# deploy.SetRevokeRolePermissions – Technical Documentation

Version: 2.0  
Date: 2025-08-01  
Page: 1 of 2

## Summary

Cleans up PostgreSQL roles by removing object-level privileges, revoking memberships, and optionally dropping roles. Function supports idempotent execution with full dry-run mode and logging.

## Function Logic

1. For each role in `p\_role\_names`:  
 a. If `p\_users\_to\_remove` is set:  
 - REVOKE role FROM each user.  
 b. If `p\_schema\_targets` provided:  
 - REVOKE table-level privileges (SELECT, INSERT, etc.)  
 - REVOKE function-level privileges (EXECUTE)  
2. If `p\_drop\_roles = TRUE`, issue DROP ROLE statements  
3. Log each action to `info.account\_log\_history` with status EXECUTED or DRY\_RUN

## Execution Flags

- `p\_execute\_flag = TRUE`: Executes all SQL commands  
- `p\_execute\_flag = FALSE`: Logs but does not change any access  
- `p\_drop\_roles = TRUE`: Drops the listed roles after revocation

## Security & Access

- Caller must be able to revoke roles and access in target schemas  
- DROP ROLE requires superuser or CREATEROLE privileges  
- Roles not found or empty schemas will result in skipped commands

## Limitations

- No dependency checking before DROP ROLE  
- Assumes schema exists and is accessible  
- Only operates on TABLES and FUNCTIONS in provided schemas

## References

- https://www.postgresql.org/docs/current/sql-revoke.html  
- https://www.postgresql.org/docs/current/sql-drop-role.html  
- https://www.postgresql.org/docs/current/sql-grant.html

## TL;DR

Strips role privileges and memberships, then optionally drops roles. Fully dry-run capable with structured schema targeting.