



Ιστορία Εγγράφου

Έκδοση	Ημ/νία	Συγγραφέας	Περιγραφή
1.0	22/06/2009	N. Κολιός	Αρχική έκδοση
2.0	12/11/2009	N. Κολιός	Αλλαγές: – Διόρθωση συνοδευτικού εκπαιδευτικού υλικού – Επιμέλεια κειμένου – Εισαγωγή ενοτήτων: 5.1.1.1 5.1.1.3 5.1.5 5.1.6 5.3.2

Ο προτεινόμενος οδηγός μπορεί να υπαχθεί στις κατηγορίες:

1. ΕΛ/ΛΑΚ για τις μικρομεσαίες επιχειρήσεις
2. ΕΛ/ΛΑΚ για την εκπαίδευση.

Προαπαιτούμενες γνώσεις:

Γενική γνώση του τρόπου λειτουργίας των Γεωγραφικών Συστημάτων Πληροφοριών σε ότι αφορά τα διανυσματικά δεδομένα. Γενική γνώση του τρόπου λειτουργίας των σχεσιακών βάσεων δεδομένων και γνώση σύνταξης βασικών ερωτημάτων σε γλώσσα SQL.

Για επισημάνσεις σφαλμάτων, διευκρινήσεις και προτάσεις για βελτίωση/επέκταση μπορείτε να επικοινωνήσετε στην διεύθυνση ηλ. ταχυδρομείου:
n.kolios@gmail.com

Πίνακας Περιεχομένων

1. Εισαγωγή.....	5
2. ΣΓΠ: Χωρικές Βάσεις Δεδομένων και Εφαρμογές Γραφείου.....	6
3. Επιμέρους στοιχεία του συστήματος	8
3.1 Το σύστημα διαχείρισης βάσεων δεδομένων PostgreSQL.....	8
3.2 Η επέκταση γεωγραφικών λειτουργιών PostGIS.....	9
3.3 Το ΣΓΠ Γραφείου QuantumGIS	10
4. Εγκατάσταση του συστήματος.....	11
4.1 ΧΒΔ PostgreSQL/PostGIS.....	11
4.2 Εγκατάσταση του λογισμικού ΓΣΠ QuantumGIS.....	22
5. Βασικές λειτουργίες.....	26
5.1. Λειτουργίες εισαγωγής και διαχείρισης δεδομένων.....	26
5.1.1 Δημιουργία βάσης δεδομένων και σύνδεση με αυτή.....	26
5.1.1.1 Δημιουργία βάσης δεδομένων με την χρήση προτύπου (template)	26
5.1.1.2 Δημιουργία βάσης δεδομένων χωρίς την χρήση προτύπου (template)	31
5.1.1.3 Σύνδεση με την βάση δεδομένων.....	32
5.1.2 Δημιουργία σχήματος.....	33
5.1.3 Δημιουργία πινάκων και εισαγωγή δεδομένων με εντολές SQL.....	34
5.1.4 Εισαγωγή δεδομένων από αρχείο *.shp.....	42
5.1.4.1 Εισαγωγή σε περιβάλλον command line	43
5.1.4.2 Εισαγωγή δεδομένων μέσω του προγράμματος QuantumGIS:.....	46
5.1.4.3 Απεικόνιση δεδομένων στο QuantumGIS.....	51
5.1.4.4 Εξαγωγή δεδομένων της βάσης δεδομένων σε μορφή *.shp	53
5.1.5 Εισαγωγή πινάκων με μη γεωμετρικά δεδομένα	55
5.1.6 Αλλαγή συστήματος συντεταγμένων	56
5.2 Λειτουργίες ανάκτησης δεδομένων.....	57
5.2.1 Μη χωρικά ερωτήματα SQL στο περιβάλλον της βάσης δεδομένων.....	58
5.2.2 Μη χωρικά ερωτήματα στο περιβάλλον του QuantumGIS.....	60
5.2.3 Χωρικά ερωτήματα.....	62
5.2.3.1 Οι τελεστές WITHIN και CONTAINS.....	62
5.2.3.2 Ο τελεστής DISTANCE.....	66
5.2.3.3 Ο τελεστής INTERSECTS.....	67
5.2.3.4 Ο τελεστής TOUCHES.....	69
5.2.3.5 Ο τελεστής AREA.....	71
5.2.3.6 Ο τελεστής ENVELOPE.....	71
5.2.3.7 Ο τελεστής LENGTH	72
5.2.3.8 Ο τελεστής NPOINTS	72
5.3 Λειτουργίες επέμβασης στα δεδομένα σε γραφικό περιβάλλον	73
5.3.1 Περιβάλλον QuantumGIS	73
5.3.1.1 Εισαγωγή στοιχείων	73
5.3.1.2 Διαγραφή στοιχείων.....	75
5.3.1.3 Μετακίνηση στοιχείου.....	75
5.3.1.4 Διάρθρωση στοιχείου.....	75

5.3.1.5 Προσθήκη κόμβου	75
5.3.1.6 Διαγραφή κόμβου.....	76
5.3.1.7 Επέμβαση στον πίνακα δεδομένων.....	76
5.3.2 Περιβάλλον UDIG.....	76
6. Προχωρημένες λειτουργίες.....	85
6.1 Η διαδικαστική γλώσσα PL/pgSQL.....	85
6.2 Σύνταξη και χρήση στοιχείων ελέγχου σε PL/pgSQL.....	86
6.2.1 Συνάρτηση.....	87
6.2.2 Σκανδαλιστής.....	90
7. Σύνοψη.....	92
8. Πηγές	93

1. Εισαγωγή

Στην εποχή μας η χρήση χωρικών πληροφοριών γίνεται ολοένα και περισσότερο μέρος της δραστηριότητας μιας μεγάλης γκάμας επιχειρήσεων, επαγγελματιών και ιδρυμάτων. Ο αυξανόμενος όγκος αυτών των χωρικών πληροφοριών οδήγησε στην ανάπτυξη συστημάτων που να είναι σε θέση να τις αποθηκεύσουν, διαχειριστούν και ανακτήσουν (να υποβάλουν ερωτήματα) με αποτελεσματικό τρόπο. Ανάμεσα σε αυτού του είδους τα συστήματα συγκαταλέγονται και οι Χωρικές Βάσεις Δεδομένων (ΧΒΔ). Ένα τέτοιο σύστημα είναι η χωρική βάση δεδομένων που αποτελείται από την βάση δεδομένων PostgreSQL και την επέκταση χωρικών λειτουργιών PostGIS. Τα λογισμικά αυτά τείνουν να χρησιμοποιούνται μαζί με ένα Σύστημα Γεωγραφικών Πληροφοριών Γραφείου, συνήθως το QuantumGIS, προκειμένου να επιτρέπεται η οπτικοποίηση των δεδομένων και η επέμβαση σε αυτά σε γραφικό περιβάλλον. Συνολικά το σύστημα που αποτελείται από τα παραπάνω λογισμικά επιτρέπει μία πλήρη γκάμα λειτουργιών αποθήκευσης, διαχείρισης και ανάκτησης χωρικών δεδομένων.

Ο παρών οδηγός έχει σαν σκοπό να εξοικειώσει τον χρήστη με τις βασικές λειτουργίες του παραπάνω συνδυασμού λογισμικών. Αυτό θα γίνεται με την βήμα προς βήμα ανάπτυξη της λειτουργίας των επιμέρους στοιχείων και την παράθεση παραδειγμάτων λειτουργίας. Αρχικά περιγράφεται σύντομα ο τρόπος λειτουργίας των Χωρικών Βάσεων Δεδομένων και των Συστημάτων Γεωγραφικών Πληροφοριών γραφείου. Έπειτα παρατίθενται πληροφορίες για τα λογισμικά PostgreSQL, PostGIS, QuantumGIS και περιγράφεται ο τρόπος εγκατάστασής τους. Στην συνέχεια περιγράφονται οι βασικές λειτουργίες εισαγωγής, διαχείρισης και ανάκτησης δεδομένων με έμφαση στην χρήση χωρικών τελεστών. Στο τελευταίο μέρος του οδηγού γίνεται αναφορά στην χρήση διαδικαστικών γλωσσών στις χωρικές βάσεις δεδομένων και συγκεκριμένα στην γλώσσα PL/pgSQL που χρησιμοποιείται στην PostgreSQL. Δίνονται δύο απλά παραδείγματα δημιουργίας στοιχείων ελέγχου (συνάρτηση, σκανδαλιστής) με την χρήση της γλώσσας PL/pgSQL.

Ο οδηγός υποστηρίζεται από εκπαιδευτικά δεδομένα που είναι διαθέσιμα στην τοποθεσία: http://www.ellak.gr/workshop_postgis.zip. Όλες οι εντολές command line, SQL και PL/pgSQL που περιλαμβάνονται στον παρόντα οδηγό βρίσκονται στο αρχείο workshop.txt ενώ τα αρχεία shapefiles (*.shp) που χρησιμοποιούνται για εισαγωγή δεδομένων βρίσκονται στον φάκελο SHP.

2. ΣΓΠ: Χωρικές Βάσεις Δεδομένων και Εφαρμογές Γραφείου.

Η αποθήκευση, διαχείριση και ανάκτηση (υποβολή ερωτημάτων) σε δεδομένα που απεικονίζουν χωρικές οντότητες (διακριτά χωρικά δεδομένα: αγροτεμάχια, κτίσματα, δρόμοι, δίκτυα, κ.α. - συνεχή χωρικά δεδομένα: υψόμετρο, θερμοκρασία κ.α) αποτελεί μέρος της δραστηριότητας μια μεγάλης γκάμας οργανισμών όπως επιχειρήσεις και επαγγελματίες, δημόσιοι φορείς και ερευνητικά ιδρύματα. Παραδείγματα αποτελούν οι τομείς του real estate, της εκπόνησης περιβαλλοντικών, δημογραφικών και άλλων τεχνικών μελετών και συναφούς έρευνας, των τεχνικών έργων, της συλλογής και διανομής χωρικών δεδομένων, της παραγωγής χαρτών και διαγραμμάτων, της πλοήγησης και διαχείρισης στόλου, της μετεωρολογίας, γεωλογίας, γεωγραφίας, της δημόσιας διοίκησης κ.α. Σε αυτό το επίπεδο η κλιμακούμενη ανάγκη για αποτελεσματική διαχείριση των χωρικών δεδομένων έχει οδηγήσει στην ανάπτυξη των Συστημάτων Γεωγραφικών Πληροφοριών (ΣΓΠ – GIS Geographical Information Systems).

Τα ΣΓΠ περιλαμβάνουν όλα εκείνα τα συστήματα που καταγράφουν, αποθηκεύουν, αναλύουν, διαχειρίζονται και οπτικοποιούν ψηφιακά χωρικά δεδομένα. Για λόγους ταξινόμησης διακρίνουμε αυτή τη μεγάλη ομάδα λογισμικών και εργαλείων που διαχειρίζονται χωρικά δεδομένα σε ΣΓΠ γραφείου (Desktop GIS) και Χωρικές Βάσεις Δεδομένων (ΧΒΔ - Spatial Databases, Geodatabases).

Τα ΣΓΠ γραφείου είναι εργαλεία που επιτρέπουν στους χρήστες την ερώτηση, αναζήτηση, ανάλυση και τροποποίηση χωρικών δεδομένων καθώς και την δημιουργία χαρτών από αυτά. Συχνά, αν και όχι πάντα, τα ΣΓΠ γραφείου περιλαμβάνουν εφαρμογές τηλεπισκόπησης, τοπογραφίας, δορυφορικών εικόνων και αεροφωτογραφίας, μαθηματικών, γεωγραφίας, χωρικής ανάλυσης φωτογραμμετρίας κ.α. Αυτά τα εργαλεία είναι συνήθως αυτοτελείς και εύχρηστες εφαρμογές που εγκαθίστανται εύκολα, και προσφέρουν την δυνατότητα επέμβασης στα δεδομένα σε γραφικό περιβάλλον.

Οι ΧΒΔ είναι εργαλεία που αποτελούνται από περισσότερα του ενός επιμέρους λογισμικά, συνήθως ένα λογισμικό βάσης δεδομένων, μία επέκταση χωρικών λειτουργιών που εξειδικεύεται στην αποθήκευση γεωμετρικών πληροφοριών και εκτέλεση γεωμετρικών υπολογισμών και ερωτημάτων και ένα λογισμικό οπτικοποίησης. Οι ΧΒΔ όπως και οι απλές βάσεις δεδομένων παρέχουν την δυνατότητα αποτελεσματικής διαχείρισης μεγάλων όγκων δεδομένων. Επιπλέον έχουν την δυνατότητα να κάνουν τα δεδομένα τους διαθέσιμα ταυτόχρονα σε περισσότερους του ενός χρήστες. Παρέχουν την δυνατότητα για εκτέλεση ευέλικτων και πολύπλοκων ερωτημάτων με χρήση της γλώσσας SQL (Structured Query Language) και συχνά διαδικαστικών προγραμματιστικών γλωσσών με ακόμη περισσότερες δυνατότητες.

Παρά τα ιδιαίτερα χαρακτηριστικά των ΣΓΠ γραφείου και των ΧΒΔ ο διαχωρισμός ανάμεσα τους, ιδιαίτερα στο επίπεδο λειτουργίας, δεν είναι εντελώς σαφής. Όλο και περισσότερο τα λογισμικά ΣΓΠ

γραφείου και οι ΧΒΔ επεκτείνουν τις δυνατότητες συμβατότητας μεταξύ τους. Συχνά τα ΓΣΠ γραφείου παίζουν το ρόλο του προγράμματος οπτικοποίησης σε μία χωρική βάση δεδομένων. Άλλες φορές τα ΣΓΠ γραφείου αναπτύσσουν λειτουργίες πέραν των απλής οπτικοποίησης επιτρέποντας την επέμβαση, μετατροπή, εισαγωγή και τροποποίηση στα περιεχόμενα της ΧΒΔ στο δικό τους γραφικό περιβάλλον. Σε κάποιες περιπτώσεις οι ΧΒΔ αναλαμβάνουν το ρόλο της διαχείρισης απαιτητικών δεδομένων ή δεδομένων που πρέπει να είναι προσβάσιμα σε πολλαπλούς χρήστες για λογαριασμό μίας εφαρμογής ΣΓΠ γραφείου.

Στον παρόντα οδηγό θα αναπτύξουμε βήμα βήμα την εγκατάσταση και λειτουργία της χωρικής βάσης δεδομένων PostgreSQL/PostGIS σε ταυτόχρονη λειτουργία με το ΣΓΠ γραφείου QuantumGIS.

3. Επιμέρους στοιχεία του συστήματος

3.1 Το σύστημα διαχείρισης βάσεων δεδομένων PostgreSQL

Η PostgreSQL είναι ένα αντικείμενο-σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (object-relational database management system - ORDBMS). Είναι ελεύθερο και ανοικτού κώδικα λογισμικό και δεν αναπτύσσεται από μία μόνο εταιρία αλλά από μία παγκόσμια κοινότητα χρηστών, εταιριών και ιδρυμάτων. Η PostgreSQL υποστηρίζει λειτουργίες όπως συναρτήσεις, δείκτες (B+-tree, hash, GiST και GiN), σκανδαλιστές, κανόνες, ένα ευρύ φάσμα από προκαθορισμένους και ορισμένους από τον χρήστη τύπους δεδομένων και αντικείμενα. Επιπλέον υποστηρίζονται λειτουργίες κληρονομικότητας χαρακτηριστικών πινάκων, περιορισμοί, όψεις, συναλλαγές, λειτουργίες κρυπτογράφησης, αποθήκευσης μεγάλων αντικειμένων κ.α.

Όπως οι περισσότερες βάσεις δεδομένων ή PostgreSQL υποστηρίζει την χρήση της SQL για την σύνταξη ερωτημάτων. Οι περιορισμένες δυνατότητες που παρέχει η γλώσσα SQL σε βασικά στοιχεία ελέγχου έχει οδηγήσει στην ενσωμάτωση ποιο σύνθετων γλωσσών. Τέτοιες είναι:

- Η διαδικαστική γλώσσα PL/pgSQL που είναι εξ' ορισμού ενσωματωμένη στην PostgreSQL και είναι αντίστοιχη της γλώσσας PL/SQL που χρησιμοποιείται στο εμπορικό σύστημα διαχείρισης βάσεων δεδομένων Oracle
- Οι γλώσσες (scripting languages) PL/Lua, PL/LOLCODE, PL/Perl, plPHP, PL/Python, PL/Ruby, PL/sh, PL/Tcl, PL/Scheme, PL/Java, PL/R
- Οι προγραμματιστικές γλώσσες C και C++.

Η PostgreSQL ακόμα και χωρίς κάποια επέκταση υποστηρίζει κάποιες βασικές χωρικές λειτουργίες όπως ειδικούς τύπους δεδομένων για την αποθήκευση απλών γεωμετρικών οντοτήτων και μηχανισμούς δεικτοδότησης χωρικών δεδομένων όπως τετραδικά και R-δένδρα. Η δεικτοδότηση με R-δένδρα είναι μία από τις πιο αποτελεσματικές μεθόδους δεικτοδότησης σε χωρικά δεδομένα. Επιπλέον υποστηρίζεται περιορισμένος αριθμός γεωμετρικών τελεστών και συναρτήσεων για την διατύπωση χωρικών ή συνδυασμένων ερωτημάτων. Στους περιορισμούς που αντιμετωπίζει στον χειρισμό χωρικών δεδομένων θα πρέπει να συγκαταλεχθεί το ότι επιτρέπει μόνο διδιάστατες γεωμετρικές οντότητες χωρίς δυνατότητα προσδιορισμού συστήματος αναφοράς.

Η PostgreSQL υποστηρίζει τη συνεργασία με πληθώρα επιπρόσθετων σπονδύλων λογισμικού (add-ons) κάθε ένα από τα οποία επιτρέπει την εκτέλεση συγκεκριμένων και εξειδικευμένων λειτουργιών. Το πιο ευρύτερα διαδεδομένο από αυτά τα λογισμικά είναι η επέκταση γεωγραφικών λειτουργιών

PostGIS.

3.2 Η επέκταση γεωγραφικών λειτουργιών PostGIS

Η PostGIS είναι επίσης λογισμικό ανοικτού κώδικα και ακολουθεί το πρότυπο OGC για τον ορισμό γεωγραφικών στοιχείων σε περιβάλλον SQL. Συγκεκριμένα η PostGIS υποστηρίζει:

Τους ακόλουθους τύπους στοιχείων:

- ο Σημεία (points)
- ο Γραμμές (linestrings)
- ο Πολύγωνα (polygons)
- ο Πολυσημεία (multipoints)
- ο Πολυγραμμές (multilinestrings)
- ο Πολύ-πολύγωνα (multipolygons)

Συλλογές Γεωμετρικών στοιχείων (Geometrycollections).

Χωρικά κατηγορήματα (spatial predicates) δηλαδή συνθήκες για τον προσδιορισμό των αλληλεπιδράσεων ανάμεσα σε γεωμετρικά στοιχεία με χρήση του 3x3 πίνακα Egenhofer.

Χωρικούς τελεστές όπως area, distance, length, perimeter, χωρικές λειτουργίες union, difference, symmetric difference και buffers.

Χωρικούς δείκτες R-δένδρα και γενικευμένα δένδρα αναζήτησης (GIST).

Η αρχιτεκτονική της PostGIS στοχεύει στην ελαχιστοποίηση απαιτούμενων πόρων υπολογιστικής ισχύος και μνήμης. Η χρήση γεωμετρικών στοιχείων με χαμηλές απαιτήσεις φυσικής μνήμης επιτρέπει την διατήρηση μεγάλου όγκου δεδομένων από την φυσική μνήμη στην υπολογιστική μνήμη (RAM) με αποτέλεσμα την ταχύτερη εκτέλεση των ερωτημάτων.

3.3 Το ΣΓΠ Γραφείου QuantumGIS

Το QuantumGIS (συχνά γίνεται αναφορά σε αυτό ως QGIS) είναι ένα ελεύθερο και ανοικτού κώδικα ΣΓΠ γραφείου που προσφέρει δυνατότητες οπτικοποίησης, τροποποίησης και ανάλυσης χωρικών δεδομένων. Η ανάπτυξη του λογισμικού ξεκίνησε από τον Gary Sherman το 2002 και σύντομα γύρω από αυτό αναπτύχθηκε το Open Source Geospatial Foundation. Το πρόγραμμα συντηρείται από μία ενεργή κοινότητα χρηστών και προγραμματιστών που σε τακτά διαστήματα κυκλοφορούν νέες εκδόσεις επεκτείνοντας τις δυνατότητες των παλιών και διορθώνοντας σφάλματα. Το QuantumGIS έχει γραφτεί σε C++ και η γραφική διεπαφή χρησιμοποιεί την βιβλιοθήκη Qt. Πολλά από τα πρόσθετα που χρησιμοποιούνται έχουν γραφτεί επίσης σε C++ ή Python. Έχει μεταφραστεί σε 14 γλώσσες και χρησιμοποιείται ευρύτατα τόσο από ερευνητικούς/ακαδημαϊκούς φορείς όσο και στον επιχειρηματικό χώρο.

Το QuantumGIS είναι συμβατό με λειτουργικά συστήματα Linux, Unix, Windows και Macintosh. Ακόμα το συγκεκριμένο ΣΓΠ γραφείου μπορεί να χρησιμοποιηθεί σαν περιβάλλον γραφικής διεπαφής τόσο για χωρικές βάσεις δεδομένων όπως η PostGIS όσο και για λιγότερο εύχρηστα ΣΓΠ όπως το GRASS. Επιπλέον έχει μικρότερες απαιτήσεις υπολογιστικής δύναμης σε σχέση με τα περισσότερα εμπορικά ΣΓΠ. Έτσι μπορεί να χρησιμοποιηθεί εύκολα ταυτόχρονα με άλλες εφαρμογές ή σε μονάδες με περιορισμένους πόρους.

Οι πρώτη έκδοση του προγράμματος που κυκλοφόρησε τον Ιούλιο του 2002 ανέπτυξε κυρίως λειτουργίες απεικόνισης και εισαγωγής δεδομένων στην χωρική βάση δεδομένων PostgreSQL/PostGIS. Σταδιακά ενσωματώθηκαν λειτουργίες υποστήριξης αρχείων shapefiles όσο και άλλων διανυσματικών format. Στην σημερινή του μορφή το πρόγραμμα προσφέρει αυξημένες δυνατότητες οπτικοποίησης, διαχειρίζεται raster αρχεία, μπορεί να εκτελεί λειτουργίες ανάλυσης (λ.χ buffer) ενώ διακρίνεται για την αξιοπιστία και σταθερότητά του. Είναι συμβατό με τους περισσότερους εμπορικούς τύπους αρχείων, την διαδικτυακή εφαρμογή χωρικών δεδομένων MapServer καθώς και τα πρότυπα WMS και WFS του Open Geospatial Consortium. Σαν λογισμικό ανοιχτού κώδικα μπορεί να τροποποιηθεί έτσι ώστε να εκτελεί διαφορετικές ή εξειδικευμένες λειτουργίες.

4. Εγκατάσταση του συστήματος

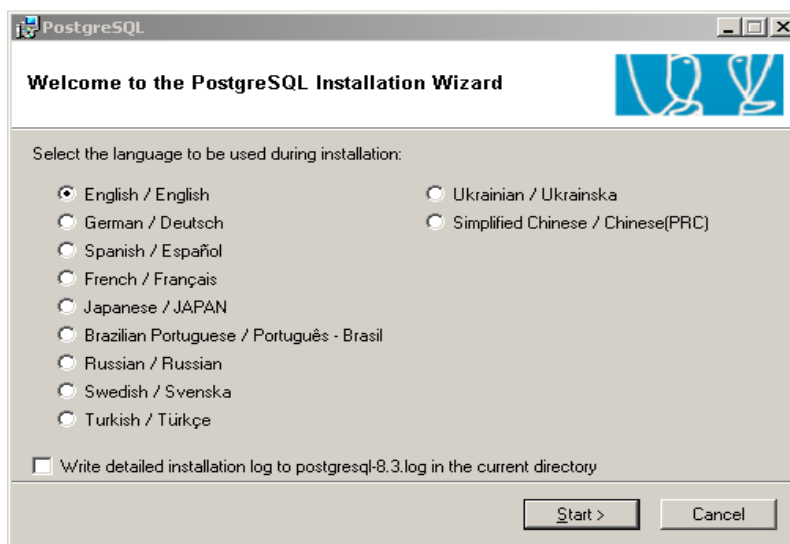
4.1 ΧΒΔ PostgreSQL/PostGIS

Αρχικά κατεβάζουμε την πιο πρόσφατη έκδοση του συμπιεσμένου πακέτου εγκατάστασης της βάσης δεδομένων PostgreSQL από την τοποθεσία <http://www.postgresql.org/download/> . Φροντίζουμε να επιλέξουμε την έκδοση που είναι συμβατή με το λειτουργικό σύστημα του υπολογιστή μας. Επιπλέον αποφεύγουμε την δοκιμαστική έκδοση (beta). Την στιγμή που γράφτηκε ο παρών οδηγός η τελευταία σταθερή έκδοση του λογισμικού ήταν η 8.3.7.1.

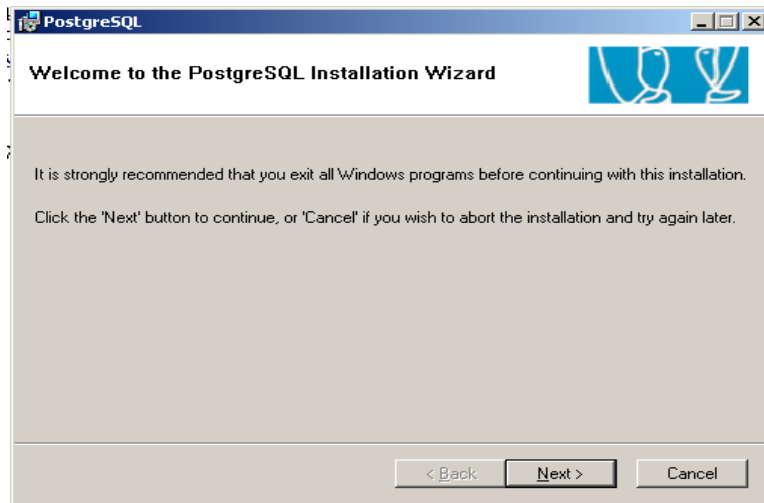
(<http://wwwmaster.postgresql.org/redir/57/h/binary/v8.3.7/win32/postgresql-8.3.7-1.zip>)

Αφού αποθηκεύσουμε το συμπιεσμένο αρχείο στο σκληρό δίσκο μας το αποσυμπιέζουμε και από τα αρχεία που περιέχει εκτελούμε το postgresql-8.3.msi. Ακολουθεί η διαδικασία εγκατάστασης σε βήματα:

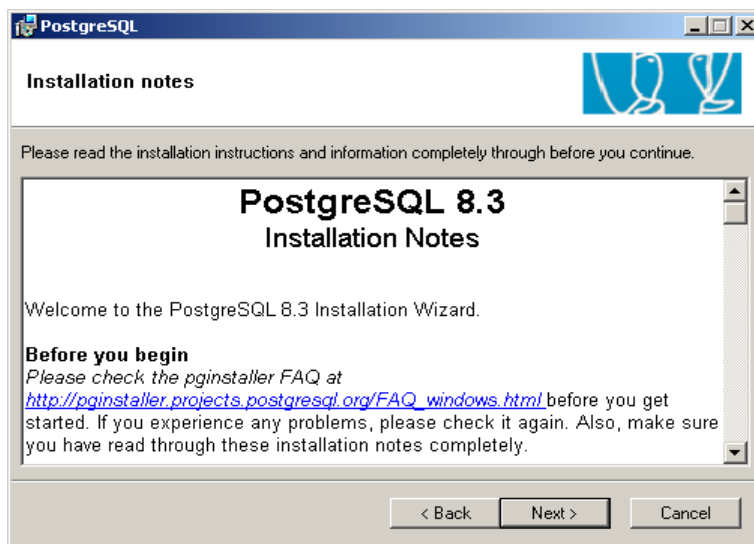
1. Αρχικά επιλέγουμε την γλώσσα που θα χρησιμοποιηθεί κατά την εγκατάσταση και επιλέγουμε start:



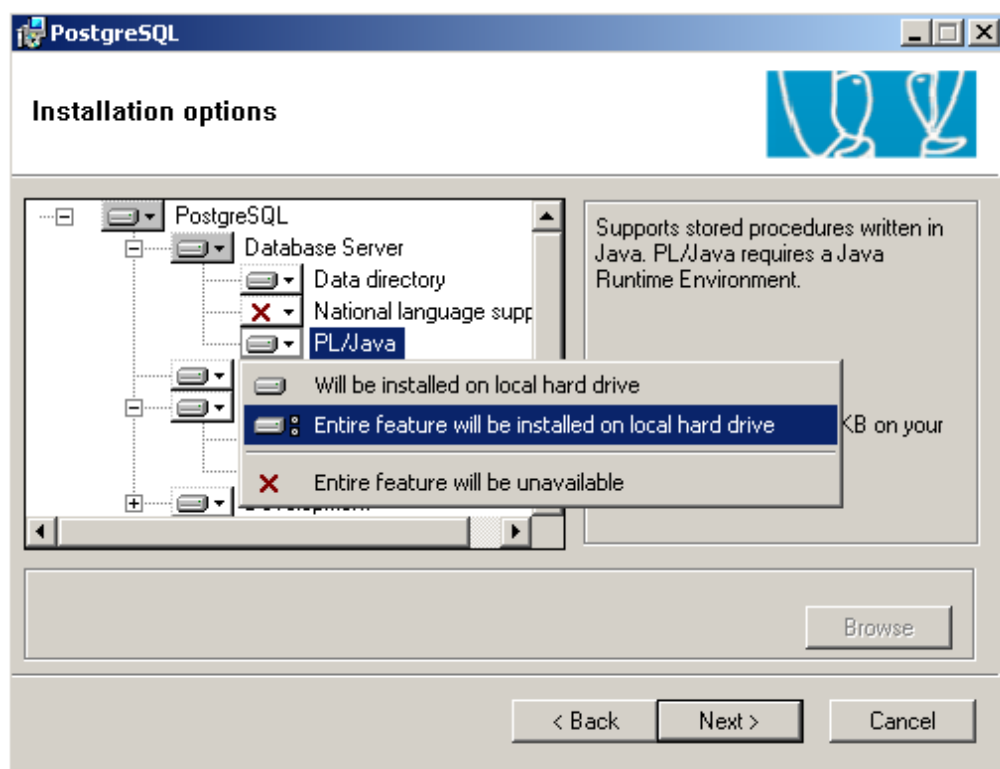
2. Φροντίζουμε ώστε να μην τρέχουν άλλες εφαρμογές κατά την διάρκεια της εγκατάστασης και επιλέγουμε Next



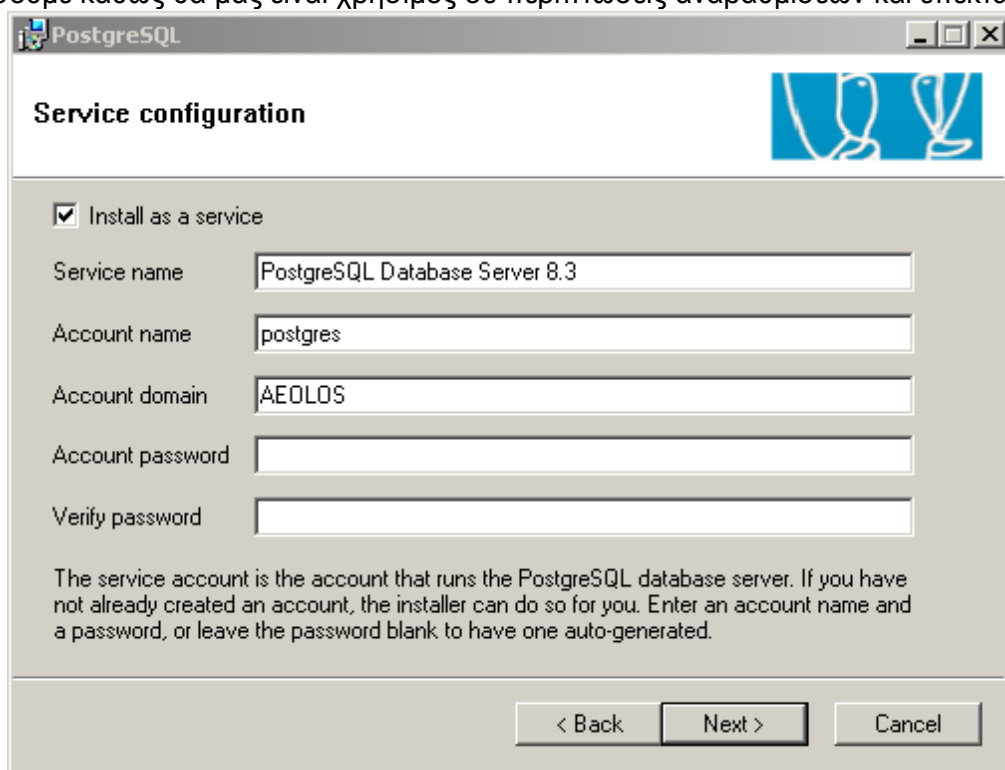
3. Επιλέγουμε “next”



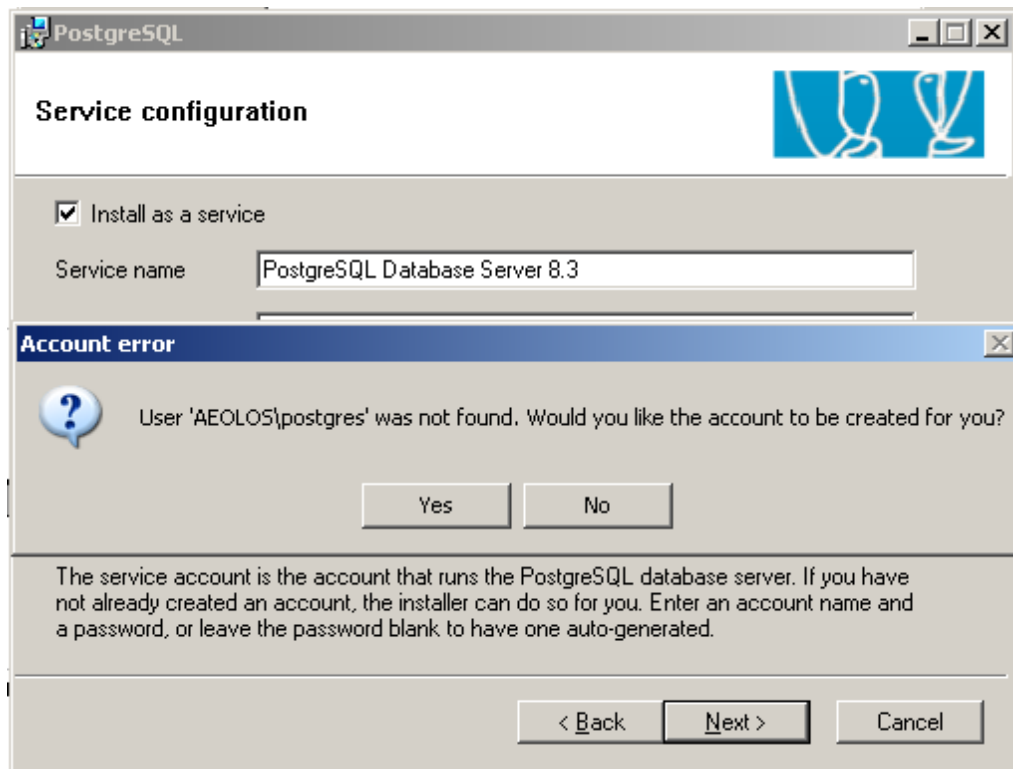
4. Επιλέγουμε “Entire Feature will be installed on local hard drive” για τα στοιχεία “PL/Java” και “Development” και προαιρετικά για το στοιχείο “National language support”. Αυτά τα στοιχεία είναι χρήσιμο να περιλαμβάνονται στην εγκατάσταση μας, ωστόσο δεν θα χρησιμοποιηθούν σε αυτό τον οδηγό.



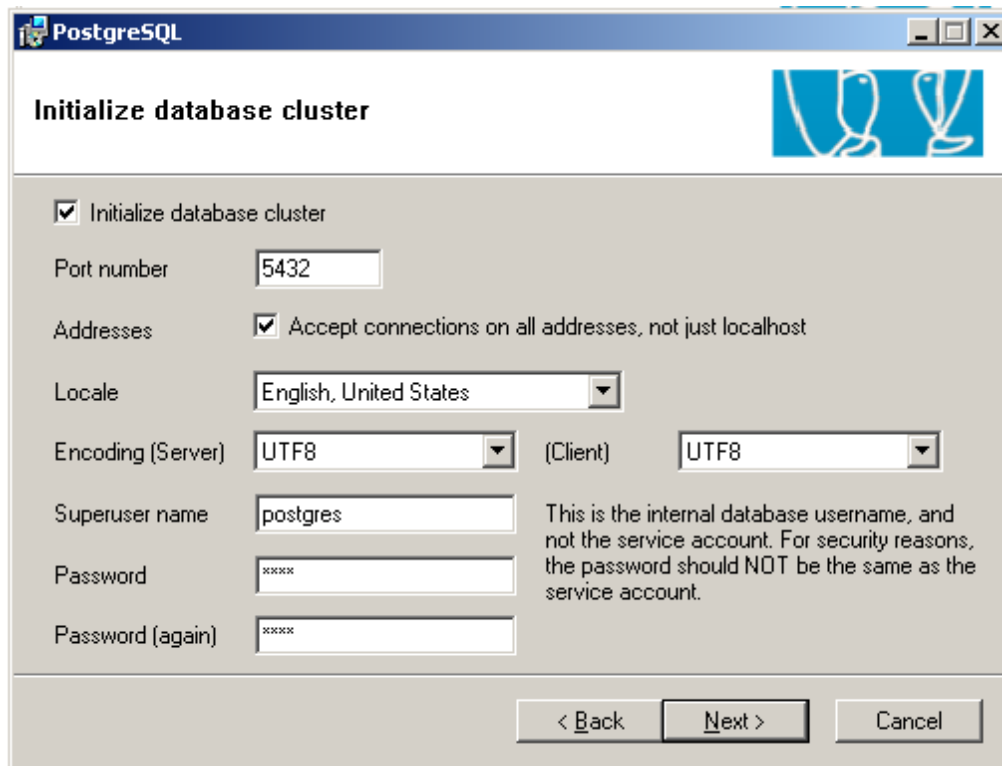
5. Στο επόμενο παράθυρο δεν συμπληρώνουμε κάτι. Τα πεδία “Service Name” “Account Name” “Account Domain” πρέπει να έχουν συμπληρωθεί αυτόματα. Αφήνουμε τα πεδία του κωδικού κενά έτσι ώστε το πρόγραμμα να επιλέξει από μόνο του έναν κωδικό που να πληρεί τα απαιτούμενα κριτήρια ασφάλειας. Το πρόγραμμα στην συνέχεια μας ενημερώνει για τον κωδικό που επέλεξε. Φροντίζουμε να τον σημειώσουμε καθώς θα μας είναι χρήσιμος σε περιπτώσεις αναβαθμίσεων και επεκτάσεων.



6. Επιλέγουμε “yes” στο επόμενο παράθυρο



7. Στο επόμενο παράθυρο επιλέγουμε το πεδίο “Accept connections on all addresses, not just localhost”. Αυτό θα είναι χρήσιμο σε περίπτωση που η βάση δεδομένων βρίσκεται εγκατεστημένη σε διαφορετική τοπική μονάδα από αυτή του χρήστη. Ωστόσο δεν είναι απαραίτητο για τα παραδείγματα που περιλαμβάνονται στον παρόντα οδηγό. Τροποποιούμε τις τιμές στα πεδία “Encoding (Server)” και “Encoding (Client)” σε 'ISO_8859_7' ή 'UTF8'. Ακόμα επιλέγουμε κάποιον κωδικό που να μπορούμε να θυμόμαστε και να πληκτρολογούμε σχετικά εύκολα. Δεν πρόκειται για τον ίδιο κωδικό με αυτόν που επέλεξε για εμάς το πρόγραμμα μετά το βήμα 5.



The image shows the 'PostgreSQL Initialize database cluster' window. It has a title bar with the PostgreSQL logo and standard window controls. The main area contains several configuration options: a checked checkbox for 'Initialize database cluster', a 'Port number' field set to '5432', an 'Addresses' section with a checked checkbox for 'Accept connections on all addresses, not just localhost', a 'Locale' dropdown set to 'English, United States', 'Encoding (Server)' and '(Client)' dropdowns both set to 'UTF8', a 'Superuser name' field set to 'postgres', and 'Password' and 'Password (again)' fields both containing 'xxxx'. A note on the right states: 'This is the internal database username, and not the service account. For security reasons, the password should NOT be the same as the service account.' At the bottom are '< Back', 'Next >', and 'Cancel' buttons.

PostgreSQL

Initialize database cluster

☒ Initialize database cluster

Port number: 5432

Addresses: ☒ Accept connections on all addresses, not just localhost

Locale: English, United States

Encoding (Server): UTF8 (Client): UTF8

Superuser name: postgres

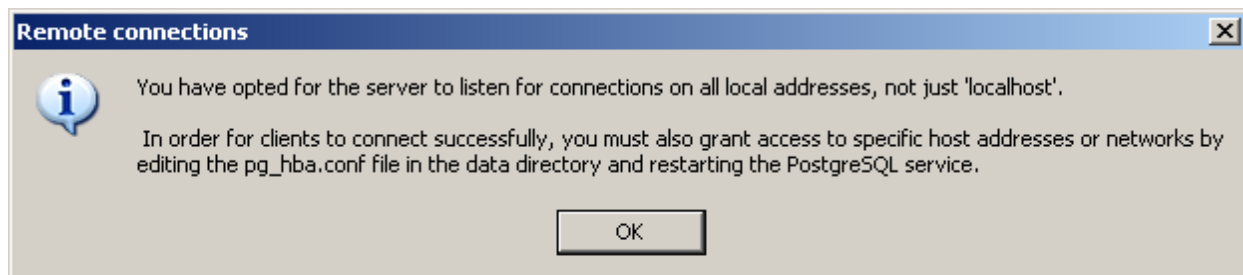
Password: xxxx

Password (again): xxxx

This is the internal database username, and not the service account. For security reasons, the password should NOT be the same as the service account.

< Back Next > Cancel

8. Στην συνέχεια πληροφορούμαστε ότι σε περίπτωση που επιθυμούμε η βάση δεδομένων βρίσκεται εγκατεστημένη σε διαφορετική τοπική μονάδα από αυτή του χρήστη τότε πρέπει να τροποποιήσουμε το αρχείο `pg_hba.conf`. Αυτό για την ώρα δεν είναι απαραίτητο.



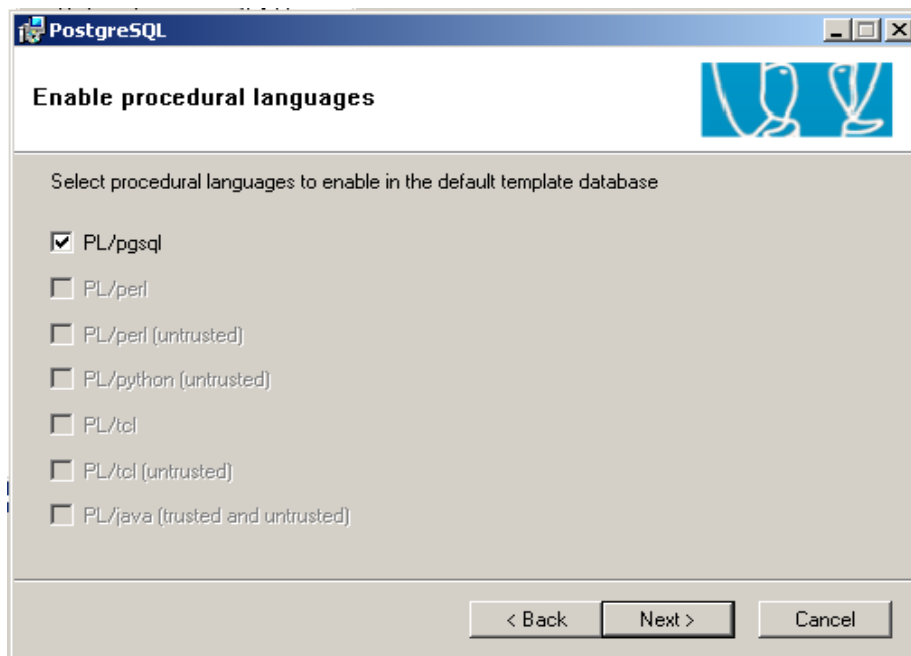
The image shows a 'Remote connections' information dialog box. It has a title bar with the text 'Remote connections' and a close button. The main area contains an information icon (i) and text: 'You have opted for the server to listen for connections on all local addresses, not just 'localhost'. In order for clients to connect successfully, you must also grant access to specific host addresses or networks by editing the pg_hba.conf file in the data directory and restarting the PostgreSQL service.' At the bottom is an 'OK' button.

Remote connections

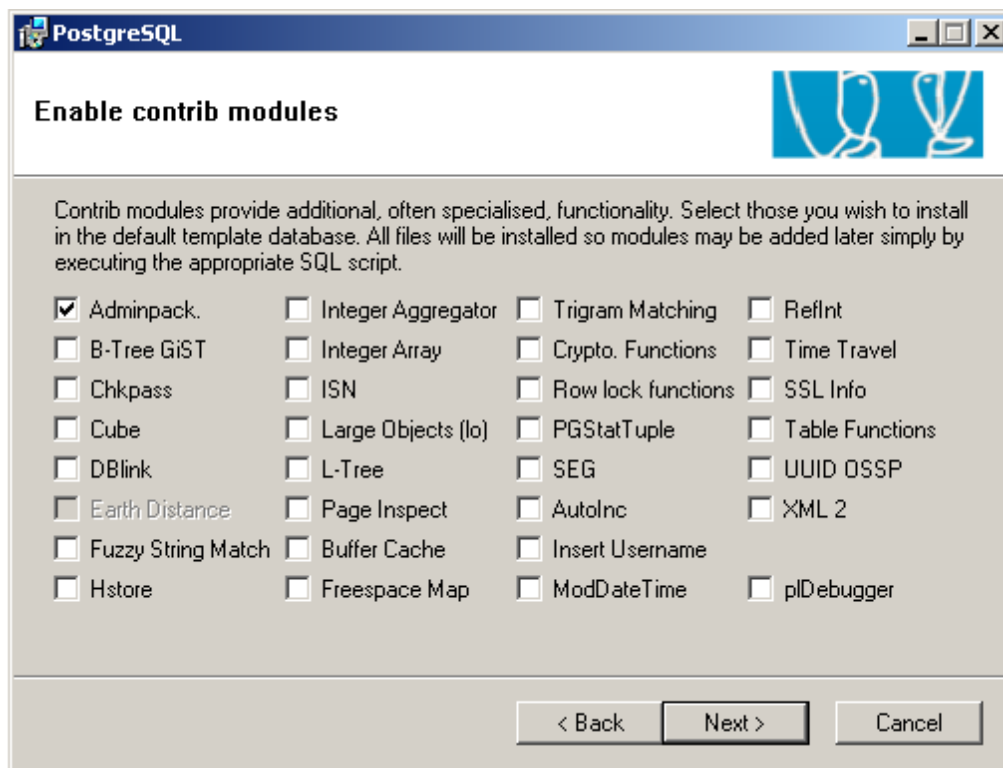
 You have opted for the server to listen for connections on all local addresses, not just 'localhost'. In order for clients to connect successfully, you must also grant access to specific host addresses or networks by editing the pg_hba.conf file in the data directory and restarting the PostgreSQL service.

OK

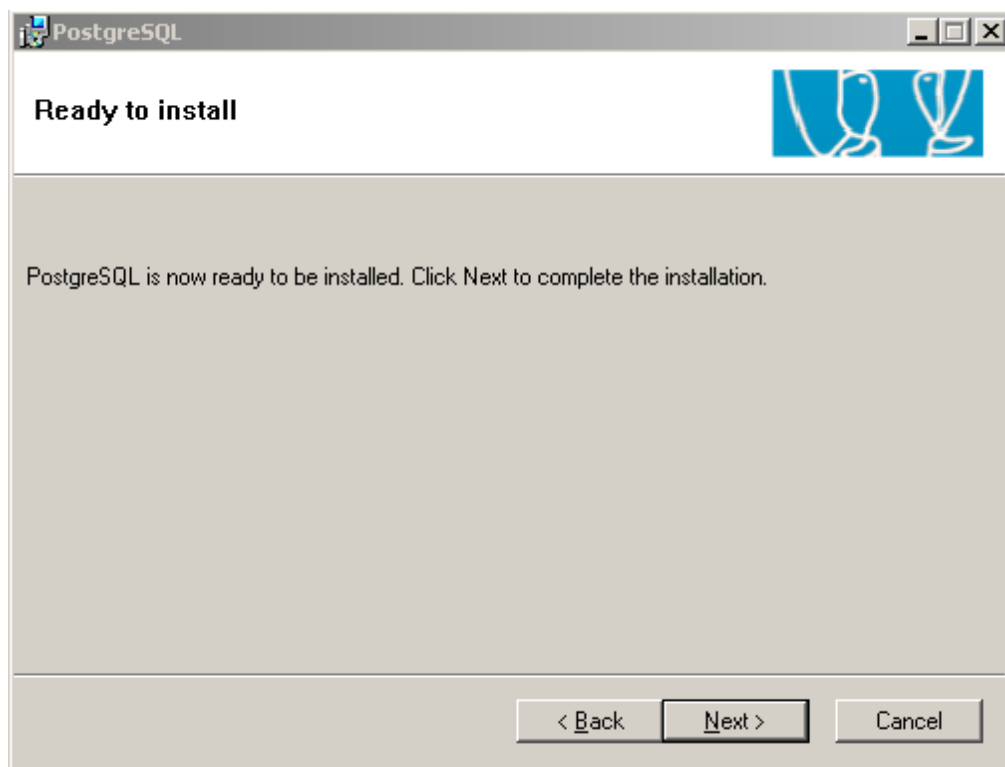
9. Στο επόμενο παράθυρο διατηρούμε επιλεγμένη την γλώσσα PL/pgSQL.



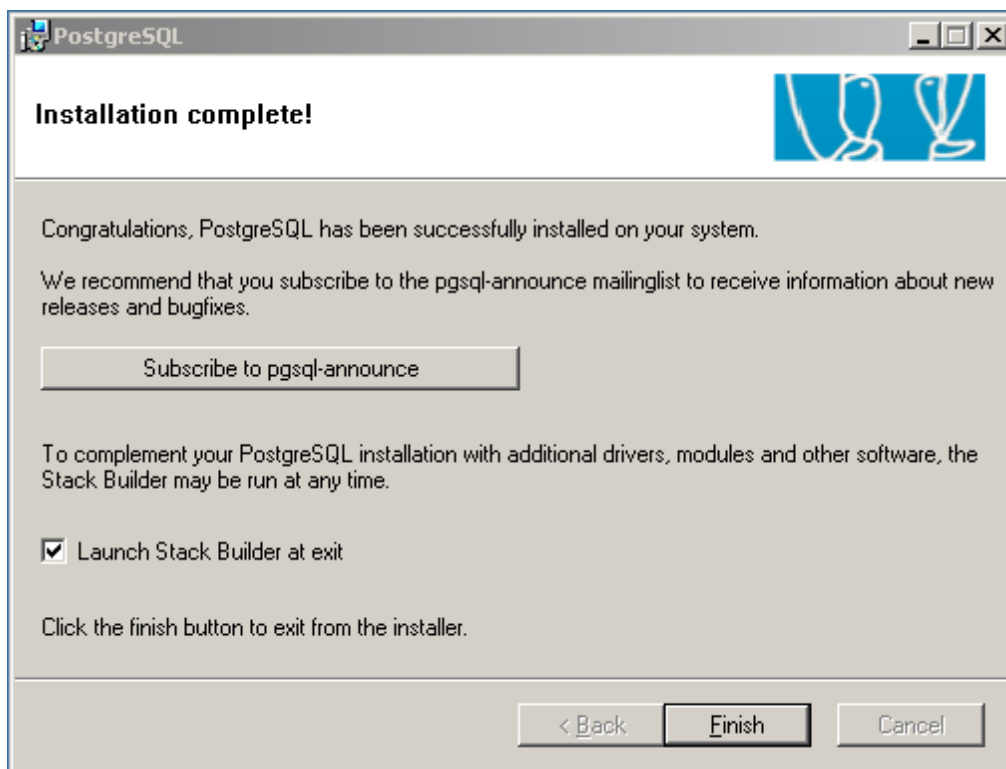
10. Στο επόμενο παράθυρο μπορούμε να επιλέξουμε κάποια εξειδικευμένη λειτουργία της βάσης δεδομένων. Οι λειτουργίες “B-Tree GiST” “Dblink” “Integer Aggregator” “L-Tree” “PGStatTuple” “ModDateTime” “TableFunctions” είναι χρήσιμες και καλό είναι να επιλεγούν.



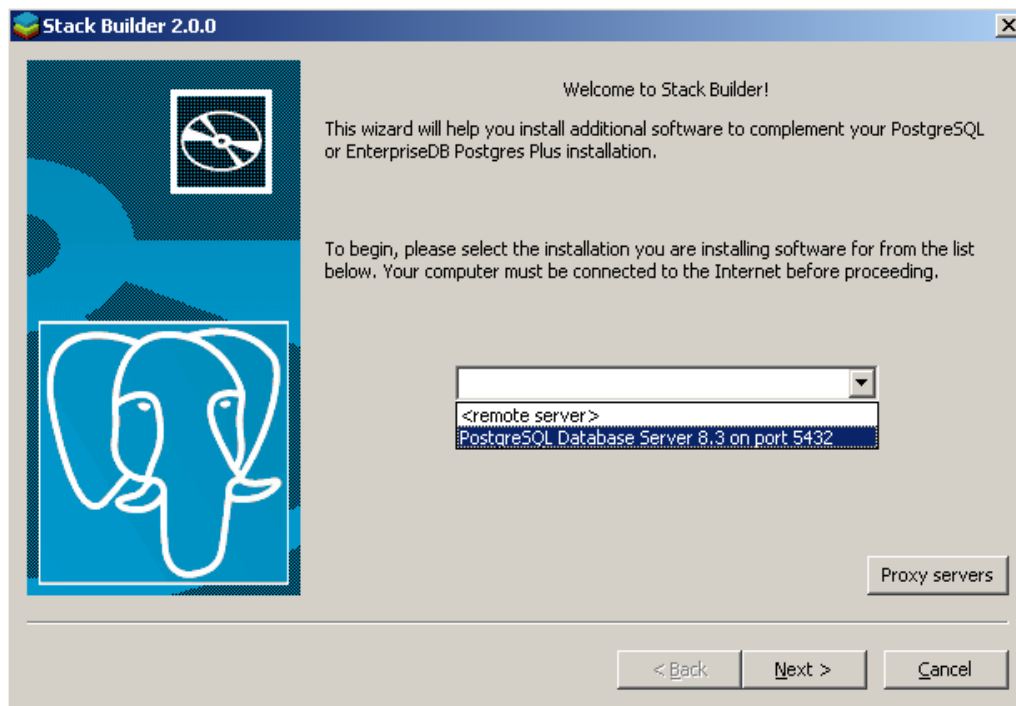
11. Επιλέγουμε “Next” και η εγκατάσταση ξεκινά.



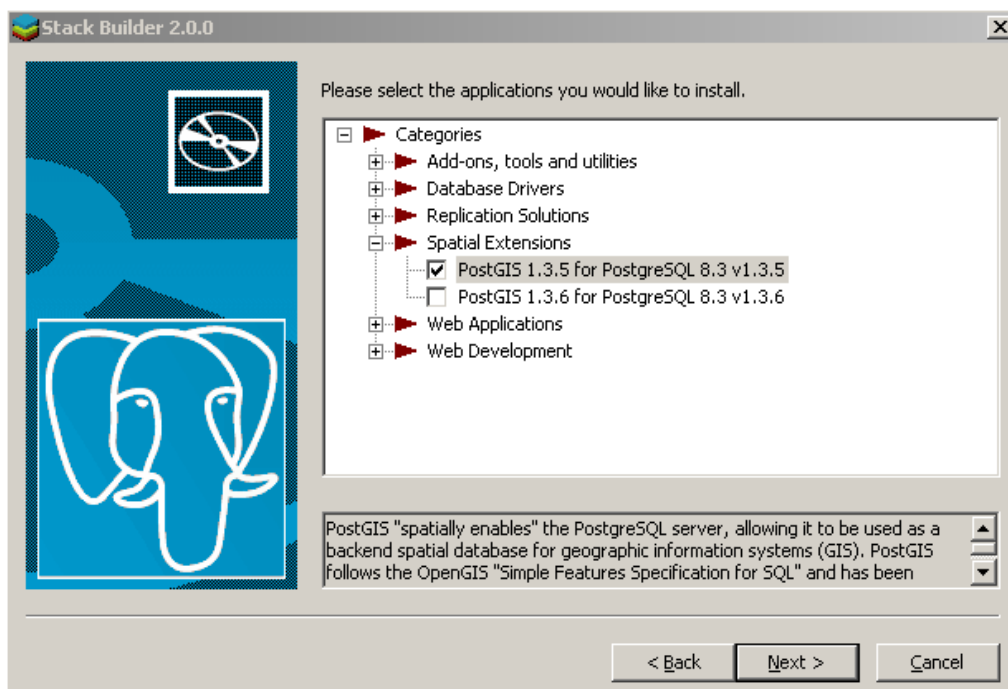
12. Όταν η εγκατάσταση έχει πλέον ολοκληρωθεί επιλέγουμε “Finish” φροντίζοντας να είναι επιλεγμένο το πεδίο “Launch Stack Builder at exit”.



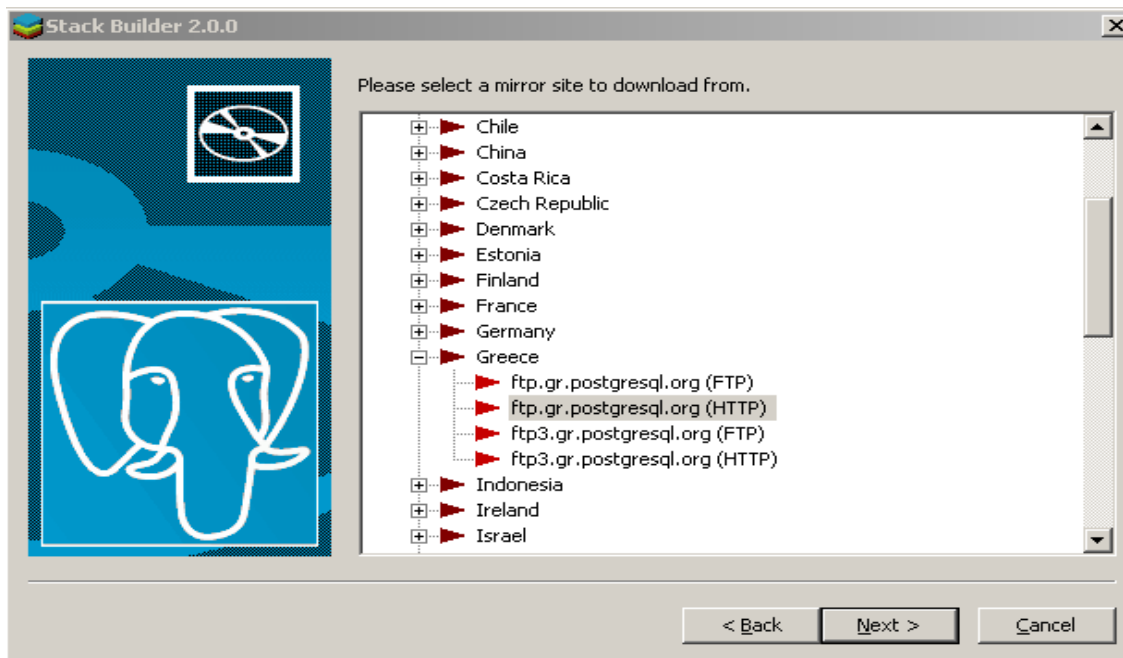
13. Στο παράθυρο του Stack Builder επιλέγουμε "PostgreSQL Database Server 8.3 on Port 5432" και στην συνέχεια "Next"



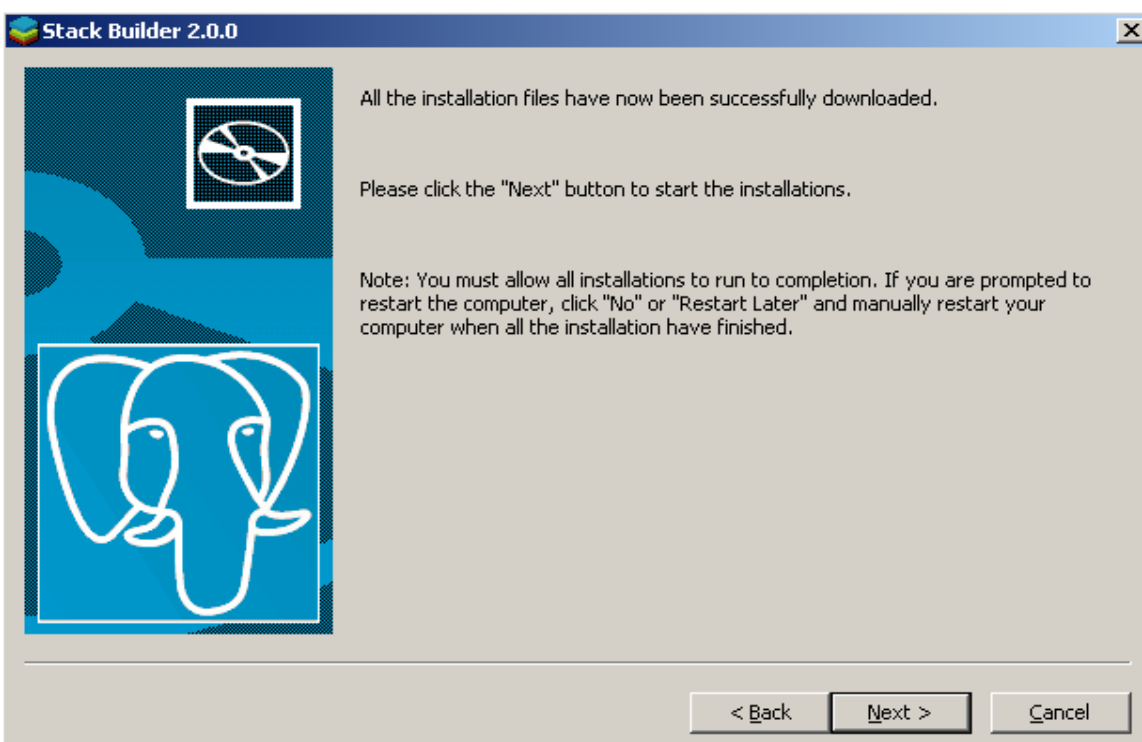
14. Στο επόμενο παράθυρο επιλέγουμε μία έκδοση της επέκτασης PostGIS. Είναι χρήσιμο να ελέγξουμε στην σελίδα <http://postgis.refractory.net/> ότι πρόκειται για σταθερή (stable) και όχι υπό ανάπτυξη (beta) έκδοση καθώς η τελευταία μπορεί να περιέχει κάποια σφάλματα που δεν έχουν ακόμα επιλυθεί.



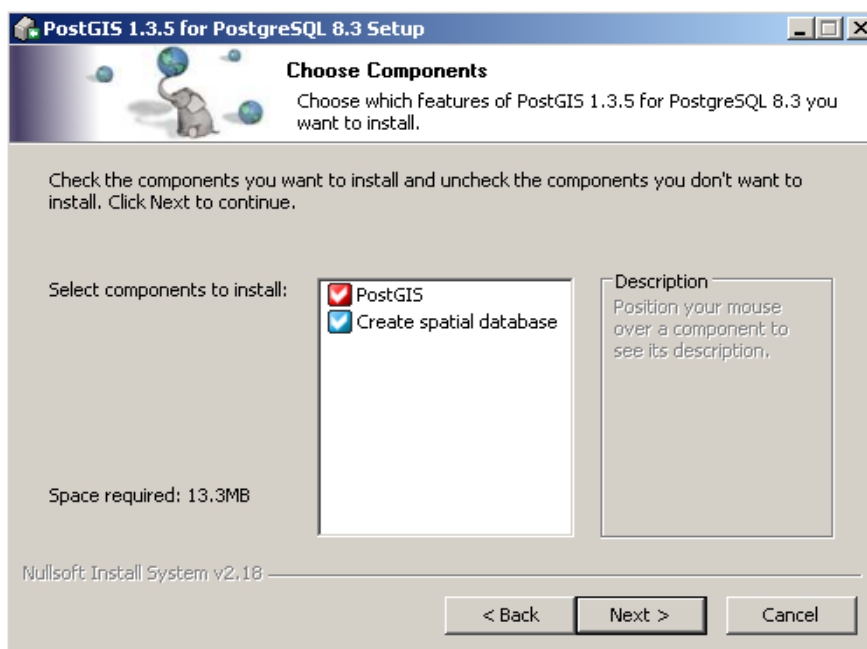
15. Στην συνέχεια επιλέγουμε κάποιον από τους διαθέσιμους εξυπηρετητές για την παροχή των σχετικών αρχείων.



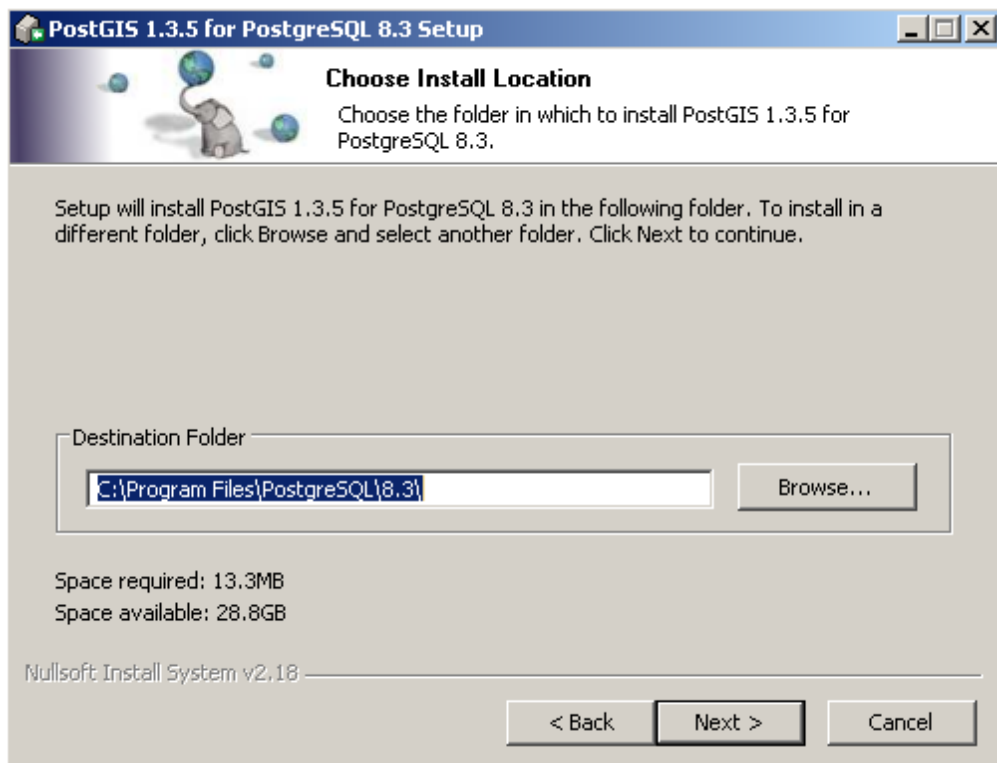
16. Όταν όλα τα απαιτούμενα για την εγκατάσταση αρχεία έχουν επιτυχώς αντιγραφεί στον δίσκο μας τότε μας ζητείται να αρχίσουμε την εγκατάσταση.



17. Επιλέγουμε το πεδίο “Create spatial database” στην οθόνη που ακολουθεί (το πεδίο “PostGIS” είναι ήδη επιλεγμένο)

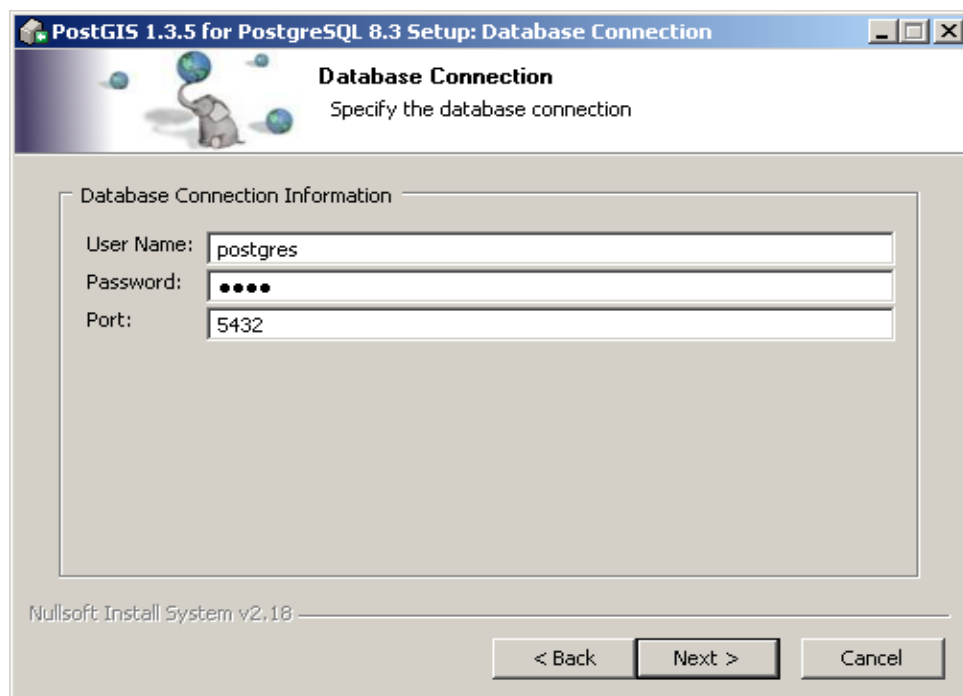


18. Ορίζουμε την τοποθεσία εγκατάστασης στον σκληρό δίσκο μας. Αν δεν υπάρχει λόγος για το αντίθετο διατηρούμε την προεπιλεγμένη τοποθεσία.

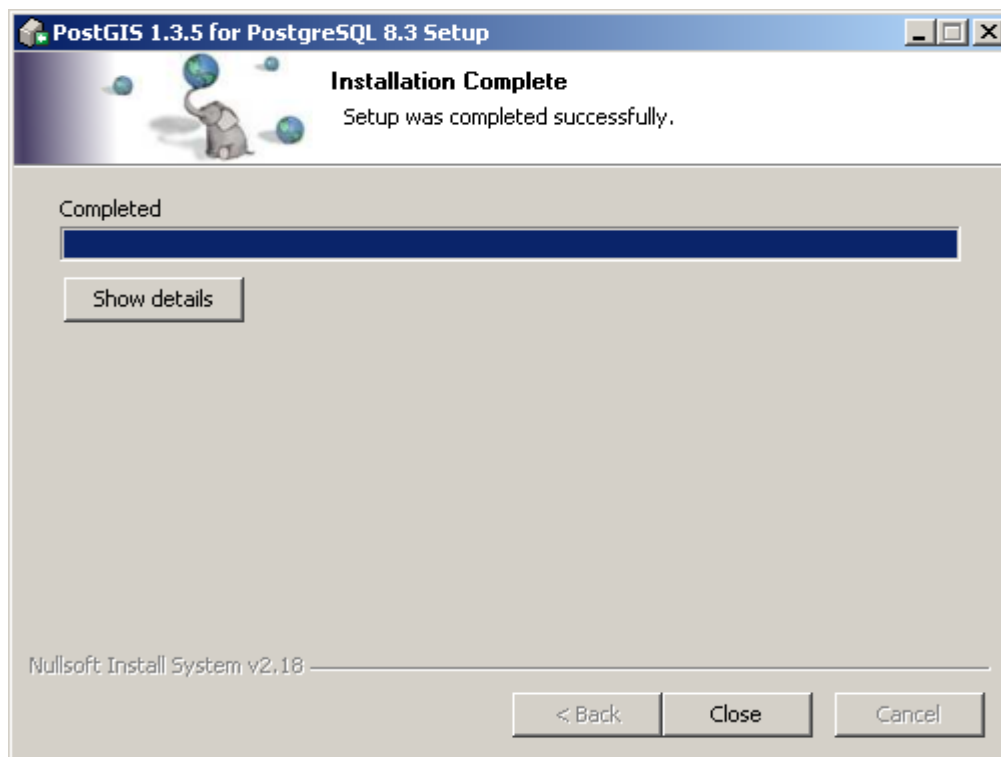


19. Συμπληρώνουμε το όνομα χρήστη (username) και συνθηματικό χρήστη (password), που

εισαγάγαμε στο βήμα 7, για την δημιουργία νέας βάσης δεδομένων.



20. Η βάση δεδομένων PostgreSQL και η επέκταση χωρικών λειτουργιών PostGIS έχουν εγκατασταθεί με επιτυχία.



4.2 Εγκατάσταση του λογισμικού ΓΣΠ QuantumGIS

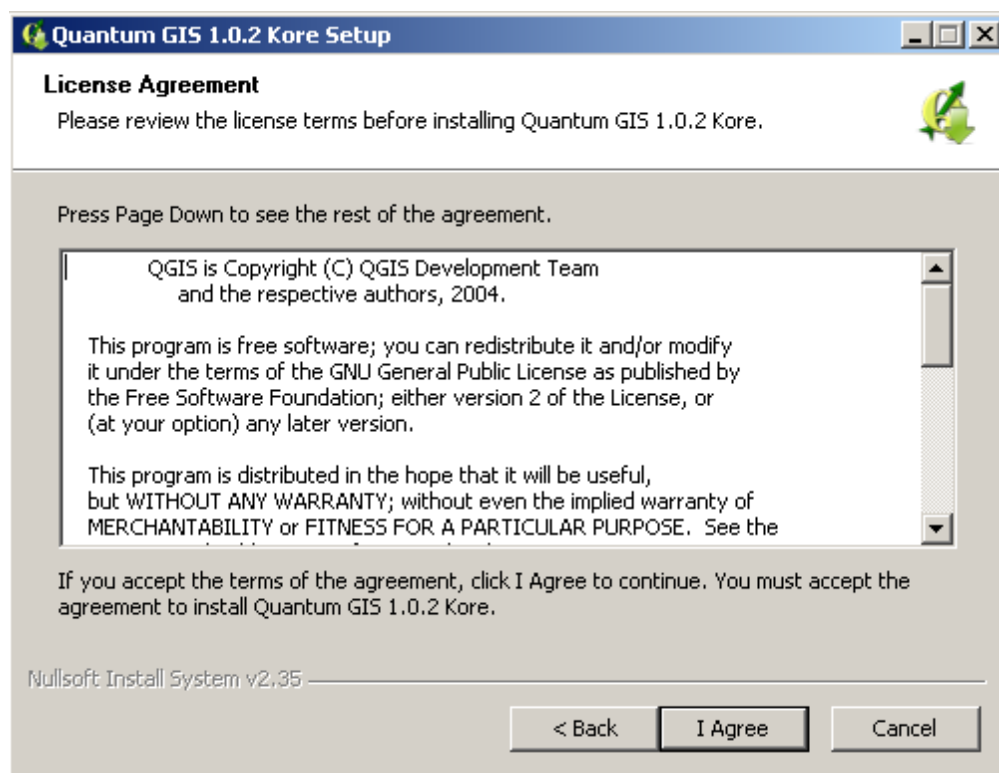
Αρχικά αντιγράφουμε στον τοπικό σκληρό δίσκο μας τα αρχεία που απαιτούνται για την εγκατάσταση του λογισμικού. Αυτά βρίσκονται στην τοποθεσία <http://www.qgis.org/en/download/binaries.html>. Επιλέγουμε την έκδοση που είναι συμβατή με το λειτουργικό σύστημά μας. Υπάρχει η δυνατότητα για να κατεβάσουμε όλα τα αρχεία στον τοπικό δίσκο μας πριν την εγκατάσταση (standalone installer ή offline installer) ή να κατεβάσουμε ένα μικρότερο αρχείο που θα κατεβάσει τα απαιτούμενα αρχεία κατά την εγκατάσταση (online installer). Εδώ ακολουθήσαμε την πρώτη επιλογή. Συγκεκριμένα χρησιμοποιήσαμε την έκδοση 1.02 για windows (<http://download.osgeo.org/qgis/win32/QGIS-1.0.2-0-Setup.exe>).

Όταν η αντιγραφή του αρχείου QGIS-1.0.2-0-Setup.exe έχει πλέον ολοκληρωθεί τότε κάνουμε διπλό κλικ σε αυτό. Έτσι ξεκινά η διαδικασία εγκατάστασης:

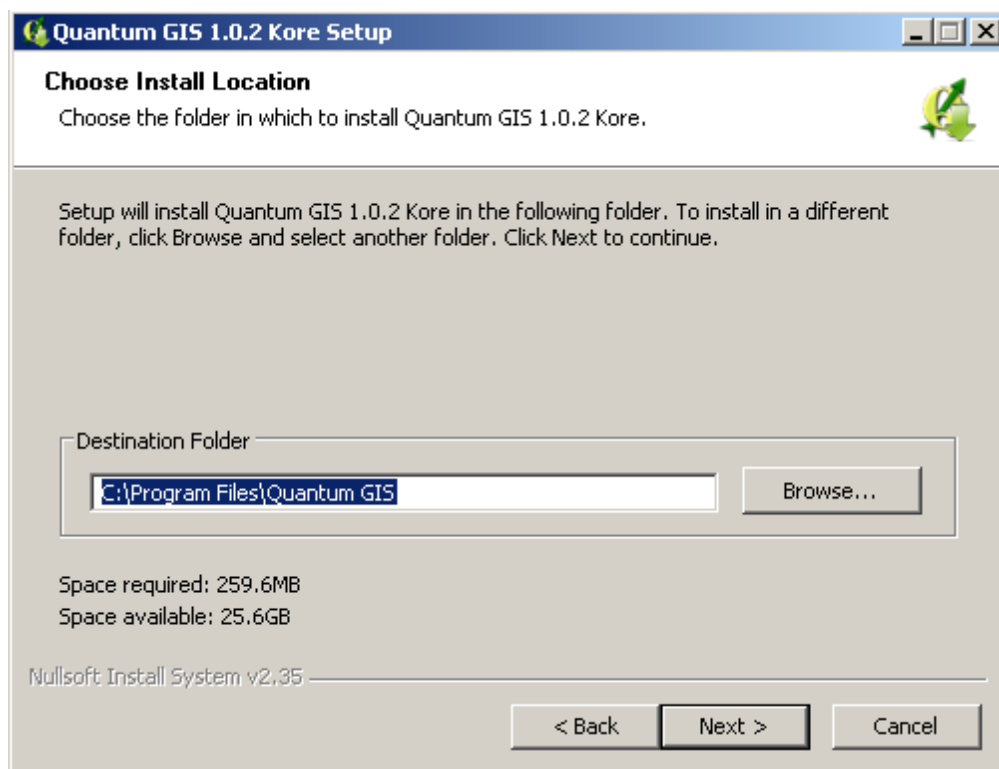
1. Στο πρώτο παράθυρο επιλέγουμε 'Next'



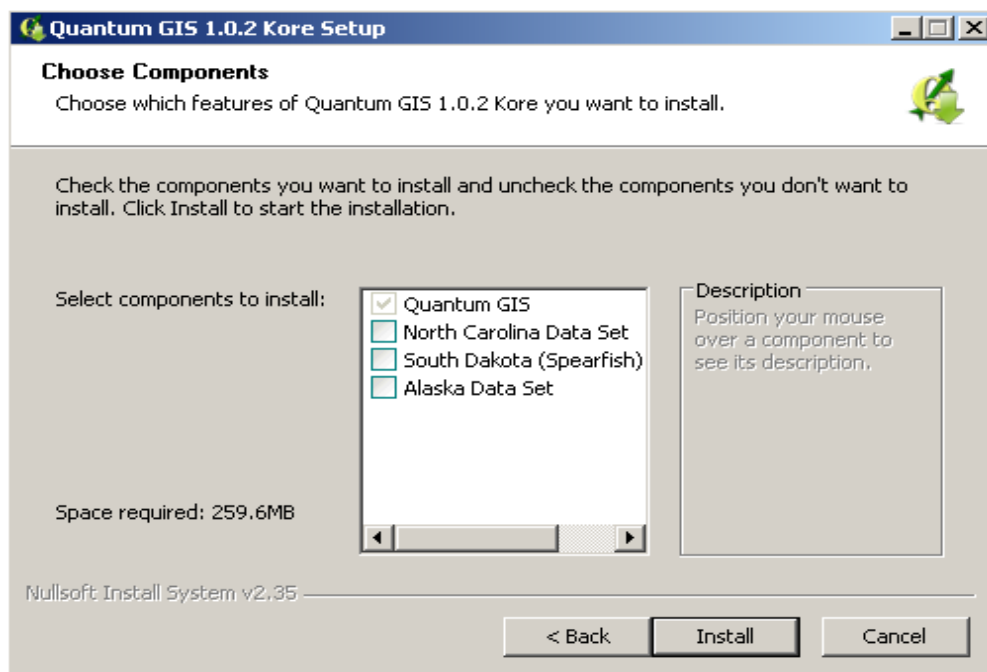
2. Στο δεύτερο παράθυρο επιλέγουμε 'I agree'



3. Επιλέγουμε την τοποθεσία εγκατάστασης (αν δεν υπάρχει λόγος να επιλέξουμε διαφορετικά διατηρούμε την προεπιλεγμένη) και στην συνέχεια 'Next'



Αν επιθυμούμε να αντιγραφούν στον τοπικό δίσκο μας τα χωρικά δεδομένα που παρέχονται για εκπαιδευτικούς λόγους μαζί με το πρόγραμμα (Βόρεια Καρολίνα, Νότια Ντακότα, Αλάσκα) επιλέγουμε τα αντίστοιχα κουτιά. Τα δεδομένα αυτά θα καταλάβουν επιπλέον 360mb επιπλέον στον σκληρό δίσκο του υπολογιστή μας. Στην συνέχεια επιλέγουμε 'Install'.



Η διαδικασία εγκατάστασης του QuantumGIS έχει πλέον ολοκληρωθεί. Όλα τα λογισμικά που συμπεριλαμβάνονται στον παρόντα οδηγό είναι πλέον έτοιμα να χρησιμοποιηθούν.



5. Βασικές Λειτουργίες

5.1. Λειτουργίες εισαγωγής και διαχείρισης δεδομένων

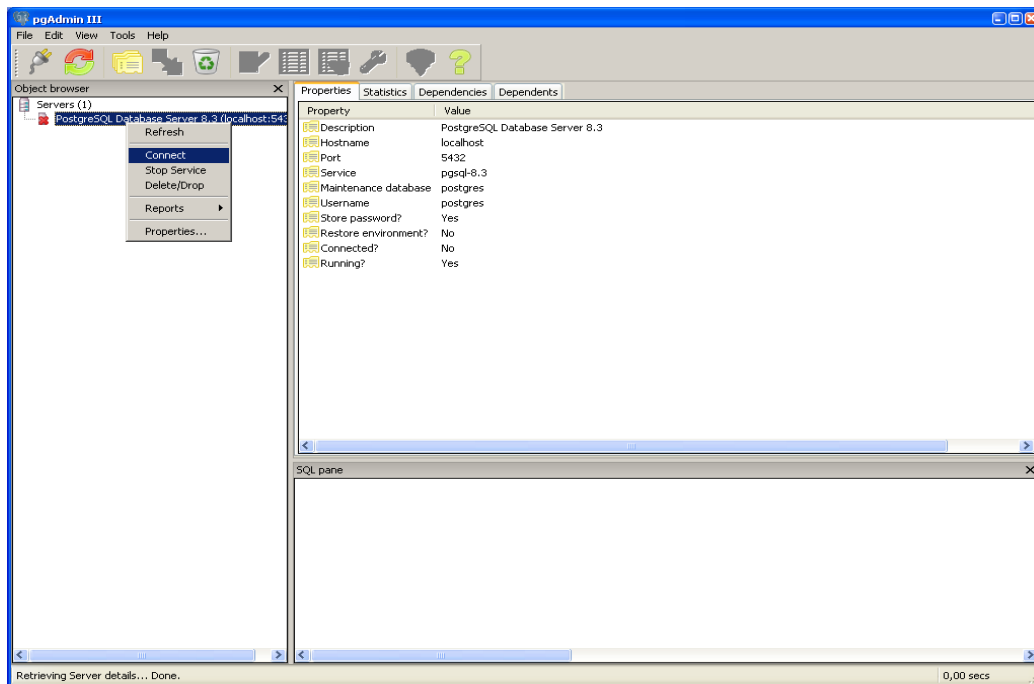
5.1.1 Δημιουργία βάσης δεδομένων και σύνδεση με αυτή

5.1.1.1 Δημιουργία βάσης δεδομένων με την χρήση προτύπου (template)

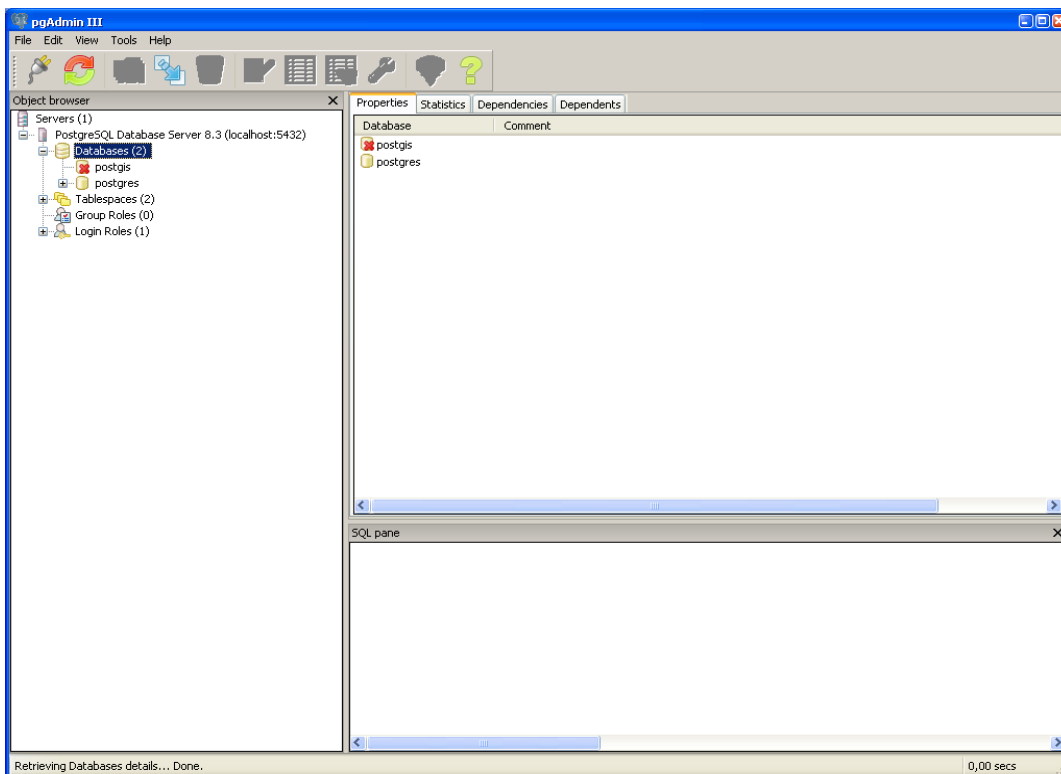
Ξεκινάμε πηγαίνοντας στο μενού “Start”, “Programs”, “PostgreSQL 8.3”, “pgAdmin III”.

Σημείωση: Η βάση δεδομένων PostGreSQL δεν έχει ή ίδια κάποιο γραφικό περιβάλλον διεπαφής με τον χρήστη. Λειτουργεί σαν υπηρεσία (service). Για την αλληλεπίδραση με την βάση δεδομένων χρησιμοποιείται ένα πρόγραμμα χρήστη (client). Ένα τέτοιο πρόγραμμα είναι και το pgAdmin III που παρέχεται και εγκαθίσταται μαζί με την βάση δεδομένων. Υπάρχει η δυνατότητα να χρησιμοποιηθεί κάποιο διαφορετικό πρόγραμμα με ιδιαίτερες δυνατότητες. Παρόλα αυτά σε αυτό τον οδηγό για λόγους ευκολίας και τυποποίησης θα χρησιμοποιήσουμε το πρόγραμμα που παρέχεται μαζί με την βάση δεδομένων, δηλαδή το pgAdmin III.

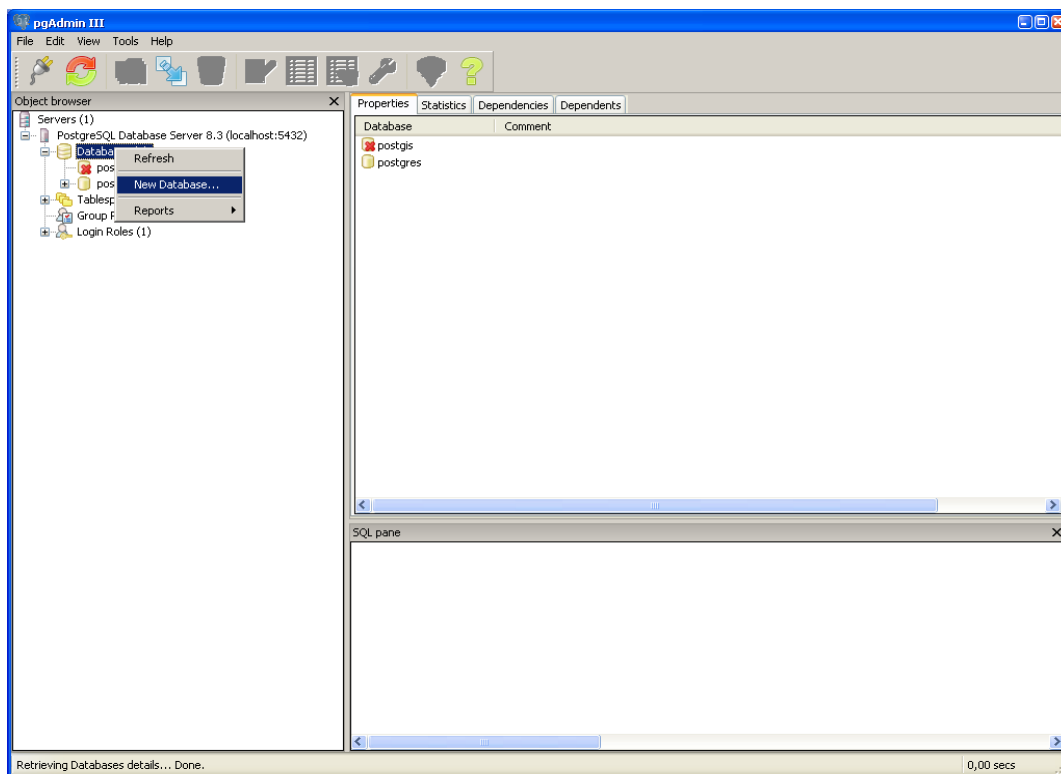
Το pgAdmin III έχει την μορφή που παρουσιάζεται στην εικόνα που ακολουθεί. Πάνω αριστερά ακριβώς κάτω από την ένδειξη “Servers (1)” υπάρχει η ένδειξη “PostgreSQL Database Server 8.3 (localhost 5432)”. Εκεί κάνουμε δεξί κλικ και επιλέγουμε “connect”. Μας ζητείται όνομα χρήστη και κωδικός και πληκτρολογούμε το όνομα postgres και τον κωδικό που επιλέξαμε στο βήμα 7 της διαδικασίας εγκατάστασης (ενότητα: 4.1 ΧΒΔ PostgreSQL/PostGIS).



Πατάμε πάνω στο σύμβολο “+” αριστερά από την ένδειξη “PostgreSQL Database Server 8.3 (localhost:5432)” και έτσι επεκτείνουμε το μενού. Κάνουμε το ίδιο και στην ένδειξη “databases”.



Βλέπουμε ότι έχουν δημιουργηθεί δύο βάσεις δεδομένων, η “postgres” και η “postgis”. Κάνουμε δεξί κλικ στην ένδειξη “Databases (2)” και επιλέγουμε “New Database”.



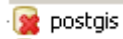
Στην φόρμα που ακολουθεί επιλέγουμε εισάγουμε τις ακόλουθες τιμές:

The 'New Database...' dialog box is shown with the 'Properties' tab selected. The fields are filled with the following values:

Field	Value
Name	template_postgis
OID	
Owner	postgres
Encoding	UTF8
Template	postgres
Tablespace	pg_default
Schema restriction	
Comment	

At the bottom of the dialog are three buttons: 'Help', 'OK', and 'Cancel'.

και επιλέγουμε “OK”. Με τον τρόπο αυτό δημιουργούμε μία βάση δεδομένων που θα μπορούμε να χρησιμοποιούμε για πρότυπο εδώ και εμπρός όταν θέλουμε να δημιουργήσουμε μία βάση δεδομένων που θα είναι ενισχυμένη με τις λειτουργίες PostGIS. Σημείωση: για να γίνει αυτό πρέπει να μην είναι ενεργοποιημένη η βάση δεδομένων PostGIS. Πρέπει δηλαδή να εμφανίζεται στο μενού έτσι:



και όχι έτσι:



Αν αυτό δεν συμβαίνει, τότε πατούμε πάνω στο Databases ()



και ανανεώνουμε την σύνδεση με την βάση δεδομένων πιέζοντας

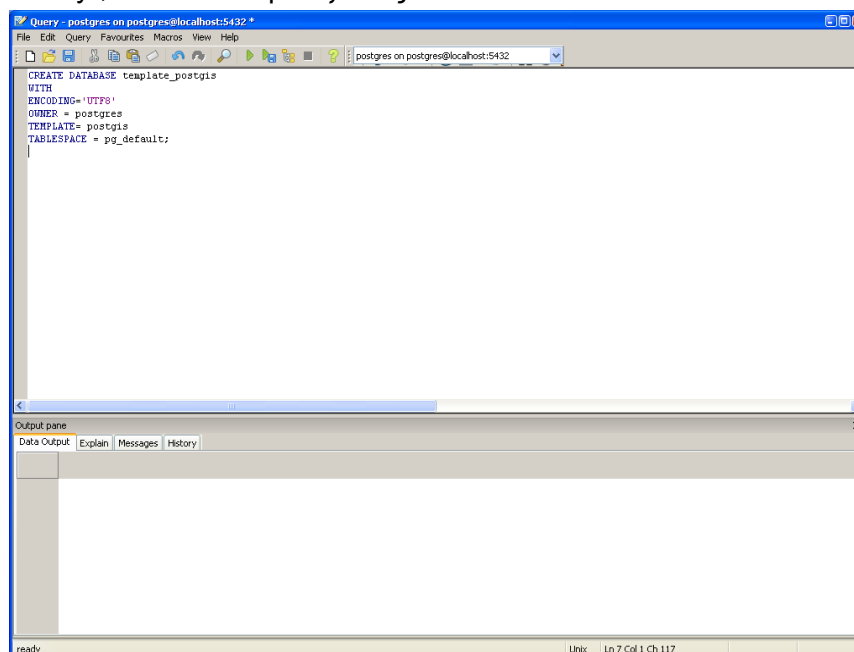


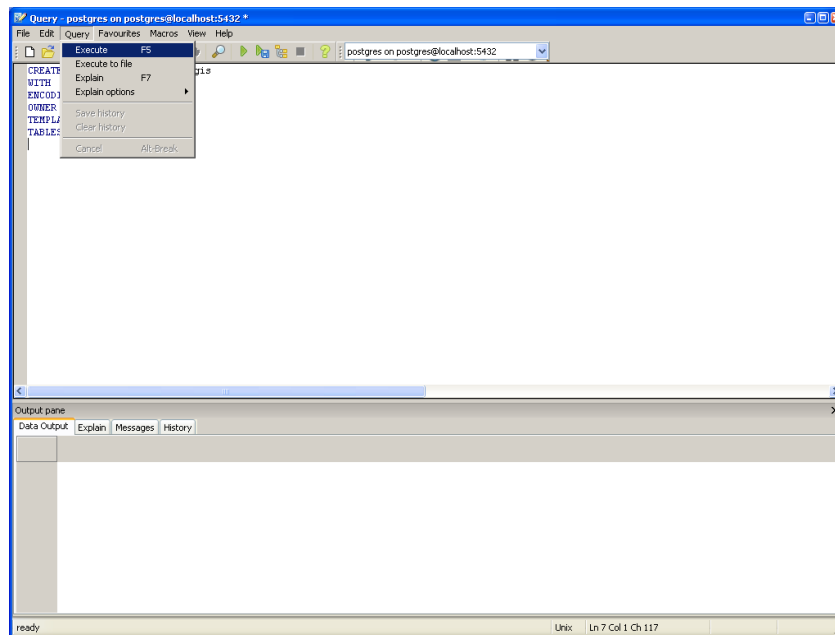
και ξαναπροσπαθούμε.

Εναλλακτικά μπορούμε να δημιουργήσουμε την βάση δεδομένων με εντολή SQL ως εξής: Πατάμε πάνω στην βάση δεδομένων postgres και στην συνέχεια επιλέγουμε από το μενού του προγράμματος στο πάνω μέρος της οθόνης “Tools”, “Query tool”. Έτσι εμφανίζεται ένα νέο παράθυρο στο οποίο μπορούμε να πληκτρολογήσουμε κώδικα σε γλώσσα SQL και ενσωματωμένες διαδικαστικές γλώσσες. Εκεί πληκτρολογούμε την ακόλουθη εντολή SQL:

```
CREATE DATABASE template_postgis
WITH
ENCODING='UTF8'
OWNER = postgres
TEMPLATE= postgres
TABLESPACE = pg_default;
```

Στην συνέχεια εκτελούμε αυτή την εντολή επιλέγοντας από το μενού στο πάνω μέρος του προγράμματος “Query”, “Execute” ή πιέζοντας F5.





Αντίστοιχα εκτελούμε την παρακάτω εντολή για να δημιουργήσουμε την βάση δεδομένων στην οποία θα εργαστούμε στα πλαίσια του παρόντα οδηγού.

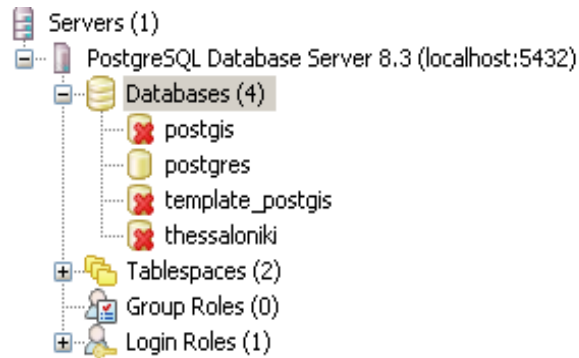
```
CREATE DATABASE thessaloniki  
WITH  
ENCODING='UTF8'  
OWNER = postgres  
TEMPLATE= template_postgis  
TABLESPACE = pg_default;
```

Όταν η βάση έχει πλέον δημιουργηθεί τότε στο κάτω μέρος του παραθύρου ερωτημάτων παίρνουμε την ένδειξη:

“Query returned successfully with no result in ms”

Γυρίζουμε στο παράθυρο του pgAdmin III. Πατάμε την ένδειξη “Databases (3)” και ανανεώνουμε την σύνδεση με την βάση δεδομένων.

Τώρα πρέπει πλέον να βλέπουμε τις ακόλουθες βάσεις δεδομένων:



Στην συνέχεια, και προκειμένου να αρχίσουμε να εργαζόμαστε στην βάση δεδομένων “thessaloniki” που μόλις δημιουργήσαμε, πατάμε πάνω της και επιλέγουμε από τα μενού “Tools”, “Query tool”. Έτσι ανοίγουμε ένα παράθυρο ερωτημάτων SQL.

5.1.1.2 Δημιουργία βάσης δεδομένων χωρίς την χρήση προτύπου (template)

Αν για κάποιο λόγο δεν υπάρχει βάση δεδομένων την οποία θα χρησιμοποιήσουμε σαν πρότυπο (λ.χ. γιατί την σβήσαμε κατά λάθος), τότε θα πρέπει να ενισχύσουμε “χειροκίνητα” την βάση δεδομένων με χωρικές λειτουργίες PostGIS.

Αρχικά δημιουργούμε μία βάση δεδομένων είτε μέσω της γραφικής διεπαφής pgAdmin III (Ακριβώς όπως κάναμε στην προηγούμενη παράγραφο, αλλά αυτή την φορά χωρίς αφήνοντας το πεδίο “Template” κενό) είτε με την ακόλουθη εντολή SQL:

```
CREATE DATABASE my_db
WITH
ENCODING='UTF8'
OWNER = postgres
TABLESPACE = pg_default;
```

Έτσι η βάση δεδομένων έχει δημιουργηθεί αλλά δεν είναι ακόμα ενισχυμένη με τις χωρικές λειτουργίες PostGIS.

Σημείωση:

Για να διαπιστώσουμε αν η βάση δεδομένων είναι ενισχυμένη ή όχι με χωρικές λειτουργίες φτάνει να δοκιμάσουμε την εισαγωγή ενός πίνακα με πεδίο γεωμετρίας. Αρχικά επιλέγουμε “Databases” στο πάνελ του object browser και κάνουμε δεξί κλικ και “Refresh”. Έτσι εμφανίζεται η βάση δεδομένων my_db που δημιουργήσαμε παραπάνω. Στην συνέχεια πατάμε πάνω στην καινούργια βάση δεδομένων my_db και από το μενού του pgAdmin επιλέγουμε Tools, Query tool. Έτσι ενεργοποιείται ένα παράθυρο ερωτημάτων για την νέα βάση δεδομένων. Σε αυτό εκτελούμε την ακόλουθη εντολή:

```
CREATE TABLE my_table(
id INTEGER PRIMARY KEY,
name VARCHAR(60)
);
SELECT AddGeometryColumn('public','my_table','the_geom',4121,'LINESTRING',2);
```

Σε περίπτωση που δεν έχουμε ενεργοποιήσει τις λειτουργίες PostGIS παίρνουμε κάποιο μήνυμα σφάλματος όπως το παρακάτω:

ERROR: function addgeometrycolumn("unknown", "unknown", "unknown", integer, "unknown", integer) does not exist
SQL state: 42883
Hint: No function matches the given name and argument types. You may need to add explicit type casts.
Character: 79

Αν ωστόσο οι βάση δεδομένων είναι ενισχυμένη με τις λειτουργίες PostGIS, τότε η εντολή θα εκτελεσθεί κανονικά.

Επιλέγουμε “Databases” στο πάνελ του object browser και κάνουμε δεξί κλικ και “Refresh”, έτσι ώστε να εμφανιστεί η βάση δεδομένων my_db που μόλις δημιουργήσαμε. Στην συνέχεια πατάμε πάνω στην καινούργια βάση δεδομένων my_db και από το μενού του pgAdmin επιλέγουμε Tools, Query tool. Στο παράθυρο ερωτημάτων που δημιουργείται επιλέγουμε “File”, “open” και πλοηγούμαστε στην τοποθεσία:

C:\Program Files\PostgreSQL\8.2\share\contrib\lwpogis.sql

Ανοίγουμε το αρχείο και το εκτελούμε επιλέγοντας “Query” “Execute”.

Με τον ίδιο τρόπο εκτελούμε το αρχείο που βρίσκεται στην τοποθεσία:

C:\Program Files\PostgreSQL\8.2\share\contrib\spatial_ref_sys.sql

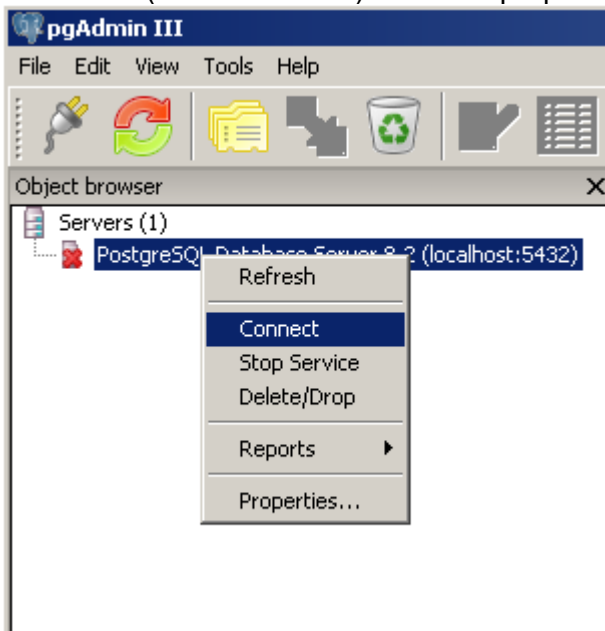
Με αυτά τα βήματα, η βάση δεδομένων my_db που δημιουργήσαμε προηγουμένως έχει ενισχυθεί με χωρικές λειτουργίες PostGIS.

5.1.1.3 Σύνδεση με την βάση δεδομένων

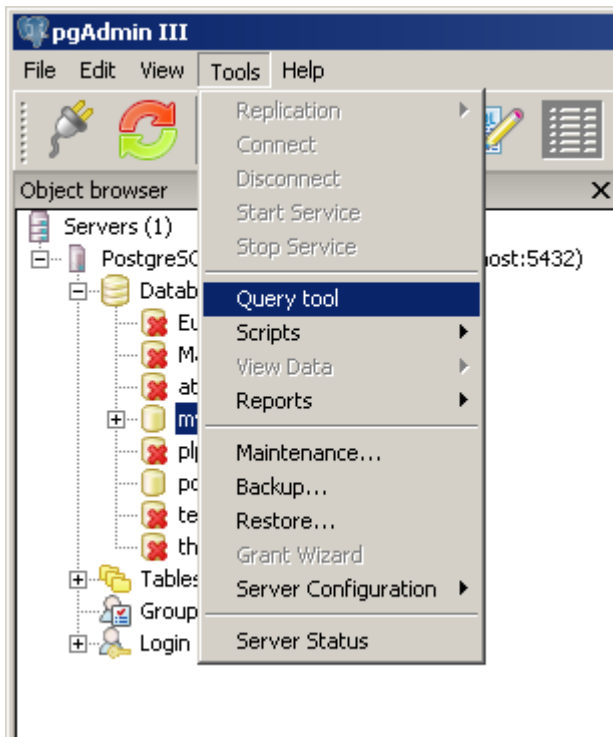
Την επόμενη φορά που θα θελήσουμε να συνδεθούμε με την βάση δεδομένων και να εργαστούμε σε αυτή θα πρέπει να κάνουμε τα εξής:

A. Ξεκινάμε το pgAdminIII

B. Στο πάνελ του Object browser κάνουμε δεξί κλικ στην βάση δεδομένων “PostgreSQL Database Server 8.2 (localhost:5432)” και επιλέγουμε “Connect”.



Στην συνέχεια επεκτείνουμε το μενού “PostgreSQL Database Server 8.2 (Localhost:5432)” και διαδοχικά το μενού “Databases”. Τότε, πατάμε πάνω στην βάση δεδομένων που θέλουμε να εργαστούμε και από το μενού του προγράμματος επιλέγουμε “Tools”, “Query tool”. Έτσι ενεργοποιείται το παράθυρο ερωτημάτων για την βάση δεδομένων και μπορούμε πλέον να εργαστούμε σε αυτή.



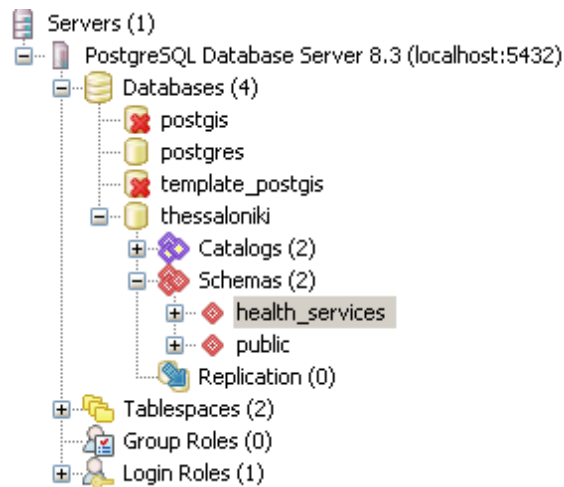
5.1.2 Δημιουργία σχήματος

Μία βάση δεδομένων περιέχει στοιχεία τα οποία είναι ομαδοποιημένα σε σχήματα. Κάθε σχήμα περιέχει πίνακες, όψεις, σκανδαλιστές, συναρτήσεις κ.α. Με το που δημιουργούμε μία βάση δεδομένων, αυτή εξ'ορισμού περιέχει ένα σχήμα το οποίο ονομάζεται “public”. Αν θελήσουμε να δημιουργήσουμε ένα άλλο σχήμα τότε μπορούμε να χρησιμοποιήσουμε την ακόλουθη εντολή SQL (έστω ότι θέλουμε να δημιουργήσουμε ένα σχήμα με την ονομασία ονομάζεται health_services):

```
CREATE SCHEMA health_services
AUTHORIZATION postgres;
GRANT ALL ON SCHEMA health_services TO postgres;
GRANT ALL ON SCHEMA health_services TO public;
```

Αφού πληκτρολογήσουμε αυτή την εντολή στο παράθυρο ερωτημάτων επιλέγουμε “execute”.

Πλέον, αν ανανεώσουμε την σύνδεση με την βάση δεδομένων, τότε βλέπουμε ότι αυτή έχει πέραν του “public” και ένα σχήμα που ονομάζεται “health_services”.



Αυτή η διαδικασία μπορεί να γίνει γραφικά από το παράθυρο pgAdminIII πατώντας πάνω στην βάση δεδομένων και στην συνέχεια επιλέγοντας “New Object”, “New Schema”.

5.1.3 Δημιουργία πινάκων και εισαγωγή δεδομένων με εντολές SQL

Το επόμενο βήμα μετά την δημιουργία της βάσης δεδομένων είναι η δημιουργία πινάκων και η εισαγωγή δεδομένων. Τα δεδομένα αποθηκεύονται στην βάση δεδομένων ομαδοποιημένα σε πίνακες. Οι πίνακες περιέχουν κατηγορίες ομοειδών στοιχείων (λ.χ. Ένας πίνακας περιέχει δρόμους, ένας άλλος δήμους, ένας άλλος νοσοκομεία). Ανάλογα με το πλήθος των γνωρισμάτων (δηλαδή στηλών) που περιέχονται στους πίνακες περιλαμβάνονται πληροφορίες για θεματικά και γεωμετρικά χαρακτηριστικά των επιμέρους στοιχείων. Στην συνέχεια θα αναπτύξουμε τρεις διαφορετικούς τρόπους δημιουργίας πινάκων και εισαγωγής δεδομένων:

- Με την χρήση εντολών SQL
- Εισαγωγή από αρχεία τύπου *.shp με την χρήση του προγράμματος shp2pgsql
- Εισαγωγή από αρχεία τύπου *.shp μέσω του QuantumGIS

Ας δούμε πώς δημιουργούμε δεδομένα και πώς εισάγουμε σε αυτούς δεδομένα με εντολές SQL. Έστω ότι στην βάση δεδομένων θέλουμε να δημιουργήσουμε έναν πίνακα με οντότητες που απεικονίζουν μέρος του οδικού δικτύου της Θεσσαλονίκης. Αρχικά πρέπει να είναι ξεκάθαρο τι θέλουμε να περιλαμβάνει ο σχετικός πίνακας. Έστω ότι για την ώρα μας αρκεί να συμπεριλάβουμε στον πίνακα τα εξής χαρακτηριστικά του οδικού δικτύου:

- Αύξων αριθμός οδικού τμήματος
- Όνομα οδικού τμήματος
- Όνομα οδικού τμήματος με λατινικούς χαρακτήρες
- Δείκτης μέσης ταχύτητας οδικού τμήματος
- Θέση (γεωμετρία) του οδικού τμήματος

Επιπλέον πρέπει να γνωρίζουμε ποιο όνομα επιθυμούμε να δώσουμε στον σχετικό πίνακα. Έστω ότι επιθυμούμε να τον ονομάσουμε “road_net”. Για να δημιουργήσουμε τον σχετικό πίνακα πληκτρολογούμε την ακόλουθη εντολή στο παράθυρο ερωτημάτων και επιλέγουμε “execute”:

```
CREATE TABLE health_services.road_net(
id    INTEGER PRIMARY KEY,
name  VARCHAR(60),
name_eng VARCHAR(60),
speed_profile INTEGER
);
```

Η παραπάνω εντολή περιέχει τα εξής:

- Στην πρώτη γραμμή περιγράφεται το σχήμα στο οποίο θα περιλαμβάνεται ο πίνακας καθώς και το όνομα που θα έχει.
- Στην δεύτερη γραμμή προσδιορίζεται το γνώρισμα αύξοντος αριθμού που θα είναι ακέραιος αριθμός και το πρωτεύον κλειδί του πίνακα. Το πρωτεύον κλειδί είναι ένα γνώρισμα που έχει μοναδική τιμή για κάθε εγγραφή του πίνακα, και είναι αναγκαίο για την διαχείριση των πινάκων από την βάση δεδομένων.
- Στην τρίτη και τέταρτη γραμμή προσδιορίζονται τα γνωρίσματα του ονόματος με ελληνικούς και λατινικούς χαρακτήρες που θα έχουν αλφαριθμητική μορφή (δηλαδή θα περιλαμβάνουν γράμματα, λοιπούς χαρακτήρες και αριθμούς) και μέγεθος 60 χαρακτήρων.
- Στην τέταρτη γραμμή προσδιορίζεται το γνώρισμα speed_profile που θα έχει μορφή ακέραιου αριθμού.

Παρόλα αυτά μέχρι στιγμής δεν έχει προσδιοριστεί το γνώρισμα γεωμετρίας του πίνακα. Αυτό γίνεται με την ακόλουθη εντολή:

```
SELECT AddGeometryColumn('health_services','road_net','the_geom',4121,'LINESTRING',2);
```

Αυτή η εντολή καλεί την λειτουργία AddGeometryColumn που δημιουργεί γνωρίσματα γεωμετρίας. Στην παρένθεση που ακολουθεί την εντολή AddGeometryColumn προσδιορίζονται τα εξής:

- Το σχήμα στο οποίο βρίσκεται ο πίνακας (health_services)
- Το όνομα του πίνακα για τον οποίον θα δημιουργηθεί το γνώρισμα γεωμετρίας (road_net)
- Το όνομα του γνωρίσματος γεωμετρίας (the_geom)
- Τον κωδικό του γεωδαιτικού συστήματος αναφοράς στο οποίο βρίσκονται τα δεδομένα που θα εισαχθούν στον πίνακα (4121, ο κωδικός αυτός υποδεικνύει το σύστημα ΕΓΣΑ 87)
- Το είδος της γεωμετρίας των στοιχείων που θα εισαχθούν (δηλαδή, σημεία, γραμμές ή πολύγωνα, πολυσημεία, πολυγραμμές, πολύ-πολύγωνα (multi-polygons) και συλλογές γεωμετρίας (geometry collections)). Εδώ προσδιορίζουμε ότι πρόκειται για γραμμή (LINESTRING).
- Το πλήθος των διαστάσεων που επιθυμούμε να υποστηρίξει η γεωμετρία του πίνακα (2).

Στην συνέχεια ενεργοποιούμε τον έλεγχο εγκυρότητας της γεωμετρίας με την ακόλουθη εντολή:

```
ALTER TABLE health_services.road_net ADD CONSTRAINT geom_valid_check CHECK
(isvalid(the_geom));
```

- Στην εντολή αυτή προσδιορίζεται αρχικά το σχήμα και ο πίνακας στον οποίο επιθυμούμε να εισαχθεί η εντολή (health_services.road_net)
- Ο έλεγχος δημιουργείται σαν ένα νέο στοιχείο της βάσης δεδομένων. Έτσι πρέπει να προσδιορίσουμε το όνομά του (geom_valid check).

- Στο τέλος προσδιορίζεται το όνομα της στήλης γεωμετρίας στην οποία θα εφαρμόζεται ο έλεγχος (the_geom).

Στην συνέχεια δημιουργούμε έναν χωρικό δείκτη γενικευμένου δένδρου αναζήτησης για τον πίνακα με την εντολή:

```
CREATE INDEX road_net_idx ON health_services.road_net USING GIST (the_geom);
```

- Τον δείκτη ονομάζουμε road_net_idx.
- Προσδιορίζουμε το σχήμα και το όνομα του πίνακα (health_services.road_net).
- Ακόμα προσδιορίζουμε το είδος του δείκτη (GIST) και το όνομα του πεδίου γεωμετρίας (the_geom).

Στην συνέχεια εισάγουμε δεδομένα με διαδοχικές εντολές INSERT. Οι εντολές έχουν την ακόλουθη μορφή:

```
INSERT INTO όνομα σχήματος.όνομα πίνακα VALUES (τιμή γωρίσματος1,τιμή γωρίσματος 2, ....,τιμή
γωρίσματος ν, GeomFromText('LINESTRING(
κόμβος γραμμής 1,
κόμβος γραμμής 2,
.....
κόμβος γραμμής ν,)',κωδικός γεωδαιτικού συστήματος));
```

Ακολουθώντας αυτό το πρότυπο οι εντολές με τις οποίες θα εισάγουμε δεδομένα οδικού δικτύου για την πόλη της Θεσσαλονίκης είναι οι ακόλουθες:

```
INSERT INTO health_services.road_net VALUES (101,'ΘΕΣΣΑΛΟΝΙΚΗΣ-
ΜΑΛΓΑΡΩΝ','THESSALONIKIS-MALGARWN',10.25,
GeomFromText('LINESTRING(
19232578.31 4929703.03,
19237529.19 4933056.09,
19240777.92 4933359.13,
19245128.96 4933842.78)',4121));
```

```
INSERT INTO health_services.road_net VALUES (102,'ΝΕΑ ΔΥΤΙΚΗ ΕΙΣΟΔΟΣ','NEA DYTIKI
EISODOS',14.25, GeomFromText
('LINESTRING(
19245128.96 4933842.78,
19246145.94 4933791.02,
19247162.92 4933490.83,
19248471.59 4932704.17,
19249220.53 4932352.26,
19249527.99 4932031.42,
19250576.51 4931513.95)',4121));
```

```
INSERT INTO health_services.road_net VALUES (203,'28ΗΣ ΟΚΤΩΒΡΙΟΥ','28is OCTOVRIOU' ,10,
GeomFromText('LINESTRING(
19247162.92 4933490.83,
19248038 4932538.56,
19248321.81 4932021.07,
19249693.55 4931565.7,
19250576.51 4931513.95
)',4121));
```

```

INSERT INTO health_services.road_net VALUES (204,'ΛΑΓΚΑΔΑ','LAGKADA',06,
GeomFromText('LINESTRING(
19250576.51 4931513.95,
19251065.75 4931618.74,
19251099.74 4933180.61,
19250941.12 4933894.7,
19251462.31 4936067.03,
19251473.64 4937004.56,
19251507.63 4937287.32,
19252448.04 4939609.27,
19253886.98 4942378.48
)',4121)));

```

```

INSERT INTO health_services.road_net VALUES (205,'ΜΟΝΑΣΤΗΡΙΟΥ','MONASTIRIOU',08,
GeomFromText('LINESTRING(
19236699.04 4941127.77,
19239055.72 4940353.61,
19240551.31 4940145.19,
19241140.48 4939430.64,
19241389.75 4938775.69,
19245559.27 4936037.27,
19246714.95 4935114.71,
19247598.71 4933835.19,
19251065.75 4931618.74
)',4121)));

```

```

INSERT INTO health_services.road_net VALUES (206,'ΕΓΝΑΤΙΑ','EGNATIA',07,
GeomFromText('LINESTRING(
19240777.92 4933359.13,
19239690.22 4936453.93,
19239690.22 4938507.76,
19241140.48 4939430.64,
19241888.28 4939847.45,
19252448.04 4939609.27,
19254306.2 4939400.87,
19254351.52 4938924.54,
19253626.38 4938388.69,
19253513.08 4937852.87,
19253603.72 4937153.38
)',4121)));

```

```

INSERT INTO health_services.road_net VALUES (207,'ΠΕΡΙΦΕΡΕΙΑΚΗ
ΛΕΟΦΩΡΟΣ','PERIFEREIAKI LEOFOROS',09, GeomFromText

```

```

('LINESTRING(
19245128.96 4933842.78,
19245106.06 4935367.66,
19245559.27 4936037.27,
19246556.33 4936945.03,
19247779.99 4937093.85,
19251473.64 4937004.56,
19252244.1 4937421.26,

```

```

19253603.72 4937153.38,
19253988.95 4935992.63,
19254351.52 4935129.59,
19254759.4 4934742.74,
19254510.14 4933522.77,
19254918.03 4931588.99,
19255529.86 4929982.76,
19256209.67 4929120.26,
19256504.26 4928049.68,
19257048.11 4927722.58,
19258022.51 4927633.38,
19258226.45 4927365.76,
19258135.81 4925938.61,
19257523.98 4924690.02,
19255756.46 4922995.77,
19254918.03 4922252.77
)',4121));

```

```

INSERT INTO health_services.road_net VALUES (208,'ΕΘΝΙΚΗΣ ΑΝΤΙΣΤΑΣΕΩΣ','ΕΘΝΙΚΙΣ
ΑΝΤΙΣΤΑΣΕΟΣ',05, GeomFromText

```

```

('LINESTRING(
19254918.03 4922252.77,
19254487.48 4922995.77,
19253966.29 4923946.89
)',4121));

```

```

INSERT INTO health_services.road_net VALUES (209,'ΒΑΣΙΛΙΣΣΗΣ ΟΛΓΑΣ','VASSILISIS
OLGAS',07, GeomFromText('LINESTRING(
19253966.29 4923946.89,
19253467.76 4924749.47,
19252969.23 4926414.3,
19253195.83 4928406.53,
19252969.23 4929625.85
)',4121));

```

```

INSERT INTO health_services.road_net VALUES (210,'ΤΣΙΜΙΣΚΗ','TSIMISKI',06,
GeomFromText('LINESTRING(
19252969.23 4929625.85,
19250576.51 4931513.95
)',4121));

```

```

INSERT INTO health_services.road_net VALUES (211,'ΕΓΝΑΤΙΑ','EGNATIA',08,
GeomFromText('LINESTRING(
19251065.75 4931618.74,
19253445.1 4929953.01,
19254056.93 4929179.74,
19254419.5 4924600.84,
19255756.46 4922995.77
)',4121));

```

```

INSERT INTO health_services.road_net VALUES (212,'ΕΛΕΥΘΕΡΙΟΥ ΒΕΝΙΖΕΛΟΥ','ELEFThERIOU
VENIZELOU',05, GeomFromText

```

```
('LINESTRING(
19254487.48 4922995.77,
19253671.7 4923382.15,
19253150.51 4923887.44
)',4121));
```

```
INSERT INTO health_services.road_net VALUES (213,'ΜΕΓΑΛΟΥ ΑΛΕΞΑΝΔΡΟΥ','MEGALOU
ALEKSANDROU',07, GeomFromText
```

```
('LINESTRING(
19253150.51 4923887.44,
19253037.21 4925135.93,
19252674.64 4926146.72,
19252991.89 4928406.53,
19252923.91 4929150
)',4121));
```

```
INSERT INTO health_services.road_net VALUES (214,'ΛΕΩΦΟΡΟΣ ΝΙΚΗΣ','LEOFOROS NIKIS',06,
GeomFromText('LINESTRING(
19252923.91 4929150,
19251994.83 4930101.73,
19250997.77 4930875.08,
19250576.51 4931513.95
)',4121));
```

Αν θέλουμε να δημιουργήσουμε έναν πίνακα με σημειακά αυτό μπορεί να γίνει με εντολές παραπλήσιες ως προς αυτές που παρουσιάστηκαν παραπάνω. Το παράδειγμα που ακολουθεί αφορά στην δημιουργία ενός πίνακα σημειακών στοιχείων:

```
CREATE TABLE health_services.hospitals(
ID INTEGER PRIMARY KEY,
NAME VARCHAR(70)
);
```

```
SELECT AddGeometryColumn('health_services','hospitals','the_geom',4121,'POINT',2);
```

```
ALTER TABLE health_services.hospitals ADD CONSTRAINT hospitals_valid_check CHECK
(isvalid(the_geom));
```

```
CREATE INDEX hospitals_idx ON health_services.hospitals USING GIST (the_geom);
```

Παραδείγματα εντολών SQL για την εισαγωγή σημειακών δεδομένων:

```
INSERT INTO health_services.hospitals VALUES (200,'ΠΑΘΟΛΟΓΙΚΟ ΝΟΣΟΚΟΜΕΙΟ
ΝΕΑΠΟΛΗΣ',GeomFromText('POINT
(19252515.21 4933565.18)',4121));
INSERT INTO health_services.hospitals VALUES (201,'ΝΟΣΟΚΟΜΕΙΟ ΑΓΙΟΣ
ΔΗΜΗΤΡΙΟΣ',GeomFromText('POINT
(19253596.71 4930995.65)',4121));
INSERT INTO health_services.hospitals VALUES (202,'ΝΟΣΟΚΟΜΕΙΟ ΔΕΡΜΑΤΙΚΩΝ
ΝΟΣΗΜΑΤΩΝ',GeomFromText('POINT
```

```
(19253527.68 4926493.21)',4121));
INSERT INTO health_services.hospitals VALUES (203,'ΝΟΣΟΚΟΜΕΙΟ
ΑΧΕΠΑ',GeomFromText('POINT(19253865.17
4929898.8)',4121));
```

Ένα άλλο παράδειγμα αφορά στην δημιουργία ενός πίνακα πολυγωνικών στοιχείων και την εισαγωγή δεδομένων σε αυτόν. Έστω ότι θέλουμε να δημιουργήσουμε έναν πίνακα με τα πολύγωνα των δήμων της Θεσσαλονίκης:

```
CREATE TABLE health_services.municipal(
id INTEGER PRIMARY KEY,
name VARCHAR(30)
);
```

```
SELECT AddGeometryColumn('health_services', 'municipal', 'the_geom',4121,'POLYGON',2);
```

```
ALTER TABLE health_services.municipal ADD CONSTRAINT geom_valid_check CHECK
(isvalid(the_geom));
```

```
CREATE INDEX municipal_idx ON health_services.municipal USING GIST (the_geom)
```

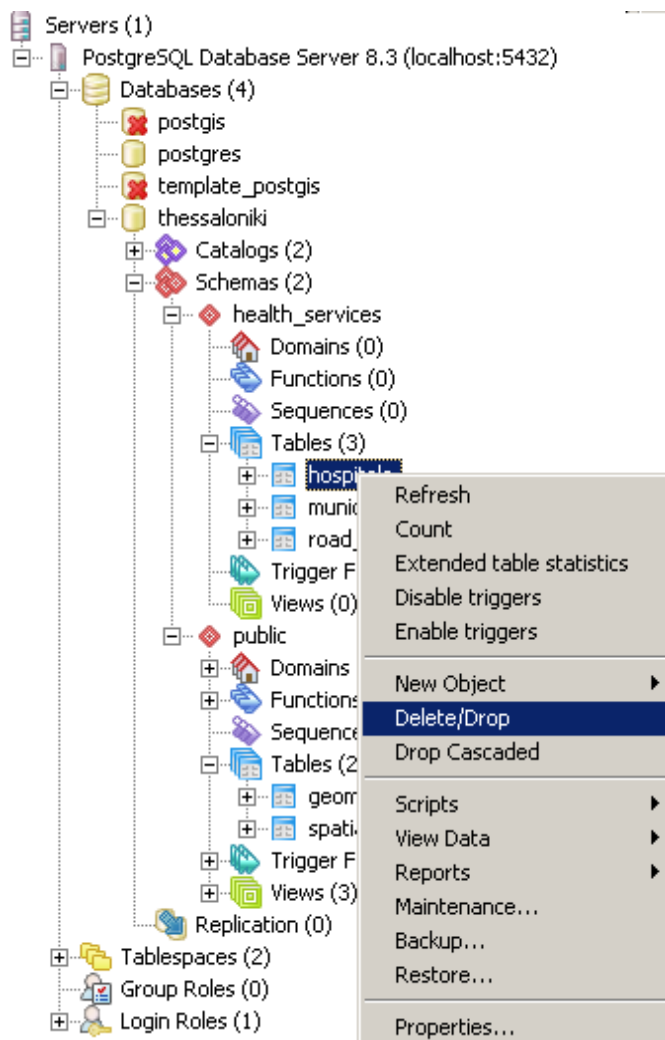
Αν τώρα θελήσουμε να εισάγουμε πολυγωνικά δεδομένα πρέπει να χρησιμοποιήσουμε μία εντολή που να μοιάζει με την ακόλουθη:

```
INSERT INTO health_services.municipal VALUES (400,'ΘΕΣΣΑΛΟΝΙΚΗ',
GeomFromText('POLYGON((
19252337.28 4924473.456,
19252400.28 4929709.089,
19248621.84 4931397.553,
19247754.52 4930485.015,
19247176.54 4931934.153,
19247087.17 4933316.181,
19249196.15 4933179.604,
19250922.02 4932784.313,
19251061.93 4933355.616,
19251240.67 4932719.875,
19251378.39 4932853.56,
19253251.49 4932012.049,
19254181.08 4932068.789,
19253488.1 4930723.478,
19254917.28 4930153.292,
19254945.12 4928520.803,
19255963.35 4928880.35,
19257276.79 4927892.109,
19254963.434 4923695.129,
19252337.28 4924473.456))',4121));
```

Σημείωση: Η τιμή του πεδίου γεωμετρίας αποτελείται από την ακολουθία συντεταγμένων των σημείων που ορίζουν το πολύγωνο. Η πρώτη τιμή από αυτά τα σημεία (εδώ 19252337.28 4924473.456) επαναλαμβάνεται και στο τέλος της ακολουθίας. Με αυτό τον τρόπο δηλώνεται ότι κλείνει το πολύγωνο.

Αν θέλουμε να διαγράψουμε έναν πίνακα από αυτούς που δημιουργήσαμε μπορούμε να το κάνουμε

σε γραφικό περιβάλλον pgAdminIII κάνοντας δεξί κλικ πάνω στον πίνακα και επιλέγοντας “Delete/Drop”. Ας διαγράψουμε λοιπόν τον πίνακα hospitals προκειμένου στην συνέχεια να τον εισάγουμε με διαφορετική μεθοδολογία.



Η ίδια λειτουργία μπορεί να γίνει και με εντολή SQL. Ας διαγράψουμε τον πίνακα “municipal”:

```
DROP TABLE health_services.municipal
```

Θα επανεισάγουμε αυτό τον πίνακα με άλλη μεθοδολογία σε παράδειγμα που θα ακολουθήσει.

Αν θέλουμε να διαγράψουμε όλα τα δεδομένα ενός πίνακα, αλλά όχι τον ίδιο τον πίνακα τότε μπορούμε να χρησιμοποιήσουμε την ακόλουθη εντολή:

```
DELETE FROM health_services.municipal
```


5.1.4 Εισαγωγή δεδομένων από αρχείο *.shp

5.1.4.1 Εισαγωγή σε περιβάλλον command line

Μαζί με το πρόγραμμα PostGIS παρέχεται και ένα πρόγραμμα γραμμής εντολών (command line tool) που επιτρέπει την μετατροπή δεδομένων *.shp σε πίνακες της χωρικής βάσης δεδομένων PostgreSQL/PostGIS. Το πρόγραμμα αυτό ονομάζεται shp2pgsql.

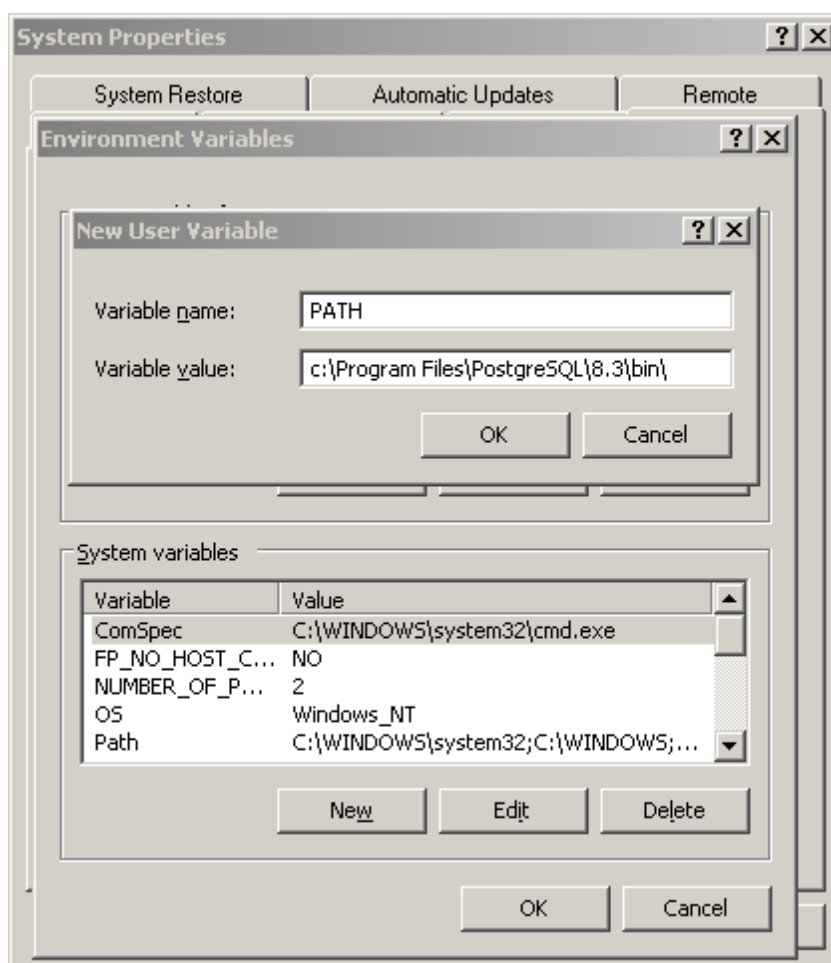
Έστω λοιπόν ότι θέλουμε να εισάγουμε στην χωρική βάση δεδομένων σημειακά δεδομένα που περιέχονται στο αρχείο hospitals.shp και αφορούν στα νοσοκομεία που βρίσκονται στην πόλη της Θεσσαλονίκης.

Αρχικά δημιουργούμε μία μεταβλητή συστήματος με το όνομα PATH και η και η οποία “δείχνει” στο φάκελο bin της PostgreSQL.

Επιλέγουμε “Start”, “My computer”, “Properties”, “Advanced”, “Environment Variables”, και στο επάνω τμήμα του παραθύρου όπου περιέχονται οι μεταβλητές χρήστη επιλέγουμε “New”,

Variable name: PATH

Variable value: c:\Program Files\PostgreSQL\8.3\bin\



και OK.

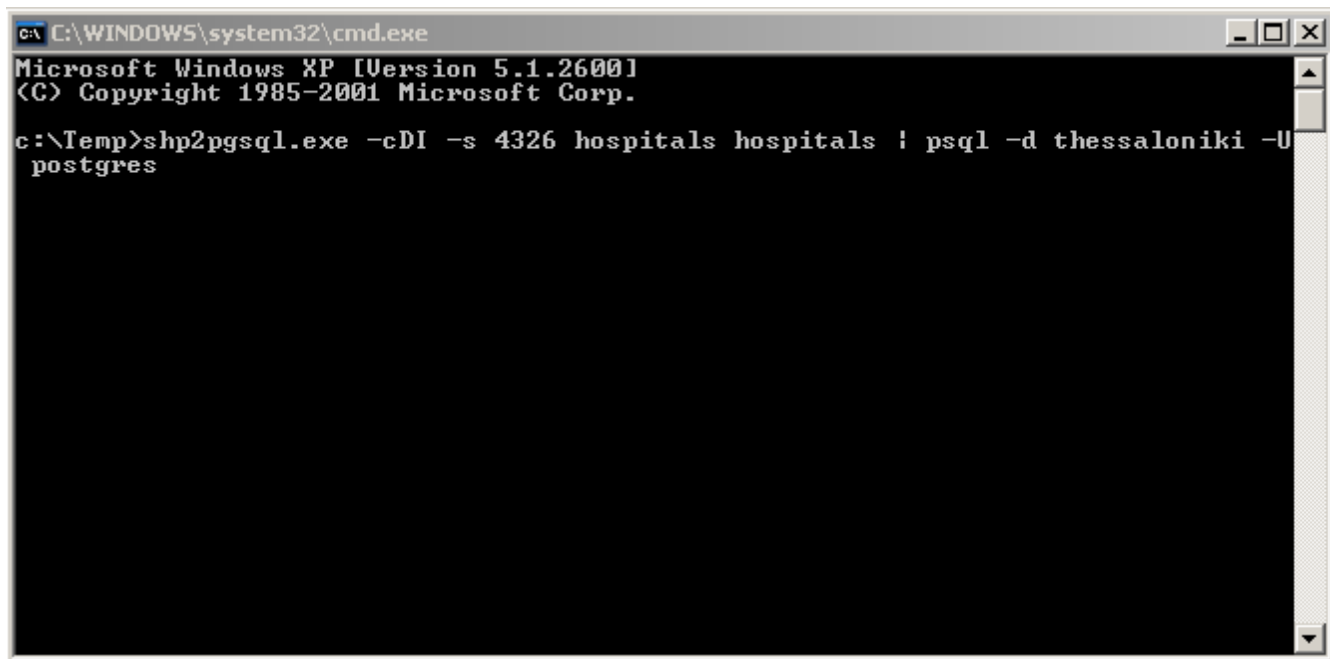
Στην συνέχεια ανοίγουμε ένα παράθυρο γραμμής εντολών και πηγαίνουμε στον φάκελο όπου

βρίσκεται το αρχείο hospitals.shp. Έστω ότι ο φάκελος αυτός είναι ο c:\Temp\.

Σημείωση: Τα αρχεία *.shp (shapefiles) είναι αρχεία που αποτελούνται από επιμέρους αρχεία και έτσι όταν “βλέπουμε” μία τοποθεσία με κάποιο μη εξειδικευμένος πρόγραμμα τότε βλέπουμε τα επιμέρους αρχεία. Για παράδειγμα, ανοίξουμε την τοποθεσία C:\temp με ένα πρόγραμμα που αναγνωρίζει τέτοιου τύπου αρχεία βλέπουμε ένα αρχείο, πχ το hospitals.shp. Αν ανοίξουμε την ίδια τοποθεσία με ένα πρόγραμμα που δεν αναγνωρίζει τέτοιου τύπου αρχεία τότε βλέπουμε κάποια επιμέρους αρχεία με επεκτάσεις *.shp, *.prj *.dbf, *.cpg, *.shx και *.shx. Όλα αυτά τα αρχεία έχουν το ίδιο όνομα, δηλαδή hospitals.

Αφού ανοίξαμε το παράθυρο εντολών και προσδιορίσαμε την τοποθεσία όπου βρίσκονται τα δεδομένα μας, πληκτρολογούμε την εντολή:

```
shp2pgsql.exe -cDI -s4121 -W UTF-8 hospitals hospitals | psql -d thessaloniki -U postgres
```



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

c:\Temp>shp2pgsql.exe -cDI -s 4326 hospitals hospitals | psql -d thessaloniki -U
postgres
```

και εκτελούμε την εντολή πατώντας enter

```
c:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

c:\Temp>shp2pgsql.exe -cDI -s 4326 hospitals hospitals | psql -d thessaloniki -U
postgres
Shapefile type: Point
Postgis type: POINT[2]
BEGIN
NOTICE: CREATE TABLE will create implicit sequence "hospitals_gid_seq" for seri
al column "hospitals.gid"
NOTICE: CREATE TABLE / PRIMARY KEY will create implicit index "hospitals_pkey"
for table "hospitals"
CREATE TABLE
        addgeometrycolumn
-----
public.hospitals.the_geom SRID:4326 TYPE:POINT DIMS:2
(1 row)

CREATE INDEX
COMMIT
c:\Temp>
```

Η εισαγωγή των δεδομένων που βρίσκονται σε μορφή *.shp έχει ολοκληρωθεί και ένας πίνακας με το όνομα hospitals βρίσκεται πλέον στην βάση δεδομένων.

Ας εξηγήσουμε πως συντάσσεται η εντολή “shp2pgsql.exe -cDI -s 4121 hospitals hospitals | psql -d thessaloniki -U postgres”:

Το πρώτο κομμάτι της εντολής (shp2pgsql.exe -cDI -s 4121 hospitals hospitals) δημιουργεί εντολή SQL για την εισαγωγή των δεδομένων στην βάση δεδομένων.

Το “-c” δημιουργεί έναν νέο πίνακα. Αν ο χρήστης βάζει τα δεδομένα σε έναν πίνακα που ήδη υπάρχει τότε μπορεί να πλητρωλογήσει “-a” αντί αυτού.

Το “-D” υποδεικνύει ότι το ενδιαμέσο αρχείο PostgreSQL θα έχει μορφή “dump” που επιτρέπει την γρηγορότερη εισαγωγή των στοιχείων στην βάση δεδομένων.

Το “-I” δημιουργεί δείκτη GiST στην στήλη της γεωμετρίας. Αν ο χρήστης εισάγει πολλαπλά αρχεία σε έναν πίνακα τότε μπορεί ο ίδιος να δημιουργήσει τον χωρικό δείκτη αφού ολοκληρώσει την εισαγωγή. Κάτι τέτοιο θα επιταχύνει την διαδικασία.

Το “-s” υποδεικνύει τον κωδικό γεωδαιτικού συστήματος (SRID) που ακολουθεί, εδώ 4121.

“hospitals” είναι το όνομα του αρχείου που εισάγεται.

Το δεύτερο “hospitals” είναι το όνομα που επιθυμούμε να έχει ο νέος πίνακας που θα δημιουργηθεί στην βάση δεδομένων.

Το δεύτερο τμήμα της εντολής (psql -d thessaloniki -U postgres) καλεί την διεπαφή psql της βάσης δεδομένων PostgreSQL και μέσω αυτής στέλνει την εντολή SQL στην βάση δεδομένων. Το “-d” προσδιορίζει ότι το όνομα που ακολουθεί είναι το όνομα της βάσης δεδομένων. Αντίστοιχα το “-U” προσδιορίζει ότι το όνομα που ακολουθεί είναι το όνομα του χρήστη.

Το μειονέκτημα που αντιμετωπίζουμε χρησιμοποιώντας αυτή τη μέθοδο είναι ότι ο νέος πίνακας δημιουργείται στο βασικό σχήμα της βάσης δεδομένων (public) και όχι στο σχήμα που δημιουργήσαμε. Αυτό μπορεί να αντιμετωπιστεί με την μετέπειτα αντιγραφή του πίνακα hospitals από το σχήμα public στο σχήμα health_services. Αυτό γίνεται με την χρήση της ακόλουθης εντολής SQL στο παράθυρο ερωτημάτων:

```
CREATE TABLE health_services.hospitals
AS (SELECT * FROM public.hospitals);
ALTER TABLE health_services.hospitals ADD CONSTRAINT hospitals_pkey PRIMARY KEY (gid);
CREATE INDEX health_services_idx ON health_services.hospitals USING GIST (the_geom);
DROP TABLE public.hospitals;
```

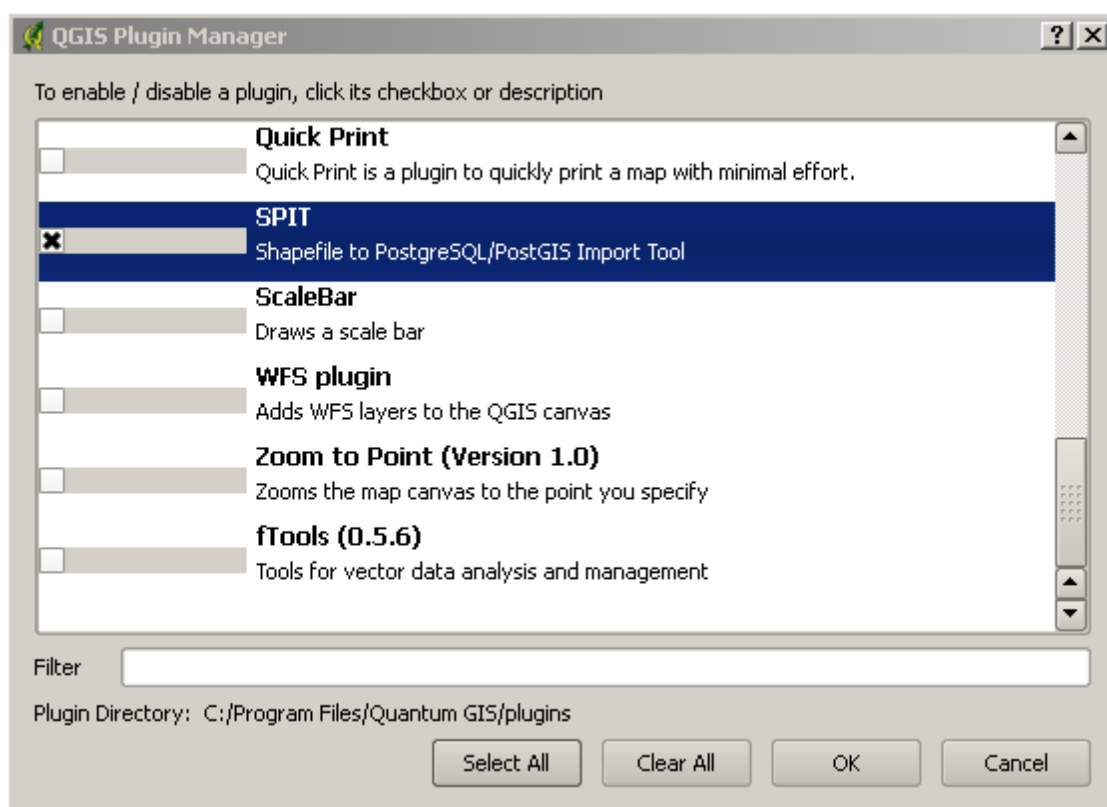
Σημείωση: Το πρόγραμμα shp2pgsql.exe έχει περισσότερες δυνατότητες από αυτή που παρουσιάζεται εδώ. Ανάμεσα σε αυτές θα πρέπει να αναφέρουμε την δυνατότητα δημιουργίας αρχείου *.sql με εντολές insert η αρχείου “dump” σε τοποθεσία του δίσκου που επιλέγει ο χρήστης. Τα αρχεία οποία μπορεί να φορτώσει “χειροκίνητα” ο χρήστης μέσω της διεπαφής psql ή του παραθύρου ερωτημάτων.

5.1.4.2 Εισαγωγή δεδομένων μέσω του προγράμματος QuantumGIS:

Το πρόγραμμα QuantumGIS παρέχει μία πολύ χρήσιμη διεπαφή για την εισαγωγή δεδομένων *.shp στην χωρική βάση δεδομένων PostgreSQL/PostGIS.

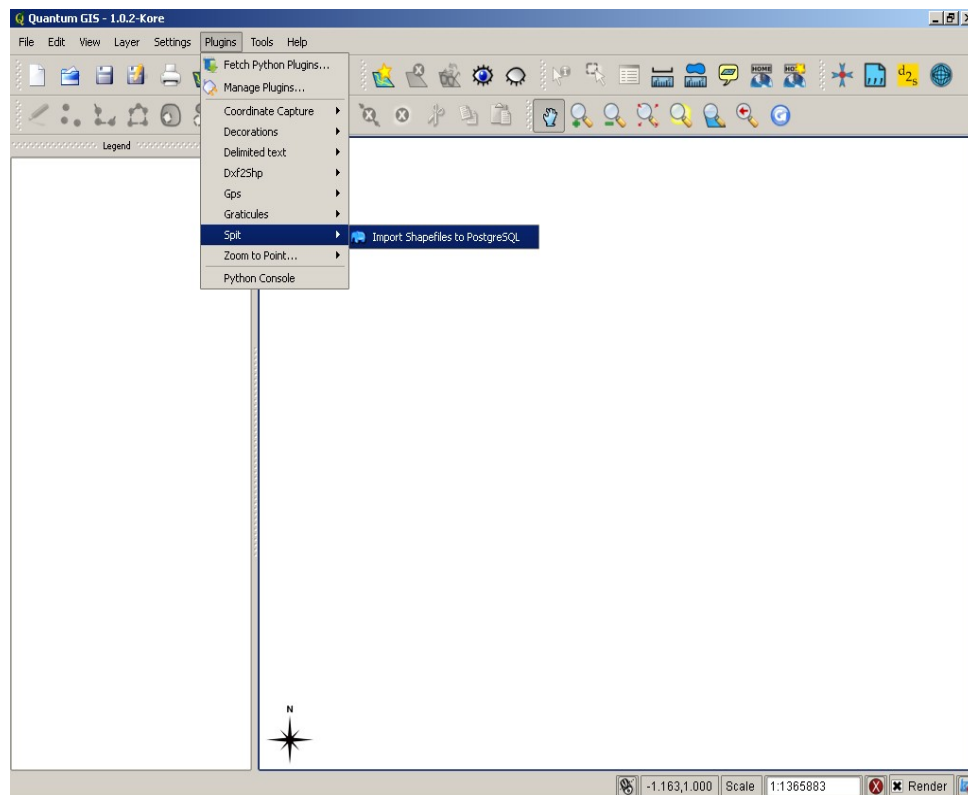
Ξεκινούμε το πρόγραμμα QuantumGIS επιλέγοντας “Start”, “All programs”, “QuantumGIS”, “QuantumGIS”.

Αρχικά βεβαιωνόμαστε ότι η επέκταση που εισαγάγει δεδομένα στην χωρική βάση δεδομένων PostgreSQL/PostGIS είναι ενεργοποιημένη. Από το μενού του προγράμματος (πάνω αριστερά) επιλέγουμε “Plugins”, “Manage plugins” και βεβαιωνόμαστε ότι η επέκταση με το όνομα “SPIT” είναι επιλεγμένη:

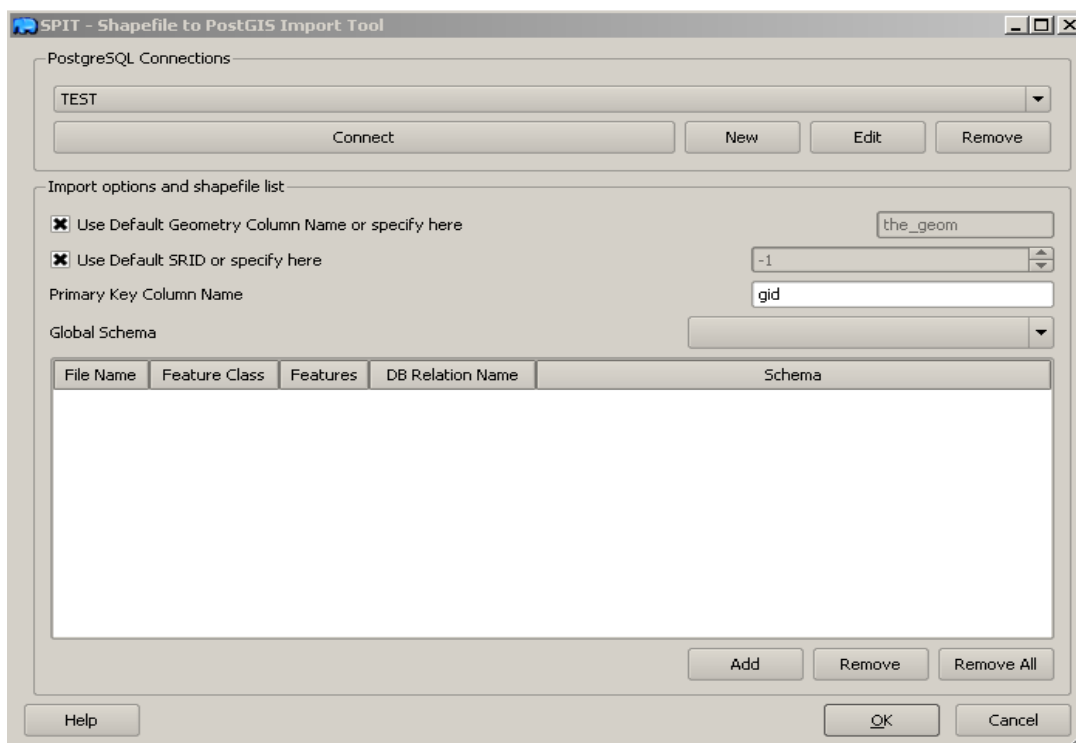


Αν δεν είναι επιλεγμένη την επιλέγουμε και πατάμε OK.

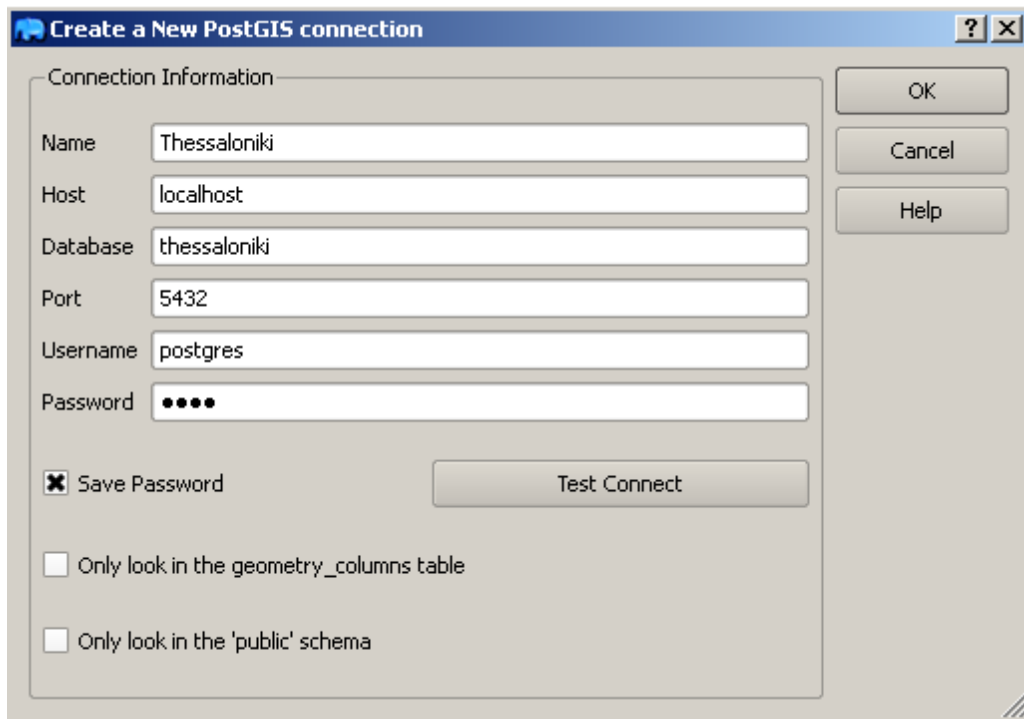
Στην συνέχεια επιλέγουμε από το μενού του προγράμματος “Plugins”, “SPIT”, “Import Shapefiles to PostgreSQL”.



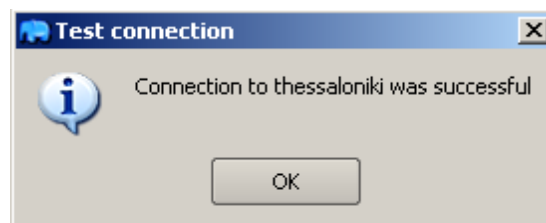
Στην συνέχεια εμφανίζεται το παράθυρο της εφαρμογής.



Αρχικά γίνεται σύνδεση με την βάση δεδομένων όπου επιθυμούμε να εισαχθεί το αρχείο *.shp. Ο χρήστης πρέπει να επιλέξει “New” και στην φόρμα που ακολουθεί εισάγει τις παρακάτω τιμές.



Πατώντας “Test Connect” ελέγχουμε την σύνδεση με την βάση δεδομένων. Αν έχουμε εισάγει τις σωστές τιμές η σύνδεση θα πρέπει να λειτουργήσει.

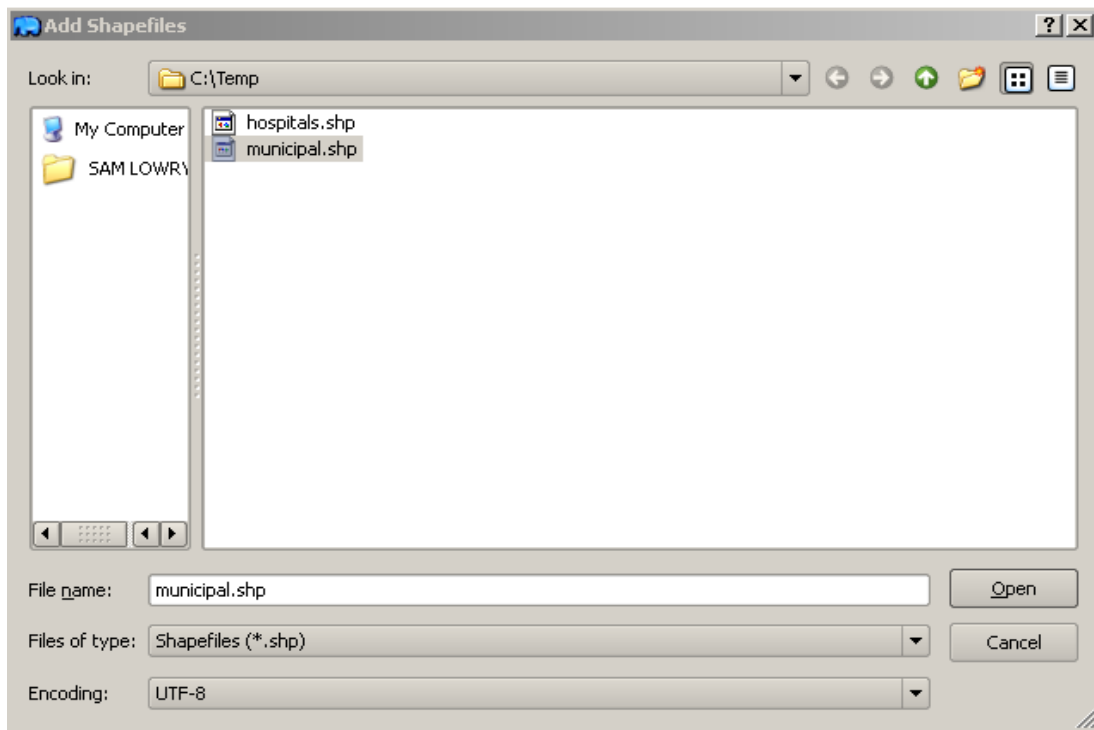


Πατάμε OK καθώς και στο παράθυρο “Create a new PostGIS connection”

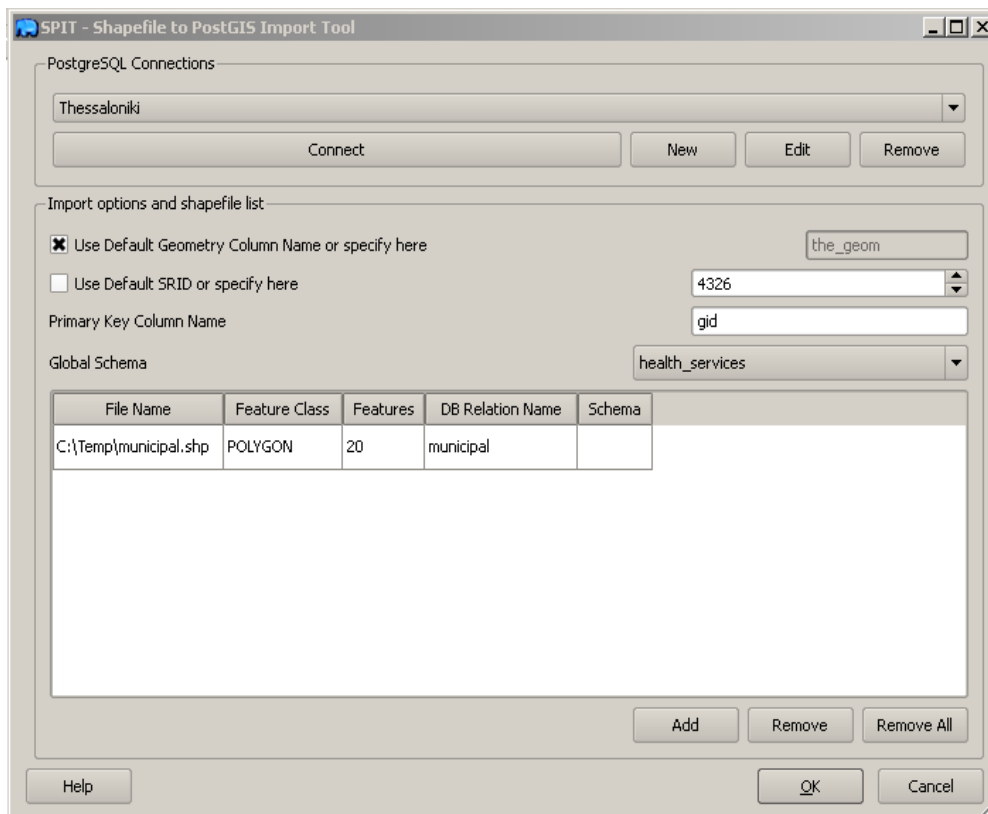
Από το παράθυρο της εφαρμογής SPIT επιλέγουμε την βάση δεδομένων “Thessaloniki” και επιλέγουμε “Connect”.

Στην συνέχεια από-επιλέγουμε το κουτί “Use Default SRID or specify here” και αντικαθιστούμε την τιμή “-1” με την τιμή 4121 που αντιστοιχεί στον κωδικό του γεωδαιτικού συστήματος στο οποίο βρίσκονται τα δεδομένα μας.

Στην συνέχεια επιλέγουμε “Add” και προσδιορίζουμε την τοποθεσία όπου έχουμε αποθηκεύσει τα δεδομένα που θα χρησιμοποιήσουμε. Ακόμα προσδιορίζουμε την συμβολοσειρά που πρέπει να χρησιμοποιηθεί (UTF-8).



Επιστρέφουμε στο παράθυρο της εφαρμογής SPIT και επιλέγουμε OK.



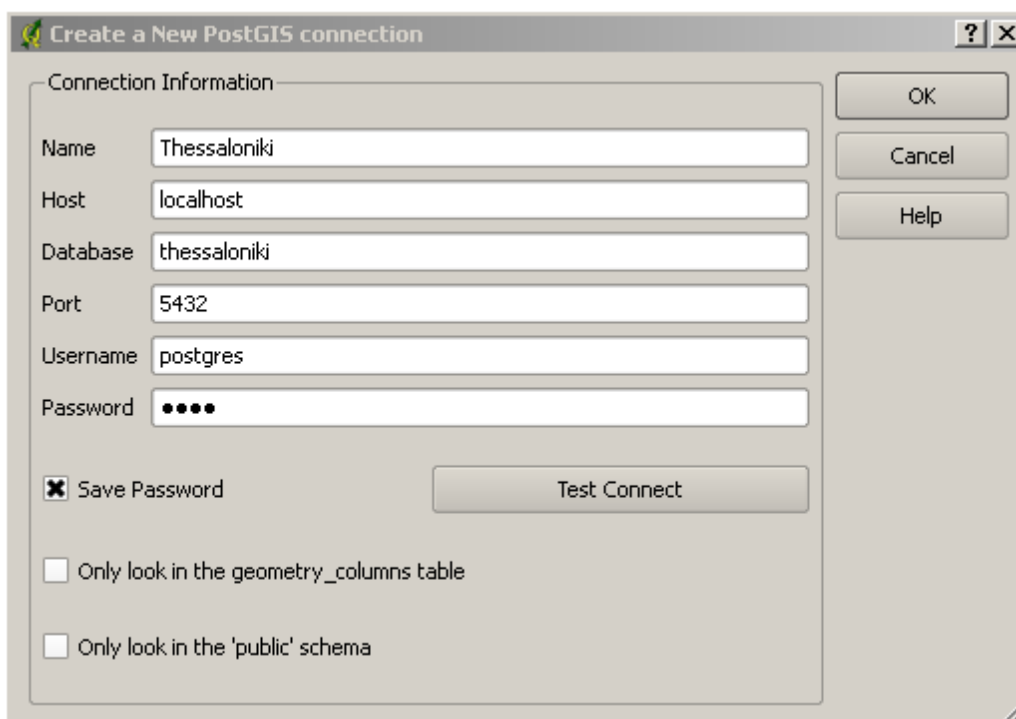
Το θεματικό επίπεδο των δήμων της Θεσσαλονίκης έχει πλέον εισαχθεί στην βάση δεδομένων.

5.1.4.3 Απεικόνιση δεδομένων στο QuantumGIS

Για να απεικονίσουμε τα δεδομένα που έχουμε εισάγει στην βάση δεδομένων στο λογισμικό QuantumGIS κάνουμε τα εξής:

Ξεκινάμε το QuantumGIS όπως κάναμε στο προηγούμενο βήμα.

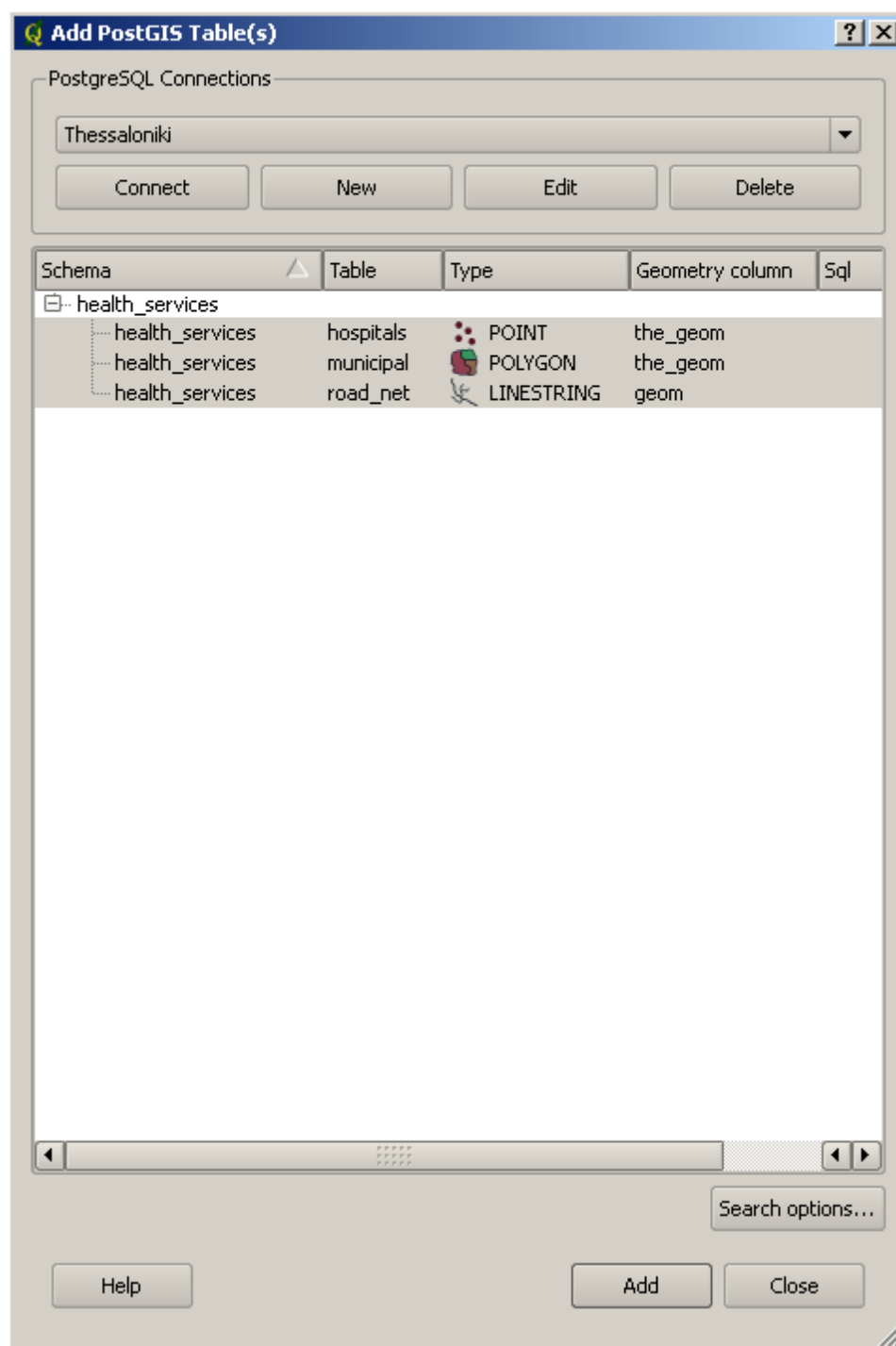
Στην συνέχεια από το μενού του προγράμματος επιλέγουμε “Layer”, “Add PostGIS Layer” και στο παράθυρο που εμφανίζεται επιλέγουμε “New”. Εκεί συμπληρώνουμε τα στοιχεία που χρειάζονται για να συνδεθεί το QuantumGIS με την PostgreSQL/PostGIS, όπως ακριβώς κάναμε και στο προηγούμενο βήμα.



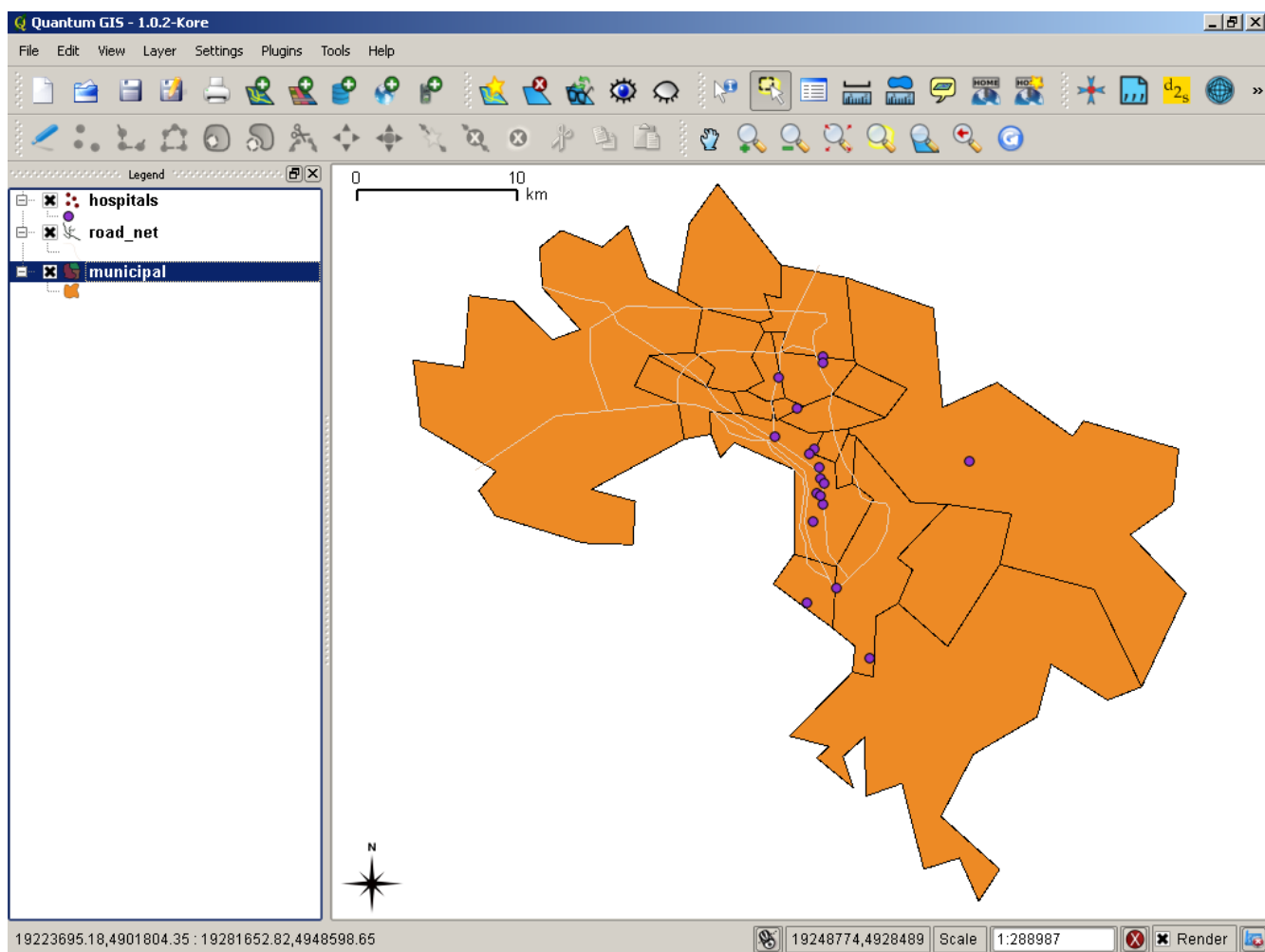
The screenshot shows a dialog box titled "Create a New PostGIS connection". It has a "Connection Information" section with the following fields: Name (Thessaloniki), Host (localhost), Database (thessaloniki), Port (5432), Username (postgres), and Password (masked with dots). There are three checkboxes: "Save Password" (checked), "Only look in the geometry_columns table" (unchecked), and "Only look in the 'public' schema" (unchecked). On the right side, there are buttons for "OK", "Cancel", and "Help". At the bottom right, there is a "Test Connect" button.

Στην συνέχεια επιλέγουμε “OK”.

Στο παράθυρο εισαγωγής δεδομένων προσδιορίζουμε το όνομα της βάσης δεδομένων όπου εργαζόμαστε και στην συνέχεια επιλέγουμε “Connect”. Τώρα πρέπει να εμφανιστούν οι πίνακες που περιέχονται στην χωρική βάση δεδομένων. Επιλέγουμε τον/τους πίνακες που θέλουμε να απεικονίσουμε και πατάμε “Add”.



Τώρα οι πίνακες της χωρικής βάσης δεδομένων που επιλέξαμε πρέπει να εμφανιστούν σαν θεματικά επίπεδα στο QuantumGIS.



5.1.4.4 Εξαγωγή δεδομένων της βάσης δεδομένων σε μορφή *.shp

Η PostgreSQL μας παρέχει την δυνατότητα να εξαγάμουμε δεδομένα πινάκων περιέχονται σε αυτή σε μορφή αρχείου *.shp. Αυτό γίνεται με την χρήση του εργαλείου γραμμής εντολών `pgsql2shp`. Ας δούμε ένα παράδειγμα: Έστω ότι θέλουμε να εξαγάμουμε τα δεδομένα οδικού δικτύου της Θεσσαλονίκης σε αρχείο *.shp και η τοποθεσία που θέλουμε να το αποθηκεύσουμε είναι ο φάκελος `c:\export`. Ανοίγουμε ένα παράθυρο γραμμής εντολών και το κατευθύνουμε σε αυτό τον φάκελο πληκτρολογώντας `c:\export`.

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

c:\export>
```

Στην συνέχεια πληκτρολογούμε την εντολή:

```
pgsql2shp -P5678 -u postgres thessaloniki health_services.hospitals
```

επιλέγουμε "enter" και η εξαγωγή ολοκληρώνεται στην επιλεγμένη τοποθεσία

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

c:\export>pgsql2shp -P5678 -u postgres thessaloniki health_services.hospitals
Initializing... Done (postgis major version: 1).
Output shape: Point
Dumping: XX [18 rows].

c:\export>
```

Η εντολή συντάσσεται ως εξής:

```
pgsql2shp -P5678 -u postgres thessaloniki health_services.hospitals
```

- **pgsql2shp**: καλείται το αρχείο pgsql2shp.exe
- **-P5678**: μετά το -P εισάγεται ο κωδικός χρήστη (βήμα 7, ενότητα: 4.1 ΧΒΔ)

PostgreSQL/PostGIS)

- -u postgres: μετά το -u προσδιορίζεται το όνομα χρήστη
- **thessaloniki**: προσδιορίζεται το όνομα της βάσης δεδομένων
- **health_services.hospitals**: προσδιορίζεται το σχήμα και το όνομα του πίνακα από τον οποίο θα αντληθούν τα στοιχεία.

5.1.5 Εισαγωγή πινάκων με μη γεωμετρικά δεδομένα

Συχνά παρουσιάζεται η ανάγκη να εισάγουμε στην βάση δεδομένων μας πίνακες με μη γεωμετρικά δεδομένα. Αυτοί οι πίνακες άλλοτε χρησιμεύουν στο να συνδέουν τους λοιπούς γεωμετρικούς πίνακες ή περιέχουν επιπρόσθετη πληροφορία για κάποιους από τους γεωμετρικούς πίνακες. Τέτοιοι πίνακες μπορούν φυσικά να εισαχθούν με διαδοχικές εντολές CREATE TABLE και στην συνέχεια διαδοχικές εντολές INSERT. Κάτι τέτοιο όμως δεν είναι πάντα εύκολο αφού οι πίνακες μπορεί να είναι μεγάλοι σε μέγεθος πράγμα που κάνει την εισαγωγή τους με την παραπάνω μεθοδολογία δύσκολο. Επιπλέον αν κάποιος προσπαθήσει να εισαγάγει πίνακες με μη γεωμετρικά δεδομένα με χρήση του προγράμματος Quantum GIS τότε θα διαπιστώσει ότι αυτό δεν είναι δυνατό αφού το πρόγραμμα του επιστρέφει σφάλμα.

Ας υποθέσουμε ότι θέλουμε να εισάγουμε στην βάση δεδομένων έναν μη γεωμετρικό πίνακα σε μορφή *.dbf με πρόσθετες πληροφορίες για κάποιο από τα υπάρχοντα θεματικά επίπεδα, λχ με λατινικά ονόματα για τα στοιχεία του σημειακού επιπέδου "Νοσοκομεία". Ο πίνακας αυτός θα πρέπει αναγκαστικά να περιέχει ένα γνώρισμα που να το συνδέει με τις εγγραφές του πίνακα με τον οποίο επιθυμούμε να τον συνδέσουμε (ξένο κλειδί), εδώ το γνώρισμα ID καθώς και το πεδίο με την επιπρόσθετη πληροφορία πχ. NAME. Ο πίνακας έχει την ακόλουθη μορφή.

	A	B
1	200	PATHOLOGIKO NOSOKOMEIO NEAPOLIS
2	201	GENIKO NOSOKOMEIO AGHOIS DIMITRIOS
3	202	PATHOLOGIKO NOSOKOMEIO THESSALONIKIS
4	203	GENIKO NOSOKOMEIO ACHEPA
5	204	GENIKO NOSOKOMEIO POLIXNIS PAPAGEORGIOU
6	205	STARTIOTIKO NOSOKOMEIO POLIXNIS 424
7	206	GENIKO NOSOKOMEIAKO IATRIKO DIAVALKANIKO KENTRO
8	207	GENIKO NOSOKOMEIO THESSALONIKIS AGHIOS PAULOS
9	208	IDIOTIKI KLINIKI PANAGIA KALAMARIA
10	209	GENIKO NOSOSKOMEIO THESSALONIKIS GEWRGIOS GENNIMATAS
11	210	GENIKO NOSOKOMEIO IPPOKRATEIO
12	211	GENIKO NOSOKOMEIO THEAGENEIO
13	212	IDIOTIKI KLINIKI DODEKA APOSTOLOI
14	213	GENIKO NOSOKOMEIO GEORGIOS PAPANIKOLAOU
15	214	IDIOTIKI KLINIKI EVAGGELISMOS
16	215	STRATIOTIKO NOSOKOMEIO THESSALONIKIS 424
17	216	PATHOLOGIKO NOSOKOMEIO EIDIKON PATHISEON
18	217	PSICHIATRIKO NOSOKOMEIO THESSALONIKIS
19		

Αρχικά θα μετατρέψουμε τον πίνακα που θέλουμε να εισάγουμε σε αρχείο τύπου *.csv (αρχείο κειμένου όπου οι τιμές των διαφορετικών γνωρισμάτων διαχωρίζονται με κόμματα). Αυτό μπορούμε να το κάνουμε με κάποιο πρόγραμμα ανοικτού λογισμικού όπως το Open Office/scalc. Ανοίγουμε τον πίνακα, φροντίζουμε να διαγράψουμε τυχόν εγγραφές με ονόματα πεδίων και αποθηκεύουμε (File → Save as → Text CSV (.csv)) σε γνωστή τοποθεσία, για παράδειγμα:

c:/TEMP/hospital_names_latin.csv .

Στην συνέχεια δημιουργούμε έναν άδειο πίνακα στην βάση δεδομένων με τα γνωρίσματα του πίνακα που επιθυμούμε να εισάγουμε.

Χρησιμοποιούμε την ακόλουθη εντολή:

```
create table hospitals_latin
(ID NUMERIC PRIMARY KEY,
NAME VARCHAR);
```

Στην συνέχεια αντιγράφουμε τα περιεχόμενα του πίνακα *.csv που δημιουργήσαμε νωρίτερα στον νέο πίνακα της βάσης δεδομένων με την ακόλουθη εντολή.

```
copy hospitals_latin from 'c:/TEMP/hospital_names_latin.csv' DELIMITERS ',' CSV;
```

Ο πίνακας με τα μη γεωγραφικά δεδομένα πρέπει να έχει μεταφερθεί με επιτυχία στην βάση δεδομένων.

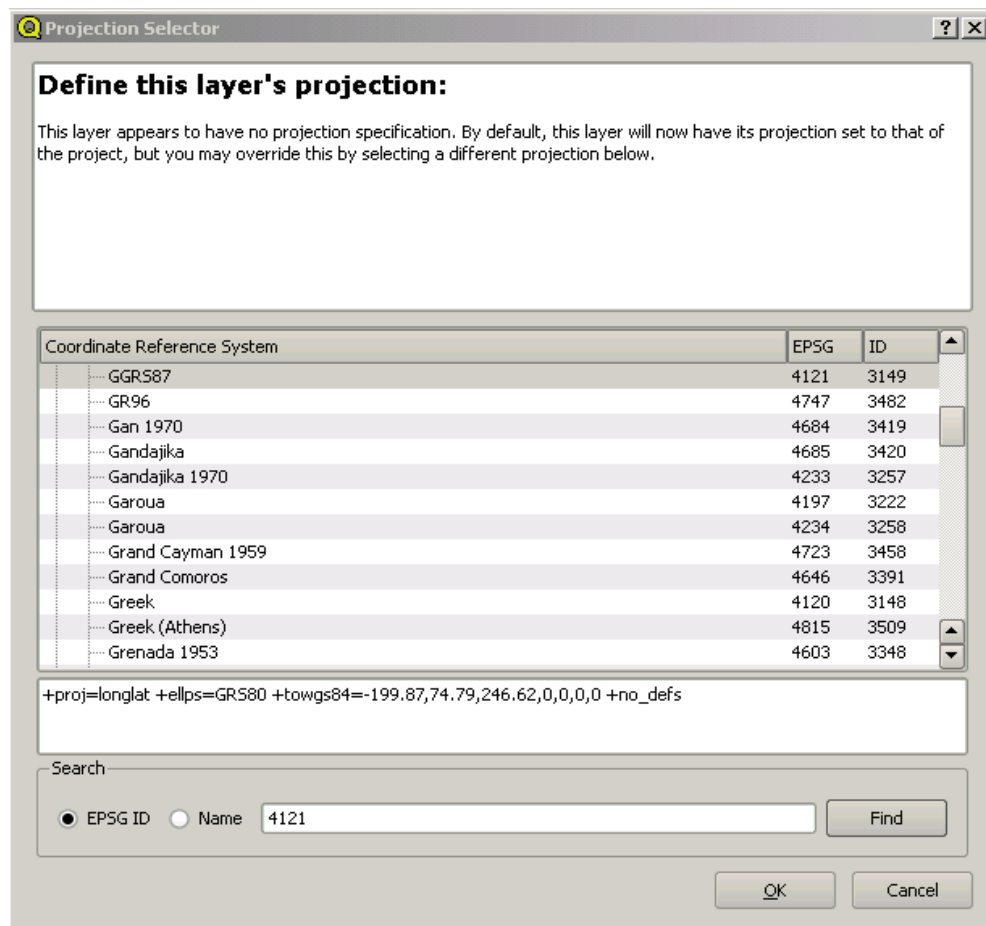
5.1.6 Αλλαγή συστήματος συντεταγμένων

Τα δεδομένα που βρίσκονται αποθηκευμένα στην βάση δεδομένων που έχουμε δημιουργήσει βρίσκονται σε κάποιο σύστημα συντεταγμένων, το οποίο έχουμε ορίσει κατά την εισαγωγή του πλέδιου γεωμετρίας του εκάστοτε πίνακα. Για παράδειγμα τα δεδομένα που έχουμε εισάγει μέχρι στιγμής βρίσκονται στο Ελληνικό Γεωδαιτικό Σύστημα Αναφοράς 1987 (ΕΓΣΑ87). Έστω ότι θέλουμε να αλλάξουμε το γεωδαιτικό σύστημα στο οποίο βρίσκονται σε Παγκόσμιο Γεωδαιτικό Σύστημα Αναφοράς 1984 (WGS84). Αρχικά πρέπει να γνωρίζουμε ποιος είναι ο κωδικός (SRID – Spatial Reference ID) με τον οποίο αναγνωρίζει αυτά τα συστήματα η βάση δεδομένων. Ο κωδικός που χρησιμοποιήσαμε για να προσδιορίσουμε το ΕΓΣΑ87 κατά την εισαγωγή των δεδομένων είναι 4121. Ο κωδικός για το WGS84 είναι 4326. Αν δεν γνωρίζουμε ποιος είναι ο κωδικός του συστήματος που μας ενδιαφέρει μπορούμε να τον αναζητήσουμε στον πίνακα `spatial_ref_sys` που βρίσκεται στο σχήμα `public` κάθε βάσης δεδομένων που έχει ενισχύθεί με χωρικές λειτουργίες.

Edit Data - PostgreSQL Database Server 8.3 (localhost:5432) - thessaloniki - spatial_ref_sys				
File Edit View Help				
No limit				
	sr_id	auth_name	auth_srid	srttext
	[PK] integer	character varying(255)	integer	character varying(2048)
1892	4051	EPSG	4051	PROJCS["RGDC 2005 / Congo TM zone 18",GEOGCS["RGDC 2005",DATUM["Reseau_Geodesique_de_la_RDC_2005",SPHEROID["GRS 1",+proj=tmcr +lat_0=0 +lon_0=1
1893	4052	EPSG	4052	GEOGCS["Unspecified datum based upon the Clarke 1866 Authalic Sphere",DATUM["Not_specified_based_on_Clarke_1866_Authalic_Sphe",+proj=longlat +a=6370997 +b=
1894	4053	EPSG	4053	GEOGCS["Unspecified datum based upon the International 1924 Authalic Sphere",DATUM["Not_specified_based_on_International_1924_",+proj=longlat +a=6371228 +b=
1895	4054	EPSG	4054	GEOGCS["Unspecified datum based upon the Hughes 1980 ellipsoid",DATUM["Not_specified_based_on_Hughes_1980_ellipsoid",SPHEROID["+proj=longlat +a=6378273 +b=
1896	4055	EPSG	4055	GEOGCS["Popular Visualisation CRS",DATUM["Popular_Visualisation_Datum",SPHEROID["Popular_Visualisation_Sphere",6378137,0,AUTHOR +proj=longlat +a=6378137 +b=
1897	4056	EPSG	4056	PROJCS["RGDC 2005 / Congo TM zone 20",GEOGCS["RGDC 2005",DATUM["Reseau_Geodesique_de_la_RDC_2005",SPHEROID["GRS 1",+proj=tmcr +lat_0=0 +lon_0=2
1898	4057	EPSG	4057	PROJCS["RGDC 2005 / Congo TM zone 22",GEOGCS["RGDC 2005",DATUM["Reseau_Geodesique_de_la_RDC_2005",SPHEROID["GRS 1",+proj=tmcr +lat_0=0 +lon_0=2
1899	4058	EPSG	4058	PROJCS["RGDC 2005 / Congo TM zone 24",GEOGCS["RGDC 2005",DATUM["Reseau_Geodesique_de_la_RDC_2005",SPHEROID["GRS 1",+proj=tmcr +lat_0=0 +lon_0=2
1900	4059	EPSG	4059	PROJCS["RGDC 2005 / Congo TM zone 26",GEOGCS["RGDC 2005",DATUM["Reseau_Geodesique_de_la_RDC_2005",SPHEROID["GRS 1",+proj=tmcr +lat_0=0 +lon_0=2
1901	4060	EPSG	4060	PROJCS["RGDC 2005 / Congo TM zone 28",GEOGCS["RGDC 2005",DATUM["Reseau_Geodesique_de_la_RDC_2005",SPHEROID["GRS 1",+proj=tmcr +lat_0=0 +lon_0=2
1902	4061	EPSG	4061	PROJCS["RGDC 2005 / UTM zone 33S",GEOGCS["RGDC 2005",DATUM["Reseau_Geodesique_de_la_RDC_2005",SPHEROID["GRS 1980",+proj=utm +zone=33 +south +t
1903	4062	EPSG	4062	PROJCS["RGDC 2005 / UTM zone 34S",GEOGCS["RGDC 2005",DATUM["Reseau_Geodesique_de_la_RDC_2005",SPHEROID["GRS 1980",+proj=utm +zone=34 +south +t
1904	4063	EPSG	4063	PROJCS["RGDC 2005 / UTM zone 35S",GEOGCS["RGDC 2005",DATUM["Reseau_Geodesique_de_la_RDC_2005",SPHEROID["GRS 1980",+proj=utm +zone=35 +south +t
1905	4071	EPSG	4071	PROJCS["Chua / UTM zone 23S",GEOGCS["Chua",DATUM["Chua",SPHEROID["International 1924",6378388,297,AUTHORITY["EPSG",702],+proj=utm +zone=23 +south +t
1906	4075	EPSG	4075	GEOGCS["SREF98",DATUM["Serbian_Reference_Network_1998",SPHEROID["GRS 1980",6378137,298.257222101,AUTHORITY["EPSG",7],+proj=longlat +ellps=GRS80 +to
1907	4081	EPSG	4081	GEOGCS["REGCAN95",DATUM["Red_Geodesica_de_Canarias_1995",SPHEROID["GRS 1980",6378137,298.257222101,AUTHORITY["EPSG",+proj=longlat +ellps=GRS80 +to
1908	4082	EPSG	4082	PROJCS["REGCAN95 / UTM zone 27N",GEOGCS["REGCAN95",DATUM["Red_Geodesica_de_Canarias_1995",SPHEROID["GRS 1980",63781 +proj=utm +zone=27 +ellps=GR
1909	4083	EPSG	4083	PROJCS["REGCAN95 / UTM zone 28N",GEOGCS["REGCAN95",DATUM["Red_Geodesica_de_Canarias_1995",SPHEROID["GRS 1980",63781 +proj=utm +zone=28 +ellps=GR
1910	4120	EPSG	4120	GEOGCS["Greek",DATUM["Greek",Bessel 1841",6377397.155,299.1528128,AUTHORITY["EPSG",7004]],AUTHORITY["EPSG",+proj=longlat +ellps=bessel +to
1911	4121	EPSG	4121	GEOGCS["GGRS87",DATUM["Greek_Geodetic_Reference_System_1987",SPHEROID["GRS 1980",6378137,298.257222101,AUTHORITY["EPSG",+proj=longlat +ellps=GRS80 +to
1912	4122	EPSG	4122	GEOGCS["ATS77",DATUM["Average_Terrestrial_System_1977",SPHEROID["Average Terrestrial System 1977",6378135,298.257,AUTHOR +proj=longlat +a=6378135 +b=
1913	4123	EPSG	4123	GEOGCS["KKJ",DATUM["Kartastokoordinaattijarjestelma_1966",SPHEROID["International 1924",6378388,297,AUTHORITY["EPSG",7022],+proj=longlat +ellps=intl +no_de
1914	4124	EPSG	4124	GEOGCS["RI90",DATUM["Rikets_koordinatsystem_1990",SPHEROID["Bessel 1841",6377397.155,299.1528128,AUTHORITY["EPSG",7004],+proj=longlat +ellps=bessel +no
1915	4125	EPSG	4125	GEOGCS["Sanboja",DATUM["Sanboja",SPHEROID["Bessel 1841",6377397.155,299.1528128,AUTHORITY["EPSG",7004]],TOWGS84[-40,+proj=longlat +ellps=bessel +to
1916	4126	EPSG	4126	GEOGCS["LKS94 (ETRS89)",DATUM["Lithuania_1994_ETRS89",SPHEROID["GRS 1980",6378137,298.257222101,AUTHORITY["EPSG",701,+proj=longlat +ellps=GRS80 +no
1917	4127	EPSG	4127	GEOGCS["Tete",DATUM["Tete",SPHEROID["Clarke 1866",6378206.4,294.9786982139006,AUTHORITY["EPSG",7008]],AUTHORITY["EPS",+proj=longlat +ellps=chr66 +no
1918	4128	EPSG	4128	GEOGCS["Madzansua",DATUM["Madzansua",SPHEROID["Clarke 1866",6378206.4,294.9786982139006,AUTHORITY["EPSG",7008]],AUT +proj=longlat +ellps=chr66 +no
1919	4129	EPSG	4129	GEOGCS["Observatorio",DATUM["Observatorio",SPHEROID["Clarke 1866",6378206.4,294.9786982139006,AUTHORITY["EPSG",7008]],+proj=longlat +ellps=chr66 +no

Αυτή την πληροφορία μπορούμε να βρούμε στο διαδίκτυο αλλά και στο Quantum GIS κάνοντας δεξί

κλίκ σε κάποιο από τα θεματικά επίπεδα που βρίσκονται στο πάνελ και στην συνέχεια επιλέγοντας “change CRS”



Στην συνέχεια εκτελούμε την ακόλουθη εντολή SQL:

```
SELECT updateGeometrySRID('health_services', 'hospitals', 'the_geom', 4326);
```

```
UPDATE health_services.hospitals set the_geom =  
transform(SetSRID(the_geom, 4121), 4326);
```

Το πρώτο κομμάτι της εντολής που προηγήθηκε αλλάζει την σήμανση του συστήματος συντεταγμένων, εσωτερικά στον πίνακα μας. Το δεύτερο τμήμα της εντολής αλλάζει την γεωμετρία των δεδομένων έτσι ώστε να μετασχηματιστούν από το παλιό στο νέο σύστημα συντεταγμένων.

5.2 Λειτουργίες ανάκτησης δεδομένων

Η χωρική βάση δεδομένων PostgreSQL/PostGIS επιτρέπει την ανάκτηση του συνόλου ή μέρους των δεδομένων που βρίσκονται σε αυτή με την υποβολή ερωτημάτων. Η πιο απλή μορφή ερωτήματος είναι αυτή που μας επιστρέφει όλα τα δεδομένα που βρίσκονται σε έναν πίνακα. Έστω για παράδειγμα ότι θέλουμε να μας επιστραφούν όλα τα δεδομένα που βρίσκονται σε κάποιον πίνακα, πχ στον πίνακα που περιέχει τα νοσοκομεία. Το ερώτημα που επιστρέφει το σύνολο του πίνακα είναι το εξής:

```
SELECT * FROM health_services.hospitals;
```

εκτελώντας αυτό το ερώτημα μας επιστρέφεται (στο κάτω μέρος του παραθύρου ερωτημάτων) ολόκληρος ο πίνακας που περιέχει τα νοσοκομεία.

	gid integer	id bigint	name character varying(200)	the_geom geometry
1	1	200	ΠΑΘΟΛΟΓΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΕΙΔΙΚΩΝ ΠΑΘΗΣΕΩΝ	0101000020E61
2	2	201	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ "ΑΓΙΟΣ ΔΗΜΗΤΡΙΟΣ"	0101000020E61
3	3	202	ΠΑΘΟΛΟΓΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ ΑΦΡΟΔΙΣΙΩΝ/ΕΙΔΙΚΩΝ ΝΟΣΗΜΑΤΩΝ	0101000020E61
4	4	203	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΧΕΠΑ	0101000020E61
5	5	204	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΠΟΛΙΣΚΗΣ ΠΑΠΑΓΕΩΡΓΙΟΥ	0101000020E61
6	6	205	ΣΤΡΑΤΙΩΤΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΠΟΛΥΓΩΝΗΣ 424	0101000020E61
7	7	206	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ/ΙΑΤΡΙΚΟ ΔΙΑΒΑΛΚΑΝΙΚΟ ΚΕΝΤΡΟ	0101000020E61
8	8	207	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ "ΑΓΙΟΣ ΠΑΥΛΟΣ"	0101000020E61
9	9	208	ΙΔΙΩΤΙΚΗ ΚΛΙΝΙΚΗ ΠΑΝΑΓΙΑ ΚΑΛΑΜΑΡΙΑ	0101000020E61
10	10	209	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ ΓΕΩΡΓΙΟΣ ΓΕΝΝΗΜΑΤΑΣ	0101000020E61
11	11	210	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΙΠΠΟΚΡΑΤΕΙΟ	0101000020E61
12	12	211	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΑΓΕΝΕΙΟ	0101000020E61
13	13	212	ΙΔΙΩΤΙΚΗ ΚΛΙΝΙΚΗ 12 ΑΠΟΣΤΟΛΩΙ	0101000020E61
14	14	213	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΓΕΩΡΓΙΟΣ ΠΑΠΑΝΙΚΟΛΑΟΥ	0101000020E61
15	15	214	ΙΔΙΩΤΙΚΗ ΚΛΙΝΙΚΗ ΕΥΑΓΓΕΛΙΣΜΟΣ	0101000020E61
16	16	215	ΣΤΡΑΤΙΩΤΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ 424	0101000020E61
17	17	216	ΠΑΘΟΛΟΓΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ	0101000020E61
18	18	217	ΨΥΧΙΑΤΡΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ	0101000020E61

Από την στιγμή που τα ερωτήματα δεν μας επιστρέφουν το σύνολο αλλά μέρος των δεδομένων που βρίσκεται σε έναν πίνακα διακρίνουμε δύο κατηγορίες ερωτημάτων ανάλογα με το αν βασίζονται σε χωρικά ή μη κριτήρια και τελεστές. Παρακάτω θα αναπτύξουμε τις δύο αυτές κατηγορίες ερωτημάτων.

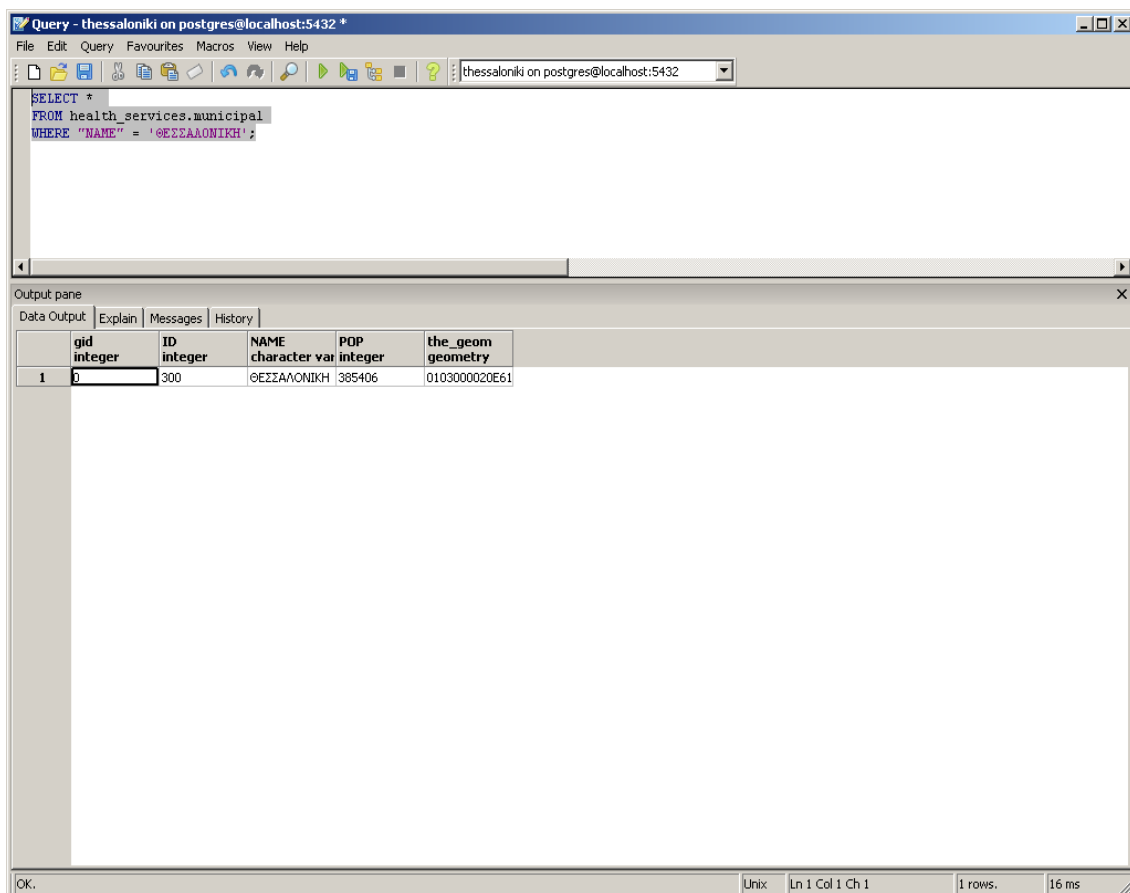
5.2.1 Μη χωρικά ερωτήματα SQL στο περιβάλλον της βάσης δεδομένων

Τα μη χωρικά ερωτήματα αφορούν σε μη χωρικά χαρακτηριστικά των δεδομένων, δηλαδή σε χαρακτηριστικά που βρίσκονται αποθηκευμένα σε όλες τις στήλες εκτός από αυτήν της γεωμετρίας. Για παράδειγμα:

Ποιες εγγραφές από τον πίνακα δήμων έχουν όνομα "ΘΕΣΣΑΛΟΝΙΚΗ;"

```
SELECT *  
FROM health_services.municipal  
WHERE "NAME" = 'ΘΕΣΣΑΛΟΝΙΚΗ';
```

Πράγματι, στο κάτω μέρος του παραθύρου ερωτημάτων μας επιστρέφεται η εγγραφή που αφορά στον δήμο Θεσσαλονίκης



Αντίστοιχα μπορούμε να ρωτήσουμε την βάση δεδομένων ποιοι είναι αυτοί οι δήμοι οι οποίοι έχουν πληθυσμό μικρότερο από 40.000 κατοίκους

```
SELECT *
FROM health_services.municipal
WHERE "POP" < 40000;
```

ή ποιοι είναι αυτοί οι δήμοι που έχουν πληθυσμό μικρότερο από 40,000 και μεγαλύτερο από 10,000 κατοίκους

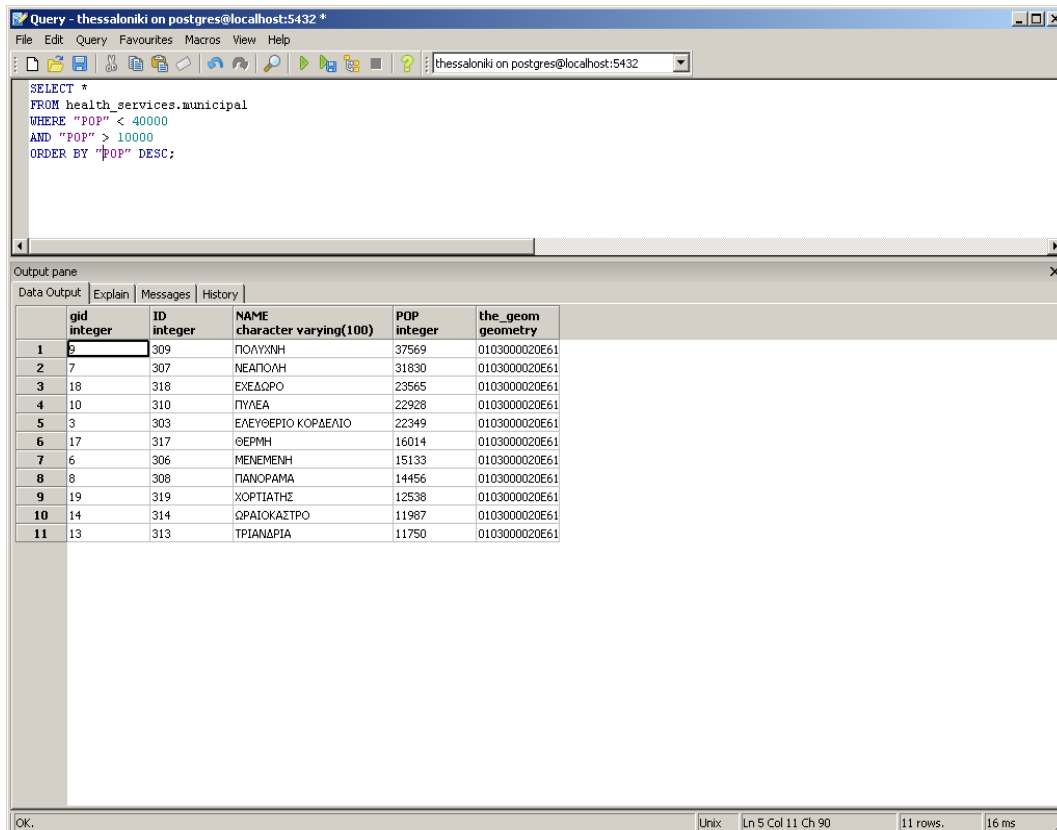
```
SELECT *
FROM health_services.municipal
WHERE "POP" < 40000
AND "POP" > 10000;
```

και ακόμα να ζητήσουμε από την βάση δεδομένων να μας επιστρέψει τα αποτελέσματα κατά αύξουσα αλφαβητική σειρά

```
SELECT *
FROM health_services.municipal
WHERE "POP" < 40000
AND "POP" > 10000
ORDER BY "NAME" ASC;
```

ή κατά φθίνουσα πληθυσμιακή σειρά


```
SELECT *
FROM health_services.municipal
WHERE "POP" < 40000
AND "POP" > 10000
ORDER BY "POP" DESC;
```



	gid integer	ID integer	NAME character varying(100)	POP integer	the_geom geometry
1	5	309	ΠΟΛΥΧΩΗ	37569	0103000020E61
2	7	307	ΝΕΑΠΟΛΗ	31830	0103000020E61
3	18	318	ΕΧΕΔΑΦΟ	23565	0103000020E61
4	10	310	ΠΥΛΕΑ	22928	0103000020E61
5	3	303	ΕΛΕΥΘΕΡΙΟ ΚΟΡΔΕΛΙΟ	22349	0103000020E61
6	17	317	ΘΕΡΜΗ	16014	0103000020E61
7	6	306	ΜΕΝΕΜΕΝΗ	15133	0103000020E61
8	8	308	ΠΑΝΟΡΑΜΑ	14456	0103000020E61
9	19	319	ΧΟΡΤΙΑΤΗΣ	12538	0103000020E61
10	14	314	ΩΡΑΙΟΚΑΣΤΡΟ	11967	0103000020E61
11	13	313	ΤΡΙΑΝΔΡΙΑ	11750	0103000020E61

Σημείωση: Τα ερωτήματα που προηγήθηκαν ζητούν την επιστροφή όλων των διαθέσιμων γνωρισμάτων για τις εγγραφές που επιλέγονται. Αυτό γίνεται με την έκφραση “SELECT *”. Αν αντί αυτού επιθυμούσαμε την επιστροφή κάποιων συγκεκριμένων πεδίων τότε την εντολή SELECT θα ακολουθούσαν τα ονόματα των πεδίων που θέλουμε να επιστραφούν.

Δεν θα επεκταθούμε περισσότερο στις εντολές SQL στον παρόντα οδηγό καθώς εδώ θα εστιάσουμε σε χωρικά ερωτήματα. Περισσότερο υλικό μπορεί κανείς να βρει [εδώ](#), [εδώ](#) και [εδώ](#) (βλ. Παράρτημα: υλικό για παραπέρα μελέτη) αλλά και σε πολλές ακόμα πηγές που είναι ελεύθερα διαθέσιμες στο διαδίκτυο.

5.2.2 Μη χωρικά ερωτήματα στο περιβάλλον του QuantumGIS

Το λογισμικό QuantumGIS μας προσφέρει την δυνατότητα να υποβάλουμε ερωτήματα SQL στα δεδομένα κατά την εισαγωγή τους. Επιλέγοντας “Layer”, “Add PostGIS Layer” ενεργοποιείται το παράθυρο εισαγωγής δεδομένων. Επιλέγουμε την βάση δεδομένων όπου εργαζόμαστε και στην συνέχεια “Connect”. Έτσι εμφανίζονται τα διαθέσιμα θεματικά επίπεδα. Κάνουμε διπλό κλικ στο θεματικό επίπεδο όπου επιθυμούμε να εφαρμοστεί το ερώτημα. Με αυτό τον τρόπο ενεργοποιείται το παράθυρο δημιουργίας ερωτημάτων SQL. Εκεί δημιουργούμε το σχετικό ερώτημα SQL και κατά την διάρκεια εισαγωγής των δεδομένων εισάγονται μόνο οι εγγραφές που το ικανοποιούν.

Έστω, για παράδειγμα, ότι θέλουμε να απεικονίσουμε τους δήμους Θεσσαλονίκης που έχουν πληθυσμό μικρότερο από 40,000 κατοίκους και μεγαλύτερο από 10,000 κατοίκους. Στο παράθυρο εισαγωγής δεδομένων επιλέγουμε την βάση δεδομένων thessaloniki και επιλέγουμε "Connect". Εμφανίζονται τα διαθέσιμα θεματικά επίπεδα. Εκεί κάνουμε διπλό κλικ στο θεματικό "municipal" και έτσι οδηγούμαστε στο παράθυρο δημιουργίας ερωτημάτων SQL. Συντάσσουμε το ακόλουθο ερώτημα: "POP" < 40000 AND "POP" > 10000

PostgreSQL Query Builder

Table "health_services"."municipal" in database thessaloniki on host localhost, user postgres

Fields

- gid
- ID
- NAME
- POP
- the_geom

Values

Sample All

Operators

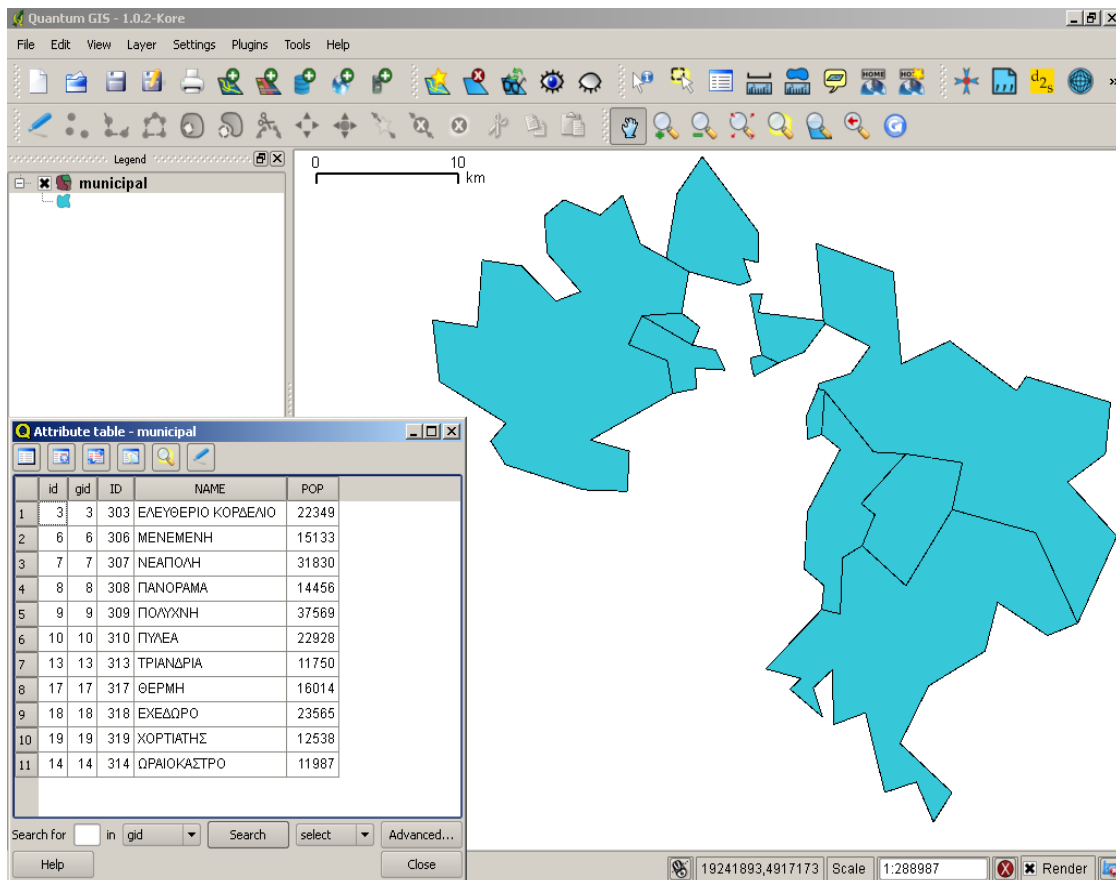
- =
- <
- >
- LIKE
- %
- IN
- NOT IN
- <=
- >=
- !=
- ILIKE
- AND
- OR
- NOT

SQL where clause

"POP" < 40000 AND "POP" > 10000

Clear Test Ok Cancel

Επιλέγουμε OK και εισάγουμε το θεματικό επίπεδο "municipal". Πράγματι έχουν εισαχθεί οι δήμοι που ο πληθυσμός τους κυμαίνεται από 10000 έως 40000 άτομα.



5.2.3 Χωρικά ερωτήματα

Τα χωρικά ερωτήματα αφορούν στα γεωμετρικά χαρακτηριστικά του κάθε στοιχείου που βρίσκεται στην βάση δεδομένων όπως αυτό περιέχεται στο πεδίο γεωμετρίας. Στην συνέχεια θα δώσουμε κάποια παραδείγματα χωρικών τελεστών και των ερωτημάτων που υλοποιούν.

5.2.3.1 Οι τελεστές WITHIN και CONTAINS

Οι τελεστές WITHIN και CONTAINS μας επιστρέφουν πληροφορίες για στοιχεία που περικλείονται σε ένα γνωστό πολύγωνο ή πολύγωνα που περικλείουν γνωστά στοιχεία.

Ο τελεστής WITHIN χρησιμοποιείται ως εξής:

```
SELECT X
FROM A,B
WHERE WITHIN (Geom.A, GeomB)
```

Ο τελεστής CONTAINS χρησιμοποιείται ως εξής:

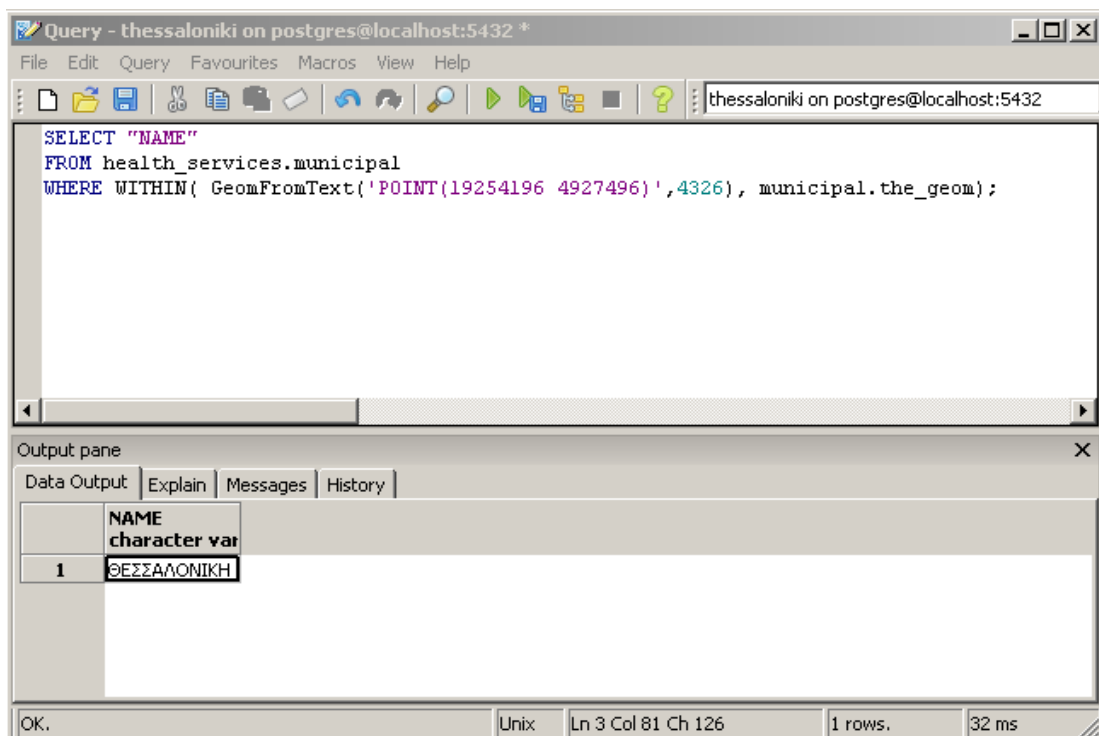
```
SELECT X
```

```
FROM A,B
WHERE CONTAINS (Geom.B, GeomA)
```

Και στις δύο περιπτώσεις A είναι το θεματικό επίπεδο με τα στοιχεία η θέση των οποίων μας ενδιαφέρει να διερευνηθεί ως προς τον πιθανό εγκλεισμό στα πολύγωνα του επιπέδου B.

Για παράδειγμα ο τελεστής WITHIN μπορεί να μας επιστρέψει πληροφορίες για το που (μέσα σε ποιο πολυγωνικό στοιχείο) βρίσκεται ένα σημείο με γνωστές συντεταγμένες: Σε ποιόν δήμο (όνομα) βρίσκεται το σημείο με συντεταγμένες XY (19254196 4927496);

```
SELECT "NAME"
FROM health_services.municipal
WHERE WITHIN( GeomFromText('POINT(19254196 4927496)',4121), municipal.the_geom);
```



Ποια νοσοκομεία βρίσκονται μέσα στο πολύγωνο του δήμου Θεσσαλονίκης;

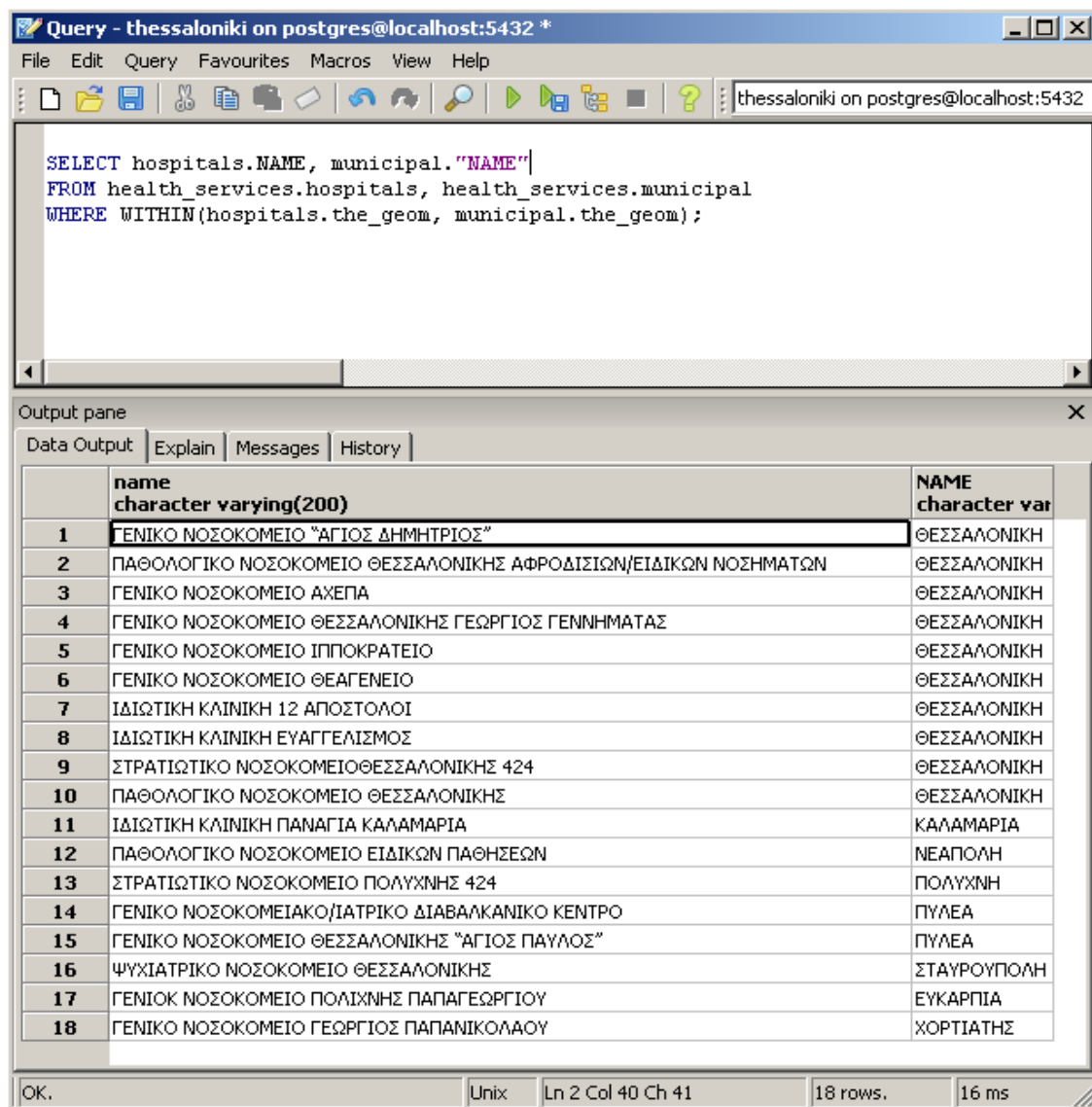
```
SELECT *
FROM health_services.hospitals
WHERE WITHIN(the_geom, (SELECT the_geom FROM health_services.municipal WHERE "NAME"
='ΘΕΣΣΑΛΟΝΙΚΗ'));
```

Ποιοι δρόμοι βρίσκονται εξολοκλήρου μέσα στον δήμο Θεσσαλονίκης;

```
SELECT *
FROM health_services.road_net
WHERE WITHIN(the_geom, (SELECT the_geom FROM health_services.municipal WHERE "NAME"
= 'ΘΕΣΣΑΛΟΝΙΚΗ'));
```

Σε ποιο δήμο βρίσκεται κάθε ένα από τα νοσοκομεία;

```
SELECT hospitals.name, municipal."NAME"  
FROM health_services.hospitals, health_services.municipal  
WHERE WITHIN (hospitals.the_geom, municipal.the_geom);
```



The screenshot shows a PostgreSQL query window titled "Query - thessaloniki on postgres@localhost:5432 *". The query is as follows:

```
SELECT hospitals.NAME, municipal."NAME"  
FROM health_services.hospitals, health_services.municipal  
WHERE WITHIN(hospitals.the_geom, municipal.the_geom);
```

The output pane displays the results in a table with two columns: "name" (character varying(200)) and "NAME" (character varying(200)). The results are as follows:

	name	NAME
1	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ "ΑΓΙΟΣ ΔΗΜΗΤΡΙΟΣ"	ΘΕΣΣΑΛΟΝΙΚΗ
2	ΠΑΘΟΛΟΓΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ ΑΦΡΟΔΙΣΙΩΝ/ΕΙΔΙΚΩΝ ΝΟΣΗΜΑΤΩΝ	ΘΕΣΣΑΛΟΝΙΚΗ
3	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΧΕΠΑ	ΘΕΣΣΑΛΟΝΙΚΗ
4	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ ΓΕΩΡΓΙΟΣ ΓΕΝΝΗΜΑΤΑΣ	ΘΕΣΣΑΛΟΝΙΚΗ
5	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΙΠΠΟΚΡΑΤΕΙΟ	ΘΕΣΣΑΛΟΝΙΚΗ
6	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΑΓΕΝΕΙΟ	ΘΕΣΣΑΛΟΝΙΚΗ
7	ΙΔΙΩΤΙΚΗ ΚΛΙΝΙΚΗ 12 ΑΠΟΣΤΟΛΟΙ	ΘΕΣΣΑΛΟΝΙΚΗ
8	ΙΔΙΩΤΙΚΗ ΚΛΙΝΙΚΗ ΕΥΑΓΓΕΛΙΣΜΟΣ	ΘΕΣΣΑΛΟΝΙΚΗ
9	ΣΤΡΑΤΙΩΤΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ 424	ΘΕΣΣΑΛΟΝΙΚΗ
10	ΠΑΘΟΛΟΓΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ	ΘΕΣΣΑΛΟΝΙΚΗ
11	ΙΔΙΩΤΙΚΗ ΚΛΙΝΙΚΗ ΠΑΝΑΓΙΑ ΚΑΛΑΜΑΡΙΑ	ΚΑΛΑΜΑΡΙΑ
12	ΠΑΘΟΛΟΓΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΕΙΔΙΚΩΝ ΠΑΘΗΣΕΩΝ	ΝΕΑΠΟΛΗ
13	ΣΤΡΑΤΙΩΤΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΠΟΛΥΧΝΗΣ 424	ΠΟΛΥΧΝΗ
14	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΙΑΤΡΙΚΟ ΔΙΑΒΑΛΚΑΝΙΚΟ ΚΕΝΤΡΟ	ΠΥΛΕΑ
15	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ "ΑΓΙΟΣ ΠΑΥΛΟΣ"	ΠΥΛΕΑ
16	ΨΥΧΙΑΤΡΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ	ΣΤΑΥΡΟΥΠΟΛΗ
17	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΠΟΛΙΧΝΗΣ ΠΑΠΑΓΕΩΡΓΙΟΥ	ΕΥΚΑΡΠΙΑ
18	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΓΕΩΡΓΙΟΣ ΠΑΠΑΝΙΚΟΛΑΟΥ	ΧΟΡΤΙΑΤΗΣ

The status bar at the bottom indicates "OK.", "Unix", "Ln 2 Col 40 Ch 41", "18 rows.", and "16 ms".

Σημείωση: Αν θελήσουμε να οπτικοποιήσουμε τα αποτελέσματα ενός χωρικού ερωτήματος, αρκεί να δημιουργήσουμε μία όψη που να περιέχει το ερώτημα και στην συνέχεια να εισάγουμε αυτή την όψη στο QuantumGIS ακριβώς σαν να επρόκειτο για κάποιο πίνακα. Έστω ότι θέλουμε να οπτικοποιήσουμε το ερώτημα που μας επιστρέφει όλα τα νοσοκομεία που βρίσκονται μέσα στον δήμο Θεσσαλονίκης. Μπορούμε να δημιουργήσουμε μία όψη που να περιέχει το ερώτημα με την ακόλουθη εντολή:

```
CREATE VIEW health_services.hospitals_thessaloniki AS  
(SELECT *  
FROM health_services.hospitals  
WHERE WITHIN(the_geom, (SELECT the_geom FROM health_services.municipal WHERE "NAME"
```

= 'ΘΕΣΣΑΛΟΝΙΚΗ')));

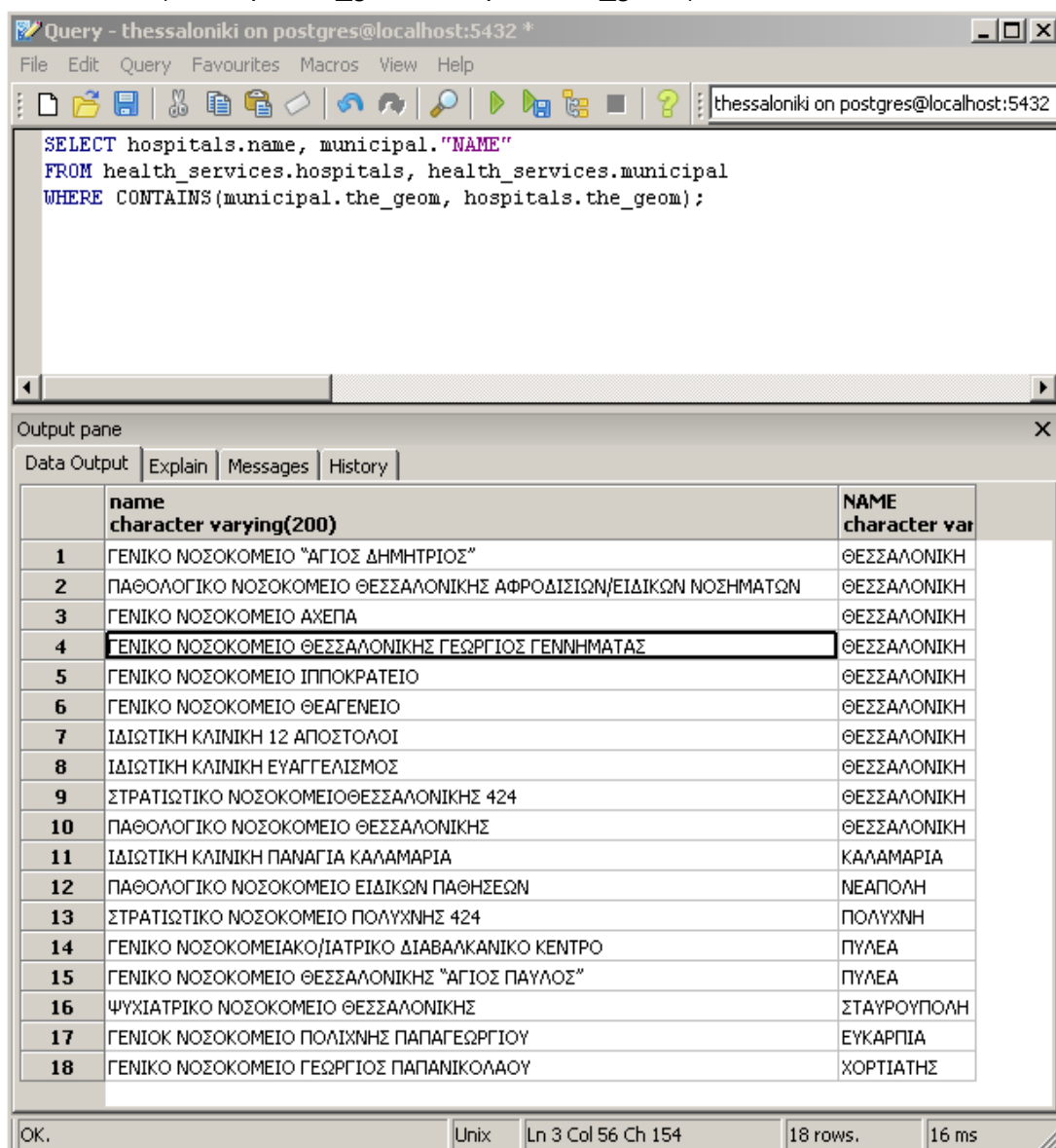
Αν στην συνέχεια συνδεθούμε με την βάση δεδομένων τότε μπορούμε να δούμε την όψη που δημιουργήσαμε παραπάνω. Αν την οπτικοποιήσουμε τότε γίνεται σαφές ότι πράγματι έχουν επιλεγεί τα νοσοκομεία που βρίσκονται στον δήμο Θεσσαλονίκης.

Παραδείγματα με τον τελεστή CONTAINS. Ποιος δήμος περιλαμβάνει το Νοσοκομείο ΑΧΕΠΑ;

```
SELECT "NAME"  
FROM health_services.municipal  
WHERE CONTAINS(the_geom, (SELECT the_geom FROM health_services.hospitals WHERE  
hospitals.name='ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΧΕΠΑ'));
```

Σε ποιο δήμο βρίσκεται κάθε νοσοκομείο ;

```
SELECT hospitals.name, municipal."NAME"  
FROM health_services.hospitals, health_services.municipal  
WHERE CONTAINS(municipal.the_geom, hospitals.the_geom);
```



The screenshot shows a PostgreSQL query window titled "Query - thessaloniki on postgres@localhost:5432 *". The query is:

```
SELECT hospitals.name, municipal."NAME"  
FROM health_services.hospitals, health_services.municipal  
WHERE CONTAINS(municipal.the_geom, hospitals.the_geom);
```

The output pane shows the results of the query, displaying 18 rows. The columns are "name" (character varying(200)) and "NAME" (character varying(200)). The results are as follows:

	name character varying(200)	NAME character varying(200)
1	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ "ΑΓΙΟΣ ΔΗΜΗΤΡΙΟΣ"	ΘΕΣΣΑΛΟΝΙΚΗ
2	ΠΑΘΟΛΟΓΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ ΑΦΡΟΔΙΣΙΩΝ/ΕΙΔΙΚΩΝ ΝΟΣΗΜΑΤΩΝ	ΘΕΣΣΑΛΟΝΙΚΗ
3	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΧΕΠΑ	ΘΕΣΣΑΛΟΝΙΚΗ
4	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ ΓΕΩΡΓΙΟΣ ΓΕΝΝΗΜΑΤΑΣ	ΘΕΣΣΑΛΟΝΙΚΗ
5	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΙΠΠΟΚΡΑΤΕΙΟ	ΘΕΣΣΑΛΟΝΙΚΗ
6	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΑΓΕΝΕΙΟ	ΘΕΣΣΑΛΟΝΙΚΗ
7	ΙΔΙΩΤΙΚΗ ΚΛΙΝΙΚΗ 12 ΑΠΟΣΤΟΛΟΙ	ΘΕΣΣΑΛΟΝΙΚΗ
8	ΙΔΙΩΤΙΚΗ ΚΛΙΝΙΚΗ ΕΥΑΓΓΕΛΙΣΜΟΣ	ΘΕΣΣΑΛΟΝΙΚΗ
9	ΣΤΡΑΤΙΩΤΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ 424	ΘΕΣΣΑΛΟΝΙΚΗ
10	ΠΑΘΟΛΟΓΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ	ΘΕΣΣΑΛΟΝΙΚΗ
11	ΙΔΙΩΤΙΚΗ ΚΛΙΝΙΚΗ ΠΑΝΑΓΙΑ ΚΑΛΑΜΑΡΙΑ	ΚΑΛΑΜΑΡΙΑ
12	ΠΑΘΟΛΟΓΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΕΙΔΙΚΩΝ ΠΑΘΗΣΕΩΝ	ΝΕΑΠΟΛΗ
13	ΣΤΡΑΤΙΩΤΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΠΟΛΥΧΝΗΣ 424	ΠΟΛΥΧΝΗ
14	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΙΑΤΡΙΚΟ ΔΙΑΒΑΛΚΑΝΙΚΟ ΚΕΝΤΡΟ	ΠΥΛΕΑ
15	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ "ΑΓΙΟΣ ΠΑΥΛΟΣ"	ΠΥΛΕΑ
16	ΨΥΧΙΑΤΡΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ	ΣΤΑΥΡΟΥΠΟΛΗ
17	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΠΟΛΙΧΝΗΣ ΠΑΠΑΓΕΩΡΓΙΟΥ	ΕΥΚΑΡΠΙΑ
18	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΓΕΩΡΓΙΟΣ ΠΑΠΑΝΙΚΟΛΑΟΥ	ΧΟΡΤΙΑΤΗΣ

The status bar at the bottom indicates "OK.", "Unix", "Ln 3 Col 56 Ch 154", "18 rows.", and "16 ms".

5.2.3.2 Ο τελεστής DISTANCE

Ο τελεστής DISTANCE προσδιορίζει την απόσταση (σε κάποιες περιπτώσεις την ελάχιστη απόσταση) ανάμεσα στην γεωμετρία στοιχείων:

Ποια νοσοκομεία απέχουν λιγότερο από 2000 μέτρα από το σημείο με συντεταγμένες XY (19254196 4927496) και ποια είναι η απόσταση αυτών των νοσοκομείων από το σημείο;

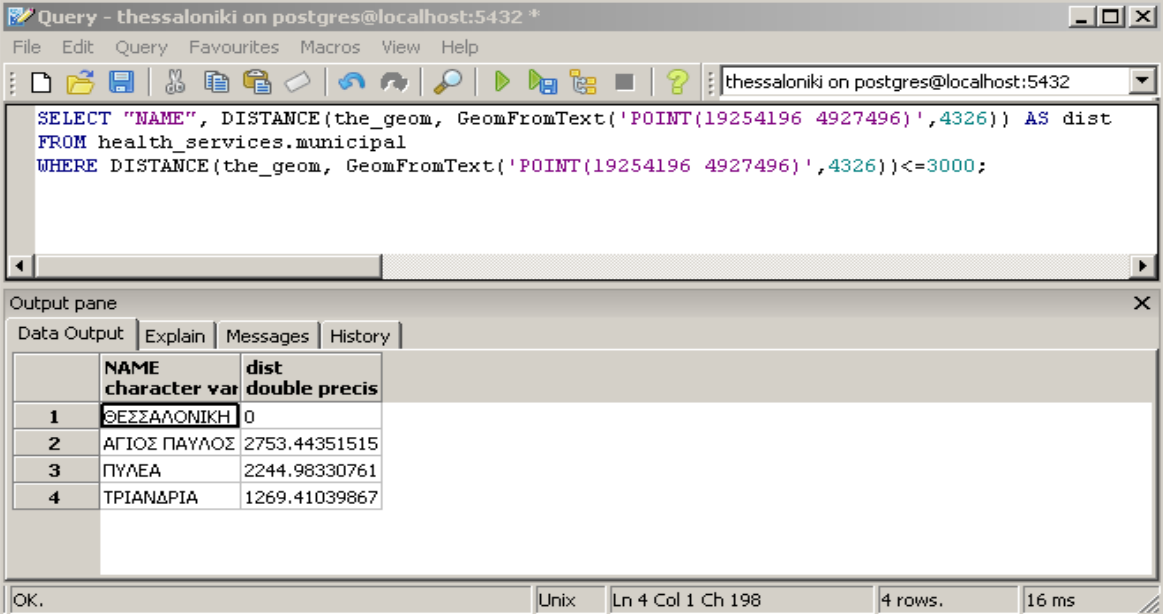
```
SELECT name, DISTANCE(the_geom, GeomFromText('POINT(19254196 4927496)',4121)) AS dist
FROM health_services.hospitals
WHERE DISTANCE(the_geom, GeomFromText('POINT(19254196 4927496)',4121))<=2000;
```

Ποιος δρόμος απέχει λιγότερο από 50 μέτρα από το σημείο με συντεταγμένες XY (19254196 4927496) και ποια είναι η απόσταση του από το σημείο;

```
SELECT name, DISTANCE(the_geom, GeomFromText('POINT(19254196 4927496)',4121)) AS dist
FROM health_services.road_net
WHERE DISTANCE(the_geom, GeomFromText('POINT(19254196 4927496)',4121))<=50;
```

Ποιοι δήμοι βρίσκονται σε απόσταση 3000 από το σημείο με συντεταγμένες XY (19254196 4927496) και ποια είναι η ελάχιστη απόστασή τους από το σημείο;

```
SELECT "NAME", DISTANCE(the_geom, GeomFromText('POINT(19254196 4927496)',4121)) AS dist
FROM health_services.municipal
WHERE DISTANCE(the_geom, GeomFromText('POINT(19254196 4927496)',4121))<=3000;
```



The screenshot shows a PostgreSQL query window titled "Query - thessaloniki on postgres@localhost:5432 *". The query is as follows:

```
SELECT "NAME", DISTANCE(the_geom, GeomFromText('POINT(19254196 4927496)',4326)) AS dist
FROM health_services.municipal
WHERE DISTANCE(the_geom, GeomFromText('POINT(19254196 4927496)',4326))<=3000;
```

The output pane shows the results of the query in a table with two columns: NAME (character var) and dist (double precis). The results are as follows:

	NAME character var	dist double precis
1	ΘΕΣΣΑΛΟΝΙΚΗ	0
2	ΑΓΙΟΣ ΠΑΥΛΟΣ	2753.44351515
3	ΠΥΛΕΑ	2244.98330761
4	ΤΡΙΑΝΔΡΙΑ	1269.41039867

The bottom of the window shows "OK.", "Unix", "Ln 4 Col 1 Ch 198", "4 rows.", and "16 ms".

Υπάρχουν νοσοκομεία που να απέχουν μεταξύ τους λιγότερο από ένα χιλιόμετρο; Αν ναι ποια είναι αυτά και ποια η απόσταση μεταξύ τους;

```

SELECT H1.name AS Hospital1, H2.name AS Hospital2, DISTANCE(H1.the_geom, H2.the_geom) AS
distance
FROM health_services.hospitals H1, health_services.hospitals H2
WHERE DISTANCE(H1.the_geom, H2.the_geom)<=1000
AND H1.id < H2.id;

```

Query - thessaloniki on postgres@localhost:5432 *

File Edit Query Favourites Macros View Help

thessaloniki on postgres@localhost:5432

```

SELECT H1.name AS Hospital1, H2.name AS Hospital2, DISTANCE(H1.the_geom, H2.the_geom) AS distance
FROM health_services.hospitals H1, health_services.hospitals H2
WHERE DISTANCE(H1.the_geom, H2.the_geom)<=1000
AND H1.id < H2.id;

```

Output pane

Data Output Explain Messages History

	hospital1 character varying(200)	hospital2 character varying(200)	distance double precis
1	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ "ΑΓΙΟΣ ΔΗΜΗΤΡΙΟΣ"	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ ΓΕΩΡΓΙΟΣ ΓΕΝΝΗΜΑΤΑΣ	429.328645381
2	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΧΕΠΑ	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ ΓΕΩΡΓΙΟΣ ΓΕΝΝΗΜΑΤΑΣ	951.389642352
3	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΑΧΕΠΑ	ΣΤΡΑΤΙΩΤΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ 424	741.638468837
4	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΠΟΛΙΧΝΗΣ ΠΑΠΑΓΕΩΡΓΙΟΥ	ΣΤΡΑΤΙΩΤΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΠΟΛΥΧΝΗΣ 424	383.819319386
5	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΙΠΠΟΚΡΑΤΕΙΟ	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΑΓΕΝΕΙΟ	758.579842527
6	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΙΠΠΟΚΡΑΤΕΙΟ	ΙΔΙΩΤΙΚΗ ΚΛΙΝΙΚΗ ΕΥΑΓΓΕΛΙΣΜΟΣ	546.098287103
7	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΑΓΕΝΕΙΟ	ΙΔΙΩΤΙΚΗ ΚΛΙΝΙΚΗ ΕΥΑΓΓΕΛΙΣΜΟΣ	263.928032296
8	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΑΓΕΝΕΙΟ	ΣΤΡΑΤΙΩΤΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ 424	939.472883347
9	ΓΕΝΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΑΓΕΝΕΙΟ	ΠΑΘΟΛΟΓΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ	728.674000623
10	ΙΔΙΩΤΙΚΗ ΚΛΙΝΙΚΗ ΕΥΑΓΓΕΛΙΣΜΟΣ	ΠΑΘΟΛΟΓΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ	769.208179581
11	ΣΤΡΑΤΙΩΤΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ 424	ΠΑΘΟΛΟΓΙΚΟ ΝΟΣΟΚΟΜΕΙΟ ΘΕΣΣΑΛΟΝΙΚΗΣ	393.581882178

OK. Unix Ln 7 Col 1 Ch 231 11 rows. 15 ms

5.2.3.3 Ο τελεστής INTERSECTS

Ο τελεστής INTERSECTS μας επιστρέφει πληροφορία σχετικά με το πότε δύο γεωμετρίες πληρούν τα κριτήρια τομής σύμφωνα με την σχετική προδιαγραφή του Open Geospatial Consortium.

Για παράδειγμα:

Ποιοι δρόμοι τέμνονται με το πολύγωνο με συντεταγμένες (19260523.58872549 4931624.24705882, 19252724.53872549 4925125.03872549, 19253794.99656863 4913350.00245098, 19268934.32892157 4931471.32450980, 19260523.58872549 4931624.24705882);

```

SELECT road_net.name
FROM health_services.road_net
WHERE INTERSECTS (the_geom, GeomFromText('POLYGON
((19260523.58872549 4931624.24705882, 19252724.53872549 4925125.03872549,
19253794.99656863 4913350.00245098, 19268934.32892157

```



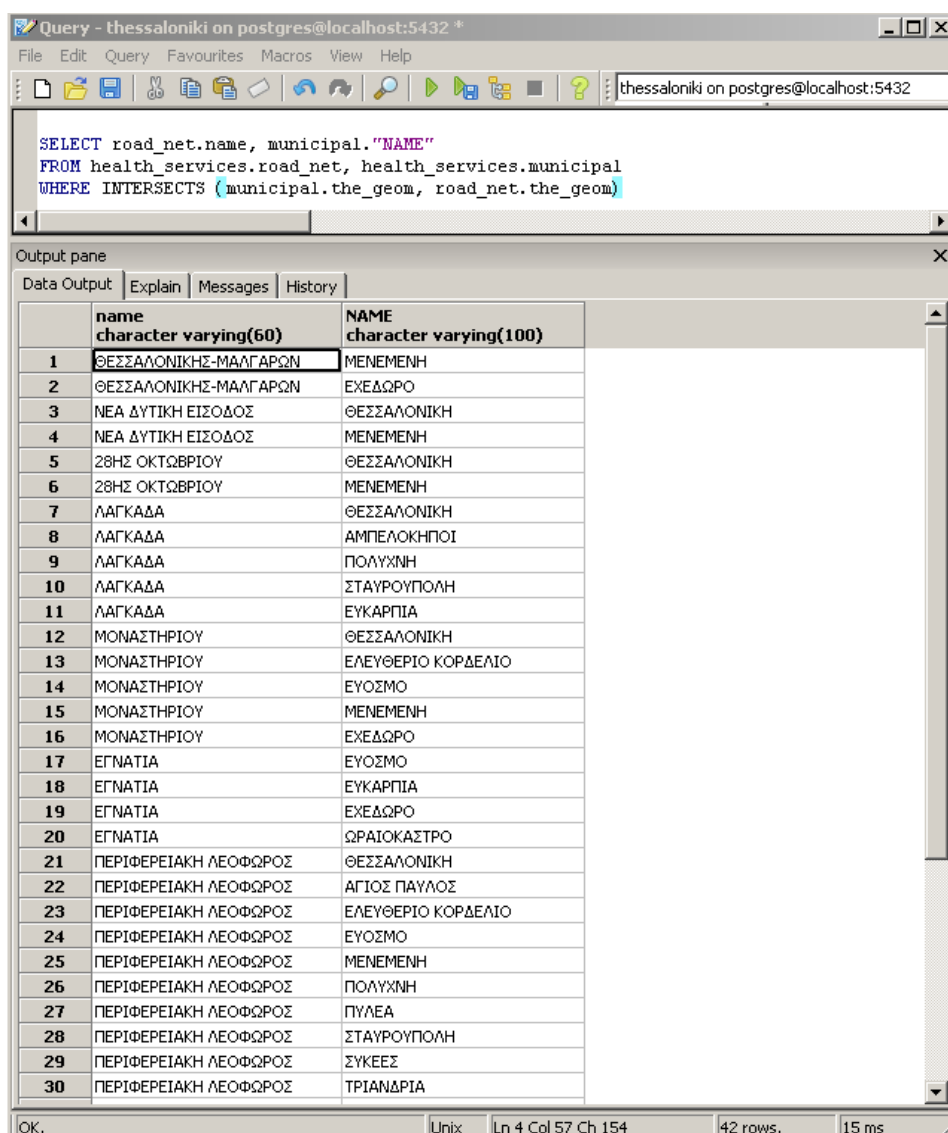
```
4931471.32450980,19260523.58872549 4931624.24705882))
',4121) );
```

Ποιοι δήμοι τέμνονται με το πολύγωνο με συντεταγμένες (19260523.58872549 4931624.24705882, 19252724.53872549 4925125.03872549, 19253794.99656863 4913350.00245098, 19268934.32892157 4931471.32450980, 19260523.58872549 4931624.24705882);

```
SELECT municipal."NAME"
FROM health_services.municipal
WHERE INTERSECTS (the_geom, GeomFromText('POLYGON
((19260523.58872549 4931624.24705882, 19252724.53872549 4925125.03872549,
19253794.99656863 4913350.00245098, 19268934.32892157
4931471.32450980,19260523.58872549 4931624.24705882))
',4121) );
```

Ποιους δήμους τέμνει κάθε τμήμα οδικού δικτύου;

```
SELECT road_net.name, municipal."NAME"
FROM health_services.road_net, health_services.municipal
WHERE INTERSECTS (municipal.the_geom, road_net.the_geom)
```



Query - thessaloniki on postgres@localhost:5432 *

```
SELECT road_net.name, municipal."NAME"
FROM health_services.road_net, health_services.municipal
WHERE INTERSECTS (municipal.the_geom, road_net.the_geom)
```

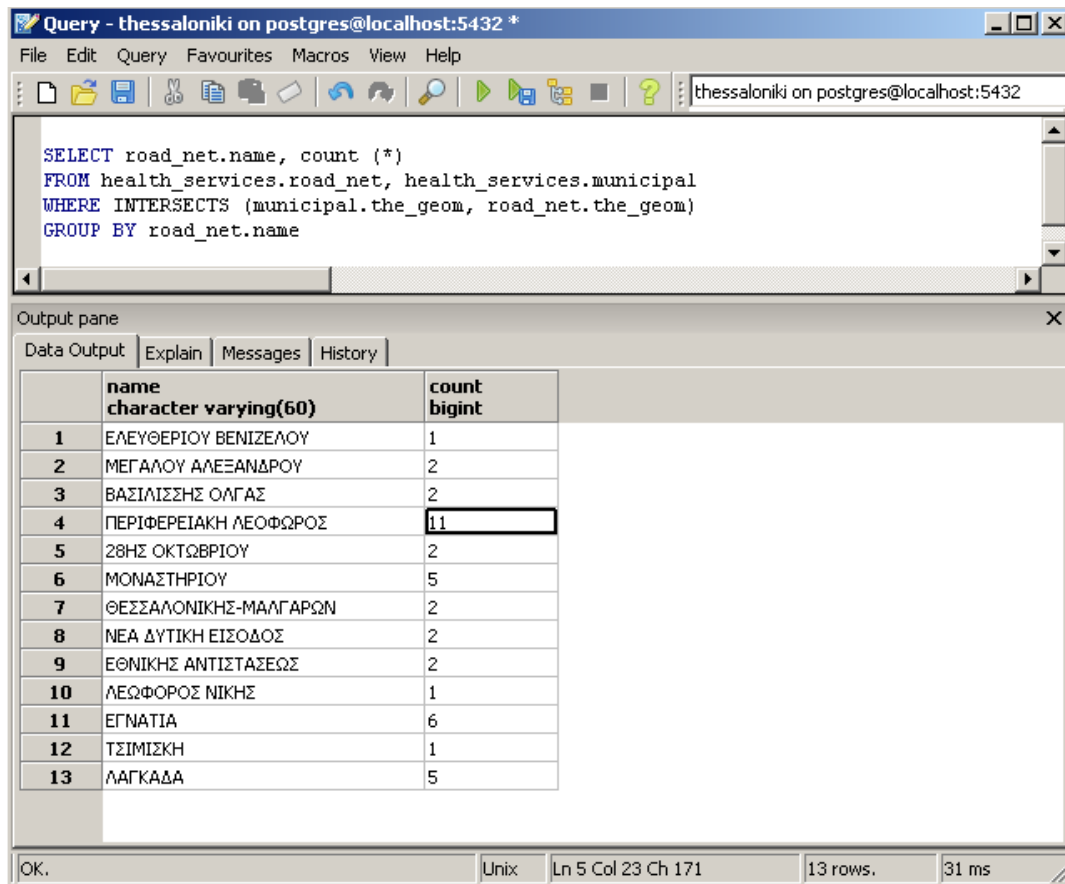
Output pane

	name character varying(60)	NAME character varying(100)
1	ΘΕΣΣΑΛΟΝΙΚΗΣ-ΜΑΛΓΑΡΩΝ	MENEMENH
2	ΘΕΣΣΑΛΟΝΙΚΗΣ-ΜΑΛΓΑΡΩΝ	ΕΧΕΔΩΡΟ
3	ΝΕΑ ΔΥΤΙΚΗ ΕΙΣΟΔΟΣ	ΘΕΣΣΑΛΟΝΙΚΗ
4	ΝΕΑ ΔΥΤΙΚΗ ΕΙΣΟΔΟΣ	MENEMENH
5	28ΗΣ ΟΚΤΩΒΡΙΟΥ	ΘΕΣΣΑΛΟΝΙΚΗ
6	28ΗΣ ΟΚΤΩΒΡΙΟΥ	MENEMENH
7	ΛΑΓΚΑΔΑ	ΘΕΣΣΑΛΟΝΙΚΗ
8	ΛΑΓΚΑΔΑ	ΑΜΠΕΛΟΚΗΠΟΙ
9	ΛΑΓΚΑΔΑ	ΠΟΛΥΧΝΗ
10	ΛΑΓΚΑΔΑ	ΣΤΑΥΡΟΥΠΟΛΗ
11	ΛΑΓΚΑΔΑ	ΕΥΚΑΡΠΙΑ
12	ΜΟΝΑΣΤΗΡΙΟΥ	ΘΕΣΣΑΛΟΝΙΚΗ
13	ΜΟΝΑΣΤΗΡΙΟΥ	ΕΛΕΥΘΕΡΙΟ ΚΟΡΔΕΛΙΟ
14	ΜΟΝΑΣΤΗΡΙΟΥ	ΕΥΟΣΜΟ
15	ΜΟΝΑΣΤΗΡΙΟΥ	MENEMENH
16	ΜΟΝΑΣΤΗΡΙΟΥ	ΕΧΕΔΩΡΟ
17	ΕΓΝΑΤΙΑ	ΕΥΟΣΜΟ
18	ΕΓΝΑΤΙΑ	ΕΥΚΑΡΠΙΑ
19	ΕΓΝΑΤΙΑ	ΕΧΕΔΩΡΟ
20	ΕΓΝΑΤΙΑ	ΩΡΑΙΟΚΑΣΤΡΟ
21	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ	ΘΕΣΣΑΛΟΝΙΚΗ
22	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ	ΑΓΙΟΣ ΠΑΥΛΟΣ
23	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ	ΕΛΕΥΘΕΡΙΟ ΚΟΡΔΕΛΙΟ
24	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ	ΕΥΟΣΜΟ
25	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ	MENEMENH
26	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ	ΠΟΛΥΧΝΗ
27	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ	ΠΥΛΕΑ
28	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ	ΣΤΑΥΡΟΥΠΟΛΗ
29	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ	ΣΥΚΕΕΣ
30	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ	ΤΡΙΑΝΔΡΙΑ

OK. Unix Ln 4 Col 57 Ch 154 42 rows. 15 ms

Πόσους δήμους τέμνει κάθε τμήμα του οδικού δικτύου;

```
SELECT road_net.name, count (*)
FROM health_services.road_net, health_services.municipal
WHERE INTERSECTS (municipal.the_geom, road_net.the_geom)
GROUP BY road_net.name
```



Query - thessaloniki on postgres@localhost:5432 *

```
SELECT road_net.name, count (*)
FROM health_services.road_net, health_services.municipal
WHERE INTERSECTS (municipal.the_geom, road_net.the_geom)
GROUP BY road_net.name
```

Output pane

	name character varying(60)	count bigint
1	ΕΛΕΥΘΕΡΙΟΥ ΒΕΝΙΖΕΛΟΥ	1
2	ΜΕΓΑΛΟΥ ΑΛΕΞΑΝΔΡΟΥ	2
3	ΒΑΣΙΛΙΣΣΗΣ ΟΛΓΑΣ	2
4	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΩΦΩΡΟΣ	11
5	28ΗΣ ΟΚΤΩΒΡΙΟΥ	2
6	ΜΟΝΑΣΤΗΡΙΟΥ	5
7	ΘΕΣΣΑΛΟΝΙΚΗΣ-ΜΑΛΓΑΡΩΝ	2
8	ΝΕΑ ΔΥΤΙΚΗ ΕΙΣΟΔΟΣ	2
9	ΕΘΝΙΚΗΣ ΑΝΤΙΣΤΑΣΕΩΣ	2
10	ΛΕΩΦΟΡΟΣ ΝΙΚΗΣ	1
11	ΕΓΝΑΤΙΑ	6
12	ΤΣΙΜΙΣΚΗ	1
13	ΛΑΓΚΑΔΑ	5

OK. Unix Ln 5 Col 23 Ch 171 13 rows. 31 ms

5.2.3.4 Ο τελεστής TOUCHES

Ο τελεστής TOUCHES ανιχνεύει αν εφάπτονται δύο γεωγραφικά στοιχεία. Παράδειγμα:

Ποια είναι τα ζεύγη γειτονικών δήμων στην Θεσσαλονίκη;

```
SELECT M1."NAME", M2."NAME"
FROM health_services.municipal M1, health_services.municipal M2
WHERE TOUCHES(M1.the_geom, M2.the_geom)
AND M1."ID" < M2."ID";
```

Εξειδικεύοντας, με ποιους δήμους γειτονεύει ο δήμος Θεσσαλονίκης;

```
SELECT M1."NAME", M2."NAME"
FROM health_services.municipal M1, health_services.municipal M2
```

```
WHERE TOUCHES(M1.the_geom, M2.the_geom)
AND M1."ID" < M2."ID"
AND M1."NAME" = 'ΘΕΣΣΑΛΟΝΙΚΗ';
```

Ποιοι δρόμοι τέμνονται με άλλα τμήματα του οδικού δικτύου και με ποια;

```
SELECT R1.name, R2.name
FROM health_services.road_net R1, health_services.road_net R2
WHERE TOUCHES(R1.the_geom, R2.the_geom)
AND R1.id < R2.id;
```

Query - thessaloniki on postgres@localhost:5432 *

File Edit Query Favourites Macros View Help

thessaloniki on postgres@localhost:5432

```
SELECT R1.name, R2.name
FROM health_services.road_net R1, health_services.road_net R2
WHERE TOUCHES(R1.the_geom, R2.the_geom)
AND R1.id < R2.id;
```

Output pane

Data Output Explain Messages History

	name character varying(60)	name character varying(60)
1	ΘΕΣΣΑΛΟΝΙΚΗΣ-ΜΑΛΓΑΡΩΝ	ΝΕΑ ΔΥΤΙΚΗ ΕΙΣΟΔΟΣ
2	ΘΕΣΣΑΛΟΝΙΚΗΣ-ΜΑΛΓΑΡΩΝ	ΕΓΝΑΤΙΑ
3	ΘΕΣΣΑΛΟΝΙΚΗΣ-ΜΑΛΓΑΡΩΝ	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ
4	ΝΕΑ ΔΥΤΙΚΗ ΕΙΣΟΔΟΣ	28ΗΣ ΟΚΤΩΒΡΙΟΥ
5	ΝΕΑ ΔΥΤΙΚΗ ΕΙΣΟΔΟΣ	ΛΑΓΚΑΔΑ
6	ΝΕΑ ΔΥΤΙΚΗ ΕΙΣΟΔΟΣ	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ
7	ΝΕΑ ΔΥΤΙΚΗ ΕΙΣΟΔΟΣ	ΤΣΙΜΙΣΚΗ
8	ΝΕΑ ΔΥΤΙΚΗ ΕΙΣΟΔΟΣ	ΛΕΩΦΟΡΟΣ ΝΙΚΗΣ
9	28ΗΣ ΟΚΤΩΒΡΙΟΥ	ΛΑΓΚΑΔΑ
10	28ΗΣ ΟΚΤΩΒΡΙΟΥ	ΤΣΙΜΙΣΚΗ
11	28ΗΣ ΟΚΤΩΒΡΙΟΥ	ΛΕΩΦΟΡΟΣ ΝΙΚΗΣ
12	ΛΑΓΚΑΔΑ	ΜΟΝΑΣΤΗΡΙΟΥ
13	ΛΑΓΚΑΔΑ	ΤΣΙΜΙΣΚΗ
14	ΛΑΓΚΑΔΑ	ΕΓΝΑΤΙΑ
15	ΛΑΓΚΑΔΑ	ΛΕΩΦΟΡΟΣ ΝΙΚΗΣ
16	ΜΟΝΑΣΤΗΡΙΟΥ	ΕΓΝΑΤΙΑ
17	ΕΓΝΑΤΙΑ	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ
18	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ	ΕΘΝΙΚΗΣ ΑΝΤΙΣΤΑΣΕΩΣ
19	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΟΦΩΡΟΣ	ΕΓΝΑΤΙΑ
20	ΕΘΝΙΚΗΣ ΑΝΤΙΣΤΑΣΕΩΣ	ΒΑΣΙΛΙΣΣΗΣ ΟΛΓΑΣ
21	ΕΘΝΙΚΗΣ ΑΝΤΙΣΤΑΣΕΩΣ	ΕΛΕΥΘΕΡΙΟΥ ΒΕΝΙΖΕΛΟΥ
22	ΒΑΣΙΛΙΣΣΗΣ ΟΛΓΑΣ	ΤΣΙΜΙΣΚΗ
23	ΤΣΙΜΙΣΚΗ	ΛΕΩΦΟΡΟΣ ΝΙΚΗΣ
24	ΕΛΕΥΘΕΡΙΟΥ ΒΕΝΙΖΕΛΟΥ	ΜΕΓΑΛΟΥ ΑΛΕΞΑΝΔΡΟΥ
25	ΜΕΓΑΛΟΥ ΑΛΕΞΑΝΔΡΟΥ	ΛΕΩΦΟΡΟΣ ΝΙΚΗΣ

OK. Unix Ln 3 Col 1 Ch 70 25 rows. 15 ms

5.2.3.5 Ο τελεστής AREA

Αυτός ο τελεστής προσδιορίζει το εμβαδό πολυγωνικών στοιχείων.

λ.χ. Ποια είναι η έκταση των δήμων του πολεοδομικού συγκροτήματος Θεσσαλονίκης;

```
SELECT "NAME", area(the_geom) AS municipal_area  
FROM health_services.municipal  
ORDER BY municipal_area DESC;
```

Ποιος είναι ο μεγαλύτερος από τους δήμους του πολεοδομικού συγκροτήματος Θεσσαλονίκης;

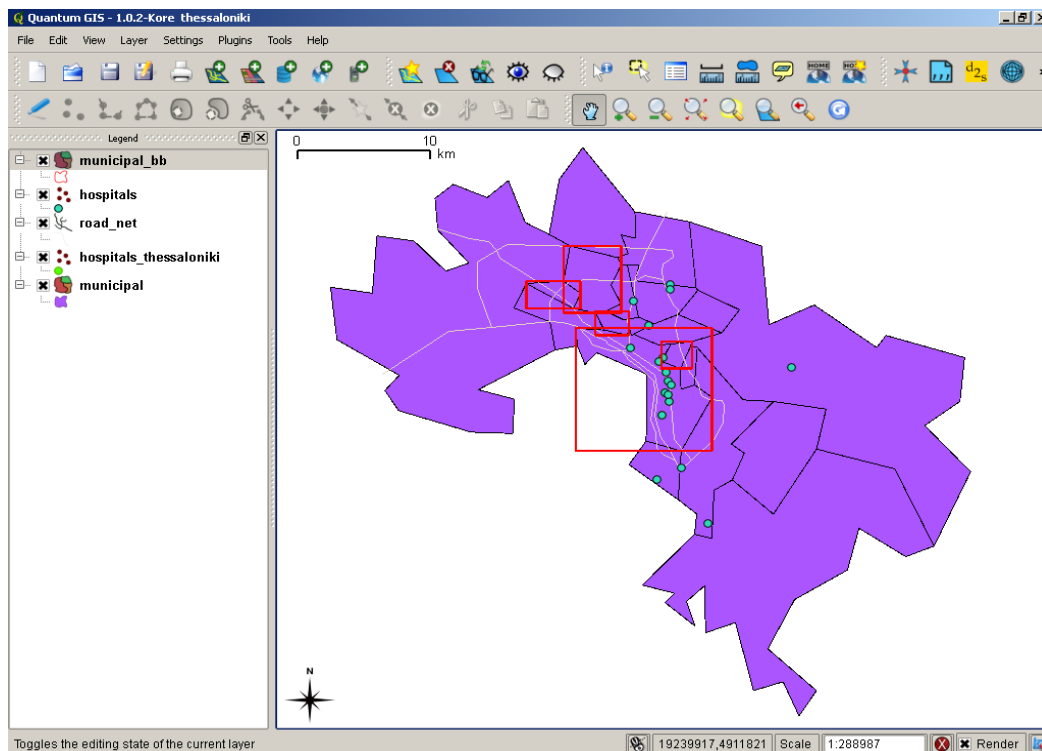
```
SELECT * FROM  
(SELECT "NAME", area(the_geom) AS municipal_area  
FROM health_services.municipal  
ORDER BY municipal_area desc)AS FOO  
LIMIT 1;
```

5.2.3.6 Ο τελεστής ENVELOPE

Αυτός ο τελεστής επιστρέφει το ελάχιστο περικλείον πολύγωνο για τα σχετικά γεωγραφικά στοιχεία.

```
SELECT "NAME", AsText(envelope(the_geom)) AS bounding_box  
FROM health_services.municipal;
```

Σημείωση: Στην παρακάτω εικόνα έχουν οπτικοποιηθεί με κόκκινο χρώμα τα ελάχιστα περικλείοντα πολύγωνα για τους δήμους Θεσσαλονίκης, Αγίου Παύλου, Αμπελοκήπων, Ελευθερίου Κορδελιού και Εύοσμου.



5.2.3.7 Ο τελεστής LENGTH

Ο τελεστής LENGTH επιστρέφει το μήκος των γραμμικών τμημάτων. Για παράδειγμα, ποιο είναι το μήκος των οδικών αξόνων που περιλαμβάνονται στον πίνακα road_net?

```
SELECT name, length(the_geom) AS segment_length
FROM health_services.road_net
ORDER BY segment_length DESC;
```

Στην περίπτωση που κάθε οδός αποτελείται από πολλαπλά τμήματα και θέλουμε να υπολογίσουμε το συνολικό μήκος της οδού μπορούμε να χρησιμοποιήσουμε κάτι σαν το παρακάτω:

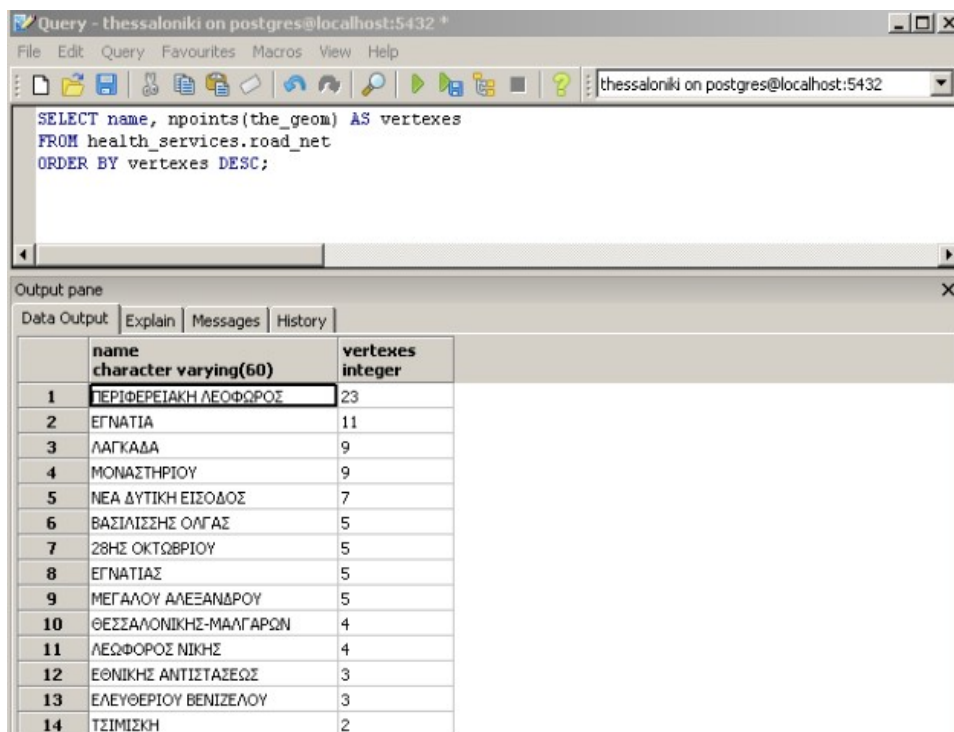
```
SELECT name, sum(length(the_geom)) AS segment_length
FROM health_services.road_net
GROUP by name
ORDER BY segment_length DESC;
```

5.2.3.8 Ο τελεστής NPOINTS

Με την βοήθεια αυτού του τελεστή υπολογίζουμε τον αριθμό κορυφών που περιέχονται σε ένα στοιχείο.

Για παράδειγμα, πόσοι κόμβοι αποτελούν την γραμμή κάθε οδικού τμήματος από τον πίνακα road_net;

```
SELECT name, npoints(the_geom) AS vertexes
FROM health_services.road_net
ORDER BY vertexes DESC;
```



The screenshot shows a PostgreSQL query editor window titled "Query - thessaloniki on postgres@localhost:5432". The query entered is:

```
SELECT name, npoints(the_geom) AS vertexes
FROM health_services.road_net
ORDER BY vertexes DESC;
```

The output pane displays the results in a table with two columns: "name" (character varying(60)) and "vertexes" (integer). The results are ordered by the number of vertices in descending order.

	name	vertexes
1	ΠΕΡΙΦΕΡΕΙΑΚΗ ΛΕΩΦΟΡΟΣ	23
2	ΕΓΝΑΤΙΑ	11
3	ΛΑΓΚΑΔΑ	9
4	ΜΟΝΑΣΤΗΡΙΟΥ	9
5	ΝΕΑ ΔΥΤΙΚΗ ΕΙΣΟΔΟΣ	7
6	ΒΑΣΙΛΙΣΣΗΣ ΟΛΓΑΣ	5
7	28ΗΣ ΟΚΤΩΒΡΙΟΥ	5
8	ΕΓΝΑΤΙΑΣ	5
9	ΜΕΓΑΛΟΥ ΑΛΕΞΑΝΔΡΟΥ	5
10	ΘΕΣΣΑΛΟΝΙΚΗΣ-ΜΑΛΓΑΡΩΝ	4
11	ΛΕΩΦΟΡΟΣ ΝΙΚΗΣ	4
12	ΕΘΝΙΚΗΣ ΑΝΤΙΣΤΑΣΕΩΣ	3
13	ΕΛΕΥΘΕΡΙΟΥ ΒΕΝΙΖΕΛΟΥ	3
14	ΤΣΙΜΙΣΚΗ	2

5.3 Λειτουργίες επέμβασης στα δεδομένα σε γραφικό περιβάλλον

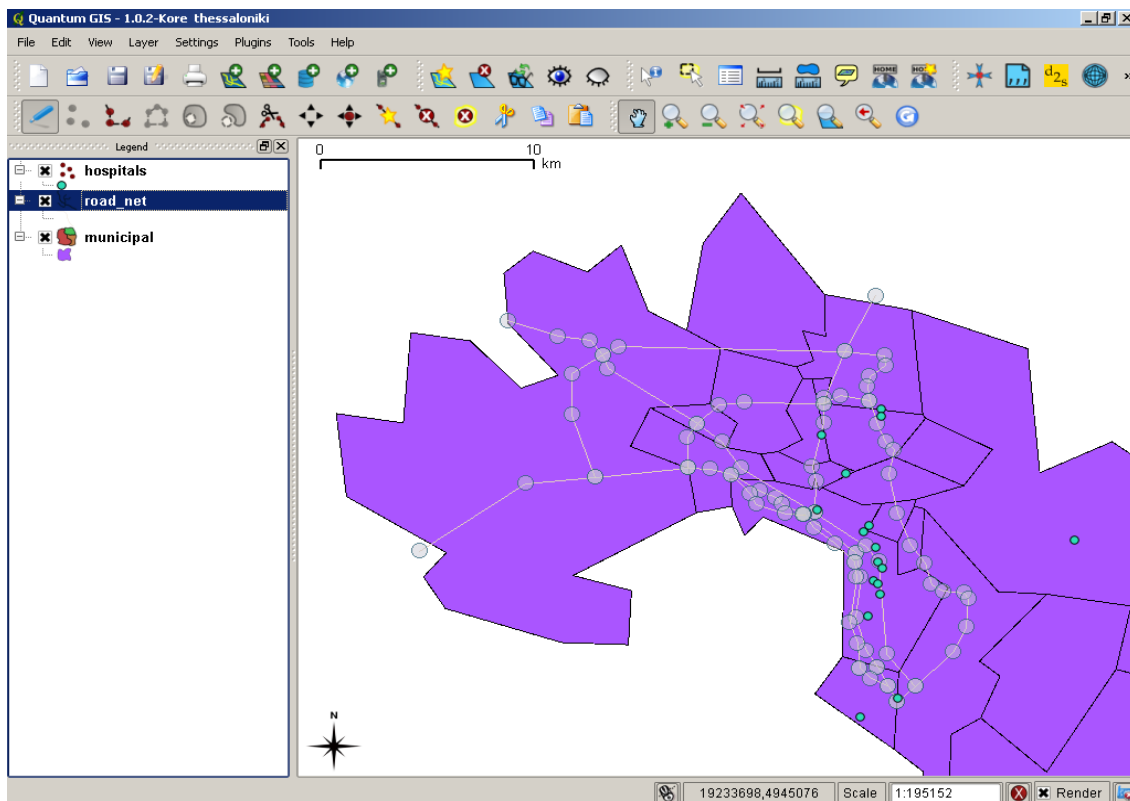
5.3.1 Περιβάλλον QuantumGIS

Προκειμένου να αρχίσουμε να επεμβαίνουμε σε ένα θεματικό επίπεδο πρέπει να κάνουμε δύο πράγματα:

- Να επιλέξουμε το θεματικό επίπεδο στο πάνελ των θεματικών επιπέδων/υπόμνημα (legend) στα αριστερά της οθόνης.
- Να επιλέξουμε από το μενού του προγράμματος “Layer”, “Toggle Editing”.

Αν πρόκειται για γραμμικό ή πολυγωνικό επίπεδο, τότε οι κόμβοι του θεματικού επιπέδου θα είναι τώρα τονισμένοι υποδεικνύοντας ότι έχει ξεκινήσει η διαδικασία επεμβάσεων.

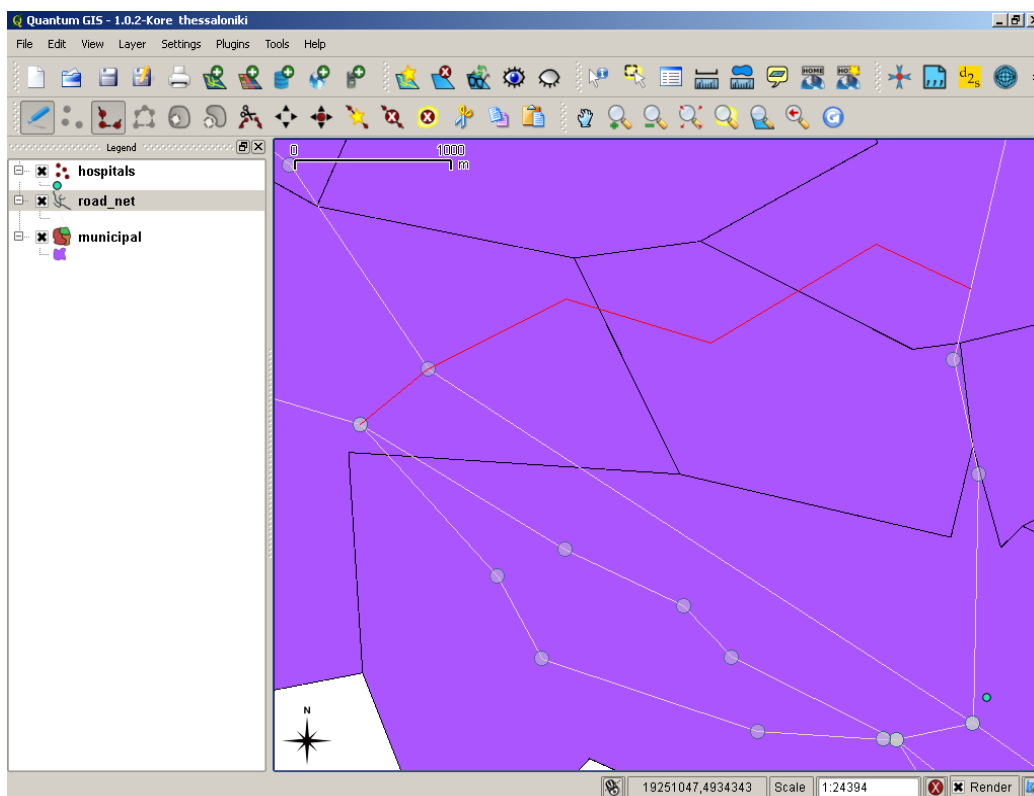
Έστω ότι θέλουμε να τροποποιήσουμε το θεματικό επίπεδο του οδικού δικτύου.



5.3.1.1 Εισαγωγή στοιχείων

Προκειμένου να εισάγουμε κάποιο στοιχείο επιλέγουμε από το μενού του προγράμματος “Edit”, “Capture Line” (Point/Polygon αν πρόκειται για σημειακό ή πολυγωνικό επίπεδο).

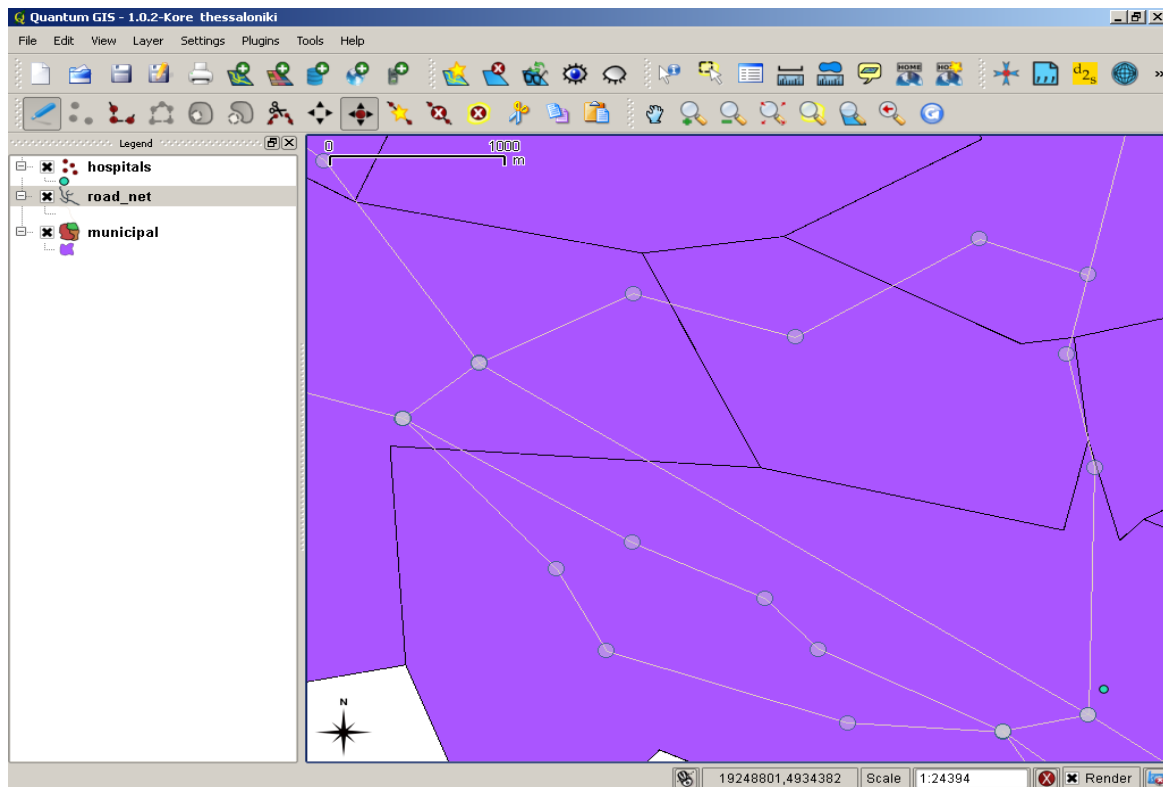
Πιέζουμε αριστερό κλικ προκειμένου να ξεκινήσουμε να ψηφιοποιούμε και δεξί κλικ όταν θέλουμε να προστεθεί ο τελευταίος κόμβος.



Στην συνέχεια καλείται η φόρμα εισαγωγής δεδομένων στον πίνακα

The image shows a dialog box titled "Enter Attribute Values". It contains four input fields with labels and values: "id (int)" with value "216", "name (txt)" with value "ANTYPΑ", "name_eng (txt)" with value "ANTYPΑ", and "speed_profile (int)" with value "5". At the bottom right, there are two buttons: "OK" and "Cancel".

Συμπληρώνουμε ότι χρειάζεται και επιλέγουμε "OK". Η εισαγωγή του νέου στοιχείου έχει ολοκληρωθεί.



5.3.1.2 Διαγραφή στοιχείων

Προκειμένου να διαγράψουμε ένα στοιχείο το επιλέγουμε με το εργαλείο επιλογής



και στην συνέχεια επιλέγουμε από το μενού του προγράμματος “Edit” και “Delete Selected”.

5.3.1.3 Μετακίνηση στοιχείου

Επιλέγουμε το αντικείμενο που θέλουμε να μετακινήσουμε με το εργαλείο επιλογής. Από το μενού του προγράμματος επιλέγουμε “Edit” και “Move Feature”. “Πιάνουμε” το στοιχείο με τον κέρσορα και το “σέρνουμε” στην επιθυμητή θέση.

5.3.1.4 Διαίρεση στοιχείου

Επιλέγουμε το αντικείμενο που θέλουμε να διαιρέσουμε με το εργαλείο επιλογής. Από το μενού του προγράμματος επιλέγουμε “Edit” και “Split Features”. Με τον κέρσορα ξεκινάμε μια γραμμή με αριστερό κλικ και την τελειώνουμε με δεξί κλικ Εκεί που η γραμμή θα τέμνει το στοιχείο θα είναι και το σημείο διαίρεσης.

5.3.1.5 Προσθήκη κόμβου

Επιλέγουμε το αντικείμενο που θέλουμε να προσθέσουμε τον κόμβο με το εργαλείο επιλογής. Από το

μενού του προγράμματος επιλέγουμε “Edit” και “Add Vertex”. Με τον κέρσορα κάνουμε κλικ στο σημείο που θέλουμε να προστεθεί ο κόμβος.

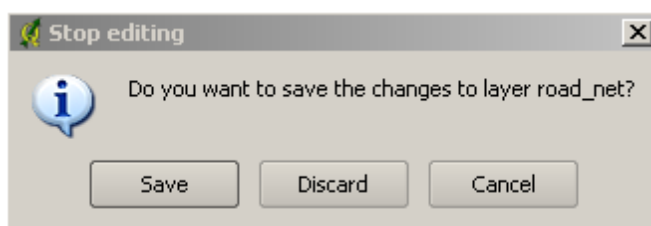
5.3.1.6 Διαγραφή κόμβου

Επιλέγουμε το αντικείμενο στο οποίο ανήκει ο κόμβος που επιθυμούμε να διαγραφεί. Από το μενού του προγράμματος επιλέγουμε “Edit” και “Delete Vertex”. Με τον κέρσορα επιλέγουμε τον κόμβο που επιθυμούμε να διαγραφεί.

5.3.1.7 Επέμβαση στον πίνακα δεδομένων.

Κάνουμε δεξί κλικ στο θεματικό πεδίο, στο πάνελ/υπόμνημα. Επιλέγουμε “Open attribute table”. Στον πίνακα δεδομένων πατάμε πάνω στην εγγραφή που θέλουμε να τροποποιήσουμε και στην συνέχεια κάνουμε διπλό κλικ στο κελί που θέλουμε να επεμβούμε. Τροποποιούμε το περιεχόμενο του κελιού και πατάμε enter.

Όταν ολοκληρώσουμε τις επεμβάσεις στο θεματικό πεδίο επιλέγουμε ξανά από το μενού του προγράμματος “Layer” και “Toggle Editing”. Έτσι ολοκληρώνουμε την διαδικασία επεμβάσεων στο θεματικό πεδίο. Το πρόγραμμα μας ρωτάει αν επιθυμούμε να σώσουμε τις αλλαγές που έγιναν.



Αν επιλέξουμε “Save” τότε οι αλλαγές αποθηκεύονται στην βάση δεδομένων. Αν επιλέξουμε “Discard” τότε τα γεωμετρικά και μη χαρακτηριστικά του θεματικού πεδίου επανέρχονται στην κατάσταση που βρίσκονταν πριν αρχίσουμε την διαδικασία επεμβάσεων.

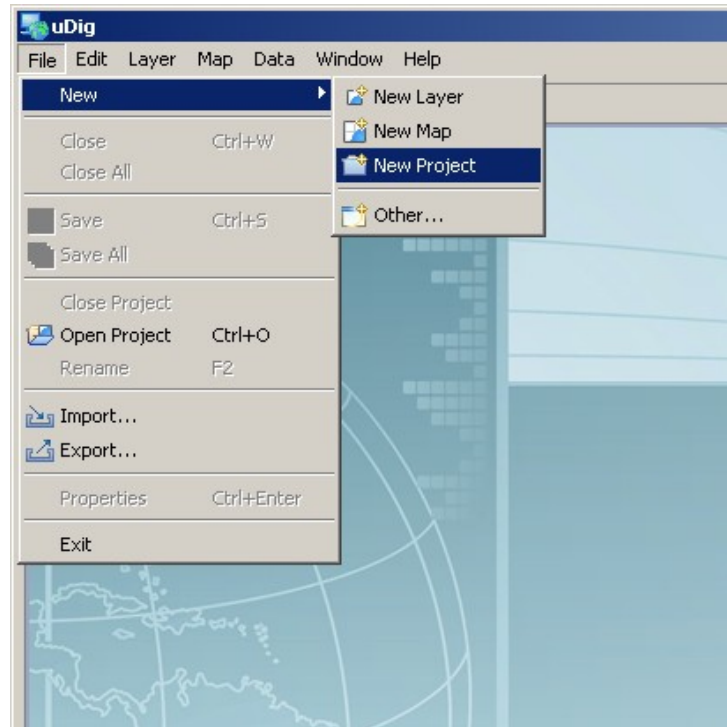
5.3.2 Περιβάλλον UDIG

Εκτός από το QuantumGIS υπάρχει η δυνατότητα να χρησιμοποιήσουμε και άλλα λογισμικά για να δούμε ή να τροποποιήσουμε τα δεδομένα που βρίσκονται αποθηκευμένα στην βάση δεδομένων PostgreSQL/PostGIS. Ένα τέτοιο λογισμικό ΣΓΠ γραφείου είναι το UDIG. Στην ενότητα που ακολουθεί θα αναφερθούμε συνοπτικά στην πρόσβαση και επέμβαση σε χωρικά δεδομένα που βρίσκονται σε βάση δεδομένων PostgreSQL/PostGIS με την χρήση του λογισμικού ΣΓΠ Γραφείου UDIG.

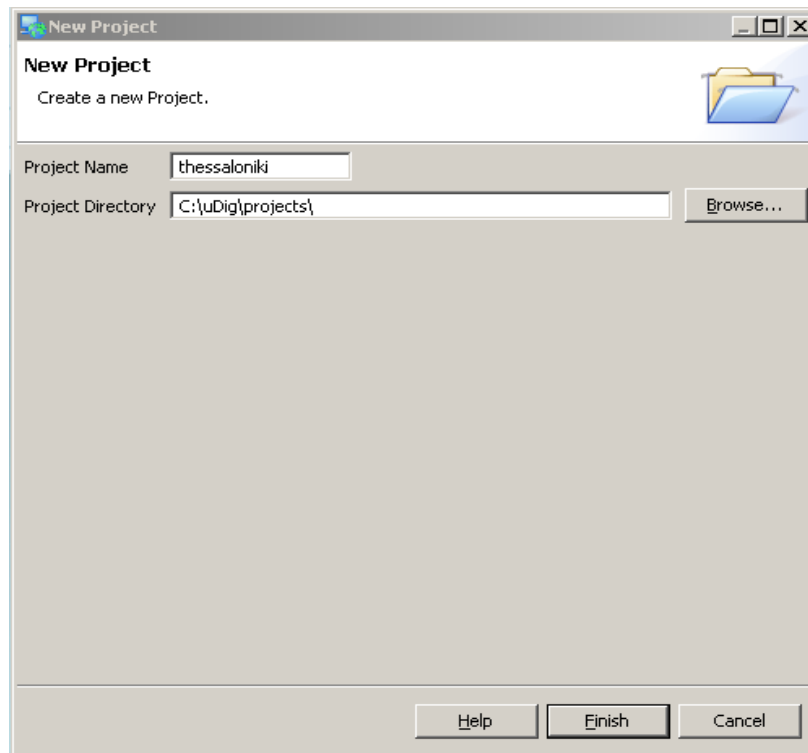
Η εγκατάσταση του UDIG είναι απλή. Εκτελέστε το αρχείο που βρίσκεται στην τοποθεσία <http://udig.refractions.net/files/downloads/udig-1.1.1.exe> ή την εκάστοτε πιο πρόσφατη σταθερή έκδοση του λογισμικού. Συνεχίστε ακολουθώντας τις οδηγίες και αν χρειαστεί συμβουλευτείτε τις οδηγίες που βρίσκονται εδώ: <http://udig.refractions.net/users/>. Όταν ολοκληρωθεί η εγκατάσταση, μπορείτε να ξεκινήσουμε το πρόγραμμα ακολουθώντας Start--> All Programs → UDIG1.1.1--> Udig.

Έπειτα δημιουργούμε διαδοχικά νέο project και νέο χάρτη όπως φαίνεται στις παρακάτω εικόνες.

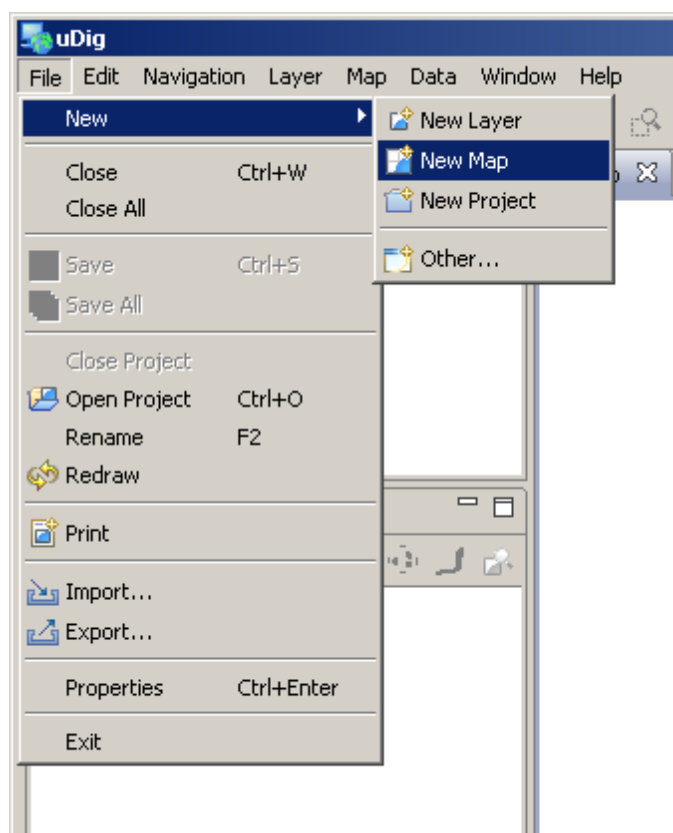
Αρχικά δημιουργούμε νέο project.



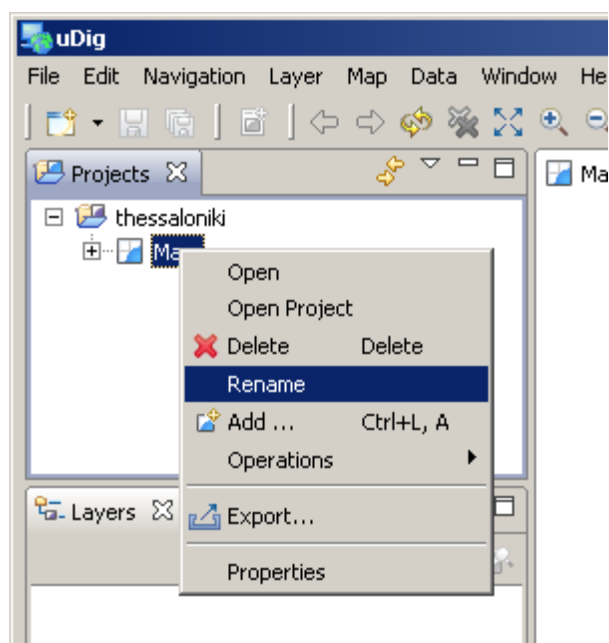
Δίνουμε στο project όνομά (thessaloniki) και ορίζουμε την τοποθεσία που θα αποθηκευθούν οι ρυθμίσεις του (C:\uDig\projects\)



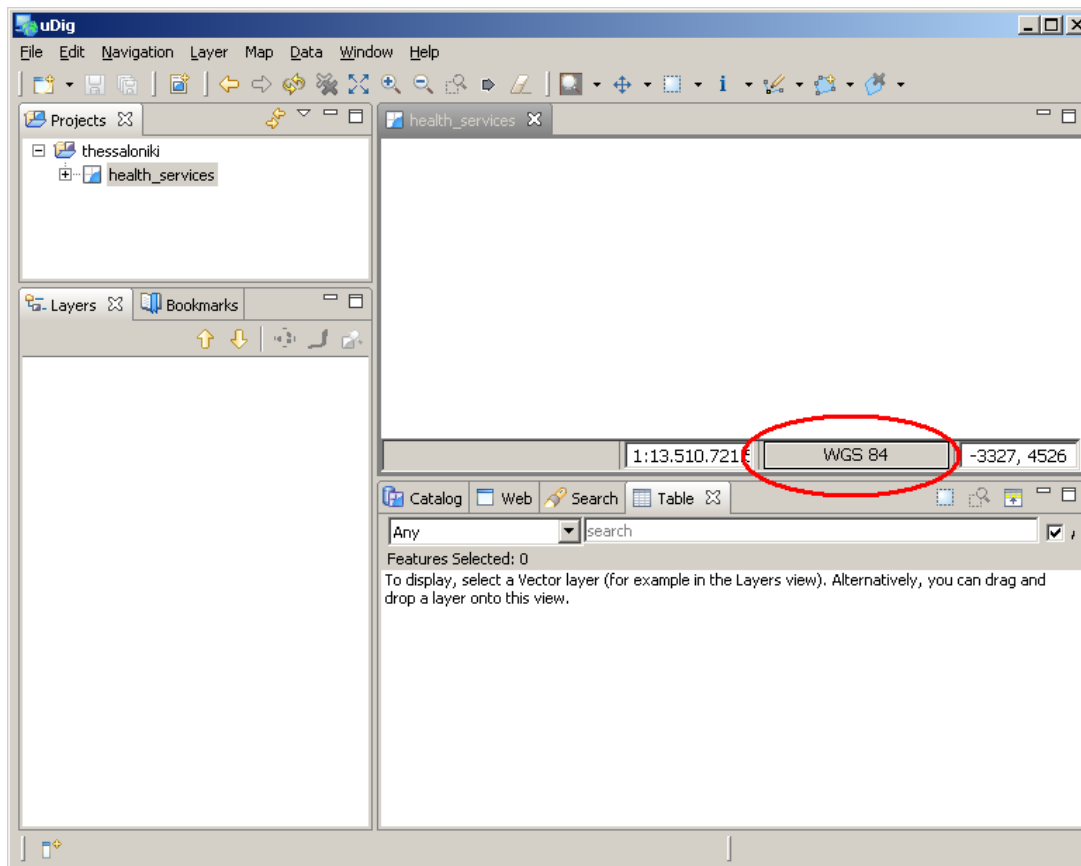
Έπειτα εισάγουμε έναν χάρτη



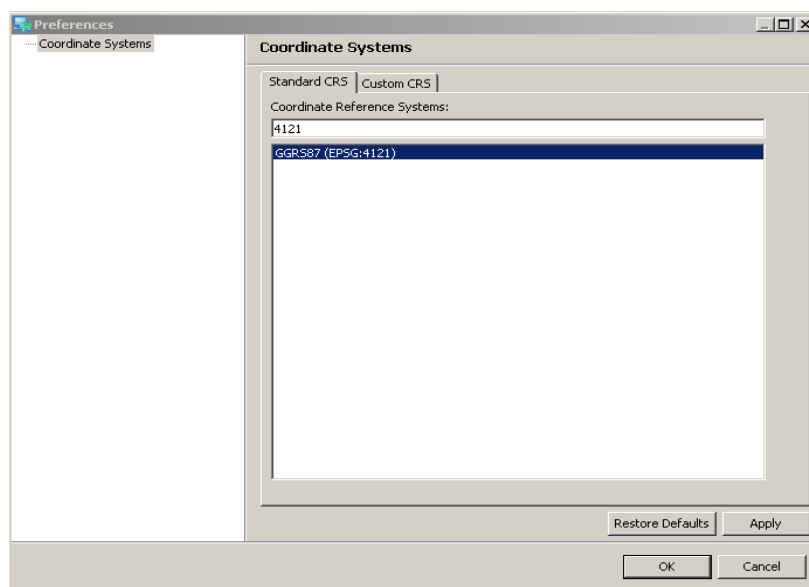
και τον μετονομάζουμε σε health services.



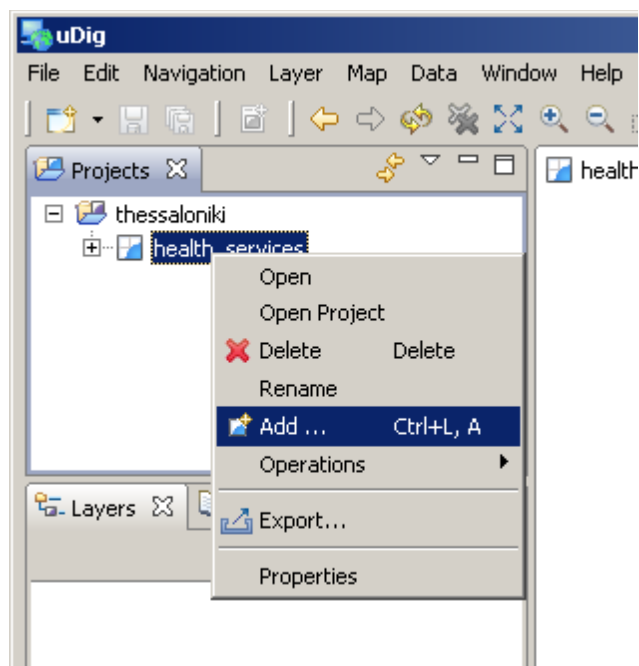
Στην συνέχεια αλλάζουμε το γεωδαιτικό σύστημα αναφοράς σε αυτό που συμφωνεί με τα δεδομένα. Χρησιμοποιούμε το κουμπί που βρίσκεται κοντά στο κέντρο της οθόνης ανάμεσα στην ένδειξη κλίμακας και την ένδειξη με τις συντεταγμένες του κέρσora. Το παράθυρο αυτό τώρα γράφει “WGS84” αφού αυτό είναι το σύστημα που χρησιμοποιείται εξ'ορισμού (by default) στους νέους χάρτες.



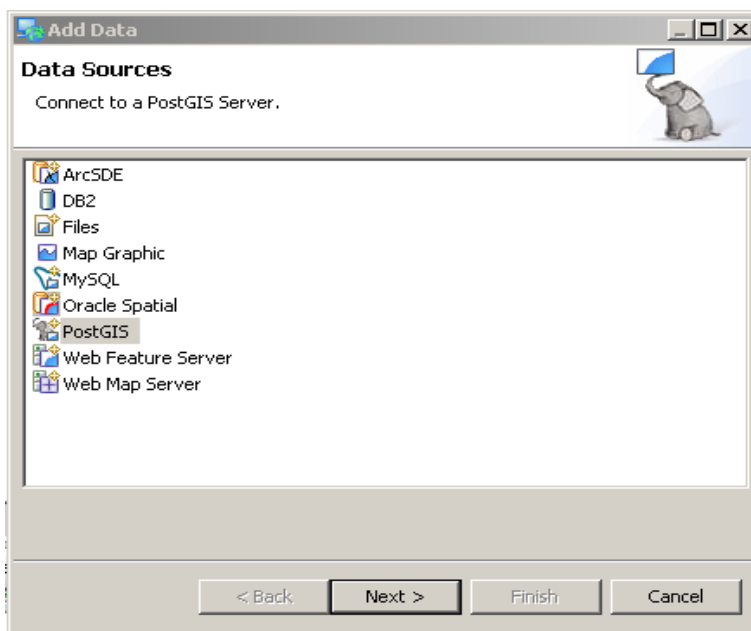
Πιέζουμε το κουμπί “WGS84” και εισάγουμε τον κωδικό 4121 που αντιστοιχεί στο ΕΓΣΑ 87.



Και στην συνέχεια εισάγουμε τα τρία θεματικά επίπεδα. Αρχικά κάνουμε δεξί κλικ στον χάρτη (health services) και επιλέγουμε Add...



Στην συνέχεια PostGIS



και εισάγουμε τις παραμέτρους της σύνδεσης με την βάση δεδομένων όπως κάναμε και νωρίτερα με το λογισμικό QuantunGIS.

Add Data

PostGIS

Connect to a PostGIS Server.

Host: localhost Port: 5432

Username: postgres

Password:

Database: thessaloniki

Schema: health_services

☐ Advanced

< Back Next > Finish Cancel

Στο παράθυρο διαλόγου επιλέγουμε τα διαθέσιμα θεματικά επίπεδα και πατάμε “Finish”

Add Data

Resource Selection

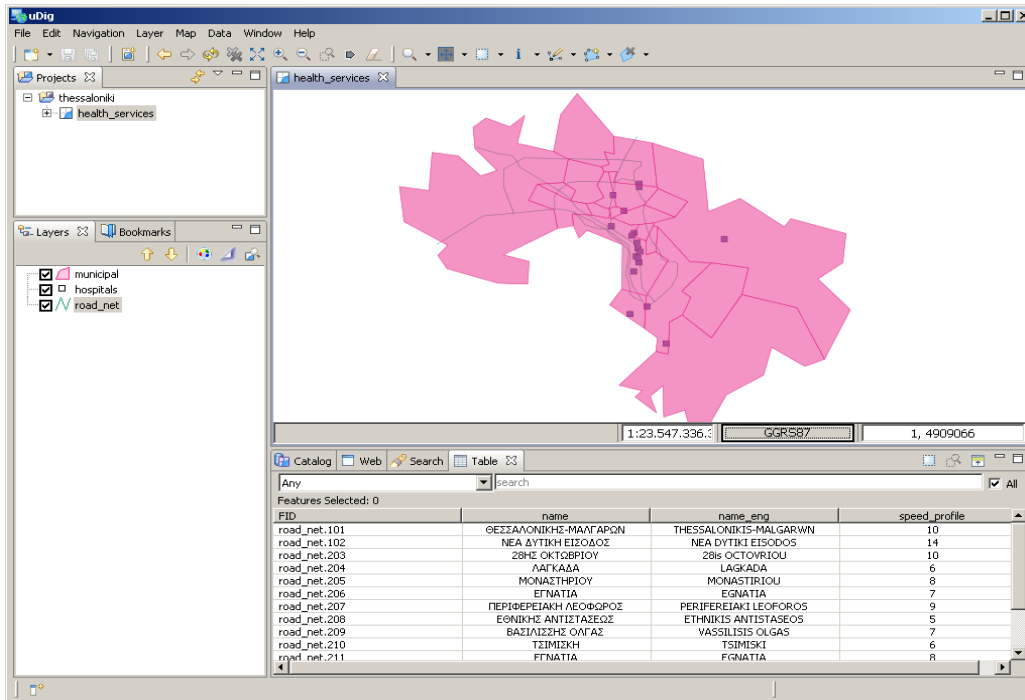
Please select a resource.

- ☒ PostGIS: postgres@localhost/health_services
 - ☐ hospitals
 - ☐ municipal
 - ☐ road_net

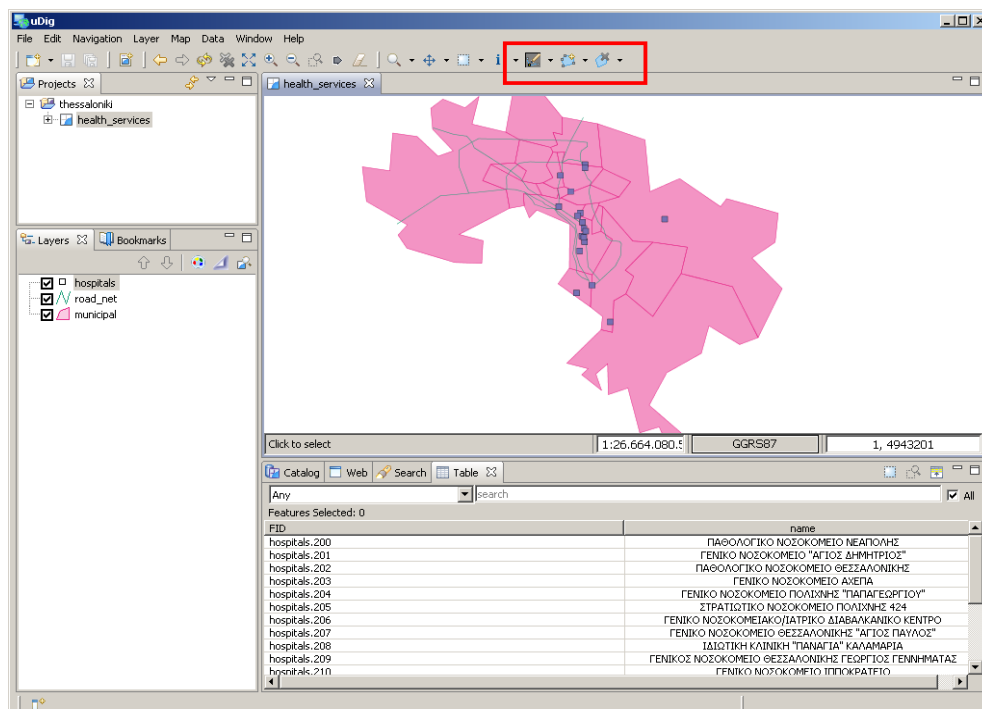
Resources Selected: 0

< Back Next > Finish Cancel

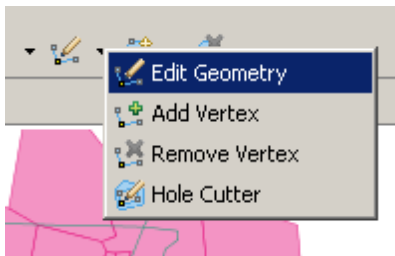
Σε αυτό το σημείο πρέπει να βλέπουμε τα δεδομένα. Αν αυτό δεν συμβαίνει επιλέγουμε “show all data” από τα εργαλεία (περίπου κάτω από το window) ή επιλέγουμε την καρτέλα “Table” και κάνουμε δεξί κλικ σε μία από τις εγγραφές και “show selection”. Αν παρ' όλα αυτά δεν βλέπετε τα δεδομένα τότε επαν-εκκινήστε το πρόγραμμα.



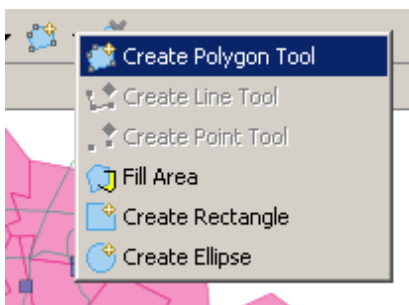
Το uDig παρέχει την δυνατότητα για επέμβασης στα δεδομένα που βρίσκονται αποθηκευμένα στην χωρική βάση δεδομένων. Τα τρία πλήκτρα επέμβασης (τροποποίηση, εισαγωγή, διαγραφή) βρίσκονται πάνω από το παράθυρο του χάρτη (βλ. παρακάτω εικόνα).



Με το πρώτο από αυτά τα εργαλεία μπορούμε να τροποποιήσουμε την γεωμετρία υπαρχόντων στοιχείων. Πατώντας στην γραμμή ενός γραμμικού στοιχείου ή την περίμετρο ενός πολυγώνου να προσθέσουμε ή να αφαιρέσουμε κόμβους. Πατώντας πάνω σε ένα σημειακό στοιχείο μπορούμε να το επιλέξουμε και να το μετακινήσουμε. Επιπλέον μπορούμε να ανοίξουμε μία οπή στην γεωμετρία ενός πολυγώνου.



Με το πλήκτρο που βρίσκεται δεξιά μπορούμε να εισάγουμε πολύγωνο, ορθογώνιο, έλλειψη και να γεμίσουμε περιοχή πολυγώνου. Επιπλέον μπορούμε να ψηφιοποιήσουμε γραμμικά και σημειακά στοιχεία.

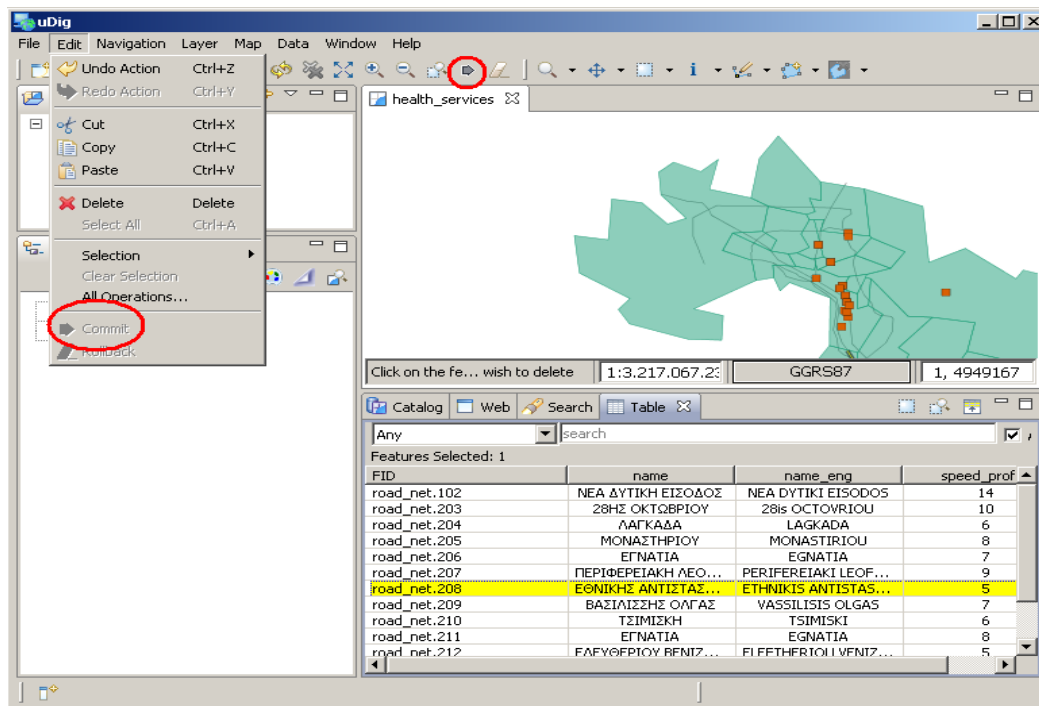


Με το τρίτο στην σειρά πλήκτρο μπορούμε να διαγράψουμε στοιχεία.

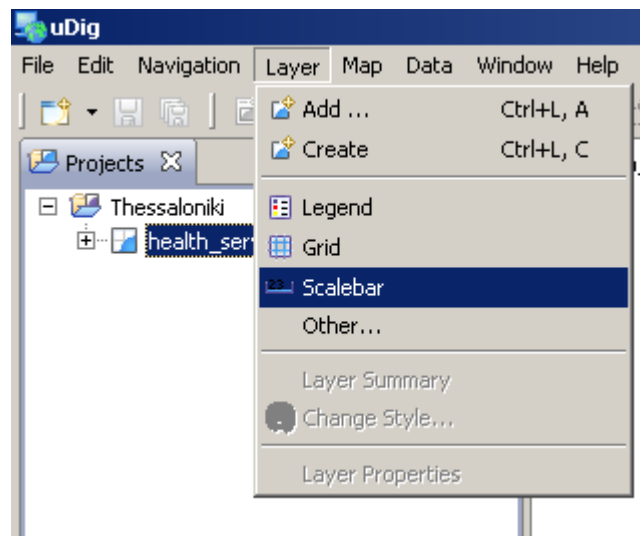


Αν θελήσουμε να αναιρέσουμε τις επεμβάσεις επιλέγουμε “Edit” και στην συνέχεια “Rollback”.

Αν θελήσουμε να επικυρώσουμε τις αλλαγές, να τις μεταφέρουμε δηλαδή στην βάση δεδομένων, επιλέγουμε Edit και στην συνέχεια commit. Μετά από την επικύρωση των αλλαγών στην βάση δεδομένων δεν είναι πια δυνατή η αναίρεσή τους.



Τέλος, το uDig δίνει την δυνατότητα να προσθέσουμε υπόμνημα (legend), κάναβο (Grid) και γραφική κλίμακα (scalebar). Προκειμένου να συμβεί αυτό επιλέγουμε Layer, Legend/Layer, Grid/Layer, Scalebar.



6. Προχωρημένες λειτουργίες

6.1 Η διαδικαστική γλώσσα PL/pgSQL

Η PL/pgSQL (Procedural Language/PostgreSQL Structured Query Language) είναι μία διαδικαστική γλώσσα που υποστηρίζεται από το λογισμικό σύστημα διαχείρισης βάσεων δεδομένων PostgreSQL. Η PL/pgSQL είναι η μόνη διαδικαστική γλώσσα εγκατεστημένη εξορισμού στο λογισμικό PostgreSQL αντίθετα με άλλες διαδικαστικές γλώσσες που είναι συμβατές μαζί του. Τέτοιες είναι οι: PL/Java, PL/Perl, plPHP, PL/Python, PL/R, PL/Ruby, PL/sh, και PL/Tcl.

Η διαδικαστική γλώσσα PL/pgSQL μοιάζει αρκετά με την αντίστοιχη γλώσσα PL/SQL που έχει αναπτυχθεί για το διαδεδομένο εμπορικό λογισμικό σύστημα διαχείρισης βάσεων δεδομένων Oracle. Σύμφωνα με τους δημιουργούς της η PL/pgSQL δημιουργήθηκε με σκοπό:

- Να επιτρέπει την δημιουργία συναρτήσεων και σκανδαλιστών,
- Να προσθέσει δομές ελέγχου στην γλώσσα SQL
- Να επιτρέπει πολύπλοκους υπολογισμούς
- Να επιτρέπει την κληρονομικότητα ορισμένων από τον χρήστη συναρτήσεων, τύπων και τελεστών
- Να μπορεί να οριστεί έτσι ώστε να διαχειρίζεται κάποιον δικτυακό εξυπηρετητή
- Να είναι εύχρηστη

Οι συναρτήσεις που δημιουργούνται με την γλώσσα PL/pgSQL μπορούν να χρησιμοποιηθούν οπουδήποτε υπάρχει περιθώριο για ενσωματωμένες συναρτήσεις (built-in functions). Για παράδειγμα είναι δυνατή η δημιουργία σύνθετων υπολογιστικών συναρτήσεων υπό όρους (conditional computation functions) που αργότερα θα χρησιμοποιηθούν για τον ορισμό τελεστών (operators) ή σε ευρετήρια (index expressions).

Η PL/pgSQL είναι μία πλήρης γλώσσα προγραμματισμού, επιτρέπει πολύ περισσότερο διαδικαστικό έλεγχο από την γλώσσα SQL, και περιλαμβάνει την δυνατότητα για χρήση βρόχων και άλλων δομών ελέγχου. Οι λειτουργίες που δημιουργούνται στην γλώσσα PL/pgSQL μπορούν να κληθούν μέσω εντολών SQL ή να ενεργοποιηθούν μέσω σκανδαλιστών.

Η γλώσσα PL/pgSQL επιτρέπει την ομαδοποίηση υπολογισμών και ερωτημάτων στον εξυπηρετητή και ως εκ τούτου παρέχει τις δυνατότητες μίας διαδικαστικής γλώσσας με την ευκολία της SQL αλλά με

δυνατότητα ελαττωμένης επιβάρυνσης διαδικτυακής επικοινωνίας μεταξύ χρήστη και εξυπηρετητή (client/server communication overhead).

Συνολικά:

- Επιπλέον ερωτήσεις και αποκρίσεις μεταξύ χρήστη και εξυπηρετητή ελαχιστοποιούνται.
- Ενδιάμεσα αποτελέσματα που ο χρήστης δεν χρειάζεται καθαυτά αλλά έχουν σημασία για επόμενα ερωτήματα δεν χρειάζεται να επεξεργαστούν από τον χρήστη και να μεταφερθούν ανάμεσα σε αυτόν και τον εξυπηρετητή.
- Πολλαπλοί κύκλοι επεξεργασίας και υποβολής ερωτημάτων μπορούν να αποφευχθούν.
- Τα παραπάνω συμβάλλουν στην δραματική βελτίωση της υπολογιστικής απόδοσης και ταχύτητας σε σχέση με ένα σύστημα που δεν αποθηκεύει συναρτήσεις (functions). Επιπλέον η γλώσσα PL/pgSQL μπορεί να συμπεριλάβει όλους τους τύπους δεδομένων, τελεστές και λειτουργίες της γλώσσας SQL.

Στην συνέχεια του οδηγού αξιοποιούνται τα στοιχεία ελέγχου που προσφέρει η PL/pgSQL όπως οι σκανδαλιστές και οι συναρτήσεις. Έτσι συντάσσονται στοιχεία ελέγχου που θα αποθηκευτούν στο σύστημα διαχείρισης βάσεων δεδομένων και θα κληθούν από τον διαχειριστή της βάσης δεδομένων ή να εκτελούνται από σκανδαλιστές που παρακολουθούν βασικές λειτουργίες του συστήματος διαχείρισης της χωρικής βάσης δεδομένων.

6.2 Σύνταξη και χρήση στοιχείων ελέγχου σε PL/pgSQL

Προκειμένου να κάνουμε ανάγλυφο τον τρόπο λειτουργίας των στοιχείων ελέγχου που δομούνται με την διαδικαστική γλώσσα PL/pgSQL θα ξεκινήσουμε με ένα παράδειγμα. Θα υποθέσουμε ότι το σύστημα που δημιουργήσαμε μέχρι τώρα χρησιμοποιείται από την υπηρεσία που μεταφέρει στο νοσοκομείο τα θύματα ατυχημάτων (πχ ΕΚΑΒ). Σε αυτό το παράδειγμα ο υπάλληλος που δέχεται τα τηλεφωνήματα εισάγει την τοποθεσία των ατυχημάτων στο σύστημα και επικοινωνεί με το πλήρωμα του ασθενοφόρου που βρίσκεται σταθμευμένο στην πλησιέστερη θέση. Στην πραγματικότητα τα ασθενοφόρα βρίσκονται σταθμευμένα σε διάφορα σημεία στην πόλη. Εδώ ας υποθέσουμε ότι τα ασθενοφόρα βρίσκονται σταθμευμένα στα νοσοκομεία. Συνεπώς ο υπάλληλος που δέχεται τα τηλεφωνήματα πρέπει να γνωρίζει άμεσα ποιο είναι το πλησιέστερο νοσοκομείο σε κάθε περιστατικό. Στο παράδειγμα που ακολουθεί θα αναπτύξουμε έναν μηχανισμό που θα παρέχει στον υπάλληλο αυτή την πληροφορία. Επιπλέον ο υπάλληλος θα πρέπει να γνωρίζει πόσο μακριά από το σημείο στάθμευσης του ασθενοφόρου είναι το συμβάν. Ο σκανδαλιστής και η συνάρτηση σκανδαλιστή που θα δημιουργήσουμε εδώ θα ενημερώνουν αυτόματα κάθε εγγραφή ενός πίνακα που αφορά σε περιστατικά για το πλησιέστερο (σε κάθε περιστατικό) νοσοκομείο και την απόσταση από αυτό.

Προκειμένου να εισάγουμε την έννοια των περιστατικών στο σύστημα μας θα δημιουργήσουμε έναν πίνακα. Τον πίνακα αυτό θα ονομάσουμε incidents. Σε αυτό τον πίνακα θα εισάγουμε τα ακόλουθα στοιχεία:

Α/α περιστατικού.

Χρονική στιγμή που το περιστατικό εισήχθη στο σύστημα.

Όνομα πλησιέστερου νοσοκομείου.

Απόσταση από το πλησιέστερο νοσοκομείο.

Θέση περιστατικού.

Θα δημιουργήσουμε έναν νέο πίνακα με αυτά τα γνωρίσματα με την ακόλουθη εντολή:

```
CREATE TABLE health_services.incidents
```

```
(gid Integer PRIMARY KEY,  
time TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,  
near_hosp_name character varying(200),  
dist NUMERIC);
```

```
SELECT AddGeometryColumn('health_services','incidents','the_geom','4121',  
'POINT',2);
```

Το σύστημα θα λαμβάνει πληροφορίες για νέα συμβάντα μέσω ενός συστήματος πελάτη (λχ μίας φόρμας) με εντολές SQL όπως η ακόλουθη:

```
INSERT INTO health_services.incidents VALUES (A/A,CURRENT_TIMESTAMP, 9999, 9999,  
GeomFromText('POINT( X Y )',4121));
```

Ο υπάλληλος αρχικά θα γνωρίζει μόνο την τοποθεσία όπου πρέπει να πάει το ασθενοφόρο. Η τιμή της χρονικής στιγμής που εισάγεται η εγγραφή στο σύστημα συμπληρώνεται αυτόματα από την βάση δεδομένων με την χρήση της λειτουργίας CURRENT_TIMESTAMP. Οι τιμές του πλησιέστερου νοσοκομείου και της απόστασης από αυτό θα συμπληρώνονται με 9999 (που υποδεικνύει ότι η τιμή του συγκεκριμένου πεδίου είναι άγνωστη στον χρήστη) και θα συμπληρώνονται αυτόματα από την βάση δεδομένων με την συνάρτηση που θα φτιάξουμε στην συνέχεια.

6.2.1 Συνάρτηση

Η συνάρτηση που θα φτιάξουμε θα λειτουργεί με τα εξής απλά βήματα:

Αρχικά η συνάρτηση θα πρέπει να διαβάζει τον πίνακα περιστατικών και να απομονώνει το περιστατικό για το οποίο πρέπει να γίνει ο υπολογισμός. Επειδή η συνάρτηση θα ενεργοποιείται κάθε φορά που εισάγεται νέο στοιχείο στον πίνακα των περιστατικών (αυτό θα εξηγηθεί στην επόμενη παράγραφο) τότε το περιστατικό για το οποίο θα γίνεται ο υπολογισμός θα είναι το πλέον πρόσφατο περιστατικό. Το περιστατικό θα ταυτοποιείται με τον αύξοντα αριθμό του. Συνολικά λοιπόν σε αυτό το βήμα η συνάρτηση θα βρίσκει τον αύξοντα αριθμό του πιο πρόσφατου περιστατικού και θα τον κρατά στην μνήμη.

Αυτό το βήμα μπορεί να υλοποιηθεί με το εξής απλό ερώτημα SQL:

```
LATEST_INCIDENT_ID = (SELECT gid  
FROM health_services.incidents  
ORDER BY time desc  
limit 1);
```

Στο επόμενο βήμα η συνάρτηση θα υπολογίζει ποιο είναι το πλέον κοντινό νοσοκομείο στην θέση του περιστατικού που προσδιορίστηκε στο παραπάνω βήμα. Αυτό θα γίνει με χρήση του τελεστή DISTANCE που περιγράψαμε νωρίτερα:

```
NEAREST_HOSP = (SELECT hospital FROM  
(SELECT name AS Hospital, DISTANCE(H.the_geom, I.the_geom) AS distance  
FROM health_services.hospitals H, health_services.incidents I  
WHERE I.gid = LATEST_INCIDENT_ID  
ORDER BY DISTANCE asc) AS NEARBY_HOSPITALS  
limit 1);
```

Στην συνέχεια με την χρήση του ίδιου ερωτήματος θα απομονώνεται η απόσταση από το πλησιέστερο νοσοκομείο:

```
HOSP_DIST = (SELECT DISTANCE
FROM
(SELECT name AS Hospital, DISTANCE(H.the_geom, I.the_geom) AS distance
FROM health_services.hospitals H, health_services.incidents I
WHERE I.gid = LATEST_INCIDENT_ID
ORDER BY DISTANCE asc) AS NEARBY_HOSPITALS
limit 1);
```

Αφού η συνάρτηση έχει προσδιορίσει αυτές τις παραμέτρους τότε μπορεί να εισάγει τις τιμές τους στα ανάλογα πεδία της πιο πρόσφατης εγγραφής. Αυτό γίνεται επίσης με εντολές SQL:

```
UPDATE health_services.incidents
SET near_hosp_name = NEAREST_HOSP, dist = HOSP_DIST
WHERE GID = LATEST_INCIDENT_ID;
```

Αν όλα αυτά τα βήματα τα συνθέσουμε σε μία εντολή δημιουργίας συνάρτησης σε γλώσσα PL/pgSQL τότε αυτή θα μοιάζει ως εξής:

```
CREATE OR REPLACE FUNCTION FIND_NEAREST_HOSPITAL() RETURNS trigger AS '
```

```
DECLARE
LATEST_INCIDENT_ID INTEGER;
NEAREST_HOSP character varying(200);
HOSP_DIST NUMERIC;
```

```
BEGIN
```

```
LATEST_INCIDENT_ID = (SELECT gid
FROM health_services.incidents
ORDER BY time desc
limit 1);
```

```
NEAREST_HOSP = (SELECT hospital FROM
(SELECT name AS Hospital, DISTANCE(H.the_geom, I.the_geom) AS distance
FROM health_services.hospitals H, health_services.incidents I
WHERE I.gid = LATEST_INCIDENT_ID
ORDER BY DISTANCE asc) AS NEARBY_HOSPITALS
limit 1);
```

```
HOSP_DIST = (SELECT DISTANCE
FROM
(SELECT name AS Hospital, DISTANCE(H.the_geom, I.the_geom) AS distance
FROM health_services.hospitals H, health_services.incidents I
WHERE I.gid = LATEST_INCIDENT_ID
ORDER BY DISTANCE asc) AS NEARBY_HOSPITALS
limit 1);
```

```
UPDATE health_services.incidents
```

```
SET near_hosp_name = NEAREST_HOSP, dist = HOSP_DIST  
WHERE GID = LATEST_INCIDENT_ID;
```

```
RETURN new;  
END  
'Language plpgsql;
```

Προκειμένου να δημιουργήσουμε αυτή την συνάρτηση εκτελούμε το παραπάνω κομμάτι κώδικά στο παράθυρο ερωτημάτων του pgAdminIII. Αν η δημιουργία της συνάρτησης λειτουργήσει όπως πρέπει τότε λαμβάνουμε το παρακάτω μήνυμα:

“Query returned successfully with no result in ... ms”

Πέρα από τα ερωτήματα SQL που διατυπώσαμε νωρίτερα έχουμε εισάγει και κάποια επιπλέον στοιχεία στον κώδικά μας:

“CREATE OR REPLACE FUNCTION FIND_NEAREST_HOSPITAL() RETURNS trigger AS ' ”

Ξεκινούμε την δημιουργία της συνάρτησης, δηλώνουμε το όνομα της, προσδιορίζουμε ότι θα χρησιμοποιηθεί μαζί με σκανδαλιστή (βλ. Επόμενη παράγραφο) και ξεκινάμε το τμήμα κώδικα που προσδιορίζει την συνάρτηση.

Αρχικά, δηλώνουμε τρεις παραμέτρους που θα χρησιμοποιεί η συνάρτηση.

```
DECLARE  
LATEST_INCIDENT_ID INTEGER;  
NEAREST_HOSP character varying(200);  
HOSP_DIST NUMERIC;
```

Μόλις εκτελεσθεί η συνάρτηση οι τρεις παράμετροι παίρνουν τιμές.

```
BEGIN
```

```
LATEST_INCIDENT_ID = (SELECT gid  
                      FROM health_services.incidents  
                      ORDER BY time desc  
                      limit 1);
```

```
NEAREST_HOSP = (SELECT hospital FROM  
                (SELECT name AS Hospital, DISTANCE(H.the_geom, I.the_geom) AS distance  
                 FROM health_services.hospitals H, health_services.incidents I  
                 WHERE I.gid = LATEST_INCIDENT_ID  
                 ORDER BY DISTANCE asc) AS NEARBY_HOSPITALS  
                limit 1);
```

```
HOSP_DIST = (SELECT DISTANCE  
             FROM  
             (SELECT name AS Hospital, DISTANCE(H.the_geom, I.the_geom) AS distance  
              FROM health_services.hospitals H, health_services.incidents I  
              WHERE I.gid = LATEST_INCIDENT_ID
```

```
ORDER BY DISTANCE asc) AS NEARBY_HOSPITALS  
limit 1);
```

Η παράμετρος LATEST_INCIDENT_ID παίρνει την τιμή του πλέον πρόσφατου περιστατικού

Η παράμετρος NEAREST_HOSP παίρνει το όνομα του κοντινότερου νοσοκομείου προς το πλέον πρόσφατο συμβάν.

Η παράμετρος HOSP_DIST παίρνει την τιμή της απόστασης του πιο πρόσφατου συμβάντος από το κοντινότερο νοσοκομείο.

Η πρώτη από αυτές τις παραμέτρους (LATEST_INCIDENT_ID) χρησιμοποιείται προκειμένου να προσδιοριστούν οι δύο επόμενες (NEAREST_HOSP, HOSP_DIST).

```
UPDATE health_services.incidents  
SET near_hosp_name = NEAREST_HOSP, dist = HOSP_DIST  
WHERE GID = LATEST_INCIDENT_ID;
```

Οι παράμετροι NEAREST_HOSP και HOSP_DIST εισάγονται στην εγγραφή του πιο πρόσφατου συμβάντος.

```
RETURN new;  
END  
'Language plpgsql;
```

Οι συναρτήσεις στην γλώσσα PL/pgSQL πρέπει να επιστρέφουν μία τιμή που να είναι συμβατή με τον τύπο της συνάρτησης όπως αυτός ορίστηκε στην εντολή CREATE FUNCTION στην αρχή της συνάρτησης. Η new είναι μία τέτοια μεταβλητή συστήματος (βλ. <http://www.postgresql.org/ons/8.3/interactive/plpgsql-trigger.html>). Στην συνέχεια ολοκληρώνεται η σειρά ενεργειών της συνάρτησης και δηλώνεται ότι το κομμάτι κώδικα που προηγήθηκε έχει γραφτεί σε γλώσσα PL/pgSQL.

6.2.2 Σκανδαλιστής

Στην προηγούμενη παράγραφο δημιουργήσαμε μία συνάρτηση που μπορεί να εκτελεσθεί με έναν σκανδαλιστή. Η υλοποίηση της συνάρτησης έχει ολοκληρωθεί αλλά ο σκανδαλιστής καθαυτός δεν έχει υλοποιηθεί ακόμα. Αυτό που θα κάνει ο σκανδαλιστής είναι να εκτελεί την συνάρτηση μετά από την ικανοποίηση κάποιας ορισμένης από τον χρήστη συνθήκης. Στην περίπτωση μας ο σκανδαλιστής θα ανιχνεύει την εκτέλεση της λειτουργίας INSERT στον πίνακα incidents και θα εκτελεί την συνάρτηση που ορίσαμε.

Ο σκανδαλιστής υλοποιείται με την ακόλουθη εντολή:

```
CREATE TRIGGER FIND_NEAREST_HOSPITAL AFTER INSERT ON health_services.incidents  
FOR EACH ROW EXECUTE PROCEDURE FIND_NEAREST_HOSPITAL();
```

Εκτελούμε την παραπάνω εντολή στο παράθυρο ερωτημάτων του pgAdminIII, όπως κάναμε νωρίτερα με την εντολή δημιουργίας της συνάρτησης. Στην συνέχεια δοκιμάζουμε την λειτουργία του σκανδαλιστή και της συνάρτησης:

Με τις ακόλουθες εντολές SQL εισάγουμε τρία συμβάντα στον πίνακα incidents. Είναι σημαντικό τα

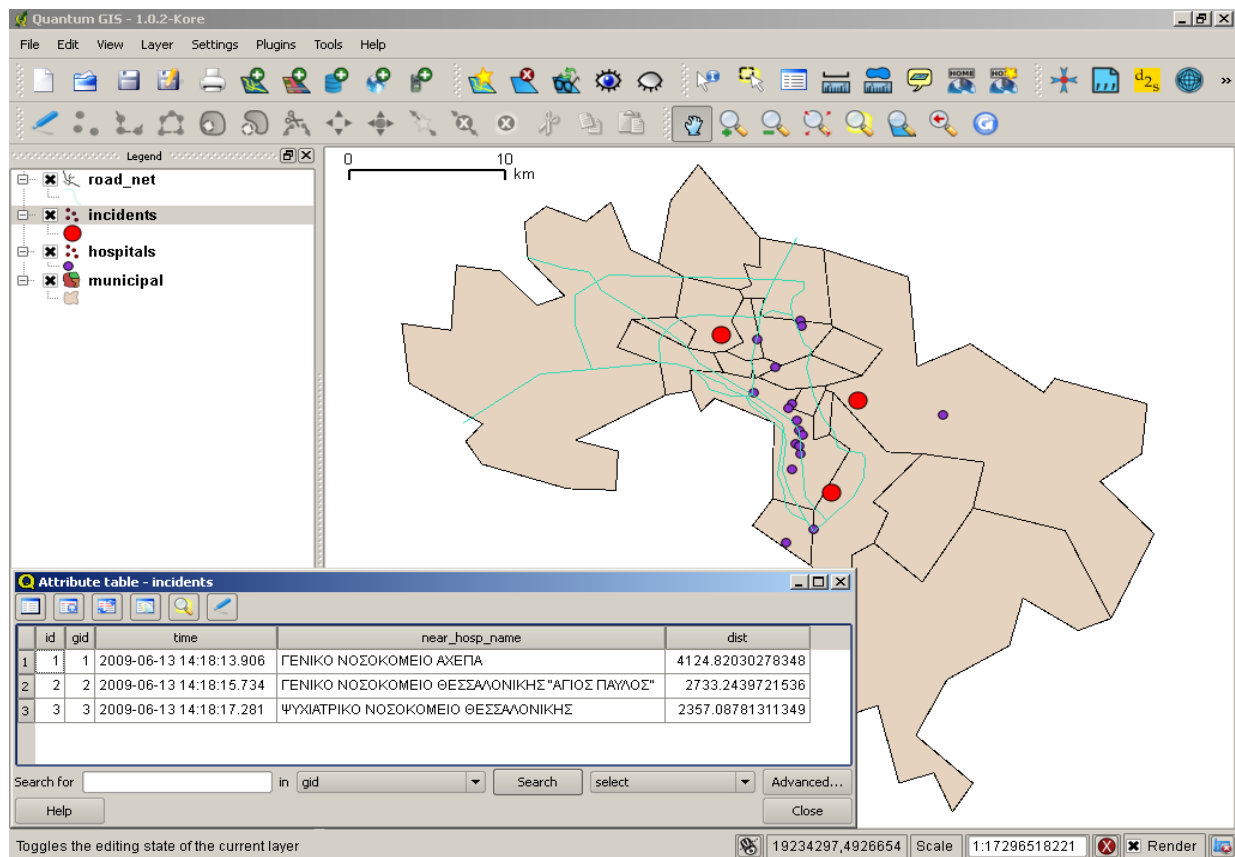
συμβάντα να καταχωρηθούν ένα ένα και όχι όλα μαζί.

```
INSERT INTO health_services.INCIDENTS VALUES (001,CURRENT_TIMESTAMP, 9999, 9999,  
GeomFromText('POINT(  
19257758.86538117 4931260.15004108)',4121));
```

```
INSERT INTO health_services.INCIDENTS VALUES (002,CURRENT_TIMESTAMP, 9999, 9999,  
GeomFromText('POINT(  
19256050.61425355 4924903.86677552)',4121));
```

```
INSERT INTO health_services.INCIDENTS VALUES (003,CURRENT_TIMESTAMP, 9999, 9999,  
GeomFromText('POINT(  
19249019.42500000 4935754.75000000)',4121));
```

Τα συμβάντα έχουν εισαχθεί. Τώρα μπορούμε να ελέγξουμε τα περιεχόμενα του πίνακα incidents. Ας το κάνουμε μέσω του QuantumGIS.



Πράγματι, στον πίνακα του θεματικού επιπέδου των περιστατικών (incidents) έχουν προστεθεί πληροφορίες πέραν αυτών που εισάγαμε με τις παραπάνω εντολές INSERT. Συγκεκριμένα, σε κάθε εγγραφή έχει εισαχθεί το όνομα του πλησιέστερου νοσοκομείου και η απόσταση από αυτό (στήλες "near_hosp_name" και "dist").

7. Σύνοψη

Ο καθημερινά αυξανόμενος όγκος διακινούμενων χωρικών πληροφοριών οδήγησε στην ανάπτυξη βάσεων δεδομένων που να είναι σε θέση να τα αποθηκεύσουν, διαχειριστούν και ανακτήσουν (να υποβάλουν ερωτήματα) σε τέτοιου είδους δεδομένα. Ένα τέτοιο σύστημα αποτελείται από τα ανοικτού κώδικα λογισμικά PostgreSQL/PostGIS/QuantumGIS. Ο παρών οδηγός έχει ως σκοπό την εξοικείωση με αυτό το σύστημα. Με την βοήθεια του οδηγού ο χρήστης θα είναι σε θέση να εγκαταστήσει τα επιμέρους λογισμικά, να δημιουργήσει χωρικές βάσεις δεδομένων και επιμέρους στοιχεία όπως πίνακες, σχήματα, όψεις, περιορισμούς, και στοιχεία ελέγχου όπως συναρτήσεις και σκανδαλιστές. Θα μπορεί να εισάγει δεδομένα με διαδοχικές εντολές INSERT, με την χρήση του προγράμματος shp2pgsql καθώς και με χρήση του λογισμικού QuantumGIS, καθώς και να εξάγει δεδομένα σε αρχείο τύπου shapefile. Θα πρέπει ακόμα να μπορεί να οπτικοποιήσει τα δεδομένα που περιέχονται στην βάση δεδομένων στο QuantumGIS αλλά και να επέμβει σε αυτά τροποποιώντας τα σε γραφικό περιβάλλον. Ακόμα θα πρέπει να μπορεί να δημιουργήσει ερωτήματα που να αφορούν σε χωρικά και μη χαρακτηριστικά των δεδομένων, τόσο στο γραφικό περιβάλλον του QuantumGIS όσο και σε παράθυρο ερωτημάτων με γλώσσα SQL. Στην ενότητα των χωρικών ερωτημάτων δίνονται παραδείγματα σχετικά με τους τελεστές WITHIN/CONTAINS, DISTANCE, INTERSECTS, TOUCHES καθώς και τους γεωμετρικούς τελεστές AREA, LENGTH, ENVELOPE, NPOINTS.

8. Πηγές

- PostgreSQL: <http://www.postgresql.org/docs/>
- PostGIS: <http://postgis.refractory.net/documentation/>
- QuantumGIS:
<http://www.qgis.org/en/documentation.html><http://www.qgis.org/en/documentation.html>
- Open GIS Consortium, Inc. OpenGIS Simple Features Specification For SQL. Revision 1.1
OpenGIS Project Document 99-049. May 05/05/1999.
http://portal.opengeospatial.org/files/?artifact_id=829
- Ramsey, P. "Introduction to PostGIS": Workshop prepared for the FOSS4G 2007:
<http://2007.foss4g.org/workshops/W-04/>
- Boston GIS, Part 1: Getting Started With PostGIS: An almost Idiot's Guide.
http://www.bostongis.com/PrinterFriendly.aspx?content_name=postgis_tut01
- Κώστας Πατρούμπας, Υλικό σχετικά με τα λογισμικά PostgreSQL και PostGIS για το μάθημα "Χωρικές Βάσεις Δεδομένων", ΔΠΜΣ Γεωπληροφορική, ΕΜΠ.
http://www.dblab.ece.ntua.gr/~kpatro/geodb/examples/GeoDB2008_PostGIS_Lab.pdf
http://www.dblab.ece.ntua.gr/~kpatro/geodb/examples/GeoDB2008_PostgreSQL_Lab.pdf
- Δεσμοί σχετικά με την σύνταξη σε γλώσσα SQL:
 - <http://www.freestuff.gr/forums/viewtopic.php?t=15660>
 - http://www.survey.ntua.gr/main/courses/general/dbintro/lectures/db2005_lecture07.pdf
 - <http://www.cs.uoi.gr/~pitoura/db2000/sql2/index.htm>