GISwater

DOCUMENTO PROTOCOLO	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

id	idval	expl_id	username	active
1	general manager	{1,2}	{postgres}	[v]
2	simple manager	{1}	{test}	[v]

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.

expl_id	username	manager_id	active
1	postgres	1	[v]
2	postgres	1	[v]
1	test	2	[v]

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.

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;
```

PREVENIR 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;
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.gully ENABLE ROW LEVEL SECURITY.....
ALTER TABLE ws.link ENABLE ROW LEVEL SECURITY.....
```

MAPZONES

```
ALTER TABLE ws.dma ENABLE ROW LEVEL SECURITY.....
ALTER TABLE ud.dqa 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 TO
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;
```

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 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 inp_junction_role_basic ON ws.inp_junction for select TO 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……
```

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