

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΜΑΘΗΜΑ: ΣΥΣΤΗΜΑΤΑ ΓΕΩΓΡΑΦΙΚΩΝ ΠΛΗΡΟΦΟΡΙΩΝ

(8^ο εξ. Προπτυχιακού – 2^ο εξ. Μεταπτυχιακού - εαρινό εξάμηνο 2009-2010)

Υπεύθυνος Εργαστηρίων: Δρ. Ηλίας Φρέντζος

Εισαγωγή στη PostgreSQL - PostGIS

Δρ. Ηλίας Φρέντζος

1. Γενικά

Σκοπός του εργαστηρίου είναι η εξοικείωση των σπουδαστών με τη βάση δεδομένων ανοικτού PostgreSQL και την επέκταση PostGIS για υποστήριξη χωρικών λειτουργιών, καθώς και το (επίσης ανοικτού κώδικα) λογισμικό GIS, Quantum GIS. Το εργαστήριο έχει σαν στόχο την παρουσίαση των βασικών λειτουργιών της χωρικής βάσης δεδομένων και του συστήματος γεωγραφικών πληροφοριών. Λίγα λόγια για το λογισμικό που θα χρησιμοποιηθεί:

Δημιουργία Γεωγραφικών Βάσεων Δεδομένων

Οι γεωγραφικές βάσεις δεδομένων βασίζονται στην ύπαρξη ενός Συστήματος Διαχείρισης Βάσεων Δεδομένων που θα έχει δυνατότητες υποστήριξης χωρικών τύπων (γεωμετριών) όπως σημεία, γραμμές και επιφάνειες. Ένα ΣΔΒΔ μπορεί να υποστηρίξει χωρικούς τύπους αυτόνομα ή να βασίζεται σε μία χωρική επέκταση για αυτό το σκοπό. Οι γεωγραφικές βάσεις δεδομένων μπορεί να περιέχουν διανυσματικά (vector) ή ψηφιδωτά (raster) δεδομένα. Η συνήθης τακτική όμως είναι να εισάγονται στο ΣΔΒΔ μόνο τα διανυσματικά δεδομένα, ενώ τα raster αποθηκεύονται σε διάφορες θέσεις στο σύστημα αρχείων (file system) του η/υ που υποστηρίζει τη λειτουργία του Συστήματος Γεωγραφικών Πληροφοριών. Στο μάθημα θα ασχοληθούμε εκτενώς με το ΣΔΒΔ PostgreSQL και την επέκταση χωρική PostGIS, τα οποία δεν έχουν τη δυνατότητα διαχείρισης raster δεδομένων.

Η υποστήριξη χωρικών τύπων από τις βάσεις δεδομένων βασίζεται στην ύπαρξη του τύπου δεδομένων geometry, ο οποίος χρησιμοποιείται για να αναπαραστήσει μια ποικιλία χωρικών δεδομένων. Η ύπαρξη επίσης χωρικής υποστήριξης σημαίνει ότι το ΣΔΒΔ πρέπει να υποστηρίζει διάφορα συστήματα προβολής και μετασχηματισμούς μεταξύ τους, χωρικούς τελεστές (π.χ. το αντικείμενο Α επικαλύπτεται με το αντικείμενο Β) συναρτήσεις (π.χ. υπολογισμός του εμβαδού μίας επιφανειακής οντότητας) και χωρικά ευρετήρια.

Όταν δημιουργείται μία Χωρική Βάση Δεδομένων, κάθε οντότητα με χωρική υπόσταση (π.χ. ένα σημείο ενδιαφέροντος), είναι μία εγγραφή σε έναν πίνακα που περιέχει όλα τα αντίστοιχα σημεία, ενώ μεταξύ των πεδίων του πίνακα υφίσταται και ένα ξεχωριστό πεδίο για τη γεωμετρία του αντικειμένου (π.χ. τη θέση του σημείου ενδιαφέροντος, η οποία δίνεται από τις συντεταγμένες του). Για να επανέλθουμε στο προηγούμενο παράδειγμα, ένας πίνακας με σημεία ενδιαφέροντος θα περιέχει τα εξής πεδία : id, name, type, geometry.

PostgreSQL (<http://www.postgresql.org/>): Το λογισμικό της Βάσης Δεδομένων PostgreSQL, είναι μία από τις δημοφιλέστερες, παγκοσμίως, **open source** βάσεις δεδομένων με ισχυρή **υποστήριξη τύπων γεωγραφικών δεδομένων** και μεγάλης κλίμακας εφαρμογές και κοινότητες. Η PostgreSQL λειτουργεί σε Windows (95 / 98 / ME / NT / 2000 / XP), Linux (RedHat / Mandrake / Suse), MacOS X. Παρέχονται γλώσσες προγραμματισμού και interfaces: Perl, Python, C/C++, Embedded SQL, Delphi/Kylix/Pascal, VB, ASP, Java, ODBC, JDBC κ.α. Η διαχείριση γίνεται κυρίως μέσω του PgAdmin III, αλλά και άλλες εφαρμογές τρίτων: (PgAccess , PhpPgAdmin, WinSQL). Το εργαστήριο είναι σχεδιασμένο για χειρισμό της βάσης δεδομένων μέσω κονσόλας, ενώ και η χρήση του PgAdmin III είναι ευπρόσδεκτη. Πρέπει να σημειωθεί ότι αν και η PostgreSQL

υποστηρίζει από μόνη της χωρικούς τύπους δεδομένων, αυτοί δεν ακολουθούν το πρότυπο OGC (Open GIS Consortium).

PostGIS (<http://postgis.refractory.net/>): Είναι μία επέκταση της PostgreSQL για να υποστηρίζει χωρικά δεδομένα, σύμφωνα με το πρότυπο του OGC. Παρέχει ειδικούς τελεστές για τη σύνταξη ερωτημάτων, λειτουργίες συνάθρισης επάνω σε χωρικά δεδομένα καθώς και χωρικές συναρτήσεις. Επιτρέπει επίσης την ανάθεση προβολικών συστημάτων στα χωρικά δεδομένα. Τέλος, μπορεί να χρησιμοποιηθεί για να οπτικοποιηθούν τα δεδομένα μέσω ειδικών εφαρμογών όπως το Quantum GIS και ο Geo Server.

Ο τύπος Geometry και ο μορφότυπος Well Known Text (WKT)

Ο τύπος Geometry του PostGIS περιέχει όλη τη γεωμετρία (τις συντεταγμένες) του κάθε αντικειμένου. Ο τύπος αυτός είναι γενικός και μπορεί να περιέχει χωρικά αντικείμενα των τριών συνηθέστερων κατηγοριών (σημείο, γραμμή, πολύγωνο), καθώς και συλλογές αυτών των αντικειμένων (δηλαδή πολλά πολύγωνα ή πολλά σημεία μαζί με κάποιες γραμμές κ.ο.κ.).

Η αναπαράσταση του τύπου Geometry σε μορφή πίνακα δε μπορεί να γίνει εύκολα αντιληπτή. Στην πραγματικότητα, κάτι τέτοιο είναι σχεδόν αδύνατο να συμβεί, από τη στιγμή που οι γεωμετρίες αναπαρίστανται από δεκαεξαδικές μορφές όπως η παρακάτω:

```
"010600002034080000010000000103000000070000008BC3E53F90901C416C1C75  
73A103504109A69A955B901C412032DD7CB70350418B00E19E04911C419FDD48A3B90350419  
B060D012B911C4166D1451AAB035041FC5482022B911C41178E9219AB03504173B28BD23E91  
1C41AA125FB4A30350418BC3E53F90901C416C1C7573A1035041"
```

Γι αυτό το λόγο, το Open GIS Consortium προτείνει τη χρήση του μορφότυπου WELL-KNOWN TEXT (WKT) ο οποίος υποστηρίζεται από το PostGIS. Με το WKT μπορούν αναπαρασταθούν με έναν εύκολα αντιληπτό τρόπο, σημεία, γραμμές πολύγωνα καθώς και πολλαπλές γεωμετρίες (π.χ. πολυ-πολύγωνα) και συλλογές γεωμετριών. Το PostGIS παρέχει συναρτήσεις που μετατρέπουν το WKT (που μπορεί να διαβαστεί και να γίνει κατανοητό σχετικά εύκολα) σε γεωμετρίες τις παραπάνω μορφής (που δε μπορούν να είναι απευθείας κατανοητές από τους ανθρώπους) και αντίστροφα. Μάλιστα, γεωμετρίες που έχουν συναχθεί σύμφωνα με το WKT format μπορούν να δημιουργηθούν / αναπαρασταθούν σε οποιοδήποτε σύστημα είναι συμβατό με τις κατευθύνσεις που δίνονται από το OGC (π.χ. Oracle, DB2 κ.α.). Παρακάτω παρατίθενται μερικά παραδείγματα του WKT format:

- Σημείο: **'POINT(X Y)'**
- Γραμμή: **'LINESTRING(X1 Y1, X2 Y2, ...)'**
- Πολύγωνο: **'POLYGON((X1 Y1, X2 Y2, ..., X1 Y1), (Xn,Yn, Xn+1 Yn+1, ..., Xn Yn))'**

όπου Χ₁ και Υ₁ οι συντεταγμένες του κάθε σημείου του αντικειμένου. Παρατηρείστε ότι το κάθε πολύγωνο μπορεί να περιγράφεται από πολλές σειρές συντεταγμένων μέσα σε παρενθέσεις (X₁ Y₁, X₂ Y₂, ..., X₁ Y₁) που αρχίζουν και τελειώνουν στο ίδιο σημείο. Από αυτές, η πρώτη σειρά είναι το εξωτερικό όριο του πολυγώνου, ενώ οι υπόλοιπες είναι πιθανά εσωτερικά όρια του πολυγώνου (νησίδες). Προφανώς ένα πολύγωνο μπορεί να μην έχει και καμία νησίδα.

Το πολύγωνο επίσης μπορεί να μην είναι απλό (δηλαδή να είναι αυτοτεμνόμενο - selfintersected). Π.χ. παρατηρήστε ότι μία αλλαγή στη σειρά των τεσσάρων σημείων στο παρακάτω σχήμα οδηγεί σε δύο εντελώς διαφορετικά πολύγωνα.

Ενδεικτικά αναφέρονται τα παρακάτω συστήματα συντεταγμένων μαζί με τα SRID τους:

	Σύστημα Συντεταγμένων	Τύπος	Μονάδες μέτρησης	SRID
1	Ελληνικό Γεωδαιτικό Σύστημα Αναφοράς 1987 (ΕΓΣΑ87)	Προβολικό	Μέτρα	2100
2	UTM Ζώνη 34 επί του WGS84	Προβολικό	Μέτρα	32634
3	UTM Ζώνη 35 επί του WGS84	Προβολικό	Μέτρα	32635
4	WGS84	Γεωγραφικό	Μοίρες	4326
5	Mercator Spheric	Γεωγραφικό	Μέτρα	900913

Από τα παραπάνω συστήματα αναφοράς το (1) είναι το πλέον διαδεδομένο προβολικό σύστημα στον Ελλαδικό Χώρο (και το επίσημο κρατικό σύστημα αναφοράς), το WGS84 είναι το παγκόσμιο σύστημα που χρησιμοποιεί το ελλειψοειδές WGS84 (και χρησιμοποιείται από τα GPS) και σε αυτό αναφέρονται συνήθως οι όροι «γεωγραφικό μήκος» και «γεωγραφικό πλάτος», τα (2) και (3) είναι Εγκάρσιες Μερκατορικές Προβολές επί του WGS84, ενώ το Mercator Spheric είναι μία προβολή επί μίας σφαίρας με περίπου τα ίδια χαρακτηριστικά με το WGS84 (χρησιμοποιεί μία σφαίρα αντί για ελλειψοειδές). Το τελευταίο είναι το σύστημα προβολής που χρησιμοποιεί ο μηχανισμός των tiles για τα Google Maps / Bing Maps.

Quantum GIS (QGIS: <http://qgis.org/>): Είναι μία εφαρμογή ανοικτού κώδικα που χρησιμοποιείται για τη δημιουργία, τροποποίηση, ανάλυση και παρουσίαση των γεωγραφικών δεδομένων. Στο περιβάλλον λειτουργίας του QGIS υπάρχει μια περιοχή εμφάνισης των χωρικών δεδομένων (map view), και μια περιοχή διαχείρισης του υπομνήματος (legend) στην οποία παρατίθενται τα ονόματα και τα σύμβολα των δεδομένων, περίπου με τον ίδιο τρόπο που αυτά παρουσιάζονται στο ArcGIS. Επίσης περιέχονται διάφορα εργαλεία και μενού επιλογών. Στο QGIS κάθε εργασία αποθηκεύεται σε ένα αρχείο το οποίο καλείται qgis project (.qgs). Διπλό κλικ σε ένα αρχείο .qgs προκαλεί αυτόματη εκκίνηση του QGIS. Το βασικό προτέρημα του QGIS είναι ότι μπορεί να εισάγει, να οπτικοποιήσει, διαχειριστεί, τροποποιήσει – δημιουργήσει δεδομένα που περιέχονται σε μία βάση δεδομένων PostgreSQL, σύμφωνα με το πρότυπο της επέκτασης PostGIS. Το QGIS δεν υποστηρίζει τα χωρικά δεδομένα που μπορεί να υπάρχουν στη PostgreSQL χωρίς το PostGIS

Τα δεδομένα για την εκτέλεση της άσκησης μπορούν να βρεθούν στο παρακάτω URL:
http://isl.cs.unipi.gr/courses/gis/lab/data/postgis_sample_data.zip

2. Δημιουργία πινάκων – ευρετηρίων

Η διαχείριση της PostgreSQL γίνεται μέσω τριών τρόπων:

- PgAdminIII (γραφικό περιβάλλον)
- Command prompt
- SQL Shell (κονσόλα psql to postgres)

Στην πορεία του εργαστηρίου θα χρησιμοποιήσουμε και τους τρεις, όπου απαιτείται.

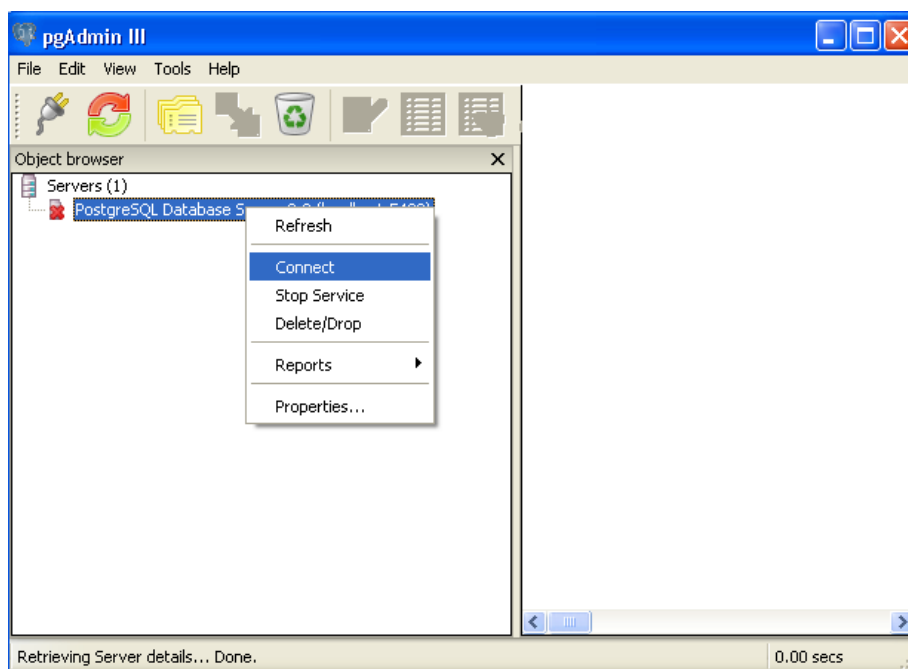
Βήμα 1ο: Θα ξεκινήσουμε από το SQL Shell για να δημιουργήσουμε μία καινούργια βάση δεδομένων. Επιλέξτε **Start > Programs > PostgreSQL 8.4 > SQL shell (psql)** και πληκτρολογήστε

```
SET ROLE postgres;
```

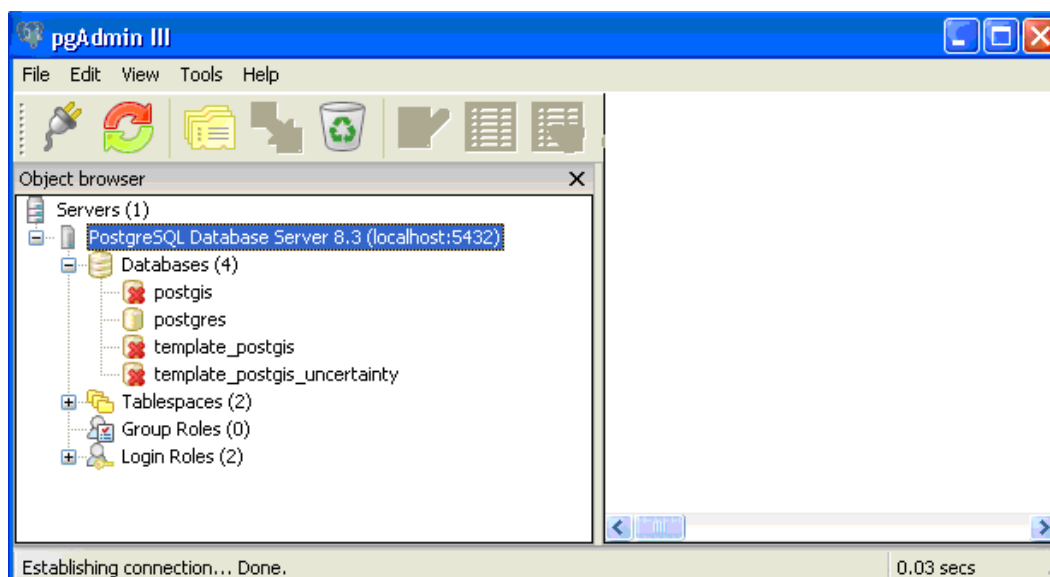
```
CREATE DATABASE lab_spatial_db  
TEMPLATE=template_postgis  
ENCODING='UTF8'  
TABLESPACE=pg_default;
```

Σημείωση: κάθε SQL εντολή στην κονσόλα SQL Shell πρέπει να τελειώνει με το σύμβολο ; (semicolon)

Εναλλακτικά μπορείτε να δημιουργήσετε τη βάση μέσω του PgAdmin. Από το **Start -> Programs -> PostgreSQL** ανοίξετε το πρόγραμμα PgAdmin III. Στη δένδρική δομή που εμφανίζεται στην αριστερή πλευρά του παραθύρου κάντε κλικ επάνω στο PostgreSQL Database Server και στο μενού που θα εμφανιστεί πατήστε το Connect όπως παρακάτω. Στο παράθυρο που ζητάει password πληκτρολογήστε postgres και πατήστε το OK.



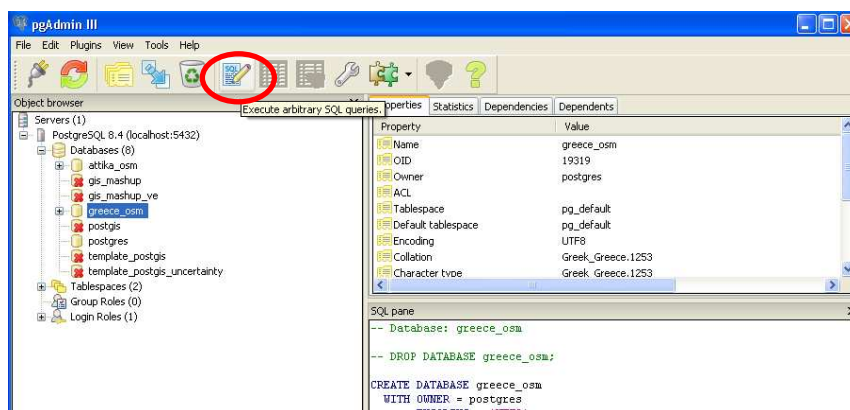
Μελετήστε το περιβάλλον εργασίας. Στα αριστερά του παραθύρου εμφανίζεται μία δένδρική μορφή με τις βάσεις δεδομένων (Databases) που αυτή τη στιγμή διαχειρίζονται από τη PostgreSQL. Το σύμβολο  postgis σημαίνει ότι η βάση δεδομένων με όνομα postgis βρίσκεται στο δένδρο της PostgreSQL και είναι κλειστή, ενώ το σύμβολο  postgres σημαίνει ότι είναι ανοιχτή.



Για τη δημιουργία της βάσης δεδομένων που θα φιλοξενήσει τα δεδομένα μας, κάντε κλικ για στον κόμβο  Databases (4) για να τον ενεργοποιήσετε (γίνεται μπλε), και στη συνέχεια δεξί κλικ και επιλέξτε το “New Database”. Η νέα βάση που θα φτιάξουμε θα έχει όνομα lab_spatial_db, θα βασίζεται στο πρότυπο template_postgis και θα ακολουθεί την κωδικοποίηση UTF-8.

Κάθε νέα χωρική βάση δεδομένων που δημιουργείται πρέπει να βασίζεται στο πρότυπο template_postgis, αλλιώς, δε θα λειτουργούν οι συναρτήσεις του PostGIS.

Βήμα 2°: Από το PgAdmin, συνδεθείτε στη βάση δεδομένων lab_spatial_db κάνοντας κλικ επάνω της, και επιλέξτε το κουμπί Execute arbitrary SQL Queries, το οποίο χρησιμοποιείται για τη σύνταξη και εκτέλεση sql ερωτημάτων



Όσα ερωτήματα εκτελεστούν παρακάτω μπορούν να εκτελεστούν μέσα από το περιβάλλον αυτού του Query Editor.

Βήμα 3°: Δημιουργία σχήματος με την ονομασία lab. Πληκτρολογήστε το παρακάτω

```
CREATE SCHEMA lab;
```

και πατήστε το πλήκτρο F5 (ή επιλέξτε το Execute από το μενού Query).

Βήμα 4°: Δημιουργία πίνακα σημείων ενδιαφέροντος με την ονομασία landmarks:

```
CREATE TABLE lab.Landmarks(  
  landmark_id integer primary key,  
  landmark_name varchar(50),  
  landmark_geometry geometry);
```

Προσέξτε το geometry column το οποίο θα περιέχει όλη τη γεωμετρία του κάθε αντικείμενου. Ο τύπος geometry είναι γενικός και μπορεί να περιέχει χωρικά αντικείμενα των τριών τύπων (σημείο, γραμμή, πολύγωνο), καθώς και συλλογές αυτών των αντικειμένων. Αν δεν χρησιμοποιούσαμε το PostGIS αλλά απλή PostgreSQL, θα έπρεπε να ορίσουμε κάποιον από τους τύπους point, path, polygon

Βήμα 5°: Δημιουργία πίνακα δρόμων με την ονομασία roads:

```
CREATE TABLE lab.roads(  
  road_id integer primary key,  
  road_name varchar(50),  
  road_geometry geometry);
```

Βήμα 6ο: Δημιουργία πίνακα γεωτεμαχίων με την ονομασία parcels:

```
CREATE TABLE lab.parcels(  
  parcel_id integer primary key,  
  parcel_name varchar(50),  
  parcel_geometry geometry);
```

Βήμα 7ο: Δημιουργία πίνακα ζωνών με την ονομασία zones:

```
CREATE TABLE lab.zones(  
  zone_id integer primary key,  
  zone_name varchar(50),  
  zone_geometry geometry);
```

Βήμα 8ο: Δημιουργία πίνακα χωρικών αντικειμένων με χρήση των προτύπων του Open GIS Consortium. Πρώτα δημιουργήστε έναν πίνακα χωρίς χωρικά αντικείμενα:

```
CREATE TABLE lab.dummy(  
  dummy_id integer primary key,  
  dummy_name varchar(50));
```

Στη συνέχεια θα χρησιμοποιήσουμε την AddGeometryColumn με Σύνταξη

```
AddGeometryColumn(<schema_name>,  
                  <table_name>,  
                  <column_name>,  
                  <srid>, <type>,  
                  <dimension>)
```

Το SRID είναι το σύστημα προβολής των δεδομένων. Το Ελληνικό σύστημα προβολής (ΕΓΣΑ'87) έχει SRID = 2100. Επομένως στη κονσόλα πληκτρολογήστε:

```
SELECT AddGeometryColumn('lab', 'dummy', 'dummy_geometry',  
                        2100, 'GEOMETRY', 2);
```

Χρησιμοποιώντας την AddGeometryColumn μπορούμε να καθορίσουμε και άλλους τύπους εκτός από GEOMETRY (π.χ. LINESTRING). Αντίθετα, όταν δημιουργώντας γεωμετρίες στο create table δε μπορούμε να καθορίσουμε άλλον τύπο εκτός από το geometry και ούτε να ορίσουμε το σύστημα προβολής.

Σημείωση: το QGIS δεν υποστηρίζει άλλους τύπους εκτός από geometry. Δεν υποστηρίζει επίσης την προσθήκη SRID στο κάθε πεδίο (αντίθετα, περιμένει να το βρει σε κάθε αντικείμενο)

Βήμα 9ο: Δημιουργία χωρικών ευρετηρίων GiST επάνω στους πίνακες landmarks, roads, parcels, zones

```
CREATE INDEX landmarks_spatial_index  
ON lab.landmarks USING GIST(landmark_geometry);
```

```
CREATE INDEX roads_spatial_index  
ON lab.roads USING GIST(road_geometry);
```

```
CREATE INDEX parcels_spatial_index  
ON lab.parcels USING GIST(parcel_geometry);
```

```
CREATE INDEX zones_spatial_index  
ON lab.zones USING GIST(zone_geometry);
```

Παλαιότερες εκδόσεις του PostGIS χρησιμοποιούσαν τα R-trees της PostgreSQL. Παρόλαυτά, από την έκδοση 0.6 και μετά τα R-trees έχουν αφαιρεθεί και η δεικτοδότηση των χωρικών δεδομένων του PostGIS γίνεται μέσω ενός σχήματος R-tree πάνω από GiST. Η τεκμηρίωση του PostGIS αναφέρει ότι η απόδοση του σχήματος αυτού σε σχετικά με τα R-trees της PostgreSQL είναι συγκρίσιμη. Επιπλέον, τα R-trees όπως είχαν υλοποιηθεί στη PostgreSQL είχαν τον περιορισμό ότι μπορούν να δεικτοδοτήσουν μόνο παραλληλόγραμμα (bounding boxes).

3. Εισαγωγή δεδομένων

Βήμα 10: Εισαγωγή δεδομένων στον πίνακα σημείων ενδιαφέροντος με διαδοχικές εντολές insert:

```
INSERT INTO lab.landmarks
VALUES(1, 'Landmark1', GeomFromText('POINT(468346 4198876)', 2100));
INSERT INTO lab.landmarks
VALUES(2, 'Landmark2', GeomFromText('POINT(468518 4198644)', 2100));
```

Βήμα 11: Εισαγωγή δεδομένων στον πίνακα δρόμων με διαδοχικές εντολές insert:

```
INSERT INTO lab.roads
VALUES(1, 'Road1', GeomFromText('LINESTRING(469180 4199872, 469171
4199855)', 2100));

INSERT INTO lab.roads
VALUES(2, 'Road2', GeomFromText('LINESTRING(468994 4199855, 469092
4199882)', 2100));
```

Βήμα 12: Εισαγωγή δεδομένων στον πίνακα ζωνών με διαδοχικές εντολές insert:

```
INSERT INTO lab.zones VALUES(1, 'Zone1', GeomFromText('POLYGON((467646
4198420, 467734 4197970, 467962 4198009, 467882 4198464, 467646
4198420))', 2100));
INSERT INTO lab.zones VALUES(2, 'Zone2', GeomFromText('POLYGON((467882
4198464, 467962 4198009, 468248 4198055, 468224 4198156, 468264 4198163,
468233 4198347, 468199 4198522, 467882 4198464, 467882 4198464))', 2100));
```

Βήμα 13: Εισαγωγή δεδομένων στον πίνακα δρόμων και σημείων ενδιαφέροντος από ASCII αρχείο sql. Πρώτα θα διαγράψουμε τα περιεχόμενα των δύο πινάκων:

```
DELETE FROM lab.roads;
DELETE FROM lab.landmarks;
```

Ανοίξτε το αρχείο roads.sql με έναν editor (π.χ. notepad) και παρατηρείστε τα περιεχόμενά του. Τι μπορείτε να πείτε για τη μορφή του αρχείου;

```
BEGIN;
INSERT INTO lab.landmarks
VALUES(1, 'Landmark1', GeomFromText('POINT(462573 4199287)', 2100));
INSERT INTO lab.landmarks
VALUES(2, 'Landmark2', GeomFromText('POINT(462740 4199054)', 2100));
* * *
COMMIT;
```

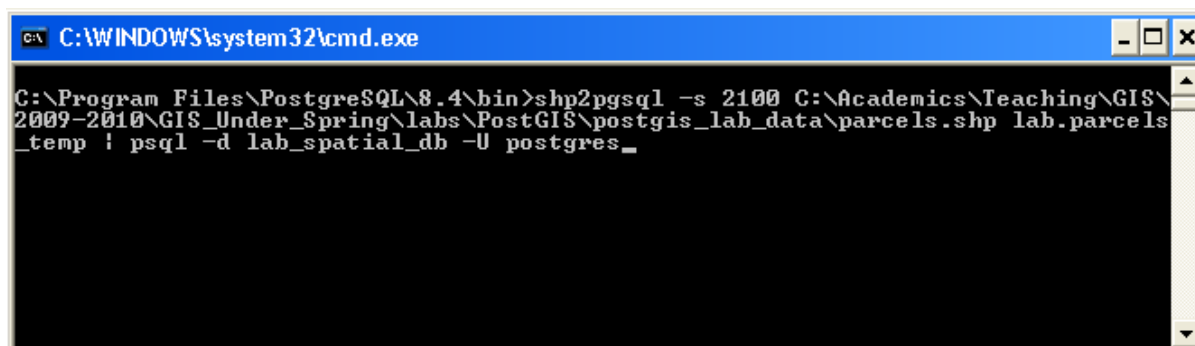
Για να εισάγουμε τα δεδομένα στη PostgreSQL, αρκεί να ανοίξουμε το αντίστοιχο sql αρχείο με τον sql editor και να το τρέξουμε κατά τα γνωστά.

Βήμα 14°: Εισαγωγή δεδομένων από shape file (αρχείο του ArcGIS). Θα χρησιμοποιήσουμε την παρακάτω εντολή που πρέπει να εκτελεστεί από το **command prompt**.

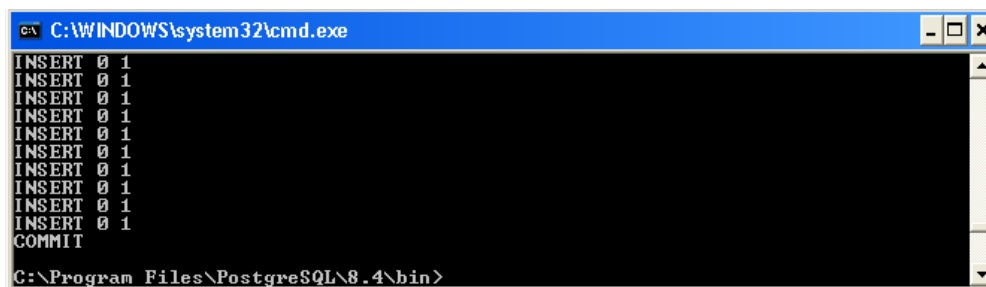
```
shp2pgsql -s 2100 [shapefile] [schema].[table] | psql -d [database] -U [user]
```

Εκτελούμε το command prompt των windows (από το Start -> Run γράφουμε cmd) και πλοηγούμαστε στο directory. C:\Program Files\PostgreSQL\8.4\bin, όπου εκτελούμε την εντολή

```
shp2pgsql -s 2100 P:\postgis_lab_data\parcels.dbf lab.parcels_temp |  
psql -d lab_spatial_db -U postgres
```



και πατήστε το Enter. Το σύμβολο | εμφανίζεται πατώντας Shift+|. Εφόσον σας ζητηθεί password πληκτρολογήστε postgres και αφήστε την εντολή να εκτελεστεί. Στο τέλος της εκτέλεσης της εντολής, η οθόνη που θα πάρετε θα πρέπει να είναι όπως παρακάτω:



Παρατηρήστε ότι η εισαγωγή των δεδομένων γίνεται σε έναν πίνακα που ονομάζεται parcels_temp.

Βήμα 15°: Μεταφορά δεδομένων στον πίνακα γεωτεμαχίων και διαγραφή του προσωρινό πίνακα:

```
INSERT INTO lab.parcels (parcel_id, parcel_geometry)  
SELECT gid, the_geom FROM lab.parcels_temp;  
  
DROP TABLE lab.parcels_temp;
```

4. Εμφάνιση των δεδομένων

Βήμα 16°: Εμφάνιση των δεδομένων σε μορφή πινάκων

```
SELECT * FROM lab.landmarks;
```

Μετατροπή κατά την εμφάνιση σε WKT format

```
SELECT landmark_id, landmark_name, AsText(landmark_geometry)
FROM lab.landmarks;
```

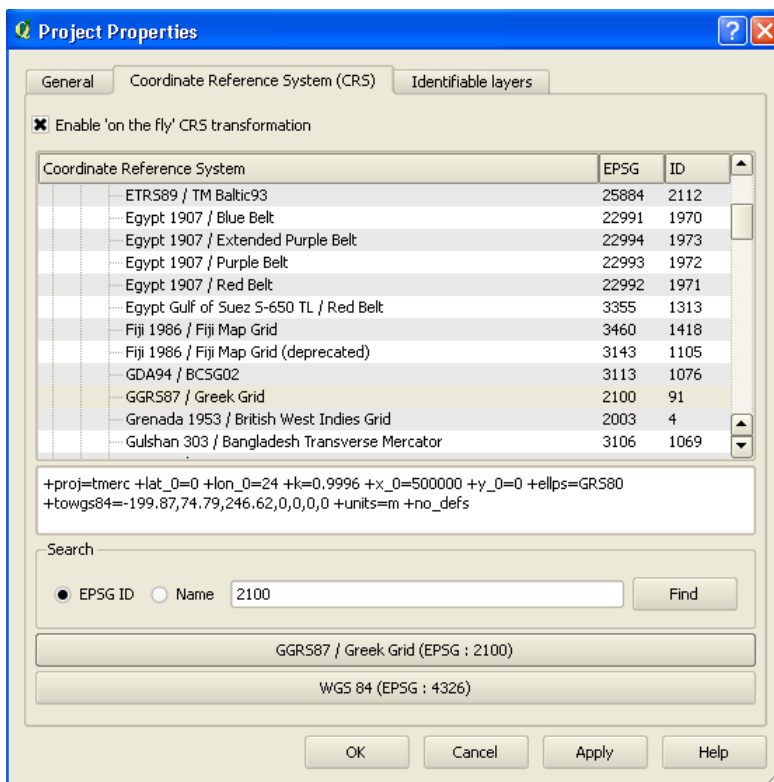
Βήμα 17^ο: Οπτικοποίηση των δεδομένων

Εκκινήστε την εφαρμογή Quantum GIS. Επιλέξτε *Start > Programs > Quantum GIS Enceladus > Quantum GIS Enceladus*.

Θα πρέπει καταρχήν να καθορίσουμε το σύστημα συντεταγμένων του project του Quantum GIS. Αυτό γίνεται από το μενού

Settings > Project Properties

Και στο ενεργοποιούμε το “Enable on the fly CRS transformation”, ενώ αναζητούμε το ελληνικό σύστημα προβολής GGRS87 (ΕΓΣΑ87), με EPSG ID 2100 (το οποίο EPSG ID είναι ίδιο με το SRID του PostGIS). Αφού επιλέξουμε το σύστημα όπως στην παρακάτω εικόνα, πατάμε το OK.



Για να προσθέσουμε τα δεδομένα, που έχουμε αποθηκεύσει στο PostGIS, εκτελέστε την εντολή

Layer > Add a PostGIS Layer > New Connection

Δώστε τις παρακάτω τιμές στα ζητούμενα πεδία:

Name: *lab_spatial_db*

host: *localhost*

database: *lab_spatial_db*

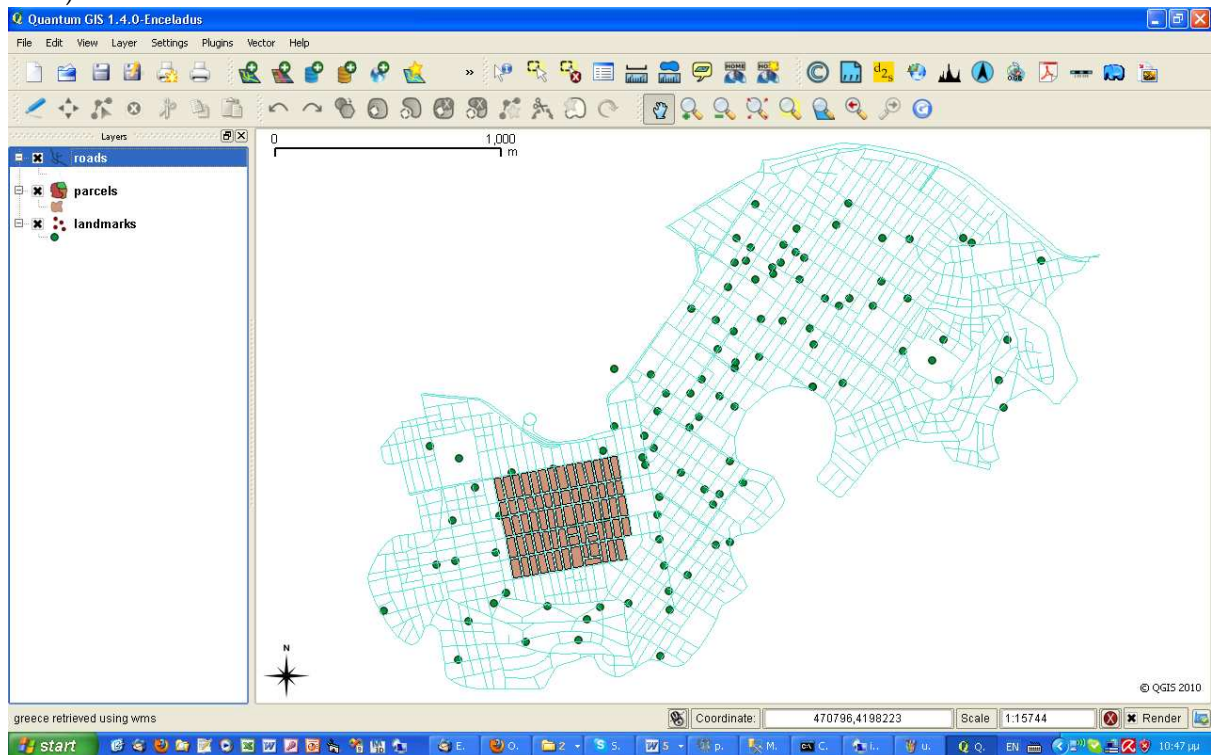
port: 5432 (ή 5433, ανάλογα με την εγκατάσταση)

username: *postgres*

password: το password που έχετε ορίσει στην εγκατάστασή σας

Ενεργοποιείτε το Save Password

και πατήστε το **Connect**. Στη συνέχεια, επιλέξτε και τα 5 Layer που εμφανίζονται και πατήστε το **Add**. Παρατηρείστε ότι το επίπεδο dummy δεν μπορεί να εισαχθεί (γιατί;) Πατήστε το OK. Ο χάρτης της περιοχής πρέπει να έχει εμφανιστεί στο κυρίως παράθυρο (map view)



Βήμα 18: Προσθέστε μία υπηρεσία Web Mapping Service εκτελώντας την εντολή

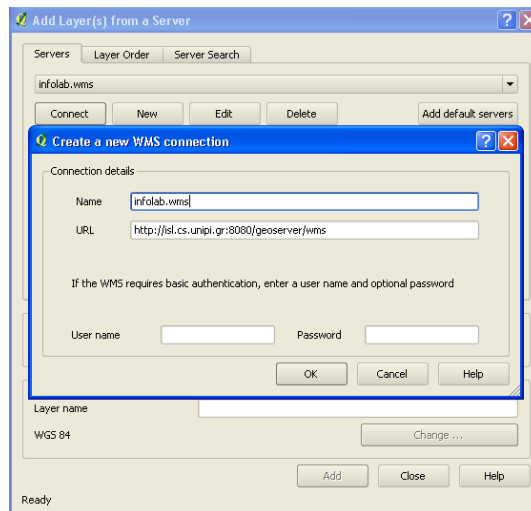
Layer > Add a WMS Layer > New Connection

Πατήστε το κουμπί New για να δημιουργήσουμε μία νέα σύνδεση, και δώστε τα παρακάτω στοιχεία:

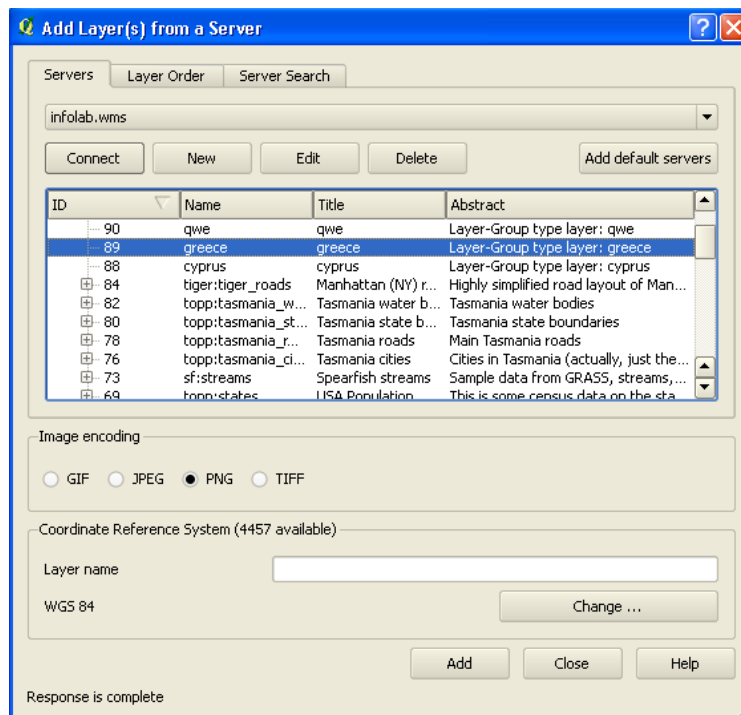
Name: *infolab.wms*

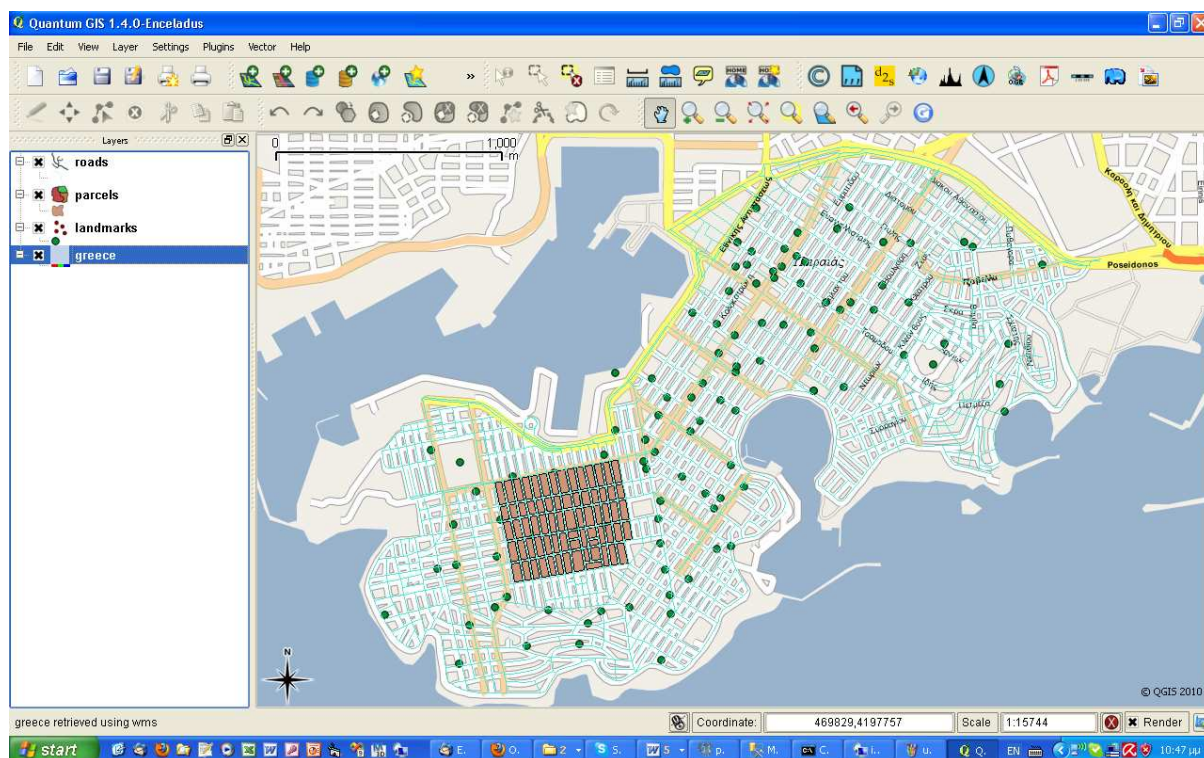
URL: *http://isl.cs.unipi.gr:8080/geoserver/wms*

όπως στην παρακάτω εικόνα, και πατήστε το ok.



Στη συνέχεια επιλέξτε την υπηρεσία στην οποία συνδεθήκαμε, και πατήστε το connect, ώστε να εμφανιστούν όλοι οι χάρτες που διαθέτει η υπηρεσία. Επιλέξτε το greece όπως στην παρακάτω εικόνα και πατήστε το Add





Εξοικειωθείτε με:

- ✓ ενεργοποιήστε / απενεργοποιήστε τα θεματικά επίπεδα (layers)
- ✓ αλλάξτε τη σειρά εμφάνισης των θεματικών επιπέδων
- ✓ αναζητήστε τις γεωγραφικές συντεταγμένες των σημείων ενδιαφέροντος
- ✓ ζητήστε πληροφορίες για τα αντικείμενα
- ✓ αλλάξτε τα σύμβολα των οντοτήτων (χρώματα, τύπους γραμμών, κλπ.)
- ✓ αλλάξτε το zoom
- ✓ ανοίξτε το attribute table χρησιμοποιώντας το κουμπί 
- ✓ Επιλέξτε αντικείμενα στο σχέδιο με το  και παρατηρήστε τι συμβαίνει στο attribute table.
- ✓ Δώστε κάποια κριτήρια για επιλογή δεδομένων στο επίπεδο roads πατώντας το κουμπί **advanced** στο attribute table

5. Εκτέλεση ερωτήσεων

Οι ερωτήσεις θα εκτελεστούν μέσω της κονσόλας εντολών ή στο PgAdminIII. Η οπτικοποίηση των αποτελεσμάτων όπου απαιτείται θα γίνει μέσω του QGIS

Βήμα 19°: Πραγματοποιήστε μία ερώτηση επιλογής με βάση ένα σημείο:

```
SELECT parcel_id  
FROM lab.parcels  
WHERE parcel_geometry && GeomFromText('POINT(468010 4198347)', 2100);
```

Ο τελεστής && ελέγχει αν τέμνονται τα bounding boxes των χωρικών αντικειμένων
Ανοίξτε το attribute table του θεματικού επιπέδου parcels στο QGIS και εντοπίστε το σημείο και το parcel που επέστρεψε σαν απάντηση το query.

Βήμα 20°: Πραγματοποιήστε μία ερώτηση επιλογής με βάση την απόσταση από σημείο:

```
SELECT *
```

```
FROM lab.roads
WHERE road_geometry && Expand(GeomFromText('POINT(468010 4198347)',
2100),100) AND Distance(GeomFromText('POINT(468010 4198347)',
2100),road_geometry)< 100;
```

Δοκιμάστε επίσης την παρακάτω ερώτηση

```
SELECT *
FROM lab.roads
WHERE Distance(GeomFromText('POINT(468010 4198347)',2100),road_geometry)
< 100;
```

Τι παρατηρείτε σε σχέση με τα αποτελέσματα των δύο ερωτήσεων;

Η πιο αποδοτική χρήση της χωρικής βάσης δεδομένων, γίνεται συνδυάζοντας ερωτήσεις με βάση την απόσταση και παράλληλο έλεγχο επικάλυψης των περιγεγραμμένων παραλληλογράμμων (bounding boxes). Ο έλεγχος με το περιγεγραμμένο παραλληλόγραμμο είναι πολύ πιο γρήγορος από τον αναλυτικό έλεγχο με βάση την απόσταση (λόγω του ότι χρησιμοποιείται το χωρικό ευρετήριο). Η συνάρτηση Expand() παρέχει έναν τρόπο ώστε να μεγαλώσει το περιγεγραμμένο παραλληλόγραμμο ενός αντικειμένου.

Για να δείτε πως λειτουργεί η Expand() εκτελέστε την παρακάτω εντολή

```
CREATE VIEW lab.expand_test AS
SELECT landmark_id, Expand(landmark_geometry, 20) AS expanded_geometry
FROM lab.landmarks;
```

και από το QGIS επιλέξτε

Layer > Add a PostGIS Layer > New Connection

επιλέξτε τη σύνδεση στη PostgreSQL, πατήστε το connect και επιλέξτε το expand_test. Παρατηρείστε τη μορφή των δεδομένων. Προσέξτε επίσης ότι το View που δημιουργήσαμε έχει και ένα πεδίο που προέρχεται από πρωτεύον κλειδί (και επιπλέον συνεχίζει να είναι μοναδικό μέσα στο View). Αυτός είναι περιορισμός του Qgis (πρέπει να υπάρχει ένα πεδίο που να μπορεί να χρησιμοποιηθεί σαν μοναδικό κλειδί).

Βήμα 21ο: Πραγματοποιήστε μία ερώτηση επιλογής με μία περιοχή:

Με περιοχή παραλληλόγραμμο:

```
SELECT *
FROM lab.landmarks
WHERE box('467350,4198000,467686,4198277') && landmark_geometry;
```

ή

```
SELECT *
FROM lab.landmarks
WHERE Contains(GeomFromText('POLYGON((467350 4198000, 467350 4198277,
467686 4198277, 467686 4198000))',
2100),landmark_geometry);
```

και οι δύο παραπάνω ερωτήσεις επιστρέφουν το ίδιο αποτέλεσμα. Η πρώτη είναι πιο γρήγορη (ελέγχει μόνο τα bounding boxes με τον τελεστή &&), αλλά και λιγότερο ακριβής. Επιπλέον, χρησιμοποιεί τη λειτουργικότητα της PostgreSQL (και όχι PostGIS) επειδή περιέχει τη συνάρτηση box().

Με περιοχή πολύγωνο (τρίγωνο):

```
SELECT *  
FROM lab.landmarks  
WHERE landmark_geometry @ GeomFromText('POLYGON((467350 4198000, 467350  
4198277, 467686 4198277, 467350 4198000))', 2100);
```

και

```
SELECT *  
FROM lab.landmarks  
WHERE Contains(GeomFromText('POLYGON((467350 4198000, 467350 4198277,  
467686 4198277, 467350 4198000))', 2100), landmark_geometry);
```

Παρατηρείστε ότι οι δύο ερωτήσεις δεν έχουν το ίδιο αποτέλεσμα. Γιατί συμβαίνει αυτό;

Βήμα 22: Πραγματοποιήστε μία ερώτηση χωρικής σύνδεσης (σημείο – πολύγωνο):

```
SELECT landmark_id, landmark_geometry  
FROM lab.landmarks, lab.parcels  
WHERE landmark_geometry @ parcel_geometry;
```

προσέξτε ότι στα αποτελέσματα εμφανίζονται δύο εγγραφές με landmark_id=9. Αυτό συμβαίνει γιατί ο τελεστής @ χρησιμοποιεί τα bounding boxes και το αποτέλεσμα είναι προσεγγιστικό. Αντίθετα, η παρακάτω ερώτηση επιστρέφει ακριβή αποτελέσματα:

```
SELECT landmark_id, landmark_geometry  
FROM lab.landmarks, lab.parcels  
WHERE Contains(parcel_geometry, landmark_geometry);
```

Βήμα 23: Πραγματοποιήστε μία ερώτηση χωρικής σύνδεσης (πολύγωνο - πολύγωνο):

```
SELECT parcel_id, parcel_geometry  
FROM lab.parcels, lab.zones  
WHERE Contains(zone_geometry, parcel_geometry) AND zone_name='Zone1'
```

Οπτικοποιήστε το αποτέλεσμα της ερώτησης δημιουργώντας ένα View (όπως στο βήμα 19).

Βήμα 24: Ερώτημα χωρικής αυτοσύνδεσης. "Επέλεξε όλα τα ζεύγη σημείων που απέχουν μεταξύ τους λιγότερο από 50 μέτρα"

```
SELECT L1.landmark_id, L1.landmark_geometry,  
       L2.landmark_id, L2.landmark_geometry,  
       distance(L1.landmark_geometry, L2.landmark_geometry)  
FROM lab.landmarks L1, lab.landmarks L2  
WHERE distance(L1.landmark_geometry, L2.landmark_geometry)<50  
       AND L1.landmark_id<>L2.landmark_id;
```

Βήμα 25: Χρήση συναρτήσεων:

Εμβαδόν περιοχής

```
SELECT parcel_id, parcel_name, area(parcel_geometry) FROM lab.parcels;
```

Μήκος αξόνων

```
SELECT road_id, road_name, length(road_geometry)
FROM lab.roads
WHERE length(road_geometry)>100;
```

Βήμα 26ο: Buffer χωρικών αντικειμένων

```
SELECT parcel_id, parcel_geometry, road_id
FROM lab.parcels, lab.roads
WHERE buffer(road_geometry, 6) && parcel_geometry;
```

αλλά και

```
SELECT parcel_id, parcel_geometry, road_id
FROM lab.parcels, lab.roads
WHERE intersects(buffer(road_geometry, 6), parcel_geometry);
```

Παρατηρήστε ότι ο χρόνος εκτέλεσης στη δεύτερη περίπτωση αυξάνεται δραματικά. Τι μπορούμε να κάνουμε ώστε να το βελτιώσουμε;

Ενεργοποιήστε το QGIS και βρείτε ποιοι δρόμοι έχουν id μεταξύ 1200 και 1300. Στη συνέχεια οπτικοποιήστε τα αποτελέσματα του παρακάτω View:

```
CREATE VIEW lab.buffer_selection AS
SELECT DISTINCT parcel_id, parcel_geometry
FROM lab.parcels, lab.roads
WHERE intersects(buffer(parcel_geometry, 6), road_geometry)
AND road_id>=1200 AND road_id<1300;
```

και εισάγετε το Layer στο QGIS. Προσέξτε ότι έτσι μπορείτε να βρείτε ποιοι δρόμοι περνάνε μπροστά από ποια γεωτεμάχια.

Βήμα 27ο: Συναθροίσεις:

```
SELECT ConvexHull(collect(landmark_geometry))
FROM lab.landmarks;
```

και

```
SELECT ConvexHull(collect(landmark_geometry))
FROM lab.landmarks
GROUP BY (landmark_id/10)::int;
```

οπτικοποίηση δε μπορούμε να κάνουμε λόγω των περιορισμών του Qgis στο πεδίο id. Μπορούμε όμως να δημιουργήσουμε έναν νέο πίνακα και να δείξουμε έτσι το αποτέλεσμα.

Τέλος
