# deploy.CreateSchemaWithPermissions – How-To Guide

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

## Summary

Creates a new schema, assigns ownership, and optionally grants USAGE and default privileges to listed roles. Includes dry-run support and detailed logging.

## Usage Instructions

Run using:  
  
 SELECT deploy.CreateSchemaWithPermissions(  
 p\_schema\_name := 'your\_schema',  
 p\_owner := 'your\_owner',  
 p\_grant\_roles := ARRAY['role1','role2'],  
 p\_execute\_flag := TRUE  
 );

## Examples

1. Dry-run new schema:  
 SELECT deploy.CreateSchemaWithPermissions('audit', 'jrussell', ARRAY['readonly'], FALSE);  
  
2. Create with grants and default privileges:  
 SELECT deploy.CreateSchemaWithPermissions('audit', 'jrussell', ARRAY['db\_datareader'], TRUE);  
  
3. Skip if exists (no error):  
 SELECT deploy.CreateSchemaWithPermissions('public', 'postgres', ARRAY['qa\_team'], TRUE);

## Parameters

- `p\_schema\_name`: TEXT – Schema name to create  
- `p\_owner`: TEXT – Role that will own the schema  
- `p\_grant\_roles`: TEXT[] – Roles to grant USAGE and future privileges  
- `p\_execute\_flag`: BOOLEAN – TRUE to run, FALSE for dry-run logging

## TL;DR

Creates schema if missing, assigns owner, grants USAGE to roles, sets default table/function grants. Safe dry-run support.