



Security

Ownership & Permissions

Martin Reisinger, presenter



Security 2 concepts intersect

- Ownership
 - Permissions (to update, delete)
- 



Ownership

- only 1 owner per record
- an owner can transfer ownership to another user
- an owner can provide permissions (Update, Delete) to multiple users



Ownership

- ▶ “What to do? The owner is no longer here!”
- ▶ Zeus can do anything
- ▶ In GG, Zeus is the database administrator (dba)
 - ▶ In NPGS, Zeus is Benjamin Haag

email him when ownership issues cannot be resolved at the site level



Not always true!

- ▶ If I create the record, I am the owner
- 



Permissions

If I own a record, I can designate my CT colleagues to be able to update or delete the records (or not)



Permissions

A permission of type:	Has the ability to:
Read*	Read existing data
Update	Update existing data
Delete	Delete existing data
Create*	Insert new data

* in the CT, ignore these types



Possible Permission Values

Value	Description
Allow	Allows access
Deny	Denies access
Inherit	Neither allows nor denies access; access is situational; it is inherited from a previous definition (typically the permission value of the parent table)

When relationships are mapped between dataviews

...the children tables inherit the security settings of the parent

Example: if someone creates an **accession_inventory_name** record, the owner is the same as the owner of the parent record, in this case the **inventory** record

Admin
Tool
screen

The screenshot shows the GRIN-Global Admin v 1.9.1.0 interface. The title bar reads "GRIN-Global Admin v 1.9.1.0 - [Table Mapping - accession_inv_name - localhost\sqlexpress - Administrator]". The menu bar includes "File", "View", "Tools", and "Help". The breadcrumb path is "localhost\sqlexpress - sqlserver > Table Mappings > accession_inv_name".


On the left, a tree view shows "Table Mappings" with a list of tables. "accession_inv_name" is selected and highlighted in blue.

The main area shows the "Table Mapping" configuration for "accession_inv_name". The "Database Area" is set to "ACCESSION" and the "Enabled" checkbox is checked. The "Fields" tab is active, displaying a table of field mappings:

Child	Type	Parent
accession_inv_name.created_by	Parent	cooperator.cooperator_id
accession_inv_name.modified_by	Parent	cooperator.cooperator_id
accession_inv_name.name_source...	Parent	cooperator.cooperator_id
accession_inv_name.owned_by	Parent	cooperator.cooperator_id
accession_inv_name.inventory_id	Parent and owner	inventory.inventory_id
accession_inv_name.name_group_id	Parent	name_group.name_group_id



Inheritance only cascades one level



You may need to give permissions at the accession, the inventory, and the order requests

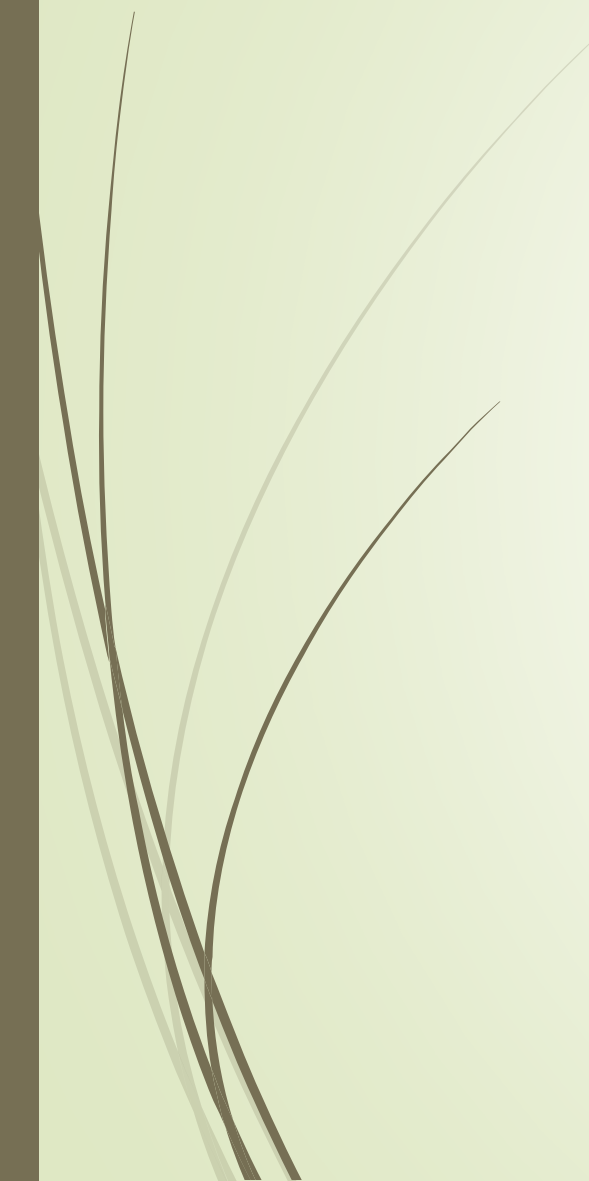


Let's do it





Demos





Special *Permission* Groups



Special *Permission* Groups

Site "Power Users" In NPGS, each site has a group defined

Admin
Tool
screen

Tag:

Description:

Applies To Data View:

Applies To Table:

Enabled

Create: Read: Update: Delete:

Restricted To:

Resource	Field	Compare	Value
site	site_id	=	16

“Super Coop” Editors

Admin
Tool
screen

Name:

Tag: **MANAGE_COOPERATOR**

Description:

Applies To Data View: -- Any Data View --

Applies To Table: **cooperator**

Enabled

Create:	Read:	Update:	Delete:
Allow	Allow	Allow	Allow

Restricted To:

Resource	Field	Compare	Value
----------	-------	---------	-------


SQL for Determining Permissions

```
SELECT su.user_name, sg.group_tag, sp.permission_tag, st.table_name,  
CONCAT(c.first_name, ' ', c.last_name) AS owner  
FROM sys_user su  
JOIN sys_group_user_map sgum ON sgum.sys_user_id = su.sys_user_id  
JOIN sys_group sg ON sg.sys_group_id = sgum.sys_group_id  
JOIN sys_group_permission_map sgpm ON sgpm.sys_group_id = sg.sys_group_id  
JOIN sys_permission sp ON sp.sys_permission_id = sgpm.sys_permission_id  
LEFT JOIN sys_table st ON st.sys_table_id = sp.sys_table_id  
JOIN cooperator c ON c.cooperator_id = sp.created_by  
WHERE sp.owned_by != 48  
AND user_name LIKE '%reisinger%'
```




SQL for Determining Ownership

```
SELECT st1.table_name AS child, st2.table_name AS owner
FROM sys_table_relationship str
JOIN sys_table_field stf1 ON stf1.sys_table_field_id = str.sys_table_field_id
JOIN sys_table st1 ON st1.sys_table_id = stf1.sys_table_id
JOIN sys_table_field stf2 ON stf2.sys_table_field_id = str.other_table_field_id
JOIN sys_table st2 ON st2.sys_table_id = stf2.sys_table_id
WHERE relationship_type_tag = 'OWNER_PARENT'
```



SQL for determining owner-parent relationships

```
SELECT st1.table_name AS child, st2.table_name AS owner
FROM sys_table_relationship str
JOIN sys_table_field stf1 ON stf1.sys_table_field_id = str.sys_table_field_id
JOIN sys_table st1 ON st1.sys_table_id = stf1.sys_table_id
JOIN sys_table_field stf2 ON stf2.sys_table_field_id = str.other_table_field_id
JOIN sys_table st2 ON st2.sys_table_id = stf2.sys_table_id
WHERE relationship_type_tag = 'OWNER_PARENT'
```



References

- Security: Ownership & Permissions
- Cooperators