Migrating from the Grouper JDBC subject source to the JDBC2 subject source

The JDBC subject source has queries to search by ID, identifier, and freeform. The JDBC2 subject source, has a table or view with those columns, and constructs the queries for you. If possible you should use the JDBC2 subject source. The JDBC source has issues with pooling if you have a less than reliable network.

Ideally you would **create a table or materialized view** and migrate data there, and build the JDBC2 subject source. This will have the best performance.

Here are some sample columns you need:

- subject_id (opaque id)
- netId (blank for external users)
- eppn
- name
- email
- description (how you want the subject to display)
- lower_search_field (lower case list of things to search e.g. comma separated. this might just be the lower case description)
- first (if you want for subject attribute)
- last (if you want for subject attribute)
- any other things you want to expose as subject attributes, usually this is a short list since Grouper is not your IdM

If you cannot load data into a table or materialized view, then you should **make a view** of the data and build the JDBC2 subject source from that view. Note, if you have a flattened table already (no concatenations, joins, etc), then a view is fine to use (table would not be faster)

If you cannot do either of those, you can **make a query** that will be used as a table

For instance, here is a table of users (example from florida)

ARP_PERSON_INFO: Created: 5/14/2018 4:06:05 PM Last DDL: 5/14/2018 4:38:49 PM

Primary Key: <none>

Columns | Indexes | Constraints | Triggers | Data | Script | Grants | Synonyms | Partitions | Subpartitions | Stats/Size | R

Column Name	I.	PK	Index Pos	Null?	Data Type	Defau
ARP_PAR_UFID	1			N	VARCHAR2 (8 Char)	
ARP_PAR_GLID	2			N	VARCHAR2 (16 Char)	
ARP_PAR_BUSINESS_NM	3			N	VARCHAR2 (44 Char)	
ARP_PAR_DISPLAY_NM	4			N	VARCHAR2 (44 Char)	
ARP_PAR_DIR_NM	5			N	VARCHAR2 (44 Char)	
ARP_PAR_FIRST_NM	6			N	VARCHAR2 (44 Char)	
ARP_PAR_MIDDLE_NM	7			N	VARCHAR2 (44 Char)	
ARP_PAR_LAST_NM	8			N	VARCHAR2 (44 Char)	
ARP_PAR_SUFFIX_NM	9			N	VARCHAR2 (44 Char)	
ARP_PAR_PHONE_STRING	10			N	VARCHAR2 (30 Char)	
ARP_PAR_POSTAL_ADDR	11			N	VARCHAR2 (254 Char)	
ARP_PAR_UF_EMAIL_AD	12			N	VARCHAR2 (254 Char)	
ARP_PAR_STATUS_FLG	13			N	VARCHAR2 (1 Char)	
ARP_PAR_PROTECT_FLG	14			N	VARCHAR2 (1 Char)	
ARP_PAR_PRI_AFF_TYPE	15			N	VARCHAR2 (1 Char)	
ARP_PAR_PRIMARY_DEPTID	16			N	VARCHAR2 (8 Char)	
ARP_PAR_CLASS_COLLEGE	17			N	VARCHAR2 (3 Char)	
ARP_PAR_MAJOR_LIST	18			N	VARCHAR2 (254 Char)	
ARP_PAR_MINOR_LIST	19			N	VARCHAR2 (254 Char)	
ARP_PAR_LOA	20			N	VARCHAR2 (20 Char)	
ARP_PAR_CERTIFICATES_LIST	21			N	VARCHAR2 (254 Char)	
ARP_PAR_NET_MANAGEDBY	22			N	VARCHAR2 (8 Char)	''

```
CREATE TABLE ARP_PERSON_INFO
(
  ARP_PAR_UFID               VARCHAR2(8 CHAR)   NOT NULL,
  ARP_PAR_GLID               VARCHAR2(16 CHAR)  NOT NULL,
  ARP_PAR_BUSINESS_NM        VARCHAR2(44 CHAR)  NOT NULL,
  ARP_PAR_DISPLAY_NM         VARCHAR2(44 CHAR)  NOT NULL,
  ARP_PAR_DIR_NM             VARCHAR2(44 CHAR)  NOT NULL,
  ARP_PAR_FIRST_NM           VARCHAR2(44 CHAR)  NOT NULL,
  ARP_PAR_MIDDLE_NM          VARCHAR2(44 CHAR)  NOT NULL,
  ARP_PAR_LAST_NM            VARCHAR2(44 CHAR)  NOT NULL,
  ARP_PAR_SUFFIX_NM          VARCHAR2(44 CHAR)  NOT NULL,
  ARP_PAR_PHONE_STRING       VARCHAR2(30 CHAR)  NOT NULL,
  ARP_PAR_POSTAL_ADDR        VARCHAR2(254 CHAR) NOT NULL,
  ARP_PAR_UF_EMAIL_AD        VARCHAR2(254 CHAR) NOT NULL,
  ARP_PAR_STATUS_FLG         VARCHAR2(1 CHAR)   NOT NULL,
  ARP_PAR_PROTECT_FLG        VARCHAR2(1 CHAR)   NOT NULL,
  ARP_PAR_PRI_AFF_TYPE       VARCHAR2(1 CHAR)   NOT NULL,
  ARP_PAR_PRIMARY_DEPTID     VARCHAR2(8 CHAR)   NOT NULL,
  ARP_PAR_CLASS_COLLEGE      VARCHAR2(3 CHAR)   NOT NULL,
  ARP_PAR_MAJOR_LIST         VARCHAR2(254 CHAR) NOT NULL,
  ARP_PAR_MINOR_LIST         VARCHAR2(254 CHAR) NOT NULL,
  ARP_PAR_LOA                VARCHAR2(20 CHAR)  NOT NULL,
  ARP_PAR_CERTIFICATES_LIST  VARCHAR2(254 CHAR) NOT NULL,
  ARP_PAR_NET_MANAGEDBY      VARCHAR2(8 CHAR)   DEFAULT ' '                  NOT NULL
);

Insert into ARP_PERSON_INFO
   (ARP_PAR_UFID, ARP_PAR_GLID, ARP_PAR_BUSINESS_NM, ARP_PAR_DISPLAY_NM, ARP_PAR_DIR_NM,
    ARP_PAR_FIRST_NM, ARP_PAR_MIDDLE_NM, ARP_PAR_LAST_NM, ARP_PAR_SUFFIX_NM, ARP_PAR_PHONE_STRING,
    ARP_PAR_POSTAL_ADDR, ARP_PAR_UF_EMAIL_AD, ARP_PAR_STATUS_FLG, ARP_PAR_PROTECT_FLG, ARP_PAR_PRI_AFF_TYPE,
    ARP_PAR_PRIMARY_DEPTID, ARP_PAR_CLASS_COLLEGE, ARP_PAR_MAJOR_LIST, ARP_PAR_MINOR_LIST, ARP_PAR_LOA,
    ARP_PAR_CERTIFICATES_LIST, ARP_PAR_NET_MANAGEDBY)
 Values
   ('123', 'mchyzer', ' ', ' ', ' ',
    'Chris', 'Michael', 'Hyzer', ' ', ' ',
    ' ', 'mchyzer@isc.upenn.edu', 'A', 'F', 'S',
    'ISC', ' ', ' ', ' ', ' ',
    ' ', ' ');
COMMIT;
```

For instance, if this is the query:

```
SELECT id,
       netId,
       name,
       email,
       description,
       LOWER (description) AS search_description
  FROM (SELECT s.ARP_PAR_GLID || '@ufl.edu' AS id,
               s.ARP_PAR_GLID AS netId,
               s.ARP_PAR_UF_EMAIL_AD AS email,
                  TRIM (s.ARP_PAR_FIRST_NM)
               || ' '
               || CASE
                     WHEN LENGTH (TRIM (s.ARP_PAR_MIDDLE_NM)) > 0
                     THEN
                        TRIM (s.ARP_PAR_MIDDLE_NM) || ' '
                  END
               || CASE
                     WHEN LENGTH (TRIM (s.ARP_PAR_LAST_NM)) > 0
                     THEN
                        TRIM (s.ARP_PAR_LAST_NM)
                  END
               || CASE
                     WHEN LENGTH (TRIM (s.ARP_PAR_SUFFIX_NM)) > 0
                     THEN
                        ' ' || TRIM (s.ARP_PAR_SUFFIX_NM)
                  END
                  AS name,
                  TRIM (s.ARP_PAR_FIRST_NM)
               || ' '
               || CASE
                     WHEN LENGTH (TRIM (s.ARP_PAR_MIDDLE_NM)) > 0
                     THEN
                        TRIM (s.ARP_PAR_MIDDLE_NM) || ' '
                  END
               || CASE
                     WHEN LENGTH (TRIM (s.ARP_PAR_LAST_NM)) > 0
                     THEN
                        TRIM (s.ARP_PAR_LAST_NM)
                  END
               || CASE
                     WHEN LENGTH (TRIM (s.ARP_PAR_SUFFIX_NM)) > 0
                     THEN
                        ' ' || TRIM (s.ARP_PAR_SUFFIX_NM)
                  END
               || ', '
               || s.ARP_PAR_GLID
               || '@ufl.edu, '
               || s.ARP_PAR_PRI_AFF_TYPE
               || ', '
               || s.ARP_PAR_PRIMARY_DEPTID
               || ', '
               || s.ARP_PAR_UF_EMAIL_AD
                  AS description
          FROM ARP_PERSON_INFO s
         WHERE s.ARP_PAR_STATUS_FLG = 'A')
```

ID	NETID	NAME	EMAIL	DESCRIPTION	SEARCH_DESCRIPTION
mchyzer@ufl.edu	mchyzer	Chris Michael Hyzer	mchyzer@isc.upenn.edu	Chris Michael Hyzer, mchyzer@ufl.edu, S, ISC, mchyzer@isc.upenn.edu	chris michael hyzer, mchyzer@ufl.edu, s, isc, mchyzer@isc.upenn.edu

You can use that query to setup a subject source in jdbc2 even though you dont have a table or view

```
##########################################
```

```
## Configuration for source id: floridaPerson
## Source configName: floridaPerson
########################################
subjectApi.source.floridaPerson.id = floridaPerson

# this is a friendly name for the source
subjectApi.source.floridaPerson.name = Florida person

# type is not used all that much.  Can have multiple types, comma separate.  Can be person, group, application
subjectApi.source.floridaPerson.types = person

# the adapter class implements the interface: edu.internet2.middleware.subject.Source
# adapter class must extend: edu.internet2.middleware.subject.provider.BaseSourceAdapter
# edu.internet2.middleware.grouper.subj.GrouperJdbcSourceAdapter2  :  if doing JDBC this should be used if
possible.  All subject data in one table/view.
# edu.internet2.middleware.grouper.subj.GrouperJdbcSourceAdapter   :  oldest JDBC source.  Put freeform queries
in here
# edu.internet2.middleware.grouper.subj.GrouperJndiSourceAdapter   :  used for LDAP
subjectApi.source.floridaPerson.adapterClass = edu.internet2.middleware.subject.provider.JDBCSourceAdapter2

subjectApi.source.floridaPerson.param.jdbcConnectionProvider.value = edu.internet2.middleware.subject.provider.
C3p0JdbcConnectionProvider

subjectApi.source.floridaPerson.param.email.value = mail

subjectApi.source.floridaPerson.param.maxPageSize.value = 100

subjectApi.source.floridaPerson.param.dbDriver.value = oracle.jdbc.driver.OracleDriver

subjectApi.source.floridaPerson.param.dbUrl.value = jdbc:oracle:thin:@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)
(HOST = orrcdbdv2-clstr.seo.int)(PORT = 1521))  (CONNECT_DATA =  (SERVER = DEDICATED)   (SERVICE_NAME = dcom)
)  )

subjectApi.source.floridaPerson.param.dbUser.value = pcdadmin

subjectApi.source.floridaPerson.param.dbPwd.value = jyc7wmz2sbr6dw

# maximum number of results from a search, generally no need to get more than 1000
subjectApi.source.floridaPerson.param.maxResults.value = 1000

# the table or view to query results from.  Note, could prefix with a schema name
subjectApi.source.floridaPerson.param.dbTableOrView.value = (select id, netId, name, email, description, lower
(description) as search_description from ( select s.ARP_PAR_GLID || '@ufl.edu' AS id, s.ARP_PAR_GLID as netId,
s.ARP_PAR_UF_EMAIL_AD as email, TRIM (s.ARP_PAR_FIRST_NM) || ' ' || CASE WHEN LENGTH (TRIM (s.
ARP_PAR_MIDDLE_NM)) > 0 THEN TRIM (s.ARP_PAR_MIDDLE_NM) || ' ' END || CASE WHEN LENGTH (TRIM (s.
ARP_PAR_LAST_NM)) > 0 THEN TRIM (s.ARP_PAR_LAST_NM) END || CASE WHEN LENGTH (TRIM (s.ARP_PAR_SUFFIX_NM)) > 0
THEN ' ' || TRIM (s.ARP_PAR_SUFFIX_NM) END AS name, TRIM (s.ARP_PAR_FIRST_NM) || ' ' || CASE WHEN LENGTH (TRIM
(s.ARP_PAR_MIDDLE_NM)) > 0 THEN TRIM (s.ARP_PAR_MIDDLE_NM) || ' ' END || CASE WHEN LENGTH (TRIM (s.
ARP_PAR_LAST_NM)) > 0 THEN TRIM (s.ARP_PAR_LAST_NM) END || CASE WHEN LENGTH (TRIM (s.ARP_PAR_SUFFIX_NM)) > 0
THEN ' ' || TRIM (s.ARP_PAR_SUFFIX_NM) END || ', ' || s.ARP_PAR_GLID || '@ufl.edu, ' || s.ARP_PAR_PRI_AFF_TYPE
|| ', ' || s.ARP_PAR_PRIMARY_DEPTID || ', ' || s.ARP_PAR_UF_EMAIL_AD AS description from ARP_PERSON_INFO s
WHERE s.ARP_PAR_STATUS_FLG = 'A'))

# the column name to get the subjectId from
subjectApi.source.floridaPerson.param.subjectIdCol.value = id

# the column name to get the name from
subjectApi.source.floridaPerson.param.nameCol.value = name

subjectApi.source.floridaPerson.param.descriptionCol.value = description

# search col where general searches take place, lower case
subjectApi.source.floridaPerson.param.lowerSearchCol.value = search_description

# optional col if you want the search results sorted in the API (note, UI might override)
subjectApi.source.floridaPerson.param.defaultSortCol.value = description

# you can count up from 0 to N of columns to search by identifier (which might also include by id)
subjectApi.source.floridaPerson.param.subjectIdentifierCol0.value = netId

# now you can count up from 0 to N of attributes for various cols.  The name is how to reference in subject.
```

```
getAttribute()
subjectApi.source.floridaPerson.param.subjectAttributeCol0.value = netId

# you can count up from 0 to N of attributes for various cols.  The name is how to reference in subject.
getAttribute()
subjectApi.source.floridaPerson.param.subjectAttributeName0.value = netId

# now you can count up from 0 to N of attributes for various cols.  The name is how to reference in subject.
getAttribute()
subjectApi.source.floridaPerson.param.subjectAttributeCol1.value = email

# you can count up from 0 to N of attributes for various cols.  The name is how to reference in subject.
getAttribute()
subjectApi.source.floridaPerson.param.subjectAttributeName1.value = email

# email attribute name
subjectApi.source.floridaPerson.param.emailAttributeName.value = email

# the 1st sort attribute for lists on screen that are derived from member table (e.g. search for member in
group)
# you can have up to 5 sort attributes
subjectApi.source.floridaPerson.param.sortAttribute0.value = description

# the 1st search attribute for lists on screen that are derived from member table (e.g. search for member in
group)
# you can have up to 5 search attributes
subjectApi.source.floridaPerson.param.searchAttribute0.value = description_lower
```

Then the subject source works

```
SUCCESS: Found subject by id in 22ms: 'mchyzer@ufl.edu'
         with SubjectFinder.findByIdAndSource("mchyzer@ufl.edu", "floridaPerson", false)
SUCCESS: Subject id in returned subject matches the subject id searched for: 'mchyzer@ufl.edu'
SUCCESS: Found subject by identifier in 3ms: 'mchyzer'
         with SubjectFinder.findByIdentifierAndSource("mchyzer", "floridaPerson", false)
SUCCESS: Found 1 subjects by search string in 15ms: 'chr hyz'
         with SubjectFinder.findAll("chr hyz", "floridaPerson")
SUCCESS: Found 1 subjects by paged search string in 8ms: 'chr hyz'
         with SubjectFinder.findPage("chr hyz", "floridaPerson")

######## SUBJECT ATTRIBUTES ########

Subject id: 'mchyzer@ufl.edu' with subject.getId()
  - the subject id should be an unchanging opaque identifier
  - the subject id is stored in the grouper_members table
Subject name: 'Chris Michael Hyzer' with subject.getName()
  - the subject name is generally first last
Subject description: 'Chris Michael Hyzer, mchyzer@ufl.edu, S, ISC, mchyzer@isc.upenn.edu' with subject.
getDescription()
  - the subject description can have more info such as the id, name, dept, etc
Subject type: 'person' with subject.getTypeName()
  - the subject type is not really used
Subject attribute 'email' has 1 value: 'mchyzer@isc.upenn.edu'
  - with subject.getAttributeValue("email")
Subject attribute 'netid' has 1 value: 'mchyzer'
  - with subject.getAttributeValue("netid")
SUCCESS: The emailAttributeName is configured to be: 'email'
SUCCESS: The email address 'mchyzer@isc.upenn.edu' was found and has a valid format

######## SUBJECT IN UI ########

Short link with icon:  Chris Michael Hyzer
  - This is configured in grouper.text.en.us.base.properties with guiSubjectShortLink
  - Also configured in grouper-ui.properties with grouperUi.screenLabel2.sourceId.X
  - By default this is the name of the subject with a tooltip for description
```

```
Long label with icon:   Chris Michael Hyzer, mchyzer@ufl.edu, S, ISC, mchyzer@isc.upenn.edu
  - This is not used in the new UI
  - It is configured in grouper-ui.properties with grouperUi.subjectImg.screenEl.
  - By default this is the description of the subject

######## SUBJECT IN WS ########

Look in grouper-ws.properties to see how the WS uses subjects.  This is the default configuation:

# subject attribute names to send back when a WsSubjectResult is sent, comma separated
# e.g. name, netid
# default is none
ws.subject.result.attribute.names =

# subject result attribute names when extended data is requested (comma separated)
# default is name, description
# note, these will be in addition to ws.subject.result.attribute.names
ws.subject.result.detail.attribute.names =

######## SOURCE CONFIGURATION ########

Adapter class: 'edu.internet2.middleware.subject.provider.JDBCSourceAdapter2'
  - configured in subject.properties: subjectApi.source.floridaPerson.adapterClass
SUCCESS: Found adapter class
SUCCESS: Instantiated adapter class
Source id: 'floridaPerson'
  - configured in subject.properties: subjectApi.source.floridaPerson.id
Source name: 'Florida person'
  - configured in subject.properties: subjectApi.source.floridaPerson.name
Source types: 'person'
  - configured in subject.properties: subjectApi.source.floridaPerson.types
Source param name: 'searchAttribute0' has value: 'description_lower'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.searchAttribute0.value
Source param name: 'dbUser' has value: 'pcdadmin'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.dbUser.value
Source param name: 'subjectAttributeCol1' has value: 'email'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.subjectAttributeCol1.value
Source param name: 'nameCol' has value: 'name'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.nameCol.value
Source param name: 'subjectAttributeCol0' has value: 'netId'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.subjectAttributeCol0.value
Source param name: 'subjectAttributeName1' has value: 'email'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.subjectAttributeName1.value
Source param name: 'subjectAttributeName0' has value: 'netId'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.subjectAttributeName0.value
Source param name: 'subjectIdCol' has value: 'id'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.subjectIdCol.value
Source param name: 'dbUrl' has value: 'jdbc:oracle:thin:@(DESCRIPTION = (ADDRESS = (PROTOCOL = TCP)(HOST =
orrcdbdv2-clstr.seo.int)(PORT = 1521))  (CONNECT_DATA =  (SERVER = DEDICATED)   (SERVICE_NAME = dcom) )  )'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.dbUrl.value
Source param name: 'email' has value: 'mail'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.email.value
Source param name: 'subjectIdentifierCol0' has value: 'netId'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.subjectIdentifierCol0.value
Source param name: 'maxResults' has value: '1000'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.maxResults.value
Source param name: 'maxPageSize' has value: '100'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.maxPageSize.value
Source param name: 'dbPwd' has value: '*******'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.dbPwd.value
Source param name: 'lowerSearchCol' has value: 'search_description'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.lowerSearchCol.value
Source param name: 'emailAttributeName' has value: 'email'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.emailAttributeName.value
Source param name: 'dbTableOrView' has value: '(select id, netId, name, email, description, lower(description)
as search_description from ( select s.ARP_PAR_GLID || '@ufl.edu' AS id, s.ARP_PAR_GLID as netId, s.
ARP_PAR_UF_EMAIL_AD as email, TRIM (s.ARP_PAR_FIRST_NM) || ' ' || CASE WHEN LENGTH (TRIM (s.ARP_PAR_MIDDLE_NM))
> 0 THEN TRIM (s.ARP_PAR_MIDDLE_NM) || ' ' END || CASE WHEN LENGTH (TRIM (s.ARP_PAR_LAST_NM)) > 0 THEN TRIM (s.
ARP_PAR_LAST_NM) END || CASE WHEN LENGTH (TRIM (s.ARP_PAR_SUFFIX_NM)) > 0 THEN ' ' || TRIM (s.
ARP_PAR_SUFFIX_NM) END AS name, TRIM (s.ARP_PAR_FIRST_NM) || ' ' || CASE WHEN LENGTH (TRIM (s.
ARP_PAR_MIDDLE_NM)) > 0 THEN TRIM (s.ARP_PAR_MIDDLE_NM) || ' ' END || CASE WHEN LENGTH (TRIM (s.
```

```
ARP_PAR_LAST_NM)) > 0 THEN TRIM (s.ARP_PAR_LAST_NM) END || CASE WHEN LENGTH (TRIM (s.ARP_PAR_SUFFIX_NM)) > 0
THEN ' ' || TRIM (s.ARP_PAR_SUFFIX_NM) END || ', ' || s.ARP_PAR_GLID || '@ufl.edu, ' || s.ARP_PAR_PRI_AFF_TYPE
|| ', ' || s.ARP_PAR_PRIMARY_DEPTID || ', ' || s.ARP_PAR_UF_EMAIL_AD AS description from ARP_PERSON_INFO s
WHERE s.ARP_PAR_STATUS_FLG = 'A'))'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.dbTableOrView.value
Source param name: 'dbDriver' has value: 'oracle.jdbc.driver.OracleDriver'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.dbDriver.value
Source param name: 'sortAttribute0' has value: 'description'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.sortAttribute0.value
Source param name: 'jdbcConnectionProvider' has value: 'edu.internet2.middleware.subject.provider.
C3p0JdbcConnectionProvider'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.jdbcConnectionProvider.value
Source param name: 'descriptionCol' has value: 'description'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.descriptionCol.value
Source param name: 'defaultSortCol' has value: 'description'
  - configured in subject.properties: subjectApi.source.floridaPerson.param.defaultSortCol.value
No internal attributes configured
No attributes configured

######## SUBJECT SEARCH RESULTS ########

Subject 0: id: mchyzer@ufl.edu, name: Chris Michael Hyzer
  - description: Chris Michael Hyzer, mchyzer@ufl.edu, S, ISC, mchyzer@isc.upenn.edu

######## SUBJECT PAGE RESULTS ########

Subject 0: id: mchyzer@ufl.edu, name: Chris Michael Hyzer
  - description: Chris Michael Hyzer, mchyzer@ufl.edu, S, ISC, mchyzer@isc.upenn.edu
```