Grouper SQL interface

This topic is discussed in the Advanced Topics training video.

Grouper's views can be granted to schemas to make it possible to access Grouper data from SQL.

Notes/Warnings

- If you do grant Grouper views to schemas you should keep track of what you grant to do some sanity tests on Grouper upgrades.
- Some views are for reporting reasons and are not intended for real time data.
- Check the views that you use to see the performance.
- The SQL interface is useful for large loads that might overwhelm or not be performant or not be possible in web services.
- This SQL interface is for SELECT only, not insert/update/delete, and is not used with the Grouper API, just SQL queries in whatever programming language you would like. Ask the Grouper team for help if you are not sure how to use the tables

Best practices to using the SQL interface

1. Do not use the Grouper schema for accesses to the Grouper SQL interface, make another schema which is readonly and only has access to the required data (where clause)
2. Make a view on top of the Grouper tables/views and grant that to the other schema. This is because when Grouper is upgraded the views are dropped (including grants), so they would be lost or need to be recreated

Groups with attribute

```
select
  gaagv.group_name
from
  grouper_attr_asn_group_v gaagv, grouper_groups gg
where
  gaagv.attribute_def_name_name = 'test:attrName'
  and gaagv.group_id = gg.id
  and gaagv.enabled = 'T'
  and gg.enabled = 'T'
```

Groups under stem with attribute

```
select
  gg.name
from
  grouper_attr_asn_stem_v gaasv, grouper_stem_set gss, grouper_groups gg
where
  gaasv.attribute_def_name_name = 'test:attrDef1'
  and gaasv.stem_id = gss.then_has_stem_id
  and gg.parent_stem = gss.if_has_stem_id
  and gg.enabled = 'T'
  and gaasv.enabled = 'T'
```

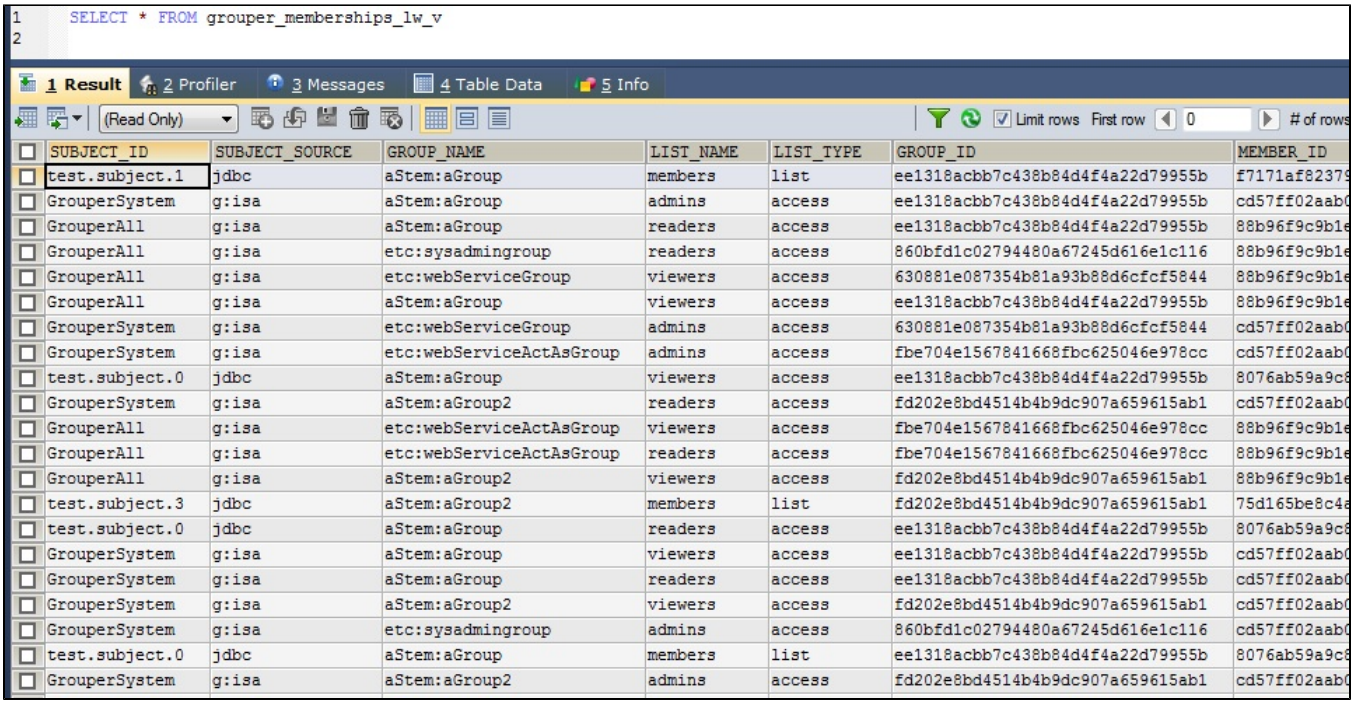
People added to group after a certain date (micros since 1970)

```
select distinct gm.subject_id, gm.description from grouper_pit_memberships_lw_v gpmlv, grouper_pit_groups gpg,
grouper_members gm
where gpg.source_id ='b9452d5e6a824b598a5fc83939f2677d'
and gpg.id  = owner_group_id  and the_start_time > '1616385600000000'
and gm.id = gpmlv.member_id
;
```

Memberships

The main views to use are:

- grouper_membership_lw_v: This view can be used for memberships or Grouper group privileges (e.g. if someone can ADMIN a group). Note: make sure you have the proper where clause which selects the proper list (generally "members").



```
1  SELECT * FROM grouper_memberships_lw_v
2
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info

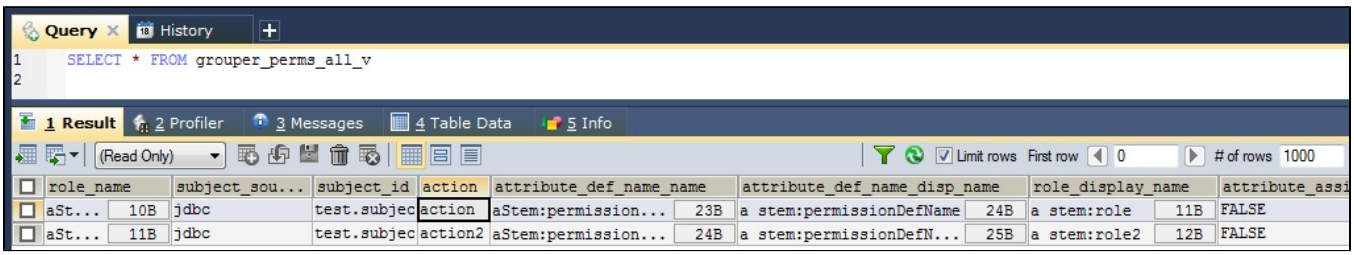
(Read Only) | Limit rows | First row 0 | # of rows

SUBJECT_ID	SUBJECT_SOURCE	GROUP_NAME	LIST_NAME	LIST_TYPE	GROUP_ID	MEMBER_ID
test.subject.1	jdbc	aStem:aGroup	members	list	ee1318acbb7c438b84d4f4a22d79955b	f7171af82379
GrouperSystem	g:isa	aStem:aGroup	admins	access	ee1318acbb7c438b84d4f4a22d79955b	cd57ff02aab0
GrouperAll	g:isa	aStem:aGroup	readers	access	ee1318acbb7c438b84d4f4a22d79955b	88b96f9c9b1e
GrouperAll	g:isa	etc:sysadmingroup	readers	access	860bfd1c02794480a67245d616e1c116	88b96f9c9b1e
GrouperAll	g:isa	etc:webServiceGroup	viewers	access	630881e087354b81a93b88d6cfcf5844	88b96f9c9b1e
GrouperAll	g:isa	aStem:aGroup	viewers	access	ee1318acbb7c438b84d4f4a22d79955b	88b96f9c9b1e
GrouperSystem	g:isa	etc:webServiceGroup	admins	access	630881e087354b81a93b88d6cfcf5844	cd57ff02aab0
GrouperSystem	g:isa	etc:webServiceActAsGroup	admins	access	fbe704e1567841668fbc625046e978cc	cd57ff02aab0
test.subject.0	jdbc	aStem:aGroup	viewers	access	ee1318acbb7c438b84d4f4a22d79955b	8076ab59a9c8
GrouperSystem	g:isa	aStem:aGroup2	readers	access	fd202e8bd4514b4b9dc907a659615ab1	cd57ff02aab0
GrouperAll	g:isa	etc:webServiceActAsGroup	viewers	access	fbe704e1567841668fbc625046e978cc	88b96f9c9b1e
GrouperAll	g:isa	etc:webServiceActAsGroup	readers	access	fbe704e1567841668fbc625046e978cc	88b96f9c9b1e
GrouperAll	g:isa	aStem:aGroup2	viewers	access	fd202e8bd4514b4b9dc907a659615ab1	88b96f9c9b1e
test.subject.3	jdbc	aStem:aGroup2	members	list	fd202e8bd4514b4b9dc907a659615ab1	75d165be8c4a
test.subject.0	jdbc	aStem:aGroup	readers	access	ee1318acbb7c438b84d4f4a22d79955b	8076ab59a9c8
GrouperSystem	g:isa	aStem:aGroup	viewers	access	ee1318acbb7c438b84d4f4a22d79955b	cd57ff02aab0
GrouperSystem	g:isa	aStem:aGroup	readers	access	ee1318acbb7c438b84d4f4a22d79955b	cd57ff02aab0
GrouperSystem	g:isa	aStem:aGroup2	viewers	access	fd202e8bd4514b4b9dc907a659615ab1	cd57ff02aab0
GrouperSystem	g:isa	etc:sysadmingroup	admins	access	860bfd1c02794480a67245d616e1c116	cd57ff02aab0
test.subject.0	jdbc	aStem:aGroup	members	list	ee1318acbb7c438b84d4f4a22d79955b	8076ab59a9c8
GrouperSystem	g:isa	aStem:aGroup2	admins	access	fd202e8bd4514b4b9dc907a659615ab1	cd57ff02aab0

This view contains all memberships (direct, indirect, groups, people, etc). To return a simple list of effective membership for people try:

```
SELECT DISTINCT subject_id FROM grouper_memberships_lw_v WHERE group_name='{path:to:group}' AND
subject_source='{people_subject_source}' AND list_name ='members'
```

- grouper_perms_all_v: This view can be used for Grouper permissions. Note, DISALLOW is exposed through the SQL interface, and is not calculated. So if DISALLOW is assigned in the data being used, you should either not use the SQL interface, or you would need to calculate the DISALLOWs after selecting the data (note, this is a complicated algorithm though it is documented on the wiki with examples).



Query | History

```
1  SELECT * FROM grouper_perms_all_v
2
```

1 Result | 2 Profiler | 3 Messages | 4 Table Data | 5 Info

(Read Only) | Limit rows | First row 0 | # of rows 1000

role_name	subject_sou...	subject_id	action	attribute_def_name_name	attribute_def_name_disp_name	role_display_name	attribute_assi
aSt... 10B	jdbc	test.subjec	action	aStem:permission... 23B	a stem:permissionDefName 24B	a stem:role 11B	FALSE
aSt... 11B	jdbc	test.subjec	action2	aStem:permission... 24B	a stem:permissionDefN... 25B	a stem:role2 12B	FALSE

- There are some useful tables to join to: grouper_groups, grouper_members, grouper_stems, etc

Privileges

Find all group admins

```
select group_name, subject_id from grouper_memberships_lw_v gmlv where list_name = 'admins' and SUBJECT_SOURCE
= 'mySubjectSource'
```

Find all stem admins

```
select stem_name, subject_id from grouper_mship_stem_lw_v gmslv where gmslv.LIST_NAME = 'stemAdmins' and gmslv.
SUBJECT_SOURCE = 'mySubjectSource' ;
```

Find groups someone can update (or admin)

```
select distinct gmlv.group_name from grouper_memberships_lw_v gmlv where gmlv.list_name in ('admins',
'updaters')
and gmlv.subject_source = 'xyz' and gmlv.subject_id = '123' order by gmlv.group_name limit 100
```

Groups provisionable to a provisioner

```
select * from grouper_sync gs, grouper_sync_group gsg
where gs.provisioner_name = 'blah'
and gs.id = gsg.grouper_sync_id
and gsg.provisionable = 'T'
```

Groups with provisioning assignments for a provisioner

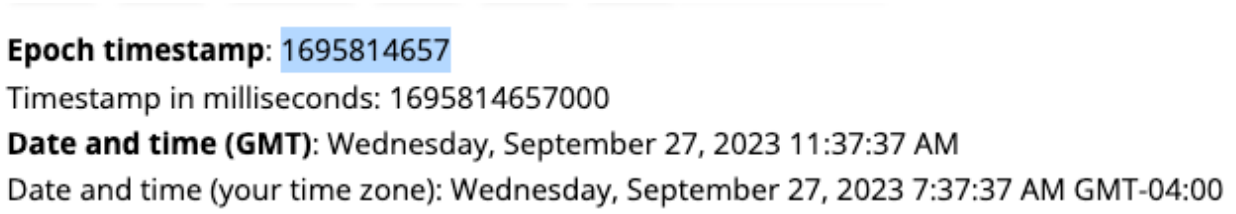
Replace provisionerId with the ID of the provisioner

```
select gaaagv.group_name, 'group' as owner_type from grouper_aval_asn_asn_group_v gaaagv
where gaaagv.attribute_def_name_name2 like '%etc:provisioning:provisioningTarget'
and gaaagv.value_string = 'provisionerId'
union all
select gaaasv.stem_name, 'stem' as owner_type from grouper_aval_asn_asn_stem_v gaaasv
where gaaasv.attribute_def_name_name2 like '%etc:provisioning:provisioningTarget'
and gaaasv.value_string = 'provisionerId'
```

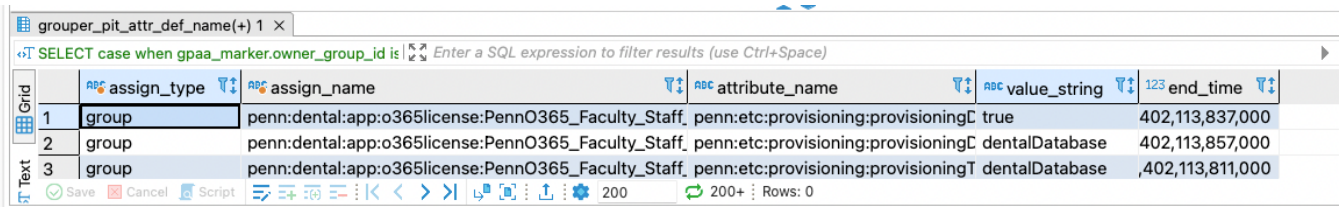
SQL loader membership queries

```
SELECT gaaag.group_name, gaaag.value_string  FROM penngrouper.grouper_aval_asn_asn_group_v gaaag where gaaag.
attribute_def_name_name2 like '%grouperLoaderQuery'
```

Point in time attribute assignments (marker and name/value pairs like provisioning)



```
create table temp_provisioning_assigns as
SELECT
  gpaa_marker.source_id as attribute_assign_id_marker,
  case when gpaa_marker.owner_group_id is not null then 'group'
  when gpaa_marker.owner_stem_id is not null then 'stem'
  end as assign_type,
  case
    when gpaa_marker.owner_group_id is not null then gpg.name
    when gpaa_marker.owner_stem_id is not null then gps.name
  end as assign_name,
  gpadn_value.name attribute_name,
  gpaav.value_string,
  gpaav.end_time
   FROM grouper_pit_attribute_assign gpaa_marker
    left outer join grouper_pit_groups gpg on gpaa_marker.owner_group_id = gpg.id
    left outer join grouper_pit_stems gps on gpaa_marker.owner_stem_id = gps.id
    join grouper_pit_attribute_assign gpaa_value on gpaa_marker.id = gpaa_value.owner_attribute_assign_id
    join grouper_pit_attr_assn_value gpaav on gpaa_value.id = gpaav.attribute_assign_id
    join grouper_pit_attr_def_name gpadn_marker on gpaa_marker.attribute_def_name_id = gpadn_marker.id
      and gpadn_marker.name = 'etc:provisioning:provisioningMarker'
    join grouper_pit_attr_def_name gpadn_value on gpaa_value.attribute_def_name_id = gpadn_value.id
  WHERE
    (gpaa_marker.owner_group_id is not null or gpaa_marker.owner_stem_id is not null)
    and gpaav.end_time/1000000 > 1695814657
    order by 1, 2, 3, 4;
```



Make a SQL to generate GSH

```
select distinct attribute_assign_id_marker, assign_name,
'new Provisionable' ||
  (case when exists (select 1 from temp_provisioning_assigns tpa2 where tpa.attribute_assign_id_marker = tpa2.
attribute_assign_id_marker
  and tpa2.assign_type = 'group') then
    'Group' else 'Stem' end)
  || 'Save().assignTargetName("'
  || (select tpa2.value_string from temp_provisioning_assigns tpa2 where tpa.attribute_assign_id_marker = tpa2.
attribute_assign_id_marker
and tpa2.attribute_name = 'etc:provisioning:provisioningTarget')
  || '").assign' ||
  (case when exists (select 1 from temp_provisioning_assigns tpa2 where tpa.attribute_assign_id_marker = tpa2.
attribute_assign_id_marker
  and tpa2.assign_type = 'group') then
    'Group' else 'Stem' end) ||
  'Name("' ||
  (select distinct assign_name from temp_provisioning_assigns tpa2 where tpa.attribute_assign_id_marker = tpa2.
attribute_assign_id_marker) ||
   '")"' ||
   (case when exists (select 1 from temp_provisioning_assigns tpa2 where tpa.attribute_assign_id_marker = tpa2.
attribute_assign_id_marker
  and tpa2.assign_type = 'group') then
    'Group' else 'Stem' end)
   || '".save();'
as script from temp_provisioning_assigns tpa
where exists (select 1 from temp_provisioning_assigns tpa2
where tpa.attribute_assign_id_marker = tpa2.attribute_assign_id_marker and tpa2.value_string = 'true'
and tpa2.attribute_name = 'etc:provisioning:provisioningDirectAssign')
and assign_type = 'stem'
order by 2;
```