PostgreSQL val bene una Grappa!

3 Dicembre 2014 – GrappaLUG

GrappaLUG

Gruppo utenti GNU/Linux di Bassano del Grappa

Smart Solutions

Sul Ponte di Bassano, ci darem la mano!

Smart Solutions

Denis Gasparin

Senior DBA and Web Developer

- Sviluppo di soluzioni software basate su PostgreSQL
- Analista e Database Administrator
- Contributor del driver PDO PostgreSQL per PHP
- Socio di IT-PUG

PostgreSQL: chi è costui?

- **1 Maggio 1995**
  - Postgres95 V0.01
- **6 Major Release dal 1995**
  - PostgreSQL95
  - PostgreSQL 1.0
  - PostgreSQL 6, 7, 8, 9
  - 23 Minor Release
- **Una versione all'anno**

Pg(SQL) = S^3

- Nota funzione matematica coniata da...
  - Michael Stonebraker?
  - Andrew Yu e Jolly Chen?
  - … Denis Gasparin :-D

- **PostgreSQL** è la moltiplicazione di tre fattori:
  - **Semplicità**
  - **Scalabilità**
  - **Sicurezza**

Semplicità

- Pacchetti di installazione disponibili per
  - Tutte le maggiori distribuzioni Linux
  - MacOSX, Freebsd e... Windows!
  - Ormai presente su tutte le piattaforme Cloud (AWS, Heroku, OpenShift)
- Clients semplici e completi
  - Psql (riga di comando)
  - Pgadmin (grafico)
- Aderenza agli standard ANSI SQL 2008 e ACID
- Configurazione pronta all'uso
- Licenza

Semplicità(installazione)

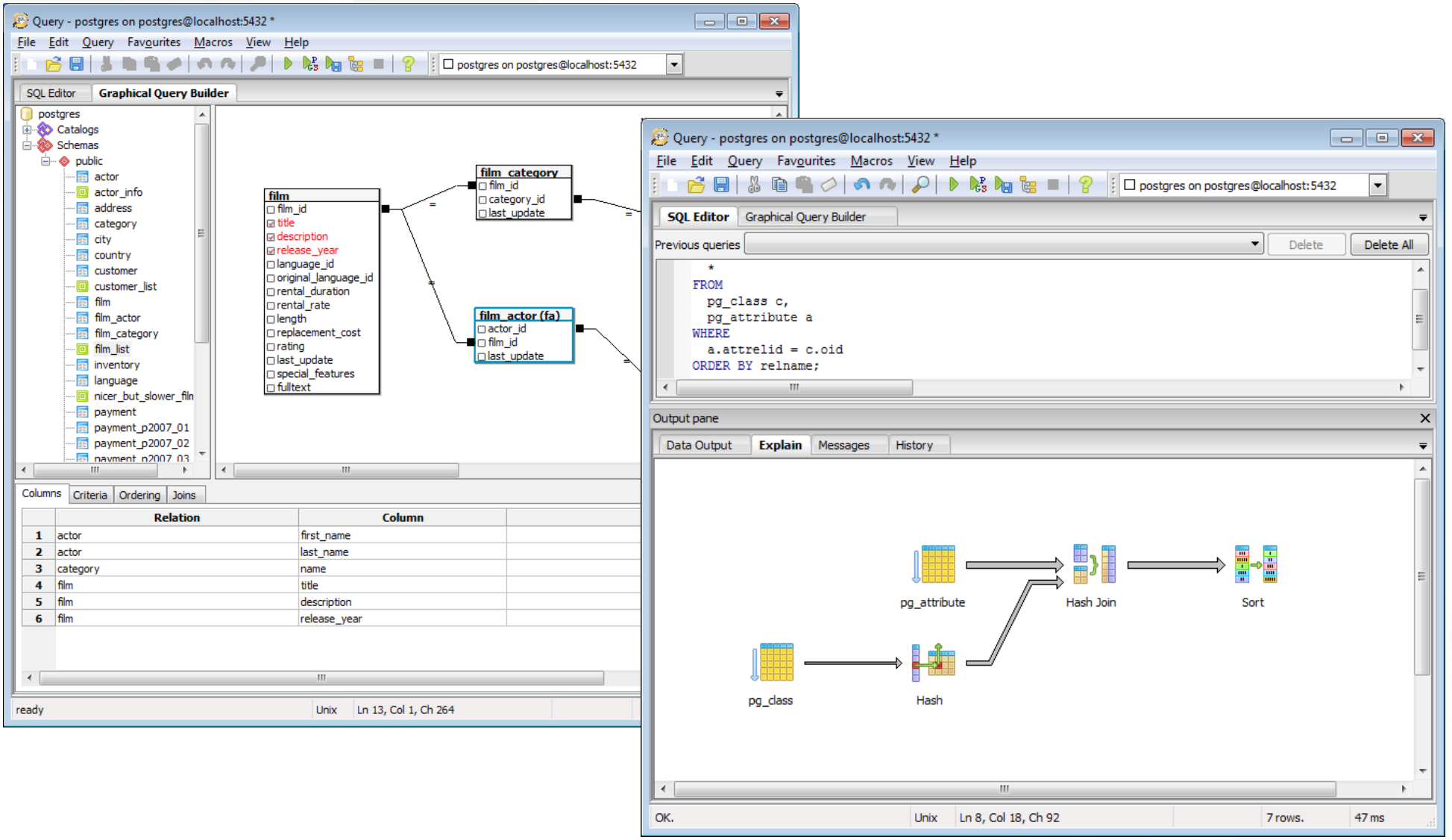
Esempio di installazione su Debian

```
$ sudo su -
$ vi /etc/apt/sources.list.d/pgdg.list

# Inserire questa riga nel file
deb http://apt.postgresql.org/pub/repos/apt/ wheezy-pgdg main

$ curl https://www.postgresql.org/media/keys/ACCC4CF8.asc |apt-key add -

$ apt-get update
$ apt-get install postgresql-9.3
$ su -l postgres
$ psql -U postgres template1
psql (9.3.5)
Digita "help" per avere un aiuto.

template1=# CREATE DATABASE pgsql_con_grappa;
CREATE DATABASE

template1=# \c pgsql_con_grappa
```

Semplicità(accesso&psql)

```
$ psql -U postgres pgsql_con_grappa
psql (9.3.5)
Digita "help" per avere un aiuto.

pgsql_con_grappa=# \i crea_tabella_produttori.sql
CREATE TABLE

pgsql_con_grappa=# \dt

           List of relations
 Schema |    Name    | Type  |  Owner
--------+------------+-------+----------
 public | produttore | table | postgres
(1 row)

pgsql_con_grappa=# SELECT * FROM produttore;

pgsql_con_grappa=# \x (abilita l'output espanso)

pgsql_con_grappa=# \set provincia 'Vicenza'  (imposta una variabile)

pgsql_con_grappa=# SELECT * FROM produttore WHERE provincia = :'provincia'
```

Semplicità(accesso&pgadmin)



Semplicità(licenza)

• PostgreSQL

PostgreSQL is released under the PostgreSQL License, a liberal Open Source license, similar to the BSD or MIT licenses.

PostgreSQL Database Management System (formerly known as Postgres, then as Postgres95)

Portions Copyright (c) 1996-2014, The PostgreSQL Global Development Group

Portions Copyright (c) 1994, The Regents of the University of California

Permission to use, copy, modify, and distribute this software and its documentation for any purpose, without fee, and without a written agreement is hereby granted, provided that the above copyright notice and this paragraph and the following two paragraphs appear in all copies.

IN NO EVENT SHALL THE UNIVERSITY OF CALIFORNIA BE LIABLE TO ANY PARTY FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, INCLUDING LOST PROFITS, ARISING OUT OF THE USE OF THIS SOFTWARE AND ITS DOCUMENTATION, EVEN IF THE UNIVERSITY OF CALIFORNIA HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

THE UNIVERSITY OF CALIFORNIA SPECIFICALLY DISCLAIMS ANY WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE SOFTWARE PROVIDED HEREUNDER IS ON AN "AS IS" BASIS, AND THE UNIVERSITY OF CALIFORNIA HAS NO OBLIGATIONS TO PROVIDE MAINTENANCE, SUPPORT, UPDATES, ENHANCEMENTS, OR MODIFICATIONS.

• MySQL

•http://www.gnu.org/licenses/gpl-2.0.html

•http://www.mysql.com/about/legal/licensing/foss-exception/

...Riassunto...

•MySQL can be used without cost if an application is locally developed and not used commercially. It is only when the resulting solution is to be sold to customers that the question of licensing comes into play. This rule is expressed on the MySQL home page as follows: **Free use for those who never copy, modify or distribute.**

•MySQL can be used freely within a web site. If you also develop a PHP application and install it with your Internet service provider, you do not have to make your PHP code freely available in the sense of GPL.

•Likewise, an Internet service provider may make MySQL available to its customers without having to pay MySQL license fees. (Since MySQL is running exclusively on the ISP computer, this application is considered internal.)

•Finally, MySQL license can be used free of charge for all projects that themselves run under the GPL or comparable free license.

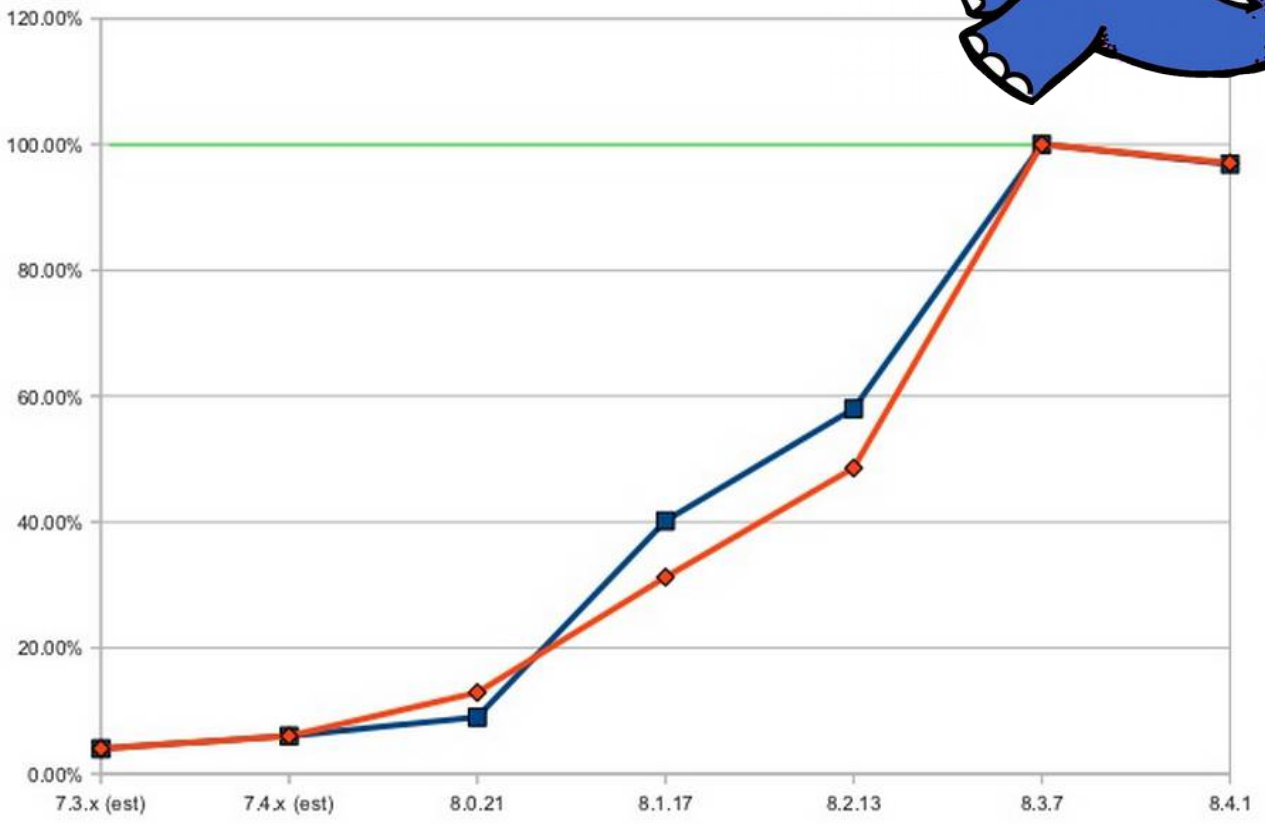
Scalabilità(performance)

- Scalabilità orizzontale
  - Data partitioning: partizionamento di tabelle nello stesso DB
  - Data sharding: partizionamento su server DB diversi
- Table spaces
- Hot standby(s)
  - Distribuzione query in sola lettura su più server DB
- Viste materializzate
- Indici GiST e GIN
  - Tosi... demose da fare... creemo gli indici GRAPPA!

Scalabilità(performance con spirito)

Altro che RedBull!

PgSql 9.3



PgSql 7.3

http://suckit.blog.hu/2009/09/29/postgresql_history

Scalabilità(applicazioni)

- MVCC: Multiversion Concurrency Control
- Transazioni:
  - anche su DDL (CREATE/DROP TABLE, ALTER TABLE, ...)
  - Savepoints (punto di ripristino all'interno di una transazione)
- Tipi di dato:
  - XML, JSON, Range, Array, Tipi geometrici, Tipi composti
- Query:
  - Window functions, CTE
- Estensioni:
  - Nuove funzionalità con un solo comando: http://pgxn.org/

Scalabilità(applicazioni)

- Motore Full Text Search
- Linguaggi procedurali:
  - PL/pgSQL, PL/Perl, PL/Python, C
- Table inheritance
- Subquery
- Espressioni regolari nelle query
- Schemi
- Triggers
- Integrità referenziale

Sicurezza(backup)

PostgreSQL offre almeno tre soluzioni di backup:

- **Backup Logico**: il classico dump SQL, pg_dump e pg_restore
  - Disaster recovery
  - Aggiornamento a nuova major release
- **Backup Fisico**: copia fisica del db con i log transazionali
  - Point in time recovery
- **Replica**: Master-Slave, anche in cascata
  - Riduzione dei tempi di ripristino in caso di crash

Barman: soluzione opensource per la gestione dei backup

Sicurezza(accesso ai dati)

- Gestione avanzata dei permessi utente
  - Utenti e gruppi di utenti (ruoli)
  - Permessi su singola colonna di una tabella

- Accesso al database:
  - Gestito dal file pg_hba.conf
  - Possibilità di limitare per indirizzo IP
  - Svariate tipologie di accesso (password, ldap, etc)
  - Possibilità di vincolare l'accesso via SSL

Pg(SQL) = S^3 : un esempio

DDL = Data Definition Language (comandi CREATE e DROP TABLE, etc)

PostgreSQL

```
db=# BEGIN;
BEGIN

db=# DROP TABLE produttore;
DROP TABLE;

db=# ROLLBACK;
ROLLBACK

db=# \dt
```

MySQL

```
mysql> BEGIN;
Query OK, 0 rows affected (0,00 sec)

mysql> DROP TABLE produttore;
Query OK, 0 rows affected (0,01 sec)

mysql> ROLLBACK;
Query OK, 0 rows affected (0,01 sec)

mysql> SHOW TABLES;
```

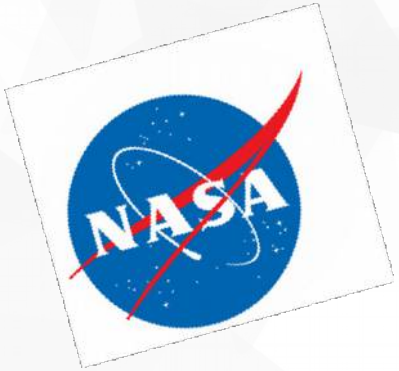
Che risultato vi aspettate?

```
              Lista delle relazioni
 Schema |      Nome      |  Tipo   | Proprietario
--------+----------------+---------+-----------
 public |   produttore   | tabella | postgres
```

```
Empty set (0,00 sec)
```

...ma non lo usa Nessuno!

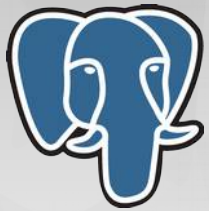
Ecco qualche esempio di "**Nessuno**"!



Courtesy of Gabriele Bartolini Keynote PgDay 2014

Ed in caso di Problemi? A chi mi rivolgo?

- **Comunità internazionale**
  - Mailing list: http://www.postgresql.org/list/
  - Canale IRC: http://www.postgresql.org/community/irc/
- In Italia esiste **IT-PUG**, fondato nel 2007
  - http://www.itpug.org
  - Organizza il PG-DAY Italiano
  - Mailing List in Italiano
- Società o **Professionisti** esperti:



Smart Solutions

Formazione ed assistenza

- Corsi di formazione
  - Per aziende o professionisti del settore
  - Personalizzati in base alle esigenze
- Consulenza su:
  - Performance
  - Sicurezza e Backup
  - Soluzioni architetturali complesse
- Corso organizzato in collaborazione con **Grappa**LUG

Un Elefante Intelligente

- **PostgreSQL** è la soluzione ideale, gratuita per la **Business Intelligence**
  - Si integra con tutte le soluzioni di BI esistenti via ODBC o JDBC
  - Funzionalità utili per la Business Intelligence
    - Partizionamento mediante inheritance
    - Viste materializzate
    - Indici Bitmap al posto di indici multicolonna
    - Tabelle Pivot
    - Query With
    - Window Functions

Esempio di Tabella Pivot!

```
 provincia | a_2013  | a_2012  | a_2011  | a_2010 | a_2009  | a_2008
-----------+---------+---------+---------+--------+---------+---------
 Padova    |  432822 |  452795 |  750852 | 748171 |  451423 |   80978
 Treviso   |  884996 |  855836 |  211019 | 371154 | 1861312 | 1301415
 Udine     |  766062 |  574152 |  940019 | 149345 |  459264 |   26562
 Verona    |  467864 |  467605 |  716269 | 458008 |   24425 |  623579
 Vicenza   | 1662862 | 1453165 | 2398290 | 625762 | 1727459 | 2151068
```

```
SELECT * FROM crosstab($$
   SELECT p.provincia, v.anno, SUM(v.totale)::integer
   FROM vendite v
   INNER JOIN produttore p ON(v.id_produttore=p.id)
   GROUP BY p.provincia,v.anno
   ORDER BY 1,2
$$)
AS ct(
    provincia TEXT,
    a_2013 INTEGER, a_2012 INTEGER,
    a_2011 INTEGER, a_2010 INTEGER,
    a_2009 INTEGER, a_2008 INTEGER
);
```

crosstab fa parte dell'estensione tablefunc

Un Elefante Cartografico

PostgreSQL, attraverso l'estensione **Postgis**, consente analisi cartografiche complesse:

- QuantumGIS: http://www.qgis.org
- uDig: http://udig.refractions.net
- OpenJump: http://www.openjump.org

```
pgsql_con_grappa=# SELECT ST_Buffer(the_geom, 30000) FROM italia WHERE citta =
'Bassano del Grappa';          – Ritorna un oggetto che rappresenta tutte le
                               – località distanti 30KM da Bassano

pgsql_con_grappa=# SELECT macroarea, ST_Union(the_geom) FROM italia GROUP BY
macroarea;                     – Ritorna 4 righe NORD, CENTRO, SUD, ISOLE

pgsql_con_grappa=# SELECT ST_Intersection(c.the_geom,f.the_geom), c.citta,
f.fiume FROM italia as c, fiumi_italia as f WHERE f.fiume = 'Brenta';
                                – Ritorna le città attraversate dal Brenta
```

Che bel Tipo sto Elefante: JSON

- **JSON:**
  - Javascript Object Notation: trasformazione in stringa di un oggetto JS
  - Principali utilizzi:
    - Chiamate AJAX di una pagina Web
    - Richieste e Risposte nei servizi REST
    - Molti database NO-SQL sono interrogati via JSON
- **JSON e PostgreSQL**
  - Due tipi:
    - Json
    - JsonB
  - Operatori specifici per l'accesso

Che bel Tipo sto Elefante: JSON

```
- Informazioni
{
    "portata": "dessert",
    "difficoltà": "media",
    "preparazione": 60,
    "cucina": "moderna"
}
- Ingredienti
{
    "Grappa": {"q": 20, "u": "ml"},
    "Pera": {"q": 1, "u": "q"},
    "Mela": {"q": 1, "u": "q"},
    "Pesca": {"q": 1, "u": "q"},
    "Chicchi d'uva": {"q": 8, "u": "q"},
    "Fogli di gelatina": {"q": 2, "u": "q"},
    "Sciroppo di zucchero": {"q": 100, "u":"ml"}
}

SELECT * FROM ricetta
   WHERE cast(informazioni->>'preparazione' AS integer) < 60;

SELECT * FROM ricetta WHERE ingredienti ? 'Grappa'; - JSONB versione 9.4
```

Window Functions

- Una window function è simile ad una funzione di raggruppamento
  - Effettua dei calcoli su un gruppo di righe (la cosiddetta finestra)
  - Non fa collassare le righe in un'unica riga
- La "finestra" viene specificata con le istruzioni "OVER" e "PARTITION BY"
- Sono disponibili tutte le funzioni di aggregazione più alcune aggiuntive:
  - row_number()
  - rank()
  - first_value()
  - last_value()
- http://www.postgresql.org/docs/9.3/interactive/functions-window.html

Window Functions

id	nome	dipartimento	salario
1	JOHNSON	ADMIN	18000.00
2	HARDING	MANAGER	52000.00
3	TAFT	SALES	25000.00
4	HOOVER	SALES	27000.00
5	LINCOLN	TECH	22500.00
6	GARFIELD	MANAGER	54000.00
7	POLK	TECH	25000.00
8	GRANT	ENGINEER	32000.00
9	JACKSON	CEO	75000.00
10	FILLMORE	MANAGER	56000.00
11	ADAMS	ENGINEER	34000.00
12	WASHINGTON	ADMIN	18000.00
13	MONROE	ENGINEER	30000.00
14	ROOSEVELT	CPA	35000.00

Window Functions

```
- confrontare lo stipendio di un dipendente rispetto alla media
- del suo dipartimento

SELECT nome, dipartimento, salario,
       avg(salario) OVER (PARTITION BY dipartimento)
FROM dipendente;
```

nome	dipartimento	salario	avg
JOHNSON	ADMIN	18000.00	18000.00
WASHINGTON	ADMIN	18000.00	18000.00
JACKSON	CEO	75000.00	75000.00
ROOSEVELT	CPA	35000.00	35000.00
GRANT	ENGINEER	32000.00	32000.00
ADAMS	ENGINEER	34000.00	32000.00
MONROE	ENGINEER	30000.00	32000.00
HARDING	MANAGER	52000.00	54000.00
GARFIELD	MANAGER	54000.00	54000.00
FILLMORE	MANAGER	56000.00	54000.00
HOOVER	SALES	27000.00	26000.00
TAFT	SALES	25000.00	26000.00
POLK	TECH	25000.00	23750.00
LINCOLN	TECH	22500.00	23750.00

Window Functions

```
– ordinare i dipendenti per stipendio descrescente nello stesso dipartimento

SELECT nome, dipartimento, salario,
       rank() OVER (PARTITION BY dipartimento ORDER BY salario DESC)
FROM dipendente;

    nome    | dipartimento | salario  | rank
------------+--------------+----------+------
 JOHNSON    | ADMIN        | 18000.00 |    1
 WASHINGTON | ADMIN        | 18000.00 |    1
 JACKSON    | CEO          | 75000.00 |    1
 ROOSEVELT  | CPA          | 35000.00 |    1
 ADAMS      | ENGINEER     | 34000.00 |    1
 GRANT      | ENGINEER     | 32000.00 |    2
 MONROE     | ENGINEER     | 30000.00 |    3
 FILLMORE   | MANAGER      | 56000.00 |    1
 GARFIELD   | MANAGER      | 54000.00 |    2
 HARDING    | MANAGER      | 52000.00 |    3
 HOOVER     | SALES        | 27000.00 |    1
 TAFT       | SALES        | 25000.00 |    2
 POLK       | TECH         | 25000.00 |    1
 LINCOLN    | TECH         | 22500.00 |    2
(14 righe)
```

Foreign Data Wrappers

- Sono estensioni che consentono di collegare Postgres ad altri database o sorgenti dati
- https://wiki.postgresql.org/wiki/Foreign_data_wrappers
- Esempio: file data wrapper

```
$ apt-get install postgresql-contrib-9.3

CREATE EXTENSION file_fdw;
CREATE SERVER filesystem_server FOREIGN DATA WRAPPER file_fdw;

CREATE FOREIGN TABLE scontrino_alcolico (
   data_ora timestamp NOT NULL,
   cassa text NOT NULL,
   importo numeric(7,2) NOT NULL,
   gradi numeric(7,1) NOT NULL
) SERVER filesystem_server
OPTIONS
(format 'text', filename '/tmp/scontrini.csv', delimiter E'\t', null '');
```

Viste Materializzate

- Le "Viste" in PostgreSQL sono una semplice riscrittura di una query

```
CREATE VIEW v_produttore AS SELECT * FROM produttore WHERE provincia='Vicenza';

EXPLAIN SELECT * FROM v_produttore;

                          QUERY PLAN
---------------------------------------------------------------
 Seq Scan on produttore  (cost=0.00..15.12 rows=2 width=168)
   Filter: (provincia = 'Vicenza'::text)
```

- Le "Viste materializzate" memorizzano i dati in una tabella:
  - Query di selezione molto più veloci
  - Non sono consentite operazioni di INSERT, DELETE, UPDATE
  - Possono essere aggiornate globalmente

Viste Materializzate

```
CREATE MATERIALIZED VIEW scontrino_materializzato
   AS SELECT * FROM scontrino_alcolico;

EXPLAIN SELECT * FROM scontrino_materializzato;
                          QUERY PLAN
-------------------------------------------------------------
 Seq Scan on scontrino_materializzato  (cost=0.00..17.20 rows=720 width=82)

– Un insert nella tabella originale non viene “visto” nella vista
materializzata

– Bisogna usare lo statemente “REFRESH”
REFRESH MATERIALIZED VIEW scontrino_materializzato;


SELECT
   (SELECT COUNT(*) FROM scontrino_materializzato)  AS scontrini_mater,
   (SELECT COUNT(*) FROM scontrino_alcolico) AS scontrini_alcolici;

\watch 5
```

Viste Materializzate e Foreign Data Wrappers

- Le viste materializzate consento di velocizzare moltissimo l'accesso a sorgenti dati esterne:
  - Importano i dati nativamente in PostgreSQL
  - Le viste materializzate sono indicizzabili
- Da PostgreSQL 9.4, le viste potranno essere aggiornate senza ottenere un LOCK esclusivo sulla vista materializzata

Previsioni meteo... in PostgreSQL!

- Usiamo in modo fantasioso*:
  - Foreign data wrapper www_fdw
  - Stored Procedures PL/PGSQL
  - Json
  - LATERAL
- Sarebbe bello se potessimo fare...

```
SELECT * FROM previsioni_meteo WHERE citta = 'Bassano del Grappa' AND data =
'2014-12-08';

     dt     |  min  |  max  | previsioni | umidita |         q
------------+-------+-------+------------+---------+--------------------
 2014-12-08 |   0.5 |   7.5 | Clear      | 0       | Bassano del Grappa
```

(*) Vedi anche Gianni Giolli – PostgreSQL Spaziale

Fantascienza? No, con PostgreSQL si può!

- No, non ho bevuto!

```
CREATE SERVER chiedile_a_bernacca FOREIGN DATA WRAPPER www_fdw
    OPTIONS (
              uri 'http://api.openweathermap.org/data/2.5/forecast/daily',
              response_deserialize_callback 'interpreta_previsioni'
            );

CREATE USER MAPPING FOR postgres SERVER chiedile_a_bernacca;

CREATE FOREIGN TABLE previsioni_meteo (
  data DATE,
  min text,
  max text,
  previsioni text,
  umidita text,
  q text
) SERVER chiedile_a_bernacca;

SELECT * FROM previsioni_meteo WHERE q = 'Bassano del Grappa';
```

Facciamo un giro dei Produttori di Grappa!

```
SELECT * FROM produttore, LATERAL previsioni(produttore.citta, '2014-12-08');

-[ RECORD 1 ]---+--------------------------------
ragione_sociale | Andrea Da Ponte
citta           | Conegliano Veneto
payoff          | Distillatori per vocazione
previsioni      | Rain, min:   5.2, max:   9.5

-[ RECORD 2 ]---+--------------------------------
ragione_sociale | Bonollo
citta           | Mestrino
payoff          | Grappa Of
previsioni      | Rain, min:   5.2, max:   9.5

-[ RECORD 3 ]---+--------------------------------
ragione_sociale | Bottega
citta           | Bibano di Godega di Sant'Urbano
payoff          | Vignaioli e Mastri Distillatori
previsioni      | Snow, min:  -1.5, max:   5.6
```

Grazie!

denis@gasparin.net

http://www.gasparin.net