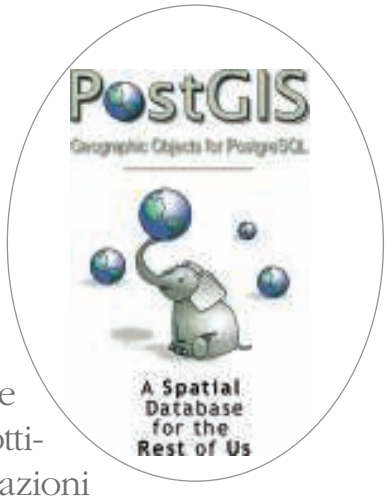
PostGIS: il database geografico Open Source

di Maurizio Napolitano, Emilia Venturato



Il database geografico, o geodatabase, è ormai una componente essenziale per le applicazioni GIS più complesse, in particolare nei gruppi di lavoro grandi ed eterogenei. In questo campo, la soluzione Open Source più potente ed affidabile è PostGIS (www.postgis.org), un'estensione del database relazionale PostgreSQL per l'archivio e la gestione dei dati geografici. È un database libero che, fondandosi su standard OGC, si pone come ottima soluzione per chi ricerchi l'interoperabilità tra applicazioni desktop, analitiche e Web nel campo del GIS. Permette l'accesso multiutente a grandi moli di dati, sia geografici che alfanumerici e la gestione delle relazioni tra di essi, garantendone l'integrità. Permette la gestione dei dati tridimensionali e si incammina verso quella dei raster.

Il GIS è per definizione basato su un database di tipo geografico. Lo schema classico delle applicazioni desktop-GIS ci ha abituati ad avere un unico strumento pensato sia per la visualizzazione dei dati che per la relativa archiviazione su file. Nel caso dei vettoriali uno dei modelli più utilizzati (ad esempio il formato ESRI Shape file) è quello in cui i dati vengono divisi in tre parti (e rispettivi file): una tabella per le geometrie (ad esempio file .shp), una per gli attributi (ad esempio file .dbf) ed una per gli indici delle relazioni fra le prime due (ad esempio file .shx).

Questo tipo di schema può dimostrarsi vincente per risolvere scenari d'uso in cui l'interazione con l'applicazione sia affidata ad un unico utente ma, quando gli utenti che accedono alla base dati sono più di uno, si sente la necessità di rivolgersi a sistemi di database più evoluti quali i DBMS (Database Management System).

I DBMS forniscono un'interfaccia verso il database che permette l'accesso multiutente a grandi quantità di dati garantendone l'integrità. L'interfaccia fondamentale al database è il linguaggio SQL (Structured Query Language), un linguaggio standard, semplice da apprendere che permette di effettuare operazioni di selezione, inserimento, modifica, calcolo, applicazione di funzioni, ecc., su una banca dati costituita da tabelle in relazione tra loro. Per chi avesse interesse ad approfondire l'argomento, è disponibile una vasta documentazione sia su Internet che su riviste e libri (per un primo approccio all'argomento può essere utile la lettura del libro di Riccardo Cervelli: "SQL - Imparare in 6 ore" ed. Tecniche nuove www.eclecticforce.com).

L'architettura di un database geografico si distingue da quella dei database classici per il tipo di dato che deve gestire: il dato geografico. Pertanto, la prima cosa richiesta ad un DBMS che volesse andare in questa direzione era una struttura in grado di gestire questa tipologia di dati. Era necessario, non solo definire la struttura del dato geografico, ma anche arricchire il linguaggio SQL con funzionalità in grado di calcolare la distanza fra due punti o l'area di un poligono su uno spazio bidimensionale e/o tridimensionale, oppure di effettuare la conversione da un sistema di riferimento ad un altro, ecc. Questo ha potenziato enormemente anche le possibilità offerte nell'incrocio con dati alfanumerici. L'Open Geospatial Consortium (precedentemente chiamato Open GIS Consortium; (www.opengis.org), ha elaborato - grazie alla collaborazione di 5 grandi aziende: due nel settore GIS (ESRI e MapInfo) e tre in quello database (IBM, Informix e Oracle) - un documento relativo agli standard per i formati dei dati geografici, pubblicato nel maggio del 1999.

Il documento dell'OGC (www.opengeospatial.org/docs/99-049.pdf) definisce:

- quali tabelle debbano essere sempre presenti in un DBMS con estensioni geografiche (tabella delle geometrie e tabella dei sistemi di riferimento);
- la tipologia di geometrie da archiviare denominate Simple Features (point, line, polygon, ecc.) offrendo due possibili schemi (Fig. 1):

uno testuale (che occupa maggior spazio ma di facile lettura) denominato WKT (Well Know Text); uno binario (con caratteristiche opposte al primo) denominato WKB (Well Know Binary);
- l'insieme delle istruzioni SQL di analisi geografica.

Questo documento è divenuto il punto di riferimento per le estensioni delle funzioni SQL Spatial di qualsiasi DBMS.

PostGIS e PostgreSQL

Nel 2001 l'azienda canadese Refractions di Victoria, British Columbia (www.refractions.net), specializzata nello sviluppo di applicazioni GIS, ebbe l'esigenza di fare uso di un DBMS con estensioni geografiche. A questo scopo avviò un'indagine delle soluzioni presenti sul mercato e, dopo aver riscontrato che tutte le alternative nei sistemi proprietari erano, o estremamente costose, o particolarmente lente, o prive di possibilità di espansione o, in alcuni casi, una combinazione di queste caratteristiche, decise di creare un software ex novo.

Costruire un DBMS ex novo in breve tempo non era ragionevole; Paul Ramsey (capo progetto della Refractions) aveva già le idee chiare: orientarsi verso il mondo del software libero con tutti i vantaggi che ne compete e creare una estensione GIS ad un DBMS già esistente. La scelta cadde su PostgreSQL che, oltre a vantare anni di sviluppo ed una grossa comunità, permetteva (e permette tuttora) ad un programmatore di creare velocemente nuove estensioni. La nuova applicazione fu chiamata PostGIS. Per ulteriori notizie si rimanda ad un'intervista effettuata da Webbit (www.webbit.it) a Paul Ramsey (www.webbit.it/html/webbit05/docosservatorio/interviste/07_PaulRamsey.rtf).

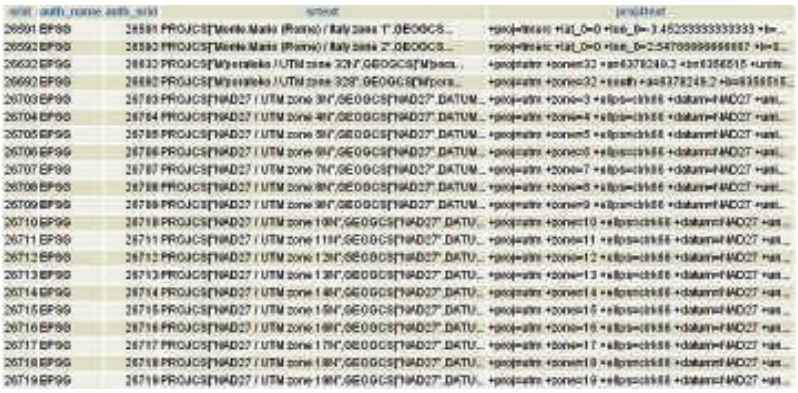
Inoltre, come risulta dal documento di valutazione dei database Open Source fatto da Fabalabs (www.fabalabs.org/research/papers/FabalabsResearchPaper-OSDBMS-Eval.pdf), PostgreSQL offre alcune caratteristiche molto interessanti:
- si tratta di uno dei pochi DBMS in grado di supportare un numero molto grande di tipi di dati binari e testuali;
- permette di gestire database molto grandi anche su sistemi operativi con capacità limitate;
- offre funzionalità quali la replicazione dei dati, load balancing, uso di multiprocessori e cluster;
- ha diverse modalità di autentificazione crittografate.

Per numerosi anni lo sviluppo di PostgreSQL si è concentrato verso i sistemi operativi UNIX-like (Solaris, Linux, HPUX, FreeBSD, ecc.) anche se era disponibile su Windows attraverso cygwin (un ambiente Linux-like per Windows). A partire dalla versione 8.0 (recentemente rilasciata) PostgreSQL gira in maniera nativa sui sistemi Microsoft (a partire da Windows 2000).

La scelta, effettuata da Refractions, di fare di PostGIS un software aperto ne ha permesso la crescita esponenziale al punto da ricoprire il 100% delle funzionalità SQL Spatial indicate dal documento dell'OGC. Negli anni si è formato un gruppo di sviluppo molto attivo (da qualche anno anche un italiano Sandro Santilli - è entrato tra gli sviluppatori in maniera talmente attiva da essere stato assunto dalla Refractions stessa) che ha portato ad importanti innovazioni. È da sottolineare inoltre la collaborazione con l'azienda canadese VividSolutions per l'integrazione di un importante pacchetto (Java Topology Suite) per l'analisi di dati vettoriali.

Figura 1 - Nel formato WKT sono riportati: lo SRID (che indica il sistema di coordinate, il tipo di geometria e la sequenza delle coordinate dei punti della forma geografica). Nel formato WKB invece i dati sono in formato binario (vedi oltre per maggiori dettagli)

Figura 2 - La figura mostra una selezione dei 2671 SRID disponibili nella spatial_ref_sys



Setup di PostGIS

Tutte le distribuzioni GNU/Linux dispongono ormai di un'ottima pacchettizzazione per l'installazione indolore di entrambe le applicazioni. Qualora si dovesse compilare PostGIS (la pacchettizzazione rende questa eventualità sempre più lontana) consigliamo vivamente di abilitare il supporto per GEOS o JTS (le librerie rispettivamente C++ e Java necessarie per avere tutte le funzionalità OGC) e PROJ (il supporto per la trasformazione da un sistema di coordinate ad un altro).

Per gli utenti Windows è disponibile un installer creato dal team di sviluppo del progetto DC Maintenance Management System (un WebGIS di analisi di tubature idriche). Per maggiori performance consigliamo comunque di utilizzare PostGIS su un server GNU/Linux.

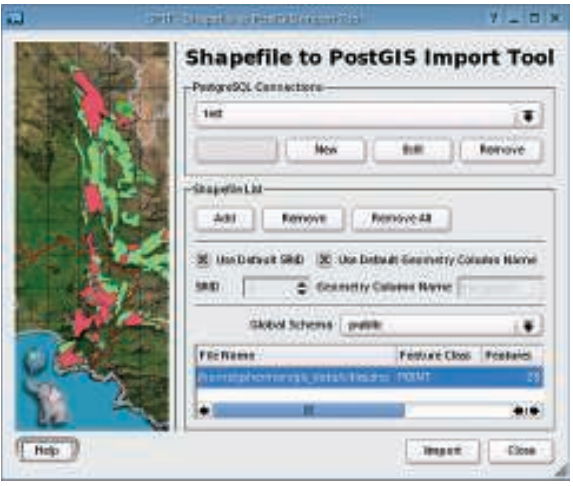
Una volta terminata l'installazione di PostgreSQL e PostGIS sono necessarie alcune operazioni, da parte dell'amministratore del DBMS, per poter abilitare l'uso delle estensioni GIS ad uno o più database. I passi da seguire sono

ben documentati sul sito ufficiale di PostGIS (http://postgis.refractions.net). Si tratta di eseguire una serie di operazioni da linea di comando per attivare le funzioni geografiche.
Vediamo nei dettagli quali sono le operazioni da seguire:

- aggiungere al database il linguaggio di programmazione plpgsql (linguaggio utilizzato per moltissime altre funzioni del database, chi usa già PostgreSQL è probabile che lo abbia già attivato per altre funzioni):
  `createlang plpgsql nomedatabase`
- importare le tabelle di supporto per le geometrie:
  `pgsql -f postgis.sql-d nomedatabase`
- importare le tabelle di supporto per i sistemi di riferimento (tabella EPGS definita dalle librerie PROJ; http://proj.maptools.org):
  `pgsql -f spatial_ref_sys.sql -d nomedatabase`.

Nel caso in cui le funzioni geografiche debbano essere attivate solo su un determinato database è necessario compiere i passi descritti, direttamente sul database interessato.
Nel caso in cui di preveda di creare più database geografici è consigliabile attivare le funzioni sul template1 (ossia sul database che viene usato da PostgreSQL come modello per la creazione di ogni nuovo database), oppure creare un modello apposito (chiamandolo ad esempio "template_gis") da richiamare tutte le volte che si voglia creare un nuovo database geografico, mantenendosi così la possibilità di creare di default database non geografici. Il fatto di lavorare sui template, piuttosto che sul database direttamente, permette di attivare le funzioni geografiche una sola volta.
Per facilitare l'installazione, alcune distribuzioni (in particolare Debian che ha un ottimo coordinamento per la pacchettizzazione delle applicazioni GIS, svolto tramite l'apposito progetto DebianGIS) durante l'installazione creano direttamente un template_gis con tutte le funzioni geografiche attivate, sollevando l'amministratore di database dal dover compiere a mano i passi di cui sopra.
Una volta finita l'installazione la gestione del database si può effettuare sia da riga di comando che dai numerosi client grafici o web disponibili; ad esempio PgAdmin (www.pgadmin.org), PhpPgAdmin (http://phppgadmin.sourceforge.net) e Rekall (www.rekallrevealed.org).

Figura 3 - Il plugin SPIT di PostGIS



Struttura e popolazione dei dati

Una volta applicate le estensioni PostGIS ad un database si ha a disposizione una struttura dati per l'archiviazione di geometrie (e quindi di dati vettoriali) secondo le specifiche Simple Features (come visto precedentemente).
Queste specifiche permettono l'archiviazione di 7 tipi diversi di dato:
POINT (punto), LINESTRING (linea), POLYGON (poligono), MULTYPOINT (collezione di punti), MULTILINESTRING (collezione di linee), MULTYPOLYGON (collezione di poligoni), GEOMETRYCOLLECTION (collezione di geometrie di vario tipo).
Ogni tipo ha una propria sintassi che permette l'archiviazione delle coordinate sia su piani bidimensionali (quindi x e y) sia su piani tridimensionali (x, y e l'altitudine z).
Per esempio, nel caso di un POINT:
coordinate piano 2D
`POINT (0 0)`
coordinate piano 3D
`POINT (0 0 0)`
Ogni record di dati viene arricchito dal "prefisso" SRID (Spatial Reference ID), variabile che indica il sistema di riferimento nel quale sono archiviate le geometrie. Il valore attribuito a SRID è scelto tra quelli presenti nella tabella spatial_ref_sys (Fig. 2), create di default in ogni database geografico. In caso di SRID ignoto si usa l'indice -1.
Riportiamo qui di seguito un esempio di archiviazione delle coordinate di un punto su un piano bidimensionale secondo il sistema di riferimento Gauss-Boaga ovest:
`SRID=26591,POINT(2455,44334356565,32).`
La forma dell'archivio dei dati lascia subito intendere un aspetto molto interessante delle funzionalità SQL implementate da PostGIS: la possibilità di calcolare misure in ambiente sia bidimensionale che tridimensionale e in relazione al sistema di riferimento scelto. Per ottimizzare le performance del database è utile associare, alla tabella che contiene i dati GIS, un indice di tipo GiST (Generalized Search Tree).
Riportiamo qui un esempio di creazione dell'indice:
`CREATE INDEX nome_indice ON nome_tabella USING GIST (colonna_geometrie GIST_GEOMETRY_OPS).`
Si tratta di un comando SQL pertanto l'istruzione va eseguita o da una shell postgres o da un tool di amministrazione di postgres che permetta l'esecuzione di comandi SQL. Alcuni client grafici o Web facilitano l'uso del database aiutando l'utente nella scrittura dell'SQL o, in alcuni casi, permettendogli di fornire comandi in modo grafico ed traducendoli in linguaggio SQL da inviare al database. Questo vale per la creazione degli indici ma anche la maggior parte delle funzioni del database.
L'inserimento dei dati geografici può avvenire secondo due modalità:

- creazione di comandi SQL secondo la sintassi Simple Features;
- importazione dei dati attraverso tool testuali o grafici.

Nel caso in cui si utilizzi direttamente un'istruzione SQL occorre prima creare una tabella secondo la sintassi classica.
`CREATE TABLE punti (ID int4, DESCRIZIONE varchar(40));` e aggiungere a questa il campo corrispondente alla colonna dei dati geografici secondo questa sintassi:

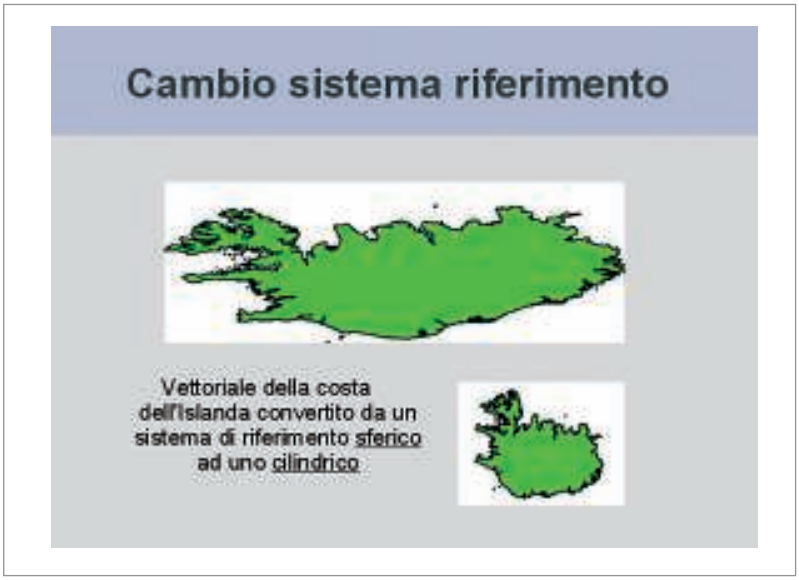


Figura 4 - La funzione SELECT TRANSFORM (geometry, integer) converte una geometria da un sistema di riferimento ad un altro

```
SELECT AddGeometryColumn ('nome_tabella', 'colonna_ geometrie', srid, 'tipo_geometria', dimensione);
```

dove:

`nome_tabella` = nome della tabella a cui applicare le geometrie;

`colonna_geometrie` = nome del campo che dovrà contenere il dato geometrico;

`srid` = id della tabella con i sistemi di riferimento (se ignoto: -1);

`tipo_geometria` = uno dei 7 valori di simple features (POINT, LINESTRING, ecc.);

`dimensione` = 2 per le geometrie bidimensionali, 3 per quelle tridimensionali.

L'inserimento delle singole geometrie avviene poi, secondo le classiche istruzioni SQL di INSERT in cui, l'istruzione legata alla geometria segue lo schema WKT.

Esempio:

```
INSERT INTO nome_tabella (colonna_geometrie, descrizione) values ('SRID=-1,POINT(0 0 0)', 'Punto inserito').
```

In alternativa alle istruzioni SQL, PostGIS offre il tool di importazione di file in formato ESRI shapefile "shp2pgsql" che crea in automatico tutto l'SQL necessario. In questo caso, se non definita dall'utente, la colonna con le geometrie prende di default il nome "the_geom". Soluzioni alternative sono rappresentate da software quali ogr2ogr o il plugin SPIT (Fig. 3) per QuantumGIS (software GIS per il desktop, molto simile ad ArcView); in quest'ultimo caso l'importazione di shapefile in PostGIS avviene esclusivamente da interfaccia grafica e non necessita di alcuna conoscenza né del linguaggio SQL né di scrittura da riga di comando.

Le query PostGIS

Per descrivere tutte le query possibili con PostGIS e le combinazioni che si possono ottenere grazie alle potenzialità di PostgreSQL/PostGIS non sarebbe sufficiente un intero numero della rivista. Ci limitiamo quindi a riassumere le funzioni di maggior interesse. Le funzionalità PostGIS sono classificabili in categorie:

Funzioni di base - Permettono la creazione/eliminazione di colonne geometriche e l'attribuzione dei dati ad un determinato sistema di riferimento.

Funzioni di relazioni fra geometrie - Variano dal calcolo della distanza fra due geometrie, alla verifica della eventuale sovrapposizione, intersezione, inclusione, ecc. tra forme geografiche distinte.

Funzioni di calcolo sulle geometrie Permettono di calcolare area, perimetro, buffer, centroide, ecc., di una data geometria. Permettono inoltre di effettuare operazioni di unione e sottrazione tra geometrie.

Funzione di "informazioni" sulle geometrie - Attraverso di esse è possibile conoscere il tipo di geometria presente in un dato campo, l'id del sistema di riferimento utilizzato, le coordinate dell'ultimo punto di una geometria, il numero di punti contenuti, il valore x o y o z di un dato record, ecc. Permettono inoltre di visualizzare i dati geografici in formato WKB quando siano archiviati in WKT, e viceversa (rendendo, ad esempio, intellegibili i dati geografici archiviati in formato binario).

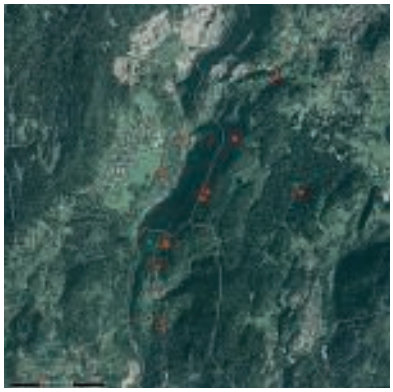
Funzioni di creazioni di geometrie - Permettono di creare dati geometrici e di archiviarli in formato standard OGC a partire da un insieme di coordinate. Questo permette di trasformare in un database geografico un database alfanumerico in cui le coordinate dei punti siano, ad esempio, archiviate in due campi 'x' e 'y'.

Oltre alle funzioni sopra descritte, e che fanno parte di quelle standard OpenGIS, PostGIS ha numerose funzioni avanzate. Ne riassumiamo per brevità alcune tra le più importanti.

Funzioni di calcolo di misure - Permettono di calcolare le aree, i perimetri, le lunghezze, gli sferoidi, ecc., in relazione al sistema di riferimento e al tipo di piano (bidimensionale o tridimensionale).

Funzioni di output - Attraverso queste funzioni è possibile trasformare le geometrie dal formato WKT al formato WKB (e viceversa), esportarle verso for-

Figura 5 - Il risultato della query: SELECT nome FROM paesi WHERE the_geom && BOX3D (3 4, 4 5): box3d; restituisce i valori del campo nome dove le geometrie sovrappongono (&&) un'area rettangolare di coordinate 3 4, 4 5



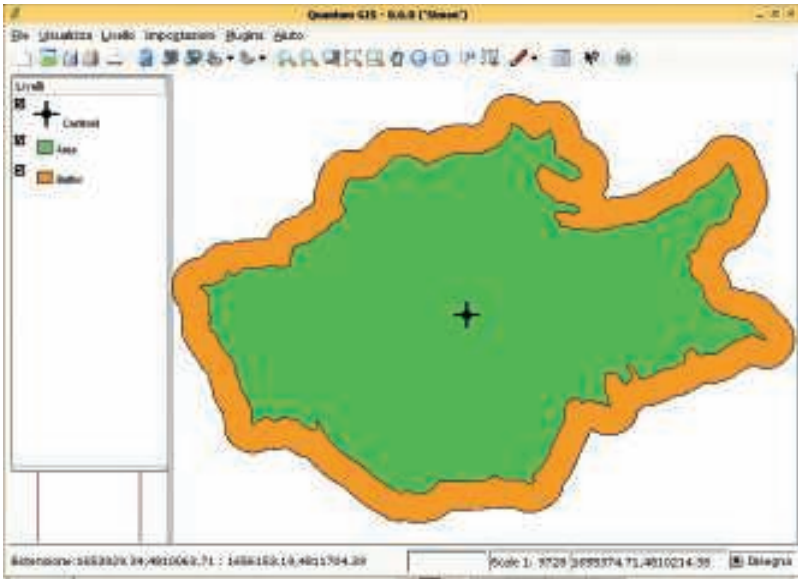


Figura 6 - Un esempio di uso semplice di QuantumGIS e PostGIS: uno shapefile (area verde), importato tramite il plugin SPIT di QuantumGIS nel database, viene visualizzato tramite QuantumGIS, insieme al risultato del calcolo di un buffer (arancio) e del centroide (croce), effettuati in PostGIS

mati XML quali l'SVG (Scalable Vector Graphics; un XML per la creazione di geometrie su browser definito dal W3C) o il GML (Geographics Markup Language; un XML, definito dall'OGC, per l'archiviazione di dati GIS).

Funzioni di modifica delle geometrie - Permettono di manipolare le geometrie effettuando ad esempio operazioni di semplificazione (tramite l'algoritmo Douglas-Peuker), di traslazione, di inversione, di conversione da un sistema di riferimento ad un altro (Fig. 4), ecc. Oltre a queste ci sono funzioni che forniscono informazioni strettamente legate a PostGIS; ad esempio permettendo di verificarne la versione, le opzioni (ad esempio se la versione in uso offre il supporto GEOS e/o PROJ), ecc. Numerosi operatori logici infine (di eguaglianza, sovrapposizione, ecc.) permettono di effettuare confronti sia tra le geometrie che tra i risultati di query anche complesse (Fig. 5). Queste e numerose altre informazioni si possono trovare tra la documentazione ufficiale del programma (http: //postgis.refractions.net/docs).

Software per PostGIS

L'uso diretto dell'SQL può risultare scomodo per un utente poco esperto, così come, in alcuni casi, anche per un utente più avanzato. Ma PostGIS è un prodotto della comunità del software libero, e tutti i programmi GIS Open Source possono interfacciarsi con esso, leggere direttamente i dati dal database e archiviarvi i risultati delle proprie elaborazioni. In particolare, software che si interfacciano senza particolari complicazioni con PostgreSQL/PostGIS sono GRASS (http://grass.itc.it), QuantumGIS (http://qgis.sourceforge.net; Fig. 6), uDIG (prodotto dalla stessa Refractions; http://udig.refractions.net), UMN Mapserver (http://mapserver.gis.umn.edu), JUMP (http://jump-pilot.sourceforge.net), GEOserver (http://geoserver.sourceforge.net/html), OGR (www.gdal.org/ogr). Sul fronte dei sistemi proprietari, le aziende Safe Software e CardCorp SIS sviluppano software per la lettura e scrittura di questa tipologia di dati oltre che applicazioni Desktop in grado di interfacciarsi con esse. Un gruppo di sviluppatori sta lavorando ad una componente per il collegamento anche verso applicazioni ESRI.

Supporto e sviluppi futuri

La comunità di utenti che ruota intorno a PostGIS cresce nel tempo sempre più velocemente. La mailing-list dedicata agli utenti ha un buon traffico di posta con richieste di aiuto che trovano spesso risposta in messaggi inviati dagli sviluppatori stessi. Questo, oltre a creare un buon supporto, favorisce la stesura di una ricca documentazione utile per gli utenti (sono inoltre in corso alcuni progetti finalizzati alla traduzione di questa documentazione dall'inglese verso altre lingue).

Qualora un utente non si sentisse sufficientemente supportato dalla documentazione disponibile e dalla mailing-list, la stessa Refractions offre un supporto tecnico a pagamento.

L'estesa comunità di persone che usano e sviluppano PostGIS, le grandi potenzialità di un database geografico così completo, basato su standard internazionali e sulla solidità di PostgreSQL, la facilità di uso come banca dati geografica per numerosi software GIS anch'essi in forte evoluzione, sono fattori che spingono ad essere molto ottimisti sul futuro di PostGIS. Gli ultimi sviluppi di PostgreSQL, che hanno portato verso un'integrazione con i server Windows, hanno inoltre ampliato ulteriormente le potenzialità di diffusione di queste applicazioni.

Nella lista dei desideri capeggia ormai da tempo la possibilità di usare PostGIS anche per i dati raster. Non è escluso che questo accada nei prossimi mesi, visto che gli sviluppatori sono intenzionati a procedere in questa direzione.

Maurizio Napolitano
ITC-irst - Divisione SRA
Via Sommarive 18 - Povo (TN)
http://sra.itc.it

Emilia Venturato
Faunalia
Piazza Garibaldi 5 - Pontedera (PI)
www.faunalia.it