως θα παρατηρήσουμε και από το ιστόγραμμα, μόνο το ευρετήριο gist χρησιμοποιείται και είναι το καταλληλότερο για αυτού του είδους ερωτημάτων.

Με βάση τους μέσους όρους των χωρικών ερωτημάτων καταλήγουμε στο συμπέρασμα ότι στην PostgreSQL το ευρετήριο GiST είναι το πιο γρήγορο(δείτε Εικόνα 29).



Εικόνα 29:Μέσος όρος χρόνων για με και χωρίς ευρετήρια

7.3.2 Χρόνοι χωρικών ερωτημάτων σε SQLite



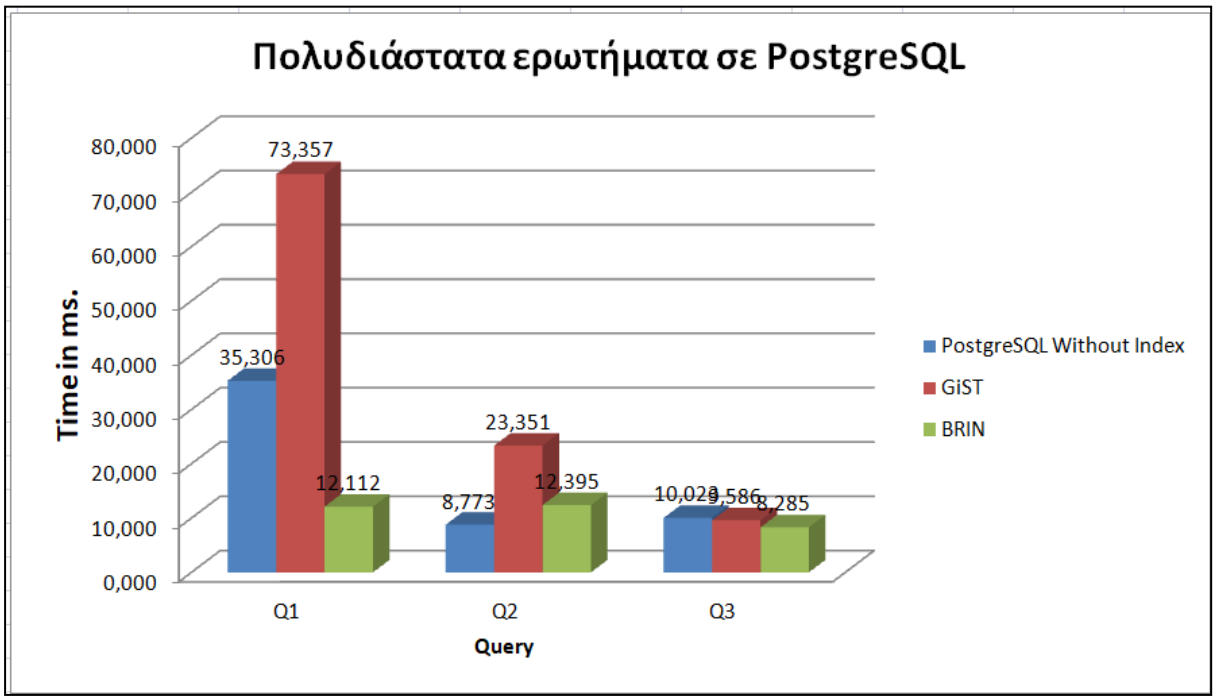
Εικόνα 30: Χωρικά ερωτήματα σε SQLite

Μπορούμε να διακρίνουμε ξεκάθαρα ότι η χρήση του ευρετηρίου R*-Tree βελτιώνει την απόδοση των ερωτημάτων(δείτε Εικόνα 30).

7.4 Χρόνοι Πολυδιάστατων ερωτημάτων

Σε αυτήν την ενότητα θα περιγράψουμε τον χρόνο εκτέλεσης κάθε κατηγορίας των χωρικών ερωτημάτων με τη βοήθεια ιστογραμμάτων. Κάθε γραμμή του ιστογράμματος θα αντιπροσωπεύει το χρόνο που απαιτείται (σε ms) από κάθε ερώτημα.

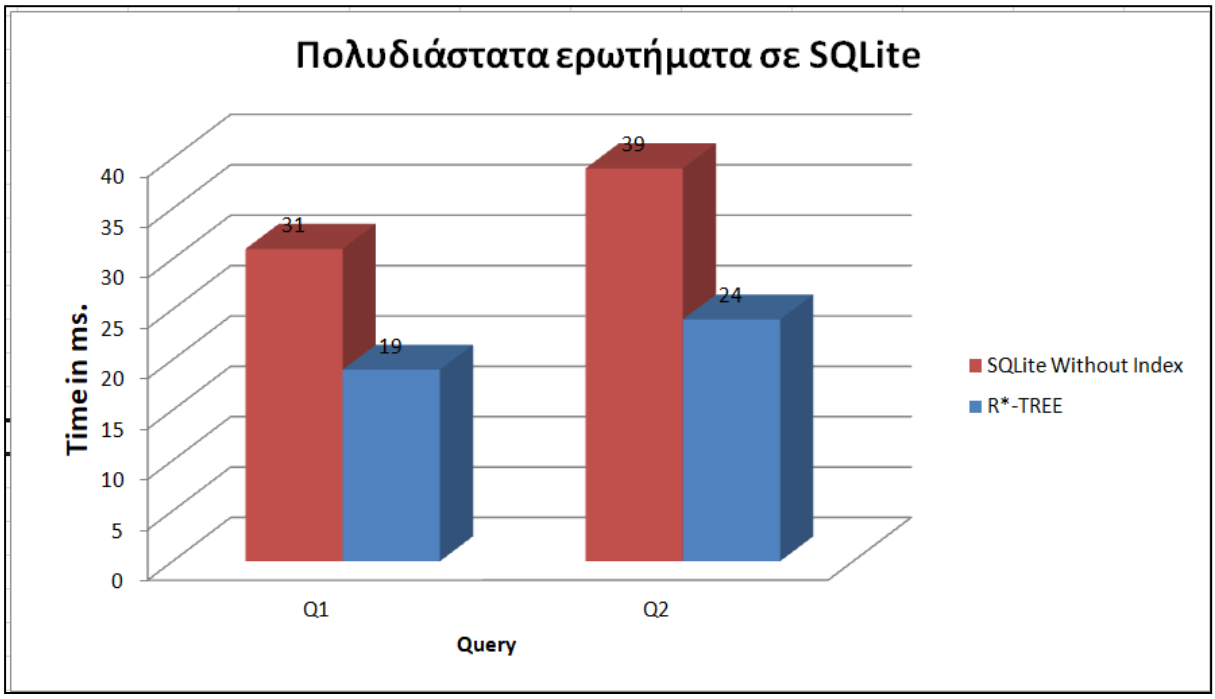
7.4.1 Χρόνοι πολυδιάστατων ερωτημάτων σε PostgreSQL



Εικόνα 31: Χρόνοι πολυδιάστατων ερωτημάτων σε PostgreSQL

Αυτό που παρατηρήσαμε από τις τρεις κατηγορίες ερωτημάτων(δείτε Εικόνα 31) είναι ότι το ευρετήριο GiST δεν είναι τόσο αποτελεσματικό όπως στα χωρικά δεδομένα. Αντίθετος το ευρετήριο BRIN παράγει καλύτερους χρόνους.

7.4.2 Χρόνοι πολυδιάστατων ερωτημάτων σε SQLite



Εικόνα 32: Χρόνοι πολυδιάστατων ερωτημάτων σε SQLite

Όπως και στα χωρικά δεδομένα, έτσι και στα πολυδιάστατα δεδομένα διακρίνουμε ξεκάθαρα ότι η χρήση του ευρετηρίου R*-Tree βελτιώνει την απόδοση των ερωτημάτων(δείτε Εικόνα 32).

ΚΕΦΑΛΑΙΟ 8 – ΣΥΜΠΕΡΑΣΜΑ ΚΑΙ ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

Σε αυτή τη διατριβή, συγκρίναμε την απόδοση τριών διαφορετικών ευρετηρίων για πέντε διαφορετικές κατηγορίες ερωτημάτων στα χωρικά δεδομένα και ερωτήματα εύρους για τα πολυδιάστατα δεδομένα. Οι δομές ευρετηρίασης που εφαρμόσαμε ήταν BRIN, GiST και R*-Trees σε δύο διαφορετικά Συστήματα Διαχείρισης Χωρικής Βάσης Δεδομένων, συγκεκριμένα PostgreSQL και SQLite.

Αφού εκτελέσαμε διάφορα εκτεταμένα ερωτήματα, καταλήξαμε στο συμπέρασμα ότι τα ευρετήρια είναι πολύ σημαντικά γιατί ο χρόνος εκτέλεσης των ερωτημάτων μειώνεται αρκετά.

Όσον αφορά τα χωρικά ερωτήματα στην PostgreSQL, το ευρετήριο GiST είχε καλύτερες επιδόσεις από το Brin. Στην SQLite διακρίναμε ξεκάθαρα ότι η χρήση του ευρετηρίου R*-Tree βελτιώνει την απόδοση των ερωτημάτων.

Στα πολυδιάστατα δεδομένα παρατηρήσαμε ότι το ευρετήριο GiST δεν είχε την καλύτερη επίδοση όπως τα χωρικά. Στην SQLite η χρήση του ευρετηρίου R*-Tree βελτιώνει σημαντικά την απόδοση των ερωτημάτων.

- Είναι σημαντικό να τονίσουμε ότι δεν μπορούμε να συγκρίνουμε τα ευρετήρια της μίας βάσης με τα ευρετήρια της άλλης.

Η έρευνα δεν θα σταματήσει σε αυτό το σημείο. Κύριος στόχος μας είναι να τρέξουμε πιο πολύπλοκα ερωτήματα και να καταγράψουμε του χρόνους.

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία

1. Intelligent Information and Database Systems: 10th Asian Conference, ACIIDS 2018, Dong Hoi City, Vietnam, March 19-21, 2018
2. S. Sumathi, S. Esakkirajan Springer(2007), Fundamentals of Relational Database Management Systems
3. John P. Wilson A. Stewart Fotheringham(2008,)The Handbook of Geographic Information Science,
4. Michael Owens (2006), The Definitive Guide to SQLite, United States of America
5. Rivero, Laura C (2005), Encyclopedia of Database Technologies and Applications
6. Ausubel, D. P. (1968). Educational Psychology: A Cognitive View. New York: Holt, Rinehart και Winston.

8. Ευστάθιος Γ. Κύρκος, Επιχειρηματική Ευφυΐα & Εξόρυξη Δεδομένων, Αθήνα 2015
9. Κωνσταντίνος Χ. Πατρούμπας, Μέθοδοι Πολυδιάστατης Προσπέλασης σε Βάσεις Δεδομένων με χρήση Δένδρων, ΑΘΗΝΑ 2004
10. Hans-Jürgen Schönig, Mastering PostgreSQL 10: Expert techniques on PostgreSQL 10 development and administration, -31 Ιαν 2018
11. Norbert Beckmann, Hans-Peter Kriegal, Ralf Schneider and Bernhard Seeger: The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles, Proceeding SIGMOD '90 Proceedings of the 1990 ACM SIGMOD international conference on Management of data.
12. Ralf Hartmut, An Introduction to Spatial Database Systems Güting Praktische Informatik IV, FernUniversität Hagen D-58084 Hagen, Germany
13. ESRI Shapefile Technical Description – An ESRI White Paper, July 1998
14. Chrisman N.R. (2003), Exploring Geographical Information Systems, 2nd Edition, Hoboken, NJ: Wiley
15. H. Samet. The Design and Analysis of Spatial Data Structures. Addison-Wesley Publishing Company, Inc, 1990.

Β.2.2 Άρθρα σε βιβλία-

Β.2.3 Ανέκδοτες Πηγές (Εργασίες / Διατριβές)

16. Ιωσηφίδης Ελευθέριος(2010), Διαχείριση Πολυδιάστατων Δεδομένων: Πειραματική και Συγκριτική Αξιολόγηση της Απόδοσης Εμπορικών και Ανοικτού Κώδικα DBMS, Πανεπιστήμιο Μακεδονίας
17. NEELABH PANT (2015), PERFORMANCE COMPARISON OF SPATIAL INDEXING STRUCTURES FOR DIFFERENT QUERY TYPES, THE UNIVERSITY OF TEXAS AT ARLINGTON

19. Κοντόπουλος Γεώργιος (2010), Ανάπτυξη διαδικτυακών εφαρμογών GIS με λογισμικό ανοιχτού κώδικα (Geoserver), Θεσσαλονίκη Πανεπιστήμιο Μακεδονίας

Β.3 Αρθρογραφία

20. Joseph M. Hellerstein, Jeffrey F. Naughton, Avi Pfeffer, Generalized Search Trees for Database Systems
21. Suhaibah Azri, Uznir Ujang, Franc¸ois Anton, Darka Mioc and Alias Abdul Rahman, Review of Spatial Indexing Techniques for Large Urban Data Management, January 2013
22. Marcel Kornacker, Access Methods for Next-Generation Database Systems, CALIFORNIA 2000

Β.4 Ηλεκτρονικά Περιοδικά

24. Boston Geographic Information Systems, Part 1: Getting Started with SpatiaLite, https://www.bostongis.com/
25. index types in postgresql 10 you should know, Sugandha Lahoti February 28, 2018, https://hub.packtpub.com/ (Ιούνιος 27 2020)

26. Egor Rogov, Indexes in PostgreSQL — 9 (BRIN), https://habr.com/, June/3/2019 (Ιούνιος 27 2020)
27. Well-Known Users of SQLite, https://www.sqlite.org/ (Ιούνιος 27 2020)
28. The SQLite R*Tree Module, https://sqlite.org/index.html(Ιούνιος 27 2020)

B.5 Ιστοσελίδες

29. 65.3. Extensibility, https://www.postgresql.org/ (Ιούνιος 27 2020)
30. PostgreSQL 9.6.18 Documentation, 11.2. Index Types, https://www.postgresql.org/ (Ιούνιος 27 2020)
31. PostgreSQL Tutorial, https://www.javatpoint.com/postgresql-tutorial, (Ιούνιος 27 2020)
32. PostGIS, Documentation, http://postgis.net/documentation/(Ιούνιος 27 2020)
33. Understanding caching in Postgres - An in-depth guide, https://madusudanan.com/, Originally Posted On: 16 May 2016 (Ιούνιος 27 2020)

Πηγές Δεδομένων:

34. https://data.london.gov.uk/, OpenStreetMap, Greater London Authority (GLA) (Τελευταία Πρόσβαση: 07/05/2020)
35. https://data.london.gov.uk/, Statistical GIS Boundary Files for London, Greater London Authority (GLA) (Τελευταία Πρόσβαση: 07/05/2020)
36. https://www.arcgis.com/, London rail network (Τελευταία Πρόσβαση: 07/05/2020)
37. https://www.ordnancesurvey.co.uk/, Order OS OpenData