



Geo-Database

Strumenti e tecniche per la gestione efficiente e scalabile di basi di dati geografiche

Corso di Sistemi Informativi Territoriali Avanzati – UD10

prof. Giovanni Borga



Geo-Database

Cos'è un geo-database

Un geo-database ha tutte le **caratteristiche proprie di un DBMS** (Data Base Management System) non geografico.

In un geo-DBMS sono installate le cosiddette **estensioni spaziali** (spatial extensions), ovvero moduli software che permettono di:

- effettuare lo **storage** delle geometrie di un livello informativo (layer GIS es. punti, linee e poligoni)
- effettuare **analisi e interrogazioni** delle proprietà geometriche degli oggetti presenti nell'archivio
- stabilire e valutare **relazioni spaziali** fra gli elementi informativi provvisti di geometria presenti nell'archivio
- effettuare il **processamento** dell'informazione geometrica presente nell'archivio utilizzando gli operatori spaziali classici degli strumenti GIS

Cos'è un geo-database

Un geoDB / geoDBMS

NON è un file

E', salvo pochissime eccezioni, un'applicazione di tipo SERVER, che:

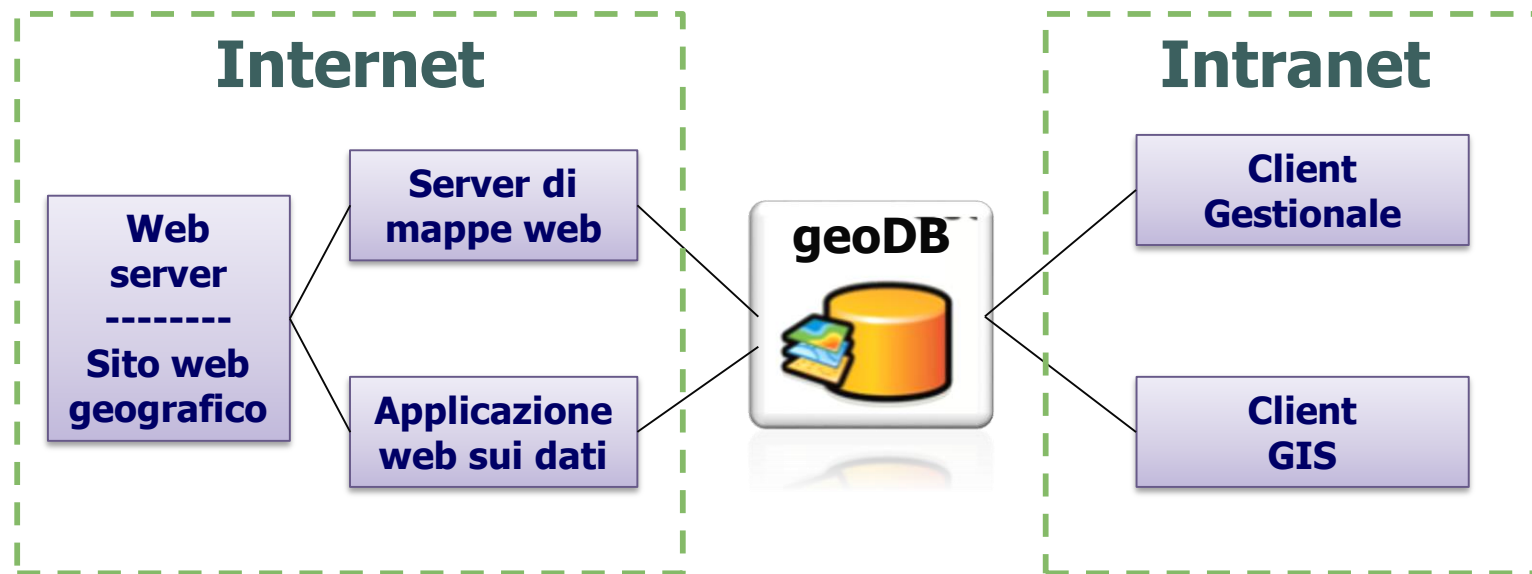
- Deve essere mandata in esecuzione all'interno un sistema operativo
- L'accesso e l'utilizzo avviene per mezzo di altre applicazioni; applicazioni di tipo CLIENT
- E' naturalmente predisposta per l'accesso simultaneo multi-utente via rete LAN o WEB e le prestazioni dipendono fortemente, oltre che dalle caratteristiche hardware e software del sistema che ospita il geoDBMS, dal numero di utenti connessi e dalla mole di richieste inoltrate da ognuno di essi al sistema.



Architetture di sistema

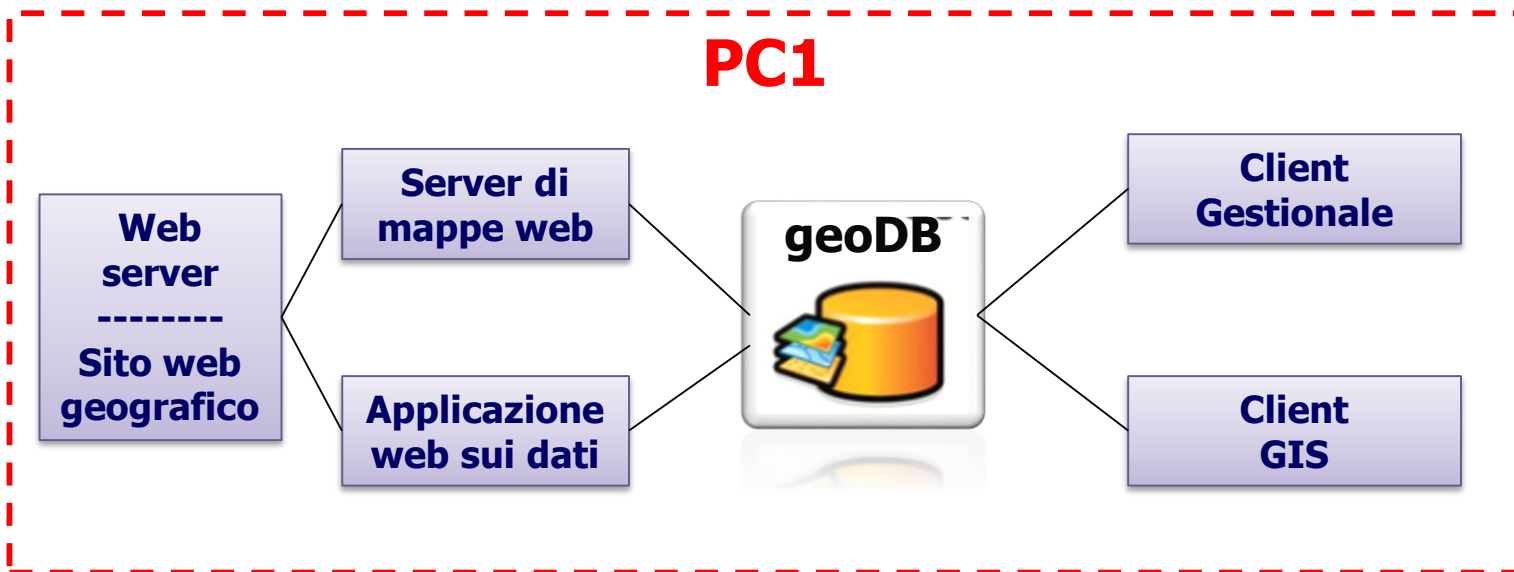
Architetture di sistema

Esempio di configurazione dei **componenti software** in un contesto di rete



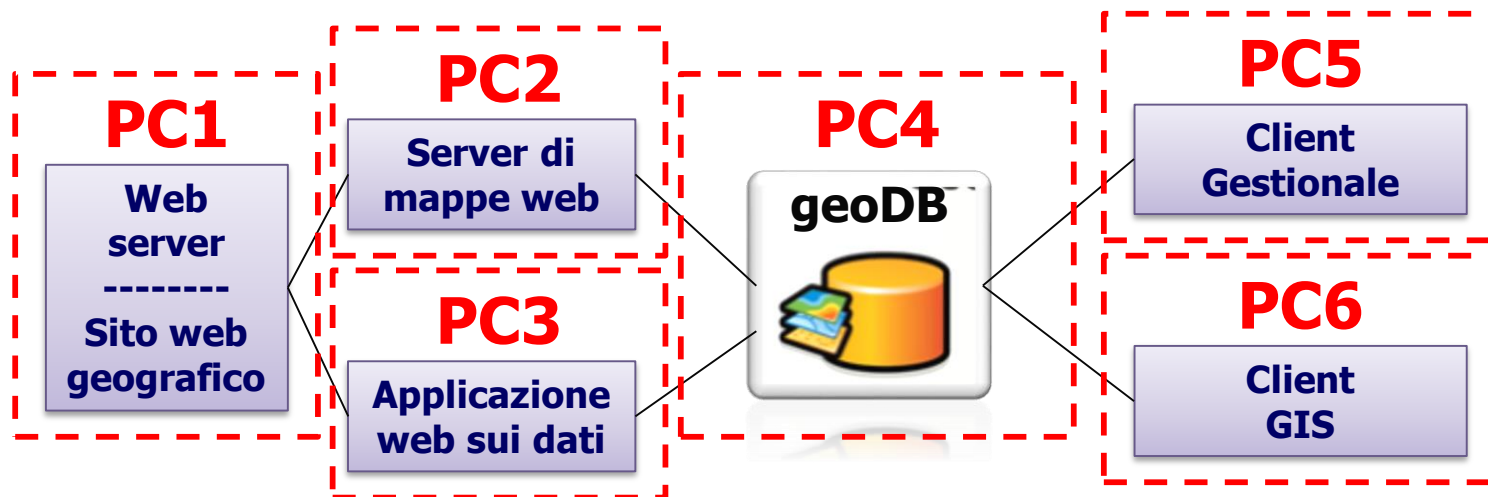
Architetture di sistema

Esempio di **soluzione hardware monolitica**



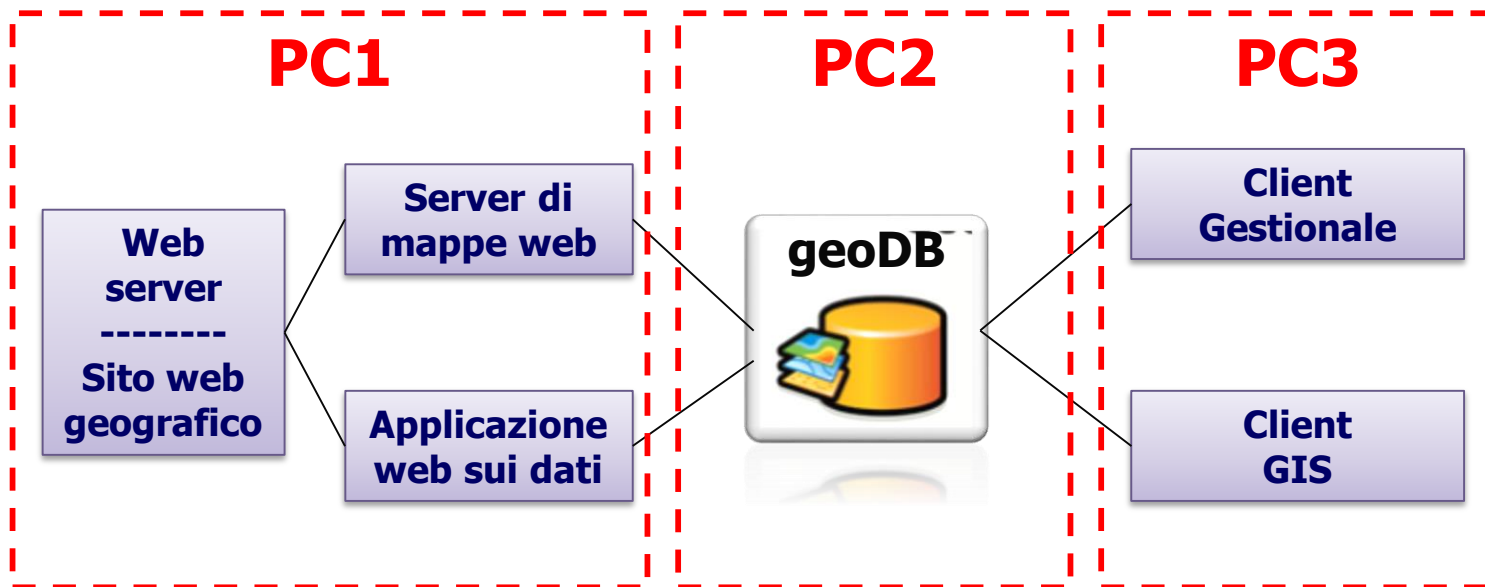
Architetture di sistema

Esempio di **soluzione hardware segmentata**



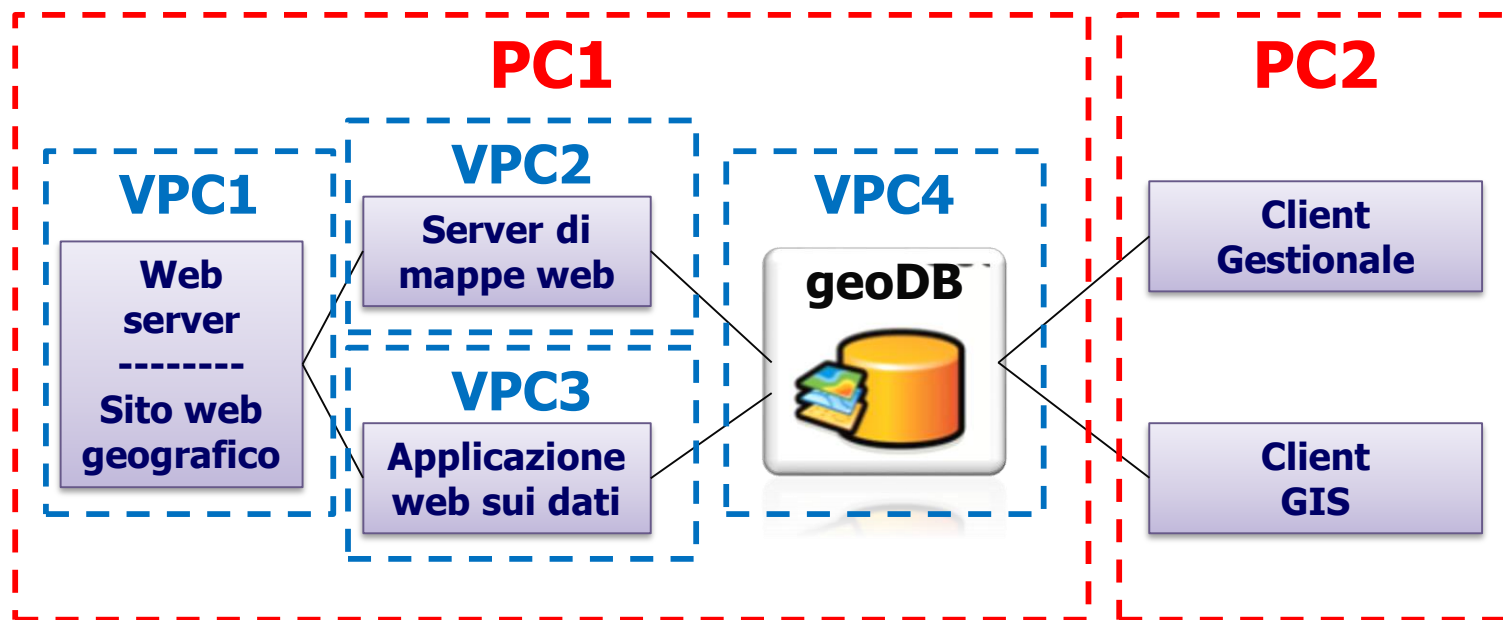
Architetture di sistema

Esempio di **soluzione hardware parzialmente segmentata**



Architetture di sistema

Esempio di soluzione hardware virtualizzata



Soluzioni software attualmente più diffuse

Corporate

- ❑ Oracle Spatial (sistema completo di tipo client server)
- ❑ MS SQL Server 2008 (sistema completo di tipo client server)
- ❑ MS MDB personal geodatabase (estensioni del sistema file-based MS Access utilizzate da ESRI e Intergraph in ambiente desktop)

Free Open Source

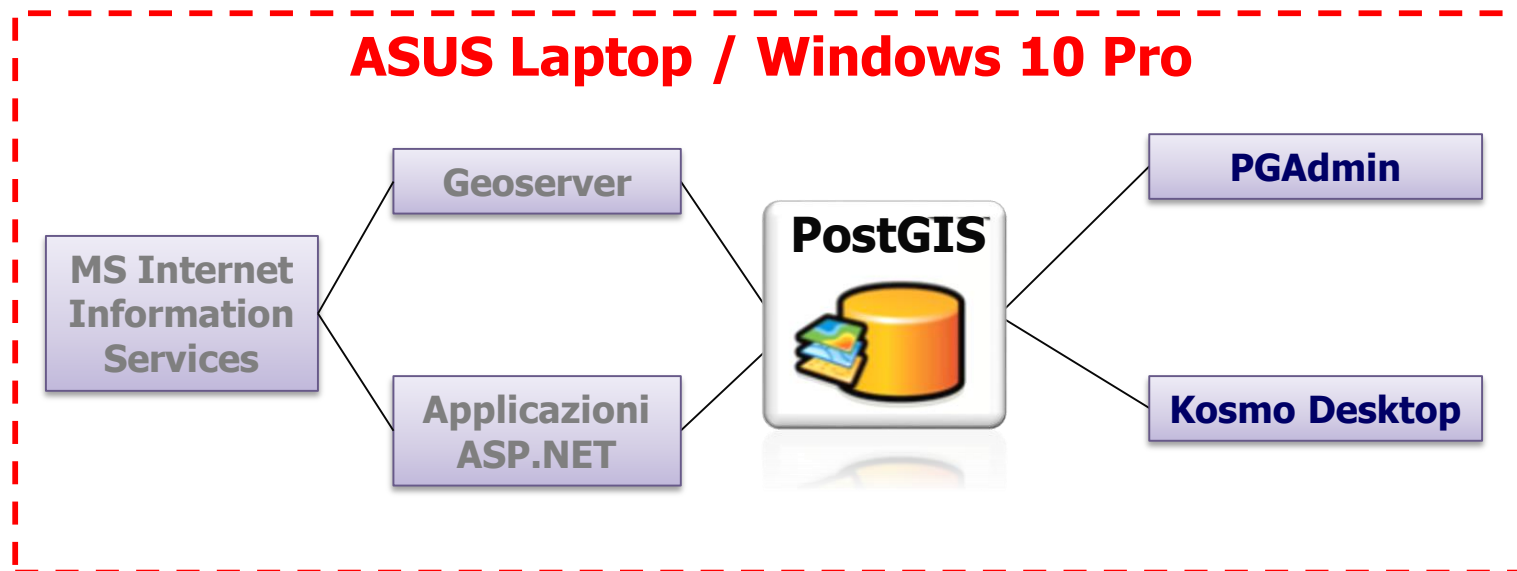
- ❑ PostGIS (sistema completo di tipo client server)
- ❑ MySQL spatial (sistema completo di tipo client server)
- ❑ Spatialite (estensione del sistema file-based SQLite)



PostGIS

Architettura utilizzata

Configurazione software



Getting started



The world's most advanced open source database.

Home About Download Documentation Community Developers Support Your account

www.postgresql.org

PostgreSQL minor versions released!

The PostgreSQL Global Development Group announces updates for all active branches of the PostgreSQL object-relational database system, including versions 9.2.1, 9.1.6, 9.0.10, 8.4.14 and 8.3.21. These releases fix a number of bugs, including a potential data corruption bug in the 9.1 and 9.2 releases.



- » [Release Announcement](#)
- » [Download](#)

FEATURED USER

We embed PostgreSQL in our Cloud Appliances and we rely on PostgreSQL in our Ubuntu packages. While the product supports multiple databases, 93% of Openbravo deployments are on PostgreSQL.

Paolo Juvara, CTO, [Openbravo](#)

- » [Case Studies](#)
- » [More Quotes](#)
- » [Featured Users](#)

LATEST NEWS

2012-09-24 [PostgreSQL Updates 2012-09-24 released](#)

UPCOMING EVENTS

2012-10-23 – 2012-10-26 [PGConf.EU 2012](#) (Prague, Czech Republic)

PLANET POSTGIS

2012-10-16 [Dimitri Fontaine](#)

2012-10-16

LATEST RELEASES

- 9.2.1 · 2012-09-24 · [Notes](#)
- 9.1.6 · 2012-09-24 · [Notes](#)
- 9.0.10 · 2012-09-24 · [Notes](#)
- 8.4.14 · 2012-09-24 · [Notes](#)
- 8.3.21 · 2012-09-24 · [Notes](#)

[Download](#)
[Why show](#)

SHORTCUTS

- » [Security](#)
- » [Internal](#)
- » [Mailing](#)
- » [Wiki](#)
- » [Report a bug](#)
- » [FAQs](#)

SUPPORT

PostgreSQL support on [donation](#).



PostGIS

OSGeo Project

HOME DOCUMENTATION DOWNLOADS SUPPORT NEWS

Home > Downloads

Downloads

Major number releases are made as substantial functionality milestones are reached (object support, spatial indexing, mapserver support, SRID support) and minor number releases are made to bring bug fixes and minor feature enhancements to the general user community.

SOURCE CODE

Stable Releases:

- [+] [2012/06/22] [postgis-2.0.1.tar.gz](#) (~5.1M)
- [+] [2012/07/20] [postgis-1.5.5.tar.gz](#) (~3.6M)
- [+] [2009/03/11] [postgis-1.4.2.tar.gz](#) (~3.2M)
- [+] [2009/05/04] [postgis-1.3.6.tar.gz](#) (~2.2M)
- [+] [2007/01/11] [postgis-1.2.1.tar.gz](#) (~2.0M)
- [+] [2007/01/31] [postgis-1.1.7.tar.gz](#) (~2.0M)
- [+] [2005/12/06] [postgis-1.0.6.tar.gz](#) (~1.4M)

Latest Development Snapshot: (revision 10448)

- [postgis-2.1.0SVN.tar.gz](#)

Latest 2.0 Snapshot: (revision 10450)

SVN

SVN SNAPSHOT

A snapshot of the SVN repository is generated after every successful commit and subsequent build and posted to the web site. Download the snapshot below:

[postgis-2.1.0SVN.tar.gz](#)

SVN ACCESS

The SVN archive is located at <http://svn.osgeo.org/postgis/>.

SVN access is only recommended for those who need the absolute latest and greatest code, and are not afraid of getting their hands dirty building the source. Here's how to access the read-only

postgis.refractor.net/download

Getting started

Linux

Windows

MAC OS X

SOURCE CODE

Stable Releases:

- [+] [2012/06/22] [postgis-2.0.1.tar.gz](#) (~5.1M)
- [+] [2012/07/20] [postgis-1.5.5.tar.gz](#) (~3.6M)
- [+] [2009/03/11] [postgis-1.4.2.tar.gz](#) (~3.2M)
- [+] [2009/05/04] [postgis-1.3.6.tar.gz](#) (~2.2M)
- [+] [2007/01/11] [postgis-1.2.1.tar.gz](#) (~2.0M)
- [+] [2007/01/31] [postgis-1.1.7.tar.gz](#) (~2.0M)
- [+] [2005/12/06] [postgis-1.0.6.tar.gz](#) (~1.4M)
- [+] [2005/07/25] [postgis-0.9.2.tar.gz](#) (~480K)

Latest Development Snapshot: (revision 10448)

- [postgis-2.1.0SVN.tar.gz](#)

Latest 2.0 Snapshot: (revision 10450)

- [postgis-2.0.2SVN.tar.gz](#)

Latest 1.5 Snapshot: (revision 10186)

- [postgis-1.5.6SVN.tar.gz](#)

Latest 1.4 Snapshot: (revision 9752)

- [postgis-1.4.3SVN.tar.gz](#)

WINDOWS

First read the [Windows PostGIS installation instructions](#) to help you decide how you want to install PostGIS and which versions of PostgreSQL and PostGIS you want to install (the latest stable releases are recommended).

You must install PostgreSQL first, before PostGIS can be installed. You can download the latest Windows PostgreSQL installer from <http://www.postgresql.org/download/windows>.

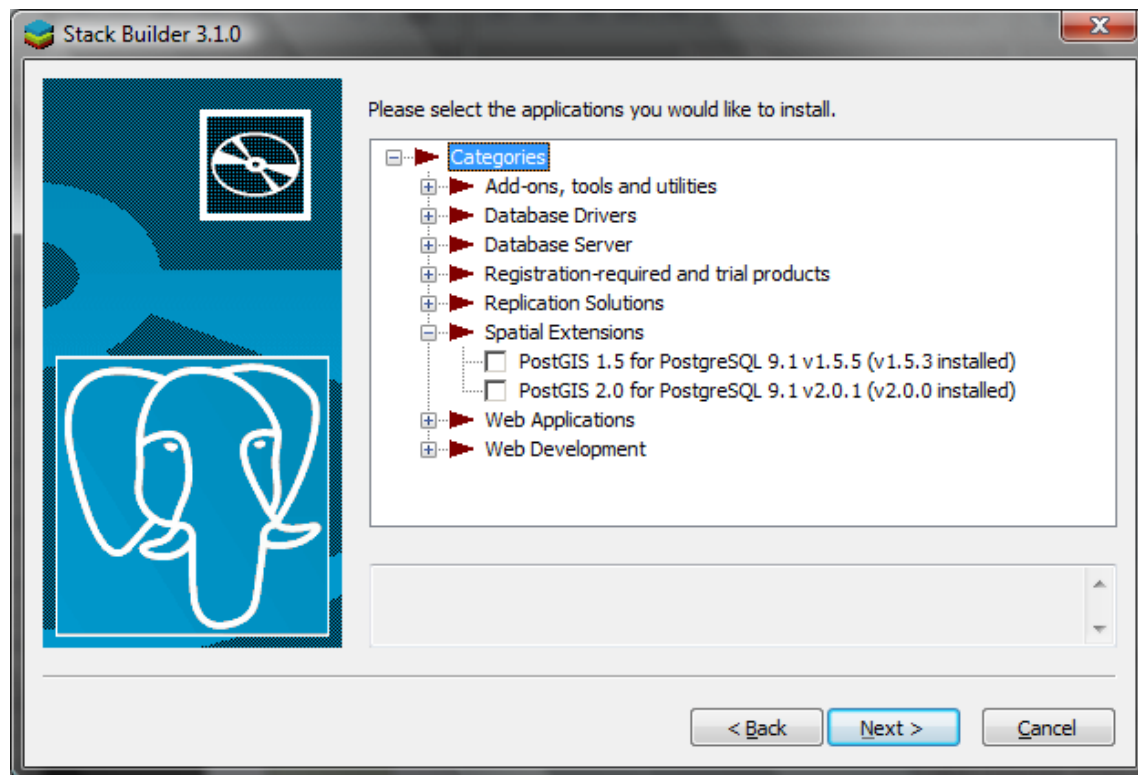
MAC OS X

[OS X binaries for PostGIS and PostgreSQL](#) are packaged regularly by William Kyngesburye.

Packages for Fink are [also available](#).

Getting started

PostgreSQL application stack builder



Struttura di una base dati geografica

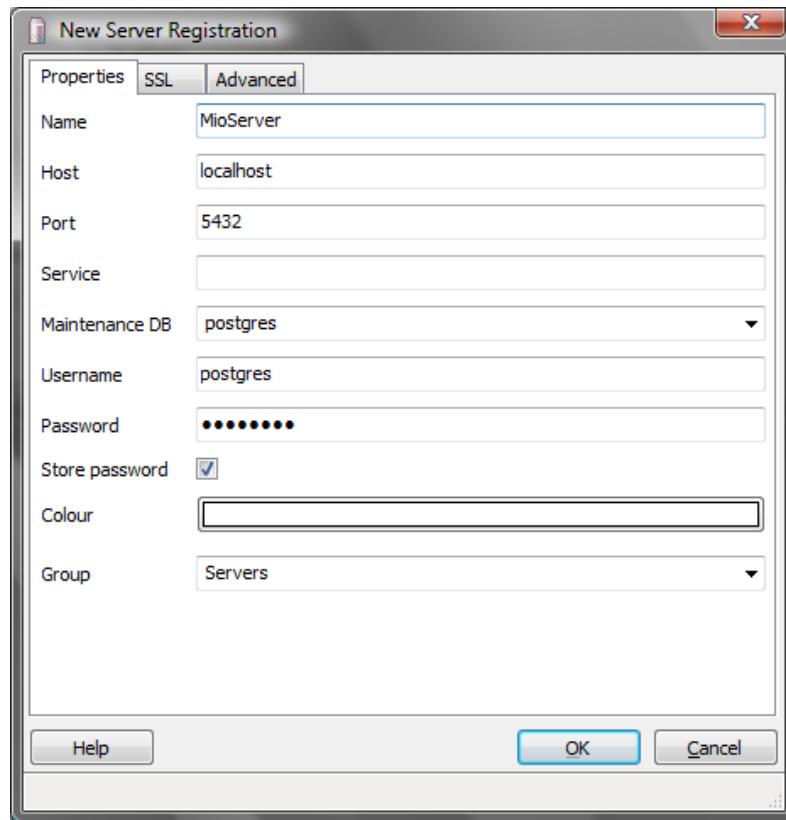
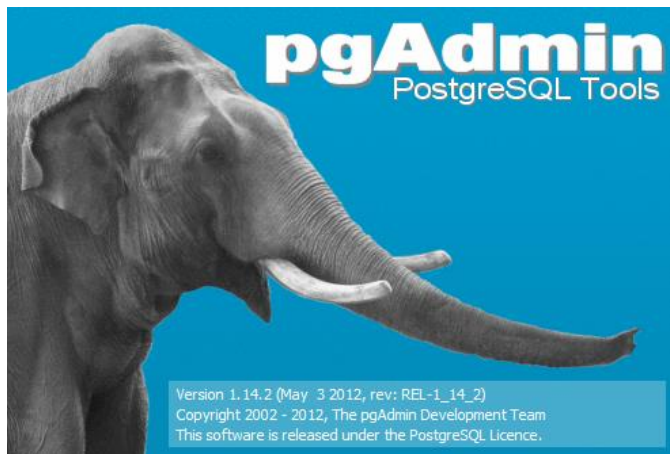
La struttura di una base dati geografica è gerarchica.
Gli elementi essenziali sono:

- **Server**
 - **Database**
 - ❖ **Schema**
 - ✓ **Tables**
 - ✓ **Views**
 - ✓ **Functions**

Client di gestione

PGAdmin III

Connessione a PostgreSQL



The screenshot shows the "New Server Registration" dialog box in PGAdmin III. The dialog has three tabs: "Properties", "SSL", and "Advanced". The "Properties" tab is selected. The fields are filled with the following information:

- Name: MioServer
- Host: localhost
- Port: 5432
- Service: (empty)
- Maintenance DB: postgres
- Username: postgres
- Password: (masked with dots)
- Store password:
- Colour: (empty)
- Group: Servers

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

L'ambiente configurato

Elementi predefiniti

The screenshot displays the PostgreSQL Enterprise Manager interface. On the left, a tree view shows the server structure. The main pane shows the 'Properties' tab for a database, with a table listing the following databases:

| Database | Owner | Comment |
|---------------------|----------|---------|
| energywebfeltr | postgres | |
| mantova | postgres | |
| ossro | postgres | |
| postgis | postgres | |
| postgres | postgres | |
| rovigo | postgres | |
| template_postgis | postgres | |
| template_postgis_20 | postgres | |

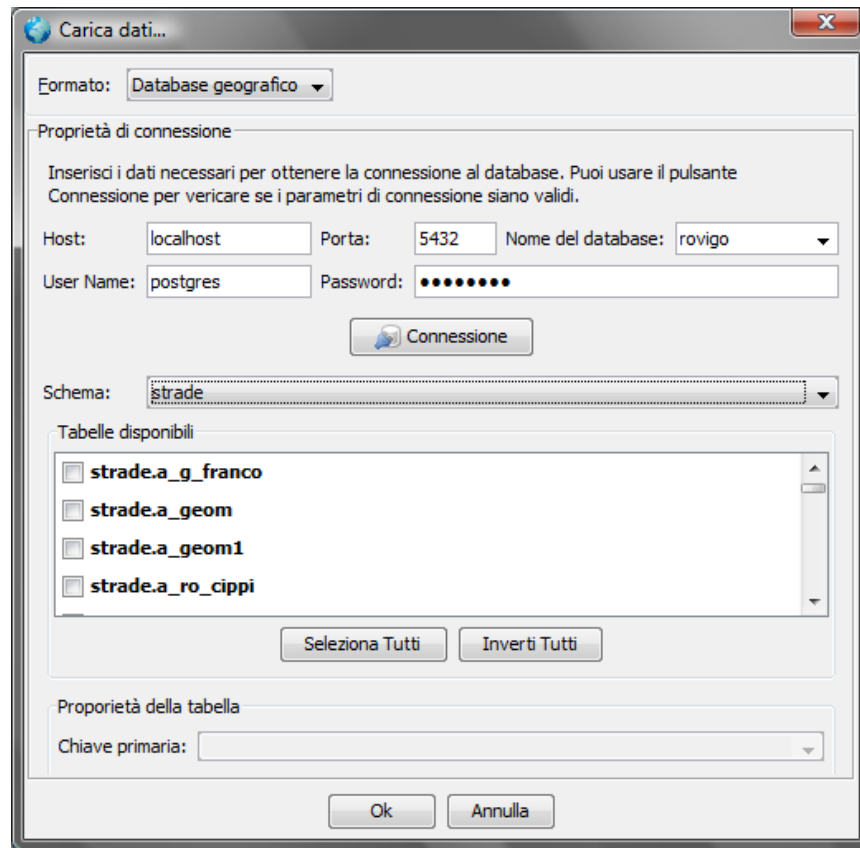
Below the table is an empty 'SQL pane'. On the right, a detailed tree view of the server configuration is shown, with the following elements:

- Servers (3)
 - PostgreSQL 9.1 (localhost:5432)
 - Databases (8)
 - energywebfeltr
 - mantova
 - ossro
 - postgis
 - postgres
 - rovigo
 - template_postgis
 - template_postgis_20
 - Tablespaces (2)
 - Group Roles (0)
 - Login Roles (1)

Client geografico

Kosmo Desktop 2.0

Connessione a PostgreSQL



Le basi

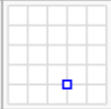
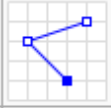
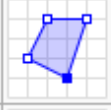
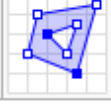
Tipi di geometrie supportate

- **Semplici:** point, line, polygon
- **Multi-part:** multipoint, multiline, multipolygon
- **Collazioni:** geometrycollection

Le geometrie sono specificate secondo il formato Open GIS:

Well Known Text

Definito da
Open Geospatial Consortium (OGC)

| Geometry primitives (2D) | | |
|--------------------------|---|--|
| Type | Examples | |
| Point | POINT (30 10) |  |
| LineString | LINestring (30 10, 10 30, 40 40) |  |
| Polygon | POLYGON ((30 10, 10 20, 20 40, 40 40, 30 10)) |  |
| | POLYGON ((35 10, 10 20, 15 40, 45 45, 35 10), (20 30, 35 35, 30 20, 20 30)) |  |

Le basi

La tabella / vista geometry_columns

| | oid | f_table_catalog [PK] character va | f_table_schema [PK] character va | f_table_name [PK] character | f_geometry_ [PK] character | coord_dimension integer | srid integer | type character va |
|----|--------|--------------------------------------|-------------------------------------|--------------------------------|-------------------------------|----------------------------|-----------------|----------------------|
| 1 | 78662 | '' | strade | a_g_franco | the_geom | 2 | 3003 | POINT |
| 2 | 66969 | '' | strade | a_geom | the_geom | 2 | 3003 | LINestring |
| 3 | 103912 | '' | strade | a_geom1 | the_geom | 2 | 3003 | POINT |
| 4 | 112439 | '' | strade | a_ro_cippi | the_geom | 2 | 3003 | POINT |
| 5 | 113673 | '' | strade | a_ro_cippi2 | the_geom | 2 | 3003 | POINT |
| 6 | 103461 | '' | strade | b | GEOMETRY | 2 | 3003 | POLYGON |
| 7 | 162132 | '' | strade | comunerovigo | GEOMETRY | 2 | 3003 | POLYGON |
| 8 | 17472 | '' | strade | confinepro | GEOMETRY | 2 | 3003 | POLYGON |
| 9 | 120860 | '' | strade | franco | the_geom | 2 | 3003 | LINestring |
| 10 | 103562 | '' | strade | grafo | GEOMETRY | 2 | 3003 | LINestring |
| 11 | 113659 | '' | strade | grafo_gplan | GEOMETRY | 2 | 3003 | POINT |
| 12 | 103703 | '' | strade | grafo_nodi | GEOMETRY | 2 | 3003 | MULTIPOINT |
| 13 | 66891 | '' | strade | grid | GEOMETRY | 2 | 3003 | POLYGON |
| 14 | 17457 | '' | strade | grid100 | GEOMETRY | 2 | 3003 | POLYGON |
| 15 | 32816 | '' | strade | IND1 | GEOMETRY | 2 | 3003 | POLYGON |
| 16 | 104110 | '' | strade | poli | the_geom | 2 | 3003 | POLYGON |
| 17 | 95260 | '' | strade | ro_a_barrier | the_geom | 2 | 3003 | LINestring |

38 rows.

Le basi

Hello world!

Creare un nuovo database:

```
CREATE DATABASE test TEMPLATE=template_postgis
```

Creare un'entità (*table*) di elementi geografici:

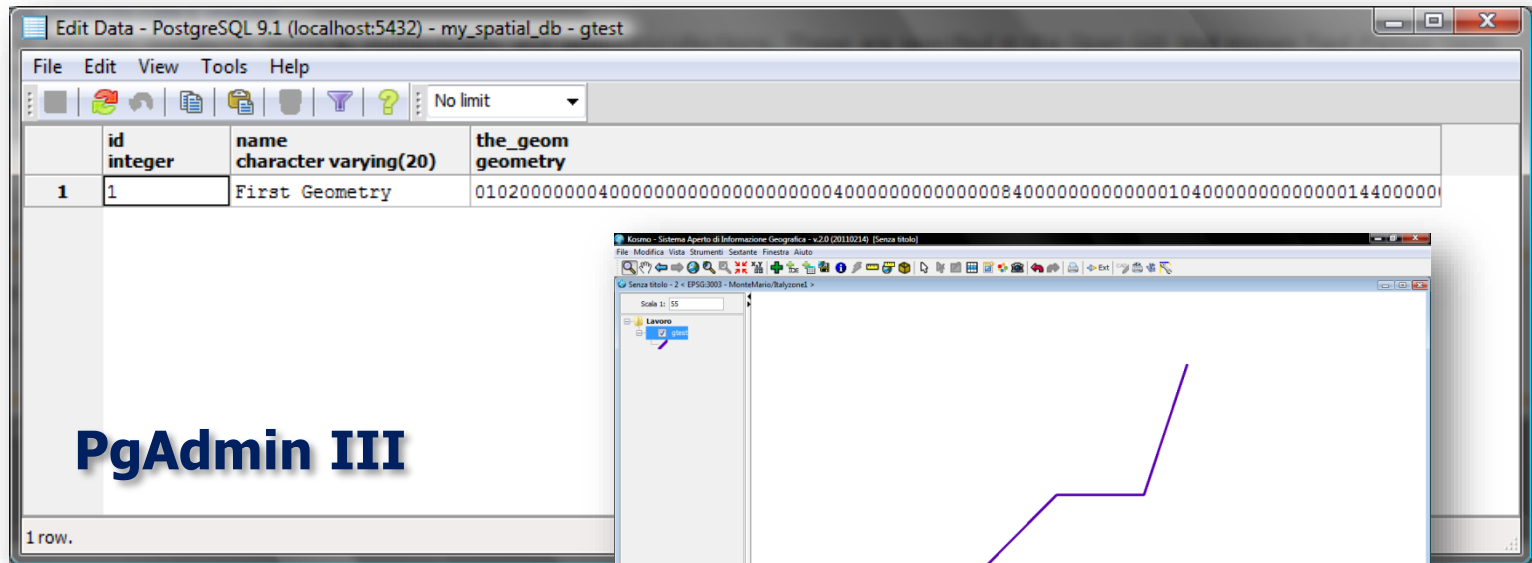
```
CREATE TABLE gtest (ID int4, NAME varchar(20));  
SELECT AddGeometryColumn("','test','the_geom",-1,'LINESTRING',2);
```

Popolare la *table*:

```
INSERT INTO gtest (ID, NAME, the_geom) VALUES (1,'First Geometry',  
GeomFromText('LINESTRING(2 3,4 5,6 5,7 8)',-1));
```


Le basi

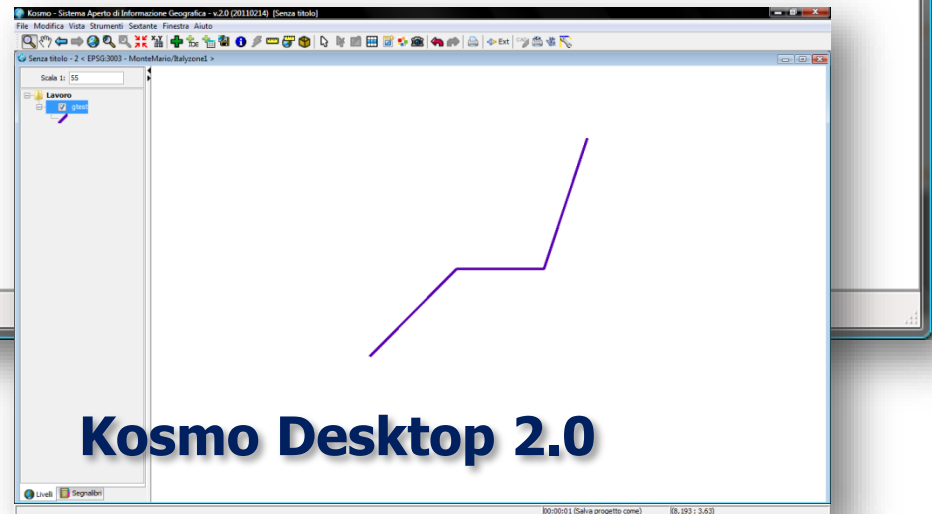
Accesso in lettura all'entità



The screenshot shows the PgAdmin III interface. The main window displays a table with the following data:

| | id integer | name character varying(20) | the_geom geometry |
|---|---------------|-------------------------------|---|
| 1 | 1 | First Geometry | 010200000004000000000000000000000400000000000000008400000000000010400000000000014400000 |

The text "PgAdmin III" is overlaid on the bottom left of the screenshot.



The screenshot shows the Kosmo Desktop 2.0 interface. The main window displays a map with a purple line. The text "Kosmo Desktop 2.0" is overlaid on the bottom of the screenshot.

Ottimizzazione delle performance

Gli indici spaziali GiST (*Generalized Search Tree*)

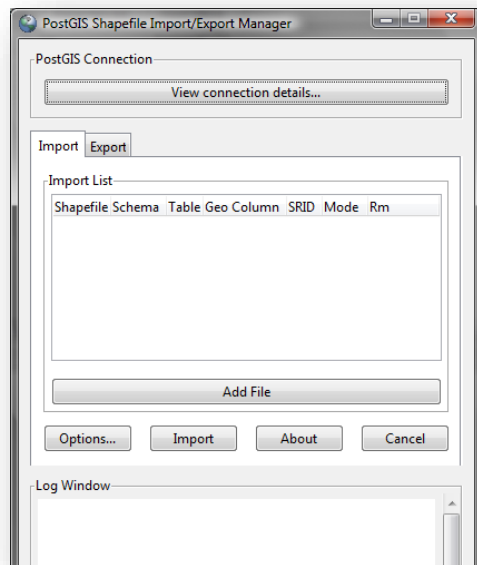
Per ottenere le migliori performance nelle interrogazioni a tabelle contenenti grandi quantità di oggetti è opportuno costruire un indice spaziale basato sulla cosiddetta “piramidizzazione” dei *bounding box*.

Creazione dell'indice per l'entità test:

```
CREATE INDEX indice ON test USING GIST (the_geom);
```

Utilizzo di una base dati PostGIS

Riversamento di un layer GIS

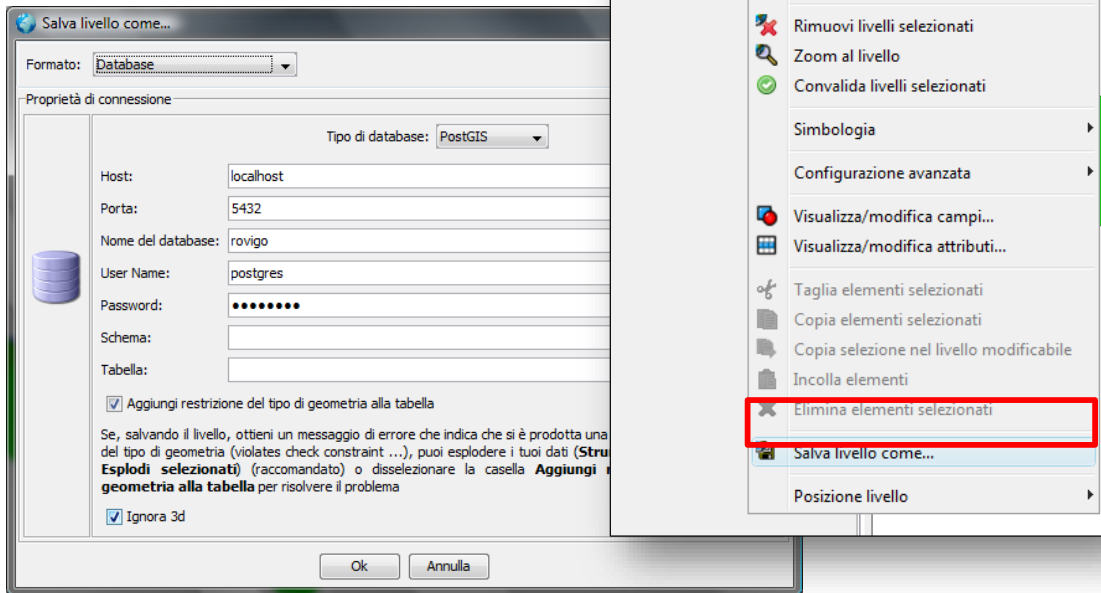


PgAdmin

Shapefile and DBF loader

Kosmo / QGIS

Salva livello come ...



Utilizzo di una base dati PostGIS

Funzioni Open GIS

- Di relazione (14) es. *distanza, contiene/contenuto, interseca ...*
- Processamento (14) es. *buffer, intersezione, differenza, unione ...*
- Accesso alle prop. (25) es. *chiuso, punto iniziale, coord X ...*
- Costruttori (23) *GeomFromText – GeomFromWKB e varianti*

Estensioni PostGIS

- Operatori (13) es. *bounding box, @, &&, ~ ...*
- Calcolo (13) es. *distanza massima, perimetro, area ...*
- Conversione (8) es. *in KML, GML, GeoJson ...*
- Costruttori (16) *come costruttori Open GIS con sintassi diverse ...*
- Editors (26) es. *aggiungi/elimina punto, scala, ruota, merge ...*
- Riferimento lineare (5) *interpola, trova punto, segmenta ...*
- Aggregazione (8), *costruttori - utilizzo nelle query di aggregazione*

Utilizzo di una base dati PostGIS

Funzioni di relazione (relationship): **ST_DWithin**

```
SELECT n.* FROM strade.grafo_nodi n, strade.comunerovigo p  
WHERE ST_DWithin(n."GEOMETRY", p."GEOMETRY", 0)
```

Selezione di tutti i record di *grafo_nodi* che ricadono all'interno dei poligoni presenti in *comunerovigo*

Il terzo argomento permette di specificare una distanza emulando un effetto *buffer*

(test: simulare una distanza di 10000m)

Utilizzo di una base dati PostGIS

Funzioni di processamento (processing): **ST_Buffer**

```
SELECT p."GID", ST_Buffer("GEOMETRY", 10000) as "GEOMETRY"  
FROM strade.comunerovigo p
```

Selezione di tutti i record di *comunerovigo* e generazione di un campo geometria *buffer* equivalente alla geometria originaria con un offset di 10000m.

(test: simulare una distanza di 5000m)

Utilizzo di una base dati PostGIS

Funzioni di accesso alle proprietà (accessors): **ST_IsClosed**

```
SELECT p.*, ST_IsClosed("GEOMETRY") as "CLOSED"  
FROM strade.comunerovigo p
```

Selezione di tutti i record di *comunerovigo* e generazione di un campo booleano che indica se punto iniziale e punto finale della geometria coincidono.

Utilizzo di una base dati PostGIS

Costruttori (constructors): **ST_PointFromText**

```
SELECT 1 as id, ST_PointFromText ('POINT(1720000 4993000)',3003) as "GEOMETRY"
```

Generazione dinamica di un livello contenente un'entità geometrica puntuale nel sistema di proiezione EPSG 3003 (Gauss Boaga fuso Ovest).

(test: modificare le coordinate)

Utilizzo di una base dati PostGIS

Operatori PostGIS (operators): &&

Overlay tra un livello di poligoni rettangolari (griglia) e semine di punti GPS

Previous queries Delete Delete All

```
SELECT g."ID", avg(s."PAV"::integer) as l
FROM strade.grid g, strade.ro_iv_pav s WHERE st_intersects(g."GEOMETRY" , s."GEOMETRY") GROUP BY g."ID"
```

Output pane X

Data Output Explain Messages History

| ID | l |
|------|----------|
| 3833 | 5.000000 |
| 3839 | 5.000000 |

Previous queries Delete Delete All

```
SELECT g."ID", avg(s."PAV"::integer) as l
FROM strade.grid g, strade.ro_iv_pav s WHERE (g."GEOMETRY" && s."GEOMETRY") GROUP BY g."ID"
```

Output pane X

Data Output Explain Messages History

| ID | l |
|------|----------|
| 3833 | 5.000000 |
| 3839 | 5.000000 |

OK. Unix Ln 2, Col 64, Ch 106 6342 row: 7854 ms

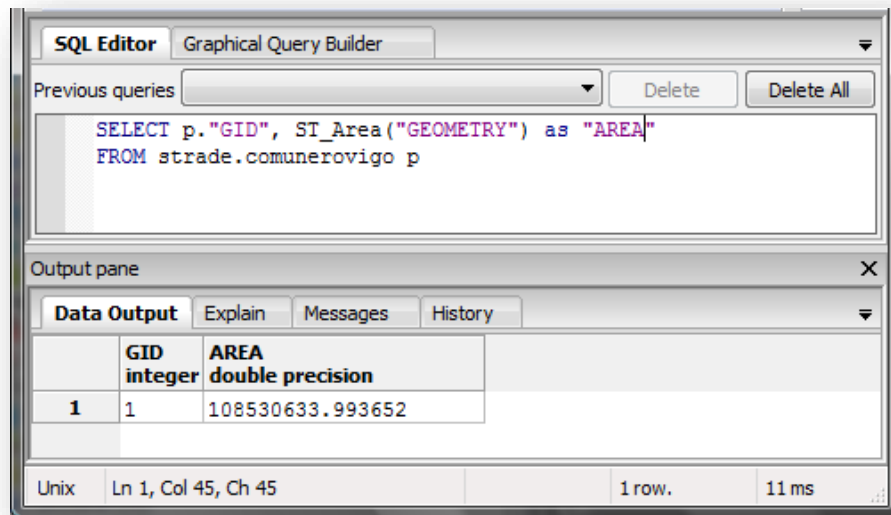
42817 ms

Utilizzo di una base dati PostGIS

Funzioni di misurazione PostGIS (measurement): [ST_Area](#)

```
SELECT p."GID", ST_Area("GEOMETRY") as "AREA" FROM strade.comunerovigo p
```

*(test: creare una vista
con campo area dinamico)*



The screenshot shows a SQL Editor window with the following content:

```
SQL Editor Graphical Query Builder
```

Previous queries: [dropdown] [Delete] [Delete All]

```
SELECT p."GID", ST_Area("GEOMETRY") as "AREA"  
FROM strade.comunerovigo p
```

Output pane: [X]

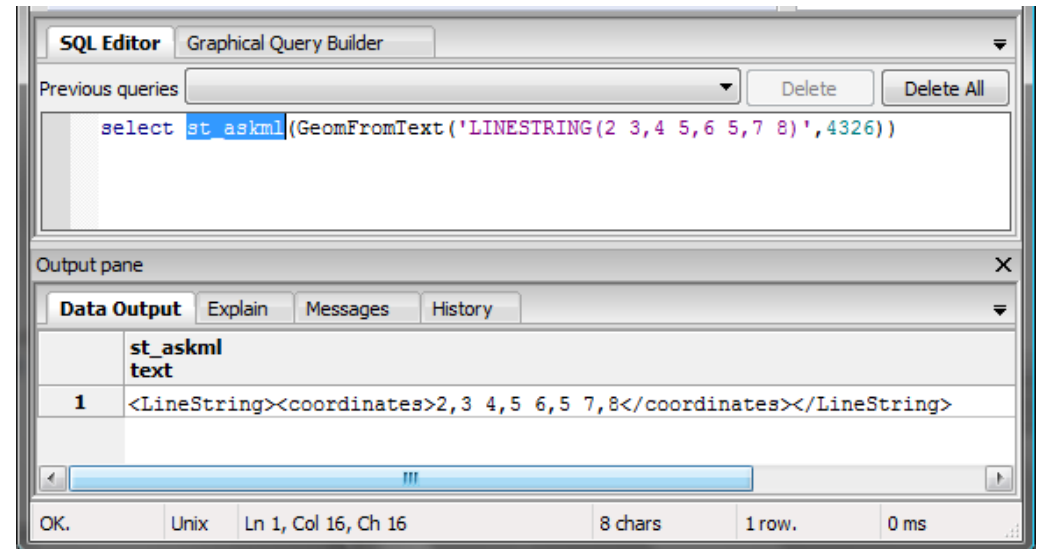
| | GID integer | AREA double precision |
|---|----------------|--------------------------|
| 1 | 1 | 108530633.993652 |

Unix Ln 1, Col 45, Ch 45 1 row. 11 ms

Utilizzo di una base dati PostGIS

Funzioni di conversione PostGIS (outputs): **ST_AsKML**

select **ST_AsKML** (GeomFromText('LINESTRING(2 3,4 5,6 5,7 8)',4326))

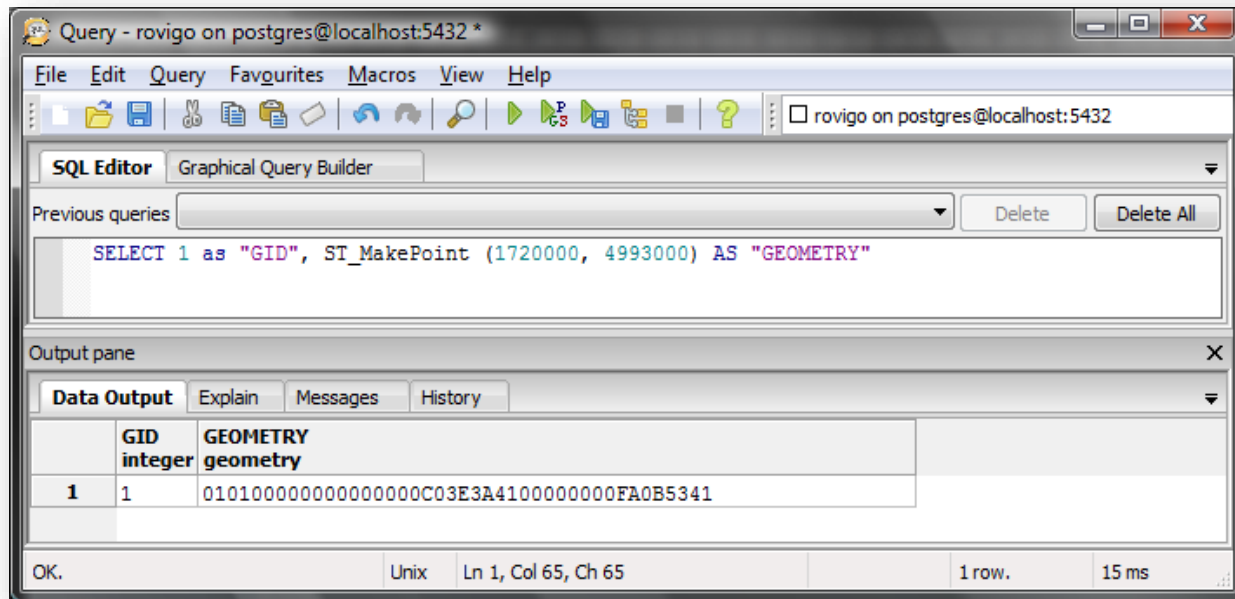


Utilizzo di una base dati PostGIS

Costruttori PostGIS (constructors): **ST_MakePoint**

SELECT 1 as "GID", ST_MakePoint (1720000, 4993000) AS "GEOMETRY"

Generazione dinamica
di un punto.



Utilizzo di una base dati PostGIS

Editors PostGIS (editors): **ST_transform**

```
SELECT p."GID", ST_Transform ("GEOMETRY", 4326) as "GEOMETRY"  
FROM strade.comunerovigo p
```

Esegue una conversione degli elementi presenti in *comunerovigo* nel sistema di proiezione 4326 (WGS84 geografico)

Utilizzo di una base dati PostGIS

Riferimento lineare PostGIS (linear referencing): [ST_line_interpolate_point](#)

```
SELECT "GID", ST_Line_interpolate_point ("GEOMETRY", 0.3) AS "GEOMETRY"  
FROM strade.grafo
```

Query che restituisce dinamicamente un livello di punti posizionati ai 3/10 di distanza dal punto iniziale delle linee contenute nel livello *grafo*.

Utilizzo di una base dati PostGIS

Operatori di aggregazione (aggregate functions):

Questi operatori si utilizzano nelle query in cui è presente l'operatore di raggruppamento GROUP BY oppure nei raggruppamenti globali

ST_Accum - costruisce un array di geometrie

ST_Collect - costruisce una collezione di geometrie (geometry collection)

ST_Extent - costruisce il bounding box di un set di geometrie

ST_Extent3D - costruisce il bounding box 3D di un set di geometrie

ST_MakeLine - costruisce una linestring.

ST_MemUnion - come ST_Union ma usa meno memoria e più tempo/processore

ST_Polygonize - costruisce una GeometryCollection di poligoni da un set di altre primitive

ST_Union - costruisce la cosiddetta point set union delle geometrie (*vedi esempio*)

```
SELECT 1 AS "GID", ST_Union ("GEOMETRY") AS "GEOMETRY" FROM strade.grafo_nodi
```


Utilizzo di una base dati PostGIS

Query complesse (1/3)

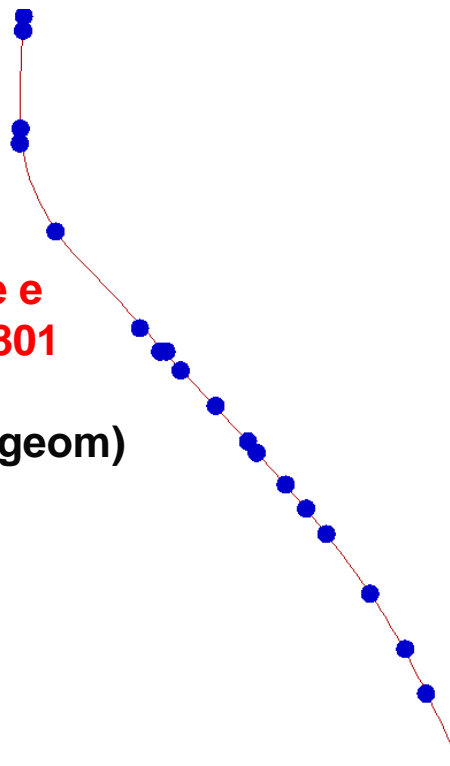
```
SELECT g.*, count (ov.value_0) as v
FROM grid g,
  (SELECT e.value_0, g.the_geom, e.idrec
   FROM a_geometry g, a_segmented_attribute e
   WHERE g.id_geo = e.id_line AND e.a_type =801
  ) ov
WHERE ST_Intersects (g."GEOMETRY", ov.the_geom)
GROUP BY g."ID"
```

Query che calcola un campo per ogni cella della griglia *grid* (*layer di rettangoli*) con il conteggio del numero di punti di una selezione da *a_segmented_attribute* rappresentanti passi carrai e accessi lungo l'asse stradale.

Utilizzo di una base dati PostGIS

Query complesse (2/3)

```
SELECT g.*, count (ov.value_0) as v
FROM grid g,
  (SELECT e.value_0, g.the_geom, e.idrec
   FROM a_geometry g, a_segmented_attribute e
   WHERE g.id_geo = e.id_line AND e.a_type =801
  ) ov
WHERE ST_Intersects (g."GEOMETRY", ov.the_geom)
GROUP BY g."ID"
```

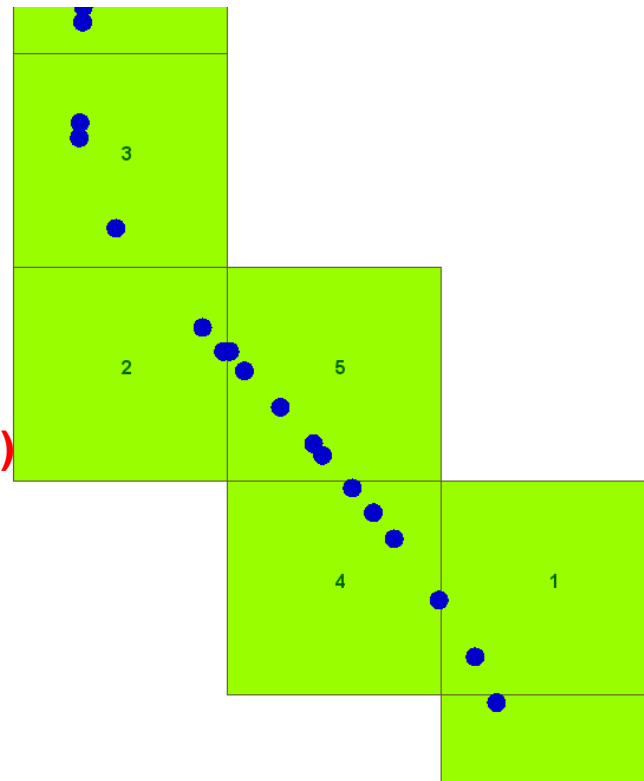


Utilizzo di una base dati PostGIS

Query complesse (3/3)

```
SELECT g.*, count (ov.value_0) as v  
FROM grid g,  
  (SELECT e.value_0, g.the_geom, e.idrec  
    FROM a_geometry g, a_segmented_attribute e  
    WHERE g.id_geo = e.id_line AND e.a_type =801  
    ) ov  
WHERE ST_Intersects (g."GEOMETRY", ov.the_geom)  
GROUP BY g."ID"
```

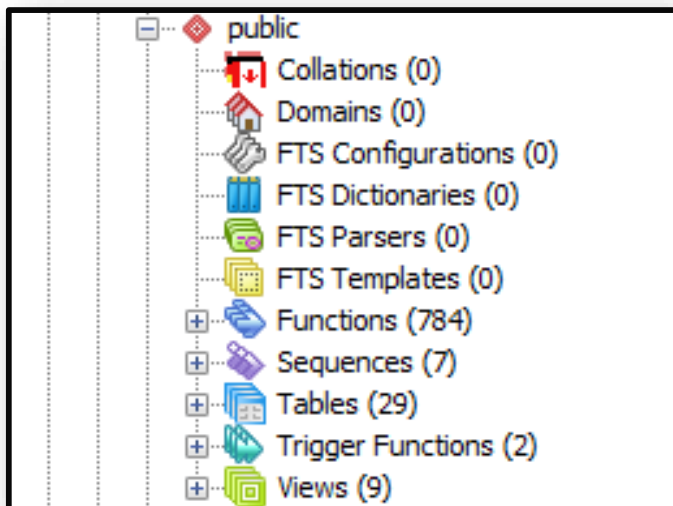
(test: modificare a_type=101)



Struttura di un progetto complesso

GeoDatBase per la gestione dell'indicatore geografico di rischio sulla rete stradale della Provincia di Rovigo

Schema «public»



Schema «strade»

