Administración de Bases de Datos

PRÁCTICA 3 – CATÁLOGOS EN IMPLEMENTACIONES SGBDR

18 de mayo de 2025

Prof. David Criado Ramón

Autor:

Ana Aragón Jerónimo

UNIVERSIDAD DE GRANADA

E.T.S. de Ingenierías Informática y de Telecomunicación

Índice

Introducción

A continuación se va a realizar el estudio de un Sistema Gestor de Bases de Datos Relacional distinto al visto anteriormente a lo largo de esta asignatura (Oracle).
El catálogo de datos es un inventario organizado de los activos de datos de la organización correspondiente. Se usa dentro de dichas organizaciones para la administración de sus datos

Se ha escogido como sistema gestor de bases de datos relacional (SGBDR) PostgreSQL, tratándose de un sistema relacional de bases de datos de código abierto, orientado a objetos y caracterizado por su estabilidad. Este SGBDR proporciona soporte a diferentes funciones de SQL, y tiene además una capacidad de ampliación del lenguaje SQL mediante funciones que escalan y reservan de forma meticulosa las cargas de trabajo de datos.

La elección de PosgreSQL como SGBDR se ha realizado debido a su gran relevancia en el ámbito empresarial, siendo ampliamente usado en el sector; así como su arquitectura clara y documentada, lo cual permite analizar en profundidad la estructura del catálogo.

Otros sistemas como MySQL, MariaDB y Microsoft SQL Server se podrían haber considerado al tratarse de SGBDR también, pero PostgreSQL resulta especialmente adecuado para la comparación con Oracle debido a su compatibilidad con tipos avanzados, sintaxis más cercana al estándar SQL y su enfoque en la integridad y extensibilidad del catálogo.

Identificación de objetos en PostgreSQL

PostgreSQL utiliza un sistema dual de identificación para los diferentes objetos de la base de datos: identificadores lógicos e identificadores físicos.

-Identificadores lógicos

Cada objeto creado dentro de PostgreSQL se identifica lógicamente por su nombre dentro de un namespace. La combinación de nombre y esquema garantiza la unicidad del objeto dentro de la base de datos.

Un ejemplo puede ser una tabla, la cual se va a llamar **tripulantes**, en un esquema nombrado **navegacion**: esta tabla se referenciaría como **navegacion.tripulantes**.
Este identificador lógico permite distinguirla de otra posible tabla tripulantes en un esquema distinto (por ejemplo, exploracion.tripulantes).

-Identificadores físicos: OID

Internamente, muchos objetos en PosgreSQL son asignados con un Object Identifier (**OID**), que se trata de un número entero de 4 bytes que actúa como clave primaria en varias tablas del catálogo.
No todos los objetos reciben un OID por defecto, especialmente desde la versión 12, donde se eliminó la asignación automática de OID a las tablas creadas por el usuario dado.

Los OID son esenciales para que el motor de PostgreSQL realice un acceso eficiente a los metadatos. Un ejemplo puede ser el siguiente:

- Cada fila en **pg_class** (donde se registran tablas y vistas) tiene un OID, el cual puede ser utilizado como referencia para otras tablas del catálogo.
- El **OID** permite relaciones internas entre objetos, como por ejemplo, en **pg_attribute**, que referencia el OID de una tabla en **pg_class** mediante attrelid.

-Comparación con Oracle

En Oracle Database, los objetos se identifican de forma similar a nivel lógico mediante un esquema y un nombre de objeto. No obstante, Oracle no hace uso de OIDs como PostgreSQL.

En su lugar, cada objeto tiene un identificador lógico único dentro del esquema, y la base de datos administra internamente una clave única (**OBJECT_ID**) en la tabla **DBA_OBJECT**, que actúa como identificador físico.

Además, Oracle mantiene una distinción clara entre objetos (tablas, vistas…) y sus propietarios (usuarios), mientras que PostgreSQL los trata como roles.

Estructura del catálogo en PostgreSQL

El catálogo de PosgreSQL está compuesto por un conjunto de tablas del sistema ubicadas en el esquema **pg_catalog**. Estas tablas almacenan toda la metainformación sobre los objetos existentes. A diferencia de otros sistemas, como MySQL, el catálogo en PostgreSQL es accesible mediante consultas SQL estándar y no requiere herramientas externas. A continuación, se describen las principales estructuras utilizadas para representar tablas, vistas y usuarios.

-Tablas y vistas

- **pg_class**
  Esta tabla contiene una fila por cada objeto relacional: tablas, vistas, índices, secuencias, etc. Cada fila incluye datos como:
  - ***relname***: nombre del objeto.
  - ***relnamespace***: OID del esquema al que pertenece.
  - ***relkind***: tipo de objeto (r para tabla, v para vista, i para índice, etc.).
  - ***relowner***: OID del rol propietario.

- **pg_attribute**
  Contiene una fila por cada columna de cada objeto relacional listado en **pg_class**. Sus columnas más relevantes incluyen:
  - ***attrelid***: OID de la tabla o vista a la que pertenece.
  - ***attname***: nombre de la columna.
  - ***atttypid***: tipo de dato (referencia a **pg_type**).

- **pg_namespace**
  Representa los esquemas. Cada registro contiene:

- ➢ ***nspname***: nombre del esquema.
- ➢ ***nspower***: OID del rol propietario.

❖ **pg_type**
Define los tipos de datos, tanto del sistema como definidos por el usuario. Cada tipo posee un OID único que se usa como referencia en **pg_attribute**.

❖ **Vistas normalizadas**
PostgreSQL también expone información del catálogo en el esquema information_schema. Este conjunto de vistas permite consultas más portables y estandarizadas:

- ➢ ***information_schema.tables***: lista las tablas y vistas visibles al usuario.
- ➢ ***information_schema.columns***: describe las columnas de cada tabla.
- ➢ ***information_schema.views***: lista las vistas y su definición (cuando es accesible).

-Usuarios (roles)

En PostgreSQL no existe una distinción técnica entre usuarios y grupos. Ambos son gestionados mediante el concepto de roles, que puede tener capacidades como iniciar sesión (LOGIN), crear bases de datos, o asumir otros roles.

❖ **pg_roles**
Contiene todos los roles definidos en el sistema. Algunas columnas destacadas son:

- ➢ ***rolname***: nombre del rol.
- ➢ ***rolcanlogin***: booleano que indica si el rol puede iniciar sesión (usuario).
- ➢ ***rolsuper***: si es un superusuario.
- ➢ ***rolvaliduntil***: fecha de expiración de la contraseña (si aplica).

❖ **pg_authid** (accesible solo para superusuarios)
Guarda información sensible de autenticación como contraseñas cifradas.

❖ **Vistas auxiliares**

- ➢ ***pg_user***: vista pública sobre **pg_roles**, limitada a roles con LOGIN.
- ➢ ***pg_shadow***: similar a **pg_user**, pero incluye información sensible y es visible solo administradores.

-Comparación con Oracle

Oracle utiliza un enfoque similar en cuanto al almacenamiento de metadatos, pero sus estructuras están organizadas en vistas del diccionario de datos, no como tablas directamente accesibles.

❖ **Tablas y vistas:**

- ➢ ***DBA_TABLES***, ***USER_TABLES***, ***ALL_TABLES***: información sobre tablas según el nivel de acceso.
- ➢ ***DBA_TAB_COLUMNS***, ***USER_TAB_COLUMNS***: columnas de cada tabla.
- ➢ ***DBA_VIEWS***, ***USER_VIEWS***: definiciones de vistas.

❖ **Usuarios:**

- ➢ ***DBA_USERS***: contiene información sobre todos los usuarios.

- ***DBA_ROLES***, **ROLE_SYS_PRIVS**: gestión de privilegios y roles.
- ***DBA_PROFILES***: configuración de seguridad por usuario.

A diferencia de PostgreSQL, Oracle segmenta el acceso a las vistas del catálogo según el privilegio del usuario (DBA, propietario o cualquier usuario).

-Listado de las principales tablas del catálogo de PosgreSQL

Se presenta una tabla con algunas de las tablas que se pueden encontrar en este SGDBR.

pg_database	Almacena información sobre las distintas bases de datos.
pg_namespace	Contiene los esquemas definidos.
pg_class	Representa tablas, vistas, índices y otras clases de objetos relacionales.
pg_attribute	Almacena columnas de las tablas y las vistas.
pg_type	Define los tipos de datos disponibles.
pg_roles	Lista todos los roles y usuarios.
pg_index	Almacena definiciones de índices secundarios.
pg_proc	Contiene información sobre funciones (definidas por el sistema/usuario).
pg_operator	Contiene información sobre los operadores definidos en el sistema.
pg_aggregate	Almacena datos sobre las funciones agregadas y sus propiedades.

-Vistas útiles del catálogo de PosgreSQL

De igual manera, se pueden realizar vistas con las siguientes opciones, ya mencionadas.

information_schema.tables	Listado de tablas visibles para el usuario actual
information_schema.columns	Columnas de las tablas.
information_schema.views	Vistas definidas en la base de datos.
pg_user	Información de los usuarios.
pg_shadow	Información de usuarios con contraseñas (acceso restringida).

-Comandos útiles en PostgreSQL para consultar el catálogo

También cuenta con una serie de comandos para consultar vistas específicas del catálogo.

\db	Lista todos los tablespace disponibles.
\dn	Lista todos los esquemas (schemas) existentes en la base de datos actual.
\dt	Devuelve las tablas definidas en el esquema por defecto del usuario actual.
\d nombre_tabla	Muestra la definición de una tabla específica (incluyendo columnas, tipos, claves, índices y restricciones).
\dp	Detalla los privilegios de acceso asignados a los diferentes objetos de la base de datos.
\da	Enumera todas las funciones de agregación disponibles.
\df	Muestra el conjunto de funciones almacenadas que pueden ser invocadas desde SQL.
\do	Presenta los operadores definidos en el sistema, junto con sus operandos y tipos de retorno.
\dT	Muestra todos los tipos de datos definidos en la base de datos.
\du	Lista los roles y usuarios definidos, junto con sus atributos.
\l	Muestra un listado completo de las bases de datos disponibles.

Referencias

Kinsta. (n.d.). *¿Qué es PostgreSQL?*. Kinsta. Recuperado el 18 de mayo de 2025, de https://kinsta.com/es/base-de-conocimiento/que-es-postgresql/

The PostgreSQL Global Development Group. (2024). *System Catalogs*. PostgreSQL Documentation. Recuperado el 18 de mayo de 2025, de https://www.postgresql.org/docs/current/catalogs.html

The PostgreSQL Global Development Group. (2024). *The Information Schema*. PostgreSQL Documentation. Recuperado el 18 de mayo de 2025, de https://www.postgresql.org/docs/current/information-schema.html

Oracle Corporation. (2024). *Oracle Database Documentation*. Oracle Help Center. Recuperado el 18 de mayo de 2025, de https://docs.oracle.com/en/database/oracle/oracle-database/

The PostgreSQL Global Development Group. (2024). *psql – PostgreSQL Interactive Terminal*. PostgreSQL Documentation. Recuperado el 18 de mayo de 2025, de https://www.postgresql.org/docs/current/app-psql.html

Además se han usado como referencia los apuntes proporcionados a lo largo de la asignatura a la hora de hacer las comparaciones con Oracle.