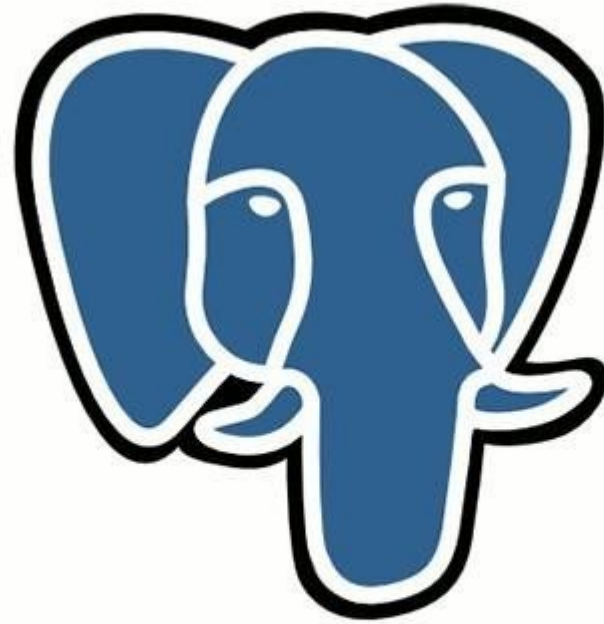


# Introducció a PostgreSQL



PostgreSQL

# Objectius

---



- Conèixer que ofereix la base de dades PostgreSQL
- Conèixer l'arquitectura de memòria i processos de PostgreSQL

# PostgreSQL

---

Per què **PostgreSQL**?

Ofereix una infraestructura (BD+Programació) **opensource** necessària per desenvolupar i gestionar aplicacions que altres SGBD no disposen.

- Escalable, robusta, fiable, disponible.
- Model de desenvolupament senzill
- Permet l'ús de SQL, PL/PGSQL, Perl, Python
- Interfície gràfica de gestió PgAdmin
- Similar a Oracle
- MySQL es veu en altres mòduls

# Introducció

---

- **PostgreSQL**, també anomenat Postgres, és un **sistema de gestió de bases de dades relacional orientat a objectes** i de **codi obert**.
- Està dissenyat per gestionar una gran quantitat de càrregues de treball, des de màquines individuals fins a magatzems de dades o serveis web amb molts usuaris concurrents.
- La primera versió va ser llançada al 1995 i des de llavors hi ha una versió mínima anual. A dia d'avui la darrera versió es la 14.

# Principal característiques

---

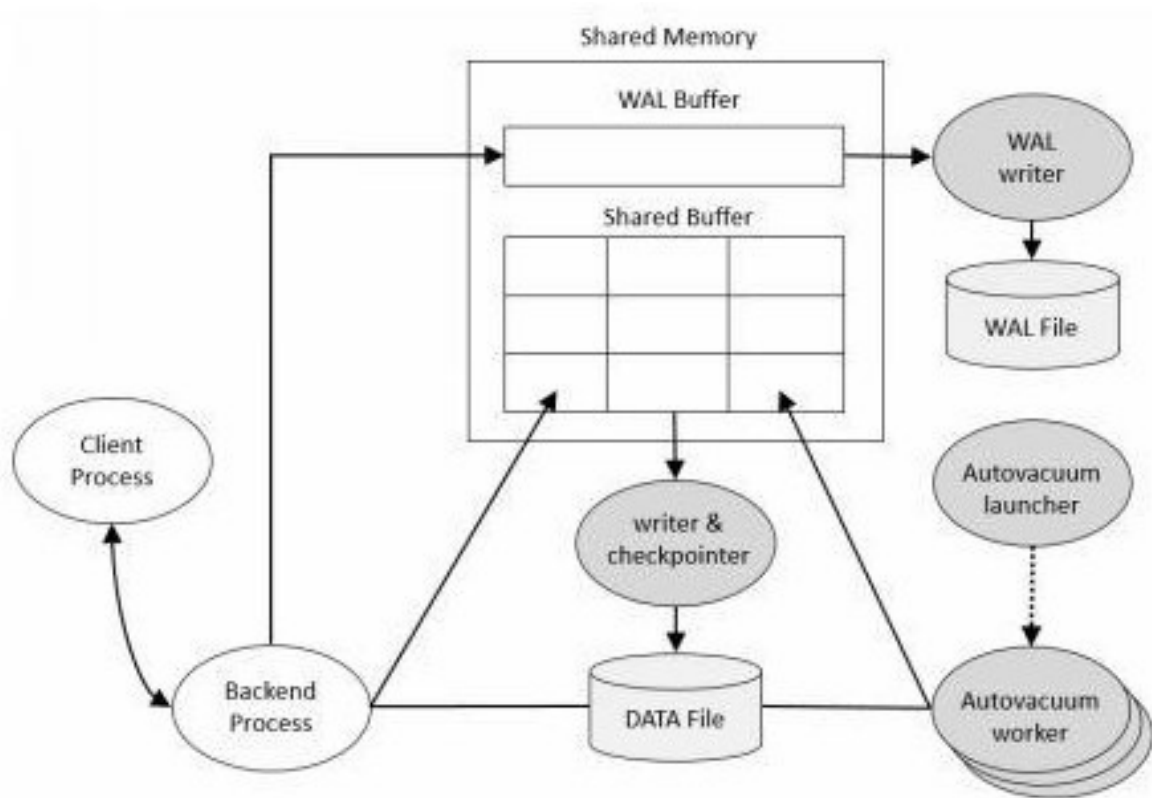
**Alta concurrència** : mitjançant un sistema denominat MVCC (Accés concurrent multiversió, per les seves sigles en anglès) PostgreSQL permet que mentre un procés escriu en una taula, d'altres accedeixin a la mateixa taula sense necessitat de bloquejos. Cada usuari obté una visió consistent.

**Àmplia varietat de tipus nadius**: números de precisió arbitrària, text de llarg il·limitat, figures geomètriques (amb una varietat de funcions associades), arrays,...

**Funcions**: pot fer servir diferents llenguatges un llenguatge propi anomenat PL/PgSQL (similar al PL/SQL de oracle), C, C++, Java PL/Java web, PL/Perl, pI PHP, PL/Python, PL/Ruby, PL/sh, PL/Tcl, PL/Scheme, llenguatge per a aplicacions estadístiques R per mitjà de PL/R.

# Arquitectura PostGreSql

- L'estructura física de PostgreSQL és molt senzilla. Consisteix en:
  - memòria compartida
  - processos de fons
  - fitxers de dades



# Arquitectura PostgreSQL: Memòria

---

- **Shared Memory** es refereix a la memòria reservada per a la memòria cau de la base de dades i la memòria cau del registre de transaccions. Els elements més importants de la memòria compartida són els buffers **Shared Buffer** i **WAL**
  - **Shared Buffer:** l'objectiu de Shared Buffer és minimitzar l'IO de disc. A aquest efecte, s'han de complir els principis següents:
    - Cal accedir ràpidament a buffers molt grans (desenes, centenars de gigabytes).
    - Hauríeu de minimitzar la contenció quan molts usuaris hi accedeixen alhora.
    - Els blocs d'ús freqüent han d'estar a la memòria intermèdia el màxim temps possible
  - **WAL Buffer:** és un buffer que emmagatzema temporalment els canvis a la base de dades. Els continguts emmagatzemats al buffer WAL s'escriuen al fitxer WAL en un moment predeterminat. Des del punt de vista de còpia de seguretat i recuperació, els buffers WAL i els fitxers WAL són molt importants.

# Arquitectura PostgreSQL: Processos

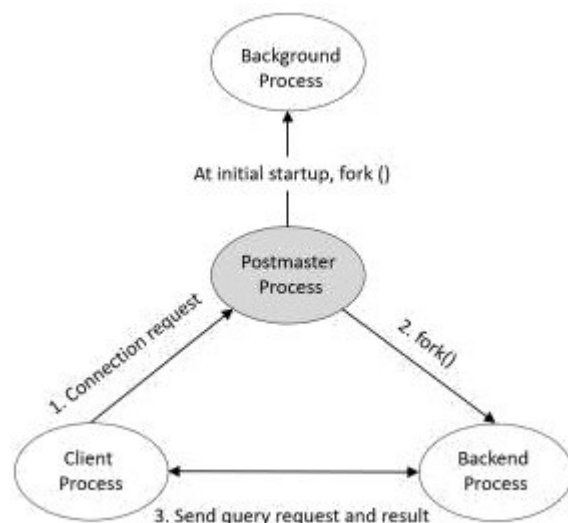
---

PostgreSQL té quatre tipus de processos.

- Postmaster (Daemon) Process
- Background Process
- Backend Process
- Client Process



# Arquitectura PostgreSQL: Procés Postmaster



```
$ pstree -p 1125
postgres(1125) /usr/local/pgsql/bin/postgres -D /usr/local/pgsql/data
├── postgres(1249) postgres: logger process
├── postgres(1478) postgres: checkpoint process
├── postgres(1479) postgres: writer process
├── postgres(1480) postgres: wal writer process
├── postgres(1481) postgres: autovacuum launcher process
├── postgres(1482) postgres: archiver process
└── postgres(1483) postgres: stats collector process
```

- El procés de **Postmaster** és el primer procés iniciat en iniciar PostgreSQL. A l'inici, realitza la recuperació, inicialitza la memòria compartida i executa processos de fons. També crea un procés de backend quan hi ha una sol·licitud de connexió del procés del client.
- Si comproveu les relacions entre processos amb la comanda pstree, podeu veure que el procés de **Postmaster** és el procés principal de tots els processos.

# Arquitectura PostgreSQL: Procés Background

- La llista de processos en segon pla necessaris per al funcionament de PostgreSQL són els següents:

Procés	Rol
logger	Escriu el missatge d'error al fitxer de registre.
checkpointer	Quan es produeix un punt de control, el buffer brut s'escriu al fitxer
writer	Escriu periòdicament el buffer brut a un fitxer.
wal writer	Escriu el WAL buffer en un fitxer WAL
Autovacuum launcher	Fork autovacuum worker when autovacuum is enabled. VACUUM recupera l'emmagatzematge ocupat per tuples esborrades. En el funcionament normal de PostgreSQL, els tuples que s'eliminen o obsolets per una actualització no s'eliminen físicament de la seva taula; romanen presents fins que es fa un VACUUM. Per tant, cal fer VACUUM periòdicament, sobretot en taules d'actualització freqüent.
archiver	Quan estigueu en mode Arxivat.log, copia el fitxer WAL al directori especificat.
stats collector	Es recopilen estadístiques d'ús de SGBD, com ara informació d'execució de sessions (pg_stat_activity) i informació estadística d'ús de taules (pg_stat_all_tables).

# Arquitectura PostgreSQL: Procés Backend

---

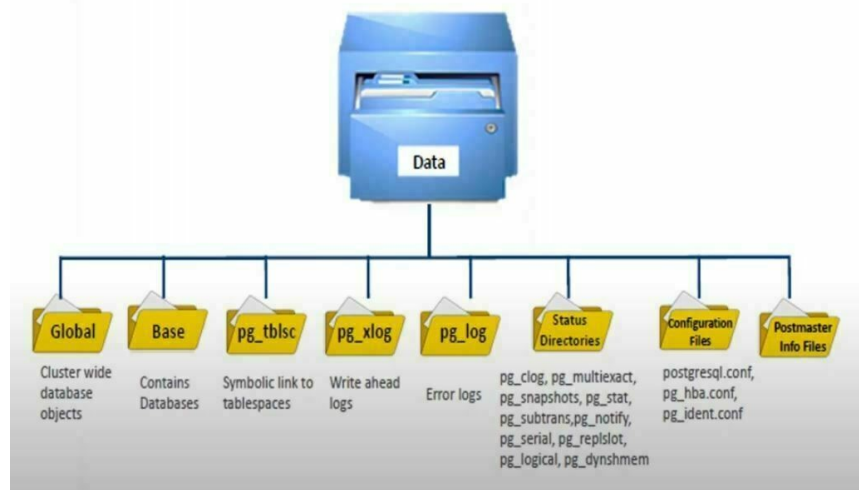
- El nombre màxim de processos de backend el defineix el paràmetre **max\_connections** i el valor per defecte és 100. El procés de backend realitza la sol·licitud de consulta del procés d'usuari i després transmet el resultat. Algunes estructures de memòria són necessàries per a l'execució de la consulta, que s'anomena memòria local. Els principals paràmetres associats a la memòria local són:
  - **work\_mem** Espai utilitzat per ordenar, operacions de mapa de bits, unions de hash i fusionar unions. La configuració predeterminada és de 4 MB.
  - **Maintenance\_work\_mem**: Espai utilitzat per Vacuum i CREAT ÍNDEX . La configuració predeterminada és de 64 MB.
  - **Temp\_buffers**: Espai utilitzat per a taules temporals. La configuració predeterminada és de 8 MB.

# Arquitectura PostgreSQL: Procés Client

---

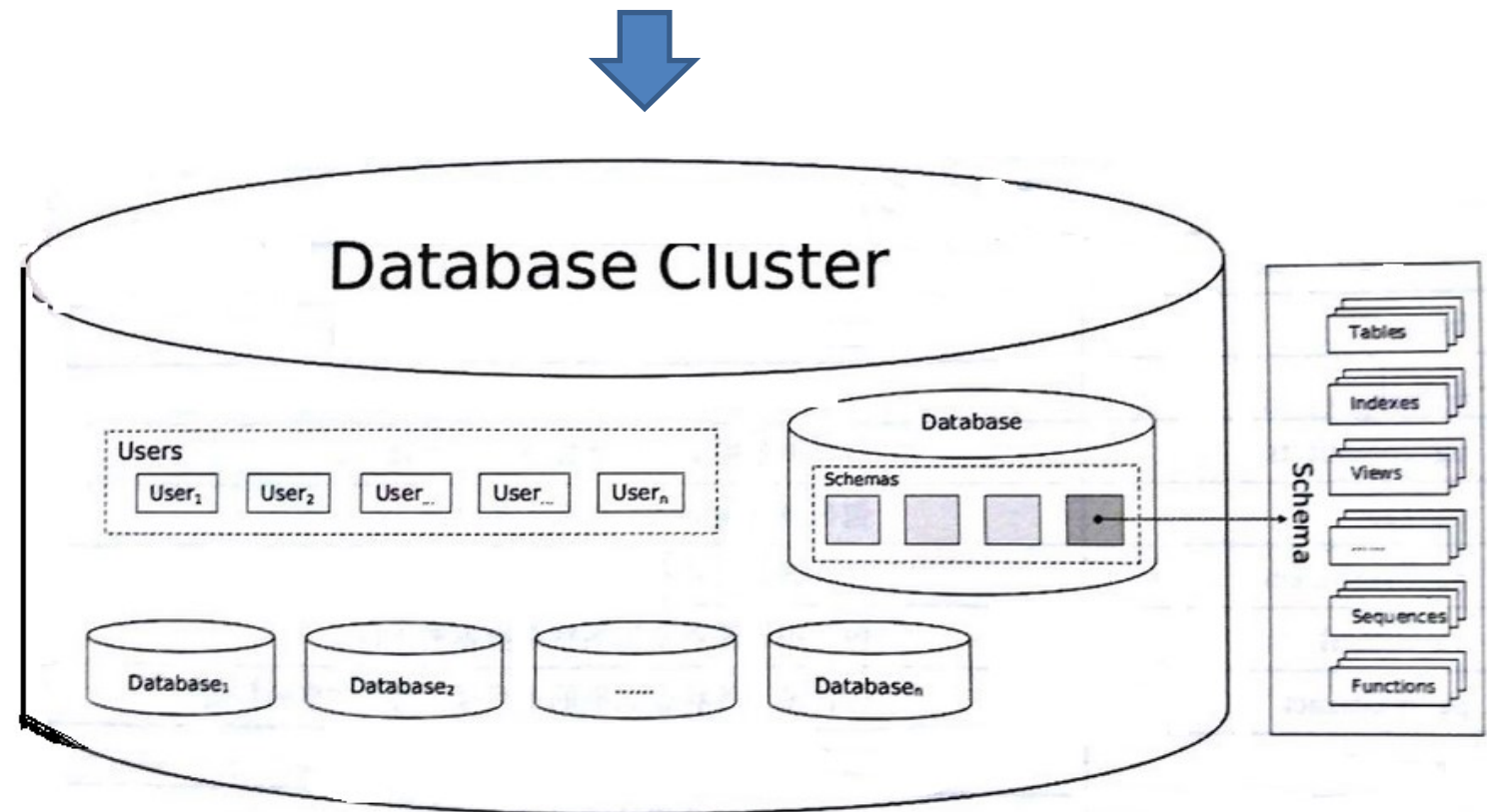
- El procés de client fa referència al procés en segon pla que s'assigna per a cada connexió d'usuari backend. Normalment, el procés de Postmaster bifurcarà un procés fill dedicat a servir una connexió d'usuari.

# Arquitectura PostgreSQL: Estructura BD



Arquitectura física

Arquitectura lògica



# Arquitectura PostgreSQL: Estructura BD

---

Aspectes important a saber quan s'intenta entendre l'estructura de la base de dades de PostgreSQL.

## **Elements relacionats amb la base de dades**

- PostgreSQL consta de diverses bases de dades. Això s'anomena clúster de bases de dades.
- Quan s'executa `initdb ()` es creen les bases de dades `template0` , `template1` , i `postgres`.
- Les bases de dades `template0` i `template1` són bases de dades de plantilles per a la creació de bases de dades d'usuaris i contenen les taules del catàleg del sistema.
- La llista de taules de les bases de dades `template0` i `template1` és la mateixa immediatament després d'`initdb ()`. No obstant això, la base de dades `template1` pot crear objectes que l'usuari necessiti.
- La base de dades d'usuaris es crea clonant la base de dades `template1`.

# Arquitectura PostgreSQL: Estructura BD-Tablespaces

---

## Elements relacionats amb el tablespace

- Els **pg\_default** i **pg\_global tablespaces** es creen immediatament després d'initdb().
- Si no especifiqueu un tablespace en el moment de la creació de la taula, s'emmagatzema al **pg\_default** tablespace.
- Les taules administrades a nivell de clúster de bases de dades s'emmagatzemen al **pg\_global** tablespace.
- La ubicació física del tablespace **pg\_default** és **\$PGDATAbase**.
- La ubicació física de **pg\_global** tablespace es **\$PGDATAglobal**.
- Un tablespace pot ser utilitzat per múltiples bases de dades. En aquest moment, es crea un subdirectori específic de base de dades al directori d'espai de taula.
- Creació d'un user tablespace crea un vincle simbòlic amb el user tablespace en el **\$PGDATAtblspc** directori.

# Arquitectura PostgreSQL: Estructura BD - Taules

---

## Elements relacionats amb la taula

- Hi ha tres fitxers per taula:
  - Un fitxer per emmagatzemar les dades de la taula. El nom del fitxer és l'OID de la taula.
  - Un fitxer per gestionar l'espai lliure de la taula. El nom del fitxer és `OID_fsm`.
  - Un fitxer per gestionar la visibilitat del bloc de taules. El nom del fitxer és `OID_vm`.
- L'índex no té un fitxer `_vm`. És a dir, `OID` i `OID_fsm` es componen de dos fitxers.



# WEBGRAFIA

---

- PostgreSQL: The world's most advanced open source database, Setembre 2022, <https://www.postgresql.org/>
- Tutorial Administració PostgreSQL, Setembre 2022, <https://www.postgresqltutorial.com/postgresql-administration/>
- Install PostgreSQL on Ubuntu Server, Setembre 2022, <https://ubuntu.com/server/docs/databases-postgresql>
- PostgreSQL: Documentation: 14: Part III. Server Administration, Setembre 2022, <https://www.postgresql.org/docs/current/admin.html>
- Postgres Architecture , Serveral9s, Setembre 2022, <https://severalnines.com/blog/understanding-postgresql-architecture/>