

# POSTGRESQL

Η PostgreSQL είναι ένα αντικείμενο-σχεσιακό σύστημα διαχείρισης βάσεων δεδομένων (object relational database management system - ORDBMS) και έχει τα παρακάτω χαρακτηριστικά :

- Λογισμικό ανοικτού κώδικα.
- Τρέχει σε όλα τα βασικά λειτουργικά συστήματα (unix/linux windows macos).
- Προσφέρει
  - Ειδικούς τύπους δεδομένων για την αποθήκευση απλών γεωμετρικών οντοτήτων,
  - Μηχανισμούς δεικτοδότησης χωρικών δεδομένων όπως τετραδικά και R-δένδρα.
  - Περιορισμένο αριθμό γεωμετρικών τελεστών και συναρτήσεων για την διατύπωση χωρικών ή συνδυασμένων ερωτημάτων.
- Η PostgreSQL επιτρέπει τη συνεργασία με πληθώρα επιπρόσθετων σπονδύλων λογισμικού (addons) μέσω των οποίων γίνεται η εκτέλεση συγκεκριμένων και εξειδικευμένων λειτουργιών.
- Η επέκταση χωρικών λειτουργιών επιτυγχάνεται σε συνεργασία με το λογισμικό ανοικτού κώδικα **PostGIS** το οποίο ακολουθεί το πρότυπο OGC για τον ορισμό γεωγραφικών στοιχείων σε περιβάλλον SQL.
- Η PostgreSQL με την επέκταση λειτουργιών PostGIS αποτελεί ένα σύστημα κατάλληλο για ανάπτυξη και διαχείριση χωρικών βάσεων δεδομένων.

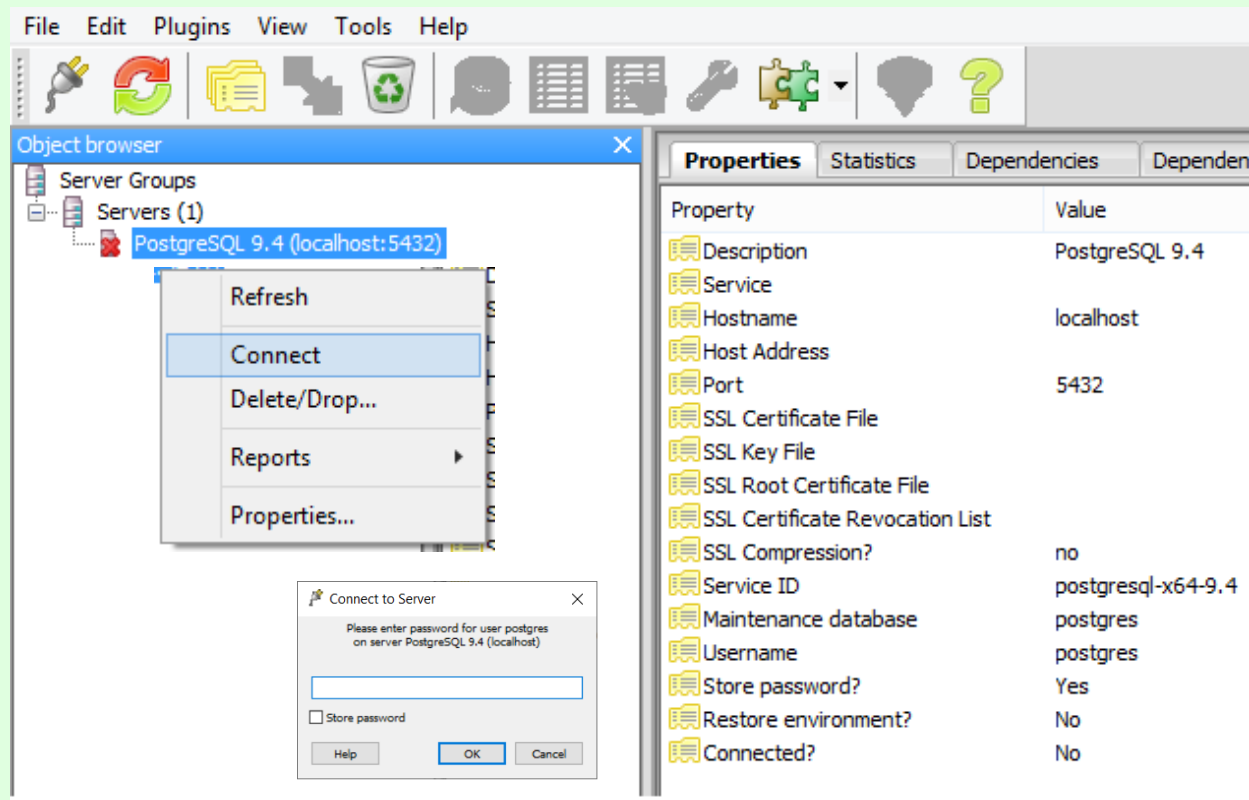


Η PostgreSQL δεν παρέχει ή ίδια κάποιο γραφικό περιβάλλον διεπαφής με τον χρήστη. Για την αλληλεπίδραση με την βάση δεδομένων χρησιμοποιείται ένα πρόγραμμα client το pgAdminn III . Παρέχεται και εγκαθίσταται μαζί με την PostgreSQL. Το pgAdmin III και έχει τα παρακάτω χαρακτηριστικά :

- Είναι ένα ολοκληρωμένο σύστημα σχεδιασμού και διαχείρισης βάσεων δεδομένων και αποτελεί το μέσο αλληλεπίδρασης της βάσης δεδομένων με το χρήστη.
- Είναι γραμμένο σε C++ και λειτουργεί σε περιβάλλοντα Linux, Mac και Windows.
- Προσφέρει ένα απλό γραφικό περιβάλλον για την ανάπτυξη πολύπλοκων βάσεων δεδομένων μέσω της διατύπωσης ερωτημάτων σε SQL, με στόχο την απλούστευση των διαδικασιών για το χρήστη.
- Είναι ελεύθερο λογισμικό και δεν απαιτεί επιπλέον προγράμματα για την επικοινωνία με τον διακομιστή της βάσης δεδομένων.







# Εκκίνηση

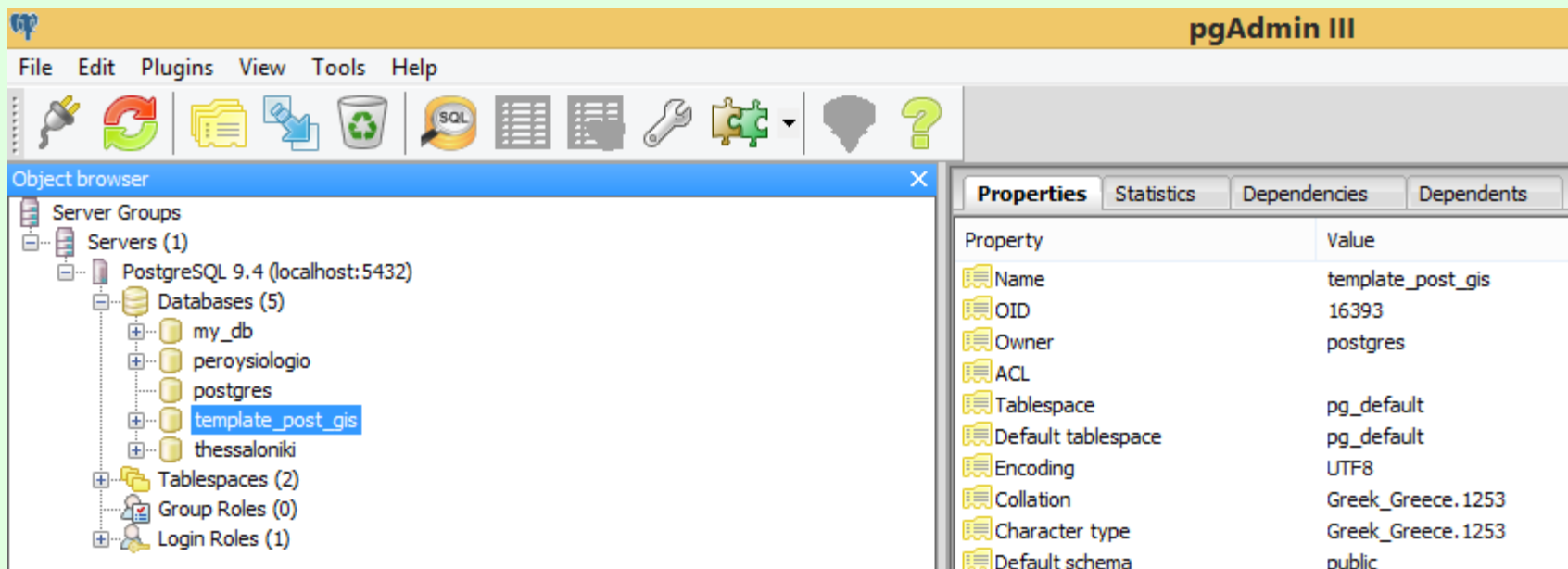
Με την εκκίνηση του client pgAdmin III εμφανίζεται το γραφικό περιβάλλον διεπαφής με την βάση postgresql. Με δεξί κλικ πάνω στο όνομα του SERVER (postgresql 9.4) εμφανίζεται μενού από μπορούμε να συνδεθούμε χρησιμοποιώντας τα συνθηματικά που έχουμε ορίσει κατά την εγκατάσταση.



# Κεντρική Σελίδα

Όταν συνδεθούμε στον server βλέπουμε ότι έχουν δημιουργηθεί 5 βάσεις δεδομένων. Επίσης εμφανίζεται το μενού και κάποια εργαλεία όπως στην εικόνα που ακολουθεί:


-  Σύνδεση σε απομακρυσμένο Server
-  Ενημερώνει το σύστημα για τις αλλαγές που έχουν γίνει στο επιλεγμένο αντικείμενο
-  Εμφανίζει τα χαρακτηριστικά του αντικειμένου που έχουμε επιλέξει
-  Για δημιουργία αντικειμένου ίδιας κατηγορίας με αυτό που έχουμε επιλέξει
-  Για διαγραφή επιλεγμένων αντικειμένων
-  Για ανάπτυξη και εκτέλεση ερωτημάτων

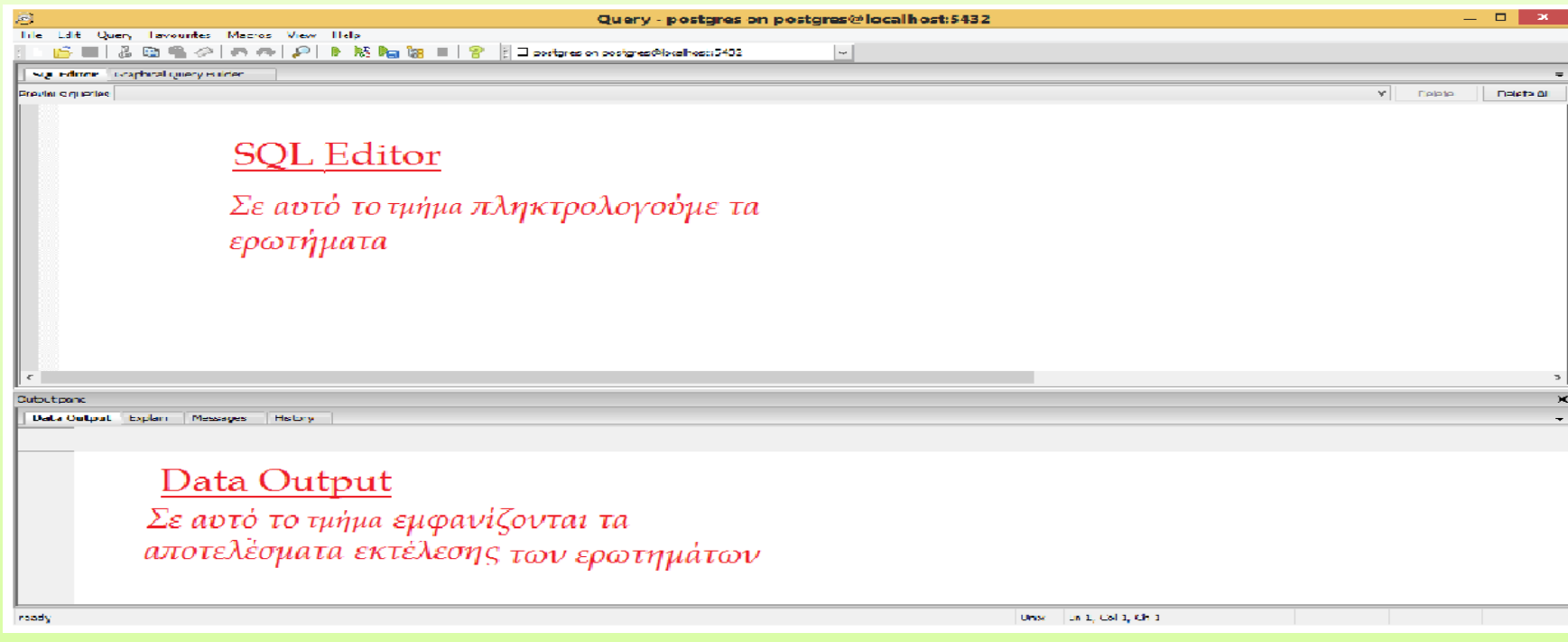


The screenshot shows the pgAdmin III interface. The top menu bar includes File, Edit, Plugins, View, Tools, and Help. Below the menu is a toolbar with icons for connecting, refreshing, viewing properties, copying, deleting, and executing SQL. The Object browser on the left shows a tree view of the database structure. The Properties window on the right displays the following details for the 'template\_post\_gis' database:

Property	Value
Name	template_post_gis
OID	16393
Owner	postgres
ACL	
Tablespace	pg_default
Default tablespace	pg_default
Encoding	UTF8
Collation	Greek_Greece.1253
Character type	Greek_Greece.1253
Default schema	public

# Περιβάλλον ανάπτυξης ερωτημάτων SQL

Η postgresql διαθέτει ένα περιβάλλον ανάπτυξης εκτέλεσης αποθήκευσης ερωτημάτων SQL. Το εν λόγω περιβάλλον αποτελείται από ένα παράθυρο χωρισμένο σε 2 τμήματα: Το τμήμα του **SQL editor** όπου πληκτρολογούμε τα ερωτήματα και το τμήμα του **Data Output** όπου εμφανίζονται τα αποτελέσματα των ερωτημάτων. Για να ανοίξουμε αυτό το παράθυρο πατάμε το εργαλείο  ή επιλέγουμε **Tools\Query tool**. Για να εκτελέσουμε ερωτήματα επιλέγουμε **Query\Execute**. Τα δεδομένα των ερωτημάτων προέρχονται από την βάση που έχουμε ανοίξει. Τα αποτελέσματα ερωτημάτων ενημέρωσης ενημερώνουν την εν λόγω βάση. Για να ανοίξουμε μία βάση δεδομένων Κάνουμε διπλό κλικ πάνω στο όνομα της.

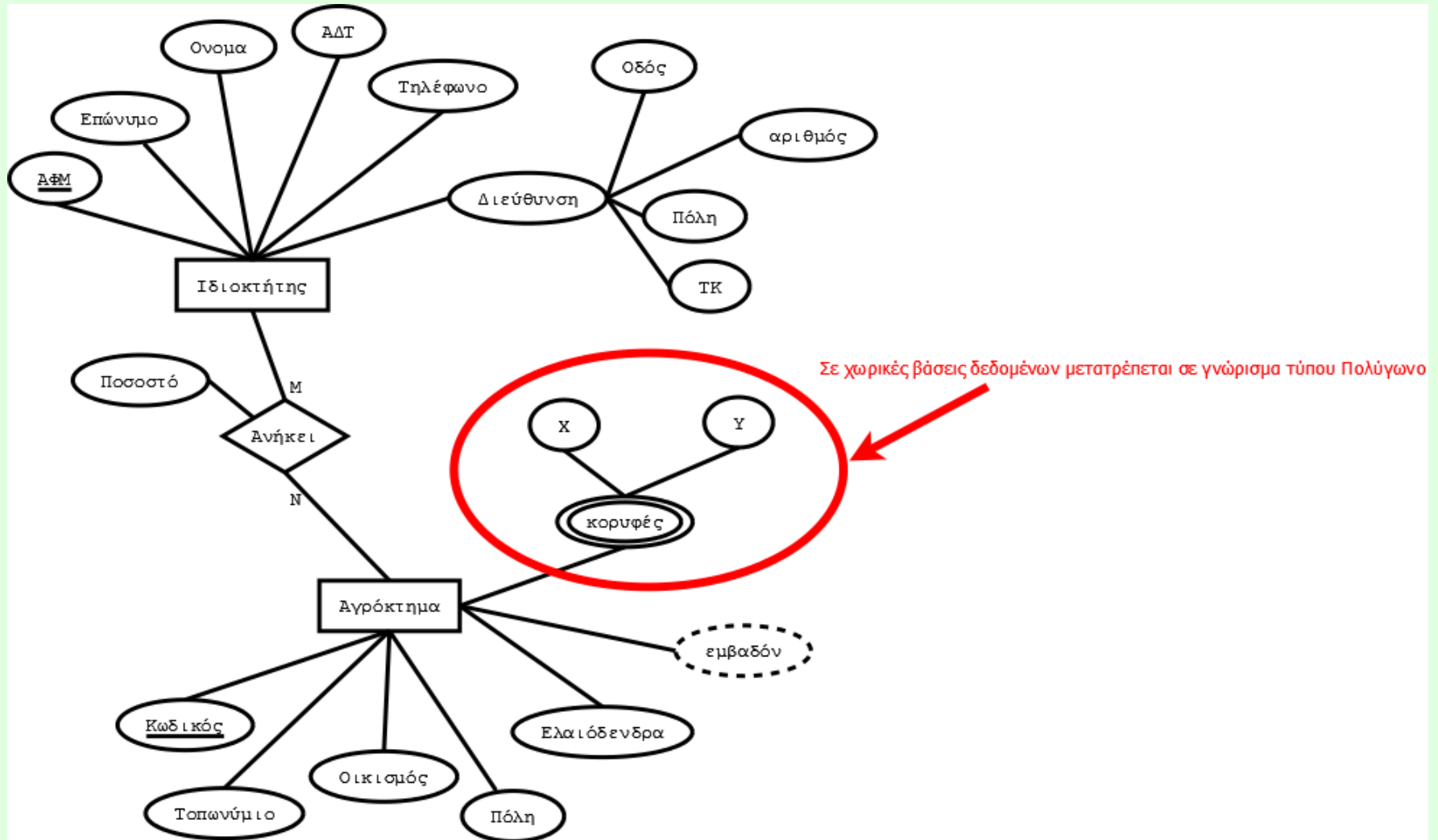


# ΧΩΡΙΚΗ ΒΑΣΗ ΔΕΔΟΜΕΝΩΝ ΠΕΡΟΥΣΙΟΛΟΓΙΟ

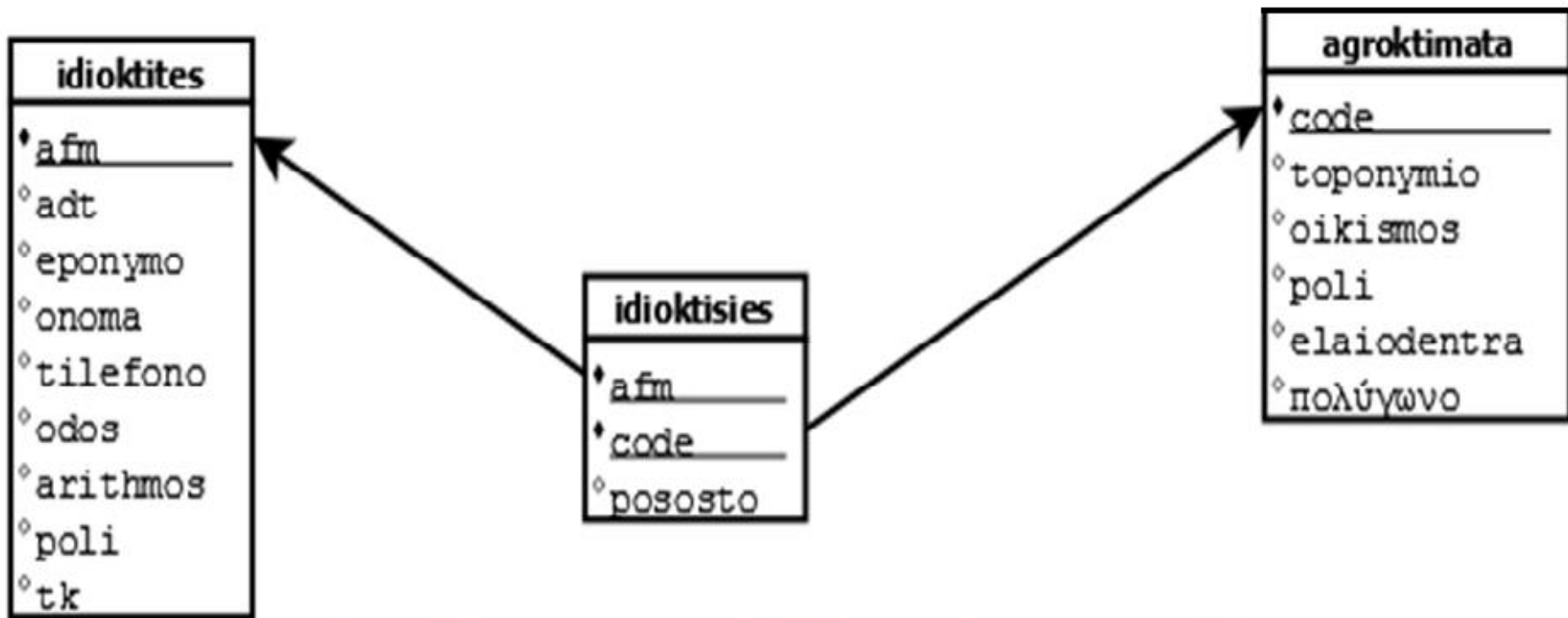
Θέλουμε να σχεδιάσουμε μία χωρική βάση δεδομένων στην οποία θα καταχωρούμε τους ιδιοκτήτες ελαιώνων και τους ελαιώνες (Ελαιοκομικό Μητρώο). Αναλυτικότερα θα περιέχει τα παρακάτω:

- Για τους ιδιοκτήτες θέλουμε να γνωρίζουμε τα παρακάτω στοιχεία: Αρ. Φορ. Μητρώου, Αρ. Δελτίου Ταυτότητας, Επώνυμο, Όνομα, Τηλέφωνο, Διεύθυνση(οδός, αριθμός, πόλη, ΤΚ). Θεωρείται γνωστό ότι ο Αρ.Φορ.Μητρώου καθώς και ο Αρ.Δελτίου Ταυτότητας έχουν την ιδιότητα της μοναδικότητας.
- Για τα αγροκτήματα θέλουμε να γνωρίζουμε τον κωδικό , το τοπωνύμιο, τον οικισμό, και την πόλη που βρίσκεται καθώς και το πλήθος των ελαιοδέντρων που περιέχει. Επίσης θέλουμε να γνωρίζουμε τις γεωγραφικές συντεταγμένες των κορυφών του και το εμβαδόν του.
- Κάθε ιδιοκτήτης κατέχει ένα ή περισσότερα αγροκτήματα και ένα αγρόκτημα ανήκει σε ένα ή περισσότερους ιδιοκτήτες. Για κάθε ιδιοκτησία πρέπει να γνωρίζουμε το ποσοστό που κατέχει ο ιδιοκτήτης. Το ποσοστό παίρνει τιμές στο διάστημα  $(0,100]$  ( ή αλλιώς  $>0$  AND  $\leq 100$  ).

# ΠΕΡΟΥΣΙΟΛΟΓΙΟ (Διάγραμμα οντοτήτων συσχετίσεων)



# ΠΕΡΟΥΣΙΟΛΟΓΙΟ (σχεσιακό σχήμα)

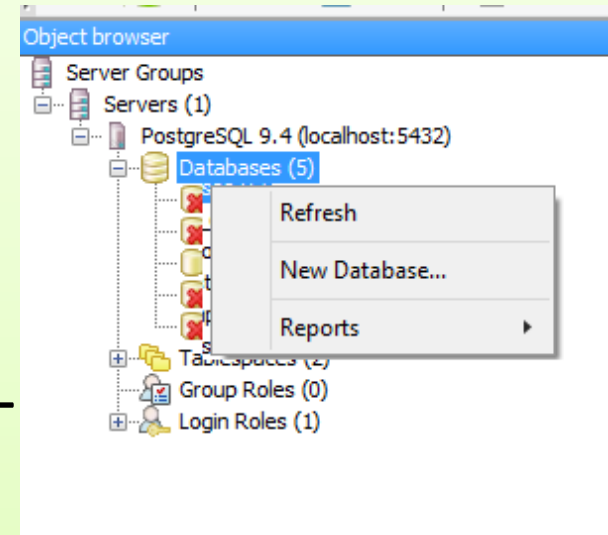


Το πολύγωνο και το εμβαδόν του θα τα παρουσιάσουμε στην διάλεξη που αφορά τα χωρικά δεδομένα

# Δημιουργία-Διαγραφή βάσης Δεδομένων

Για να δημιουργήσουμε μία βάση δεδομένων υπάρχουν 2 τρόποι:

Με δεξί κλικ στο εικονίδιο Databases εμφανίζεται μενού όπου κλικάρουμε New Database συμπληρώνουμε τα απαραίτητα ορίσματα για την δημιουργία της βάσης.



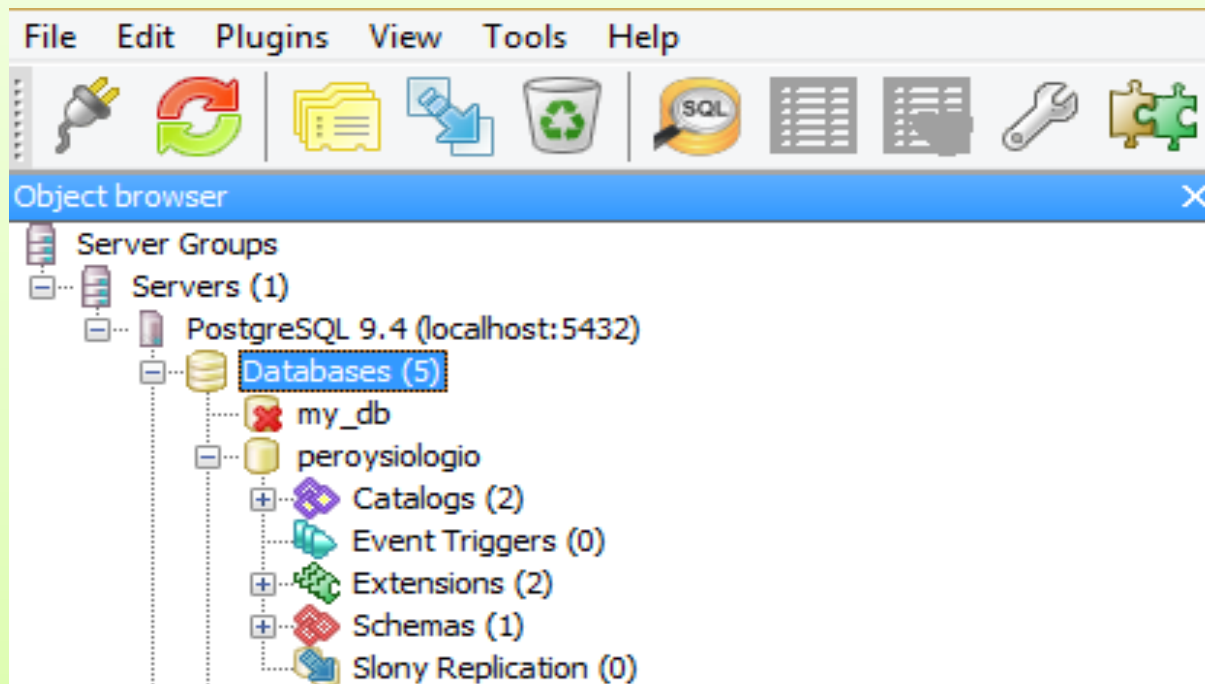
Με ερώτημα SQL. Δημιουργία βάσης peroysiologio.


CREATE DATABASE ονομα βάσης WITH  
ENCODING=κωδικοποιητής χαρακτήρων  
OWNER = ιδιοκτήτης TABLESPACE = .....

Διαγραφή βάσης: DROP DATABASE όνομα\_βάσης.

# Δημιουργία αντικειμένων της βάσης

Εμφανίζουμε τις βάσεις πατώντας το + που έχουμε δημιουργήσει και ξανά πατώντας το + που υπάρχει δίπλα από το όνομα της βάσης εμφανίζονται τα αντικείμενα της βάσης βλέπε εικόνα που ακολουθεί:



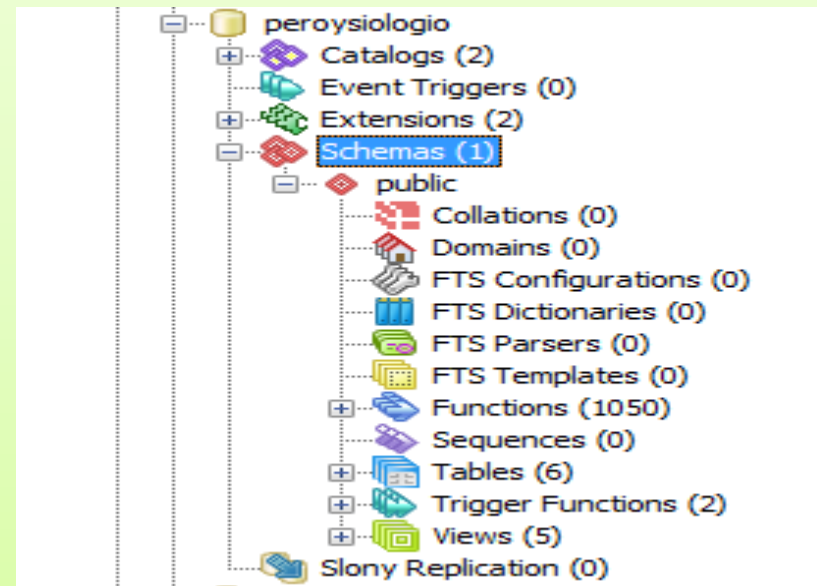
Σε περίπτωση που δεν εμφανίζεται ή βάση που δημιουργήσαμε ή οποιοδήποτε νέο αντικείμενο κάνουμε refresh πατώντας το 2ο εργαλείο της γραμμής εργαλείων 

# Δημιουργία σχημάτων στη βάση

Μία βάση δεδομένων περιέχει στοιχεία τα οποία είναι ομαδοποιημένα σε σχήματα. Το κάθε σχήμα είναι σαν κατάλογος που περιέχει πίνακες, όψεις, σκανδαλιστές, συναρτήσεις κ.α. Με την δημιουργία μίας βάσης δεδομένων, δημιουργείται αυτόματα ένα σχήμα το οποίο ονομάζεται **“public”**. Αν θέλουμε να δημιουργήσουμε και άλλο σχήμα κάνουμε δεξί κλικ στο αντικείμενο Schema και επιλέγουμε new schema ή με το ερώτημα **CREATE SCHEMA όνομα\_σχήματος ;**

Στο διπλανό σχήμα βλέπουμε τα αντικείμενα που περιέχει το σχήμα public. Βασικό αντικείμενο σε μια βάση δεδομένων είναι οι πίνακες που εμφανίζονται πατώντας το + στο αντικείμενο **Tables**.

Ανοίγοντας το εν λόγω αντικείμενο μπορούμε να διαχειριστούμε τους πίνακες που περιέχονται στη βάση.



# ΟΡΙΣΜΟΣ ΔΕΔΟΜΕΝΩΝ ΠΕΡΟΥΣΙΟΛΟΓΙΟΥ

Το πρώτο βήμα είναι να δημιουργήσουμε την βάση δεδομένων :

```
CREATE DATABASE perouysiologio WITH OWNER = postgres ENCODING = 'UTF8'  
TABLESPACE = pg_default LC_COLLATE = 'Greek_Greece.1253'  
LC_CTYPE = 'Greek_Greece.1253' CONNECTION LIMIT = -1;
```

Το παραπάνω Ερώτημα δημιουργεί μια βάση δεδομένων με όνομα perouysiologio. Η κωδικοποίηση των χαρακτήρων ορίζεται από το UTF-8. Ο χώρος αποθήκευσης των αντικειμένων της βάσης είναι ο προεπιλεγμένος χώρος. Η σειρά ταξινόμησης των χαρακτήρων ορίζεται από τον κώδικα Greece.1253. Δεν υπάρχει περιορισμός ως προς τον αριθμό των χρηστών που θα συνδεθούν στην βάση.

Όταν η βάση δεδομένων περιέχει μεγάλο αριθμό αντικειμένων(πίνακες, όψεις, ...) απαιτείται να γίνει οργάνωση του χώρου αποθήκευσης ώστε να είναι εύκολη η αναζήτηση τους. Για την εν λόγω οργάνωση τα στοιχεία της βάσης ομαδοποιούνται σε σχήματα. Η κάθε βάση περιέχει ένα σχήμα με όνομα **public** το οποίο δημιουργείται με την δημιουργία της.

Η δημιουργία σχήματος γίνεται με το ερώτημα : **CREATE SCHEMA όνομα\_σχήματος ;**

Η διαγραφή σχήματος γίνεται με το ερώτημα : **DROP SCHEMA όνομα\_σχήματος;**

Η παρούσα βάση δεδομένων περιέχει μικρό αριθμό πινάκων και δεν δημιουργήσουμε νέα σχήματα αλλά θα χρησιμοποιήσουμε μόνο το σχήμα public.

# ΕΡΩΤΗΜΑΤΑ ΔΗΜΙΟΥΡΓΙΑΣ- ΔΙΑΓΡΑΦΗΣ- ΜΕΤΑΒΟΛΗΣ ΠΙΝΑΚΩΝ

## **Δημιουργία Πίνακα.**

CREATE TABLE όνομα πίνακα (στήλη1 data type [NOT NULL] [DEFAULT αρχική τιμή], στηλη2 data type [NOT NULL] [DEFAULT αρχική τιμή], ... στήληN ..., PERIORISMOS\_AKAIREOTHTAS\_1, PERIORISMOS\_AKAIREOTHTAS\_2, ... PERIORISMOS\_AKAIREOTHTAS\_N, );

## **Διαγραφή Πίνακα.**

DROP TABLE όνομα πίνακα.

**Μεταβολή δομής Πίνακα** (εισαγωγή-διαγραφή στήλης, προσθήκη περιορισμού).

ALTER TABLE όνομα πίνακα ADD COLUMN στήλη data type );

ALTER TABLE όνομα πίνακα DROP COLUMN στήλη;

ALTER TABLE όνομα πίνακα ADD CONSTRAINT περιορισμός ακεραιότητας;

# ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΑ ΑΓΡΟΚΤΗΜΑΤΩΝ.

```
CREATE TABLE agroktimata
(  
code VARCHAR(10) NOT NULL,  
toponymio VARCHAR(30),  
oikismos VARCHAR(20),  
poli VARCHAR(20),  
elaiodentra INTEGER,  
CONSTRAINT pr_key_agro  
PRIMARY KEY (code)  
);
```

agroktimata	
•	<u>code</u>
◊	toponymio
◊	oikismos
◊	poli
◊	elaiodentra
◊	πολύγωνο

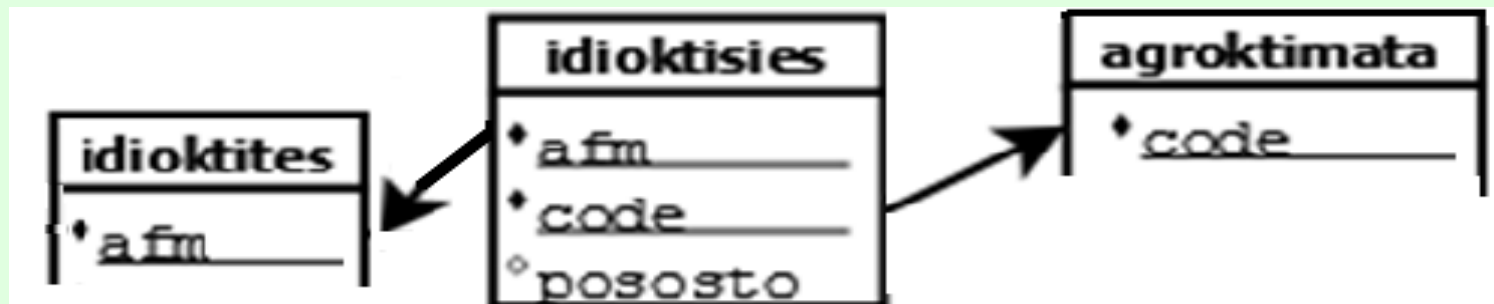
# ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΑ ΙΔΙΟΚΤΗΤΩΝ.

```
CREATE TABLE idioktites
(
  afm VARCHAR(12) NOT NULL,
  adt VARCHAR (8) NOT NULL,
  eponymo VARCHAR(20) NOT NULL,
  onoma VARCHAR(15),
  telefono VARCHAR(15),
  Odos VARCHAR(20),
  arithmos INTEGER,
  poli VARCHAR(20),
  tk VARCHAR(10),
  CONSTRAINT pr_key_idiokt
  PRIMARY KEY(AFM), CONSTRAINT
  uniq_idiokt UNIQUE (ADT)
);
```

idioktites
* <u>afm</u>
◊ adt
◊ eponymo
◊ onoma
◊ telefono
◊ odos
◊ arithmos
◊ poli
◊ tk

# ΔΗΜΙΟΥΡΓΙΑ ΠΙΝΑΚΑ ΙΔΙΟΚΤΗΣΙΩΝ

```
CREATE TABLE idioktisies( afm VARCHAR(12) NOT NULL,  
code VARCHAR(10) NOT NULL, pososto smallint,  
CONSTRAINT pr_key_idioktis PRIMARY KEY(AFM,code),  
CONSTRAINT check_psosto CHECK(pososto<=100 AND  
pososto>0), CONSTRAINT forkey_afm FOREIGN KEY (afm)  
REFERENCES idioktites(afm) MATCH SIMPLE ON UPDATE  
RESTRICT ON DELETE RESTRICT, CONSTRAINT forkey_code  
FOREIGN KEY(code) REFERENCES agroktimata(code) MATCH  
SIMPLE ON UPDATE RESTRICT ON DELETE RESTRICT);
```



# Καταχώριση Δεδομένων.

Μπορεί να γίνει με 2 τρόπους:

A) Με δεξί κλικ πάνω στο όνομα του πίνακα εμφανίζεται μενού από όπου με την επιλογή **view data** μπορούμε να καταχωρήσουμε νέες εγγραφές δεδομένων και γενικότερα να διαγράψουμε, να μεταβάλλουμε κ.λπ. τα περιεχόμενα του πίνακα.

B) Με την χρήση Ερωτημάτων INSERT INTO με 2 τρόπους:

B1) INSERT INTO όνομα\_πίνακα VALUES (value1, value2, ... ),(...),...;

B2) INSERT INTO όνομα\_πίνακα (col1, col2, col3) VALUES(value\_1, value\_2, value\_3);

Στον τρόπο B1 το πλήθος των ορισμάτων στις παρενθέσεις είναι όσες είναι οι στήλες του πίνακα που ενημερώνεται και υπάρχει αντιστοιχία της σειράς των τιμών και των στηλών του πίνακα, ενώ στον τρόπο B2 υπάρχει αντιστοιχία τιμών και των στηλών που αναφέρονται στις παρενθέσεις πριν το Values.

## Εισαγωγή εγγραφών στην βάση peroysiologio στον πίνακα ιδιοκτήτες.

```
INSERT INTO idioktites VALUES('123456789', 'ZΛ123456',  
'Παπαδάκης', 'Γιάννης', '2345678', 'Καλοκαιρινού',  
100, 'Ηράκλειο', '71410');
```

```
INSERT INTO idioktites (afm, adt, eponymo, onoma, poli)  
VALUES ('1234562000', 'ZH123456', 'Τσαγκάρης', 'Νίκος',  
'Ηράκλειο');
```

```
INSERT INTO idioktites  
( '4000', 'HI345678', 'Μπελής', 'Γιάννης', '345', NULL, '0', 'Ιεράπε  
τρα', NULL),  
( '3000', 'ZΛ123478', 'Δεληγιάννης', 'Ηλίας', '2345678', 'Κατεχα  
κη', '5', 'Ηράκλειο', '71410');
```

# Εισαγωγή εγγραφών στην βάση peroysiologio στον πίνακα αγροκτήματα.

```
INSERT INTO agroktimata VALUES  
( 'Σ1', 'Αποσαμι Μετοχι', 'Σκαλανι', 'Ηράκλειο', '33'),  
( 'Κ1', 'Κορακοβούνι', 'Αμφιθέα', 'Ηράκλειο', '2');  
( 'ΚΜ1', 'Κάμπος', 'Αρκαλοχώρι', 'Μινώα Πεδιάδος', '23'),  
( 'ΚΜ2', 'Κάμπος', 'Αρκαλοχώρι', 'Μινώα Πεδιάδος', '4'),  
( 'ΒΝ1', 'Βουνό', 'Αρκαλοχώρι', 'Μινώα Πεδιάδος', '27'),  
( 'ΦΛ1', 'Φλέγα', 'Μακρυλιά', 'Ιεράπετρα', '10'),  
( 'ΦΛ2', 'Φλέγα', 'Μακρυλιά', 'Ιεράπετρα', '8'),  
( 'ΒΝ2', 'Βουνό', 'Αρκαλοχώρι', 'Μινώα Πεδιάδος', '15'),  
( 'ΠΝ1', 'Ζουρβες', 'Μακρυλιά', 'Ιεράπετρα', '26'),  
( 'ΨΡΚ1', 'Ψαροκεφάλα', 'Αρκαλοχώρι', 'Μινώα Πεδιάδος', '16'),  
( 'ΨΡΚ2', 'Ψαροκεφάλα', 'Αρκαλοχώρι', 'Μινώα Πεδιάδος', '27');
```

## Εισαγωγή εγγραφών στην βάση peroysiologio στον πίνακα ιδιοκτησίες.

```
INSERT INTO idioktisies  
VALUES('1000','BN2','50'),('1000','Σ1','100'),  
('3000','BN2','60'),('3000','BN1','100'),  
('3000','Κ1','50'),('4000','ΚΜ1','100'),('4000','Κ1','50'),  
('3000','ΦΛ1','100'),('2000','ΚΜ2','10'),  
('3000','ΦΛ2','100'),('4000','ΨΡΚ1','100'),  
('4000','ΨΡΚ2','100'),('3000','ΠΝ1','100'),  
('2000','ΠΝ1','20');
```

# Ενημέρωση εγγραφών.

Τα ερωτήματα ενημέρωσης, μεταβάλουν τα περιεχόμενα των πεδίων των εγγραφών ενός πίνακα οι οποίες ικανοποιούν κάποιο κριτήριο(συνθήκη).

Η εντολή συντάσσεται:

UPDATE ονομα\_πίνακα set col\_1=value1, ..., col\_N=value\_N where συνθήκη;

Παραδείγματα:

α) Το Ερώτημα που ακολουθεί μεταβάλλει το επώνυμο και την πόλη του ιδιοκτήτη που έχει Αρ. Δελτίου Ταυτότητας ΖΗ123456

```
UPDATE idioktites SET eponymo='Τσαγκαράκης', poli='Χανιά' WHERE adt='ZH123456';
```

β) Το Ερώτημα που ακολουθεί μεταβάλλει το όνομα την πόλης των αγροκτημάτων σε Ηράκλειο όταν αυτή είναι Herakleion ή Hraklio.

```
UPDATE agroktimata set poli='Ηράκλειο' where poli='Herakleion' or poli='Hraklio';
```

# Διαγραφή εγγραφών.

Με τα ερωτήματα διαγραφής μπορούμε να διαγράψουμε τις εγγραφές ενός πίνακα οι οποίες ικανοποιούν κάποιο κριτήριο(συνθήκη).

Η εντολή συντάσσεται:

```
DELETE From onoma_pinaka where συνθήκη;
```

Παραδείγματα:

α)το ερώτημα που ακολουθεί διαγράφει τον ιδιοκτήτη με ΑΦΜ 1234562000

```
DELETE FROM idioktites WHERE afm='1234562000';
```

β)το ερώτημα που ακολουθεί διαγράφει όλους τους ιδιοκτήτες των οποίων το ΑΦΜ δεν υπάρχει στις ιδιοκτησίες(δεν έχουν αγροκτήματα)

```
DELETE FROM idioktites WHERE afm not in (select afm from idioktisies);
```

# Παραδείγματα Ερωτημάτων Επιλογής Δεδομένων

**Εμφάνιση όλων των στοιχείων των ιδιοκτητών από το Ηράκλειο.**

```
SELECT * FROM idioktites WHERE poli='Ηράκλειο';
```

**Εμφάνιση των ΑΔΤ, ΕΠΩΝΥΜΩΝ και ΟΝΟΜΑΤΩΝ ιδιοκτητών από το Ηράκλειο με ταξινόμηση ως προς το ΕΠΩΝΥΜΟ και το ΟΝΟΜΑ.**

```
SELECT adt, eponymo , onoma FROM idioktites WHERE  
poli="Ηράκλειο" order by eponymo,onoma;
```

**Εμφάνιση όλων των αγροκτημάτων που έχουν περισσότερα από 5 ελαιόδενδρα και βρίσκονται στο Ηράκλειο ή στην Ιεράπετρα.**

```
select * from agroktimata where poli in ('Ηράκλειο','Ιεράπετρα') and  
elaiodentra>5;
```

# Παραδείγματα Επιλογής Δεδομένων από πολλούς πίνακες

Όταν σε ένα ερώτημα χρησιμοποιούμε περισσότερους από ένα πίνακες πρέπει:

A) να αναφέρουμε την σύνδεση που υπάρχει μεταξύ τους αλλιώς δημιουργείται καρτεσιανό γινόμενο (το οποίο χρησιμοποιείται σπάνια).

B) Όταν χρησιμοποιούμε ένα πεδίο το οποίο υπάρχει σε περισσότερους από ένα πίνακες με το ίδιο όνομα για να διευκρινήσουμε από ποιο πίνακα προέρχεται γράφουμε το ονομα\_πίνακα.ονομα\_πεδίου.

**Εμφάνιση των αγροτεμαχίων του ιδιοκτήτη με ΑΦΜ 3000.**

```
select idioktisies.code, toponymio, oikismos, poli, elaiodentra, pososto from idioktisies  
join agroktimata on idioktisies.code=agroktimata.code where afm='3000';
```

**Εμφάνιση όλων των αγροκτημάτων και των κατόχων τους.**

```
select idioktisies.code, toponymio, oikismos, agroktimata.poli,  
elaiodentra,pososto,idioktites.afm,eponymo,onoma from (idioktisies join agroktimata  
on idioktisies.code=agroktimata.code) join idioktites on idioktites.afm=idioktisies.afm;
```

# Ερωτήματα συναθροιστικών αποτελεσμάτων

Τα ερωτήματα συναθροιστικών αποτελεσμάτων ομαδοποιούν τις εγγραφές ως προς τιμές κάποιου πεδίου (group by πεδίο). Σε κάθε ομάδα εγγραφών που δημιουργείτε εφαρμόζουμε την συναθροιστική συνάρτηση (αθροισμα, μέσο όρο, τυπική απόκλιση, μέγιστο, ελάχιστο κ.λ.π). Παράδειγμα αν θέλω να βγάλω τους μέσους όρους των βαθμών που έχει περάσει τα μαθήματα του ο κάθε σπουδαστής, ομαδοποιώ τις βαθμολογίες ανά αριθμό μητρώου και για κάθε ομάδα που δημιουργείται υπολογίζεται ο μέσος όρος των βαθμών.

## Παραδείγματα:

A) Εμφάνιση των στοιχείων όλων των ιδιοκτητών καθώς και τον συνολικό αριθμό ελαιόδέντρων που κατέχει ο καθένας.

```
select idioktites.afm,adt,eponymo,onoma,sum(elaiodentra*pososto/100) as  
plithos_elaiodentron from (idioktisies join agroktimata on  
idioktisies.code=agroktimata.code) join idioktites on idioktites.afm=idioktisies.afm  
group by idioktites.afm;
```

B) Εμφάνιση των στοιχείων όλων των αγροτεμαχίων που το συνολικό ποσοστό ιδιοκτησίας των κατόχων του είναι διάφορο του 100.

```
select agroktimata.code, toponymio, oikismos, agroktimata.poli, elaiodentra,  
sum(pososto) from idioktisies join agroktimata on idioktisies.code=agroktimata.code  
group by agroktimata.code having sum(pososto)<>100
```