

# Grouper SQL interface

- [Wiki Home](#)
- [Download Grouper](#)
- [Grouper Guides](#)
- [Community Contributions](#)
- [Developer Resources](#)
- [Grouper Website](#)

 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. Note, if you do this you should keep track of what you grant to do some sanity tests on Grouper upgrades. Note, 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

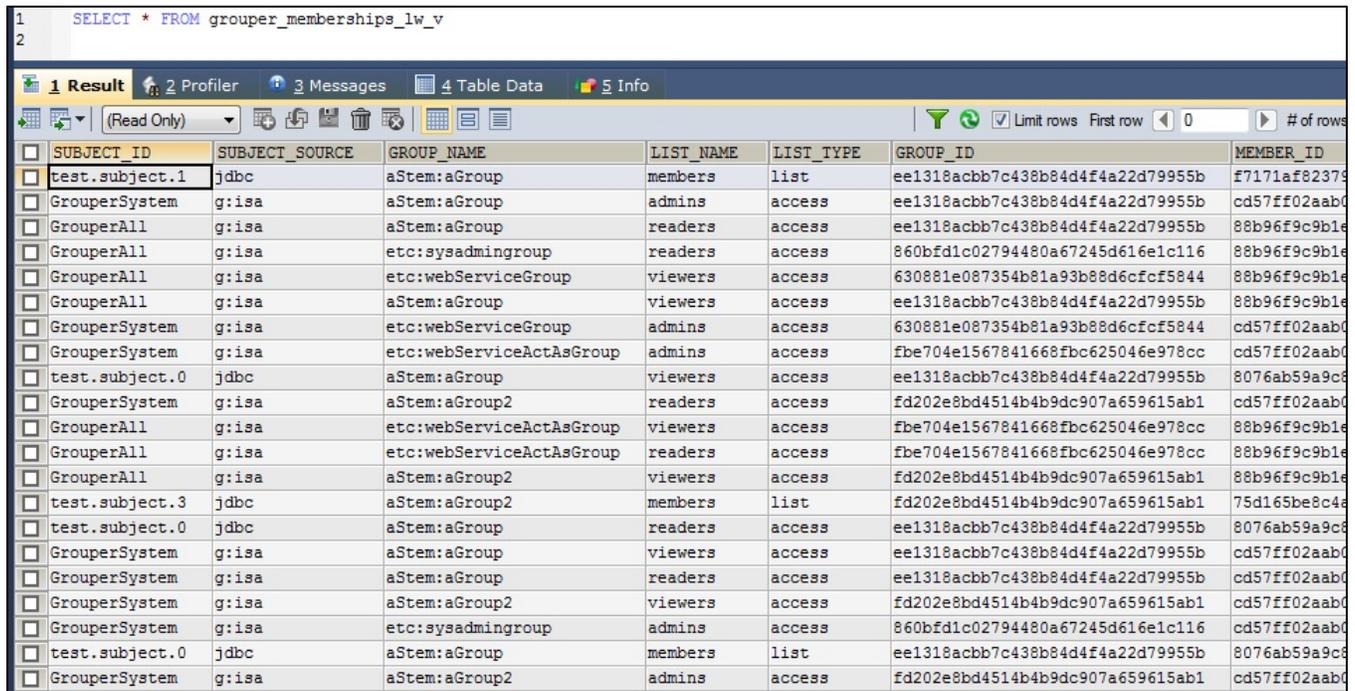
Best practices to using the SQL interface

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

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
```



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	8076ab59a9c6
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	75d165be8c4e
test.subject.0	jdbc	aStem:aGroup	readers	access	ee1318acbb7c438b84d4f4a22d79955b	8076ab59a9c6
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	8076ab59a9c6
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_membership_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 x 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