**PostgreSQL**

**-**

**version 12**

Distribuito tramite il sito www.manualioracle.it

Copyright © 2021 Assi Loris

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

Version: 1.9

Alcuni termini usati sono trademarks registrati dei rispettivi proprietari.

**Sommario**

[Cap. 1 - I Database 5](#_Toc181995719)

[1.1 – I database 5](#_Toc181995720)

[1.2 – DBMS e RDBMS 5](#_Toc181995721)

[1.3 – L’SQL e le sue estensioni 6](#_Toc181995722)

[Cap. 2 - Prime nozioni relative a Postgres 9](#_Toc181995723)

[2.1 – Introduzione 9](#_Toc181995724)

[2.2 – Postgres Version e Cluster Database 10](#_Toc181995725)

[2.3 – Accesso al database 10](#_Toc181995726)

[2.4 – Il psql 11](#_Toc181995727)

[2.5 – Comandi psql e Comandi SQL 12](#_Toc181995728)

[2.6 – Casi particolari di Connessione al db 13](#_Toc181995729)

[Cap. 3 - Installare e Disinstallare Postgres 15](#_Toc181995730)

[3.1 – Installazione e Creazione di un cluster database 15](#_Toc181995731)

[3.2 – Creare un altro cluster database nella stessa macchina 16](#_Toc181995732)

[3.3 – Startup e Shutdown del cluster database 17](#_Toc181995733)

[3.4 – Creare e Rimuovere un Database 20](#_Toc181995734)

[3.5 – Disinstallare Postgres: Tutti i casi 21](#_Toc181995735)

[Cap. 4 - SQL Statement 24](#_Toc181995736)

[4.1 – Creare una Tabella 24](#_Toc181995737)

[4.2 – I constraint 24](#_Toc181995738)

[4.3 – Query e Subquery 25](#_Toc181995739)

[4.4 – DML sulle Tabelle 26](#_Toc181995740)

[4.5 – Esempi di Query 27](#_Toc181995741)

[4.6 – Le Viste 27](#_Toc181995742)

[4.7 – Indici 28](#_Toc181995743)

[4.8 – Utenti 29](#_Toc181995744)

[4.9 – Transazioni 30](#_Toc181995745)

[Cap. 5 - Database di Esempio 32](#_Toc181995746)

[5.1 – Le tabelle EMP e DEPT 32](#_Toc181995747)

[5.2 – Popolare EMP e DEPT 33](#_Toc181995748)

[5.3 – Interrogare EMP e DEPT 34](#_Toc181995749)

[Cap. 6 - SQL e PL/pgSQL in Postgres 36](#_Toc181995750)

[6.1 – SQL in Postgres 36](#_Toc181995751)

[6.2 – PL/pgSQL 36](#_Toc181995752)

[6.3 – JSON Types 37](#_Toc181995753)

[6.4 – Approfondimenti relativi al JSONB 40](#_Toc181995754)

[6.5 – JSON Functions e Operators 42](#_Toc181995755)

[Cap. 7 - Postgres: Server Administration 44](#_Toc181995756)

[7.1 – Settare i parametri del db 44](#_Toc181995757)

[7.2 – Esempi di modifica di parametri del database 46](#_Toc181995758)

[7.3 – PG\_SETTINGS e SET SESSION 48](#_Toc181995759)

[7.4 – Template Database 49](#_Toc181995760)

[Cap. 8 - File Postgres 51](#_Toc181995761)

[8.1 – Collocazione dei File 51](#_Toc181995762)

[8.2 – Datafile e Tablespace 51](#_Toc181995763)

[8.3 – Il Write-Ahead Logging (WAL) 53](#_Toc181995764)

[8.4 – Client Authentication e File pg\_hba.conf 54](#_Toc181995765)

[Cap. 9 - Role, User e Schema 57](#_Toc181995766)

[9.1 – Database Roles e Users 57](#_Toc181995767)

[9.2 – Role Attributes 58](#_Toc181995768)

[9.3 – Role Membership 59](#_Toc181995769)

[9.4 – Default Roles 60](#_Toc181995770)

[9.5 – Gli Schemi 61](#_Toc181995771)

[Cap. 10 - Performance & Tuning 64](#_Toc181995772)

[10.1 – Processi Postgres e System Catalog 64](#_Toc181995773)

[10.2 – Maintenance Tasks e vacuuming 65](#_Toc181995774)

[10.3 – Standard Unix Tool per monitorare il Database 65](#_Toc181995775)

[10.4 – L’EXPLAIN di una query 66](#_Toc181995776)

[10.5 – PG\_STAT\_STATEMENTS 68](#_Toc181995777)

[10.6 – LOG\_MIN\_DURATION\_STATEMENT 69](#_Toc181995778)

[10.7 – Statistics Collector 70](#_Toc181995779)

[10.8 – Viste relative alle Statistiche 72](#_Toc181995780)

[10.9 – Overview sul Partizionamento 73](#_Toc181995781)

[Cap. 11 - Backup e Recovery 75](#_Toc181995782)

[11.1 – Introduzione al SQL Dump 75](#_Toc181995783)

[11.2 – *pg\_dump, pg\_dumpall* e *pg\_restore* 75](#_Toc181995784)

[11.3 – Dump di grandi database 77](#_Toc181995785)

[11.4 – Esempi di Dump di tabelle 77](#_Toc181995786)

[11.5 – File System Level Backup 78](#_Toc181995787)

[11.6 – Continuous Archiving 78](#_Toc181995788)

[11.7 – Fare un Base Backup 81](#_Toc181995789)

[11.8 – Esempi di Backup configuration 82](#_Toc181995790)

[11.9 – Upgrade di un Cluster Postgres 83](#_Toc181995791)

[11.10 – Migrazione di un singolo database 84](#_Toc181995792)

[Cap. 12 - Barman 85](#_Toc181995793)

[12.1 – Backup fisico di Postgres tramite tool “esterni” 85](#_Toc181995794)

[12.2 – Il tool Barman 85](#_Toc181995795)

[12.3 – Barman: Design e Architettura 86](#_Toc181995796)

[12.4 – Barman: Scenari Tipici per il Backup 87](#_Toc181995797)

[12.5 – Barman: Installazione e Configurazione 89](#_Toc181995798)

[12.6 – Comandi Barman 90](#_Toc181995799)

[12.7 – General Commands 91](#_Toc181995800)

[12.8 – Server Commands 92](#_Toc181995801)

[12.9 – Backup Commands 97](#_Toc181995802)

[Cap. 13 - Standby database 99](#_Toc181995803)

[13.1 – Alta Disponibilità e Standby database 99](#_Toc181995804)

[13.2 – Standby database: Alta Disponibilità e Load Balancing 101](#_Toc181995805)

[13.3 – Standby database: Configurazioni esistenti 102](#_Toc181995806)

[13.4 – Standby database: dettagli sulla Streaming replication 103](#_Toc181995807)

[13.5 – Hot Standby 104](#_Toc181995808)

[13.6 – Failover 105](#_Toc181995809)

[13.7 – Cascade Replication 106](#_Toc181995810)

[13.8 – Replication slot 107](#_Toc181995811)

[Cap. 14 - Logical Replication 109](#_Toc181995812)

[14.1 – Introduzione alla Logical Replication 109](#_Toc181995813)

[14.2 – La Publication 110](#_Toc181995814)

[14.3 – La Subscription 111](#_Toc181995815)

[14.4 – Esempio di configurazione della Logical Replication 112](#_Toc181995816)

[Cap. 15 - Miscellaneous 114](#_Toc181995817)

[15.1 – Overview sul TOAST 114](#_Toc181995818)

[15.2 – System Catalogs 115](#_Toc181995819)

[15.3 – New Features in Postgres 12 116](#_Toc181995820)

Introduzione al Manuale

 **Contenuto**

Argomenti trattati in questo manuale:

 - introduzione ai database relazionali (con esempi di comandi SQL che puoi usare in Postgres)

 - dettagli sul database Postgresql (detto anche Postgres).

Ci riferiamo a Postgres versione 12 su Linux. Ma la maggioranza degli argomenti qui trattati sono validi anche per altre versioni Postgres.

 **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)

 **Principali Versioni**

- 19.10.2021 : version 1.0

- 25.10.2021 : version 1.1 rilascio al pubblico

**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*):

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

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

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

## Prime nozioni relative a Postgres

### 2.1 – Introduzione

*PostgreSQL* (detto anche *Postgres*) è un *Object-Relational DBMS open source,* supportato dall'azienda *2ndQuadrant.*

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

Qui uno schema riassuntivo delle differenze fra Postgres e "EDB Postgres":

 <https://www.enterprisedb.com/compare-postgres-databases>

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

Postgres può girare nella maggior parte dei sistemi operativi Linux, Unix, Windows.

I tool client più usati per interfacciarsi a Postgres sono:

* il client testuale *psql*
* il client grafico *pgAdmin*

*Postgres* 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* (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)

Inoltre Postgres usa l' *ANSI-SQL:2008 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:

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

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

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

## Installare e Disinstallare Postgres

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

Puoi installare postgres su vari sistemi operativi:

 https://www.postgresql.org/download

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

* *source code distribution:* è il metodo più "manuale". Puoi farlo seguendo quanto indicato qui:

https://www.postgresql.org/docs/12/install-short.html

* *pre-packaged distribution:* è quello che normalmente viene utilizzato. Esso sfrutta il *package manager* della tua distribuzione Linux.

In questo paragrafo descriviamo velocemente come installare una *pre-packaged distribution* come *Debian package* o *Ubuntu package.*

Supponiamo di essere in una macchina Ubuntu, connesso come *root,* ecco come puoi vedere la release del sistema operativo:

root@ubuntu:~# cat /etc/os-release

NAME="Ubuntu"

**VERSION="20.04.2 LTS (Focal Fossa)"**

ID=ubuntu

ID\_LIKE=debian

PRETTY\_NAME="Ubuntu 20.04.2 LTS"

VERSION\_ID="20.04"

HOME\_URL="https://www.ubuntu.com/"

SUPPORT\_URL="https://help.ubuntu.com/"

BUG\_REPORT\_URL="https://bugs.launchpad.net/ubuntu/"

PRIVACY\_POLICY\_URL="https://www.ubuntu.com/legal/terms-and-policies/privacy-policy"

VERSION\_CODENAME=focal

UBUNTU\_CODENAME=focal

Dunque segui quanto indicato in

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

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

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

## SQL Statement

I *SQL statement* sono le istruzioni SQL da lanciare sul database per fare delle interrogazioni, creare oggetti o utenti, droppare tabelle o indici, ecc... ecc..

Qui vediamo i *SQL statement* più semplici ed usati.

### 4.1 – Creare una Tabella

Nei database relazionali, i record in relazione fra loro vengono contenuti in tabelle.

Ad esempio, i record degli impiegati di una azienda potrebbero essere contenuti in una singola tabella «anagrafica» chiamata EMPLOYEE:

 CREATE TABLE employee

 ( empno INTEGER,

 name VARCHAR(30),

 surname VARCHAR(30),

 deptno INTEGER

 );

Così abbiamo creato una tabella che ha 4 colonne (dette anche "campi") di 2 tipi:

* VARCHAR(30): qui potrai inserire delle stringhe lunghe al massimo 30 caratteri
* INTEGER: qui potrai inserire solo numeri interi

Questi 2 tipi visti sono probabilmente i tipi più usati nei database relazionali. Ma esistono tanti tipi di dato che possono essere usati; qui eccone altri molto diffusi (tutti esistenti in Postgres):

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

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

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