

P-25

Role administration and management

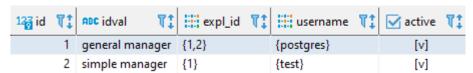
GOALS

When an organization is of considerable size, it is necessary to organize the structure into managements. Giswater allows you to organize the management of information by managers in such a way that an operator is assigned to a management, and the management to a set of exploitations and everything is linked. In this way, it is possible to manage the permission to consult one or other exploitations for users, even if all the information is in the same data schema.

DESCRIPTION

The configuration is very simple. You just have to:

- 1) Put variable admin_exploitation_x_user of config_param_system in TRUE.
- 2) Fill the management records in the **cat_manager** table where they are linked:
 - Management
 - Roles
 - Exploitations



The names of **username** must already exist in the table **cat_users**.

3) After the configuration, we can consult the table **config_user_x_expl** where the values for the relationship between user and exploitation will have been inserted.



4) We can now verify that the restrictions for configured users are applied in the QGIS project. In this example, the user 'postgres' can see exploits 1 and 2, but the user 'test' can only see the 1.



P-25

Role administration and management

ADDITIONAL

ENABLE CONFIG_X_USER TO MANAGE PERMISSIONS FOR CREATE SECTORS BY USERS

```
GRANT UPDATE (sector_id) ON cat_manager to role_edit;
GRANT ALL ON config_user_x_sector to role_edit;
```

REMOVE vu_arc, vu_node, vu_connec & vu_gully INTEGRATING THEM ON 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 wsga ON node.dga id = dga.dga 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');
```

PREVENT REMOVE ROWS ON INP TABLES WHEN GO2EPA AUTOREPAIR

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



P-25

Role administration and management

ROW SECURITY 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.dga ENABLE ROW LEVEL SECURITY ............
ALTER TABLE ws.presszone ENABLE ROW LEVEL SECURITY ............
CHILD MAN
ALTER TABLE ws.man junction ENABLE ROW LEVEL SECURITY;
```

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 tank ENABLE ROW LEVEL SECURITY.....



P-25

Role administration and management

```
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.....
```

CHANGE CONFIG_FORM_FIELDS DV_QUERYTEXT FOR SECTOR_ID USING JOIN TO 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'

REVIEWS

Action	User	Date
Created	Xavi	16/04/2020
Updated	Albert	13/09/2021
Updated	Xavi	01/12/2021
Updated	Xavi	30/12/2021
Updated	Albert	18/01/2022