

Administración y Aspectos internos de una BBDD



Entornos del Proceso de Desarrollo



DESARROLLO

En donde se desarrolla la aplicación. Puede haber diferentes opciones: el propio ordenador del programador o incluso un servidor compartido por los desarrolladores para que creen la aplicación.

Este entorno debe parecerse en gran medida al entorno de producción.



PRUEBA / QA

En donde se realizan las pruebas de lo que hayamos desarrollado. Este entorno debe parecerse al máximo al entorno de producción.

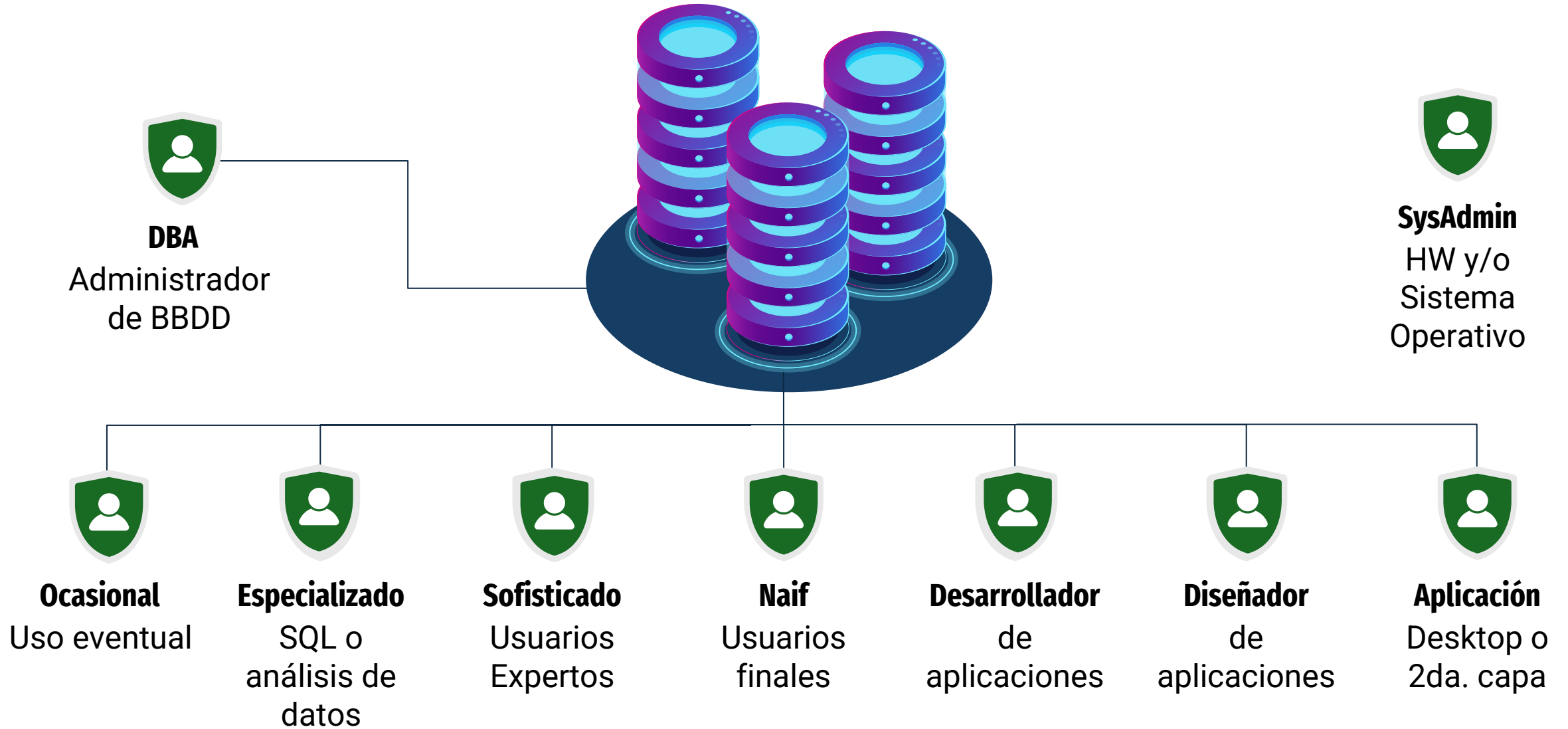
Aquí los desarrolladores y/o usuarios reales realizan pruebas para asegurarse de que los requisitos de desarrollo de software se han cumplido, y se consigue la funcionalidad tal y como se ha pedido y el software es completamente usable.



PRODUCCIÓN

Es accesible a todos los usuarios finales. Si hemos configurado todos nuestros entornos de la misma manera, realizado pruebas exhaustivas del software, tests automatizados y seguido buenas prácticas, no deberíamos tener ningún problema en el despliegue. Y si lo tuviéramos, simplemente tendríamos que comenzar de nuevo el ciclo de desarrollo: código, pruebas y despliegue.

Tipos de usuarios de una BBD



Tipos de usuarios de una BBD

NAIF

Los usuarios NAIF son usuarios finales que no saben mucho sobre tecnología, ni SQL, y usan el sistema llamando a uno de las aplicaciones que ya se han desarrollado. Por ejemplo, un cajero de banco seleccionará la opción del menú “Transferir” de la aplicación bancaria para mover \$15.000 de la cuenta A a la cuenta B. Este proceso le pide al cajero la cantidad de dinero que se debe mover, la cuenta de la que proviene el dinero y la cuenta a la que se destinará el dinero.

OCASIONAL

Los usuarios con conocimientos de lenguaje SQL que por alguna razón se les permite conectarse al servidor de BBDD para explorarlo y/o consultar, durante breves períodos, y con permisos de acceso limitados, en BBDD no críticas.

ESPECIALIZADOS

Son usuarios avanzados que escriben programas de bases de datos que no se adaptan a la forma tradicional de procesar datos. Entre estas aplicaciones se encuentran los sistemas de diseño asistido por computadora, las bases de conocimiento y los sistemas expertos, los sistemas que almacenan datos con tipos de datos complejos (como datos audiovisuales) y los sistemas que modelan el entorno.

Tipos de usuarios de una BBD

SOFISTICADOS

Saben cómo usar el sistema de BBDD sin escribir programas. Este grupo está formado por analistas que utilizan consultas para ver los datos de la BBDD. Por ejemplo, las herramientas de procesamiento analítico en línea (OLAP) facilitan el trabajo de los analistas al permitirles ver los resúmenes de datos de diferentes maneras. Por ej., los analistas pueden ver el total de productos por categorías, por número de ventas o por una combinación de categorías y número de ventas.

DESARROLLADOR

Profesionales informáticos que escriben aplicaciones para interactuar con las BBDD. Los programadores de aplicaciones pueden elegir entre muchas herramientas para desarrollar interfaces de usuario. Tienen distintos niveles de permisos dependiendo del entorno de desarrollo al que estén accediendo.

DE APLICACIÓN

Usuarios que sólo se utilizan para que una aplicación se conecte a las BBDD. Sólo tiene acceso desde el equipo en que se ejecute dicha aplicación, y tiene permisos determinados por su funcionalidad. Se los utiliza configurando su acceso en la cadena de conexión de la app.

Tipos de usuarios de una BBD

DISEÑADOR

Usuario que colabora durante el proceso de análisis y diseño de la aplicación, generando desde los modelos conceptuales hasta el físico para dicha aplicación.

Durante la etapa de desarrollo interactúa con la BBDD para crear sus esquemas y objetos hasta conseguir el definitivo. Luego lo entrega al DBA para que se implemente en los entornos de TEST y PRODUCCIÓN.

DBA

El Administrador de Base de Datos es una persona responsable de administrar y mantener el sistema de base de datos en una organización, garantizando que la base de datos sea accesible, segura y optimizada para el rendimiento. Desempeñan un papel crucial en el control y la supervisión de todos los aspectos de una base de datos, incluida su estructura, almacenamiento de datos, controles de acceso, copias de seguridad, recuperación y ajuste del rendimiento.

SYSADMIN

El Administrador de Sistemas interactúa con el DBA para brindar el apoyo necesario en lo que respecta a los requerimientos del Sistema Operativo, el hardware y las estructuras de almacenamiento y memoria necesarias para que el DBMS funcione de manera óptima.

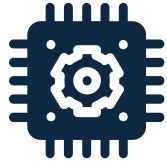
Funciones del DBA

- Instalación y actualización del servidor de BDD y las herramientas de aplicación.
- Asignación de almacenamiento del sistema y planificación de los requisitos de almacenamiento futuros para el sistema de base de datos.
- Creación de estructuras de almacenamiento de base de datos primarias (tablespaces).
- Creación de objetos primarios (tablas, vistas, índices, etc.) a partir de un modelo físico generado durante el proceso de desarrollo de la aplicación (Diseñadores).
- Modificación de la estructura de la base de datos, según sea necesario.
- Gestión de usuarios y mantenimiento de la seguridad del sistema (Adm. de Seguridad).
- Control y supervisión del acceso de los usuarios a la base de datos (Adm. de Seguridad).
- Supervisión y optimización del rendimiento de la base de datos.
- Realización de copias de seguridad y restauración de la base de datos.
- Mantenimiento de datos archivados en almacenamientos externos (cintas, storage, etc.).

Tareas comunes de un DBA

- Evaluar el hardware del servidor de base de datos.
- Instalar el software de base de datos Oracle.
- Planificar la base de datos.
- Crear y abrir la base de datos, sus esquemas y objetos.
- Realizar una copia de seguridad de la base de datos completamente funcional.
- Probar la restauración de las copias de seguridad periódicamente..
- Administrar los usuarios del sistema, sus permisos y accesos.
- Implementar el diseño de la base de datos.
- Ajustar el rendimiento de la base de datos.
- Descargar e instalar parches de software.
- Implementar en hosts adicionales.
- Ejecutar scripts SQL, analizando su impacto.
- Garantizar que haya suficiente espacio libre en el disco para las operaciones normales.
- Supervisar los trabajos que se ejecutan en la base de datos y garantizar que el rendimiento no se vea afectado por tareas muy costosas enviadas por algunos usuarios.

Configuración y Tuning del DBMS



Configuración de memoria

Se definen los parámetros de uso de memoria, desde los buffers de uso compartido, hasta la memoria asignada a tareas de mantenimiento, ordenamiento, datos temporales, o asignada a cada proceso de usuario, entre otras.



Configuración de almacenamiento

Se definen tanto la ubicación de los archivos de datos, así como los transactions logs, ya sea locales o remotos.

Por otro lado, se definen dentro de esas ubicaciones, los espacios de tablas, y qué bases y/o objetos se almacenan en ellos.



Configuración de red y accesos

Se determinan direcciones IP, protocolos, puertos, rangos de acceso, límites de conexiones, etc..

Desde el lado de la aplicación cliente se recomienda configurar un POOL de conexiones para optimizar los recursos de red, CPU y memoria.

Archivos de configuración

postgresql.conf

Se definen los parámetros de configuración del DBMS.

Se ubica por defecto en la carpeta de datos.

Por defecto se instala uno, con parámetros mínimos, como para un equipo de prueba.

Los cambios en sus datos requieren el reinicio del servicio, o la recarga del mismo.

```
# This is a comment
log_connections = yes
log_destination = 'syslog'
search_path = '$user', public'
shared_buffers = 4GB
max_connections = 200
maintenance_work_mem = 512MB
```

Consejo: Usar como guía las sugerencias de parámetros del sitio

<https://pgtune.leopard.in.ua/>



PGTune

Parameters of your system

DB version [what is this?](#)

OS Type [what is this?](#)

DB Type [what is this?](#)

Total Memory (RAM) [what is this?](#)

Required

Number of CPUs [what is this?](#)

Number of Connections [what is this?](#)

Data Storage [what is this?](#)

You need provide basic information about your hardware configuration, where is working PostgreSQL database. Results will be calculated after clicking "Generate" button

More information about "DB Type" setting:

- Web Application (web)
 - Typically CPU-bound
 - DB much smaller than RAM
 - 90% or more simple queries
- Online Transaction Processing (oltp)
 - Typically CPU- or I/O-bound
 - DB slightly larger than RAM to 1TB
 - 20-40% small data write queries
 - Some long transactions and complex read queries
- Data Warehouse (dw)
 - Typically I/O- or RAM-bound
 - Large bulk loads of data
 - Large complex reporting queries
 - Also called "Decision Support" or "Business Intelligence"
- Desktop application
 - Not a dedicated database
 - A general workstation, perhaps for a developer
- Mixed type of application
 - Mixed DW and OLTP characteristics
 - A wide mixture of queries

Configuración de parámetros en **postgresql.conf**



Es necesario reiniciar el
servicio del DBMS para que
se apliquen los cambios

```
# DB Version: 15
# OS Type: linux
# DB Type: oltp
# Total Memory (RAM): 32 GB
# CPUs num: 4
# Connections num: 199
# Data Storage: ssd
```

```
max_connections = 199
shared_buffers = 8GB
effective_cache_size = 24GB
maintenance_work_mem = 2GB
checkpoint_completion_target = 0.9
wal_buffers = 16MB
default_statistics_target = 100
random_page_cost = 1.1
effective_io_concurrency = 200
work_mem = 21076kB
huge_pages = try
min_wal_size = 2GB
max_wal_size = 8GB
max_worker_processes = 4
max_parallel_workers_per_gather = 2
max_parallel_workers = 4
max_parallel_maintenance_workers = 2
```

Configuración de parámetros a través de sentencias **ALTER SYSTEM**

Escribe los parámetros
dados en el archivo
postgresql.auto.conf que
es leído junto con
postgresql.conf cuando
se inicia el servicio

```
-- DB Version: 15
-- OS Type: linux
-- DB Type: oltp
-- Total Memory (RAM): 32 GB
-- CPUs num: 4
-- Connections num: 199
-- Data Storage: ssd
```

```
ALTER SYSTEM SET max_connections = '199';
ALTER SYSTEM SET shared_buffers = '8GB';
ALTER SYSTEM SET effective_cache_size = '24GB';
ALTER SYSTEM SET maintenance_work_mem = '2GB';
ALTER SYSTEM SET checkpoint_completion_target = '0.9';
ALTER SYSTEM SET wal_buffers = '16MB';
ALTER SYSTEM SET default_statistics_target = '100';
ALTER SYSTEM SET random_page_cost = '1.1';
ALTER SYSTEM SET effective_io_concurrency = '200';
ALTER SYSTEM SET work_mem = '21076kB';
ALTER SYSTEM SET huge_pages = 'try';
ALTER SYSTEM SET min_wal_size = '2GB';
ALTER SYSTEM SET max_wal_size = '8GB';
ALTER SYSTEM SET max_worker_processes = '4';
ALTER SYSTEM SET max_parallel_workers_per_gather = '2';
ALTER SYSTEM SET max_parallel_workers = '4';
ALTER SYSTEM SET max_parallel_maintenance_workers = '2';
```

Archivos de configuración

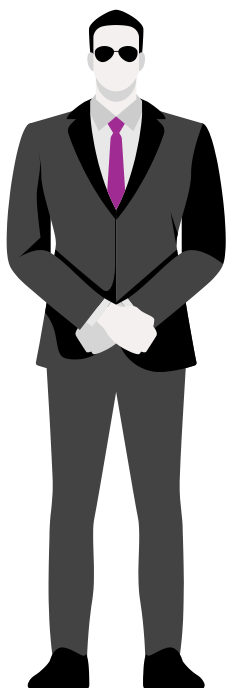
pg_hba.conf

Define qué conexiones se permiten a las bases, desde dónde y para qué usuarios, definiendo el método de autenticación.

También se ubica por defecto en la carpeta de datos.

Por defecto se instala uno, con parámetros iniciales, como para un equipo de prueba.

Los cambios en sus datos requieren el reinicio del servicio, o la recarga del mismo.



```
# TYPE      DATABASE      USER      ADDRESS      METHOD
local      all            all            trust

# The same using local loopback TCP/IP connections.
#
# TYPE      DATABASE      USER      ADDRESS      METHOD
host        all            postgres   127.0.0.1/32   trust

host        postgres     jpereyra    192.168.93.0/24 md5

# TYPE      DATABASE      USER      ADDRESS      METHOD
host        ventas        all        192.168.12.10/32 scram-sha-256
```

Configuración de espacios de almacenamiento

Tablespaces

Permiten a los DBA definir ubicaciones en el sistema de archivos donde se pueden almacenar los archivos que representan objetos de la base de datos. Una vez creado, se puede hacer referencia a un espacio de tabla por su nombre al crear objetos de base de datos.

Al utilizar tablespaces, un DBA puede controlar la distribución del. Si la partición o el volumen en el que se inicializó el clúster se queda sin espacio y no se puede ampliar, se puede crear un espacio de tabla en una partición diferente y utilizarlo hasta que se pueda reconfigurar el sistema.

También el DBA analiza el patrón de uso de los objetos de la base de datos para optimizar el rendimiento. Por ejemplo, un índice que se utiliza mucho se puede colocar en un disco muy rápido y de alta disponibilidad. Al mismo tiempo, una tabla que almacena datos archivados que rara vez se utilizan o que no son críticos para el rendimiento se puede almacenar en un sistema de disco más lento y menos costoso.

```
CREATE TABLESPACE ventas_datos LOCATION '/opt/data';  
  
ALTER TABLE artículos TABLESPACE ventas_datos;  
  
ALTER DATABASE ventas TABLESPACE ventas_datos;
```

Usuarios y Roles de un DBMS

Usuarios? Roles?

Un rol es una entidad que puede poseer objetos de base de datos y tener privilegios de base de datos; un rol puede considerarse un “usuario”, un “grupo” o ambos, según cómo se utilice.

Los usuarios, grupos y roles son lo mismo en PostgreSQL, con la única diferencia que los usuarios tienen permiso para iniciar sesión de forma predeterminada.

Los roles también son esenciales para el concepto de propiedad de objetos dentro de PostgreSQL. Cada base de datos y tabla, por ejemplo, tiene exactamente un rol configurado como propietario. Aparte de los superusuarios, el rol de propietario es el único rol que puede modificar o eliminar el objeto en sí.

Para poder utilizarlos para la conexión inicial al DBMS, los roles primero deben tener configurado el atributo LOGIN. Las reglas de autenticación en sí se definen en el archivo de configuración `pg_hba.conf`.

Tenga en cuenta que los roles se definen en el nivel del DBMS y, por lo tanto, son válidos en todas las bases de datos del mismo.

Creación de usuarios/roles

CREATE ROLE

```
CREATE ROLE name WITH  
    SUPERUSER | NOSUPERUSER  
    | CREATEDB | NOCREATEDB  
    | CREATEROLE | NOCREATEROLE  
    | INHERIT | NOINHERIT  
    | LOGIN | NOLOGIN  
    | CONNECTION LIMIT connlimit  
    | [ ENCRYPTED ] PASSWORD 'password' | PASSWORD NULL  
    | VALID UNTIL 'timestamp' ...
```

```
CREATE USER davide WITH PASSWORD 'jw8s0F4';
```

```
CREATE ROLE miriam WITH LOGIN PASSWORD 'jw8s0F4' VALID UNTIL  
'2005-01-01';
```

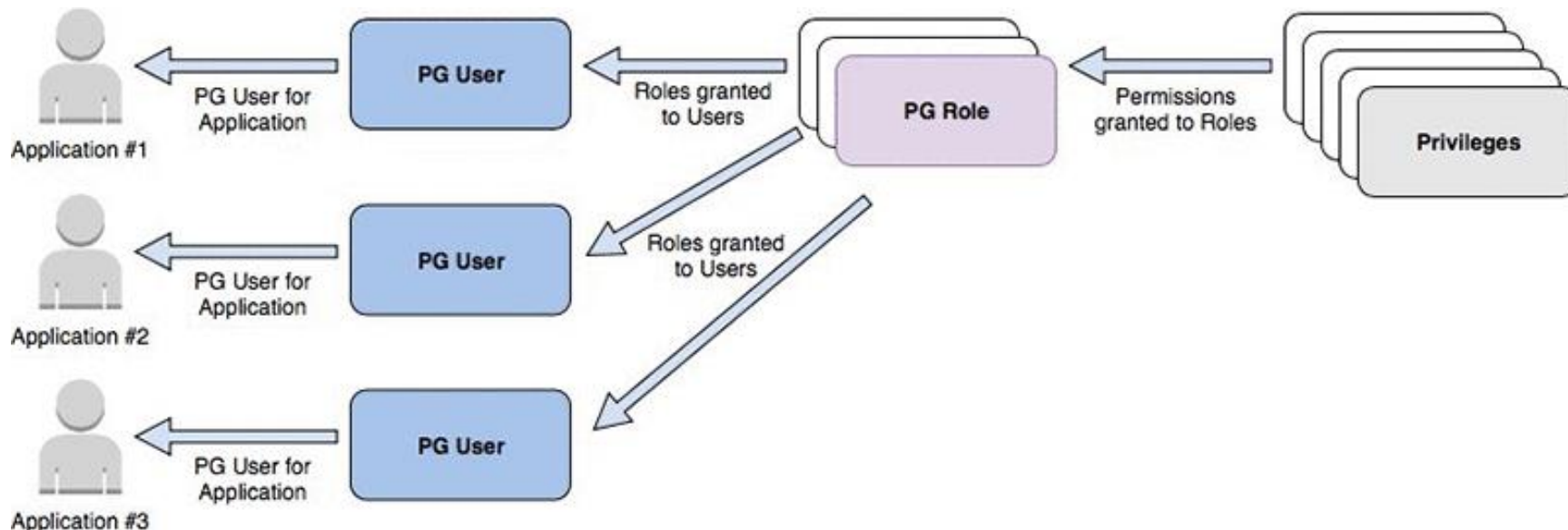
```
CREATE ROLE jpereyra WITH SUPERUSER PASSWORD 'jw8s0F4';
```

Seguridad y permisos

PostgreSQL le permite otorgar permisos directamente a los usuarios de la base de datos.

Sin embargo, como buena práctica, se recomienda que cree varios roles con conjuntos específicos de permisos según los requisitos de acceso y de la aplicación. Luego, asigne el rol adecuado a cada usuario.

Los roles se deben usar para aplicar un modelo de privilegios mínimos para acceder a los objetos de la base de datos. El usuario con permisos de SUPERUSUARIO se debe usar solo para tareas de administración de la base de datos, como crear otros usuarios, roles y bases de datos. La aplicación nunca debe usar esos usuarios.



Seguridad y permisos

Buenas prácticas

El enfoque recomendado para configurar un control de acceso detallado en PostgreSQL es el siguiente:

- Use el usuario maestro para crear roles por aplicación o caso de uso, como **readonly** (sólo lectura) y **readwrite** (lectura y escritura).
- Agregue permisos para permitir que estos roles accedan a varios objetos de la base de datos. Por ejemplo, el rol readonly sólo puede ejecutar consultas SELECT.
- Otorgue a los roles la menor cantidad posible de permisos necesarios para la funcionalidad.
- Cree nuevos usuarios para cada aplicación o funcionalidad distinta, como `app_user` y `reporting_user`.
- Asigna los roles correspondientes a estos usuarios para otorgarles rápidamente los mismos permisos que el rol. Por ejemplo, otorga el rol de lectura y escritura a `app_user` y el rol de solo lectura a `reporting_user`.
- En cualquier momento, puedes quitar el rol del usuario para revocar los permisos.

Asignación de permisos

GRANT (tablas)

```
GRANT { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE |  
        REFERENCES | TRIGGER }  
      | ALL [ PRIVILEGES ] }  
ON { [ TABLE ] table_name [, ...]  
    | ALL TABLES IN SCHEMA schema_name [, ...] }  
TO role_specification  
[ WITH GRANT OPTION ] ...
```

```
GRANT INSERT ON films TO PUBLIC;
```

```
GRANT ALL PRIVILEGES ON kinds TO manuel;
```

```
GRANT SELECT, REFERENCES ON detalle_factura  
TO appventas_sololectura;
```

Asignación de permisos

GRANT (BBDD)

```
GRANT { { CREATE | CONNECT | TEMPORARY | TEMP }  
      | ALL [ PRIVILEGES ] }  
ON DATABASE database_name [, ...]  
TO role_specification  
[ WITH GRANT OPTION ] ...
```

```
GRANT CONNECT ON DATABASE ventas TO PUBLIC;
```

```
GRANT ALL ON DATABASE ventas TO jpereryra;
```

```
GRANT CONNECT, CREATE ON DATABASE ventas  
TO appventas_desarrollo;
```

Quita o revocación de permisos

REVOKE (tablas)

```
REVOKE [ GRANT OPTION FOR ]
      { { SELECT | INSERT | UPDATE | DELETE | TRUNCATE |
        REFERENCES | TRIGGER }
        | ALL [ PRIVILEGES ] }
ON { [ TABLE ] table_name [, ...]
     | ALL TABLES IN SCHEMA schema_name [, ...] }
FROM role_specification [, ...]
```

```
REVOKE INSERT ON films FROM PUBLIC;
```

```
REVOKE ALL PRIVILEGES ON kinds FROM manuel;
```

```
REVOKE SELECT, REFERENCES ON detalle_factura
TO appventas_sololectura;
```

Quita o revocación de permisos

REVOKE (BBDD)

```
REVOKE [ GRANT OPTION FOR ]  
      { { CREATE | CONNECT | TEMPORARY | TEMP } [, ...]  
        | ALL [ PRIVILEGES ] }  
ON DATABASE database_name [, ...]  
FROM role_specification [, ...]
```

```
REVOKE CONNECT ON DATABASE ventas FROM PUBLIC;
```

```
REVOKE ALL ON DATABASE ventas FROM jpereryra;
```

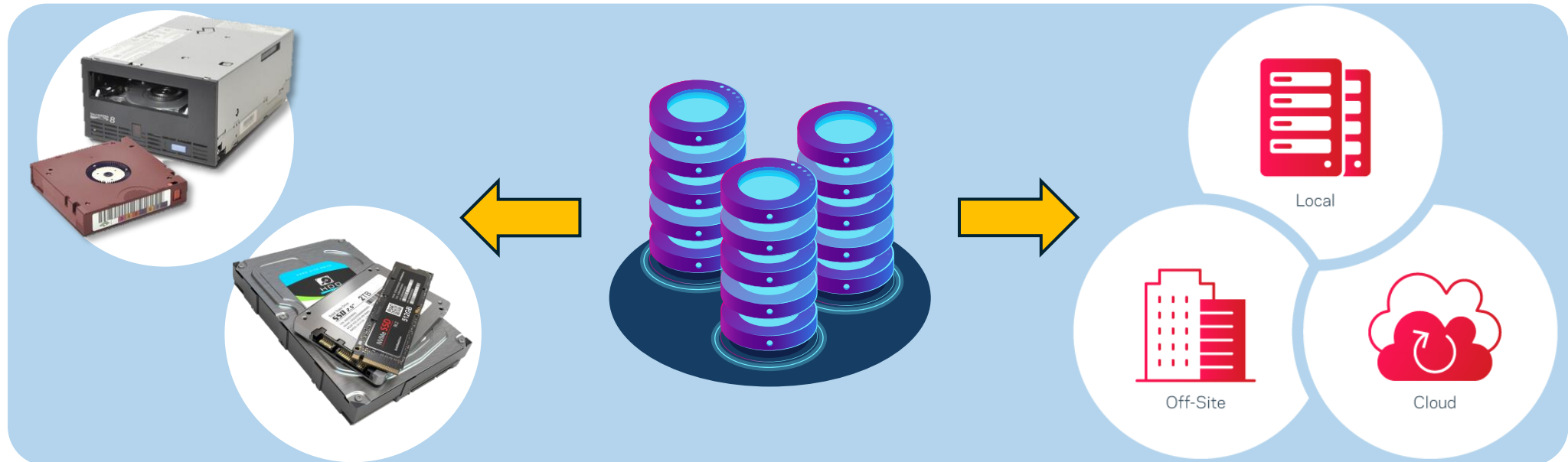
```
REVOKE CONNECT, CREATE ON DATABASE ventas  
FROM appventas_desarrollo;
```

Copias de Seguridad de BBDD (Backups)

Una copia de seguridad de una base de datos es el proceso de crear una copia de los datos de una base de datos para garantizar su disponibilidad en caso de pérdida, corrupción o fallo del sistema.

Esta copia de seguridad se puede utilizar para restaurar la base de datos a su estado anterior, lo que protege la información crítica de problemas inesperados.

Las copias de seguridad se pueden realizar con regularidad y almacenar en las instalaciones o en entornos de nube, o sitios remotos, lo que proporciona una protección fiable para los datos empresariales valiosos.



Copias de Seguridad de BBDD (Backups)

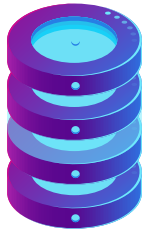
¿Por qué es importante la copia de seguridad de la base de datos?

Después de las personas, **los datos son el activo más valioso** de una empresa. Los utiliza para gestionar su negocio. Sus operaciones diarias dependen de datos financieros, información de clientes, registros de empleados, detalles de productos, fórmulas y procesos patentados y software de aplicación. El tiempo de inactividad prolongado debido a la interrupción de los datos perjudica tanto su reputación como sus resultados.

La corrupción y la pérdida de datos pueden ocurrir a través de un desastre natural como incendios e inundaciones, fallas de energía o hardware, error humano y ataques cibernéticos. **Cuando ocurre un desastre de datos, necesita la capacidad de recuperar y restaurar rápidamente esos archivos y su negocio al estado operativo normal.**

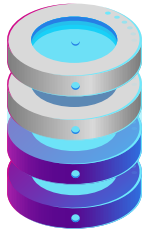
Las estrategias de planificación de la copia de seguridad y recuperación de bases de datos no podrían ser más cruciales para la continuidad del negocio (**Business Continuity**). Junto con la recuperación ante desastres (**Disaster Recovery**), el acrónimo BC/DR abarca dos caras de la misma moneda. Mientras que la recuperación ante desastres restaura la tecnología necesaria, la continuidad del negocio trata de mantener el negocio en su conjunto en funcionamiento.

Tipos de Copias de Seguridad



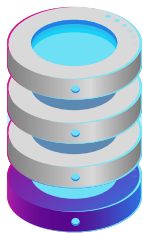
FULL

Se copia y almacena toda la base de datos cada vez que se ejecuta una copia de seguridad de la base de datos.



DIFERENCIAL

Sólo se copian y guardan los cambios realizados en la base de datos desde la última copia de seguridad completa.



INCREMENTAL

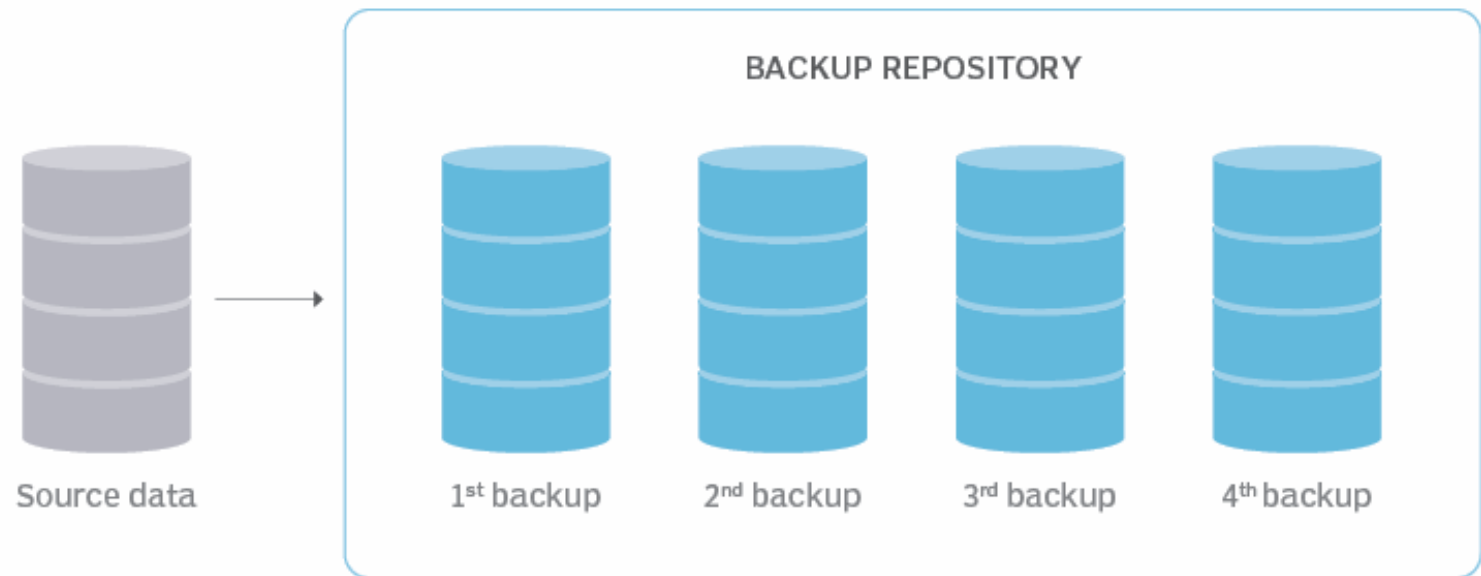
Sólo se copian y guardan los cambios realizados en la base de datos desde la última copia de seguridad.

Backup FULL (Completo)

Se realiza una copia de seguridad de todos los datos en una única versión y se traslada a un dispositivo de almacenamiento. Es la protección perfecta contra la pérdida de datos si se tiene en cuenta la velocidad y la simplicidad de la recuperación.

Sin embargo, el tiempo y el gasto necesarios para copiar todos los datos (todo el tiempo) pueden convertirla en una opción poco recomendable para muchos profesionales de TI.

Los datos son copiados en su totalidad cada vez.

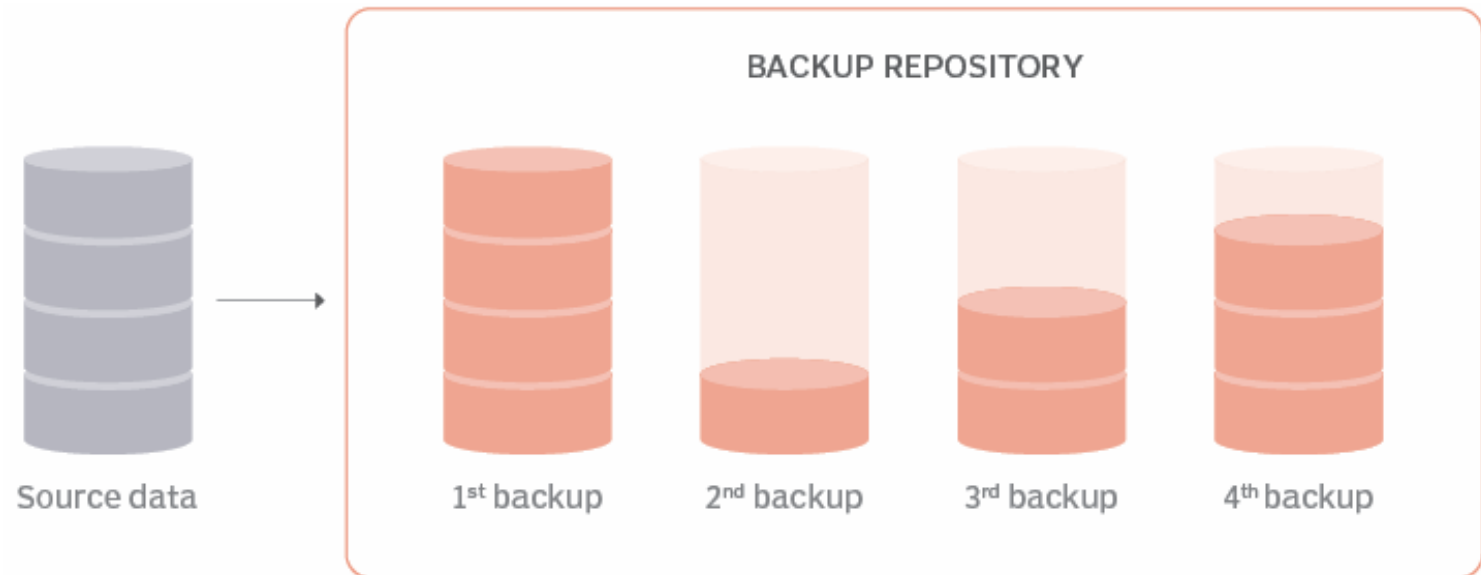


Backup DIFERENCIAL

Se trata de realizar copias de seguridad de datos que se insertaron o modificaron desde la última copia de seguridad completa (en comparación con solo los cambios desde la última copia de seguridad incremental). Sólo se realiza una copia de seguridad de un pequeño volumen de datos entre el intervalo de tiempo de la última copia de seguridad y la actual, lo que consume menos espacio de almacenamiento y requiere menos tiempo e inversión.

En caso de que necesites recuperar toda la base, se deberán restaurar **SÓLO EL ÚLTIMO** de los archivos de backup desde la última copia de seguridad completa, incluida ésta y sólo la última copia de seguridad diferencial.

En 1ra. instancia, los datos se copian en su totalidad y, luego, sólo se realiza una copia de seguridad de los datos nuevos o actualizados desde esa copia de seguridad completa.

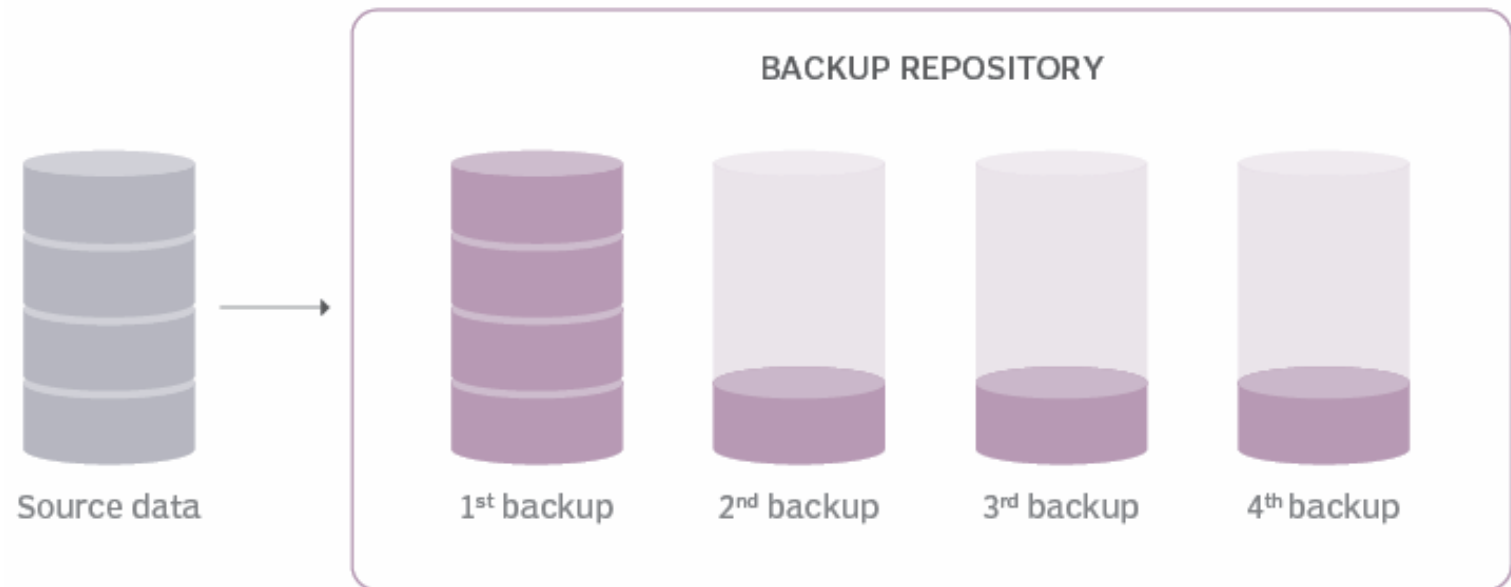


Backup INCREMENTAL

La copia de seguridad incremental implica realizar una copia de seguridad de todos los datos que hayan cambiado desde la última actividad de copia de seguridad. Sólo se realiza una copia de seguridad de los cambios recientes (incrementos), lo que consume menos espacio de almacenamiento y da como resultado una copia de seguridad más rápida. Sin embargo, el tiempo de recuperación es mayor, ya que será necesario acceder a más archivos de copia de seguridad.

En caso de que necesites recuperar toda la base, se deberán restaurar TODOS los archivos de backup desde la última copia de seguridad completa, incluida la última copia de seguridad completa y las copias de seguridad incrementales posteriores.

En 1ra. instancia, los datos se copian en su totalidad y, luego, sólo se realiza una copia de seguridad de los datos nuevos o actualizados.



Bibliografía y fuentes consultadas

- ***PostgreSQL 15.8 Documentation***
(1996–2024), The PostgreSQL Global Development Group
- ***Oracle Database Database 18c Administrator's Guide***
(2024), Oracle University
- ***Documentación de Cloud SQL para PostgreSQL***
(2024), Google Cloud
- ***Temas Destacados: ¿Qué es la arquitectura de tres niveles?***
IBM® Cloud
- ***Search Data Backup: Full vs. incremental vs. differential: Comparing backup types***
Tech Target Network