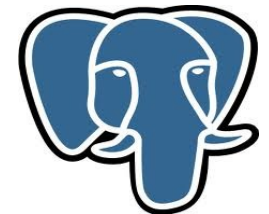


PostgreSQL

Il database **Open Source** piu' avanzato



Argomenti trattati

◆ **Introduzione**

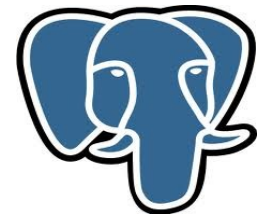
- ORDBMS

◆ PostgreSQL

- Architettura
- SQL

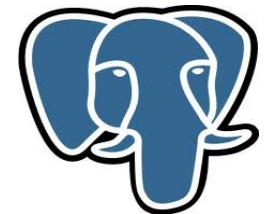
◆ etc

- Tools
- Storia



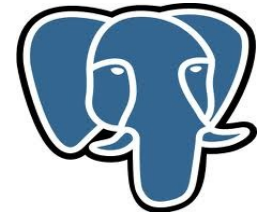
◆ Obiettivi

- ◆ Fornire una conoscenza di base su **PostgreSQL** dal punto di vista architetturale e funzionale
- ◆ Presentare le principali caratteristiche dell'SQL di PostgreSQL
- ◆ Fornire una panoramica sugli strumenti più comunemente utilizzati su PostgreSQL
- ◆ Storia di PostgreSQL e... futuro!



ORDBMS

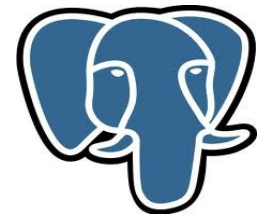
- ◆ PostgreSQL e' DBMS relazionale Open Source piu' avanzato al mondo e non ha nulla da invidiare a sistemi commerciali. I suoi principali punti di forza sono:
 - Free! Gratis e distribuito con una licenza molto libera
 - Funzionalita' molto ampie (eg. Object Oriented, GIS, ...)
 - Robustezza ed integrita'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



ORDBMS

◆ 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



ORDBMS

- ◆ Le caratteristiche object-oriented sono potenti ma di semplice utilizzo:

- Ereditarietà:

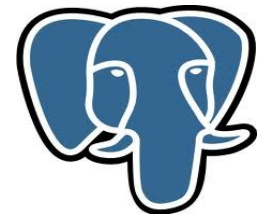
```
CREATE TABLE emp (sal numeric) INHERITS  
(person);
```

- Constraints:

```
ON DELETE/UPDATE [RESTRICT|CASCADE|SET  
NULL|SET DEFAULT]
```

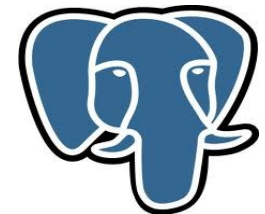
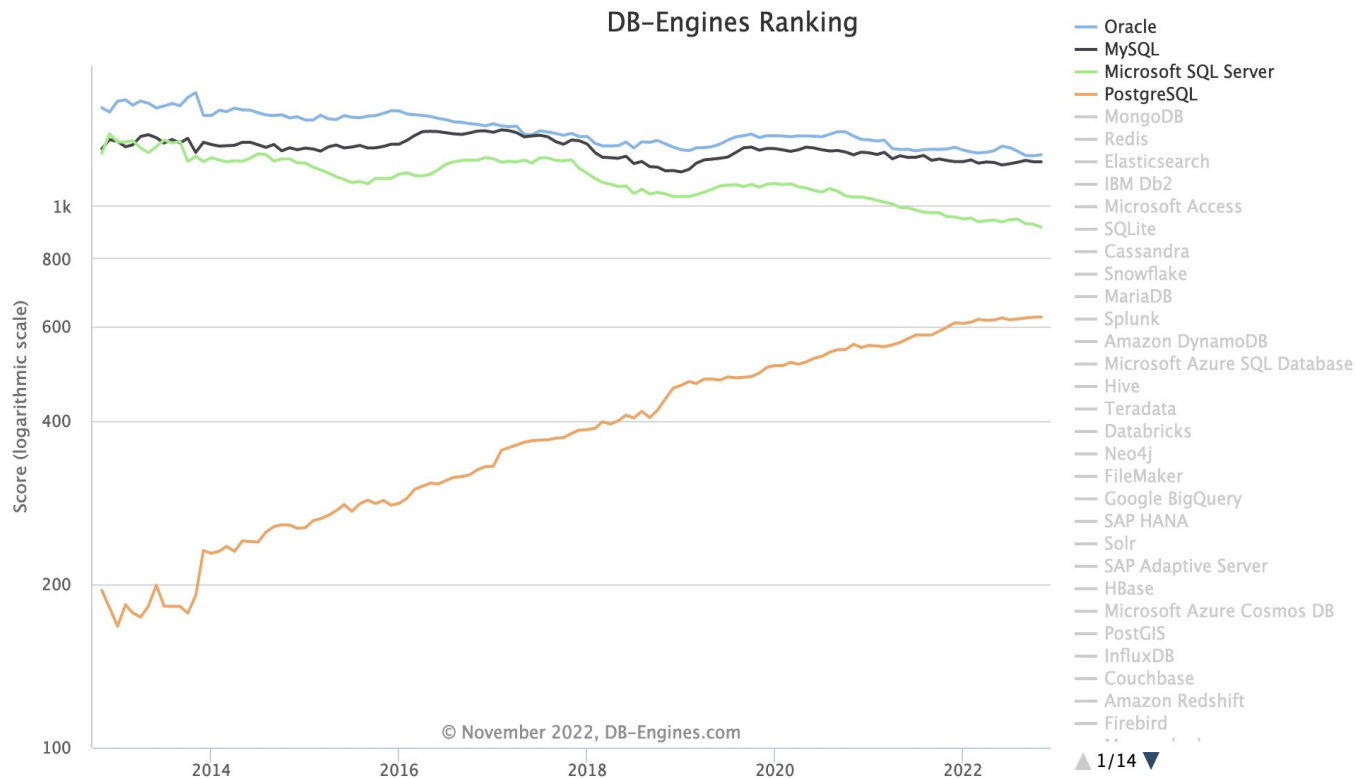
- Arrays, User defined data type:

```
INTEGER[3]
```

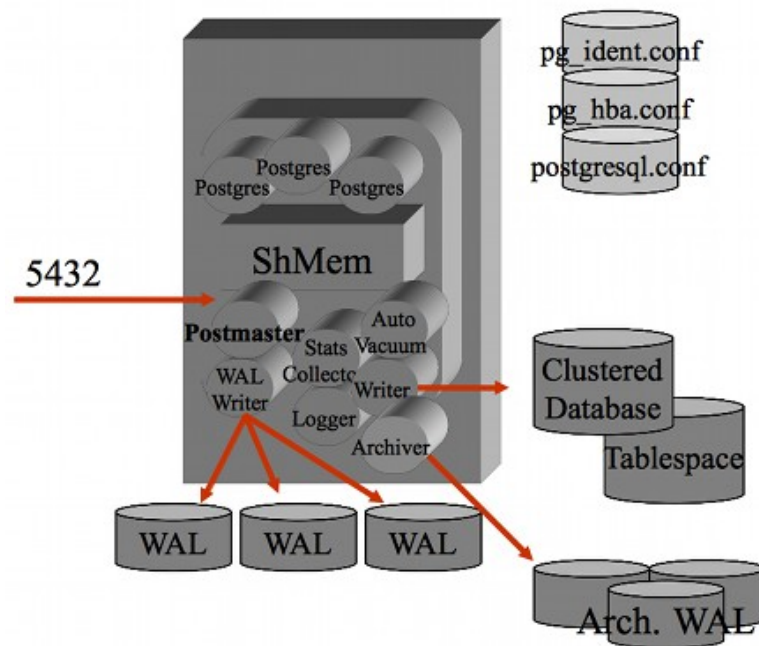


Diffusione

◆ PostgreSQL e' utilizzato in moltissimi progetti ed ambienti:



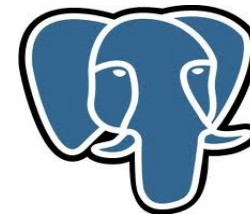
Architettura



L'architettura di PostgreSQL e' *relativamente* semplice. La struttura dei processi presenta una serie di processi di background, tra cui il principale (Postmaster) e gli eventuali processi utente (Postgres Server). Il processo postmaster e' il processo principale che si occupa della gestione delle connessioni (e' in LISTEN sulla porta socket di default 5432) ed e' il "padre" di tutti i processi, sia di sistema (eg. wal writer) sia quelli relativi alle connessione utente. Su ogni processo utente e' riportata in argv[*] l'origine e l'attivita' in corso.

Tutti i processi girano come utente postgres ed eseguono un attach al segmento di shared memory su cui vengono mantenuti buffer e lock. Nel tempo l'architettura dei processi si e' mantenuta sempre costante anche se nelle versioni piu' recenti sono presenti piu' processi di sistema.

La consistenza dei dati e' mantenuta con l'MVCC.



Architettura - Utilizzo

- ◆ PostgreSQL si utilizza con interfacce grafiche Client-Server (pgAdmin):

The screenshot displays the pgAdmin III interface. On the left, the 'Object browser' shows a tree view of server groups and servers. The 'Servers (7)' folder is expanded, showing several servers, including 'tst-benchpost3 (10.102.102.155:5432)'. The 'bench' role is selected under the 'Login Roles (2)' for this server.

The main pane shows the 'Properties' tab for the selected role, displaying the following table:

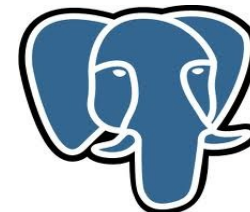
Property	Value
Name	bench
OID	16388
Account expires	
Can login?	Yes
Superuser?	No

Below the properties, the 'Server Status - tst-benchpost3 (10.102.102.155:5432)' window is open, showing the 'Activity' tab. This tab displays a table of active connections:

PID	Application name	Database	User	Client	Client start
1918	pgAdmin III - Browser	postgres	postgres	158.102.160.26:577...	2044-12-30 11:59:33
1919	pgAdmin III - Browser	bench	postgres	158.102.160.26:577...	2044-12-30 11:59:34
2114		bench	bench	158.102.160.26:579...	2044-12-30 12:11:27

At the bottom of the activity window, there is a 'Locks' tab with a table structure:

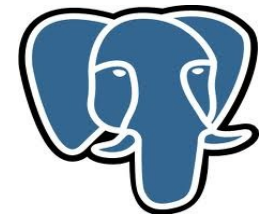
PID	Database	Relation	User	XID	TX	Mode	Granted
-----	----------	----------	------	-----	----	------	---------



SQL

- ◆ PostgreSQL fornisce un SQL ANSI SQL:2016 molto completo e ricco. PostgreSQL è semplice da utilizzare:

```
$ psql [-d bench]
bench=# \dt
          List of relations
 Schema |   Name   | Type  | Owner
-----+-----+-----+-----
 public | accounts | table | bench
 public | branches | table | bench
 ...
bench=# \h
Available help:
      ABORT          COMMIT          DELETE          FETCH
 ALTER AGGREGATE    COMMIT PREPARED  DISCARD         GRANT
 ...
bench=# ^D
```



SQL (DCL)

- ◆ Un'istanza PostgreSQL gestisce piu' database: e' un clustered database
- ◆ Ogni DB e' distinto ed ha un suo catalogo. La scelta del DB da utilizzare va fatta in connessione e non puo' essere cambiata
- ◆ Gli utenti/ruoli valgono per tutto il DB cluster. Le autorizzazioni vengono concesse con il comando di GRANT
- ◆ Ogni Database ha un ruolo PUBLIC
- ◆ Non esistono sinonimi ma la ricerca degli oggetti avviene con il SEARCH_PATH

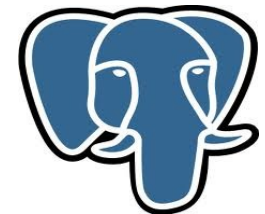
◆ SQL da linea di comando: **psql**

\XYZ permette di visualizzare informazioni sugli oggetti presenti e sullo stato della base dati oppure impostare variabili



SQL (DDL)

- ◆ I comandi di DDL sono SQL Standard (eg. create table):
 - ◆ E' possibile indicare attributi sulle colonne (eg. NOT NULL, UNIQUE, ...)
 - ◆ E' possibile dichiarare le foreign key ed i relativi constraint
 - ◆ **INTEGER**, BIGINT 4/8 byte, **NUMERIC**(s,p) o DECIMAL a virgola mobile, DOUBLE 8 byte in binario
 - ◆ DATE solo data, TIME solo ora (fino ai microsecondi), **TIMESTAMP** data ed ora (fino ai microsecondi)
 - ◆ **CHAR**, **VARCHAR** e **TEXT** arrivano sino ad 1GB; BYTEA dati binari
 - ◆ SERIAL assegna automaticamente valori univoci alle colonne
 - ◆ Altro: ENUM, XML, INTERVAL, BOOLEAN, OID, INET, MONEY, composite, ANY,...
- ◆ Sono presenti diversi tipi di indici. Quello di default e' il BTREE



SQL (DML)

◆ SELECT

```
postgres=# select substring(version() from 1 for 16), current_user, now();
```

```
    substring    | current_user |          now
```

```
-----+-----+-----  
PostgreSQL 9.0.5 | postgres    | 2011-10-10 10:10:10.10101+02
```

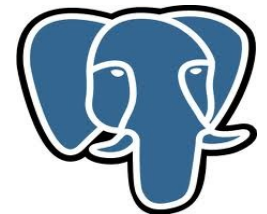
```
postgres=# select * from pg_tables;
```

...

```
postgres=# SELECT *
```

```
FROM weather LEFT OUTER JOIN cities ON (weather.city = cities.name);
```

...



SQL (DML)

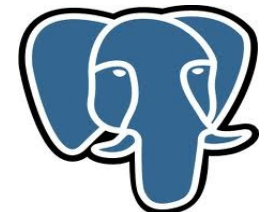
◆ SELECT, PL/pgSQL e TABLE FUNCTIONS

```
postgres=# select loc, count(*), sum(sal)
-> from emp, dept
-> where emp.deptno = dept.deptno
-> group by loc order by loc limit 5 offset 2;
...
```

```
postgres=# CREATE OR REPLACE FUNCTION increment(i integer) RETURNS integer AS
$$
    BEGIN
        RETURN i + 1;
    END;
$$ LANGUAGE plpgsql;
```

```
postgres=# create or replace function empDept(int) returns setof emp7 as $$
    select * from emp7 where deptno=$1;
$$ language SQL;
```

```
postgres=# select ename, increment(deptno) from empDept(10);
...
```



SQL (DML)

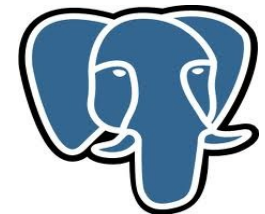
◆ INSERT, UPDATE, DELETE

```
postgres=# insert into votazione(scelta, voti) values ("pizza",1),  
("hot-dog",1);
```

```
postgres=# update emp set sal=sal+100;
```

```
postgres=# delete from emp where empno=10;
```

```
postgres=# COPY emp FROM '/home/user/emp.txt';
```



SQL (funzioni)

◆ PostgreSQL ha un insieme molto ampio di operatori, funzioni e clausole:

◆ Operatori:

AND, OR, NOT, BETWEEN, LIKE, SIMILAR, ~,, ...

◆ Funzioni su stringhe, numeri, date:

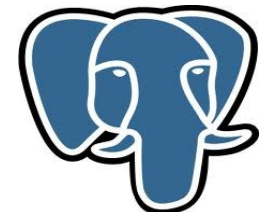
||, SUBSTRING(x FROM y FOR z), LENGTH, UPPER, LOWER, ... SIN, ROUND, ... NOW, pg_sleep()...

◆ Funzioni logiche:

CASE WHEN THEN, NULLIF(), GREATEST, LEAST

◆ Funzioni di gruppo:

COUNT(), SUM(), AVG(), HAVING, ...

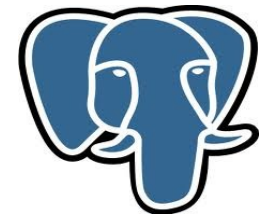


SQL (funzioni, variabili)

◆ PostgreSQL ha un insieme molto ampio di operatori, funzioni e clausole. Tipicamente ogni comando e' richiamato con una funzione:

◆ Funzioni di amministrazione:

```
pg_switch_xlog() pg_cancel_backend(pid)  
pg_terminate_backend(pid) current_setting(setting_name)  
set_config(setting_name, new_value, is_local)  
pg_reload_conf() pg_start_backup() ...
```



Altre funzionalita'

◆ Molti aspetti funzionali di PostgreSQL non sono stati riportati in questa breve presentazione:

◆ Linguaggio procedurale

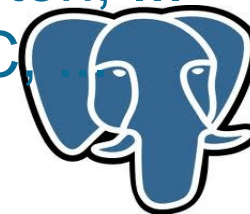
- PostgreSQL dispone di diversi linguaggi procedurali per la creazione di funzioni e la programmazione dei trigger.
- Il piu' utilizzato e' il PL/pgSQL

◆ Strumenti di amministrazione

- psql, pg_dump, pg_dumpall, pg_ctl, pgAdminIII, pg_restore, Replication, ...

◆ Interfacce di programmazione e connettori

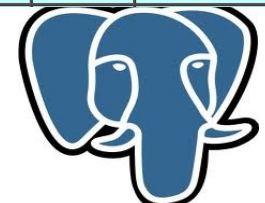
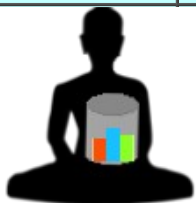
- Linguaggi: C, C++ via libpq; ecpg; Perl, PHP, Python, ...
- Interfacce: JDBC 3/4 type 4, DBD::Pg, plsqlODBC, ...



Storia

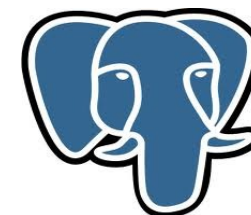
◆ Dal 1995 PostgreSQL ha avuto una notevole evoluzione:

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	Suggested
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	Suggested
PostgreSQL 12	Production	Performance improvements on partitioning, decreased WAL usage for indexes, JSON path expressions, stored generated columns, REINDEX CONCURRENTLY, ... Incompatibilities: 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	Suggested
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. Incompatibilities: 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	



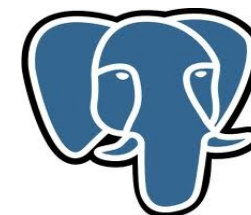
Storia

PostgreSQL 9.6	Production	Parallel sequential scans, joins and aggregates; Elimination old data autovacuuming; Synchronous replication on multiple standby servers; Full-text search for phrases; Support for remote joins, sorts, and updates in postgres_fdw; Better multi-core scalability. (9.6.1 2016-10) many bug fixes. Incompatible change: pg_stat_activity. (9.6.8 2018-03): CVE-2018-1058 fix. (9.6.20 2020-11): CVE-2020-25695 fix. This version and the previous ones are probably affected by CVE-2022-1552 published on 2022-05.	9.6.24	2016-09	2021-11	
PostgreSQL 9.5	Production	INSERT ON CONFLICT UPDATE (UPSERT), Grouping Sets, CUBE and ROLLUP analytic operations, Row-Level Security, better multi-core and large memory scalability. Desupport: PostGIS 2.1. (9.5.2 2016-03): fixed a bug affecting 9.5 only (text indexes optimization). (9.5.12 2018-03): CVE-2018-1058 fix. (9.5.24 2020-11): CVE-2020-25695 fix.	9.5.25	2016-01	2021-02	
PostgreSQL 9.4	Production	Native JSON support; new API for managing replication streams; logical WAL decoding; performance improvements on GIN indexes; concurrently updatable materialized views; ALTER SYSTEM (postgresql.auto.conf); pg_stat_archiver; security: SSL ECDH key exchange, log_connections shows SSL usage. (9.4.1 2015-02): fixed a bug affecting 9.4 only. (9.4.17 2018-03): CVE-2018-1058 fix.	9.4.26	2014-12	2020-02	
PostgreSQL 9.3	Production	Disk page checksum, updatable views, writeable foreign tables, parallel pg_dump, copy freeze, lateral join, materialized views, new functions for JSON, easier switchover with replication. Desupport: PostGIS 2.0. (9.3.2 2013-12) replication and PITR serious bug fixed. (9.3.4 2014-03) fixed a serious bug affecting 9.3 only. (9.3.22 2018-03) CVE-2018-1058 fix.	9.3.25	2013-09	2018-11	
PostgreSQL 9.2	Production	Cascading replication, on-slave backup; performance improvements (index-only scan); new SQL features (range data type, JSON data type). Incompatible change: pg_stat_activity. Desupport: PostGIS 1.5. (9.2.6 2013-12) replication and PITR serious bug fixed.	9.2.24	2012-09	2017-11	
PostgreSQL 9.1	Production	Synchronous replication; serializable snapshot; Extensions; per-column collation; unlogged tables; K-nearest-neighbor indexes; SE-Postgres; foreign data wrappers (DB gateway); SQL or PL/PgSQL: CREATE TABLE IF NOT EXISTS, INSTEAD OF triggers on views, FOREACH; ... (9.1.11 2013-12) replication and PITR serious bug fixed.	9.1.24	2011-09	2016-10	
PostgreSQL 9.0	Production	Streaming Replication and Hot Standby. GRANT/REVOKE IN SCHEMA for mass permission changes. Enhanced stored procedure support, DO with anonymous blocks, Python3. Dereferrable unique constraints for bulk load. 64 bit build for WinX. Desupport: PostGIS 1.4 (9.0.15 2013-12) replication and PITR serious bug fixed.	9.0.23	2010-09	2015-10	



Storia

PostgreSQL 8.4	Production	Performance enhancements (hash index, EXISTS), better free space management, windows functions (OVER clause), CTE (WITH clause), column permission, returns table in functions, ... (8.4.19 2013-12) replication and PITR serious bug fixed.	8.4.21	2009-07	2014-07	
PostgreSQL 8.3	Unsupported	Very stable. Full text search, XML, updatable cursors, performance&tuning (eg. asynchronous commit, dedicated writes, default autovacuum)	8.3.23	2008-02	2013-02	
PostgreSQL 8.2	Unsupported	Multirow DML, index DDL during DML (CONCURRENTLY clause), SQL:2003 statistical functions, performance&tuning (eg. fast locking, FILLFACTOR, monitor/logging)	8.2.23	2006-12	2011-12	
PostgreSQL 8.1	Unsupported	2PC, Role based security, IN/OUT/INOUT SQL function parameters, partitioning (constraint_exclusion param.), autovacuum in core distribution (it was an extension)	8.1.23	2005-11	2010-11	
PostgreSQL 8.0	Unsupported	MS Windows porting, savepoint, point-in-time recovery, tablespaces, Perl server-side language	8.0.26	2005-01	2010-10	
PostgreSQL 7.0	Unsupported	Foreign keys, SQL-92 joins, many optimizer enhancements; (7.1 2001-04) GiST, TOAST, Outer joins; (7.2 2002-04) not locking vacuum, column histograms in analyze, better password encryption; (7.3 2002-11) schemas, drop column, table functions, prepared queries; (7.4 2003-11) standard information schema, more efficient with: IN, GROUP BY, explicit JOIN, regular expressions, function-inlining, full-text	7.4.30	2000-05	2010-10	
PostgreSQL 6.0	Unsupported	JDBC 2.0, sqlcli.h, SET CONSTRAINTS, plperl	6.5.3	1997-01	2004-06	
1.0	Unsupported	PostgreSQL : SQL Engine	1.09	1996		
0	Unsupported	Postgres95 : SQL interpreter, ANSI C, psql, GNU make	0.03	1995		
4	Unsupported	"post" Ingres project at University of California at Berkeley: POSTGRES with Object-Relational features, PostQUEL, BSD make	4.2	1987		



Varie ed eventuali

◆ Link utili

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

Sito ufficiale PostgreSQL

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

Non ufficiale ma... in italiano!

