

Almacenes y Minería de Datos

- Prácticas
- 30 horas
- Requisitos previos:
Bases de Datos I, II



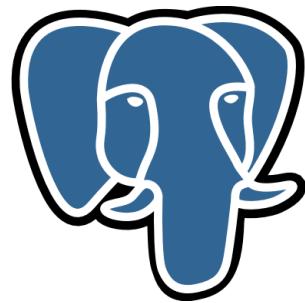
Joaquín Ángel Triñanes Fernández

Instituto de Investigaciónes Tecnológicas

Joaquin.Trinanes@usc.es

Ext: 16001

Externo: 881816001



Entorno

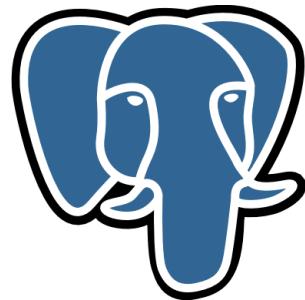
Motivación

- En esta primera parte emplearemos PostgreSQL.
- Proyecto Open Source independiente.
- Especialmente adaptado para el entorno de almacenes de datos (a pequeña-mediana escala).
- Muy estable y con un buen optimizador de consultas.
- Permite el tratamiento de datos geo-espaciales.
- Integración con múltiples lenguajes de programación y compatible con múltiples herramientas OLAP.
- FDW. Podemos extenderlo con nuevos tipos de datos y funciones.
- Podemos usarlo en la nube. Ejemplo: Amazon RDS, Google cloud SQL, Microsoft Azure, ...
- Nube: DBaaS, contenedores y Kubernetes, VMs.
- Amazon Redshift está basado en PostgreSQL 8



Redshift vs PostgreSQL

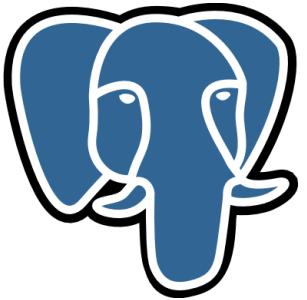
DIFERENCIAS PRINCIPALES



- Datos ordenados en columnas Vs datos ordenados en filas. En Redshift cada columna tiene un fichero asociado
- Los almacenes columnares pueden representar una ventaja para consultas analíticas mientras que los almacenes en filas pueden ser ventajosos para análisis exploratorios
- Uso de todos los recursos computacionales para una consulta. Capacidad de manejar operaciones de entrada salida masiva es mayor
- PostgreSQL permite el acceso a vistas materializadas. Redshift necesita add-ons.
- Redshift opera en entornos clúster. Escala mucho mejor.
- SQL: Similares pero presentan algunas diferencias
- Postgresql soporta un mayor número de tipos de datos

<https://blog.panoply.io/redshift-vs-postgresql>

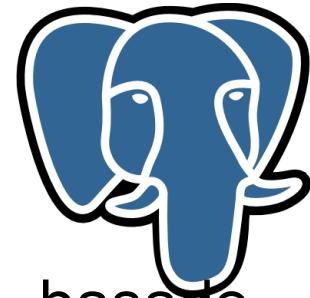
<https://www.flydata.com/blog/redshift-vs-postgres>



PostgreSQL en DW

Pequeña-mediana escala

- En muchos casos el sistema de almacenamiento de un almacén de datos es una base de datos relacional
- Una tabla calendario es una tabla que consta de unas fechas y los componentes de las mismas. Algunos de esos componentes pueden ser difíciles de computar usando SQL. (Ej. días festivos)
- Es muy útil para diseccionar los resultados por día de semana, semana del año, etc
- Suelen ser estáticas y de solo lectura. Se precargan a la granularidad deseada. Con granularidad diaria son pequeñas pero con granularidad por minuto, son mucho mas grandes.
- Son necesarias para estudiar los cambios en el tiempo
- PostgreSQL permite crear nuevos tipos y dominios y procedimientos almacenados



PostgreSQL en DW

- PostgreSQL se integra con múltiples herramientas de minería de datos
- Espacios de tablas: permiten almacenar en diferentes localizaciones, fomenta la escalabilidad y rendimiento.
- Podemos escalar horizontalmente, usando una característica de la tabla de hechos usando herencia
- Particionado vertical, basado en columnas
- Permite esquemas y vistas
- Funciones ventana y consultas WITH
- Permite claves subrogada (normalmente secuencias)
- Incluye vistas materializadas
- Tablas puente de jerarquías

Descripción

Tarea Inicial:

Instalación de PostgreSQL

Puede ser en máquinas virtuales

Ej. OS: CentOS

Basado en RHEL

Código Abierto y gratis

Muy estable (FC puede ser bastante inestable)

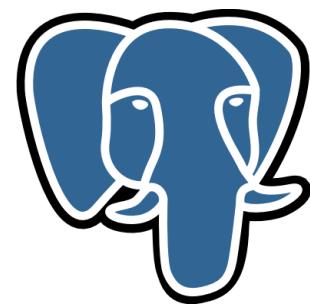
No hay apoyo “oficial” en caso de problemas

Yum (Yellowdog Updater, Modified), dnf

En Debian y derivados (ej. Ubuntu): apt-get/aptitude

Mac: Homebrew, Macports

Windows: EDB installer

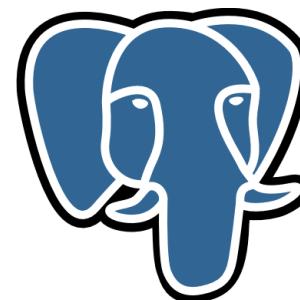


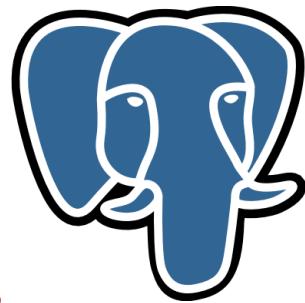
CentOS version	Release date	Full updates	Maintenance updates
3	2004-03-19	2006-07-20	2010-10-31
4	2005-03-09	2009-03-31	2012-02-29
5	2007-04-12	2014-01-31	2017-03-31
6	2011-07-10	2017-05-10	2020-11-30
7	2014-07-07	2020-08-06	2024-06-30
8	2019-09-24	2024-05	2029-05-31

Legend: Old version (Red), Older version, still maintained (Yellow), Latest version (Green)

PostgreSQL

- **yum install postgresql-server**
- Como usuario postgres:
 - ➊ Inicializa los ficheros en la base de datos: initdb
 - ➋ pg_ctl –D /var/lib/pgsql/data start
 - ➌ createdb test
 - ➍ psql test
 - ➎ \q



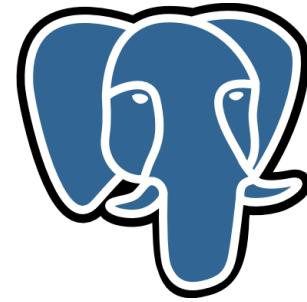


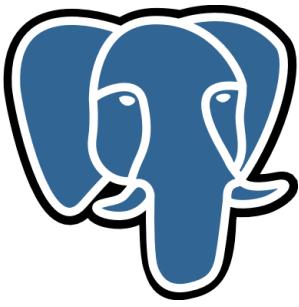
PostgreSQL

- Podemos instalar desde código fuente:
 - www.postgresql.org v14
 - Creamos el Makefile, compilamos e instalamos:
 - configure, make, make install
 - Directorios:
 - bin: Ejecutables
 - lib: librerías estáticas y dinámicas
 - doc: documentos
 - include: ficheros de cabecera
 - share: plantillas de inicialización
- **Creamos usuario postgres**
 - adduser postgres
 - En /etc/passwd:
postgres:x:1001:1001::/usr/local/pgsql/data:/bin/bash
 - En /etc/group
postgres:x:1001:
- **Localización de las bases de datos:**
 - mkdir -p /usr/local/pgsql/data ;
 - chown postgres /usr/local/pgsql/data
 - Inicializar el clúster de BD (como postgres):
initdb –D /usr/local/pgsql/data

PostgreSQL

- **Ficheros de configuración:**
- **postgresql.conf** - Fichero de configuración principal
- **pg_hba.conf** – Mecanismos de autenticación de clientes
- **pg_ident.conf** – En caso de emplear acceso de tipo ident en el fichero anterior





PostgreSQL

- **Variables de entorno .bash_profile**

```
PGDATA=/usr/local/pgsql/data
```

```
PATH=/usr/local/pgsql/bin:$PATH
```

```
export PATH
```

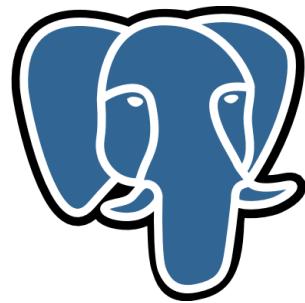
```
export PGDATA
```

- Inicializamos el servidor:

```
pg_ctl start (con systemctl, sudo systemctl start servicio_postgresql)
```

Una sesión de PostgreSQL tiene varios procesos:

- ⊕ **postgres** – proceso supervisor que lanza otros procesos y responde a conexiones de usuarios
- ⊕ Un proceso de usuario (como psql) para lanzar consultas
- ⊕ Procesos lanzados por postgres para atender las peticiones de datos de los usuarios
- ⊕ Los procesos anteriores se comunican con cada otro usando semáforos y memoria compartida



PostgreSQL

- **Almacenamiento columnar: cstore_fdw**

En ocasiones, resulta deseable acceder por columnas en lugar de por filas

Imaginémonos una tabla con 200 columnas y consultas del tipo:

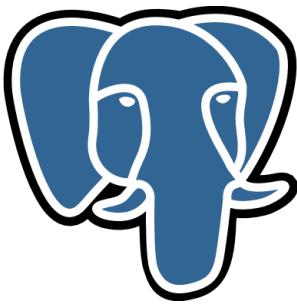
```
SELECT AVG(edad), MIN(edad), MAX(edad) FROM población WHERE nacimiento > '1980-01-01'
```

GROUP BY residencia;

En el almacenamiento por filas, habrá una cantidad muy superior de operaciones de I/O que en una almacenamiento por columnas, donde solo accederíamos a las columnas: edad, población, nacimiento y residencia.

El factor de compresión es mucho mayor

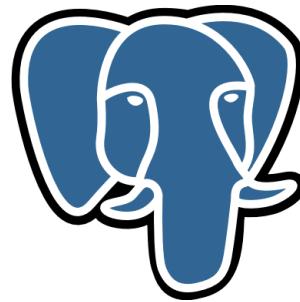
https://github.com/citusdata/cstore_fdw



PostgreSQL

● Importar datos en PostgreSQL

1. Vamos a realizar un primer ejercicio en el que vamos a importar los datos del Censo de población y viviendas 2001 del IGE. Implementa y describe los métodos que puedes utilizar para importar estos datos: <http://www.ige.eu/igebdt/esqv.jsp?paxina=002001&c=-1&ruta=parroquias/parroquias.jsp>
2. ¿Cuales son las 4 parroquias con más habitantes de la provincia de Pontevedra?
3. Muestra, para cada provincia, los municipios con población superior a 15000 habitantes.
4. Instala cstore_fdw. Habilita esta extensión, importa los registros y compara los tamaños de las tablas resultantes.
5. Realiza consultas sobre ambas tablas y observa los tiempos (podemos usar EXPLAIN ANALYZE). Intenta explicar los resultados.
6. Vamos a instalar la BBDD Adventure Works: <https://github.com/lorint/AdventureWorks-for-Postgres>
Explica lo que has hecho y cómo has resuelto los posibles problemas que hayan podido aparecer.
Describe las diferentes secciones del fichero install.sql y su propósito.



PostgreSQL

- Importar datos en PostgreSQL

7. <https://github.com/jonathanDuenas/AdventureWorksDW-PostgreSQL>

Importadla y realizar un par de consultas que tengan sentido analítico.

Analizad la estructura de esta BBDD comparándola a la que vosotros habéis importando en el paso anterior.

PRÁCTICA INDIVIDUAL

Los resultados de la práctica será un archivo detallado en el que debe figurar:

Nombre y número de horas fuera de horas de clase dedicadas a la práctica. Incluiremos en ese archivo una copia de la interacción con la base de datos. Se valorará la claridad expositiva del conjunto de la práctica, así como la presencia de posibles alternativas en el caso de poder realizar el ejercicio de múltiples formas. Plazo de presentación: 2 Oct. Se penalizará a los trabajos presentados fuera de plazo. Medio de entrega: Campus virtual.