

# Consultas de la Base de Datos

Usuario	Consulta	Fecha
postgres	SELECT query, calls, total_exec_time, rows FROM pg_stat_statements WHERE dbid = (SELECT oid FROM pg_database WHERE datname = \$1)	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT rl.*, r.rolname AS user_name, db.datname as db_name FROM pg_catalog.pg_db_role_setting AS rl LEFT JOIN pg_catalog.pg_roles AS r ON rl.setrole = r.oid LEFT JOIN pg_catalog.pg_database AS db ON rl.setdatabase = db.oid WHERE setdatabase = \$1	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	select * from auditoria	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT pid, username, query, query_start FROM pg_stat_activity WHERE datname = \$2 AND backend_start::date = \$1	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	INSERT INTO menu (nombre, icono, activo, fecharegistro) VALUES (\$1, \$2, \$3, \$4)	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	CREATE EXTENSION pg_stat_statements	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SET client_encoding='utf-8'	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT n.nspname, r.relname FROM pg_catalog.pg_class r LEFT JOIN pg_catalog.pg_namespace n ON (r.relnamespace = n.oid) WHERE r.oid = \$1	Mon Jul 29 2024 19:51:30

Usuario	Consulta	Fecha
		GMT-0500 (hora de Ecuador)
postgres	SELECT set_config(\$1,\$2,\$3) FROM pg_settings WHERE name = \$4	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	/*pga4dash*/ SELECT \$1 AS chart_name, pg_catalog.row_to_json(t) AS chart_data FROM (SELECT (SELECT count(*) FROM pg_catalog.pg_stat_activity WHERE datname = (SELECT datname FROM pg_catalog.pg_database WHERE oid = \$2)) AS "Total", (SELECT count(*) FROM pg_catalog.pg_stat_activity WHERE state = \$3 AND datname = (SELECT datname FROM pg_catalog.pg_database WHERE oid = \$4)) AS "Active", (SELECT count(*) FROM pg_catalog.pg_stat_activity WHERE state = \$5 AND datname = (SELECT datname FROM pg_catalog.pg_database WHERE oid = \$6)) AS "Idle" ) t UNION ALL SELECT \$7 AS chart_name, pg_catalog.row_to_json(t) AS chart_data FROM (SELECT (SELECT sum(xact_commit) + sum(xact_rollback) FROM pg_catalog.pg_stat_database WHERE datname = (SELECT datname FROM pg_catalog.pg_database WHERE oid = \$8)) AS "Transactions", (SELECT sum(xact_commit) FROM pg_catalog.pg_stat_database WHERE datname = (SELECT datname FROM pg_catalog.pg_database WHERE oid = \$9)) AS "Commits", (SELECT sum(xact_rollback) FROM pg_catalog.pg_stat_database WHERE datname = (SELECT datname FROM pg_catalog.pg_database WHERE oid = \$10)) AS "Rollbacks" ) t UNION ALL SELECT \$11 AS chart_name, pg_catalog.row_to_json(t) AS chart_data FROM (SELECT (SELECT sum(tup_inserted) FROM pg_catalog.pg_stat_database WHERE datname = (SELECT datname FROM pg_catalog.pg_database WHERE oid = \$12)) AS "Inserts", (SELECT sum(tup_updated) FROM pg_catalog.pg_stat_database WHERE datname = (SELECT datname FROM pg_catalog.pg_database WHERE oid = \$13)) AS "Updates", (SELECT sum(tup_deleted) FROM pg_catalog.pg_stat_database WHERE datname = (SELECT datname FROM pg_catalog.pg_database WHERE oid = \$14)) AS "Deletes" ) t UNION ALL SELECT \$15 AS chart_name, pg_catalog.row_to_json(t) AS chart_data FROM (SELECT (SELECT sum(tup_fetched) FROM pg_catalog.pg_stat_database WHERE datname = (SELECT datname FROM pg_catalog.pg_database WHERE oid = \$16)) AS "Fetched", (SELECT sum(tup_returned) FROM pg_catalog.pg_stat_database WHERE datname = (SELECT datname FROM pg_catalog.pg_database WHERE oid = \$17)) AS "Returned" ) t UNION ALL SELECT \$18 AS chart_name, pg_catalog.row_to_json(t) AS chart_data FROM (SELECT (SELECT sum(blks_read) FROM pg_catalog.pg_stat_database WHERE datname = (SELECT datname FROM pg_catalog.pg_database WHERE oid = \$19)) AS "Reads", (SELECT sum(blks_hit) FROM pg_catalog.pg_stat_database WHERE datname = (SELECT datname FROM pg_catalog.pg_database WHERE oid = \$20)) AS "Hits" ) t	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT oid, pg_catalog.format_type(oid, \$2) AS typename FROM pg_catalog.pg_type WHERE oid = ANY(\$1) ORDER BY oid	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT roles.oid as id, roles.rolname as name, roles.rolsuper as is_superuser, CASE WHEN roles.rolsuper THEN \$1 ELSE roles.rolcreatorrole END as can_create_role, CASE WHEN roles.rolsuper THEN \$2 ELSE roles.rolcreatedb END as can_create_db, CASE WHEN \$3=ANY(ARRAY(WITH RECURSIVE cte AS ( SELECT pg_roles.oid,pg_roles.rolname FROM pg_roles WHERE pg_roles.oid = roles.oid UNION ALL SELECT m.roleid,pgr.rolname FROM cte cte_1 JOIN pg_auth_members m ON m.member = cte_1.oid JOIN pg_roles pgr ON pgr.oid = m.roleid) SELECT rolname FROM cte)) THEN \$4 ELSE \$5 END as can_signal_backend FROM pg_catalog.pg_roles as roles WHERE rolname = current_user	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT nsp.nspname as schema_name, (nsp.nspname = \$1 AND EXISTS (SELECT \$2 FROM pg_catalog.pg_class WHERE relname = \$3 AND relnamespace = nsp.oid LIMIT \$4)) OR	Mon Jul 29 2024

Usuario	Consulta	Fecha
	(nsp.nspname = \$5 AND EXISTS (SELECT \$6 FROM pg_catalog.pg_class WHERE relname = \$7 AND relnamespace = nsp.oid LIMIT \$8)) OR (nsp.nspname = \$9 AND EXISTS (SELECT \$10 FROM pg_catalog.pg_class WHERE relname = \$11 AND relnamespace = nsp.oid LIMIT \$12)) AS is_catalog, CASE WHEN nsp.nspname = ANY(\$13) THEN \$14 ELSE \$15 END AS db_support FROM pg_catalog.pg_namespace nsp WHERE nsp.oid = \$16::OID	19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT gss_authenticated, encrypted FROM pg_catalog.pg_stat_gssapi WHERE pid = pg_backend_pid()	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SET client_min_messages=notice	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT DISTINCT att.attname as name, att.attnum as OID, pg_catalog.format_type(ty.oid,\$1) AS datatype, att.attnotnull as not_null, att.atthasdef as has_default_val, des.description, seq.sectypid FROM pg_catalog.pg_attribute att JOIN pg_catalog.pg_type ty ON ty.oid=atttypid JOIN pg_catalog.pg_namespace tn ON tn.oid=ty.typnamespace JOIN pg_catalog.pg_class cl ON cl.oid=att.attrelid JOIN pg_catalog.pg_namespace na ON na.oid=cl.relnamespace LEFT OUTER JOIN pg_catalog.pg_type et ON et.oid=ty.typelem LEFT OUTER JOIN pg_catalog.pg_attrdef def ON adrelid=att.attrelid AND adnum=att.attnum LEFT OUTER JOIN (pg_catalog.pg_depend JOIN pg_catalog.pg_class cs ON classid=\$2::regclass AND objid=cs.oid AND cs.relkind=\$3) ON refobjid=att.attrelid AND refobjsubid=att.attnum LEFT OUTER JOIN pg_catalog.pg_namespace ns ON ns.oid=cs.relnamespace LEFT OUTER JOIN pg_catalog.pg_index pi ON pi.indrelid=att.attrelid AND indisprimary LEFT OUTER JOIN pg_catalog.pg_description des ON (des.objoid=att.attrelid AND des.objsubid=att.attnum AND des.classoid=\$4::regclass) LEFT OUTER JOIN pg_catalog.pg_sequence seq ON cs.oid=seq.seqrelid WHERE att.attrelid = \$5::oid AND att.attnum > \$6 AND att.attisdropped IS FALSE ORDER BY att.attnum	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT * FROM pg_stat_statements WHERE dbid = (SELECT oid FROM pg_database WHERE datname = \$1)	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT COUNT(*) FROM pg_extension WHERE extname IN ( \$1, \$2)	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT ca.oid, pg_catalog.concat(pg_catalog.format_type(st.oid,\$1),\$2,pg_catalog.format_type(tt.oid,tt.tytypmod)) as name, des.description FROM pg_catalog.pg_cast ca JOIN pg_catalog.pg_type st ON st.oid=castsource JOIN pg_catalog.pg_namespace ns ON ns.oid=st.typnamespace JOIN pg_catalog.pg_type tt ON tt.oid=casttarget JOIN pg_catalog.pg_namespace nt ON nt.oid=tt.typnamespace LEFT JOIN pg_catalog.pg_proc pr ON pr.oid=castfunc LEFT JOIN pg_catalog.pg_namespace np ON np.oid=pr.pronamespace LEFT OUTER JOIN pg_catalog.pg_description des ON (des.objoid=ca.oid AND des.objsubid=\$3 AND des.classoid=\$4::regclass) WHERE ca.oid > \$5::OID ORDER BY st.typname, tt.typname	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)

Usuario	Consulta	Fecha
postgres	SELECT at.attname, ty.typname, at.attnum FROM pg_catalog.pg_attribute at LEFT JOIN pg_catalog.pg_type ty ON (ty.oid = at.atttypid) JOIN pg_catalog.pg_class as cl ON cl.oid=at.attrelid JOIN pg_catalog.pg_namespace as nsp ON nsp.oid=cl.relnamespace WHERE attrelid=\$1::oid AND at.attnum > \$2 AND at.attisdropped = \$3	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT pid, username, application_name, client_addr, client_hostname, client_port, backend_start, xact_start, query_start, state_change, wait_event_type, wait_event, state, query FROM pg_stat_activity WHERE datname = \$2 AND backend_start::date = \$1	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT pid, username, application_name, client_addr, client_hostname, client_port, backend_start, xact_start, query_start, state_change, wait_event_type, wait_event, state, query FROM pg_stat_activity WHERE datname = \$3 AND backend_start >= \$1 AND backend_start <= \$2	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT db.oid AS did, db.oid, db.datname AS name, db.dattablespace AS spcoid, spcname, dataallowconn, pg_catalog.pg_encoding_to_char(encoding) AS encoding, pg_catalog.pg_get_userbyid(datdba) AS datowner, (select pg_catalog.current_setting(\$1)) as datcollate, (select pg_catalog.current_setting(\$2)) as datctype, datconnlimit, pg_catalog.has_database_privilege(db.oid, \$3) AS cancreate, pg_catalog.current_setting(\$4) AS default_tablespace, descr.description AS comments, db.datistemplate AS is_template, \$5 AS tblacl, \$6 AS seqacl, \$7 AS funcacl, pg_catalog.array_to_string(datacl::text[], \$8) AS acl FROM pg_catalog.pg_database db LEFT OUTER JOIN pg_catalog.pg_tablespace ta ON db.dattablespace=ta.OID LEFT OUTER JOIN pg_catalog.pg_shdescription descr ON ( db.oid=descr.objoid AND descr.classoid=\$9::regclass ) WHERE db.oid = \$10::OID ORDER BY datname	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT * from ( (SELECT CASE (e.deftype) WHEN \$1 THEN \$2 WHEN \$3 THEN \$4 WHEN \$5 THEN \$6 WHEN \$7 THEN \$8 END AS deftype, \$9 as acltype, COALESCE(gt.rolname, \$10) AS grantee, g.rolname AS grantor, pg_catalog.array_agg(e.privilege_type) as privileges, pg_catalog.array_agg(e.is_grantable) as grantable FROM( SELECT (d.acl).grantee as grantee, (d.acl).grantor AS grantor, (d.acl).is_grantable AS is_grantable, CASE (d.acl).privilege_type WHEN \$11 THEN \$12 WHEN \$13 THEN \$14 WHEN \$15 THEN \$16 WHEN \$17 THEN \$18 WHEN \$19 THEN \$20 WHEN \$21 THEN \$22 WHEN \$23 THEN \$24 WHEN \$25 THEN \$26 WHEN \$27 THEN \$28 WHEN \$29 THEN \$30 WHEN \$31 THEN \$32 WHEN \$33 THEN \$34 ELSE \$35 END AS privilege_type, d.defaclobjtype as deftype FROM (select b.defaclobjtype, pg_catalog.aclxplode(b.revoke_priv) as acl from (select a.defaclobjtype, a.defaclrole, a.defaultprivileges, a.acldefault, array(select unnest(a.acldefault) except select unnest(a.defaultprivileges)) as revoke_priv from (SELECT defaclobjtype, defaclrole, defaclacl as defaultprivileges, CASE WHEN defaclnamespace = \$36 THEN acldefault(CASE WHEN defaclobjtype = \$37 THEN \$38::"char" ELSE defaclobjtype END, defaclrole) ELSE \$39 END AS acldefault FROM pg_catalog.pg_default_acl dacl WHERE dacl.defaclnamespace = \$40::OID ) a ) b where not b.revoke_priv = \$41 ) d ) e LEFT JOIN pg_catalog.pg_roles g ON (e.grantor = g.oid) LEFT JOIN pg_catalog.pg_roles gt ON (e.grantee = gt.oid) GROUP BY g.rolname, gt.rolname, e.deftype ORDER BY e.deftype) UNION ( SELECT CASE (e.deftype) WHEN \$42 THEN \$43 WHEN \$44 THEN \$45 WHEN \$46 THEN \$47 WHEN \$48 THEN \$49 END AS deftype, \$50 as acltype, COALESCE(gt.rolname, \$51) AS grantee, g.rolname AS grantor, pg_catalog.array_agg(e.privilege_type) as privileges, pg_catalog.array_agg(e.is_grantable) as grantable FROM( SELECT (d.acl).grantee as grantee, (d.acl).grantor AS grantor, (d.acl).is_grantable AS is_grantable, CASE (d.acl).privilege_type WHEN \$52 THEN \$53 WHEN \$54 THEN \$55 WHEN \$56 THEN \$57 WHEN \$58 THEN \$59 WHEN \$60 THEN \$61 WHEN \$62 THEN \$63 WHEN \$64 THEN \$65 WHEN \$66 THEN \$67 WHEN \$68 THEN \$69 WHEN \$70 THEN \$71 WHEN \$72 THEN \$73 WHEN \$74 THEN \$75 ELSE \$76 END AS privilege_type, d.defaclobjtype	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)

Usuario	Consulta	Fecha
	as deftype FROM( select *, pg_catalog.aclexplode(b.grant_priv) as acl from (select a.defaclobjtype, a.defaclrole, a.defaultprivileges, a.acldefault, array(select unnest(a.defaultprivileges) except select unnest(a.acldefault)) as grant_priv from (SELECT defaclobjtype, defaclrole, defaclacl as defaultprivileges, CASE WHEN defaclnamespace = \$77 THEN acldefault(CASE WHEN defaclobjtype = \$78 THEN \$79::"char" ELSE defaclobjtype END, defaclrole) ELSE \$80 END AS acldefault FROM pg_catalog.pg_default_acl dacl WHERE dacl.defaclnamespace = \$81::OID ) a ) b where not b.grant_priv = \$82 ) d ) e LEFT JOIN pg_catalog.pg_roles g ON (e.grantor = g.oid) LEFT JOIN pg_catalog.pg_roles gt ON (e.grantee = gt.oid) GROUP BY g.rolname, gt.rolname, e.deftype ORDER BY e.deftype)) f order by f.acltype DESC	
postgres	SELECT db.oid as did, db.datname, db.dataallowconn, pg_encoding_to_char(db.encoding) AS serverencoding, has_database_privilege(db.oid, \$1) as cancreate, datistemplate FROM pg_catalog.pg_database db WHERE db.datname = current_database()	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT at.attname, at.attnum, ty.typname FROM pg_catalog.pg_attribute at LEFT JOIN pg_catalog.pg_type ty ON (ty.oid = at.atttypid) JOIN pg_catalog.pg_class as cl ON cl.oid=AT.attrelid JOIN pg_catalog.pg_namespace as nsp ON nsp.oid=cl.relnamespace WHERE attrelid=\$1::oid AND attnum = ANY ( (SELECT con.conkey FROM pg_catalog.pg_class rel LEFT OUTER JOIN pg_catalog.pg_constraint con ON con.conrelid=rel.oid JOIN pg_catalog.pg_namespace as nsp ON nsp.oid=REL.relnamespace AND con.contype=\$2 WHERE rel.relkind IN (\$3,\$4,\$5, \$6) AND rel.oid = (\$7)::oid )::oid[])	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT version()	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	select * from pg_stat_statements	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	select * from menu	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	/*pga4dash*/ SELECT pid, datname, username, application_name, client_addr, pg_catalog.to_char(backend_start, \$1) AS backend_start, state, wait_event_type    \$2    wait_event AS wait_event, array_to_string(pg_catalog.pg_blocking_pids(pid), \$3) AS blocking_pids, query, pg_catalog.to_char(state_change, \$4) AS state_change, pg_catalog.to_char(query_start, \$5) AS query_start, pg_catalog.to_char(xact_start, \$6) AS xact_start, backend_type, CASE WHEN state = \$7 THEN ROUND((extract(\$8 from now() - query_start) / \$9)::numeric, \$10) ELSE \$11 END AS active_since FROM pg_catalog.pg_stat_activity WHERE datname = (SELECT datname FROM pg_catalog.pg_database WHERE oid = \$12)ORDER BY pid	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT query, calls, total_exec_time, rows, shared_blks_hit, shared_blks_read, shared_blks_dirtied, shared_blks_written FROM pg_stat_statements WHERE dbid = (SELECT oid FROM pg_database WHERE datname = \$1)	Mon Jul 29 2024 19:51:30 GMT-

Usuario	Consulta	Fecha
		0500 (hora de Ecuador)
postgres	SET DateStyle=ISO	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT e.oid, e.evtname AS name, pg_catalog.obj_description(e.oid, \$1) AS comment FROM pg_catalog.pg_event_trigger e ORDER BY e.evtname	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	DELETE FROM menu WHERE idmenu = \$1	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT username FROM pg_user WHERE usesysid = \$1	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT userid, dbid, query, calls, total_exec_time, rows FROM pg_stat_statements WHERE dbid = (SELECT oid FROM pg_database WHERE datname = \$1)	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT nsp.oid, nsp.nspname as name, pg_catalog.has_schema_privilege(nsp.oid, \$1) as can_create, pg_catalog.has_schema_privilege(nsp.oid, \$2) as has_usage, des.description FROM pg_catalog.pg_namespace nsp LEFT OUTER JOIN pg_catalog.pg_description des ON (des.objoid=nsp.oid AND des.classoid=\$3::regclass) WHERE nspname NOT LIKE \$4 AND NOT ( (nsp.nspname = \$5 AND EXISTS (SELECT \$6 FROM pg_catalog.pg_class WHERE relname = \$7 AND relnamespace = nsp.oid LIMIT \$8)) OR (nsp.nspname = \$9 AND EXISTS (SELECT \$10 FROM pg_catalog.pg_class WHERE relname = \$11 AND relnamespace = nsp.oid LIMIT \$12)) OR (nsp.nspname = \$13 AND EXISTS (SELECT \$14 FROM pg_catalog.pg_class WHERE relname = \$15 AND relnamespace = nsp.oid LIMIT \$16)) ) ORDER BY nspname	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT pid, username, application_name, client_addr, client_hostname, client_port, backend_start, xact_start, query_start, state_change, wait_event_type, wait_event, state, query FROM pg_stat_activity WHERE datname = \$1	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)

Usuario	Consulta	Fecha
postgres	SELECT pid, username, application_name, client_addr, client_hostname, client_port, backend_start, xact_start, query_start, state_change, wait_event_type, wait_event, state, query FROM pg_stat_activity WHERE datname = \$2 AND backend_start::date = \$1 AND (now() - backend_start) <= interval \$3	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)
postgres	SELECT \$1 AS deftype, COALESCE(gt.rolname, \$2) AS grantee, g.rolname AS grantor, pg_catalog.array_agg(privilege_type) AS privileges, pg_catalog.array_agg(is_grantable) AS grantable FROM (SELECT d.grantee, d.grantor, d.is_grantable, CASE d.privilege_type WHEN \$3 THEN \$4 WHEN \$5 THEN \$6 WHEN \$7 THEN \$8 WHEN \$9 THEN \$10 WHEN \$11 THEN \$12 WHEN \$13 THEN \$14 WHEN \$15 THEN \$16 WHEN \$17 THEN \$18 WHEN \$19 THEN \$20 WHEN \$21 THEN \$22 WHEN \$23 THEN \$24 WHEN \$25 THEN \$26 ELSE \$27 END AS privilege_type FROM (SELECT (d).grantee AS grantee, (d).grantor AS grantor, (d).is_grantable AS is_grantable, (d).privilege_type AS privilege_type FROM (SELECT pg_catalog.aclexplode(db.datacl) AS d FROM pg_catalog.pg_database db WHERE db.oid = \$28::OID) a ) d ) d LEFT JOIN pg_catalog.pg_roles g ON (d.grantor = g.oid) LEFT JOIN pg_catalog.pg_roles gt ON (d.grantee = gt.oid) GROUP BY g.rolname, gt.rolname ORDER BY grantee	Mon Jul 29 2024 19:51:30 GMT-0500 (hora de Ecuador)