REL MID-ATLANTIC
Regional Educational Laboratory
at ICF International

District Data Coordinator Toolbox: Implementing Database Connections in SAS

Jason Schoeneberger, Ph.D.
Senior Researcher & Task Lead

Mid-Atlantic: Delaware, Maryland, New Jersey, Pennsylvania, Washington, D.C.

REL MID-ATLANTIC Regional Educational Laboratory at ICF International

Data, data, everywhere

The volume of and the push to make use educational data is growing:

- More people must become data savvy (teachers, coordinators, etc.)
- Leadership may request cyclical reporting to establish and monitor trends
- Little time to document business rules or standardize data storage practices
- Quality control can take time or be difficult to manage

Teachers, principals, administrators and analysts often have difficulty keeping pace.

REL MID-ATLANTIC Regional Educational Laboratory at ICF International

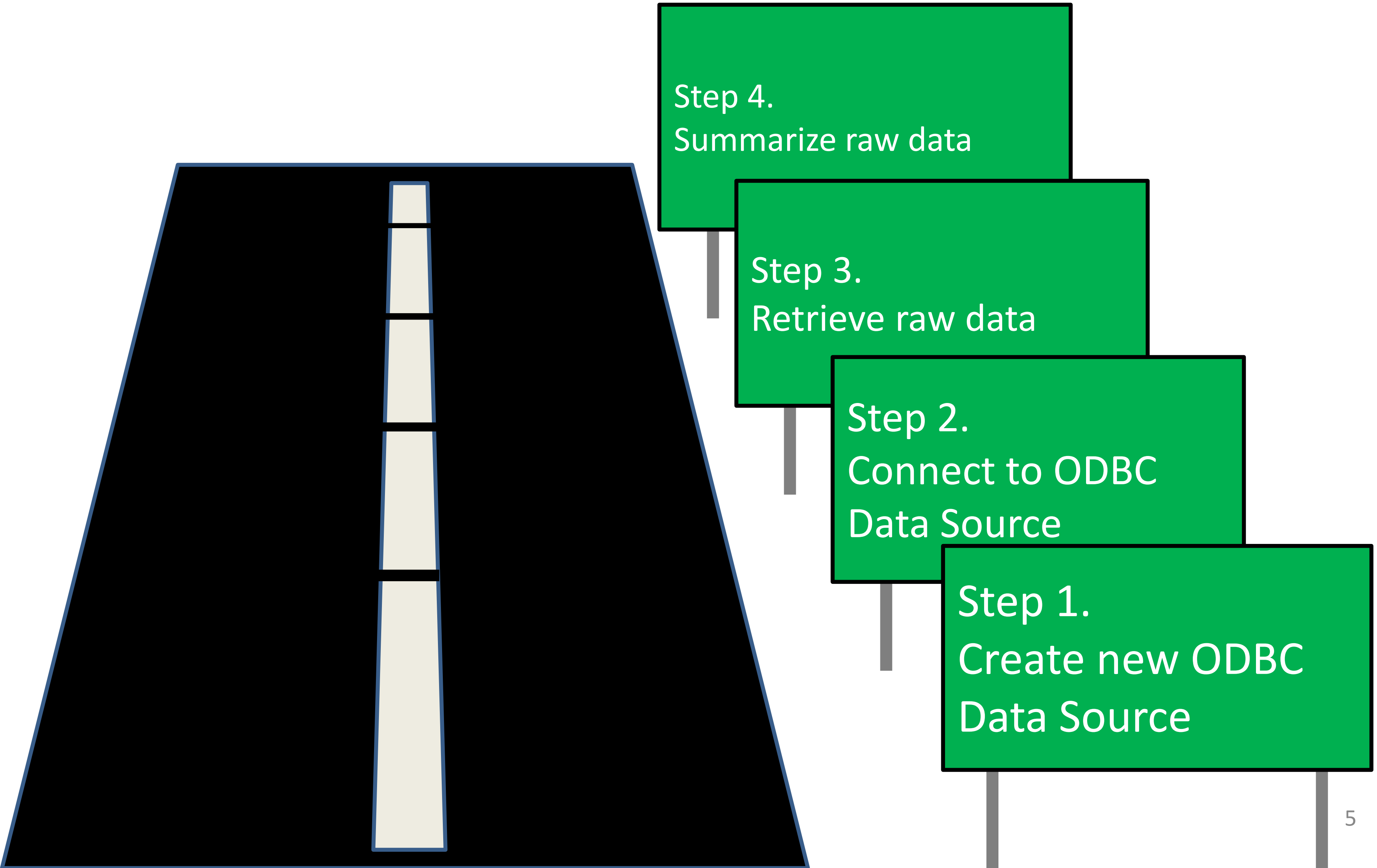
Some familiar scenarios (using data stored in SQL, Oracle, Access, etc.)

- The same data points are necessary across reporting cycles
- Process to acquire and report data is repetitive across reporting cycles
- A non-technical person may be tasked with reporting responsibility
- Lack of documentation
- Analysts report shortage of storage space on network or external hard drives
- Analysts are maintaining idiosyncratic versions of various data elements (e.g. test score files, student attendance files, etc.)
- Idiosyncratic versions have commonalities across analyst versions
- Separate data requests completed by different analysts yield conflicting results (e.g. a school mean test score)

Database connections

- Databases (e.g., SQL, Oracle, Access, etc.) allow for basic data base connectivity:
  - Open Database Connectivity (ODBC)
  - Object Linking and Embedding Database (OLEDB)
  - These are often standard on computers
- ODBC/OLEDB connections are frameworks to allow data manipulation software (e.g. Excel, SPSS, SAS) to communicate with databases

Road map to data connectivity



Traveling the road by example

- To follow the steps in our road map to connectivity, let’s assume the following example:
  - District leadership has asked us to examine reading achievement as measured by reading assessment achievement levels
  - Leadership is specifically interested in 6th grade student performance
  - They want to examine performance by student Limited English Proficient (LEP) status.
  - The data we need to obtain are stored in an Access database

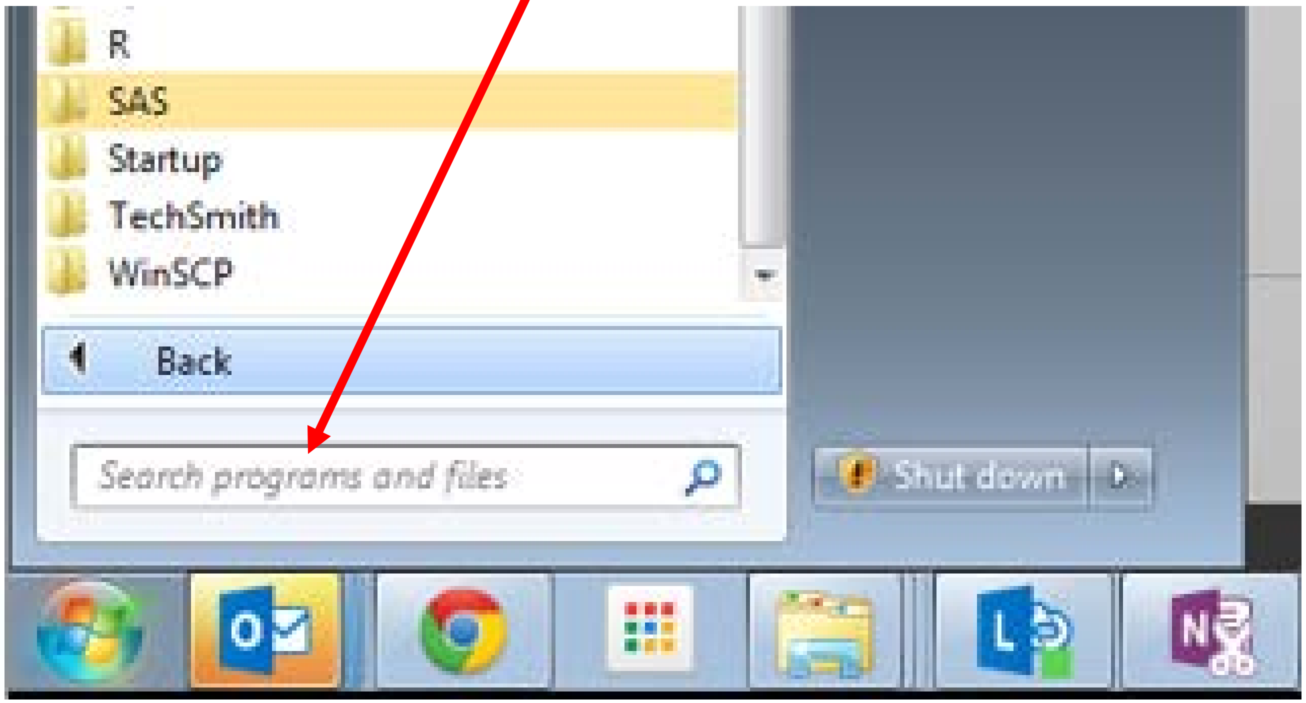
Creating an ODBC data source

- The first step is to create an ODBC Data Source centered on an existing database such as Access, SQL, or Oracle. ODBC Data Sources are frameworks, or linkages for software packages such as SAS to communicate with databases

REL MID-ATLANTIC
Regional Educational Laboratory
at ICF International

Open ODBC administrator window

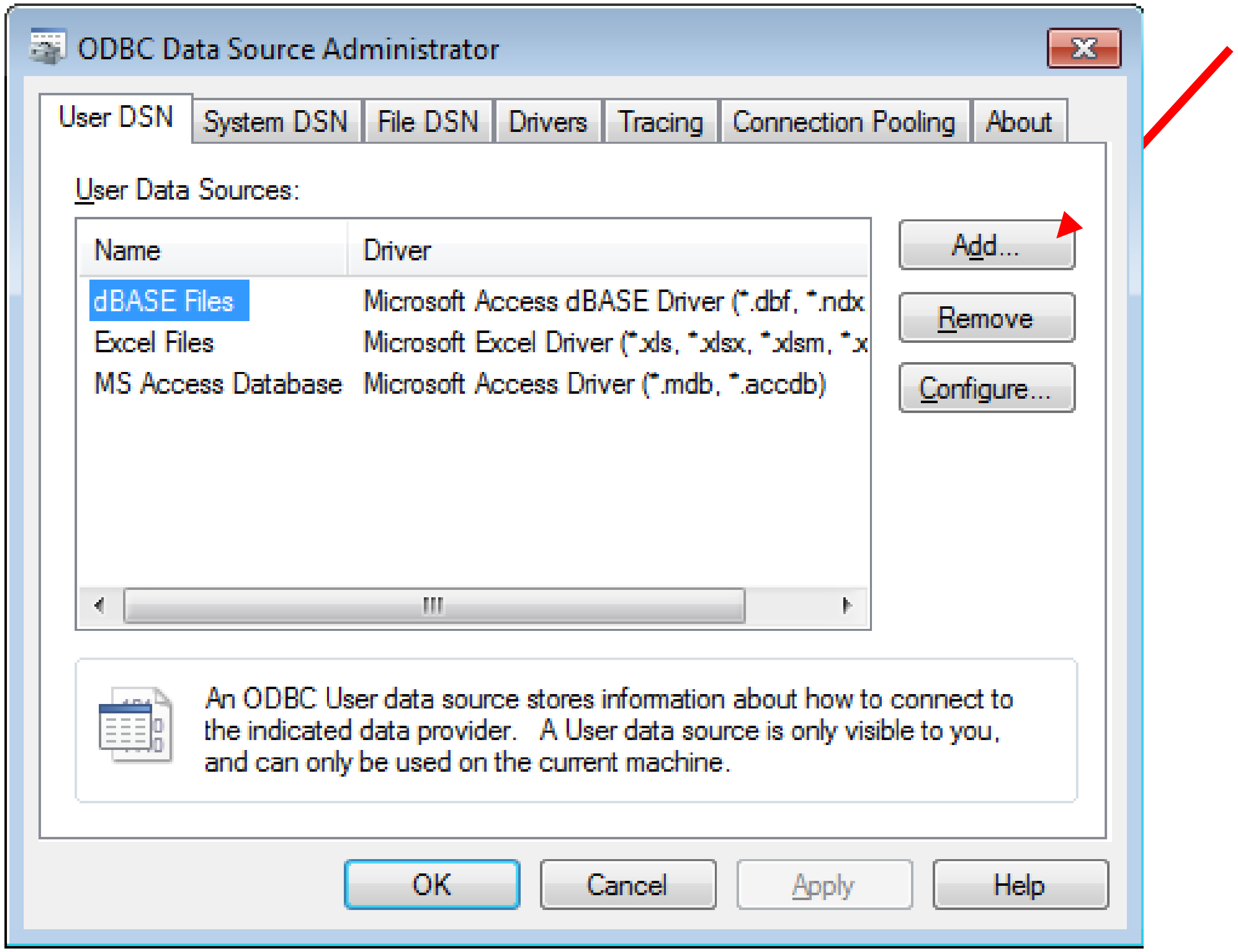
- Type ‘ODBC’ in Search Box and press Enter



REL MID-ATLANTIC
Regional Educational Laboratory
at ICF International

Add a new data source

- Click ‘Add’ to begin adding a new ODBC data source



ODBC Data Source Administrator

User DSN | System DSN | File DSN | Drivers | Tracing | Connection Pooling | About

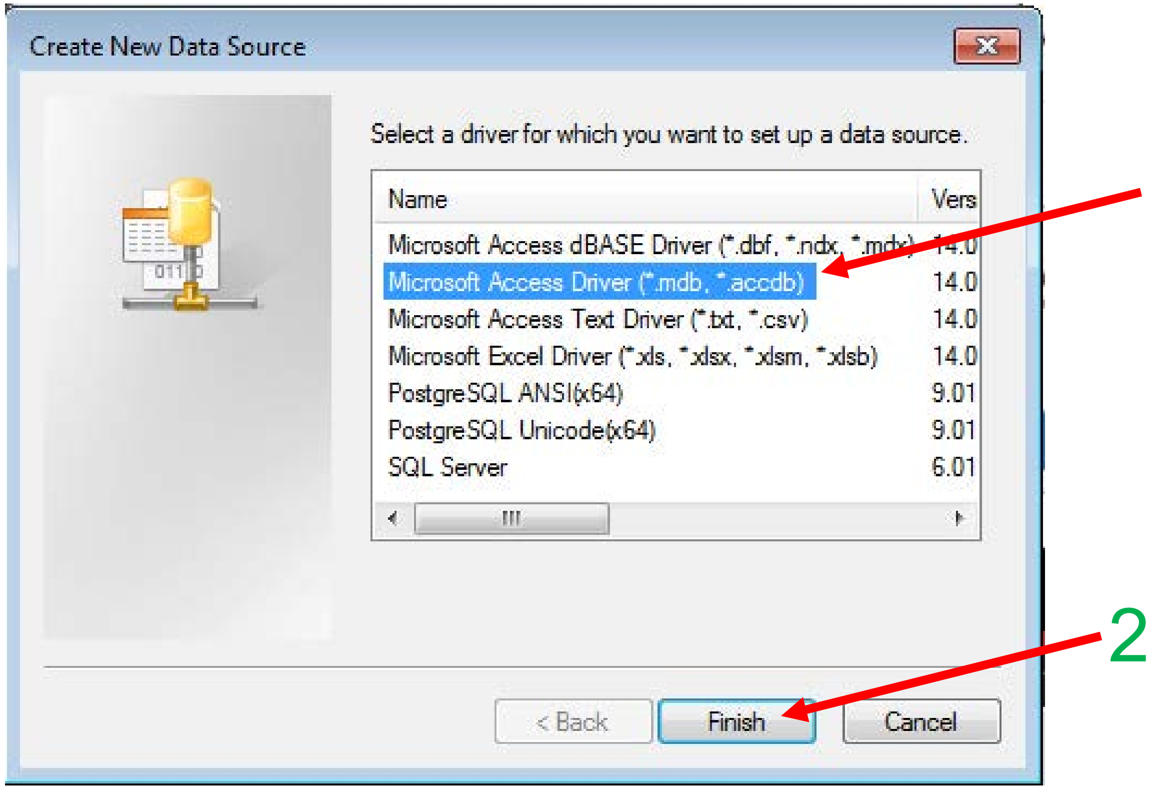
User Data Sources:

Name	Driver
dBASE Files	Microsoft Access dBASE Driver (*.dbf, *.ndx
Excel Files	Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.x
MS Access Database	Microsoft Access Driver (*.mdb, *.accdb)

An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

Choose a driver for the data source

1. Choose driver for connection to a source (in this example, we connect to an Access database)
2. Click Finish



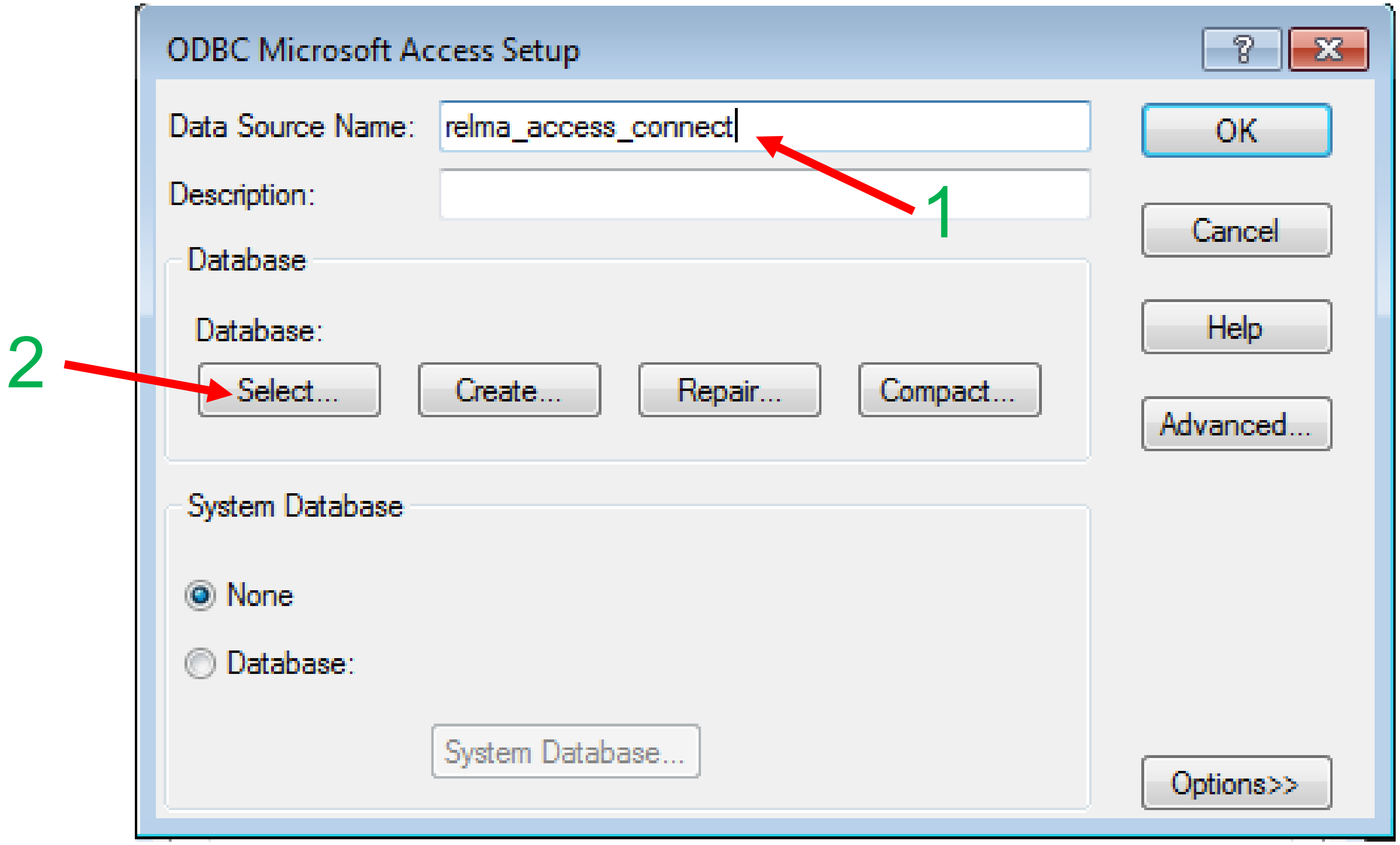
Create New Data Source

Select a driver for which you want to set up a data source.

Name	Vers
Microsoft Access dBASE Driver (*.dbf, *.ndx, *.mdx)	14.0
Microsoft Access Driver (*.mdb, *.accdb)	14.0
Microsoft Access Text Driver (*.txt, *.csv)	14.0
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb)	14.0
PostgreSQL ANSI(x64)	9.01
PostgreSQL Unicode(x64)	9.01
SQL Server	6.01

Name the database connection

1. Name the connection to the database
2. Click ‘Select’ button under Database



ODBC Microsoft Access Setup

Data Source Name: relma_access_connect

Description:

Database

Database:

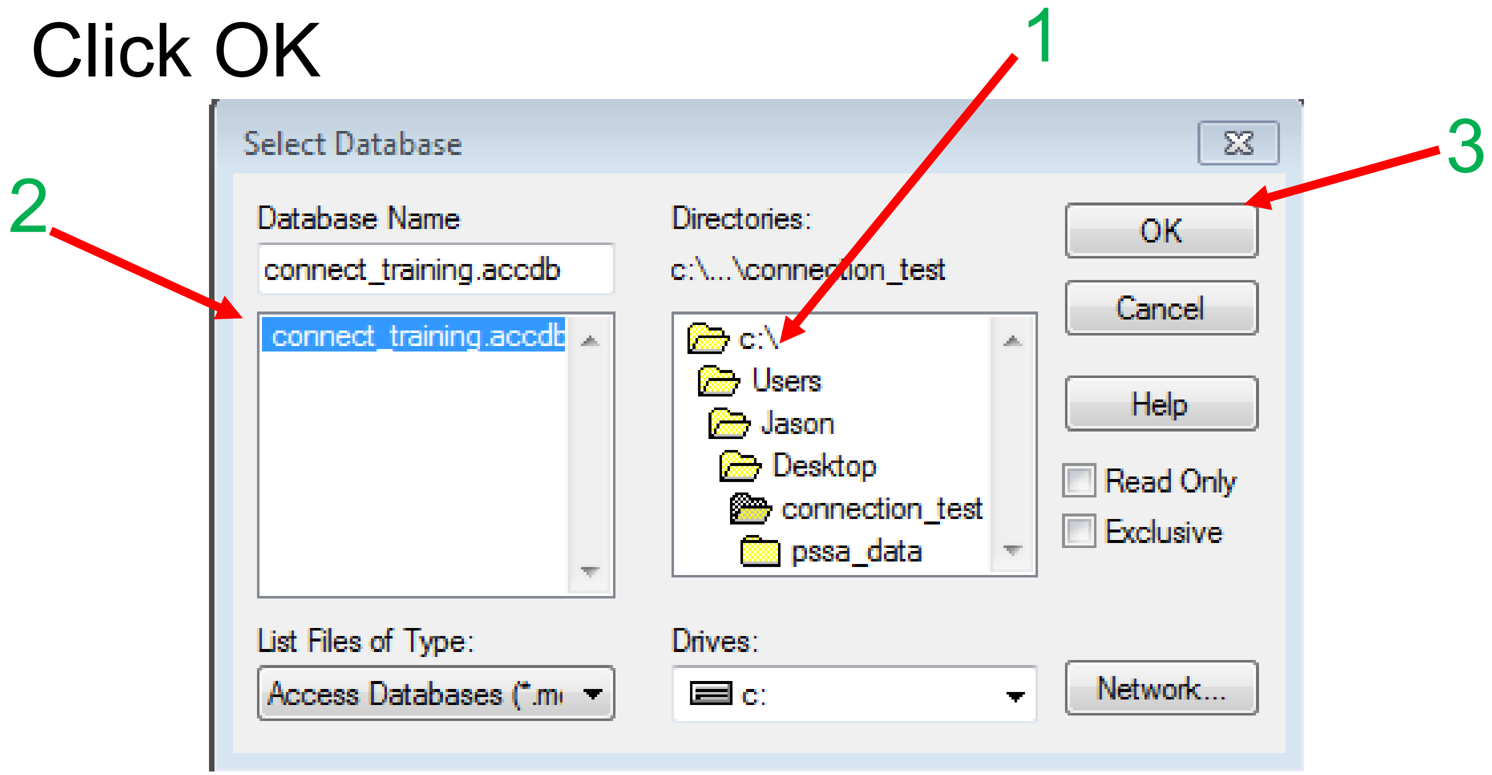
System Database

(● None) Database:

REL MID-ATLANTIC Regional Educational Laboratory at ICF International

Select source database

1. Navigate to location of the database (the Access database we want to connect to in this example)
2. Select source database
3. Click OK


Select Database

Database Name: connect_training.accdb

connect_training.accdb

Directories: c:\...\connection_test

c:\
Users
Jason
Desktop
connection_test
pssa_data

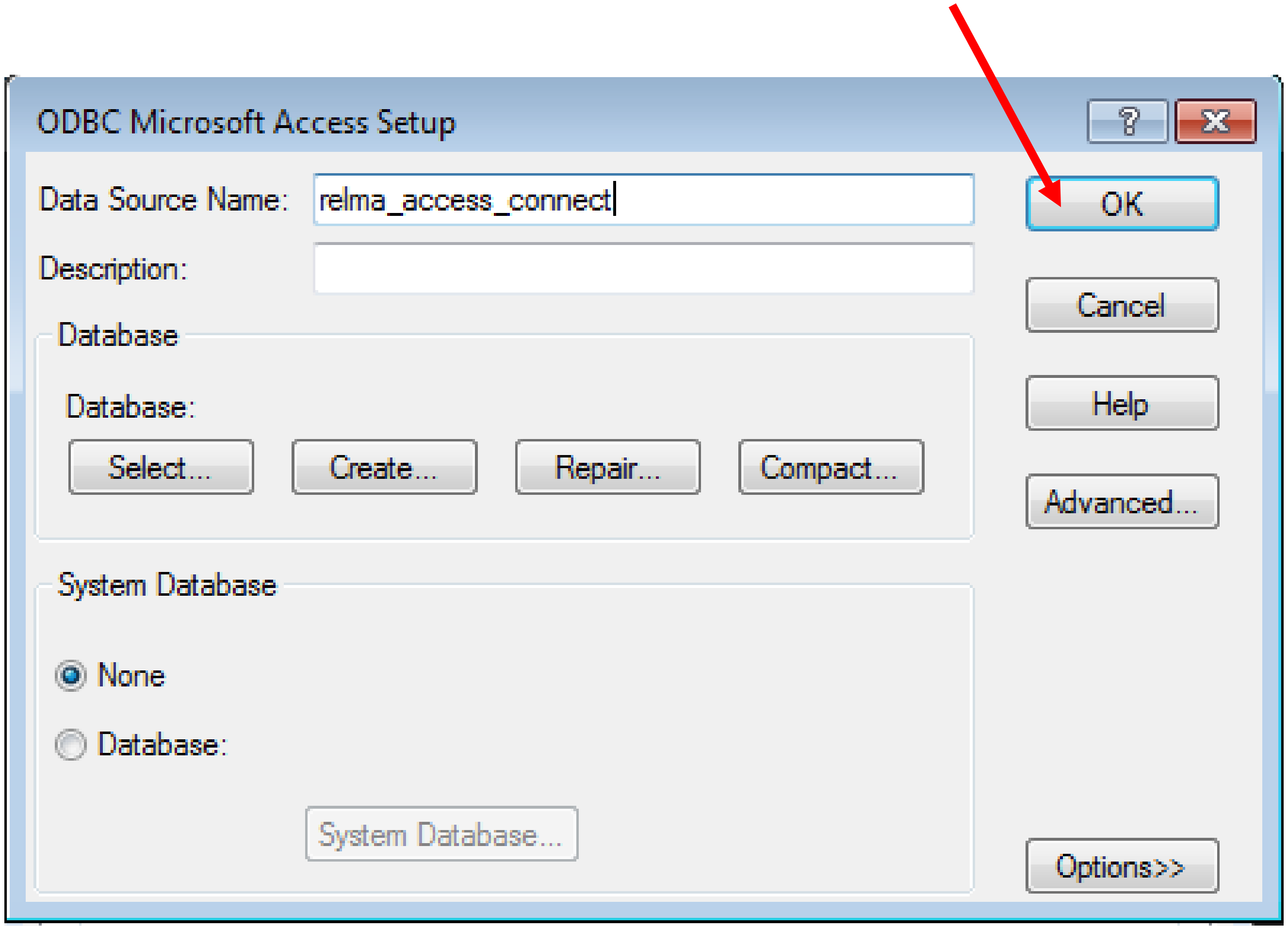
List Files of Type: Access Databases (*.mı

Drives: c:

REL MID-ATLANTIC
Regional Educational Laboratory
at ICF International

Click OK for data source name

- Click OK button under Database



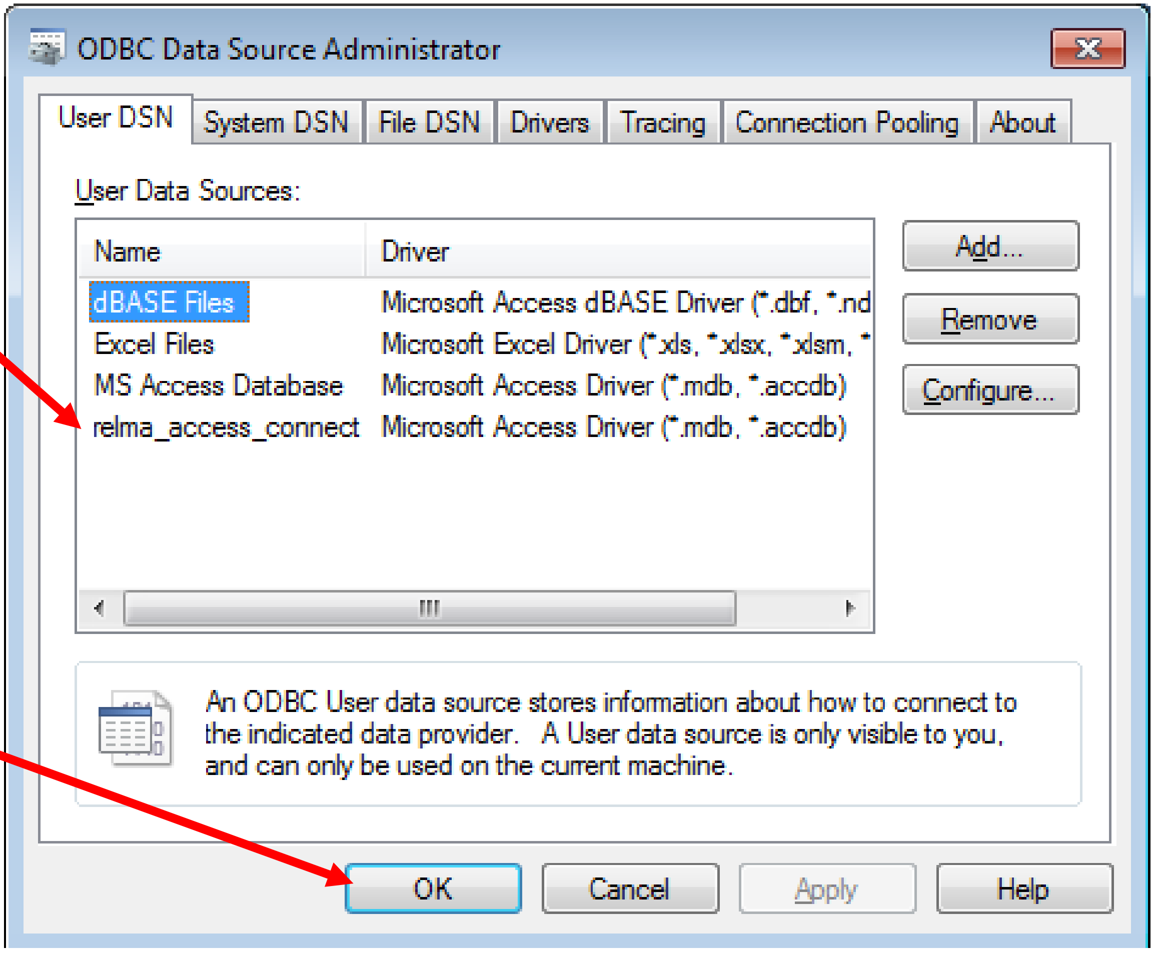
ODBC Microsoft Access Setup

Data Source Name: relma_access_connect

Description:

New data source appears in ODBC directory

1. New data source listed in ODBC directory
2. Click OK



User Data Sources:

Name	Driver
dBASE Files	Microsoft Access dBASE Driver (*.dbf, *.nd
Excel Files	Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *
MS Access Database	Microsoft Access Driver (*.mdb, *.accdb)
relma_access_connect	Microsoft Access Driver (*.mdb, *.accdb)

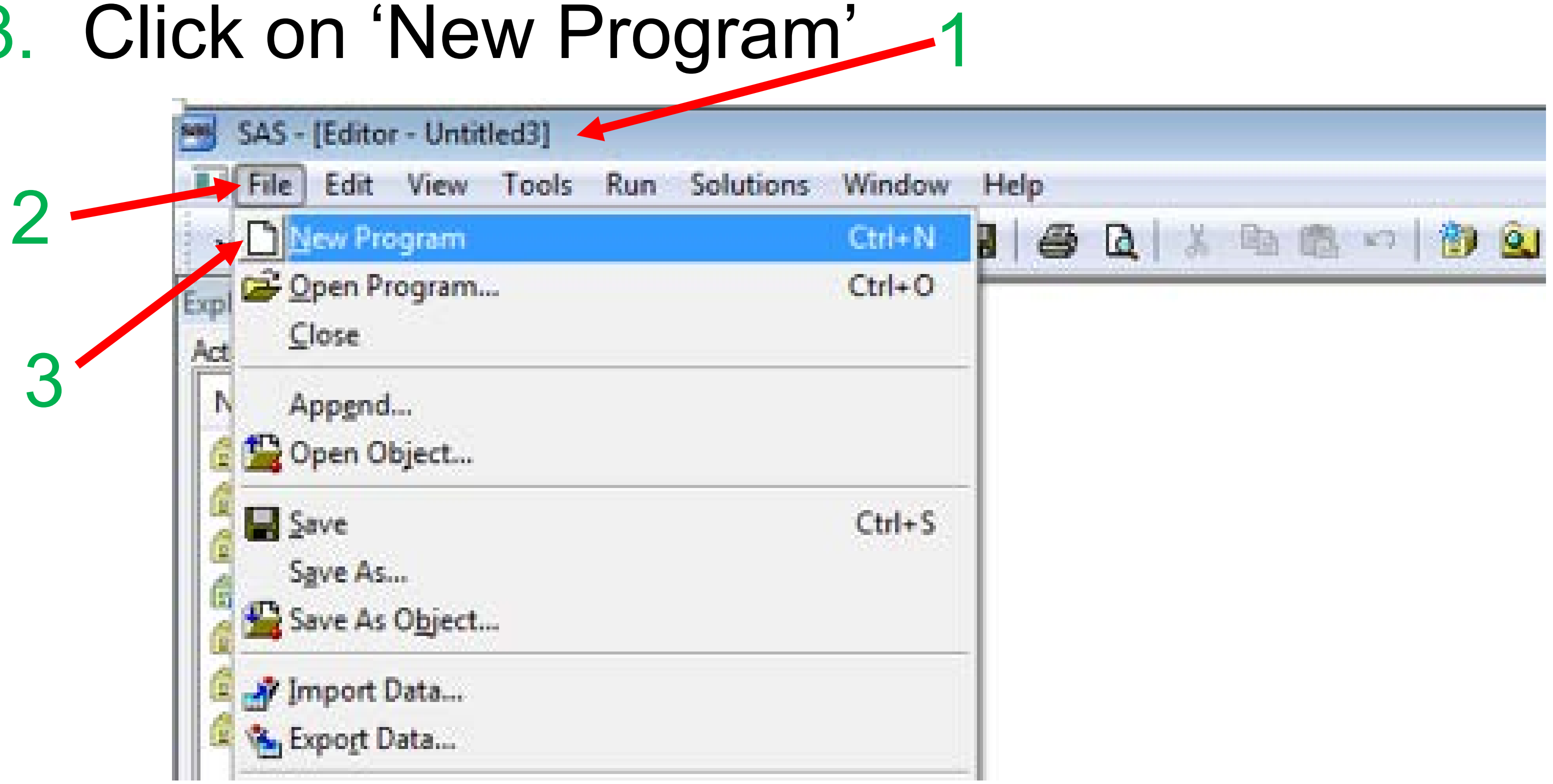
An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you, and can only be used on the current machine.

Connecting to a Database

- Now that our ODBC data source exists for communicating with the database, the information in the database can be extracted directly into other software packages (e.g. SAS) for further manipulation

Connect to database using SAS

1. Open an instance of SAS (screenshots use SAS 9.4)
2. Navigate to the File menu in SAS
3. Click on ‘New Program’



Define SAS library name (libname)

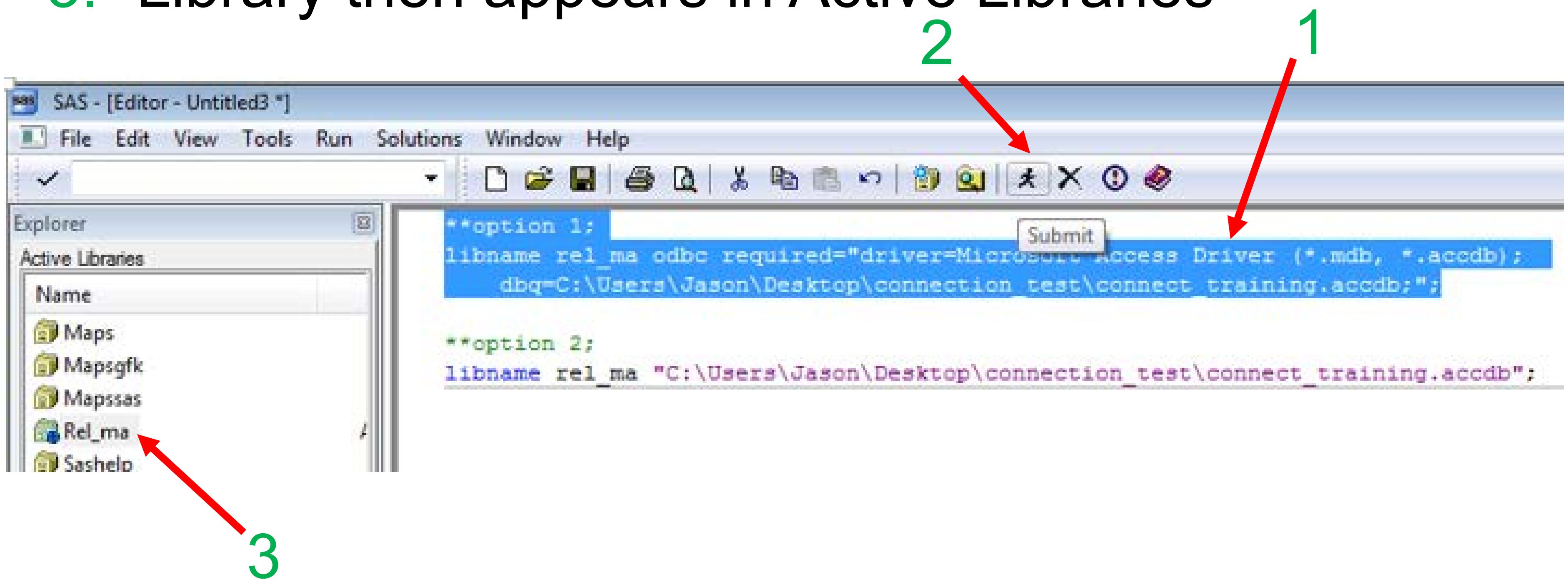
1. Copy and paste one of the options below into the SAS program editor file
   a) Option 2 relevant if using the latest (9.4) version of SAS
   b) This is the name you can assign to your library (libname)

```
**option 1;
libname rel_ma odbc required="driver=Microsoft Access Driver
(*.mdb, *.accdb);
       dbq=C:\Users\Jason\Desktop\connection_test\connect_traini
ng.accdb;";

**option 2;
libname rel_ma
"C:\Users\Jason\Desktop\connection_test\connect_training.accdb";
```

b

Execute libname syntax

1. Highlight an option
2. Click Submit (SAS Running Man)
3. Library then appears in Active Libraries



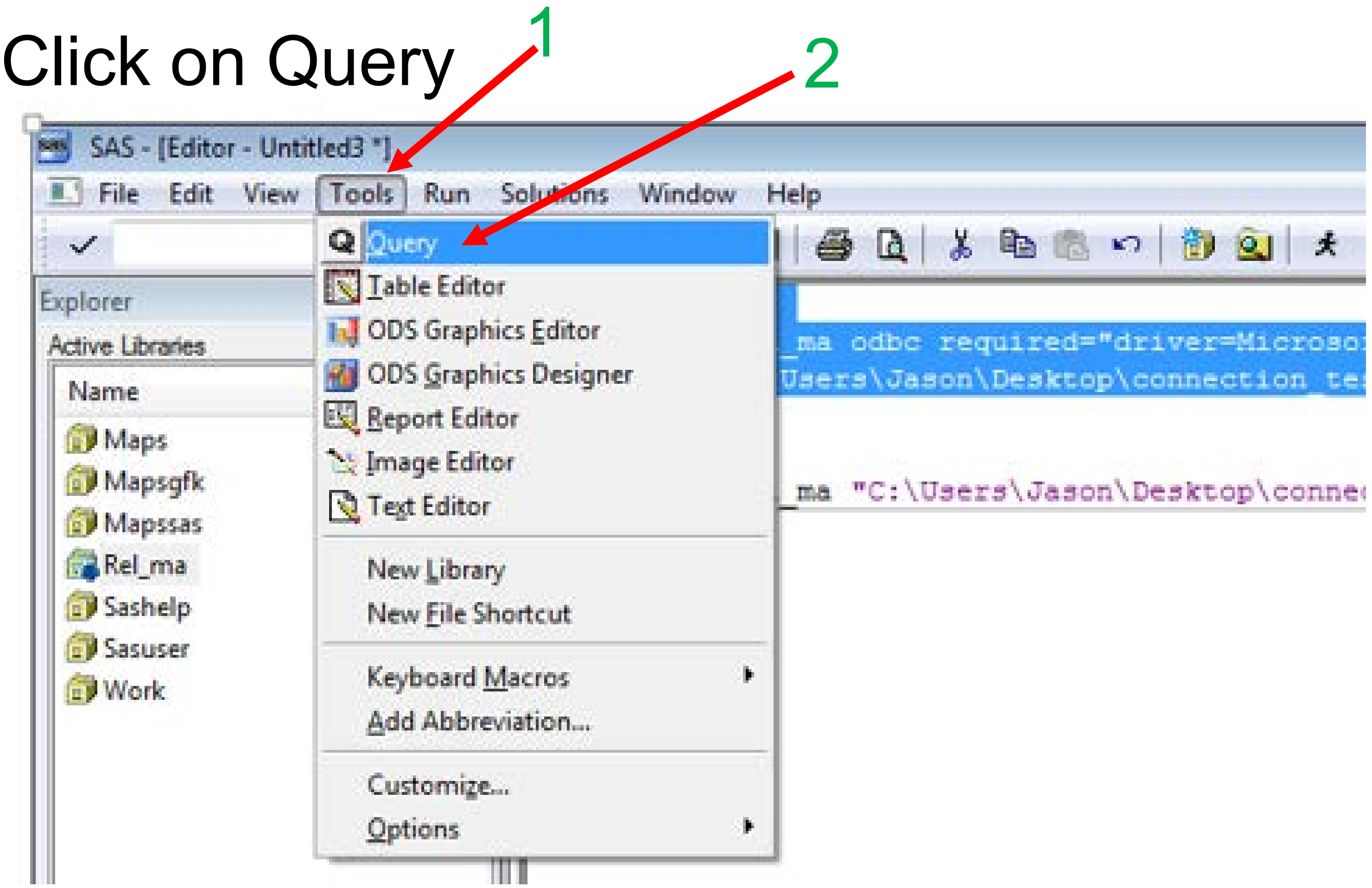
2
1
3
SAS - [Editor - Untitled3 *]
File Edit View Tools Run Solutions Window Help
Submit
Explorer
Active Libraries
Name
Maps
Mapsgfk
Mapssas
Rel_ma
Sashelp

```
**option 1;
libname rel_ma odbc required="driver=Microsoft Access Driver (*.mdb, *.accdb);
    dbq=C:\Users\Jason\Desktop\connection_test\connect_training.accdb;";

**option 2;
libname rel_ma "C:\Users\Jason\Desktop\connection_test\connect_training.accdb";
```

REL MID-ATLANTIC
Regional Educational Laboratory at ICF International

Open SAS query tool

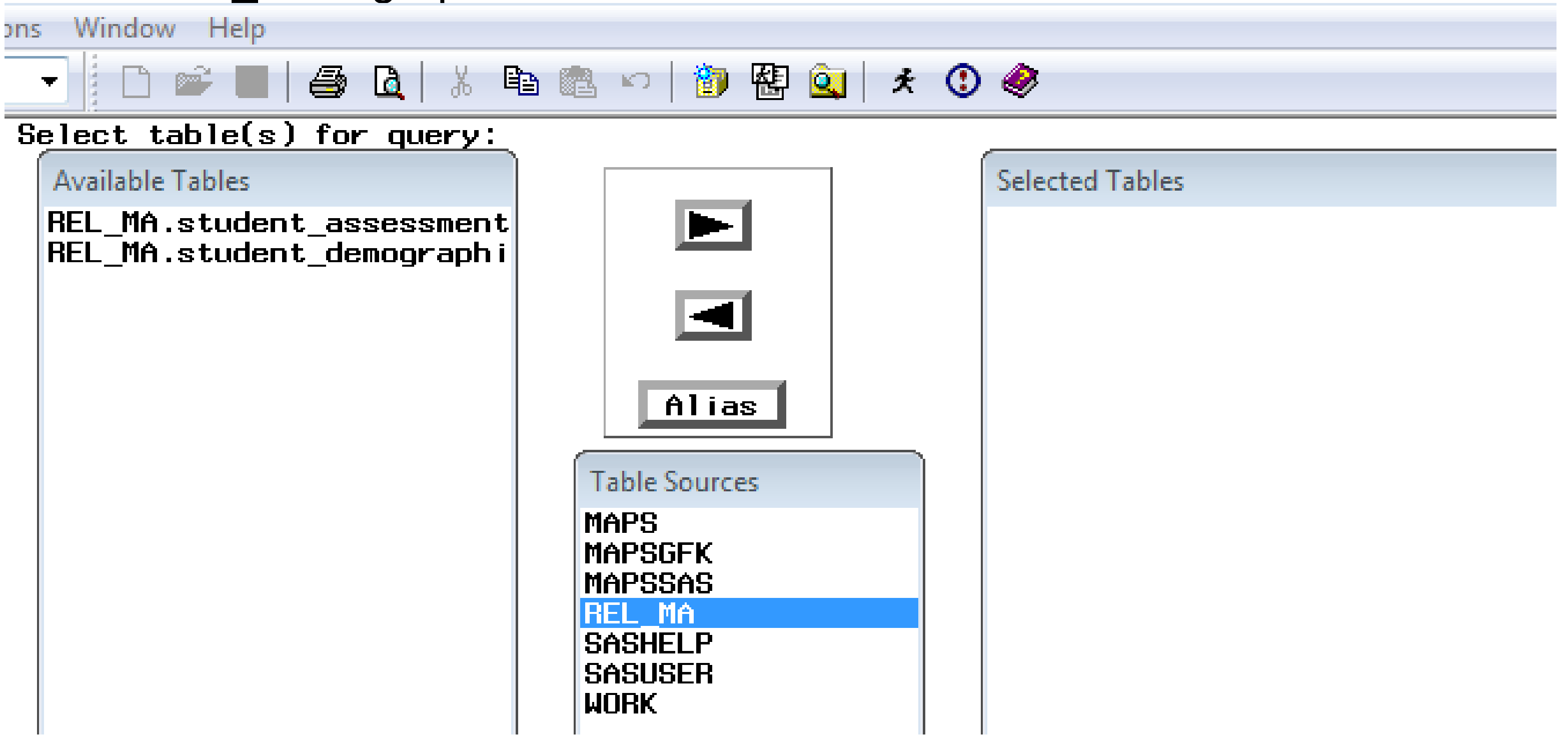
1. Click Tools
2. Click on Query



```
_ma odbc required="driver=Microso
Users\Jason\Desktop\connection te
_ma "C:\Users\Jason\Desktop\conne
```

View available tables in data source

1. Ensure the libname you created is selected under Table Sources
2. Now we can view data tables in the database
   - student_assessments
   - student_demographics



Select table(s) for query:

Available Tables

REL_MA.student_assessment
REL_MA.student_demographi

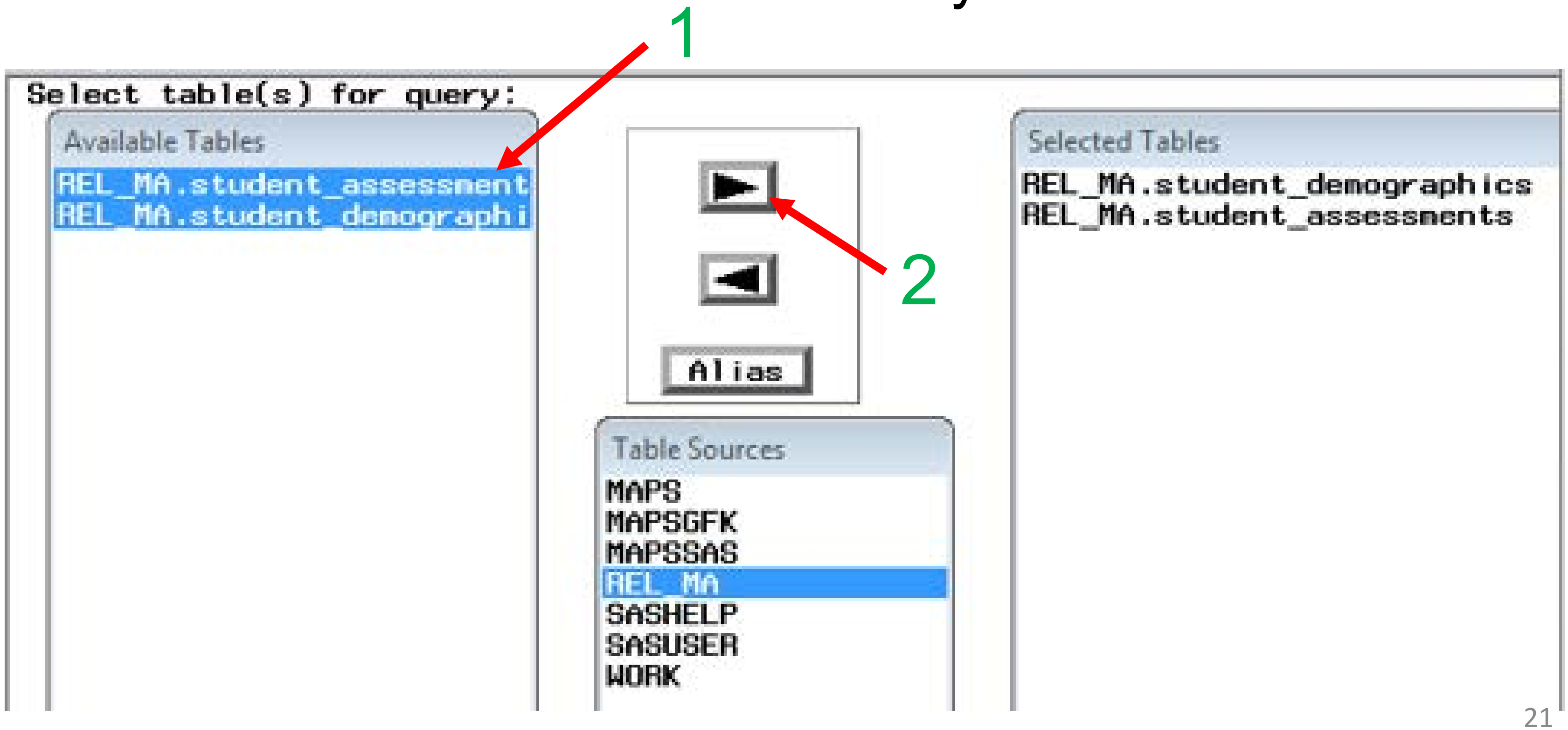
Selected Tables

Table Sources

MAPS
MAPSGFK
MAPSSAS
REL_MA
SASHELP
SASUSER
WORK

Select tables to query

1. Select each table you want to query
   - student_assessments
   - student_demographics
2. Click Right Arrow to move selections to Selected Tables
3. Click OK in lower left of SAS Query screen


1
2
Select table(s) for query:
Available Tables
REL_MA.student_assessment
REL_MA.student_demographi
Alias
Table Sources
MAPS
MAPSGFK
MAPSSAS
REL_MA
SASHELP
SASUSER
WORK
Selected Tables
REL_MA.student_demographics
REL_MA.student_assessments

Select variables

1. Click on variables to retrieve from each table
2. Click ➡ arrow to move variables to Selected Columns

From student_demographics

a) Student_id (student identification number)

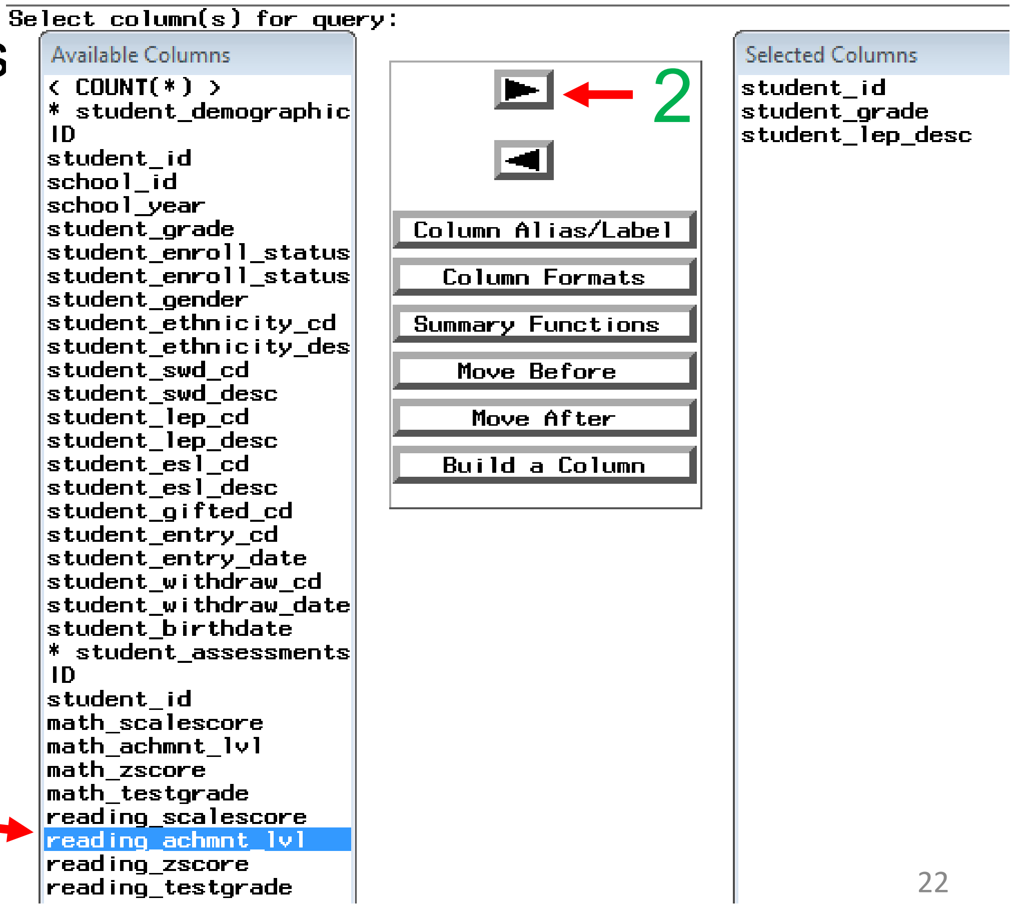
b) Student_grade (student grade level)

c) Student_lep_desc (student LEP status description)

From student_assessments

a) reading_achmnt_lvl (student reading proficiency)

1



Select column(s) for query:

Available Columns

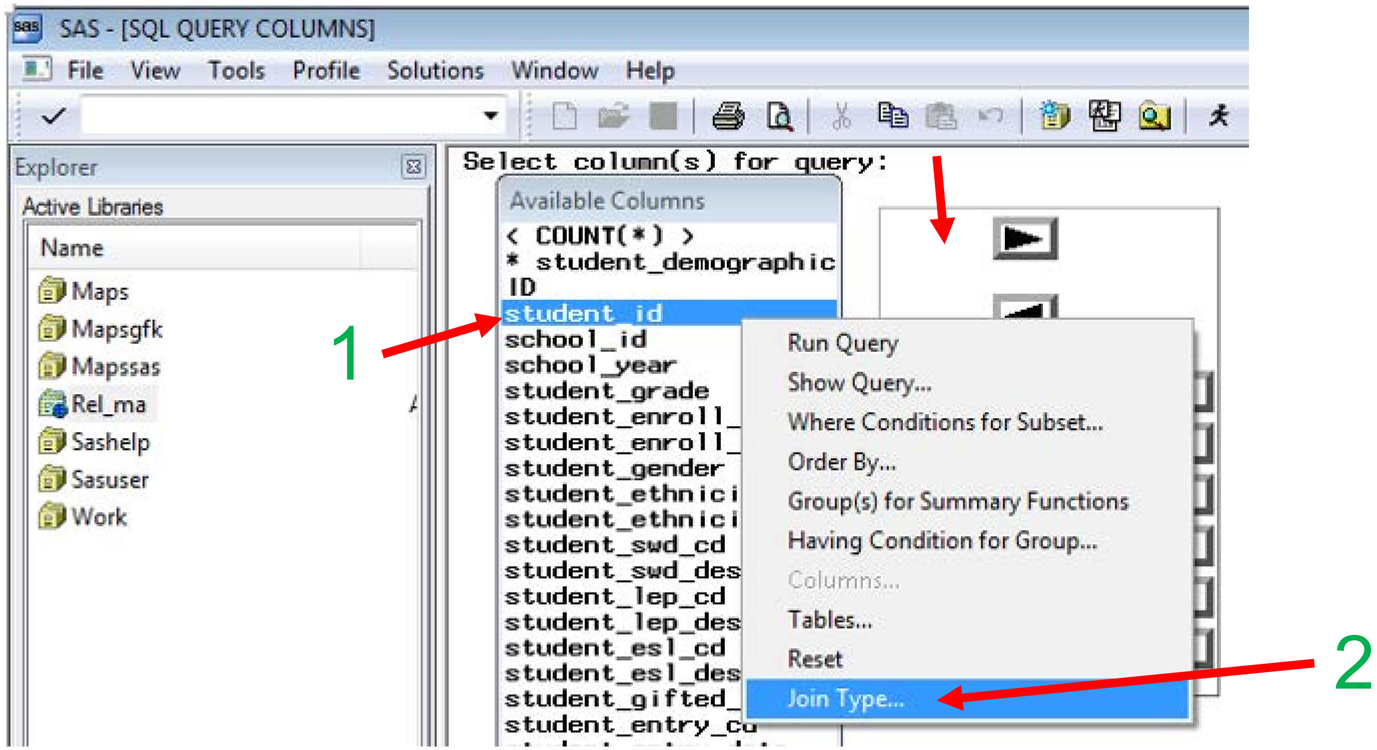
```
< COUNT(*) >
* student_demographic
ID
student_id
school_id
school_year
student_grade
student_enroll_status
student_enroll_status
student_gender
student_ethnicity_cd
student_ethnicity_des
student_swd_cd
student_swd_desc
student_lep_cd
student_lep_desc
student_esl_cd
student_esl_desc
student_gifted_cd
student_entry_cd
student_entry_date
student_withdraw_cd
student_withdraw_date
student_birthdate
* student_assessments
ID
student_id
math_scalescore
math_achmnt_lvl
math_zscore
math_testgrade
reading_scalescore
reading_achmnt_lvl
reading_zscore
reading_testgrade
```

Selected Columns

```
student_id
student_grade
student_lep_desc
```

Adding/deleting relationships

1. Right-click in Available Columns box
2. Select Join Type…in the menu that appears

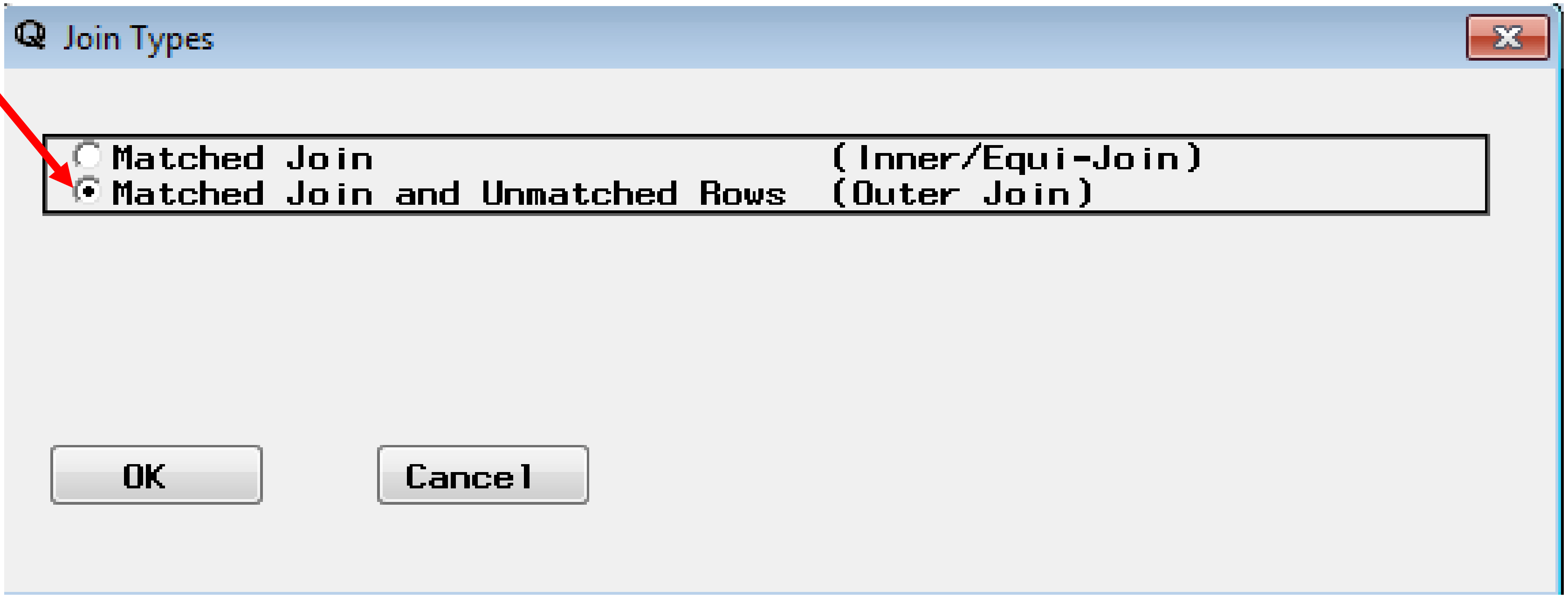

SAS - [SQL QUERY COLUMNS]
File View Tools Profile Solutions Window Help
Explorer
Active Libraries
Name
Maps
Mapsgfk
Mapssas
Rel_ma
Sashelp
Sasuser
Work
1
2
Run Query
Show Query...
Where Conditions for Subset...
Order By...
Group(s) for Summary Functions
Having Condition for Group...
Columns...
Tables...
Reset
Join Type...

```
Select column(s) for query:
Available Columns
< COUNT(*) >
* student_demographic
ID
student_id
school_id
school_year
student_grade
student_enroll_
student_enroll_
student_gender
student_ethnici
student_ethnici
student_swd_cd
student_swd_des
student_lep_cd
student_lep_des
student_esl_cd
student_esl_des
student_gifted_
student_entry_cd
```

Join types

1. Change type of join using radio button
   a) In this instance, we want Matched Join and Unmatched Rows

1a



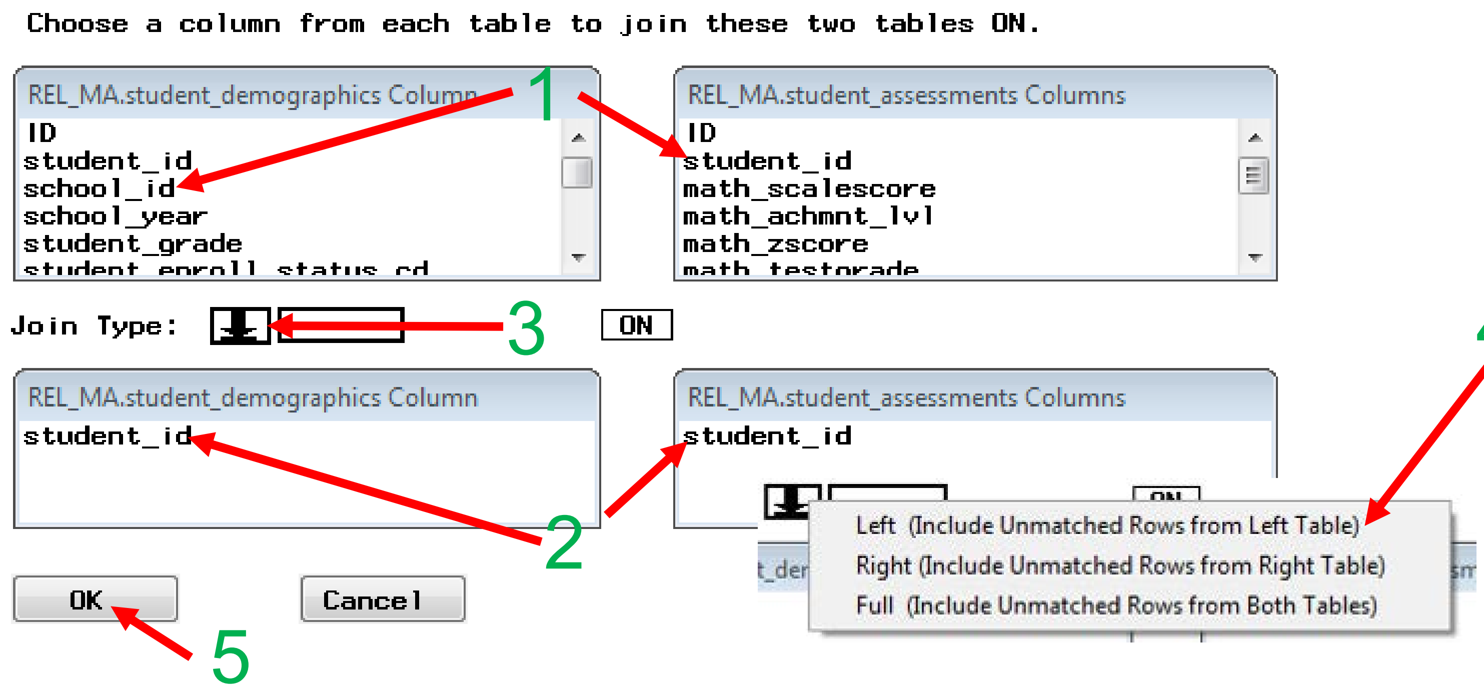
Join Types

Matched Join (Inner/Equi-Join)
[✓ Matched Join and Unmatched Rows (Outer Join)]

REL MID-ATLANTIC Regional Educational Laboratory at ICF International

Specify variable and join type

1. Highlight variable(s) to join on in each column (student_id)
2. Each will appear in box below their respective table
3. Click the Down-Arrow Icon
4. Click to choose Join type (in this case, Left Join)
5. Click OK



Choose a column from each table to join these two tables ON.

REL_MA.student_demographics Column
```
ID
student_id
school_id
school_year
student_grade
student_enroll_status_cd
```

REL_MA.student_assessments Columns
```
ID
student_id
math_scalescore
math_achmnt_lvl
math_zscore
math_testgrade
```

Join Type: ON

REL_MA.student_demographics Column
```
student_id
```

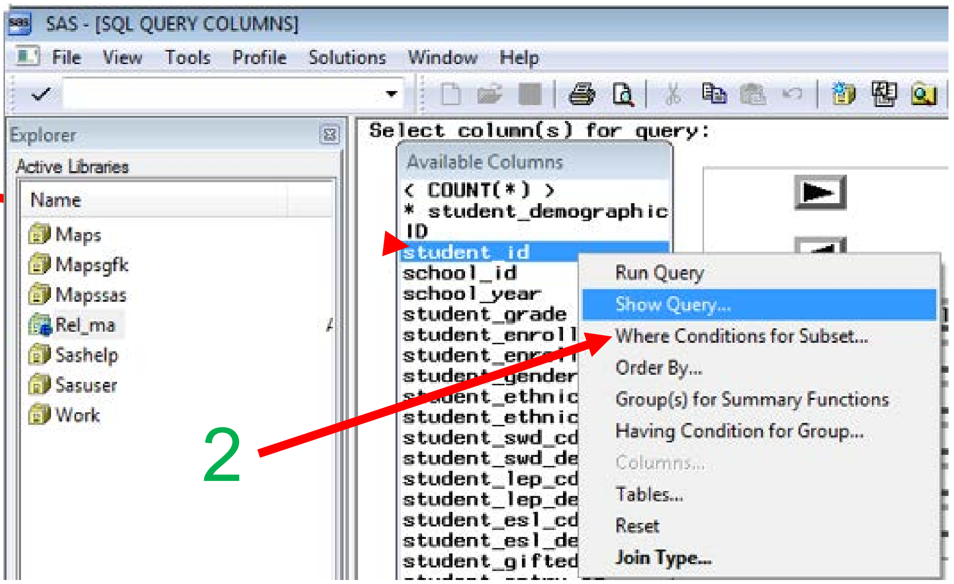
REL_MA.student_assessments Columns
```
student_id
```

Left (Include Unmatched Rows from Left Table)
Right (Include Unmatched Rows from Right Table)
Full (Include Unmatched Rows from Both Tables)

OK Cancel

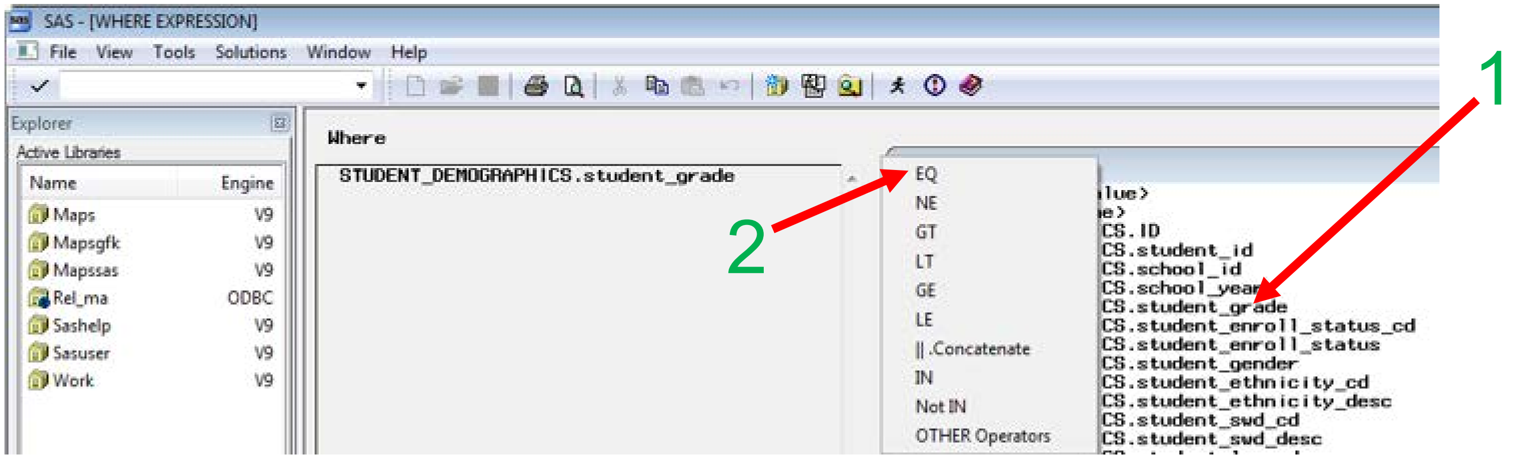
Restrict data with 'where' statement

1. Right click in the Available Columns area
2. Select Where Conditions for Subset…



Restrict data with 'where' statement

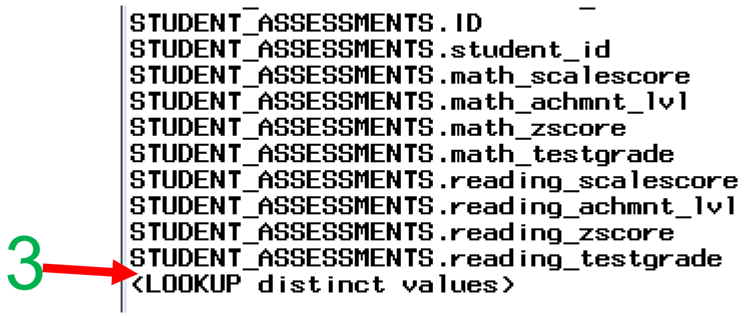
1. Click on the student_grade variable
2. Select 'EQ' from the dropdown menu



```
STUDENT_DEMOGRAPHICS.student_grade
```

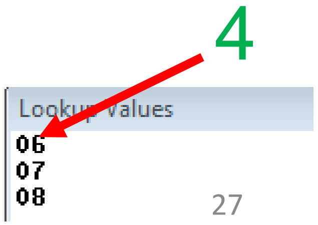
```
lue>
e>
CS.ID
CS.student_id
CS.school_id
CS.school_year
CS.student_grade
CS.student_enroll_status_cd
CS.student_enroll_status
CS.student_gender
CS.student_ethnicity_cd
CS.student_ethnicity_desc
CS.student_swd_cd
CS.student_swd_desc
```

3. Click on <LOOKUP distinct values> that appears



```
STUDENT_ASSESSMENTS.ID
STUDENT_ASSESSMENTS.student_id
STUDENT_ASSESSMENTS.math_scalescore
STUDENT_ASSESSMENTS.math_achmnt_lvl
STUDENT_ASSESSMENTS.math_zscore
STUDENT_ASSESSMENTS.math_testgrade
STUDENT_ASSESSMENTS.reading_scalescore
STUDENT_ASSESSMENTS.reading_achmnt_lvl
STUDENT_ASSESSMENTS.reading_zscore
STUDENT_ASSESSMENTS.reading_testgrade
<LOOKUP distinct values>
```

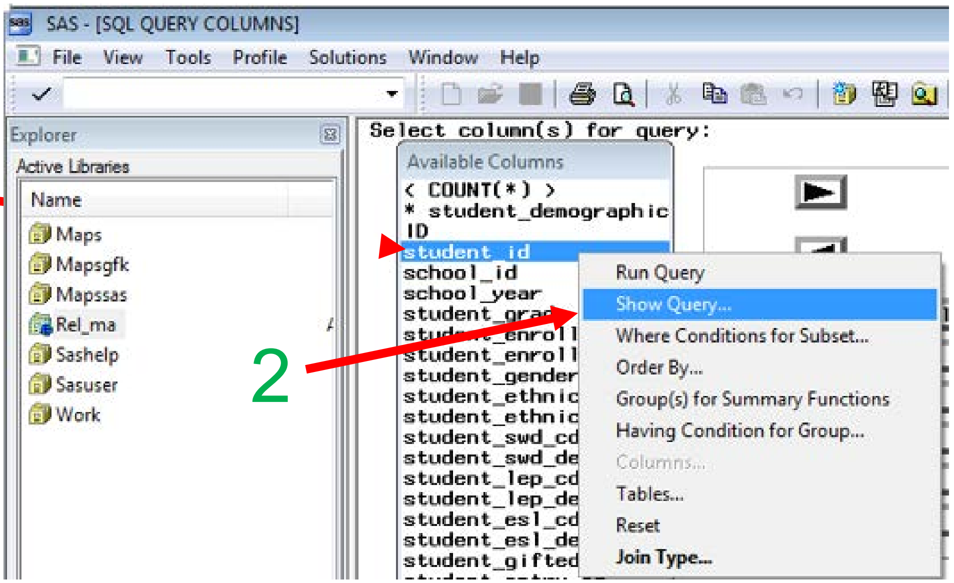
4. Select '06' to retrieve 6th graders
5. Click OK at bottom
6. Click OK at bottom

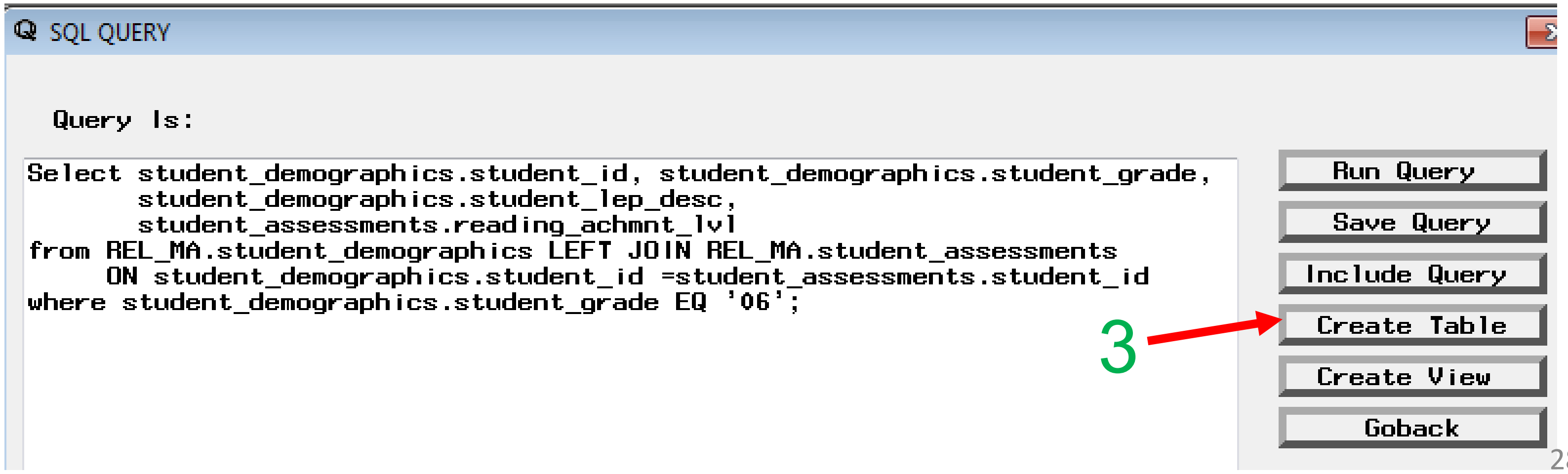


```
06
07
08
```

View SQL query

1. Right click in the Available Columns area
2. Select Show Query from drop-down menu
3. Click Create Table



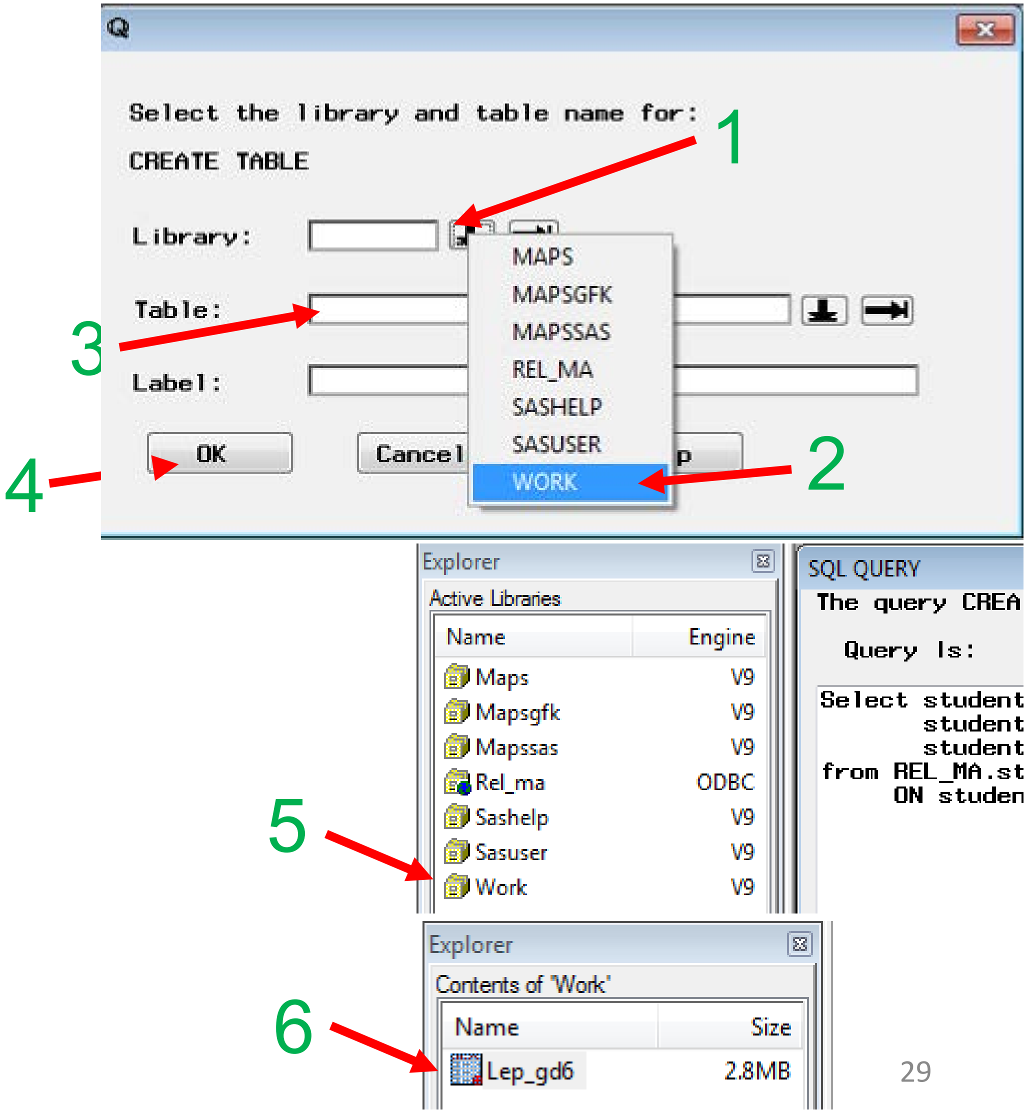


Query Is:

```
Select student_demographics.student_id, student_demographics.student_grade,
       student_demographics.student_lep_desc,
       student_assessments.reading_achmnt_lvl
from REL_MA.student_demographics LEFT JOIN REL_MA.student_assessments
     ON student_demographics.student_id =student_assessments.student_id
where student_demographics.student_grade EQ '06';
```

Create a data table

1. In Create table pop-up, click down arrow button
2. Select 'WORK' as table destination
3. Click inside Table box, enter name for returned data table ('lep_gd6')
4. Click OK
5. Double-Click Work library folder
6. SAS data file should appear in Work library



Name	Engine
Maps	V9
Mapsgfk	V9
Mapssas	V9
Rel_ma	ODBC
Sashelp	V9
Sasuser	V9
Work	V9

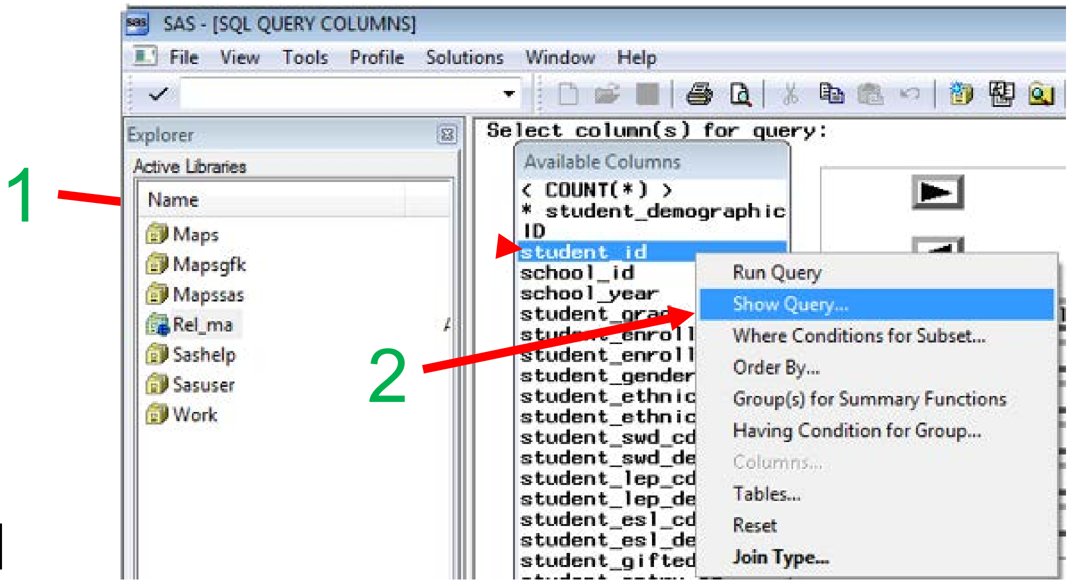
```
The query CREA
  Query Is:
Select student
       student
       student
from REL_MA.st
   ON studen
```

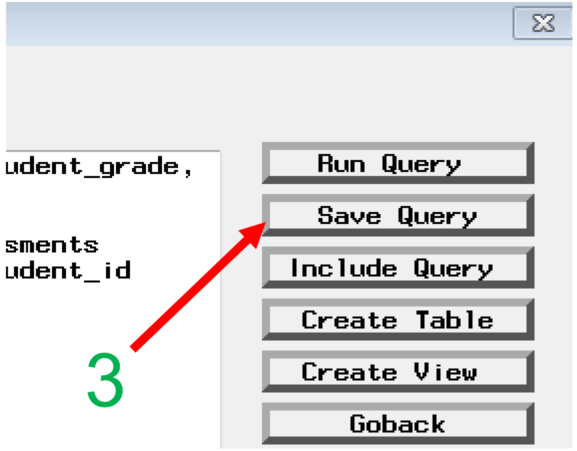
Name	Size
Lep_gd6	2.8MB

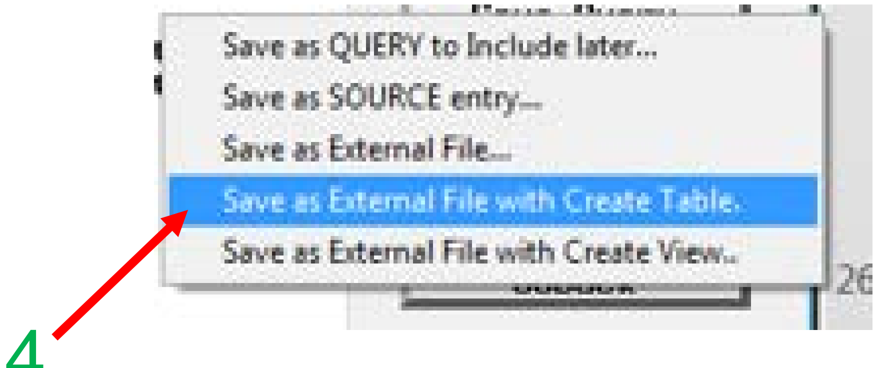
REL MID-ATLANTIC Regional Educational Laboratory at ICF International

Save SQL query

1. Right click in the Available Columns area
2. Select Show Query from drop-down menu
3. Click Save Query
4. Click Save as External File with Create Table

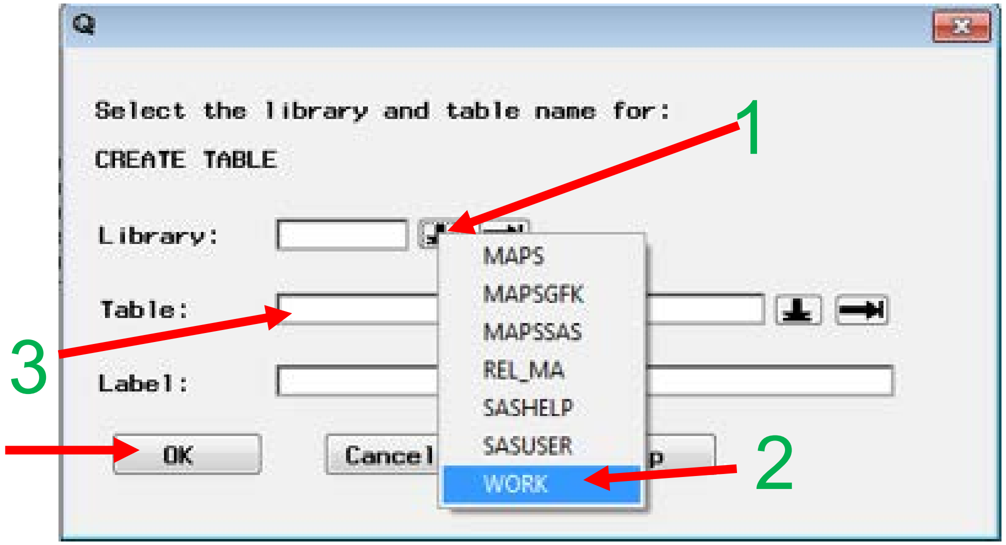


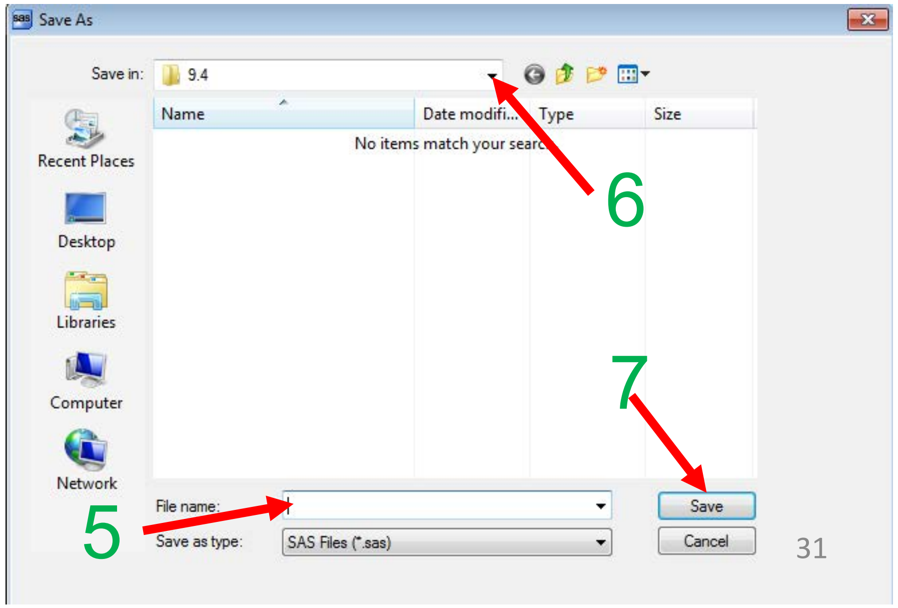



4
Save as QUERY to Include later...
Save as SOURCE entry...
Save as External File...
Save as External File with Create Table.
Save as External File with Create View..

Save SQL query - continued

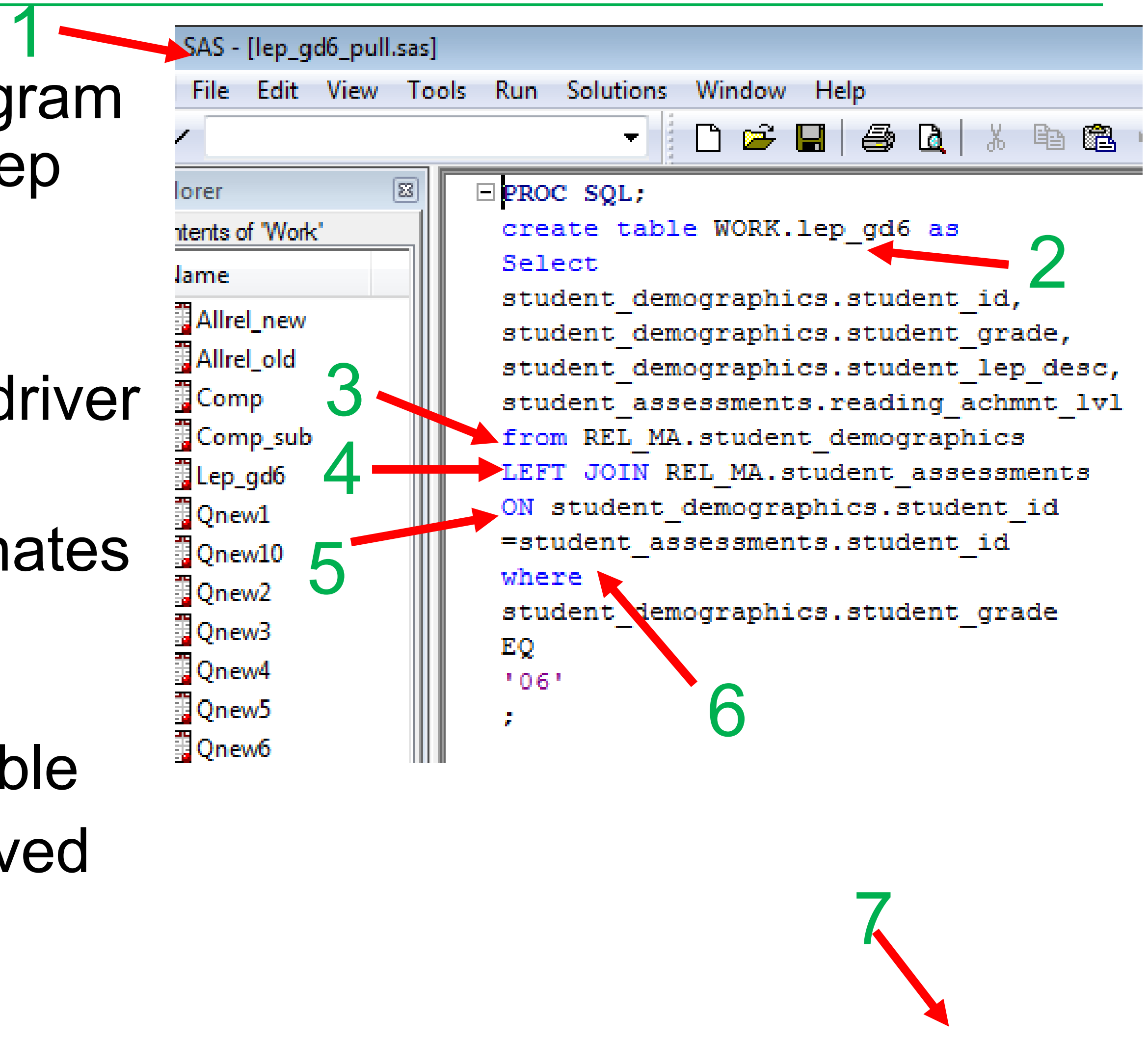
1. Click Down Arrow to designate library
2. Choose 'WORK' from drop-down
3. Enter 'lep_gd6' as Table name
4. Click OK
5. Enter SAS program file name
6. Save program file in your chosen directory
7. Click Save





SQL query in SAS

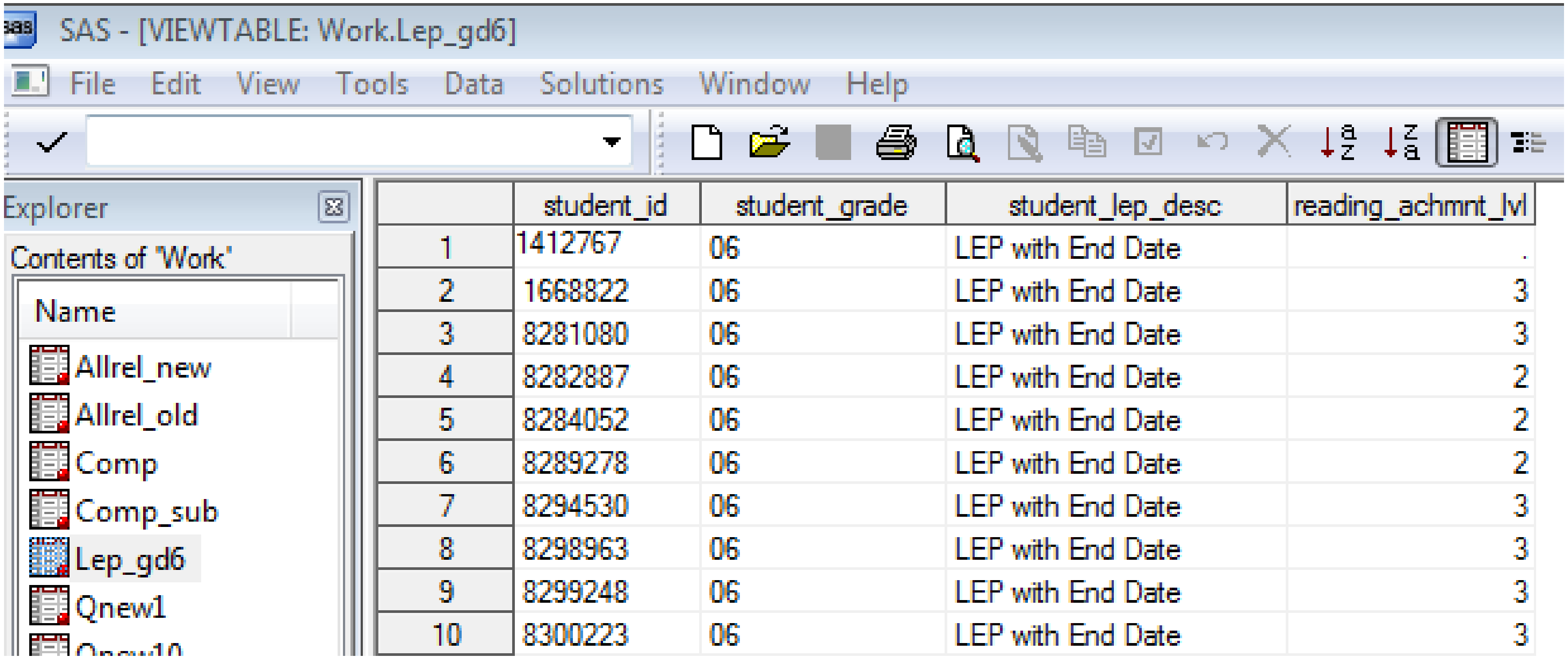
1. Open the SAS program saved in the last step
2. ‘Create table’ code names data
3. ‘From’ designates driver table
4. ‘LEFT JOIN’ designates table to be merged
5. ‘ON’ determines matching key variable
6. ‘where’ limits retrieved data



```
PROC SQL;
create table WORK.lep_gd6 as
Select
student_demographics.student_id,
student_demographics.student_grade,
student_demographics.student_lep_desc,
student_assessments.reading_achmnt_lvl
from REL_MA.student_demographics
LEFT JOIN REL_MA.student_assessments
ON student_demographics.student_id
=student_assessments.student_id
where
student_demographics.student_grade
EQ
'06'
;
```

Retrieved data in table form

- Resulting data returned to SAS, ready for analysis


SAS - [VIEWTABLE: Work.Lep_gd6]
File Edit View Tools Data Solutions Window Help
Explorer
Contents of 'Work'
Name
Allrel_new
Allrel_old
Comp
Comp_sub
Lep_gd6
Qnew1

	student_id	student_grade	student_lep_desc	reading_achmnt_lvl
1	1412767	06	LEP with End Date	.
2	1668822	06	LEP with End Date	3
3	8281080	06	LEP with End Date	3
4	8282887	06	LEP with End Date	2
5	8284052	06	LEP with End Date	2
6	8289278	06	LEP with End Date	2
7	8294530	06	LEP with End Date	3
8	8298963	06	LEP with End Date	3
9	8299248	06	LEP with End Date	3
10	8300223	06	LEP with End Date	3

Summarize data retrieved from connection

- If we want something that is more friendly for leadership, we may want to summarize the raw data
- For this example, we will use a Crosstabs analysis to summarize the data

Generating crosstabs

- Let’s create a Crosstabs table using the returned data
  - We’ll specify reading achievement levels as columns and student LEP status as rows
  - We accomplish this using PROC FREQUENCY syntax in SAS
  - Copy-paste the syntax below into the SAS program editor

1. data= specified the table to use (lep_gd6)
2. The first variable (lep) in the table statement will be the rows of the table, the second (reading achievement) will be the columns
3. Nocol and nopercent mean there will be no percentages based on column totals or cell totals. Only percentages within each row will be displayed

1

2

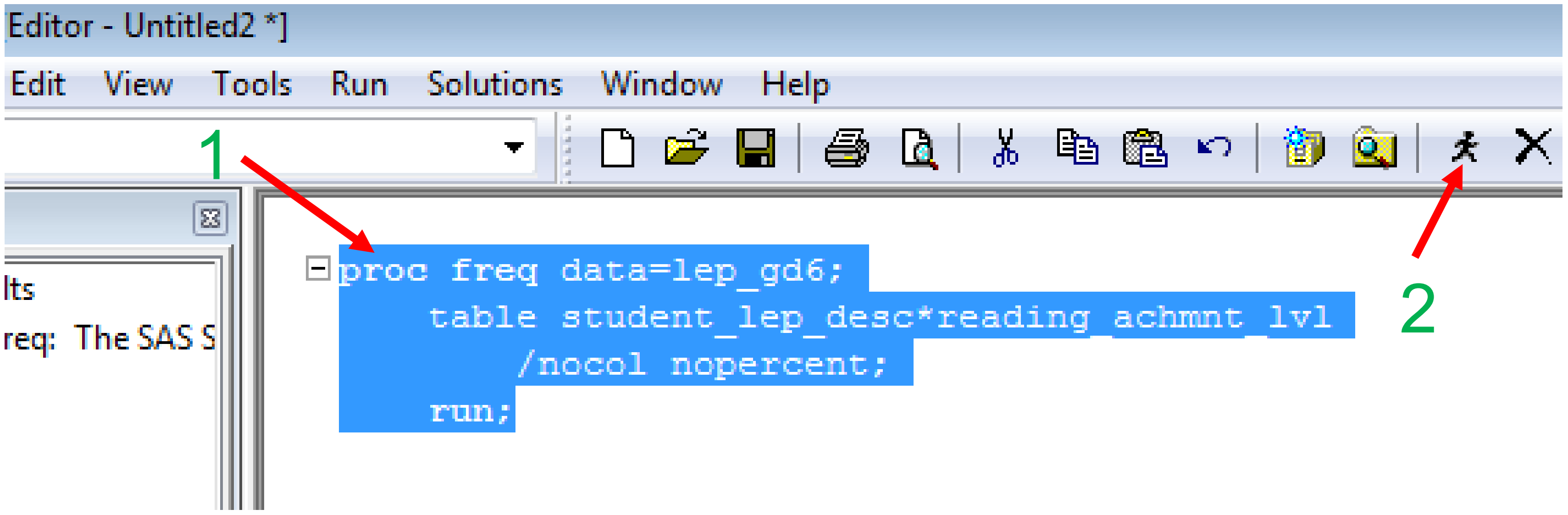
```
proc freq data=lep_gd6;
       table student_lep_desc*reading_achmnt_lvl/
             nocol nopercent;
       run;
```

3

SAS PROC FREQ syntax

1. Highlight the PROC FREQ syntax using your mouse
2. Click the SAS running man icon to execute the syntax

5



```
proc freq data=lep_gd6;
    table student_lep_desc*reading_achmnt_lvl
        /nocol nopercent;
    run;
```

REL MID-ATLANTIC Regional Educational Laboratory at ICF International

PivotTable – Finished product

- Now we have counts and percentages within each LEP Status across Reading achievement levels
- Should the parameters of the request change, or new data become available, the data can quickly be refreshed using the connection

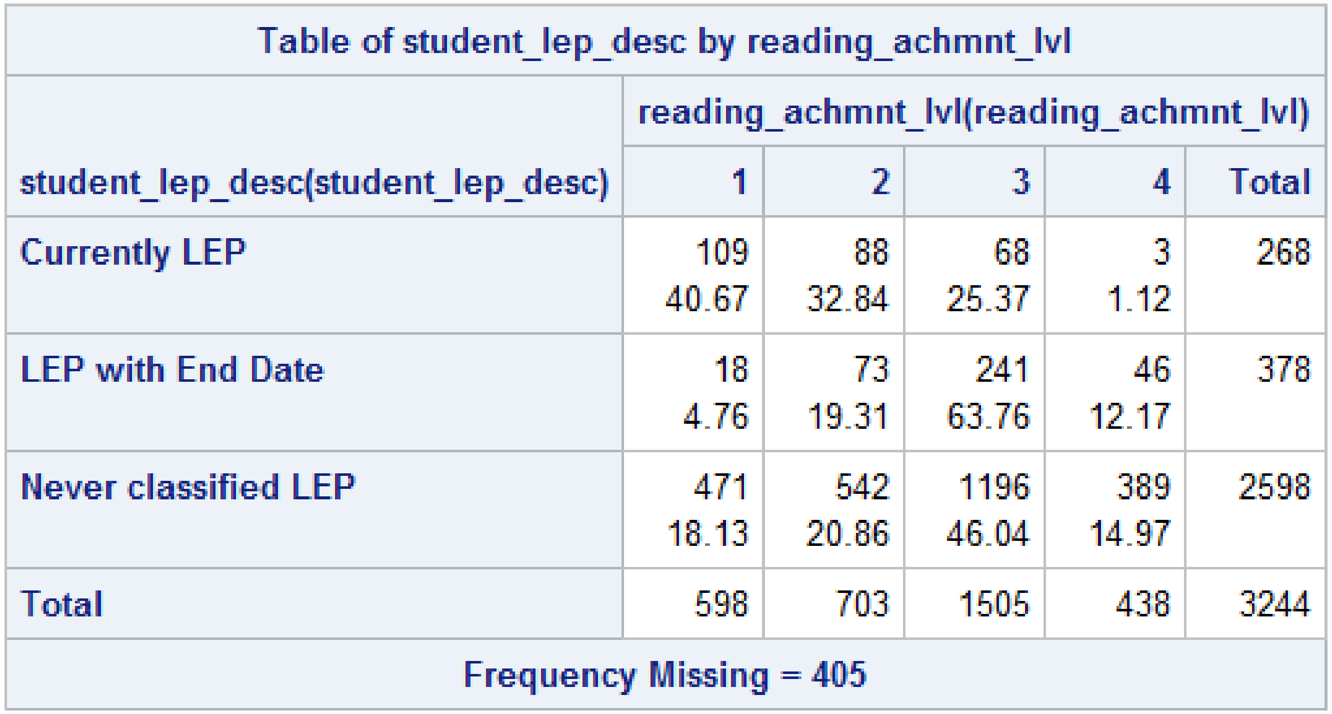


Table of student_lep_desc by reading_achmnt_lvl

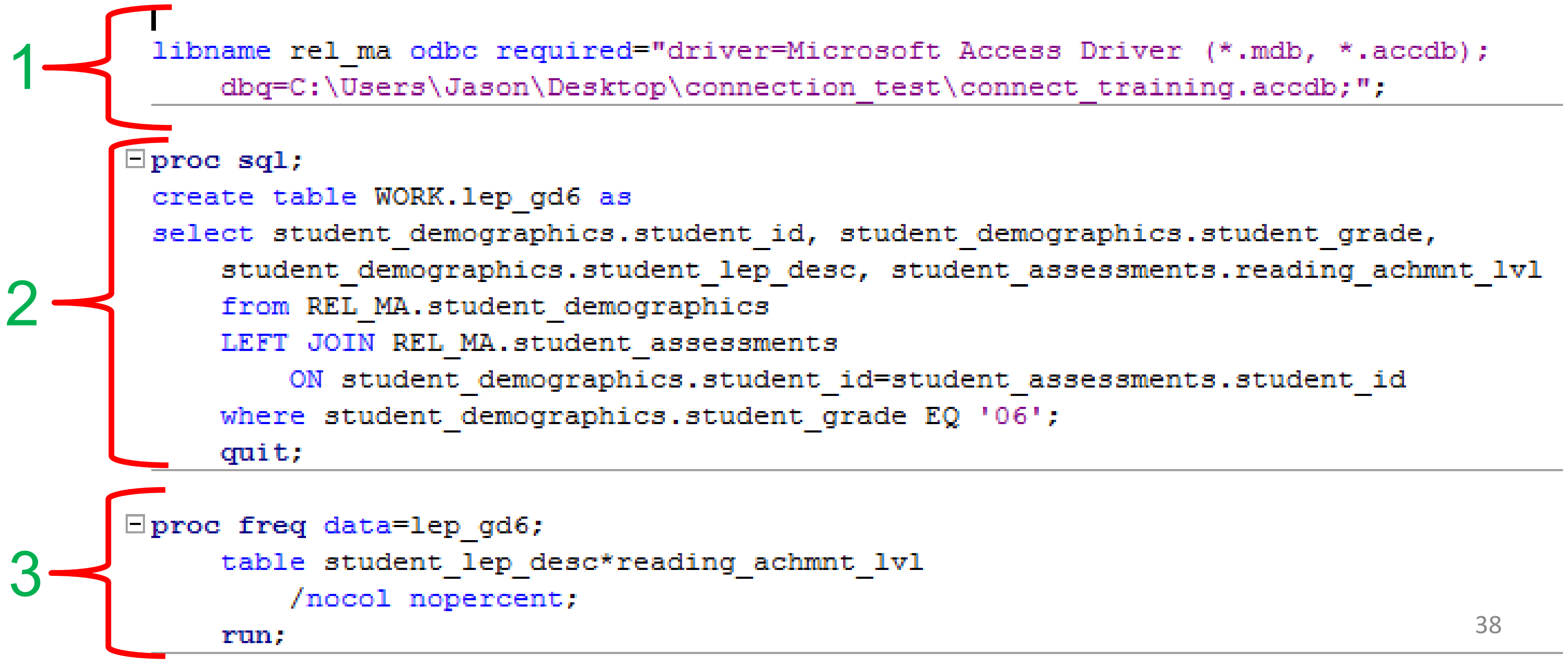
	reading_achmnt_lvl(reading_achmnt_lvl)				
student_lep_desc(student_lep_desc)	1	2	3	4	Total
Currently LEP	109 40.67	88 32.84	68 25.37	3 1.12	268
LEP with End Date	18 4.76	73 19.31	241 63.76	46 12.17	378
Never classified LEP	471 18.13	542 20.86	1196 46.04	389 14.97	2598
Total	598	703	1505	438	3244

Frequency Missing = 405

Comprehensive syntax

- Syntax can be advantageous for repetitive tasks

1. Establish libname connection to data source
2. Syntax for retrieving data from database
3. Syntax for generating crosstab table

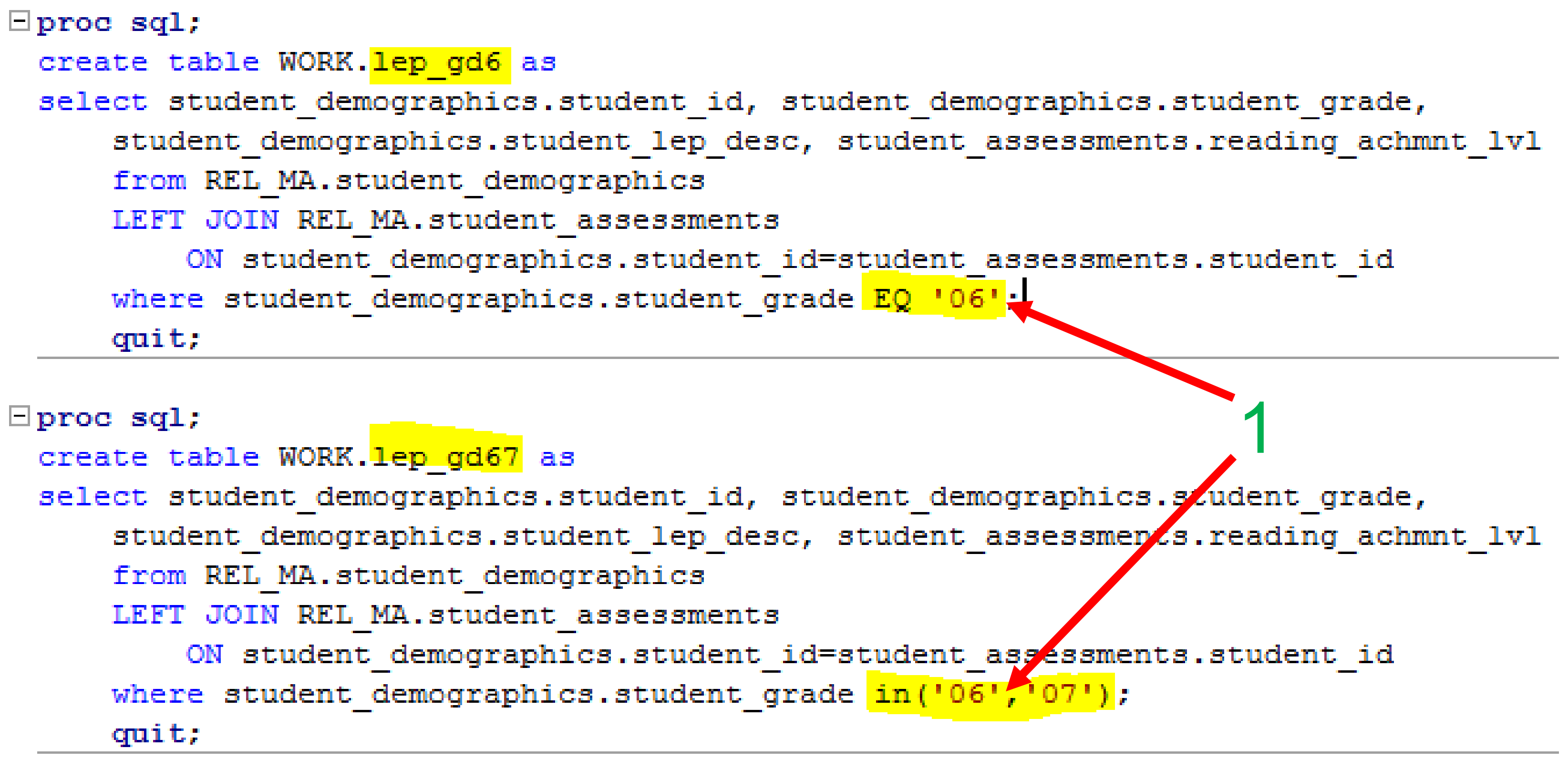


```
libname rel_ma odbc required="driver=Microsoft Access Driver (*.mdb, *.accdb);
    dbq=C:\Users\Jason\Desktop\connection_test\connect_training.accdb;";

proc sql;
create table WORK.lep_gd6 as
select student_demographics.student_id, student_demographics.student_grade,
    student_demographics.student_lep_desc, student_assessments.reading_achmnt_lvl
    from REL_MA.student_demographics
    LEFT JOIN REL_MA.student_assessments
        ON student_demographics.student_id=student_assessments.student_id
    where student_demographics.student_grade EQ '06';
    quit;

proc freq data=lep_gd6;
    table student_lep_desc*reading_achmnt_lvl
        /nocol nopercent;
    run;
```

Editing the query syntax

- Maybe leadership wants data for 6th and 7th graders

1. Edit SQL syntax to pull both grade levels (compare highlighted sections

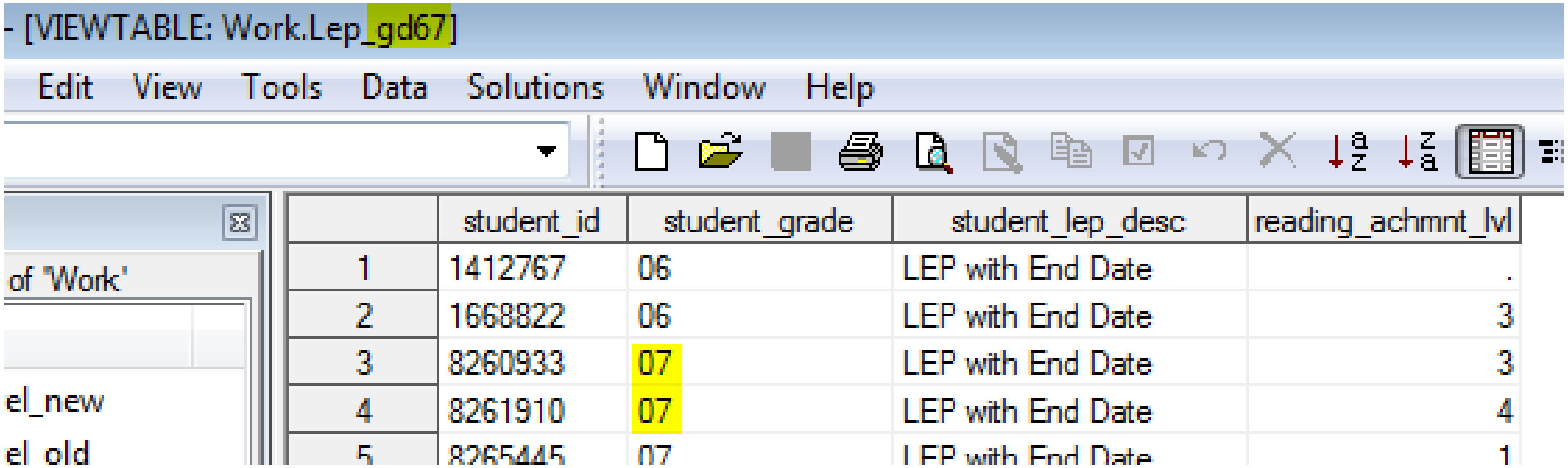


```
proc sql;
create table WORK.lep_gd6 as
select student_demographics.student_id, student_demographics.student_grade,
    student_demographics.student_lep_desc, student_assessments.reading_achmnt_lvl
    from REL_MA.student_demographics
    LEFT JOIN REL_MA.student_assessments
        ON student_demographics.student_id=student_assessments.student_id
    where student_demographics.student_grade EQ '06';
    quit;
```

```
proc sql;
create table WORK.lep_gd67 as
select student_demographics.student_id, student_demographics.student_grade,
    student_demographics.student_lep_desc, student_assessments.reading_achmnt_lvl
    from REL_MA.student_demographics
    LEFT JOIN REL_MA.student_assessments
        ON student_demographics.student_id=student_assessments.student_id
    where student_demographics.student_grade in('06','07');
    quit;
```

Retrieved data from edited query

- Resulting data, with 6th and 7th graders, returned to SAS, ready for analysis


- [VIEWTABLE: Work.Lep_gd67]
Edit View Tools Data Solutions Window Help
of 'Work'
el_new
el_old

	student_id	student_grade	student_lep_desc	reading_achmnt_lvl
1	1412767	06	LEP with End Date	.
2	1668822	06	LEP with End Date	3
3	8260933	07	LEP with End Date	3
4	8261910	07	LEP with End Date	4
5	8265445	07	LEP with End Date	1

Questions/Need help

Contact:

Jason Schoeneberger, Ph.D.

Senior Researcher and Task Lead

REL Mid-Atlantic at ICF International

jason.schoeneberger@icfi.com

704-307-9395

Please visit www.relmidatlantic.org for other data tools!