

# INTRODUZIONE A POSTGRESQL e POSTGIS

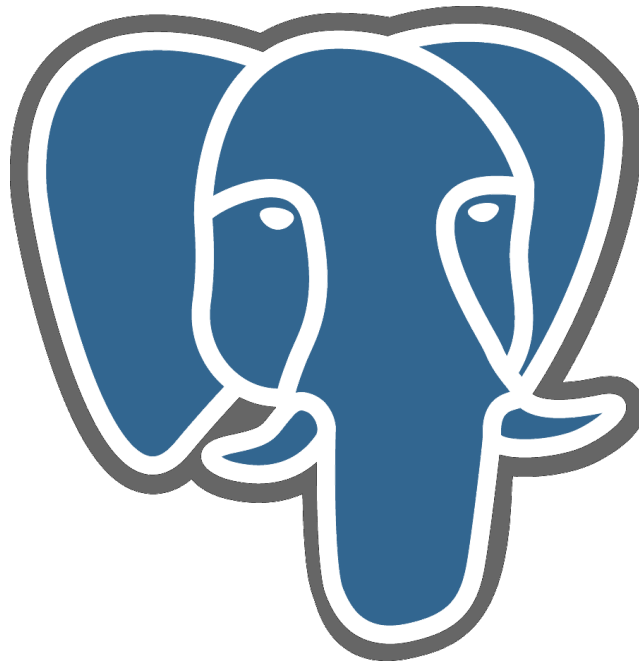


**Luca Lanteri**



**Rocco Pispico**

# The world's most advanced open source database



PostgreSQL è un potente **database relazionale** Open Source. Il suo sviluppo è attivo da oltre 20 anni e la sua architettura si è guadagnata una forte reputazione per l'affidabilità e l'integrità dei dati.

PostgreSQL è stato sviluppato originariamente su UNIX ma oggi è disponibile per tutti i principali sistemi operativi. E' testato sulle principali distribuzioni Linux, su UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), e Windows.

PostgreSQL non è controllata da nessuna ditta privata o corporation ed il codice sorgente è distribuito liberamente e gratuitamente.

E' rilasciato sotto secondo la **PostgreSQL License**, simile alla licenza BSD o alla licenza MIT.

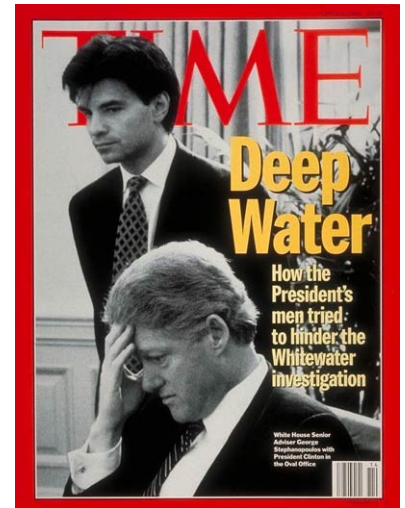
<https://opensource.org/licenses/postgresql>

# Storia

Nel **1986** la Defense Advanced Research Projects Agency (DARPA), l'Army Research Office (ARO), il National Science Foundation (NSF), sponsorizzarono l'università di Berkeley per lo sviluppo di del progetto **POSTGRES** guidato da Michael Stonebraker.



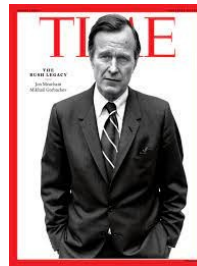
Nel **1994**, dopo diverse fasi di sviluppo, viene aggiunto il supporto per un interprete al linguaggio SQL e in seguito viene rilasciato sul web sotto il nome di **Postgres95**.



# Storia



Dal **1996**, il progetto viene ribattezzato PostgreSQL.



Oggi è disponibile la **versione 13** di PostgreSQL. Vengono comunque ancora mantenute tutte le versioni precedenti fino alla 9.6.

PostgreSQL 9.6 will stop receiving fixes on November 11, 2021. If you are running PostgreSQL 9.6 in a production environment, we suggest that you make plans to upgrade to a newer, supported version of PostgreSQL.







Viene stimato che circa il 30% delle compagnie tecnologiche usano PostgreSQL per applicazioni core (2012).

La versione 8.0 ha avuto circa un milione di download dopo sette mesi dal rilascio

La mailing list internazionale conta circa 35,000 iscritti



# Funzionalità

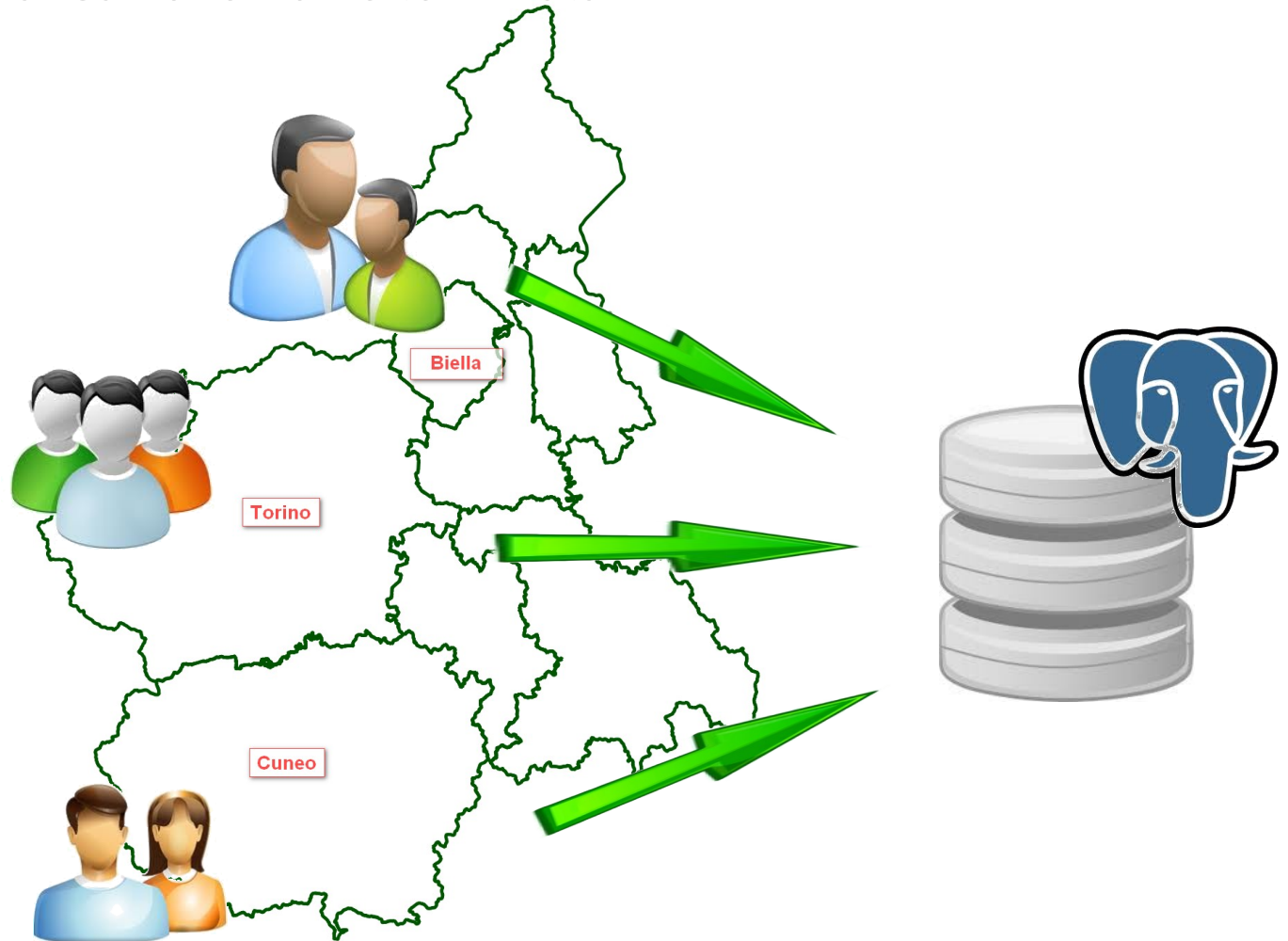
PostgreSQL offre moltissime delle funzionalità avanzate messe a disposizione dai più blasonati database commerciali:

- Profilazione degli utenti e degli accessi
- Tipi definiti dall'utente
- Ereditarietà delle tavole
- Foreign key integrità referenziale
- Viste, regole, subquery
- Replica asincrona
- Trigger
- Tablespaces
- Point-in-time recovery



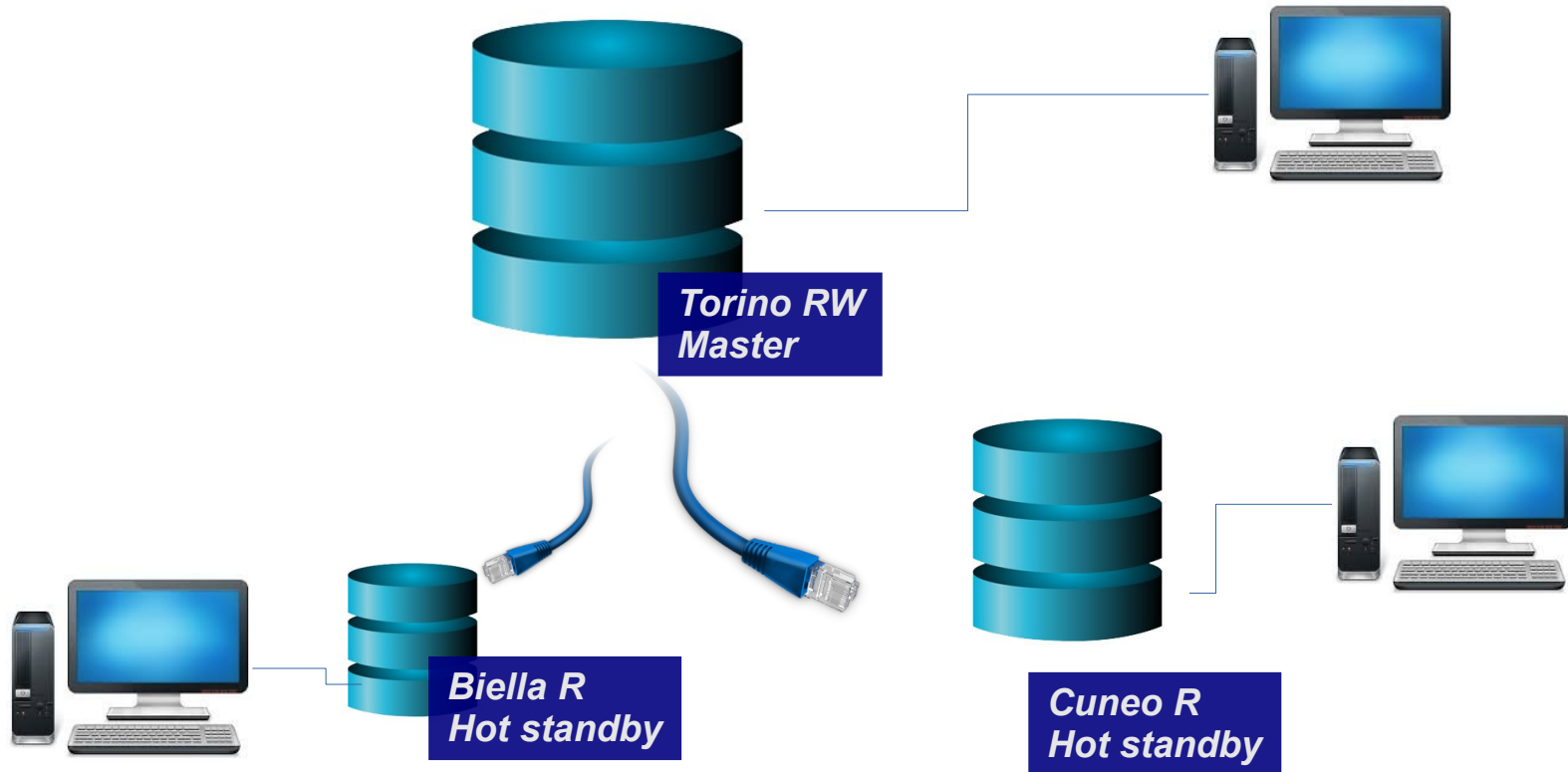
# Multiutenza

*Il Multiversion concurrency control (MVCC) permette di gestire l'accesso contemporaneo concorrente. L'azione di ogni utente non è visibile agli altri fino a quando la transazione non è terminata.*



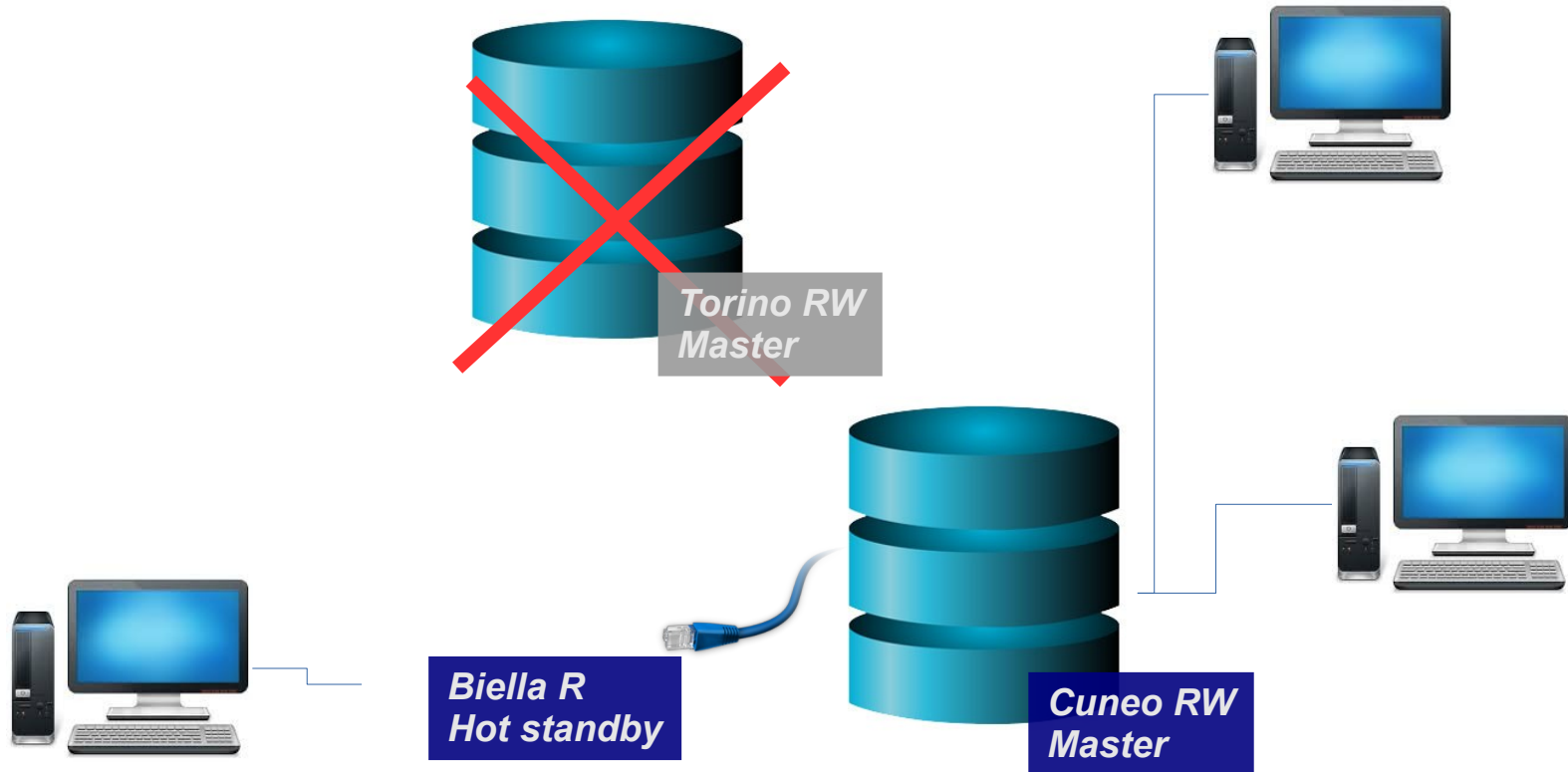
# Replica

A partire dalla versione 8, PostgreSQL ha ampiamente migliorato la sua scalabilità e le sue performance

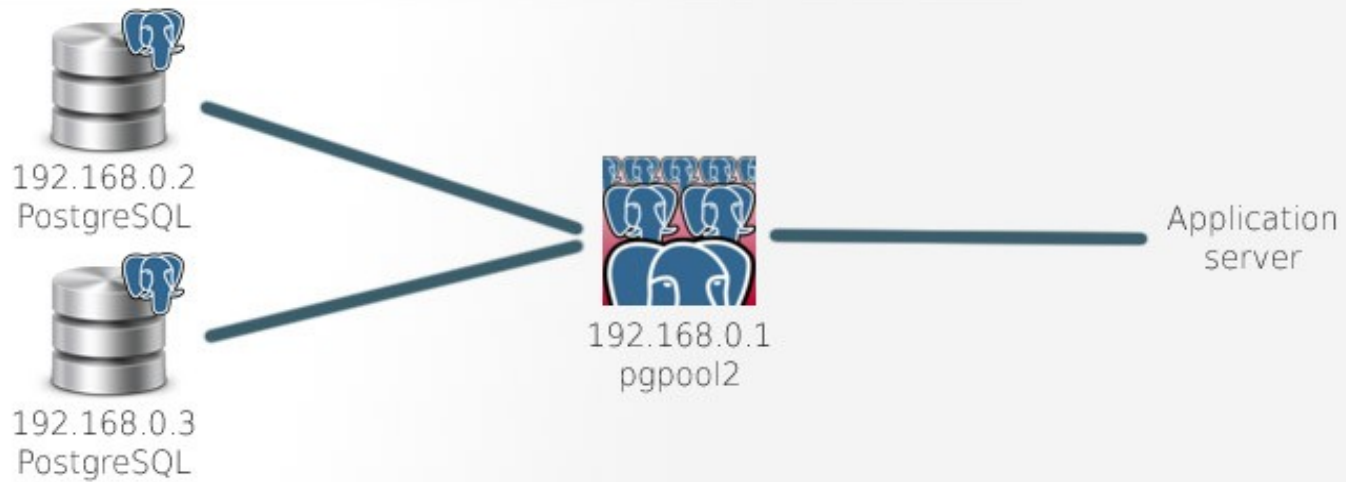


# Alta disponibilità

In caso di failover è possibile promuovere uno degli hot standby a master server



# Load Balancing





# Linguaggi procedurali

PostgreSQL supporta diversi linguaggi procedurali che permettono all'utente di scrivere codice personalizzato che viene eseguito direttamente dal database server:

- **PL/pgSQL,**
- PL/Tcl,
- PL/Perl and
- PL/Python.

Pacchetti esterni permettono il supporto di altri linguaggi non standard:

- PL/PHP,
- PL/V8,
- PL/Ruby,
- PL/Java etc.
- **PL/R**

# Indici, trigger, viste e regole

## Indici:

PostgreSQL supporta nativamente diversi tipi di indici : **B+-tree**, hash, generalized search trees (**GiST**) e generalized inverted indexes (**GIN**). Possono essere anche creati indici personalizzati dagli utenti.

## Triggers:

E' possibile eseguire azioni personalizzate, generalmente scatenate od ogni INSERT o UPDATE che permettono di personalizzare in modo estremamente versatile il proprio database

## Viste e regole:

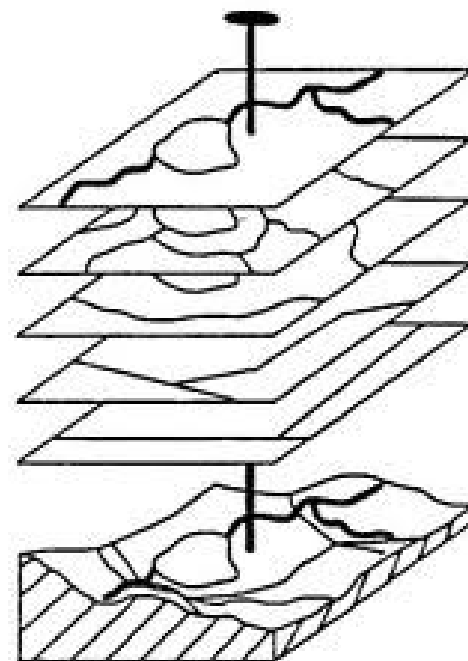
PostgreSQL supporta le viste, le tavole virtuali e le regole.



## Analisi spaziali automatizzate

Contestualmente alla digitalizzazione di un poligono di frana è possibile ricavare immediatamente diverse informazioni:

- Statistiche di base dati interferometrici
- Elementi antropici che intersecano o che si trovano in prossimità del perimetro di frana
- Dati di base utili alla caratterizzazione del fenomeno (sondaggi, monitoraggi, uso del suolo, pluviometri ecc..)

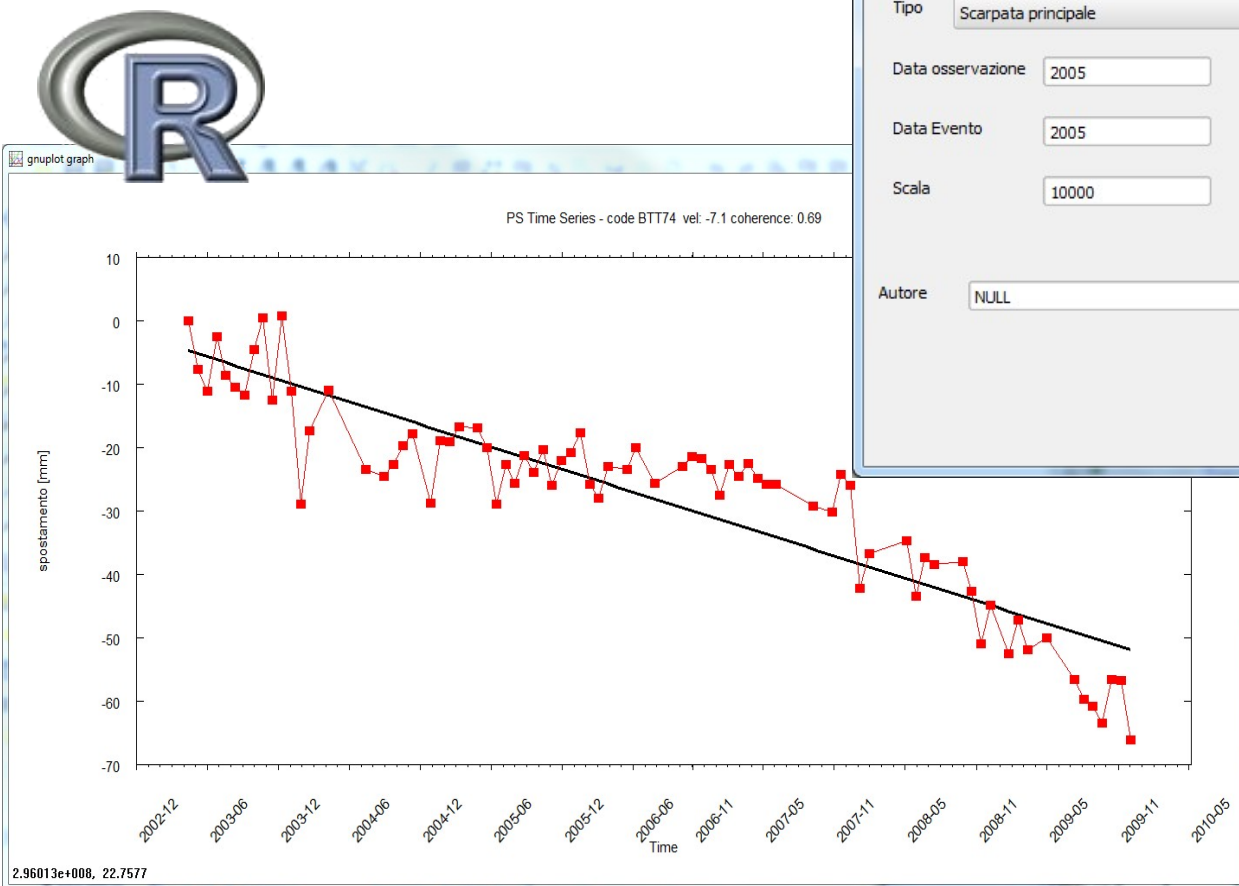


Sono **supportati nativamente moltissimi tipi di dati**: Boolean, Arbitrary precision numeric, Character (text, varchar, char), Binary, Date/time (timestamp/time with/without timezone, date, interval), Money Enum, Bit strings, Text search type, Composite Variable length arrays (including text and composite types) up to 1 GB in total storage size, Geometric primitives, IPv4 and IPv6 addresses, CIDR blocks and MAC addresses, XML supporting XPath queries, UUID.

**E possibile creare tipi dati, oggetti e funzioni personalizzate** come: Casts, Conversions, Data types, Domains, Functions including aggregate functions and window functions, Indexes including custom indexes for custom types Operators.



# Strumenti per gestione ed analisi dati



Dialog

SIFRAP - Elementi Morfologici Poligonal

Tipo:

Data osservazione:

Data Evento:

Scala:

Autore:

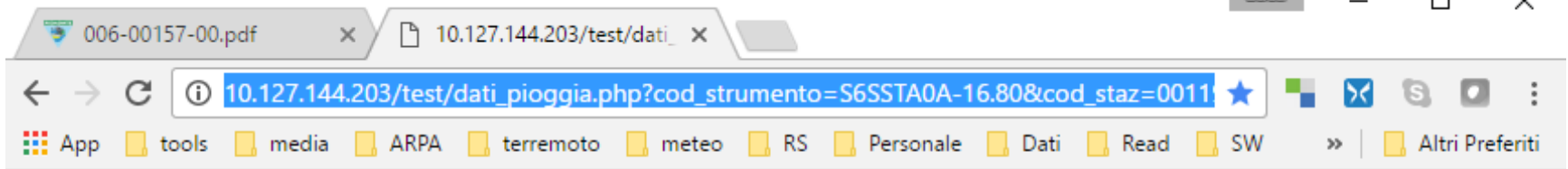
Fonti:

Descrizione: Lorem Ipsum, Lorem IpsumLorem IpsumLorem IpsumLorem IpsumLorem IpsumLorem IpsumLorem IpsumLorem IpsumLorem IpsumLorem IpsumLorem IpsumLorem Ipsum

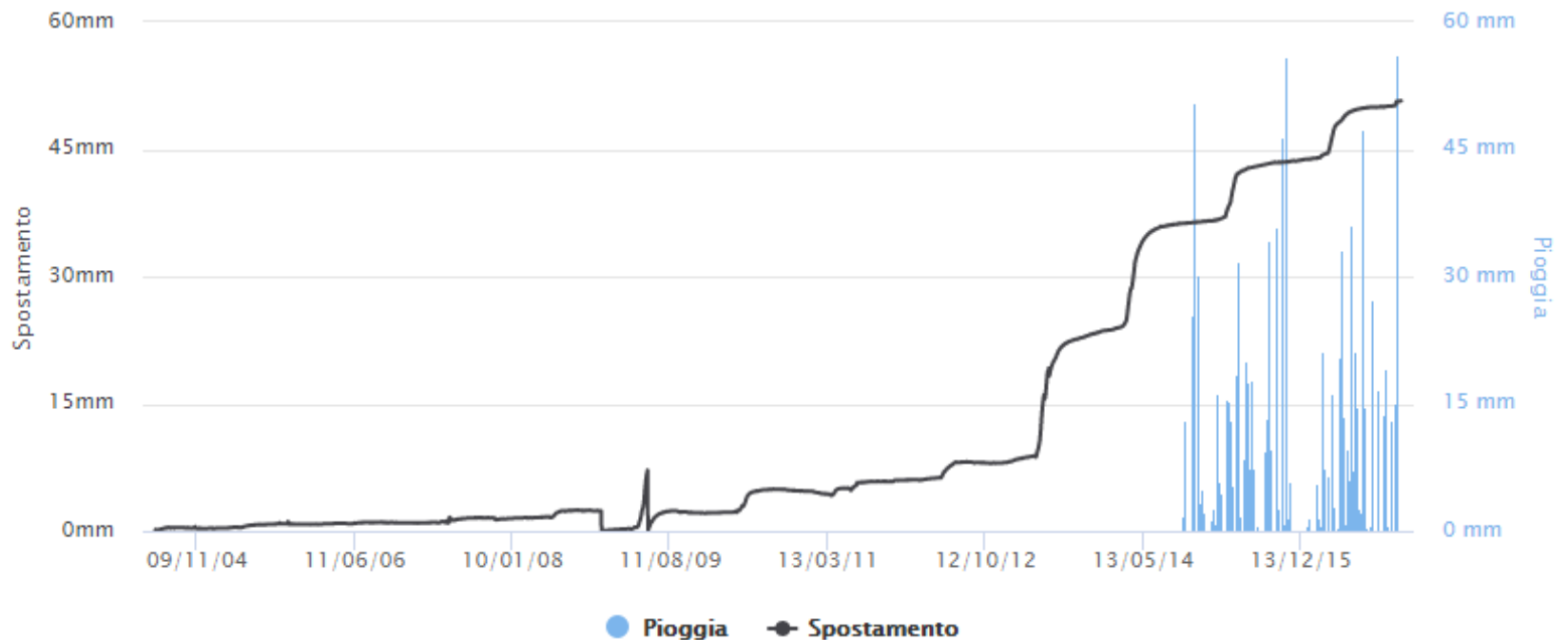
OK Cancel



# Strumenti per gestione ed analisi dati



## Grafici spostamento inclinometri fissi



Highcharts.com

# Estensioni

Grazie alle estensioni possono essere installate nuove funzionalità.

PostgreSQL è progettato per essere estensibile. In PostgreSQL, possono essere definiti tipi di dati indici, linguaggi e funzioni personalizzate e distribuite tramite le estensioni.

# PostGIS

PostGIS è un'estensione spaziale per PostgreSQL. Fornisce il supporto per la gestione di oggetti geografici.

Aggiunge nuovi tipi di oggetti (geometry, geography, raster, indici, funzioni e altro) a PostgreSQL:

- Aggiungendo in PostgreSQL librerie in C.
- Collegando le funzioni e le strutture messe a disposizione dalle librerie C con i tipi e le funzioni in SQL.

PostGIS è sviluppato in C, e si appoggia su altre librerie:

- GEOS mette a disposizione algoritmi di processing
- Proj.4 supporto per le coordinate geografiche e funzioni di proiezione
- GDAL gestione dei raster e supporto ai formati
- LibXML2 parsing XML
- JSON-C parsing JSON
- SFCGAL supporto 3D e algoritmi aggiuntivi di geoprocessing





# Quale DBMS scegliere

**Formato standard aperto, per gestire dati geospaziali**

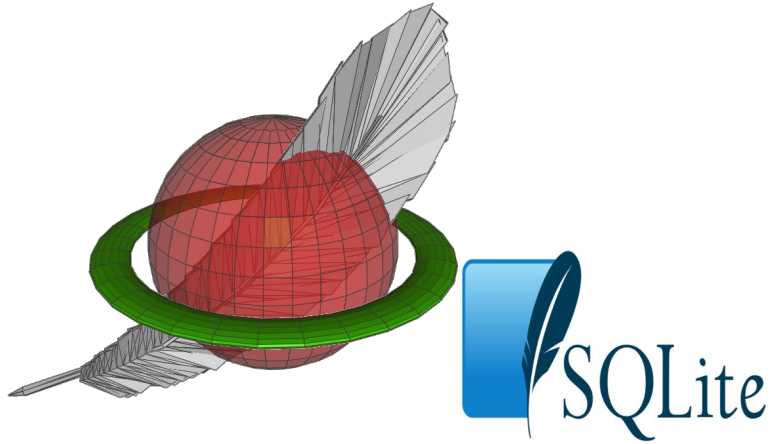


- Indipendente dalla piattaforma
- Portabile (no installazione)
- compatto
- Gestisce vettoriali e raster
- Attributi (non-spatial data)
- Estensibile
- Standard OGC

- Tipi dati e linguaggio limitati
- Accesso a utente singolo
- Non permette analisi spaziali

**Alternativa allo shapefile**

# Quale DBMS scegliere



- Indipendente dalla piattaforma
- Portabile (no installazione)
- Compatto
- Gestisce vettoriali
- Attributi (non-spatial data)
- Estensibile

- Tipi dati e linguaggio limitati
- Accesso a utente singolo
- **Comprende funzioni per analisi spaziali**

# Quale DBMS scegliere



- Estremamente potente, flessibile e scalabile
- Accesso multiutente
- Accesso LAN/web
- Estensibile
- Innumerevoli formati dati

- Installazione server
- Maggior complessità

# Installazione - Linux

Tutte le principali distribuzioni Linux forniscono i pacchetti di PostgreSQL, per installarlo quindi basta digitare da linea di comando:

```
sudo apt-get update
sudo apt-get install postgresql postgresql-contrib

sudo apt-get install postgis
sudo apt-get install pgadmin3 qgis
```



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



<https://apt.postgresql.org/pub/repos/apt/>











# Installazione - Windows

## Interactive installer by EnterpriseDB

<http://www.enterprisedb.com/products/pgdownload.do#windows>

## Graphical installer by BigSQL

<https://www.biasal.ora/postgresal/installers.isp>

PostgreSQL 9.6.1 - Stable (27-Oct-16)	PostgreSQL 9.5.5 - Proven (27-Oct-16)
 <a href="#">postgresql-9.6.1-win64.exe</a>	 <a href="#">postgresql-9.5.5-win64.exe</a>
 <a href="#">postgresql-9.6.1-osx64.dmg</a>	 <a href="#">postgresql-9.5.5-osx64.dmg</a>
 <a href="#">postgresql-9.6.1-x64.rpm</a>	 <a href="#">postgresql-9.5.5-x64.rpm</a>
 <a href="#">postgresql-9.6.1-x64.deb</a>	 <a href="#">postgresql-9.5.5-x64.deb</a>



# Ruoli

Postgresql usa il concetto di ruoli per permettere l'autenticazione al database. Dopo l'installazione è disponibile un nuovo utente unix/linux chiamato **postgres**, a cui è associato il **ruolo postgres** sul db. Con tale utente è pertanto possibile accedere sia al database che a tutti i file presenti sul filesystem per la configurazione di Postgres. E' possibile accedere all'utente con il comando:

```
sudo -i -u postgres
```

# Configurazione di Base – pg\_hba.conf

Permette di gestire le policy di accesso al DB

Debian/Ubuntu	/etc/postgres/[versione]/main
Fedora/Redhat	/var/lib/pgsql/data
Windows	c:/programmi/postgres/data

```
# TYPE DATABASE USER CIDR-ADDRESS METHOD

# local for domain socket only
local all postgres ident

# IPv4 connections
# TYPE DATABASE USER CIDR-ADDRESS METHOD

# local for domain socket only
local all postgres ident

# IPv4 connections
host all postgres md5
host all all 127.0.0.1/32 md5
host all all 192.168.0.0/24 md5

# IPv6
host all all ::1/128 md5
```

# Configurazione di Base – postgres.conf

Permette di gestire la configurazione del DB

Debian/Ubuntu	/etc/postgres/[versione]/main
Fedora/Redhat	/var/lib/pgsql/data
Windows	c:/programmi/postgres/data

```
listen_address = 'localhost'

listen_address = '*'
```

# File di log

Tiene traccia di tutto quello che

Debian/Ubuntu    /var/log/postgres/

Windows: /programmi/postgres/log

```
2017-02-01 09:47:45 CET LOG:  database system is ready to accept
connections
2017-02-01 09:47:45 CET LOG:  autovacuum launcher started
2017-02-01 18:33:46 CET LOG:  received fast shutdown request
2017-02-01 18:33:46 CET LOG:  aborting any active transactions
2017-02-01 18:33:46 CET LOG:  autovacuum launcher shutting down
2017-02-01 18:33:46 CET LOG:  shutting down
2017-02-01 18:33:46 CET LOG:  database system is shut down
2017-02-01 21:37:03 CET LOG:  database system was shut down at 2017-02-01
18:33:46 CET
2017-02-01 21:37:03 CET LOG:  incomplete startup packet
2017-02-01 21:37:03 CET LOG:  MultiXact member wraparound protections are
now enabled
2017-02-01 21:37:03 CET LOG:  database system is ready to accept
connections
2017-02-01 21:37:03 CET LOG:  autovacuum launcher started
```

# Tools per gestire PostgreSQL

Esistono diversi front-end Open Source e proprietari per PostgreSQL.

**Psql:** gestore da linea di comando. È il principale strumento per accedere a PostgreSQL.

**PgAdmin:** La più diffusa GUI per amministrare PostgreSQL e accede ai dati.

**PhpPgAdmin:** Tool per amministrare PostgreSQL via web, scritto in PHP. Molto utile per amministrare database remoti

**PgFouine:** Log analyzer che crea reports basati sul file di log di PostgreSQL. (Proprietario)



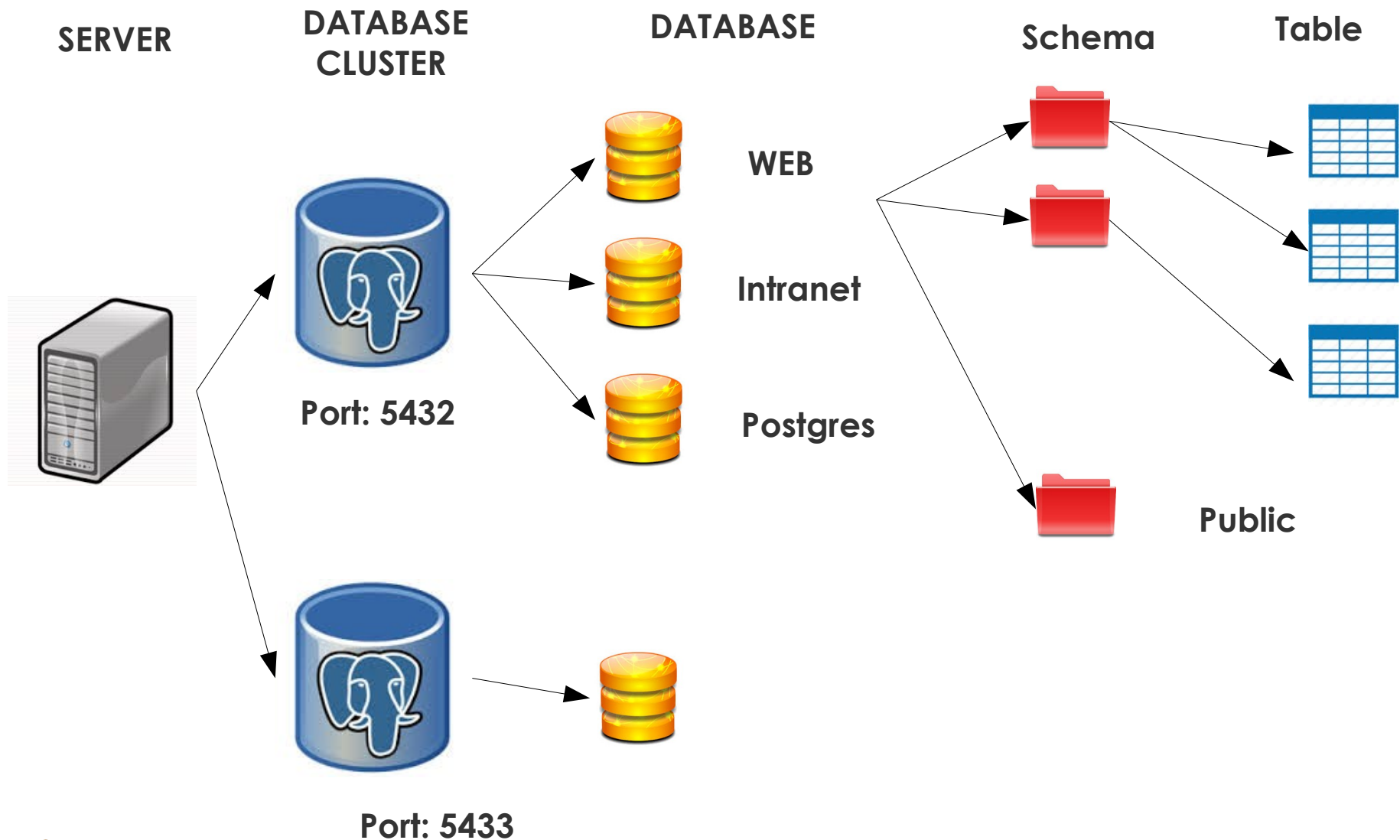
# Tools per gestire PostgreSQL

Esistono diversi front-end Open Source e proprietari

Altri strumenti di amministrazione per PostgreSQL:

Borland Kylix, DBOne, DBTools Manager PgManager, Rekall, Data Architect, SyBase Power Designer, Microsoft Access, eRWin, DeZign for Databases, PGExplorer, Case Studio 2, pgEdit, RazorSQL, MicroOLAP Database Designer, Aqua Data Studio, Tuples, EMS Database Management Tools for PostgreSQL, Navicat, SQL Maestro Group products for PostgreSQL, Datanamic DataDiff for PostgreSQL, Datanamic SchemaDiff for PostgreSQL, DB MultiRun PostgreSQL Edition, SQLPro, SQL Image Viewer, SQL Data Sets etc.

# Strutturare un server postgres



# Regole

PostgreSQL gestisce l'accesso al database utilizzando il concetto di ruoli (ROLES).

Un ruolo può essere visto come un UTENTE o come un GRUPPO DI UTENTI.

I ruoli possono essere proprietari gli oggetti del database e possono assegnare privilegi su questi oggetti ad altri ruoli.

```
sudo - su postgres  
psql
```

```
CREATE ROLE miriam WITH LOGIN PASSWORD 'jw8s0F4' VALID UNTIL '2005-01-01';
```



Ogni ruolo può avere diverse proprietà, es:

CAN CREATE ROLES;

SUPERUSER;

# User & Grant

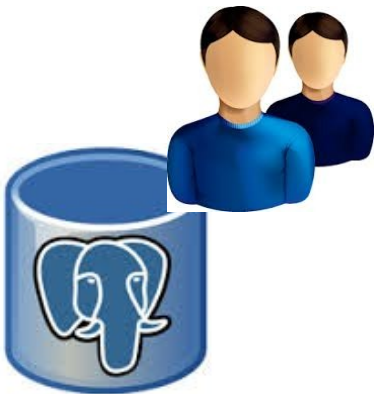
E' possibile profilare i diritti di accesso ai principali oggetti del database.  
Possono essere forniti i diritti a singoli utenti (USERS) o a gruppi (ROLES).  
Ogni oggetto ha differenti tipi di diritti:

DATABASE: Create, Temp, Connect

SCHEMA: Usage, Connect

TABLE, VIEW: Insert, Select, Update, Delete, Truncate, Trigger, References

FUNCTION: Execute



# Configurazione del DB

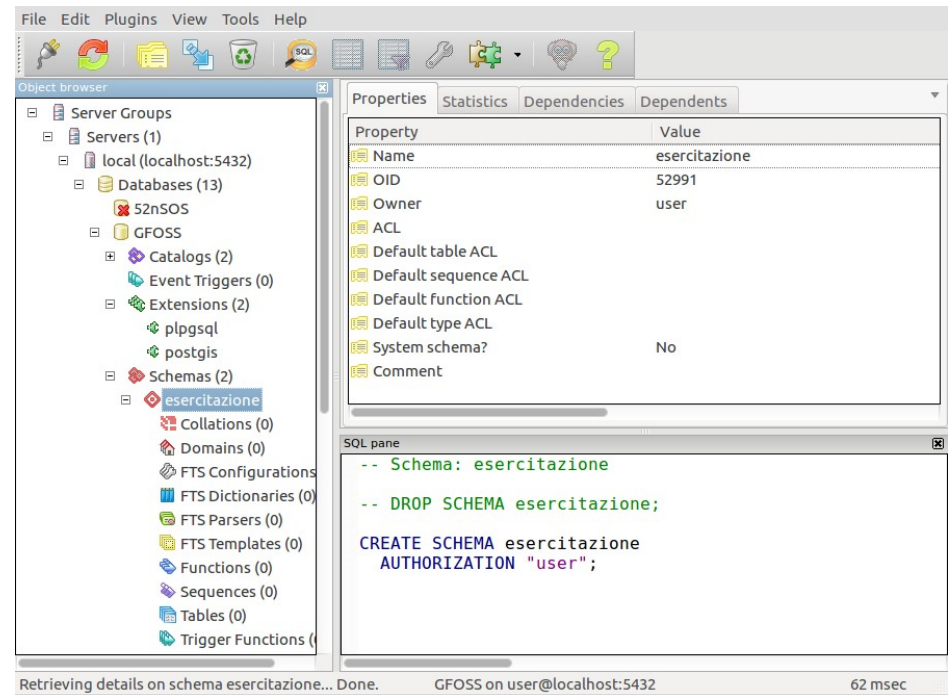
1. Creare un nuovo database
2. Abilitare postGIS
3. Creare un nuovo schema

## Psql

```
sudo - su postgres  
psql
```

```
CREATE DATABASE "gfooss2019";  
CREATE SCHEMA dati;
```

## PGadmin3

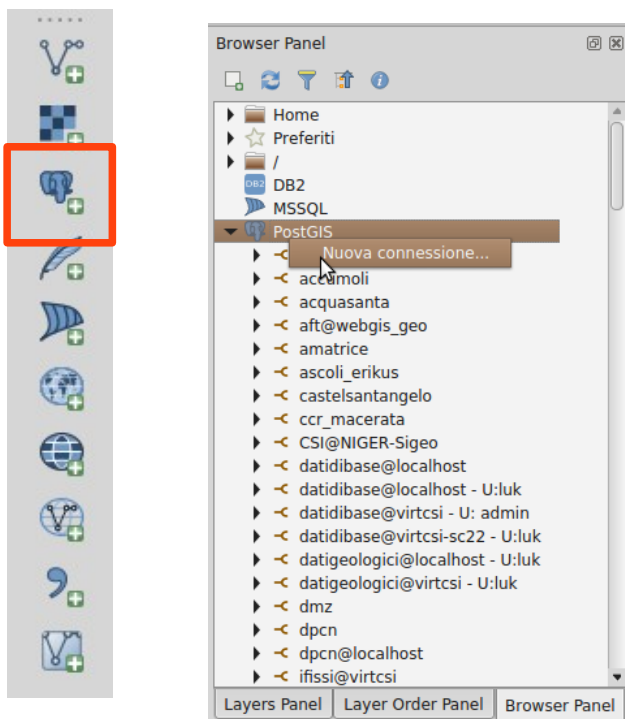


# Importare i dati con QGIS



## 1- Creare una nuova connessione

Browser panel - > PostGIS - > tasto dx - > nuova connessione



**Crea una nuova connessione PostGIS**

**Informazioni Connessione**

Nome:

Servizio:

Host:

Porta:

Database:

Modalità SSL:

**Authentication** **Configurations**

Nome utente:  ☐ Salva

Password:  ☐ Salva

☐ Cerca solamente nei registri del layer

☐ Non risolvere tipo di geometria senza restrizioni (GEOMETRY)

☐ Cerca solamente nello schema 'public'

☐ Mostra anche le tabelle senza geometria

☐ Usa i metadati stimati della tabella



if you need any support, an active community is available to help. You can always find the answers from the PostgreSQL's community for the issues that you may have when working with PostgreSQL. Many companies offer commercial support services in case you need one.

