info.GetPrivileges - Technical Documentation

# Summary

The `info.GetPrivileges` function is a PostgreSQL stored procedure designed to retrieve detailed privilege mappings for any given role (or all roles) within the current database context. It compiles privileges across schemas, database-level access rights, and optionally default ACLs if supported by the current PostgreSQL version.

# Technical Breakdown

1. Version Awareness:  
- Checks `server\_version\_num` to conditionally use `pg\_default\_acl` if version >= 14.  
  
2. Role Expansion:  
- Uses `pg\_auth\_members` to recursively determine inherited roles.  
  
3. Schema Privileges:  
- Pulls `CREATE` and `USAGE` rights using `has\_schema\_privilege`.  
  
4. Database Privileges:  
- Checks for `CREATE`, `TEMP`, `CONNECT` via `has\_database\_privilege`.  
  
5. Default ACL Expansion:  
- Extracts object-level defaults per schema (if version supports it), then decodes privilege bitmasks.  
  
6. Output Formatting:  
- Result set returns 5 columns: `object\_scope`, `grantee`, `privilege`, `inherited\_role`, `source`.

# Common Use Cases

- Verify if roles have appropriate access before revoking.  
- Confirm inherited permissions for team or service accounts.  
- Audit privileges post-deployment in multi-schema environments.  
- Ensure secure privilege escalation boundaries.

# References

1. PostgreSQL Documentation - Privileges and ACLs

https://www.postgresql.org/docs/current/ddl-priv.html

2. PostgreSQL Function Development

https://www.postgresql.org/docs/current/sql-createfunction.html

3. PostgreSQL Default ACLs

https://www.postgresql.org/docs/current/sql-alterdefaultprivileges.html

## TL;DR

Use `SELECT \* FROM info.GetPrivileges('<role\_name>');` to view all schema, database, and inherited privileges for a role. Pass an empty string to see results for all roles.