# deploy.CreateSchemaWithPermissions – Technical Documentation

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

## Summary

This function creates a schema and applies permission structures automatically. It checks for existence, logs each step, and supports dry-run.

## Flow Logic

1. Check if schema exists in `information\_schema.schemata`.  
2. If it does: log and return without error.  
3. If not:  
 - Build CREATE SCHEMA with AUTHORIZATION clause.  
 - Only execute if `p\_execute\_flag` is TRUE.  
4. For each `p\_grant\_roles` value:  
 - GRANT USAGE ON SCHEMA to that role  
 - ALTER DEFAULT PRIVILEGES to grant:  
 - SELECT, INSERT, UPDATE, DELETE on future TABLES  
 - EXECUTE on future FUNCTIONS  
5. Log every action to `info.object\_log\_history` with action\_type CREATE\_SCHEMA

## Security Considerations

- Schema owner must have permission to grant privileges  
- Caller must be superuser or have create-schema rights in the database  
- All commands are optionally dry-run and logged with context

## Limitations & Edge Cases

- Will not overwrite existing schema  
- Does not validate role existence for `p\_grant\_roles`  
- No DDL audit trigger integration unless built externally

## References

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

## TL;DR

Safe schema creation with auto-permissions and dry-run support. Respects existing objects, logs everything, grants future access.