

P-25

Gestión de roles y gerencias

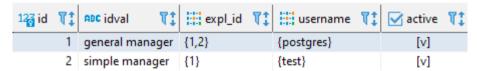
OBJETIVOS

Cuando una organización tiene un tamaño considerable, es necesario organizar la estructura en gerencias. Giswater permite organizar la gestión de la información por gerencias de manera que se asigna un operario a una gerencia, y la gerencia a un conjunto de explotaciones y todo queda vinculado. De este modo se puede gestionar el permiso para consultar unas u otras explotaciones para los usuarios, aunque toda la información se encuentre en el mismo esquema de datos.

DESCRIPCIÓN

La configuración es muy sencilla. Simplemente hay que:

- 1) Poner variable admin_exploitation_x_user de config_param_system en TRUE.
- 2) Llenar los registros de gerencias en la tabla cat_manager donde quedan vinculados:
 - Gerencias
 - Roles
 - Explotaciones



Los nombres de **username** deben existir previamente en la tabla **cat_users**.

3) Después de la configuración, podemos consultar la tabla **config_user_x_expl** dónde se habrán insertado los valores para la relación entre usuario y explotación.



4) Ya podemos comprovar que en el proyecto de QGIS se aplican las restricciones para los usuarios configurados. En este ejemplo, el usuario 'postgres' puede ver las explotaciones 1 y 2, pero el usuario 'test' sólo puede ver la 1.



P-25

Gestión de roles y gerencias

ADDITIONAL

HABILITAR CONFIG_USER_X_SECTOR PARA PERMITIR CREAR SECTORES EN USUARIOS

```
GRANT UPDATE (sector_id) ON cat_manager to role_edit;
GRANT ALL ON ud sample.config user x sector to role edit;
```

BORRAR vu_arc, vu_node, vu_connec & vu_gully INTEGRANDO EN v_arc, v_node, v_connec & v_gully. Example for vu_node:

```
CREATE OR REPLACE VIEW v node AS
WITH vu node query AS
 (SELECT node.node id,
   node......
   Node......
   node.asset id
   FROM ws.node
   LEFT JOIN cat node ON cat node.id::text = node.nodecat id::text
   JOIN cat feature ON cat feature.id::text = cat node.nodetype id::text
   LEFT JOIN dma ON node.dma id = dma.dma id
   LEFT JOIN sector ON node.sector id = sector.sector id
   LEFT JOIN exploitation ON node.expl id = exploitation.expl id
   LEFT JOIN wsqa ON node.dqa id = dqa.dqa id
   LEFT JOIN presszone ON presszone.presszone id::text = node.presszone id::text
   LEFT JOIN ws.v ext streetaxis a ON a.id::text = node.streetaxis id::text
   LEFT JOIN ws.v ext streetaxis b ON b.id::text = node.streetaxis2 id::text)
   SELECT vu_node_query.*
     FROM vu_node_query
     JOIN ws.v_state_node USING (node_id);
DROP VIEW vu_node;
DROP VIEW vu_arc;
DROP VIEW vu_connec;
DROP VIEW vu_gully;
DELETE FROM sys_table WHERE id IN ('vu_node','vu_arc','vu_connec','vu_gully');
```

ELIMINAR FILAS EN TABLAS INP CUANDO SE EJECUTA GO2EPA AUTOREPAIR

```
inp_options_debug→>'autoRepair' ---> FALSE
```

ROW SECUTIRY LEVEL

PARENT TABLES

ALTER TABLE ws.arc ENABLE ROW LEVEL SECURITY;



P-25

Gestión de roles y gerencias

```
CREATE POLICY arc role edit ON ws.arc for all TO role edit using (expl id IN
(SELECT expl id FROM ws.config user x expl WHERE username = current user));
CREATE POLICY arc role basic ON ws.arc for select TO role basic using (expl id
IN (SELECT expl id FROM ws.config user x expl WHERE username = current user);
ALTER TABLE ws.arc DISABLE ROW LEVEL SECURITY;
ALTER TABLE ws.node ENABLE ROW LEVEL SECURITY;
CREATE POLICY node role edit ON ws.node for all TO role edit using (expl id IN
(SELECT expl id FROM ws.config user x expl WHERE username = current user);
CREATE POLICY node role basic ON ws.node for select TO role basic using (expl id
IN (SELECT expl id FROM ws.config user x expl WHERE username = current user));
ALTER TABLE ws. node DISABLE ROW LEVEL SECURITY;
ALTER TABLE ws.connec ENABLE ROW LEVEL SECURITY .........
ALTER TABLE ud.qully ENABLE ROW LEVEL SECURITY ...........
ALTER TABLE ws.link ENABLE ROW LEVEL SECURITY ......
MAPZONES
ALTER TABLE ws.dma ENABLE ROW LEVEL SECURITY.........
ALTER TABLE ud.dga ENABLE ROW LEVEL SECURITY ............
ALTER TABLE ws.presszone ENABLE ROW LEVEL SECURITY ......
```

CHILD MAN

ALTER TABLE ws.man junction ENABLE ROW LEVEL SECURITY; CREATE POLICY man_junction_role_edit ON ud_sample.man_junction for all TO role_edit using (node_id IN (SELECT node id FROM ws.node JOIN ud_sample.config_user_x_expl USING (expl_id) WHERE username = current user)); CREATE POLICY man_junction_role_basic ON ws.man_junction for select role_basic using (node_id IN (SELECT node_id FROM ws.node ud sample.config_user_x_expl USING (expl_id) WHERE username = current_user)); ALTER TABLE ws.node DISABLE ROW LEVEL SECURITY;

CHILD INP

ALTER TABLE ws.inp_junction ENABLE ROW LEVEL SECURITY; CREATE POLICY inp junction role edit ON ud sample.inp junction for all TO (node id IN (SELECT node id FROM ws.node role edit using ud_sample.config_user_x_expl USING (expl_id) WHERE username = current user)); CREATE POLICY inp_junction_role_basic ON ws.inp junction for select role basic using (node id IN (SELECT node id FROM ws.node JOIN ud sample.config user x expl USING (expl id) WHERE username = current user)); ALTER TABLE ws.node DISABLE ROW LEVEL SECURITY;

```
ALTER TABLE ws.inp pump ENABLE ROW LEVEL SECURITY .....
ALTER TABLE ws.inp reservoir ENABLE ROW LEVEL SECURITY.....
ALTER TABLE ws.inp tank ENABLE ROW LEVEL SECURITY.....
ALTER TABLE ws.inp valve ENABLE ROW LEVEL SECURITY .....
ALTER TABLE ws.inp shortpipe ENABLE ROW LEVEL SECURITY .....
ALTER TABLE ws.inp virtualvalve ENABLE ROW LEVEL SECURITY.....
```



P-25

Gestión de roles y gerencias

CAMBIAR EL DV_QUERYTEXT CONFIG_FORM_FIELDS EN SECTOR_ID USANDO JOIN A CONFIG_USER_X_SECTOR

UPDATE config_form_fields SET dv_querytext = 'SELECT sector_id as id, name as idval FROM config_user_x_sector JOIN sector USING (sector_id) WHERE username = current_user and sector_id > -1 and sector.active is true' WHERE columnname = 'sector id'

REVISIONES

Acción	Usuario	Fecha
Creado	Xavi	16/04/2020
Actualizado	Albert	13/09/2021
Actualizado	Xavi	01/12/2021
Actualizado	Albert	18/01/2022

