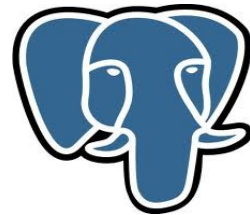


# PostGIS

L'estensione geografica a PostgreSQL



# Argomenti trattati

## ◆ Presentazione PostGIS

### ◆ **Introduzione**

- PostgreSQL
- OpenGIS Consortium (OGC)

### ◆ PostGIS

- Oggetti geometrici
- Oggetti geografici
- Funzioni
- SQL
- Applicazioni WEB



## ◆ Obiettivi

- ◆ Fornire una conoscenza di base su **PostGIS** dal punto di vista funzionale
- ◆ Presentare le principali interrogazioni SQL su oggetti PostGIS
- ◆ Fornire una panoramica sugli strumenti Open Source piu' comunemente utilizzati con PostGIS
- ◆ Storia di PostGIS e... futuro!



# Introduzione

- ◆ PostGIS e' un estensione al DBMS object-relational Open Source PostgreSQL che consente di memorizzare sulla base dati oggetti di tipo GIS (Geographic Information System).
- ◆ PostGIS implementa ed estende le funzioni definite dallo standard OGC.
- ◆ PostGIS supporta oggetti geometrici e geografici, indici spaziali GiST ed un numero molto elevato di funzioni per l'analisi e il processing di oggetti GIS.



# PostgreSQL

- ◆ PostgreSQL e' DBMS relazionale Open Source piu' avanzato al mondo e non ha nulla da invidiare a sistemi commerciali. I suoi principali punti di forza sono:
  - Free! Gratis e distribuito con una licenza molto libera
  - Funzionalita' molto ampie (eg. Object Oriented, GIS, ...)
  - Robustezza ed integrita' dei dati
  - Notevole diffusione
  - Un ottimo e completo SQL utilizzabile direttamente e con i piu' diffusi linguaggi di programmazione
  - Fornito anche con una licenza commerciale ed con supporto
  - Distribuito con Linux, Unix ed anche WinX



# PostgreSQL

◆ PostgreSQL risulta particolarmente vantaggioso perche':

- Buone prestazioni
- Ricco di funzionalita'
- Robustezza ed integrita' dati: full ACID, referential integrity, object-oriented
- Semplice nell'utilizzo, nell'amministrazione e per la programmazione...
- Object-Oriented Relational Database Management System



# PostgreSQL e PostGIS

- ◆ Le caratteristiche object-oriented di PostgreSQL consentono di estenderlo in modo semplice. Molte funzioni PostGIS sono implementate in **plpgsql**.
- ◆ L'installazione di PostGIS avviene con i classici download, untar, ./configure, make, make install. Oppure... rpm!
- ◆ Per configurare su un database PostgreSQL le funzionalità PostGIS è sufficiente lanciare lo script psql **postgis.sql**. Oppure... utilizzare un template!



# PostgreSQL e PostGIS

- ◆ PostGIS introduce alcuni data type (eg. geometry e geography), due meta-tabelle e le funzioni (oltre 500) per gestire gli oggetti GIS. L'utilizzo di PostGIS e'...  
SQL:

```
CREATE TABLE testgg(gid serial PRIMARY KEY, name  
  varchar(16), the_geog geography(POINT,4326) );
```

...

```
SELECT ST_Distance(c1.the_geog, c2.the_geog)  
FROM testgg c1, testgg c2  
WHERE c1.name='Torino' and c2.name='Milano'
```





# PostgreSQL e PostGIS

- ◆ Per rispettare lo standard OGC il datatype geometry deve essere aggiunto in modo esplicito ad una tabella:

```
CREATE TABLE testgm (ID int4, NAME varchar(64) );
```

```
SELECT AddGeometryColumn('', 'testgm', 'geom', -1,  
    'LINESTRING',2);
```

```
INSERT INTO testgm (ID, NAME, GEOM)  
VALUES ( 1, 'Esempio geometrico',  
    ST_GeomFromText('LINESTRING(2 3,4 5,6 5,7 8)',-1));
```



# OGC

- ◆ L'OGC (OpenGis Consortium) e' un organismo che definisce gli standard per i sistemi GIS. Le specifiche OpenGIS prevedono due modi standard per definire spatial objects: Well-Known Text (WKT) e Well-Known Binary (WKB).
- ◆ PostGIS implementa tutte le “Simple Features for SQL” di OGC. PostGIS implementa tutti i tipi di oggetti, tutte le funzioni e le due meta-tabelle standard (SPATIAL\_REF\_SYS, GEOMETRY\_COLUMNS). PostGIS estende lo standard per gestire coordinate 3DZ, 3DM, 4D e trattare come embedded l'SRID (spatial referencing system identifier). Per motivi prestazionali, di default PostGIS non controlla la validita' degli oggetti, ma e' possibile farlo in automatico con un trigger (usando la funzione ST\_IsValid).



# OGC - WKT

Esempi di GEOMETRY in WKT:

```
POINT(0 0)
```

```
LINESTRING(0 0,1 1,1 2)
```

```
LINEARRING(0 0,1 1,1 2)
```

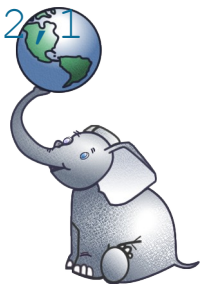
```
POLYGON((0 0,4 0,4 4,0 4,0 0),(1 1,2 1,2 2,1 2,1 1))
```

```
MULTIPOINT(0 0,1 2)
```

```
MULTILINESTRING((0 0,1 1,1 2),(2 3,3 2,5 4))
```

```
MULTIPOLYGON(((0 0,4 0,4 4,0 4,0 0),(1 1,2 1,2 2,1 2,1 1)),((-1 -1,-1 -2,-2 -2,-2 -1,-1 -1)))
```

```
GEOMETRYCOLLECTION(POINT(2 3),LINESTRING(2 3,3 4))
```



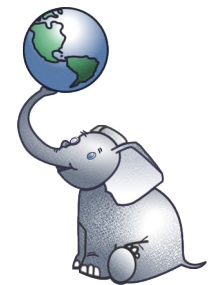
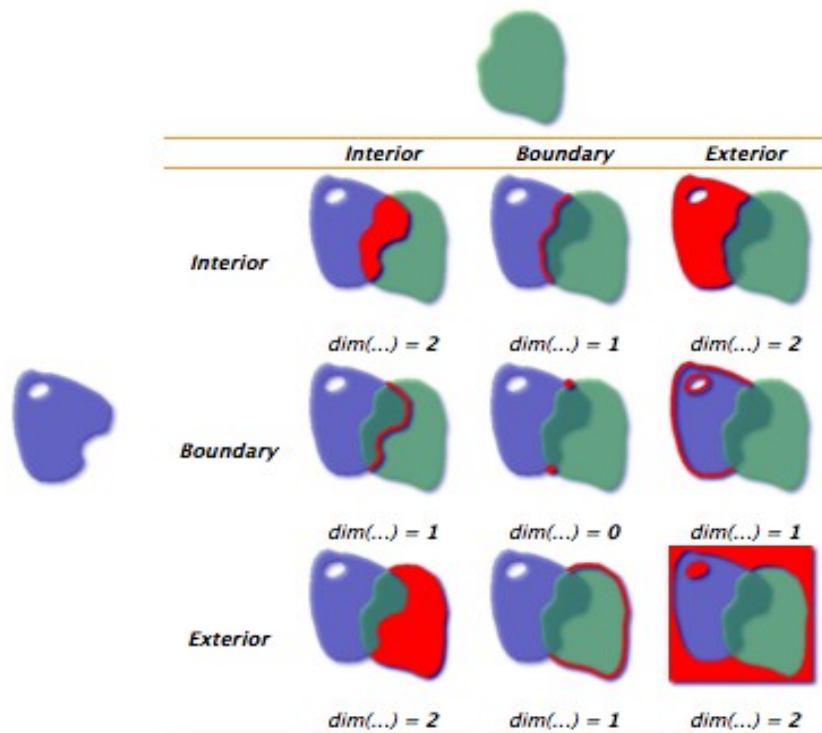
# Sistemi di coordinate

- ◆ Esistono parecchi sistemi di coordinate differenti...
- ◆ In realta' sono necessari due elementi per definire un sistema completo: il riferimento (datum) ed il sistema di coordinate.
- ◆ Il sistema di coordinate piu' diffuso e' l'UTM: Universal Transverse Mercator. Per i poli si utilizza l'UPS: Universal Polar Stereographic, ...
- ◆ Tra i datum piu' utilizzati vi sono il WGS84 (World Geodetic System 1984, utilizzato dai GPS), NAD83, NAD27, ED50, ...
- ◆ PostGIS consente la conversione tra tutti i sistemi di coordinate descritti nella meta-tabella SPATIAL\_REF\_SYS (circa 3000).
- ◆ Il sistema di coordinate utilizzato per gli oggetti geografici e' WGS84 (con SRID=4326).



# Intersezione di oggetti

- La definizione di intersezione di oggetti non e' banale. La funzione `ST_Crosses` controlla solo se due linee hanno un punto in comune, si puo' risolvere con un'intersezione ma i casi diventano molti... L'OGC utilizza DE-9IM con la funzione `ST_Relate`. Per due linee la matrice di dimensioni 212101212:



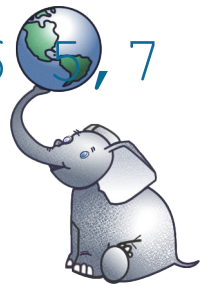
# PostGIS - Oggetti geometrici

```
CREATE TABLE testgm (ID int4, NAME  
varchar(64) );
```

```
SELECT AddGeometryColumn('', 'testgm',  
'geom', -1, 'LINESTRING',2);
```

```
CREATE INDEX testgm_gix ON testgm USING  
GIST ( geom );
```

```
INSERT INTO testgm (ID, NAME, GEOM)  
VALUES ( 1, 'Esempio geometrico',  
ST_GeomFromText('LINESTRING(2 3,4 5,6 7  
8) ',-1));
```



# PostGIS - Oggetti geografici

```
CREATE TABLE citta (  
    id SERIAL PRIMARY KEY,  
    name VARCHAR(64),  
    location GEOGRAPHY (POINT,4326) );  
CREATE INDEX citta_gix ON citta USING GIST ( location );
```

```
INSERT INTO citta (name, location) VALUES ('Torino',  
    ST_GeographyFromText ('SRID=4326;POINT(45.07055 7.686653)') );
```

```
INSERT INTO citta (name, location) VALUES ('Parigi',  
    ST_GeographyFromText ('SRID=4326;POINT(48.856 2.352)') );
```

```
# Distanza da un'ortodromica (cfr. http://gc.kls2.com/cgi-bin/gc?PATH=MXP-JFK)
```

```
SELECT
```

```
ST_Distance ('LINESTRING (45.631 8.724, 40.644 -73.782) '::geography,  
location)  
FROM citta  
WHERE name='Parigi';
```



# PostGIS - SQL

```
SELECT sum(ST_Length(the_geom))/1000 AS km_roads  
FROM bc_roads;
```

```
SELECT ST_Area(the_geom)/10000 AS hectares  
FROM bc_municipality  
WHERE name = 'PRINCE GEORGE';
```

```
SELECT name, ST_Area(the_geom)/10000 AS hectares  
FROM bc_municipality  
ORDER BY hectares DESC  
LIMIT 5;
```

```
SELECT sum(ST_Length(r.the_geom))/1000 AS kilometers  
FROM bc_roads r, bc_municipality m  
WHERE r.name = 'Douglas St' AND m.name = 'VICTORIA'  
AND ST_Contains(m.the_geom, r.the_geom);
```



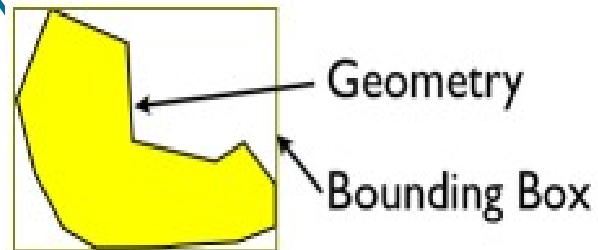


# PostGIS - GiST

Gli indici tradizionali (B-tree) non sono adatti per le ricerche spaziali poiche' sono basati su una sola dimensione.

PostGIS utilizza gli indici GiST (R-tree).

Per rendere efficienti le ricerche e le funzioni si utilizza la tecnica che selezionare gli oggetti interessati mediante il bounding box:



# PostGIS - Caricamento dati

- ◆ Il caricamento e la verifica dei dati sono fondamentali!
- ◆ L'SQL e il WKT sono utilizzabili anche in modo massivo
- ◆ Il comando **Shp2PgsqI** consente di caricare il formato fileshape (introdotto da ESRI ArcGIS). Altro comando, con molti formati supportati e' **ogr2ogr**.
- ◆ I dati introdotti non vengono validati poiche' computazionalmente pesante, e' pero' possibile farlo con la `ST_validate()`



# Applicazioni

PostGIS e' tipicamente utilizzato in un'architettura multi layer integrata con prodotti di visualizzazione Open Source:

OpenLayers

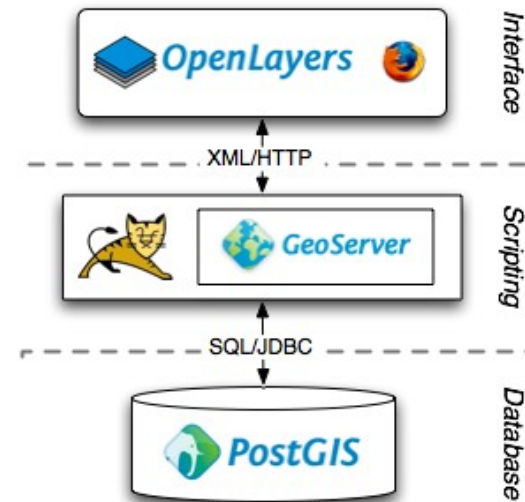
Mapserver

GeoServer

SharpMap SDK

MapGuide

PostGIS !!



Altri: ESRI ArcGIS, FME ...; uDIG, GRASS, ...



# Applicazioni WEB

**OpenStreetMap**  
The Free Wiki World Map

View Edit History Export GPS Traces User Diaries

Welcome, meo63 | inbox (0) | logout

Search [Where am I?](#)  
   
examples: 'Alkmaar', 'Regent Street, Cambridge', 'CB2 5AQ', or 'post offices near Lünen' [more examples...](#)

Help Centre  
Documentation  
Copyright & License  
Community Blogs  
Foundation  
Map Key

[Make a Donation](#)

200 m  
1000 ft

Permalink Shortlink



# Storia

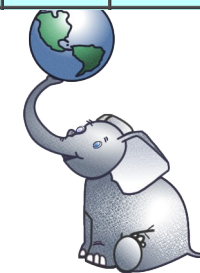
## ◆ Le versioni PostGIS:

Version	Status	Features	Last release	Date (from)	Date (last)	Notes
3.4	Development					
3.3	Production	Require PG 11+, GEOS 3.6+, Proj 5.2+	3.3.2	2022-08	2022-11	
3.2	Production	Requires PG 9.6+, GEOS 3.6+ Proj 4.9+; better with GEOS 3.10+, PG 14+	3.2.4	2021-12	2022-11	
3.1	Production	GEOS 3.9 new features, performance enhancements	3.1.8	2020-12	2022-11	
3.0	Production	Requires Postgres 9.5+, best with Postgres 12+; GEOS 3.6+, Proj 6+.	3.0.8	2019-10	2022-11	
2.5	Production	Postgres 9.4-12 support.	2.5.9	2018-09	2022-11	
2.4	Production	Changed < = > B-Tree index operators, dropped support for PostgreSQL 9.2. (2.4.4 2018-04): PgSQL 11 support	2.4.8	2017-09	2019-08	
2.3	Production	Custom TOC in postgis_restore.pl, TopoGeom_addElement, TopoGeom_remElement, populate_topology_layer, support for 9.6 parallel query mode	2.3.10	2016-09	2019-08	
2.2	Production	PROJ 4.6+, KNN works for all geometry types, Topology API in liblwgeom	2.2.8	2015-10	2018-11	
2.1	Production	Faster! PostgreSQL 9.3 support. New functions for geography, geometry and topology.	2.1.9	2013-08	2017-09	
2.0	Production	GEOS 3.3, PostgreSQL 9.1. New features: 3D (ST_3D*). Raster (based on GDAL). Topology. Geocoder -> TIGER 2010 US Census.	2.0.7	2012-04	2015-03	
1.5	Production	up to pg 8.5; new Geography type for lat/lon; GML, KML format readers; better performance on distance calculation. (1.5.3): up to pg 9.1 (PostgreSQL 9.2 requires PostGis 2.0 or upper) (1.5.6): Added support for pg 9.2 (but not suggested)	1.5.9	2010-02	2012-11	
1.4	Production	New API. Requirements: pg >= 8.2, >= GEOS 3.0, >= PROJ4 4.5. Support for compound curves in a curve polygon (SQL-MM)	1.4.2	2009-03	2010-03	
1.3	Production	Early adoption of SQL-MM convention. (1.3.6) up to pg 8.4	1.3.6	2007-08	2009-05	
1.2	Production	Curved geometry type support for serialization/deserialization; Point-in-polygon shortcircuit to the Contains and Within functions	1.2.1	2006-12	2007-01	
1.1	Production	transform() performance; stable GEOS connectivity (GEOS 2.2); new functions; draft topology support.	1.1.7	2005-12	2007-01	
1.0	Production	Small footprint data representation as default	1.0.6	2005-04	2005-12	
0.x	Pre	First release from Refrations Research. (0.8): Full support for SFSQL.	0.9	2001-05	2005-07	



# Storia

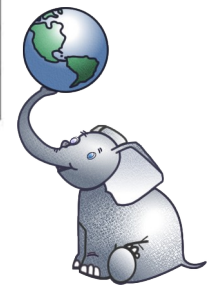
Version	Status	Features	Last release	Date (from)	Date (to)	Notes
PostgreSQL 16	Planned	Planned features (to be confirmed in 2023-04): SQL/JSON features for v15, 64-bit XIDs, native transparent data encryption, improvements to logical replication, parallelism, partitioning, and vacuuming.	16	2023-11	2028-11	
PostgreSQL 15	Production	MERGE command, server-side compression in backup, logical replication for ALL TABLES IN SCHEMA, logical replication filters, new regexp functions, security_invoker in views, optimized sort algorithms, jsonlog logging format, ... Revoked PUBLIC CREATE from public schema. Desupport: exclusive backup mode, python 2, PG9.2 specific code.	15.1	2022-10	2027-11	
PostgreSQL 14	Production	OUT in Stored procedures, SEARCH and CYCLE options in CTE, subscribing operators on jsonb and hstore, multiranges data type, extended statistics also on expressions, new statistics in pg_stat_database (eg. active_time), reduced bloat on B-tree index after updates, more aggressive VACUUM, libpq query pipeline, numerous performance improvements on: parallel queries, heavily-concurrent workloads, partitioned tables, logical replication, vacuuming! (14.3 2022-05): CVE-2022-1552 fix. (14.4 2022-06): fixed a bug affecting 14 only (data corruption using CONCURRENTLY).	14.6	2021-09	2026-11	<b>Suggested</b>
PostgreSQL 13	Production	Better performance with aggregates and partitioning, parallel VACUUM for indexes, space saving in indexes, new monitoring views, trusted extensions, ... (13.1 2020-11): CVE-2020-25695 fix. (13.7 2022-05): CVE-2022-1552 fix.	13.9	2020-09	2025-11	<b>Suggested</b>
PostgreSQL 12	Production	Performance improvements on partitioning, decreased WAL usage for indexes, JSON path expressions, stored generated columns, REINDEX CONCURRENTLY, ... Incompatibilites: replication parameters in postgres.conf. (12.5 2020-11): CVE-2020-25695 fix. (13.7 2022-05): CVE-2022-1552 fix.	12.13	2019-10	2024-11	<b>Suggested</b>
PostgreSQL 11	Production	Improvements to partitioning (by hash, better query pruning, full referential constraint support, upsert, row move, ...), transaction control in PL/pgSQL, SQL stored procedures, enhanced parallelism, full SQL:2011 standard support for window functions, JIT compilation, ... quit and exit in psql! (11.4 2019-06): CVE-2019-10164 fix. (11.10 2020-11): CVE-2020-25695 fix. (11.16 2022-05): CVE-2022-1552 fix.	11.18	2018-10	2023-11	
PostgreSQL 10	Production	Built-in Logical Replication, native Table Partitioning, conditional scripting in psql, enhanced Query Parallelism, quorum commit on Synchronous Replication. Driver updates: SCRAM-SHA-256 Authentication, Multi-host failover, Read-only/Read-Write connections. Incompatibilites: directory changes, wal functions renamed, version numbering. Desupport: pg_dump for DB older than 8.0. (10.3 2018-03): CVE-2018-1058 fix. (10.9 2019-06): CVE-2019-10164 fix. (10.15 2020-11): CVE-2020-25695 fix. (10.21 2022-05): CVE-2022-1552 fix.	10.23	2017-10	2022-11	



# PostgreSQL/PostGIS Compatibility Matrix

◆ Compatibilita' tra le versioni PostgreSQL / PostGIS:

PostgreSQL	PostGIS												
	1.3	1.4	1.5	2.0	2.1	2.2	2.3	2.4	2.5	3.0	3.1	3.2	3.3
7.x	Yes	No	No	No	No	No	No	No	No	No	No	No	No
8.0	Yes	No	No	No	No	No	No	No	No	No	No	No	No
8.1	Yes	No	No	No	No	No	No	No	No	No	No	No	No
8.2	Yes	Yes	No	No	No	No	No	No	No	No	No	No	No
8.3	Yes	Yes	Yes	No	No	No	No	No	No	No	No	No	No
8.4	Yes	Yes	Yes	Yes	No	No	No	No	No	No	No	No	No
9.0	No	No	Yes	Yes	Yes	No	No	No	No	No	No	No	No
9.1	No	No	Yes	Yes	Yes	Yes	No	No	No	No	No	No	No
9.2	No	No	Yes	Yes	Yes	Yes	No	No	No	No	No	No	No
9.3	No	No	No	No	Yes	Yes	Yes	Yes	No	No	No	No	No
9.4	No	No	No	No	Yes	Yes	Yes	Yes	Yes	No	No	No	No
9.5	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes	No	No	No
9.6	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
10	No	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	No
11	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes
12	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes	Yes
13	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes	Yes
14	No	No	No	No	No	No	No	No	No	No	Yes	Yes	Yes
15	No	No	No	No	No	No	No	No	No	No	No	Yes	Yes
16	No	No	No	No	No	No	No	No	No	No	No	No	Yes



# Varie ed eventuali

## ◆ Link utili

<https://www.postgis.org/>

Sito ufficiale PostGIS

<https://www.postgresql.org/>

Sito ufficiale PostgreSQL

<https://www.opengeospatial.org/standards/sfs>

Standard OGC: Simple Feature for SQL

<https://www.md-c.it/meo/index.htm#post>

Pagine su PostgreSQL non ufficiali ma... in italiano!

