**PostgreSQL**

**version 14**

Distribuito tramite il sito www.manualioracle.it

Copyright © 2024 Assi Loris

Qualsiasi abuso sarà perseguito e punito secondo i termini di legge.

Version: 1.6

Alcuni termini usati sono trademarks registrati dei rispettivi proprietari.

**Sommario**

Introduzione al Manuale 4

Cap. 1 - I Database 5

1.1 – I database 5

1.2 – DBMS e RDBMS 5

1.3 – L’SQL e le sue estensioni 6

1.4 – Approccio DTAP nei database 8

Cap. 2 - Prime nozioni relative a Postgres 10

2.1 – Introduzione 10

2.2 – Postgres Versions 12

2.3 – Cluster Database 12

2.4 – Struttura logica di un Cluster Database 13

2.5 – Struttura Fisica di un Cluster Database 14

2.6 – Accesso al database 14

2.7 – Oggetti Locali e Globali 15

2.8 – Case sensitive in Postgres 16

Cap. 3 - Installare e Disinstallare Postgres 17

3.1 – Installazione e Creazione di un cluster database: Intro 17

3.2 – Installazione e Creazione di un cluster db su Debian e Ubuntu 17

3.3 – Installazione e Creazione di un cluster db su Red Hat: Premesse 21

3.4 – Installazione e Creazione di un cluster db su Red Hat: Tutti i casi 22

3.5 – Spostare il cluster database 23

3.6 – Creare un altro cluster database nella stessa macchina 25

3.7 – Disinstallare Postgres: Tutti i casi 25

Cap. 4 - Startup e Shutdown del Cluster Database 29

4.1 – Introduzione al comando *pg\_ctl* e i files *postmaster.\** 29

4.2 – *pg\_ctl* e contenuto dei files *postmaster.\** 29

4.3 – Start e Stop del cluster db 33

4.4 – Start e Stop del cluster db tramite *systemctl command* 33

4.5 – Start e Stop del cluster db tramite pg\_ctl 34

4.6 – Start e Stop del cluster db tramite pg\_ctlcluster 36

Cap. 5 - User Interfaces 37

5.1 – Introduzione 37

5.2 – Il psql 37

5.3 – Comandi psql e Comandi SQL 38

5.4 – Casi particolari di Connessione al db 39

Cap. 6 - Internals e Client Interface 41

6.1 – L’Information Schema 41

6.2 – System Catalogs 42

6.3 – Overview sul DB physical storage e sul TOAST 43

Cap. 7 - Gestire i Database 46

7.1 – Creare un Database 46

7.2 – Rimuovere o Rinominare un Database 47

7.3 – Template Database 47

7.4 – Clonare un Database 48

Cap. 8 - Parametri del database 51

8.1 – Server configuration file 51

8.2 – PG\_SETTINGS 52

8.3 – Settaggio dei parametri del db 54

8.4 – SET SESSION 54

8.5 – ALTER SYSTEM 55

8.6 – Modifica del file postgresql.conf 58

Cap. 9 - File Postgres 59

9.1 – Collocazione dei File 59

9.2 – Tablespace e Datafile 60

9.3 – WAL e checkpoint 62

9.4 – Client Authentication e File pg\_hba.conf 63

9.5 – Vista *pg\_hba\_file\_rules* ed altre considerazioni 66

9.6 – Esempio di utilizzo di *pg\_authid, password\_encryption e* *pg\_hba.conf* 67

Cap. 10 - Role, User e Schema 69

10.1 – Database Roles e Users 69

10.2 – Troubleshooting relativi ad utenti e connessioni 70

10.3 – Role Attributes 71

10.4 – Role Membership 72

10.5 – Predefined Roles 73

10.6 – Gli Schemi 74

10.7 – Create e Drop Schema 76

10.8 – Schema Public e Schema Search Path 76

Cap. 11 - Miscellaneous 79

11.1 – Introduzione al Vacuum 79

11.2 – Autocommit 79

11.3 – Overview delle New Features in Postgres 14 80

## Introduzione al Manuale

 **Contenuto**

Argomenti trattati in questo manuale:

 - introduzione ai database relazionali

 - dettagli sul database Postgresql (detto anche Postgres).

Ci riferiamo a Postgres versione 14 su Linux (soprattutto Debian e Red Hat) ma quasi tutti gli argomenti qui trattati sono validi anche su altre piattaforme.

 **Audience**

Il presente manuale è rivolto ai tecnici informatici, agli studenti e a chiunque voglia capire che cosa sia un database e voglia conoscere le principali caratteristiche di Postgres.

 **Particolarità**

Il presente manuale è in italiano ma molti termini tecnici in esso contenuti sono in lingua inglese.

Abbiamo fatto tale scelta perché spesso a lavoro tali termini vengono detti in inglese e perché riteniamo più utile che i lettori del manuale conoscano tali termini nella lingua usata nella documentazione ufficiale.

Molti argomenti sono schematizzati al fine di fornire una comprensione ed una memorizzazione superiore.

Alcune abbreviazioni usate:

 db : database

 grantare : dare un grant

 restorare : fare il restore

 O.S. : *Operating System* (Sistema Operativo)

 *<data\_directory>* : valore del parametro Postgres chiamato *data\_directory*

 **Principali Versioni**

- 06.03.2024 : version 1.0

- 12.03.2024 : version 1.1 rilascio al pubblico

- 31.10.2024 : version 1.5 varie modifiche

**Disclaimer**

Non si fornisce alcuna garanzia relativamente al fatto che il presente documento sia privo di errori. Non si assume nessuna responsabilità sugli eventuali errori o danni derivanti dall’uso delle informazioni qui contenute.

## I Database

### 1.1 – I database

Il database è un archivio di dati usato per contenere e restituire informazioni correlate fra loro.

Tali dati sono gestiti sui server attraverso appositi software detti DBMS e sono organizzati tramite tecniche di “modellazione” che hanno i seguenti obiettivi:

- immagazzinare molti dati pur non avendo a disposizione un grande storage

- restituire facilmente e velocemente le informazioni pur non avendo a disposizione un server molto potente

I database hanno 2 tipi di struttura:

* + - Logica:

E’ rappresentata dai componenti che si possono vedere dentro il database (tabelle, indici, ecc..)

* + - Fisica:

Sono i file fisici che un database usa per immagazzinare i dati

Un aspetto importante relativo ai database è quello della **sicurezza:** i dati contenuti in un database possono essere molto importanti per una azienda per cui è importante proteggerli da:

- perdite di informazioni (accidentali o no). Per questo si usano le tecniche di Backup e Recovery

- lettura o modifica dei dati stessi da parte di persone non autorizzate. Per questo sul database si usano i GRANT ed altre tecniche atte ad evitare il cosiddetto *SQL injection*.

La maggioranza dei database ha una architettura composta da almeno 2 parti (chiamate anche *tier*):

- Server:

E’ la macchina in cui è installato il database.

- Client:

 E’ la macchina in cui si collega l'utente che vuole usare il database.

In questo manuale parleremo soprattutto del lato server di tale architettura.

Un altro aspetto che spesso è importante è il **performance tuning**: è l'insieme delle attività usate per ottimizzare le performance di un database.

Il *performance tuning* può ulteriormente essere suddiviso in:

..................

..................

..................

## Prime nozioni relative a Postgres

### A logo with a blue and black design  Description automatically generated2.1 – Introduzione

*PostgreSQL* (detto anche *Postgres*) è un *Object-Relational DBMS open source* nato nel 1996e supportato da diverse aziende, fra le quali una delle più importanti si chiama *2ndQuadrant.*

Come detto all'inizio del manuale, esiste anche una versione a pagamento di Postgres chiamata *EDB Postgres Advanced Server (EPAS)*, tale versione è gestita dall'azienda *EnterpriseDB.*

In questo manuale, descriveremo Postgres solo nella sua versione *open source*, in quanto è quella più diffusa.

Qui l’elenco delle aziende che danno supporto su Postgresql in Europa:

<https://www.postgresql.org/support/professional_support/europe/>

Postgres può girare nella maggior parte dei sistemi operativi Linux, Unix, Windows e supporta oggetti come:

* *foreign keys, joins, views, triggers*
* *stored procedures* scritte in una dozzina di linguaggi, incluso *Java, Perl, Python, Ruby, C/C++* e *PL/pgSQL* (quest’ultimo è il linguaggio nativo di *Postgres,* simile al *PL/SQL* di *Oracle*)
* *data types* come INTEGER, NUMERIC, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, TIMESTAMP e *binary large objects* (come pictures, sounds o video).

Un particolare tipo di dato esistente in Postgres è il *pg\_lsn :* è usato per contenere un LSN (*Log Sequence Number*) che è puntatore alla location del WAL.

Inoltre Postgres usa l' *ANSI-SQL standard* e supporta:

* *subquery*
* *auto-increment columns* attraverso le sequence
* LIMIT/OFFSET che permettono di ritornare *result set* parziali

PostgreSQL supporta anche features avanzate come:

*- Multi-Version Concurrency Control (MVCC)*

..................

..................

..................

## Installare e Disinstallare Postgres

### 3.1 – Installazione e Creazione di un cluster database: Intro

Puoi installare postgres su vari sistemi operativi:

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

Per quanto riguarda l'installazione su Unix/Linux, puoi farla in 2 modi:

* *source code distribution:*

E’ il metodo più "manuale", usato normalmente in sistemi come AIX e Solaris. Puoi farlo seguendo quanto indicato qui:

<https://www.postgresql.org/docs/14/install-short.html>

* *pre-packaged distribution:*

E’ il metodo che normalmente viene utilizzato ed è quello descritto in questo manuale.

Esso sfrutta il *package manager* (come RPM o Debian package) della tua distribuzione Linux.

In questo capitolo vedremo l’installazione di postgres in alcune delle distribuzioni Linux più diffuse:

* Debian
* Ubuntu
* Red Hat

Per dettagli sulla installazione su altre piattaforme diffuse:

..................

..................

..................

## Startup e Shutdown del Cluster Database

### 4.1 – Introduzione al comando *pg\_ctl* e i files *postmaster.\**

Il comando *pg\_ctl* (che vedremo in dettaglio in questo paragrafo e nei prossimi) può essere usato per startare, stoppare e controllare il *cluster database Postgres.*

Tale comando usa i seguenti file contenuti nella directory *<data\_directory>*:

* *postmaster.id*

Il *pg\_ctl* esamina questo file per determinare se il *cluster database* sta girando

* *postmaster.opts*

Se esiste questo file, il *pg\_ctl* (in *restart mode*) passerà il contenuto del file stesso come opzione del comando, a meno che non venga sovrascritto dall’opzione *-o* (*pg\_ctl -o ...*) .

Parte del suo contenuto può essere visto lanciando, con l’OS user *postgres*, il seguente comando:

pg\_ctl status

 Se questo comando non viene trovato allora in *Debian* puoi lanciare:

..................

..................

..................

## User Interfaces

### 5.1 – Introduzione

Alcune delle più comuni intefacce utenti usate per lavorare su Postgressono:

* **psql** :

E’ il tool più usato dai DBA. E’ particolarmente adatto se devi lanciare degli script o se sei già connesso al server Postgres(ad esempio tramite PuTTY).

Viene installato assieme all’installazione del *postgres server e* del *postgres client* e non è molto user-friendly.

* **pgAdmin** :

E’ molto usato da chi sviluppa codice sul database. E’ molto user-friendly e tipicamente è installato sulla macchina (client) in cui stai lavorando.

* **DBeaver** :

E’ molto usato da chi sviluppa codice sul database. E’ molto user-friendly e tipicamente è installato sulla macchina (client) in cui stai lavorando.

Può essere usato per accedere a vari RDBMS (Postgres, Oracle e altri)

* **IDE (IntelliJ, Eclipsed)** :

..................

..................

..................