



ΠΑΝΕΠΙΣΤΗΜΙΟ ΜΑΚΕΔΟΝΙΑΣ

ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ

ΤΜΗΜΑΤΟΣ ΕΦΑΡΜΟΣΜΕΝΗΣ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΕΥΡΕΤΗΡΙΟΠΟΙΗΣΗ ΣΕ ΥΨΗΛΕΣ ΔΙΑΣΤΑΣΕΙΣ - ΠΕΙΡΑΜΑΤΙΚΗ ΜΕΛΕΤΗ

ΕΥΡΕΤΗΡΙΩΝ GIST, BRIN ΚΑΙ R-TREE ΣΤΙΣ PostgreSQL ΚΑΙ SQLite*

Τριανταφύλλου Αναστασία

«mai19075@uom.edu.gr»

Ερωτήματα που θα απαντηθούν;

1. Τι είναι ευρετήρια;
2. Πως λειτουργούν και ποιες είναι οι ιδιότητες των ευρετηρίων GiST, BRIN και R*-TREE.
3. Τι είναι χωρικά και τι πολυδιάστατα Δεδομένα;
4. Ποιες DBMS υποστηρίζουν τα συγκεκριμένα ευρετήρια και πως;
5. Πετύχαμε τον αρχικό μας στόχο;

Μεθοδολογία και Στόχος Έρευνας

Θα μελετήσουμε τα ευρετήρια GiST, BRIN και R*-Tree σε δύο διαφορετικά συστήματα διαχείρισης βάσης δεδομένων, PostgreSQL και SQLite.

Συνολικά έχουμε 4 κατηγορίες ερωτημάτων με και χωρίς ευρετήρια:

1. PostgreSQL – Χωρικά Δεδομένα (14 Ερωτήματα[Simple SQL, Geometry, Spatial relationship, Spatial Joins, Nearest Neighborhoods])
2. PostgreSQL – Πολυδιάστατα Δεδομένα
3. SQLite – Χωρικά Δεδομένα
4. SQLite – Πολυδιάστατα Δεδομένα

⇒ Χρησιμοποιούμε γεωγραφικά δεδομένα με μορφή shapefile.

⇒ Τα πολυδιάστατα δεδομένα είναι οι εγγραφές Αναγνώρισης Γραμμάτων (Letter Recognition)

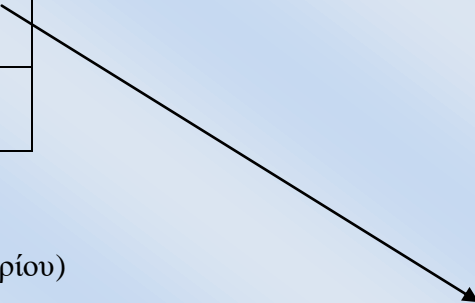
Ο κύριος στόχος αυτής της έρευνας είναι να δούμε αν όντως τα ευρετήρια βελτιώνουν την απόδοση των ερωτημάτων στους δύο τύπους δεδομένων.

Εισαγωγή στα Ευρετήρια

Τι είναι ευρετήριο (index);

γνώρισμα	

Ευρετήριο (Αρχείο Ευρετηρίου)



γνώρισμα	υπόλοιπα γνωρίσματα

Αρχείο Δεδομένων

Εγγραφή στο ευρετήριο:

Πηγή: Βάσεις Δεδομένων 2002-2003 Ευαγγελία Πιτουρά

Τιμή Πεδίου Ευρετηριοποίησης	Δείκτης στο block της εγγραφής
------------------------------	--------------------------------

GiST(Generalized Search Tree) Ευρετήριο

Τα γενικευμένα δένδρα αναζήτησης (Generalized Search Trees, GiST) [HNP95] επινοήθηκαν για να προσφέρουν καλύτερη προσπέλαση:

- σε σύνθετους τύπους δεδομένων (λ.χ. με πολλές διαστάσεις)
- σε νέους τύπους δεδομένων (π.χ. σύνολα τιμών)
- καθώς και υποστήριξη σε ευρύτερη ποικιλία ερωτημάτων που τίθενται σ' αυτά.

Δομή:

Είναι ισοζυγισμένα αφού όλα τα φύλλα βρίσκονται στο ίδιο επίπεδο.

Στηρίζονται σε μια αφηρημένη έννοια της αναζήτησης: κάθε κλειδί σε ένα κόμβο θα πρέπει να ικανοποιεί όλες τις πλειάδες των υποδένδρων του.

Λειτουργίες της μεθόδου GiST:

- Αναζήτηση
- Εισαγωγή
- Διαγραφή

Διαφορές με ένα ευρετήριο BTree:

- Τα δεδομένα δεν είναι οργανωμένα
- Τα βασικά εύρη μπορούν να αλληλεπικαλύπτονται

Operators:

<< &< &> >> <<| &<| |&> |>> @> <@ ~= &&|

Εισαγωγή ευρετηρίου στην PostgreSQL:

```
CREATE INDEX ("index_name")  
ON ("table_name")  
USING gist ("column");
```

BRIN (Block Range Index)

Η PostgreSQL 9.5 εισήγαγε μια δυνατότητα που ονομάζεται ευρετήριο εύρους μπλοκ, "Block Range Index".

- Είναι γρήγορο.
- Δεν είναι ένα ισορροπημένο δέντρο.
- Προορίζονται να βελτιώσουν την απόδοση ερωτημάτων σε εξαιρετικά μεγάλους πίνακες
- Για κάθε εύρος, υπολογίζει μια 'σύνοψη' των μπλοκ
 - i. Ελάχιστο/Μέγιστο: για αριθμητικές τιμές. Βασικές τιμές προστίθενται ακόλουθα κριτήρια ταξινόμησης + τελεστές ταξινόμησης
 - ii. Σύνοψη: για πιο δύσκολους τύπους δεδομένων. Οι βασικές τιμές προστίθενται σύμφωνα με τα κριτήρια Σύνοψης + τελεστές Σύνοψης.

Λειτουργίες της μεθόδου BRIN:

- Αναζήτηση
- Εισαγωγή

- Διαγραφή

Operators:

&& @ ~ (2D) &&& (3D, 4D)

Δεν υποστηρίζονται ερωτήματα kNN

Εισαγωγή ευρετηρίου στην PostgreSQL:

CREATE INDEX (“*index_name*”)

ON (“*table_name*”)

USING brin (“*column*”);

R*-TREE Index

Στο R*-Tree [BKSS90] : Norbert Beckmann, Hans-Peter Kriegel, Ralf Schneider και Bernard Seeger το 1990

Είναι μία παραλλαγή ενός R-Tree.

- Κριτήρια που αφορούν την ελαχιστοποίηση της έκτασης κάθε παραλληλογράμμου MBR
- Ελαχιστοποίηση επικάλυψης μεταξύ κάθε παραλληλογράμμου,
- Η ελαχιστοποίηση της περιμέτρου
- Βέλτιστη αξιοποίηση της χωρητικότητας των κόμβων του δένδρου.

Λειτουργίες της μεθόδου R*-Tree:

- Εισαγωγή (Insert)
- Reinsert
- Splitting

Η εισαγωγή του ευρετηρίου γίνεται διαφορετικά στα δύο είδη δεδομένων μας.

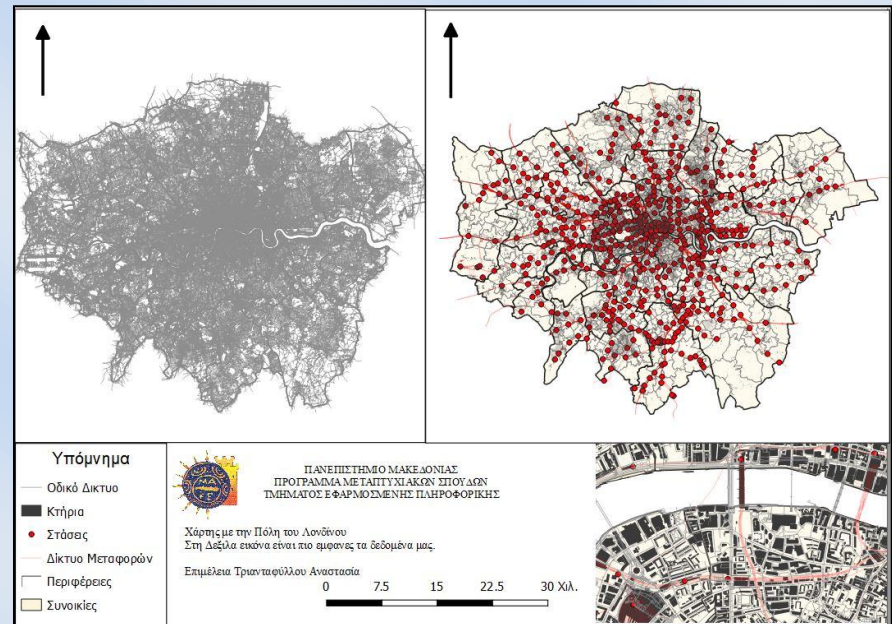
Χωρικά Δεδομένα (Spatial Data)

Τα χωρικά δεδομένα είναι της μορφής των shapefiles και είναι στοιχεία αναφοράς της πόλης του Λονδίνου στη Μεγάλη Βρετανία.

Είναι 2D – Δισδιάστατα δεδομένα

Αποτελείται από εγγραφές:

- σημεία: stations,
- γραμμές: roads, railways,
- πολύγωνα: borough, buildings, wards.



Εικόνα 1: Χάρτης με τα δεδομένα της έρευνας

Πολυδιάστατα Δεδομένα

Ο όρος "πολυδιάστατο" τείνει να εφαρμόζεται μόνο σε σύνολα δεδομένων με τρεις ή περισσότερες διαστάσεις. Τα πολυδιάστατα δεδομένα που θα χρησιμοποιήσουμε στην έρευνα μας είναι τα Δεδομένα αναγνώρισης γραμμάτων(Letter Image Recognition Data).

- Δημιουργήθηκαν από τον David J. Slate, Odesta Corporation; 1890 Maple Ave; Suite 115; Evanston, IL 60201, τον Ιανουάριο το 1991.
- Περιέχει 20.000 εγγραφές
- Περιέχει 17 Στήλες (1 κατηγορία γραμμάτων και 16 αριθμητικά χαρακτηριστικά)

Attribute Information:

1. letter	capital letter	(26 values from A to Z)	9. x2bar	mean x variance	(integer)
2. x-box	horizontal position of box	(integer)	10. y2bar	mean y variance	(integer)
3. y-box	vertical position of box	(integer)	11. xybar	mean x y correlation	(integer)
4. width	width of box	(integer)	12. x2ybr	mean of $x * x * y$	(integer)
5. height	height of box	(integer)	13. xy2br	mean of $x * y * y$	(integer)
6. onpix	total # on pixels	(integer)	14. x-ege	mean edge count left to right	(integer)
7. x-bar	mean x of on pixels in box	(integer)	15. xegvy	correlation of x-ege with y	(integer)
8. y-bar	mean y of on pixels in box	(integer)	16. y-ege	mean edge count bottom to top	(integer)
			17. yegvx	correlation of y-ege with x	(integer)

ΕΠΙΣΚΟΠΗΣΗ ΤΩΝ DBMS

Θα μελετήσαμε 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

Η PostgreSQL αποτελεί μια ανοικτού κώδικα σχεσιακή βάση δεδομένων (σχέση-αντικειμένου) με πολλές δυνατότητες (ORDBMS).

PostgreSQL + PostGIS (Refractions Research Inc)

Η SQLite (D. Richard Hipp) 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

SQLite + SpatiaLite

Εισαγωγή Χωρικών Δεδομένων στην PostgreSQL

1. Εκτελέσαμε από το μενού "PostgreSQL 9.6" το "pgAdmin4" και συνδεθείτε ως χρήστης postgres στην PostgreSQL. Δημιουργήσαμε μια νέα βάση με όνομα LondonUK. Επιλέξαμε την LondonUK και από το μενού Tools εκτελέστε το Query Tool.
2. Εκτελέσαμε την sql εντολή "create extension postgis".
3. 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: 1234, Database: LondonUK) και φορτώσαμε τα shapefiles.
4. 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..

Εισαγωγή Πολυδιάστατων Δεδομένων στην PostgreSQL

1. Αφού δημιουργήσαμε μια νέα βάση με όνομα letter-recognition..
2. 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);

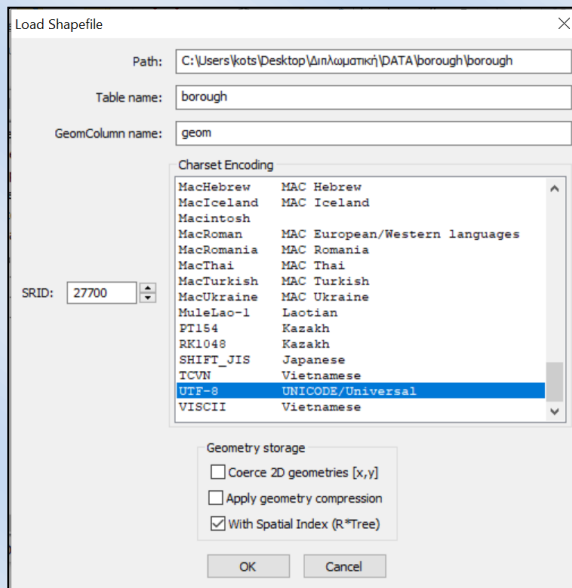
Στην συνέχεια θα πρέπει να κάνουμε εισαγωγή το αρχείο .data που κατεβάσαμε από το <https://archive.ics.uci.edu/>, USI, Machine Learning Repository, Letter Recognition Data Set. Εκτελούμε την παρακάτω εντολή στο Query Tool:

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

Εισαγωγή Χωρικών Δεδομένων στην SQLite

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

1. Αφού έχουμε εγκαταστήσει το spatialite-gui, το ανοίγουμε και δημιουργούμε μία βάση δεδομένων με όνομα LondonUK, όπως και στην PostgreSQL.
2. 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.

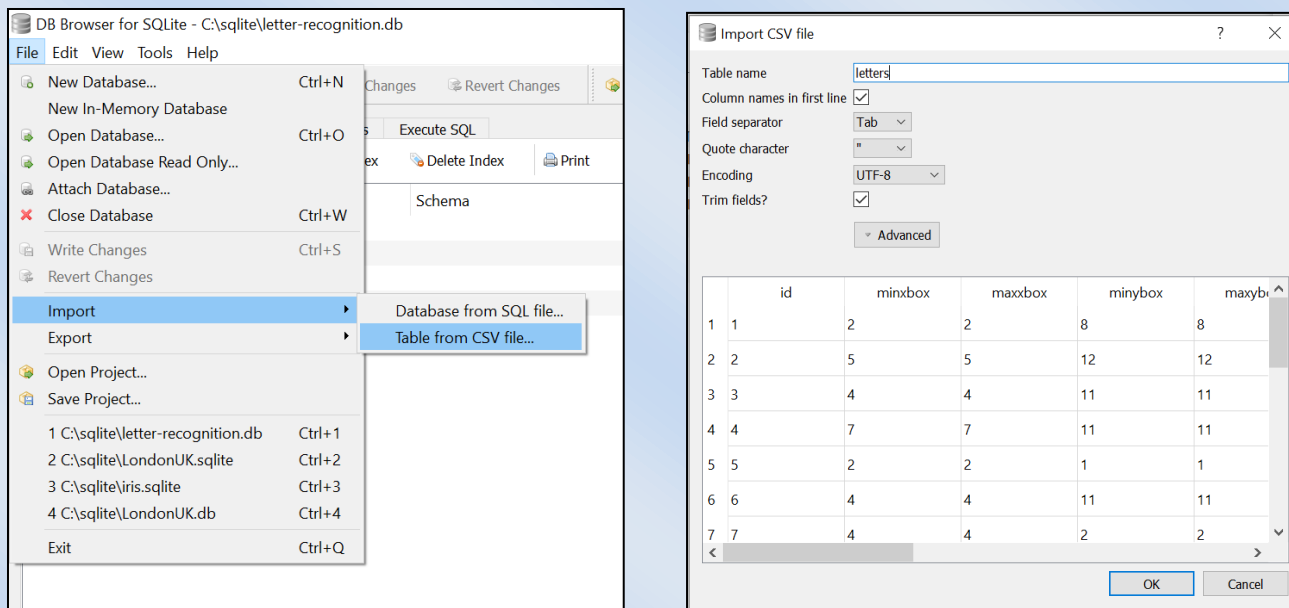


Εικόνα 2: Εισαγωγή Χωρικών Δεδομένων στην SQLite

Εισαγωγή Πολυδιάστατων Δεδομένων 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 3: Εισαγωγή Πολυδιάστατων Δεδομένων στην SQLite

Δημιουργία Ευρετηρίων σε Χωρικά Δεδομένα σε PostgreSQL

1. 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”);
```

2. Με τον παραπάνω τρόπο δημιουργούμε ένα ευρετήριο GiST στη στήλη γεωμετρίας (geom) και στους υπόλοιπους πίνακες.
3. Καταγράφουμε τον χρόνο εκτέλεσης και διαγράφουμε τα ευρετήρια.
4. Στη συνέχεια ευρετηριάσαμε τις γεωμετρίες με τη δομή BRIN και εκτελέσαμε το ίδιο σύνολο διαφορετικών χωρικών ερωτημάτων και καταγράψαμε τον χρόνο εκτέλεσης.
5. Η σύνταξη για τη δημιουργία του ευρετηρίου BRIN στις στήλες γεωμετρίας είναι:

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

Query Editor	Query History
<pre>1 CREATE INDEX inx_gist_london_borough 2 ON borough 3 USING gist (geom);</pre>	<pre>1 CREATE INDEX inx_brin_london_borough 2 ON borough 3 USING brin (geom);</pre>
Data Output Explain <u>Messages</u> Notifications	Data Output Explain <u>Messages</u> Notifications
CREATE INDEX	CREATE INDEX
Query returned successfully in 124 msec.	Query returned successfully in 76 msec.

Εικόνα 4: Δημιουργία Ευρετηρίων σε Χωρικά Δεδομένα σε PostgreSQL

Δημιουργία Ευρετηρίων σε Πολυδιάστατων Δεδομένα σε PostgreSQL

Στα πολυδιάστατα δεδομένα 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.

Δημιουργήσαμε ένα ευρετήριο για τις πρώτες 5 στήλες του αρχείου letter.

Πρώτα θα εισάγουμε το ευρετήριο GiST και έπειτα το BRIN:

```
CREATE INDEX gist5dim ON letters
```

```
USING gist
```

```
(
```

```
  xbox , ybox ,
```

```
  width , height,
```

```
  onpix
```

```
);
```

Δημιουργία Ευρετηρίων σε Χωρικά Δεδομένα σε SQLite

Κάθε R*Tree στη SQLite απαιτεί τέσσερις αλληλοσυσχετισμένους πίνακες:

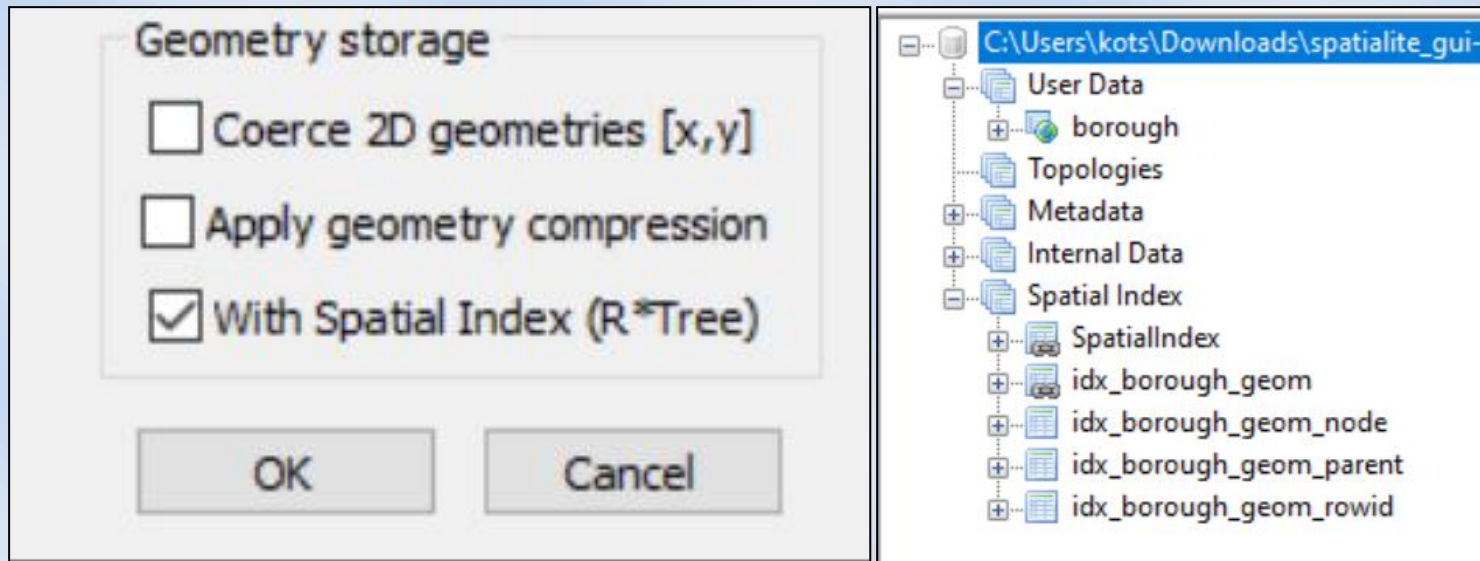
- `rtreebasename_node` αποθηκεύει (δυναμική μορφή) τους στοιχειώδεις κόμβους R * Tree.
- `rtreebasename_parent` αποθηκεύει σχέσεις που συνδέουν parent and child κόμβους.
- `rtreebasename_rowid` αποθηκεύει τιμές ROWID που συνδέουν έναν κόμβο R * Tree και μια αντίστοιχη σειρά στον πίνακα ευρετηρίου.

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 μέσω 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 5: Δημιουργία Ευρητηρίων σε Χωρικά Δεδομένα σε SQLite

Δημιουργία Ευρετηρίων σε Πολυδιάστατα Δεδομένα σε SQLite

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, maxheight,  -- Minimum and maximum 4 coordinate  
  minonpix, maxonpix  -- Minimum and maximum 5 coordinate  
);
```

ΕΡΩΤΗΜΑΤΑ (QUERIES)

Χωρικά Ερωτήματα σε PostgreSQL:

Simple SQL:

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

```
SELECT char_length(name)  
FROM roads  
WHERE fclass = 'residential';
```

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';
```

Spatial relationship:

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

);

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';
```

Nearest Neighborhoods:

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;
```

Πολυδιάστατα Ερωτήματα σε PostgreSQL:

Τα ερωτήματα μας θα βασίζονται σε 3 κατηγορίες

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

Βάση του μέσου όρου:

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
```

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.

Ένα από τα ερωτήματα:

1. 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);
```

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 maxheight<=5.0  
AND minonpix>=1.0 AND maxonpix<=1.0 ;
```

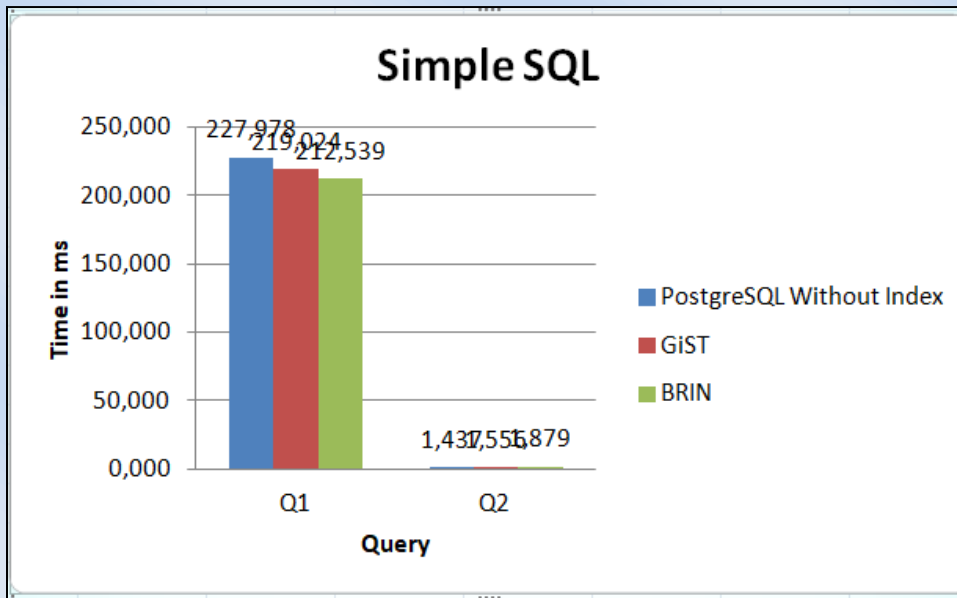
Με ευρετήριο:

```
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 maxheight<=5.0  
AND minonpix>=1.0 AND maxonpix<=1.0 ;
```

ΑΠΟΤΕΛΕΣΜΑΤΑ ΕΡΕΥΝΑΣ

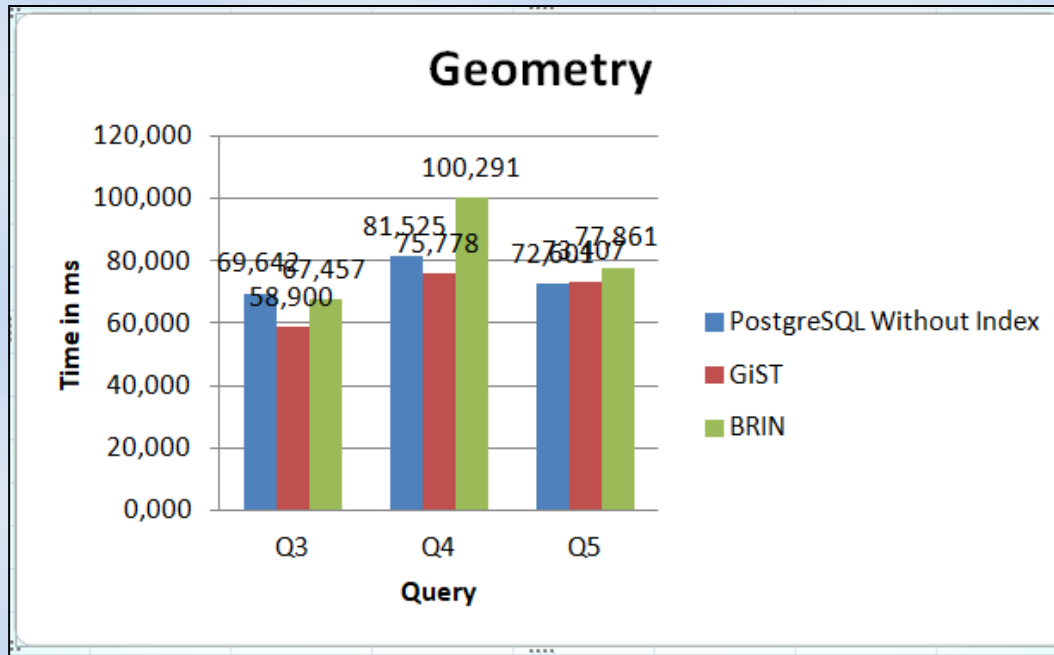
Χρησιμοποιήσαμε επεξεργαστή Intel core i7-8550U, 1.80GHz - 2.00GHz, με μνήμη 8 GB στο λειτουργικό σύστημα Windows 10 64 bit.

Χρόνοι χωρικών ερωτημάτων σε PostgreSQL



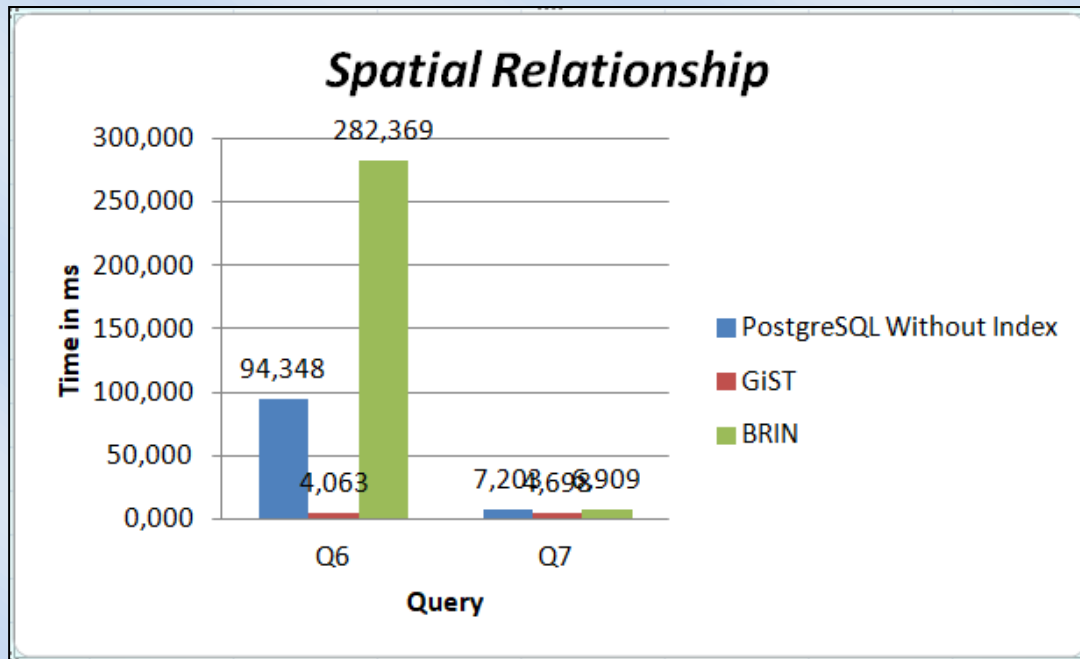
Εικόνα 6: Χρόνος που απαιτείται Simple SQL

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



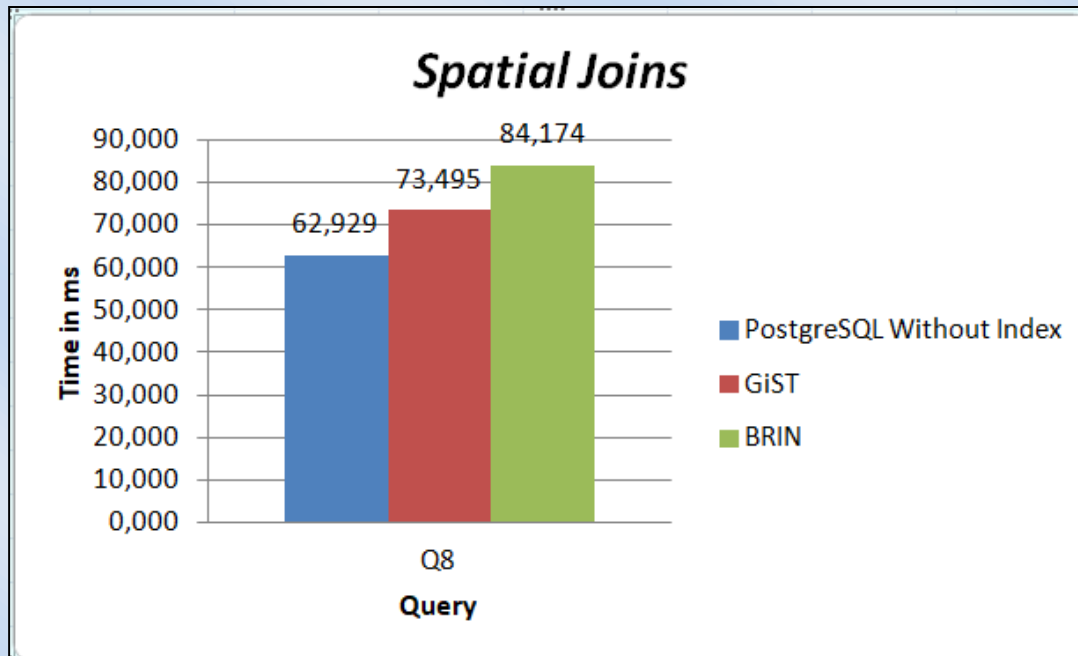
Εικόνα 7: Χρόνος που απαιτείται Geometry

Το ίδιο ακριβώς παρατηρείται και στα ερωτήματα Γεωμετρία. Δεν γίνεται κάποια χρήση ευρετηρίου ακολουθούν seq. ανάλυση.



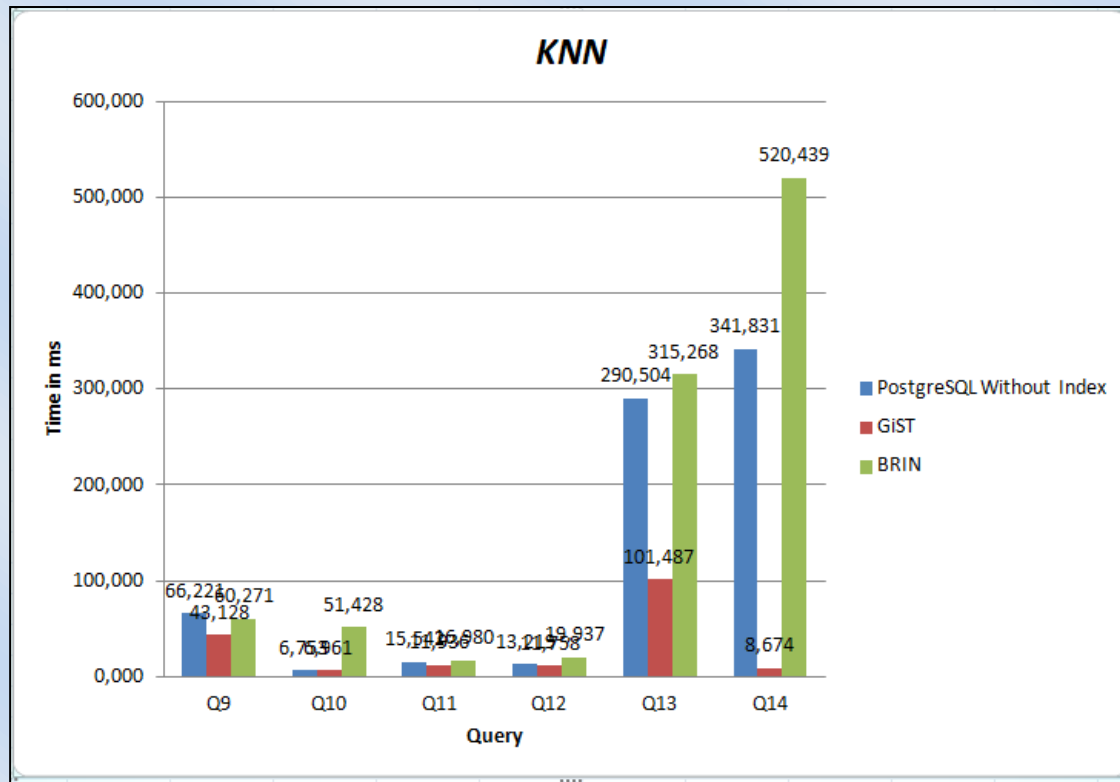
Εικόνα 8: Χρόνος που απαιτείται Spatial Relationship

Στα ερωτήματα χωρικών σχέσεων 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.



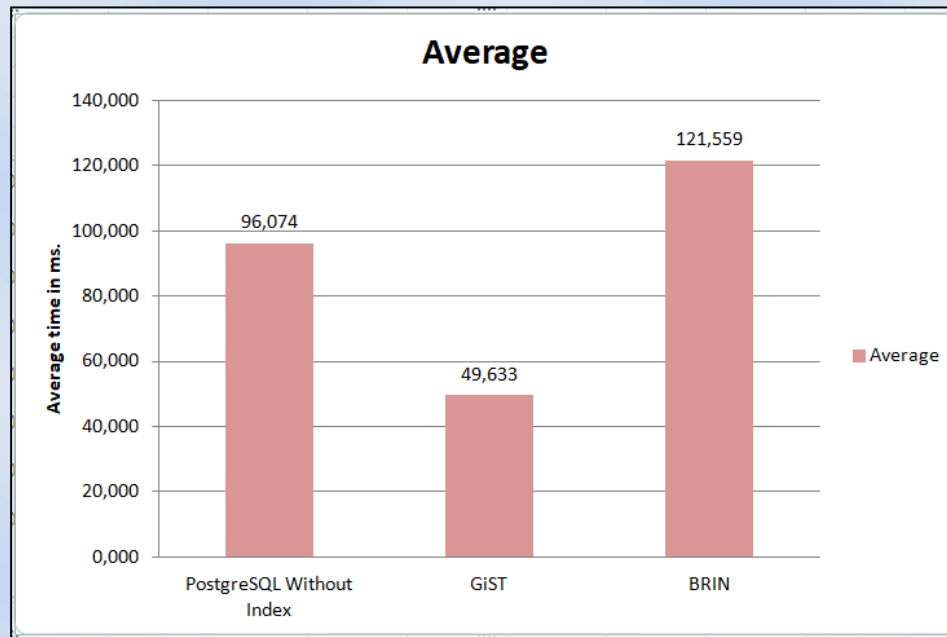
Εικόνα 9: Χρόνος που απαιτείται Spatial Joins

Στο ερώτημα που αφορά τα spatial joins παρατηρούμε ότι κανένα ευρετήριο δεν είναι αποτελεσματικό.



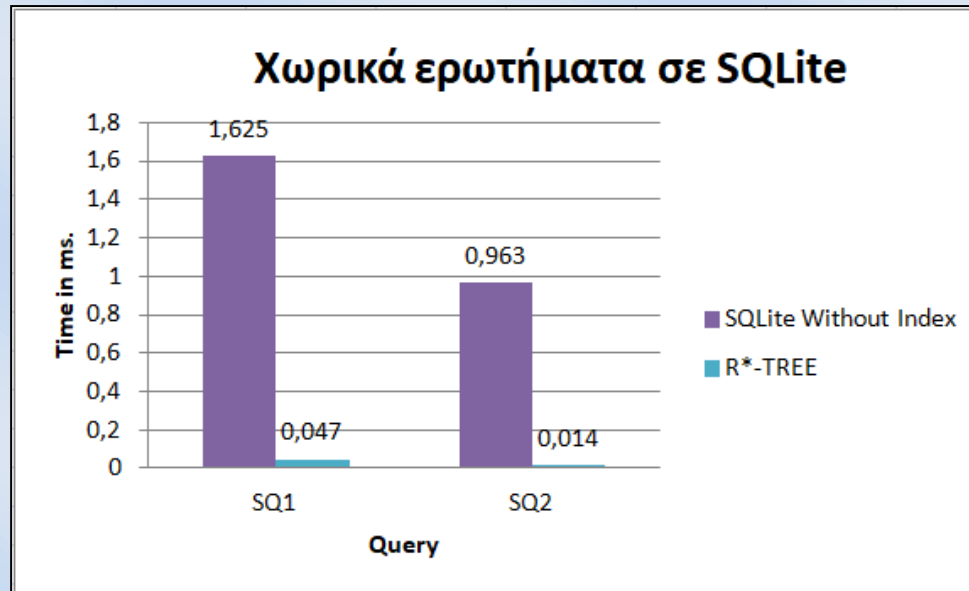
Εικόνα 10: Χρόνος που απαιτείται KNN

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



Εικόνα 11: Μέσος όρος χρόνων για με και χωρίς ευρετήρια

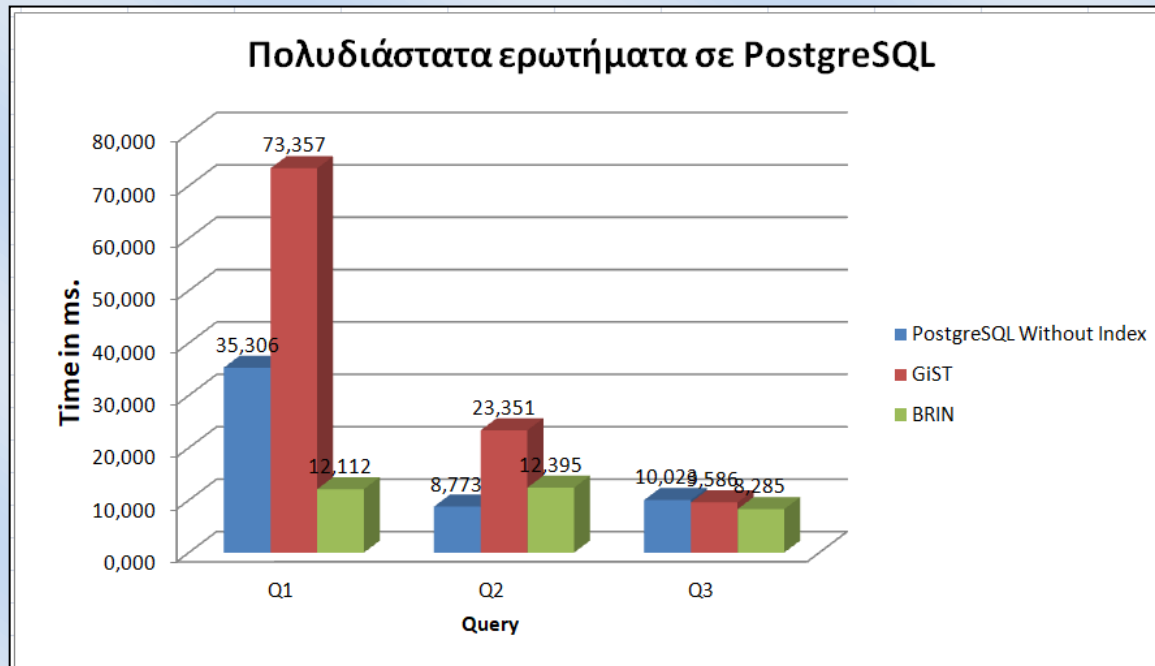
Χρόνοι χωρικών ερωτημάτων σε SQLite:



Εικόνα 12: Χωρικά ερωτήματα σε SQLite

Μπορούμε να διακρίνουμε ξεκάθαρα ότι η χρήση του ευρετηρίου R*-Tree βελτιώνει την απόδοση των ερωτημάτων.

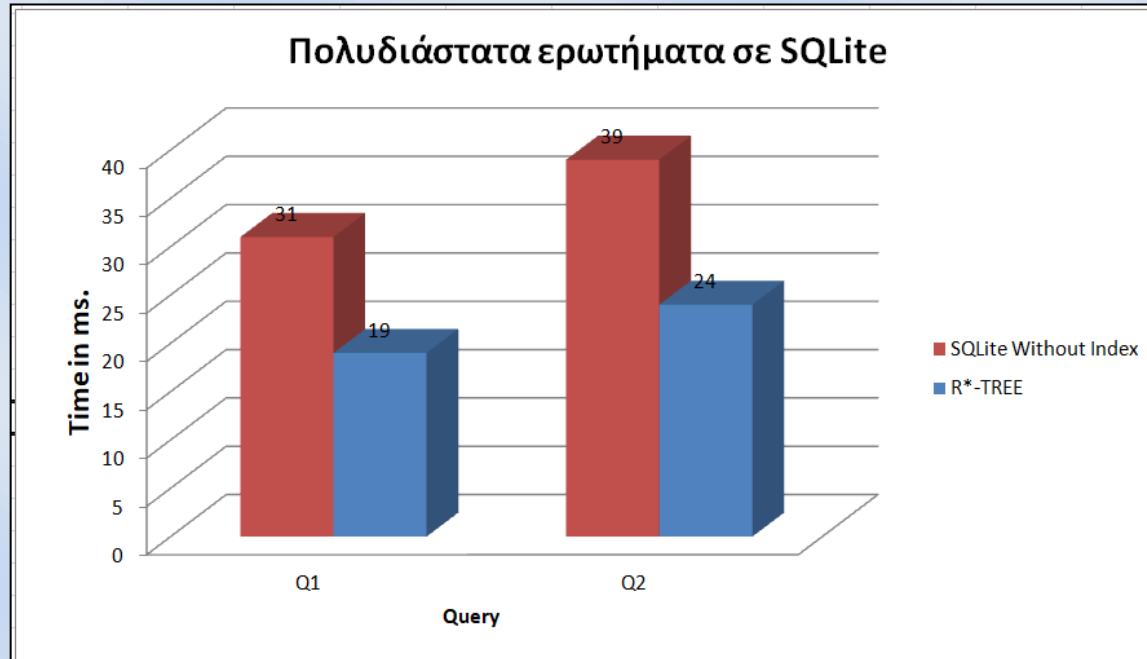
Χρόνοι πολυδιάστατων ερωτημάτων σε PostgreSQL:



Εικόνα 13: Χρόνοι πολυδιάστατων ερωτημάτων σε PostgreSQL

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

Χρόνοι πολυδιάστατων ερωτημάτων σε SQLite:



Εικόνα 14: 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.

ΣΥΜΠΕΡΑΣΜΑ

Οι δομές ευρετηρίασης που εφαρμόσαμε ήταν BRIN, GiST και R*-Trees σε δύο διαφορετικά Συστήματα Διαχείρισης Βάσης Δεδομένων, συγκεκριμένα PostgreSQL και SQLite, 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.

Ευχαριστώ!

ΒΙΒΛΙΟΓΡΑΦΙΑ

Βιβλία

Intelligent Information and Database Systems: 10th Asian Conference, ACIIDS 2018, Dong Hoi City, Vietnam, March 19-21, 2018

S. Sumathi, S. Esakkirajan Springer(2007), Fundamentals of Relational Database Management Systems

John P. Wilson A. Stewart Fotheringham(2008,)The Handbook of Geographic Information Science, Michael Owens (2006), The Definitive Guide to SQLite, United States of America

Rivero, Laura C (2005), Encyclopedia of Database Technologies and Applications

Ausubel, D. P. (1968). Educational Psychology: A Cognitive View. New York: Holt, Rinehart και Winston.

Whitehead, D. & Δ. Μακρίδου - Μπούσιου (1995). Οικονομική Εκπαίδευση - Διδακτική των Οικονομικών. Ένα Εγχειρίδιο για τους Καθηγητές των Οικονομικών.

Ευστάθιος Γ. Κύρκος, Επιχειρηματική Ευφυΐα & Εξόρυξη Δεδομένων, Αθήνα 2015

Κωνσταντίνος Χ. Πατρούμπας, Μέθοδοι Πολυδιάστατης Προσπέλασης σε Βάσεις Δεδομένων με χρήση Δένδρων, ΑΘΗΝΑ 2004

Hans-Jürgen Schönig, Mastering PostgreSQL 10: Expert techniques on PostgreSQL 10 development and administration, -31 Ιαν 2018

Norbert Beckmann, Hans-Peter Kriegel, 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.

Ralf Hartmut, An Introduction to Spatial Database Systems Güting Praktische Informatik IV, FernUniversität Hagen D-58084 Hagen, Germany

ESRI Shapefile Technical Description – An ESRI White Paper, July 1998

Chrisman N.R. (2003), Exploring Geographical Information Systems, 2nd Edition, Hoboken, NJ: Wiley

B.2.2 Άρθρα σε βιβλία

-

B.2.3 Ανέκδοτες Πηγές (Εργασίες / Διατριβές)

Ιωσηφίδης Ελευθέριος(2010), Διαχείριση Πολυδιάστατων Δεδομένων: Πειραματική και Συγκριτική Αξιολόγηση της Απόδοσης Εμπορικών και Ανοικτού Κώδικα DBMS, Πανεπιστήμιο Μακεδονίας

NEELABH PANT (2015), PERFORMANCE COMPARISON OF SPATIAL INDEXING STRUCTURES FOR DIFFERENT QUERY TYPES, THE UNIVERSITY OF TEXAS AT ARLINGTON

ΧΡΗΣΤΟΣ ΣΑΒΒΙΔΗΣ (2010), ΕΞΥΠΗΡΕΤΗΤΕΣ ΜΕΓΑΛΟΥ ΟΓΚΟΥ ΧΩΡΙΚΩΝ ΔΕΔΟΜΕΝΩΝ. ΜΕΛΕΤΗ ΠΕΡΙΠΤΩΣΗΣ ΤΟΥ ΑΝΟΙΚΤΟΥ ΛΟΓΙΣΜΙΚΟΥ POSTGIS, Πανεπιστήμιο Μακεδονίας.

B.3 Αρθρογραφία

Joseph M. Hellerstein, Jeffrey F. Naughton, Avi Pfeffer, Generalized Search Trees for Database Systems

Suhaibah Azri, Uznir Ujang, Francois Anton, Darka Mioc and Alias Abdul Rahman, Review of Spatial Indexing Techniques for Large Urban Data Management, , January 2013

Marcel Kornacker, Access Methods for Next-Generation Database Systems, CALIFORNIA 2000

[Yannis Manolopoulos-Alexandros Nanopoulos-Apostolos N. Papadopoulos-Yannis Theodoridis](#), R-Trees: Theory and Applications, 8 Σεπτεμβρίου 2010

B.4 Ηλεκτρονικά Περιοδικά

Boston Geographic Information Systems, Part 1: Getting Started with SpatiaLite, <https://www.bostongis.com/>

index types in postgresql 10 you should know, Sugandha Lahoti February 28, 2018, <https://hub.packtpub.com/> (Ιούνιος 27 2020)

Egor Rogov, Indexes in PostgreSQL — 9 (BRIN), <https://habr.com/>, June/3/2019 (Ιούνιος 27 2020)

Well-Known Users of SQLite, <https://www.sqlite.org/> (Ιούνιος 27 2020)

B.5 Ιστοσελίδες

65.3. Extensibility, <https://www.postgresql.org/> (Ιούνιος 27 2020)

PostgreSQL 9.6.18 Documentation, 11.2. Index Types, <https://www.postgresql.org/> (Ιούνιος 27 2020)

PostgreSQL Tutorial, <https://www.javatpoint.com/postgresql-tutorial>, (Ιούνιος 27 2020)

PostGIS, Documentation, <http://postgis.net/documentation/>(Ιούνιος 27 2020)

Understanding caching in Postgres - An in-depth guide, <https://madusudanan.com/>, Originally Posted On: 16 May 2016 (Ιούνιος 27 2020)

Spatialite, Software that supports Spatialite, <https://en.wikipedia.org/>(Ιούνιος 27 2020)

Πηγές Δεδομένων:

<https://data.london.gov.uk/>, OpenStreetMap, Greater London Authority (GLA) (Τελευταία Πρόσβαση: 07/05/2020)

<https://data.london.gov.uk/>, Statistical GIS Boundary Files for London, Greater London Authority (GLA) (Τελευταία Πρόσβαση: 07/05/2020)

<https://www.arcgis.com/>, London rail network (Τελευταία Πρόσβαση: 07/05/2020)