How can I use odbc to load, write, or view a dBASE file, Excel file or Access file in Stata?

Authored by

stats writer

June 30, 2024

ODBC (Open Database Connectivity) is a standard interface that allows different programs to communicate with databases. In the context of Stata, ODBC can be used to load, write, or view data from dBASE, Excel, or Access files. This means that users can easily import data from these file types into Stata for analysis, or export data from Stata into these file types for further processing. By establishing a connection between Stata and the ODBC driver for the specific file type, users can seamlessly transfer data between programs without the need for manual data entry. This provides a convenient and efficient way to work with data from different sources within the Stata environment.

How can I load, write, or view a dBASE file, Excel file or Access file using odbc? | Stata FAQ

Note: This page is written using Stata 11 and Windows XP.

Stata command odbc allows Stata to load, write, or view data from ODBC sources. ODBC, an acronym for Open DataBase Connectivity, is a standardized set of function calls that can be used to access data stored in database management systems. A dBASE file, an Excel file, or an ACCESS file are all examples of data files created by database systems.

Why might I use odbc?

On this page, we will go through loading a multi-sheet Excel file and inserting columns from within Stata and an ACCESS database with linked tables and a query that can be modified from within Stata.

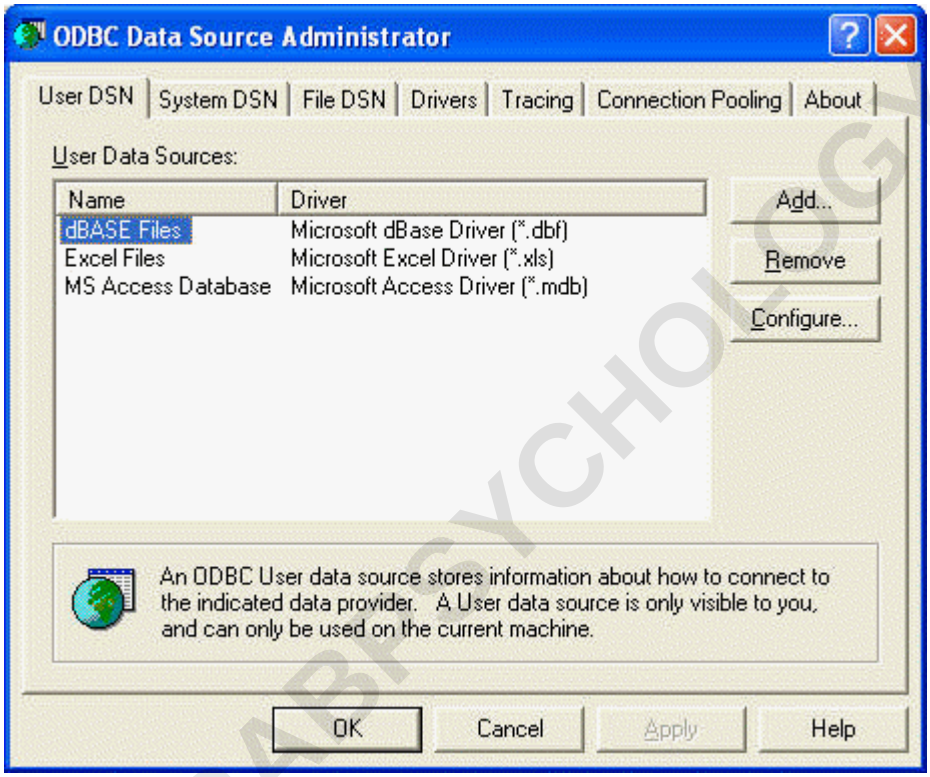
Setting Up a Data Source in Windows for Reading Excel and Access files via odbc in Stata

To read an Excel or Access file into Stata, we must establish the file as a via Data Source. The key step here is to register your database with your computer's ODBC system. Here is an example showing how to do it for Windows XP. You may also want to read Stata manual on Data Management for more details. In this example, we have one Access file, /stata/faq/hsbdb.mdb. The same procedure works for both Excel and Access files.

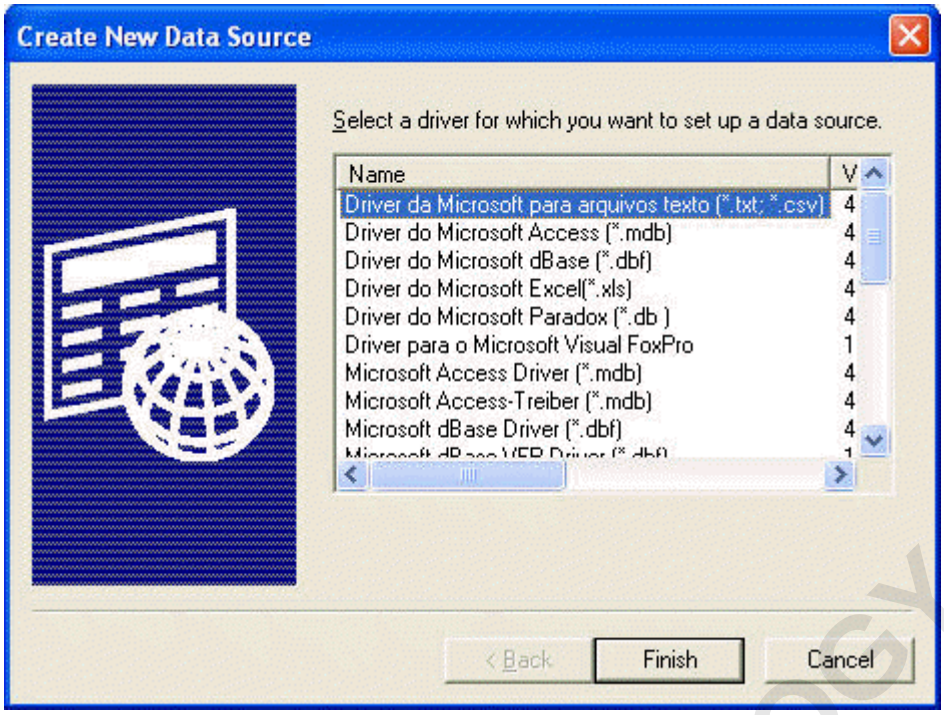
Step 1: From the Start Menu, select Settings and then the

Control Panel.

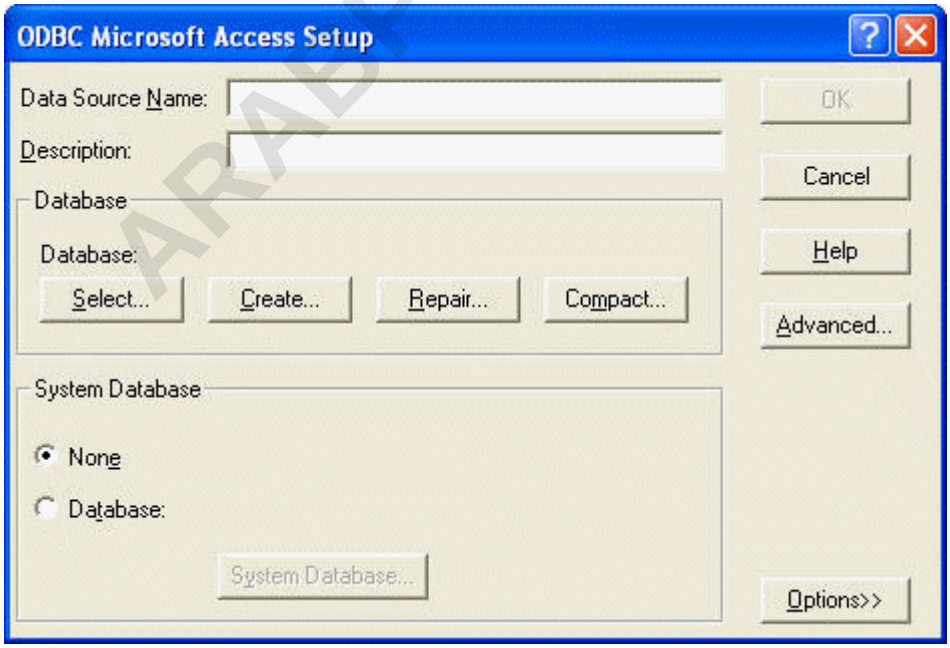
Step 2: Select Administrator Tools and then select Data Sources (ODBC). This will bring up a window like the following window:



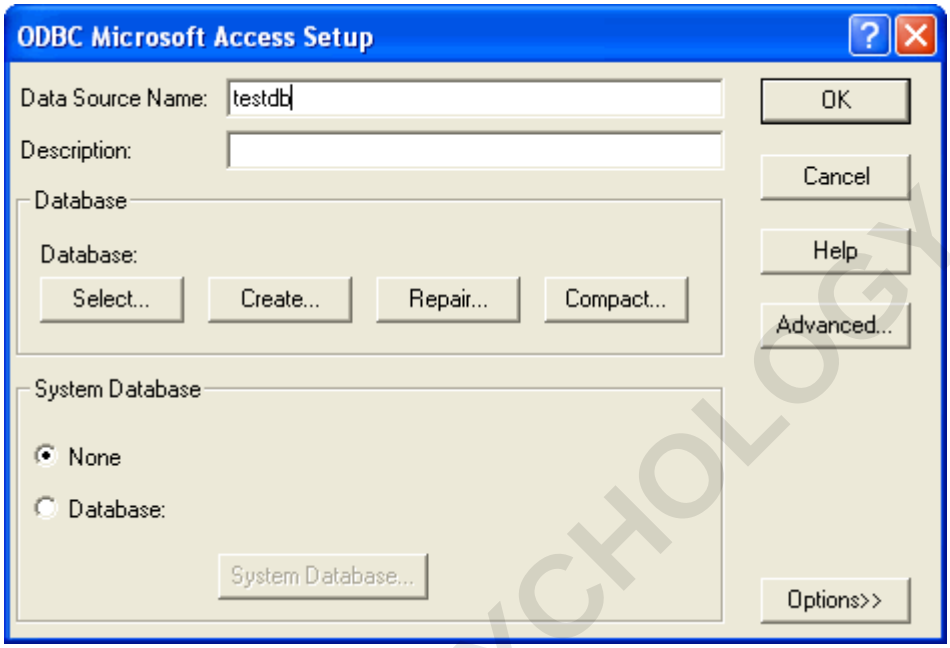
Step 3: Click on Add... and this will bring up the following window.



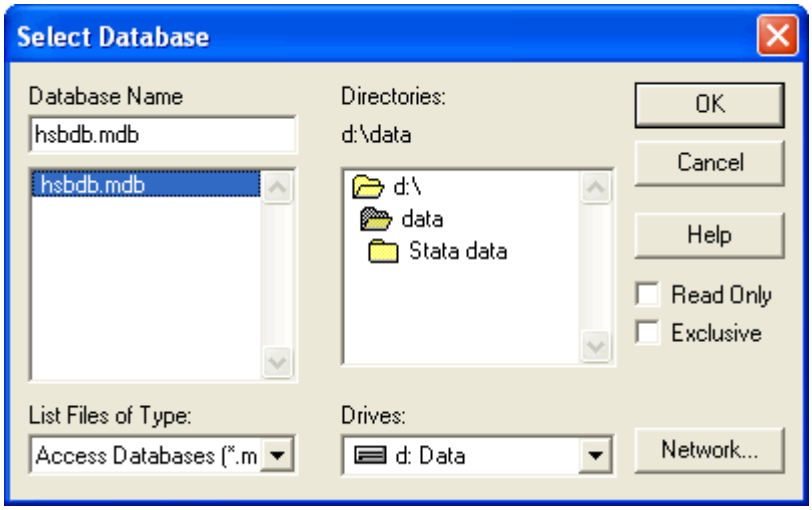
Step 4: Select Microsoft Access Driver (*.mdb) and then click on Finish.



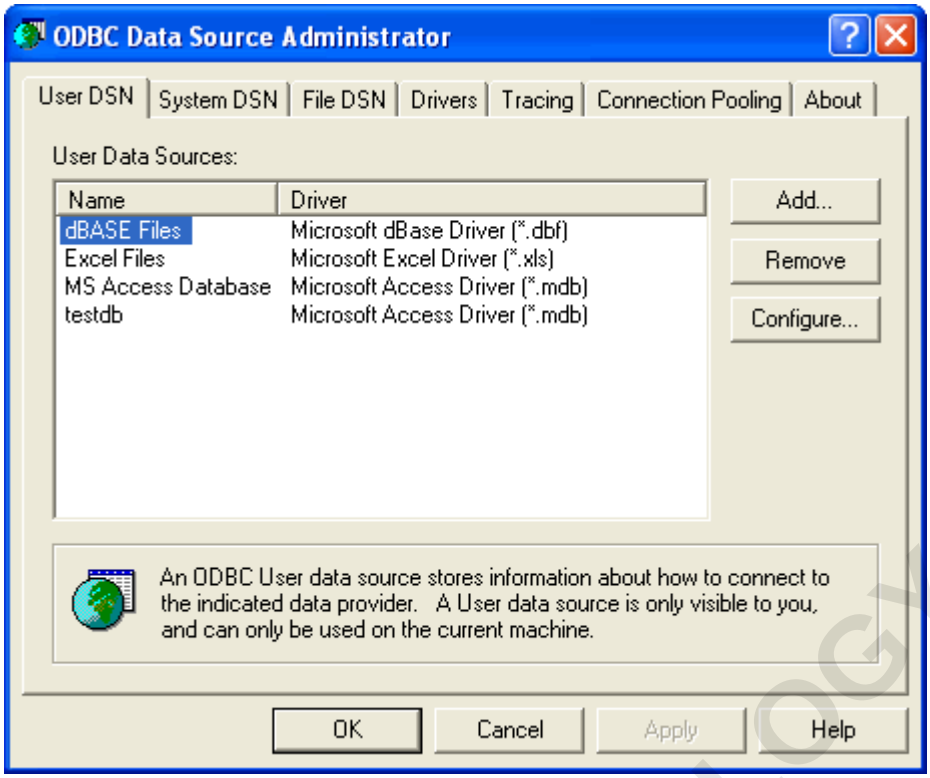
Step 5: The name for the Data Source can be arbitrary. Here we will use testdb as our data source name for the Access database.



Step 6: Now, the last step is to select all the Access files that you want to be associated with this data source. This is done by "Select". Click on "Select", you will see something similar to the following:

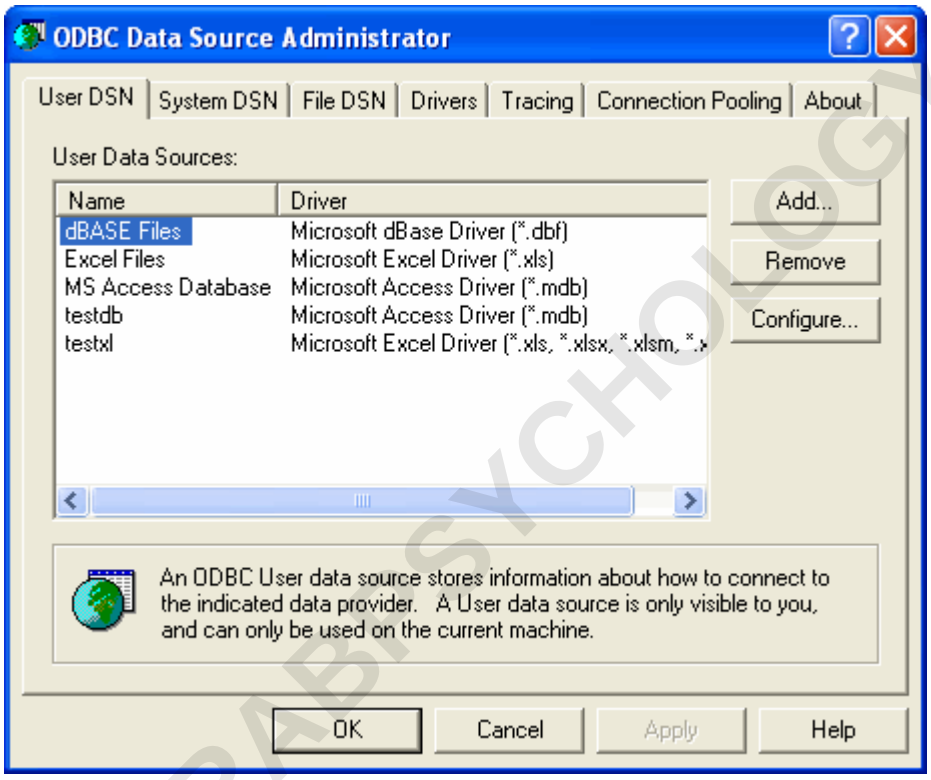


Step 7: Now you can select your .mdb file to be in the Data Source testdb that we are creating. After select your file, you can then
click on OK to close this window and click on OK again to close previous
window.



Notice that we have a new entry in the window above, that is our testdb
Data Source. We will do the same for a multisheet Excel file,
https://stats.idre.ucla.edu/wp-content/uploads/2016/02/multiplesheets.xls,
assigning it the Data Source name testxl. If your computer has multiple versions of Microsoft Office, you will need to
be pickier about the driver you choose. You should select the driver that
supports all of the file extensions. For Excel files, look

for
Microsoft Excel Driver (*.xls, *.xlsx, *.xlsm, *.xlsb). After we have
added our Excel file as a data source, our window list includes both of our
additions.



Reading and Altering Excel files via odbc in Stata

After setting up our Data Sources, we are ready to get access to our Excel file via odbc in Stata. After typing "odbc list", you can just
follow the links to access a given sheet in our Excel

data source testxl:
click on testxl, then from the query list, then click on scores$, and
then from the describe output for that sheet, click on load.
Alternatively, you can enter the equivalent commands following the example
below. If there is a dataset loaded in Stata prior to these steps, you
must enter clear before loading the dataset from the Data Source.

odbc list

Data Source Name Driver

MS Access Database Microsoft Access Driver (*.mdb)
Excel Files Microsoft Excel Driver (*.xls)
dBASE Files Microsoft dBase Driver (*.dbf)
testdb Microsoft Access Driver (*.mdb)
testxl Microsoft Excel Driver (*.xls, *.xlsx, *.xl
----

odbc query "testxl"

DataSource: testxl
Path : D:datamultiplesheets
---
demo$
scores$
demo$_FilterDatabase
scores$_FilterDatabase
---

odbc desc "scores$"

DataSource: testxl (query)
Table: scores$ (load)

Variable Name Variable Type
---
id NUMBER
read NUMBER
write NUMBER
math NUMBER
science NUMBER
socst NUMBER
---

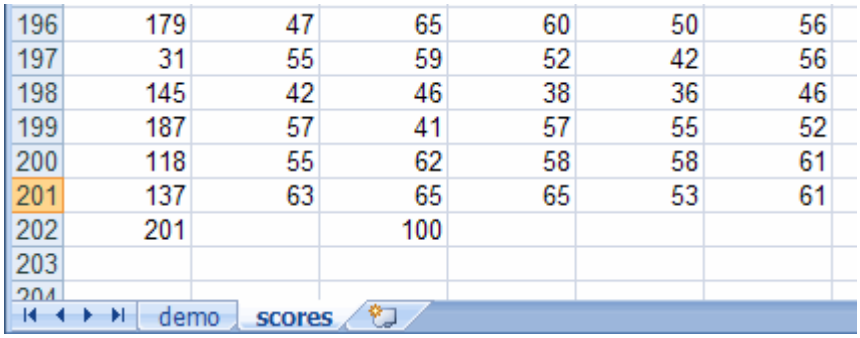
odbc load, table("scores$")

With this new dataset loaded into Stata, we can now make changes to the dataset and output them to Excel using odbc.

Adding observations: We can add a new observation to the dataset by creating a new dataset with 1 observation, defining some values in the observation, and then inserting the observation into our scores tab in the Excel file:

```
clear
set obs 1
gen id =201
gen write = 100
odbc insert id write , table("") dsn("testxl")
```

We can look at the scores tab in Excel to see this added observation.



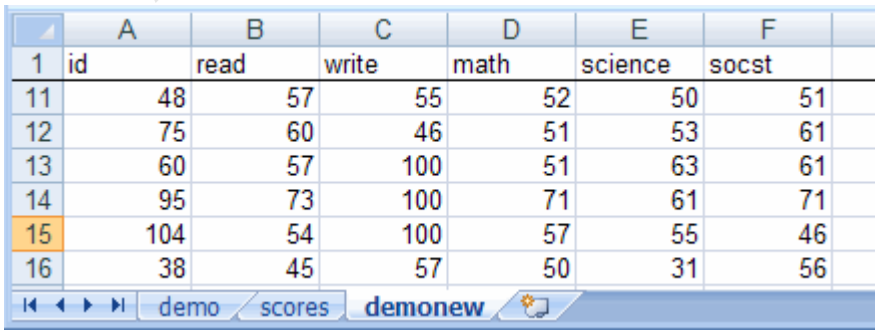
196	179	47	65	60	50	56
197	31	55	59	52	42	56
198	145	42	46	38	36	46
199	187	57	41	57	55	52
200	118	55	62	58	58	61
201	137	63	65	65	53	61
202	201		100			
203						

demo scores

Creating a new data tab: We can make changes to our data and output
the altered dataset to a new tab in our Excel file.

```
odbc load, table("scores$") dsn("testxl")
replace write = 100 if write >=60
odbc insert id read write math science socst, table("")
dsn("testxl") create
```

We can see that a new tab has been added to our Excel file in which write
values of 60 or greater have been replaced with 100.

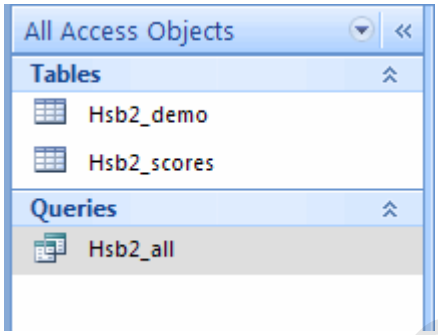


	A	B	C	D	E	F
1	id	read	write	math	science	socst
11	48	57	55	52	50	51
12	75	60	46	51	53	61
13	60	57	100	51	63	61
14	95	73	100	71	61	71
15	104	54	100	57	55	46
16	38	45	57	50	31	56

demo scores demonew

Reading and Altering Access Database files via odbc in Stata

Our example Access file has two tables-one containing demographic
information for each student and one containing scores for each student-and one
query linking these two students into a single table with both demographics and
scores.



We can view the full dataset by selecting the Hsb2_all query.

Hsb2_all

id	female	race	ses	schtyp	prog	read	write	math	science	socst
1	"female"	"hispanic"	"low"	"public"	"vocation"	34	44	40	39	41
2	"female"	"hispanic"	"middle"	"public"	"vocation"	39	41	33	42	41
3	"male"	"hispanic"	"low"	"public"	"academic"	63	65	48	63	56
4	"female"	"hispanic"	"low"	"public"	"academic"	44	50	41	39	51
5	"male"	"hispanic"	"low"	"public"	"academic"	47	40	43	45	31
6	"female"	"hispanic"	"low"	"public"	"academic"	47	41	46	40	41
7	"male"	"hispanic"	"middle"	"public"	"academic"	57	54	59	47	51
8	"female"	"hispanic"	"low"	"public"	"academic"	39	44	52	44	48
9	"male"	"hispanic"	"middle"	"public"	"vocation"	48	49	52	44	51
10	"female"	"hispanic"	"middle"	"public"	"general"	47	54	49	53	61
11	"male"	"hispanic"	"middle"	"public"	"academic"	34	46	45	39	36
12	"male"	"hispanic"	"middle"	"public"	"vocation"	37	44	45	39	46

Record: 5 of 200 No Filter Search

Following the same steps used for the Excel file, you can load the
Hsb2_scores table into Stata as a dataset (remember to clear any
existing datasets first).

```
clear
odbc load, table("Hsb2_scores") dsn("testdb")
```

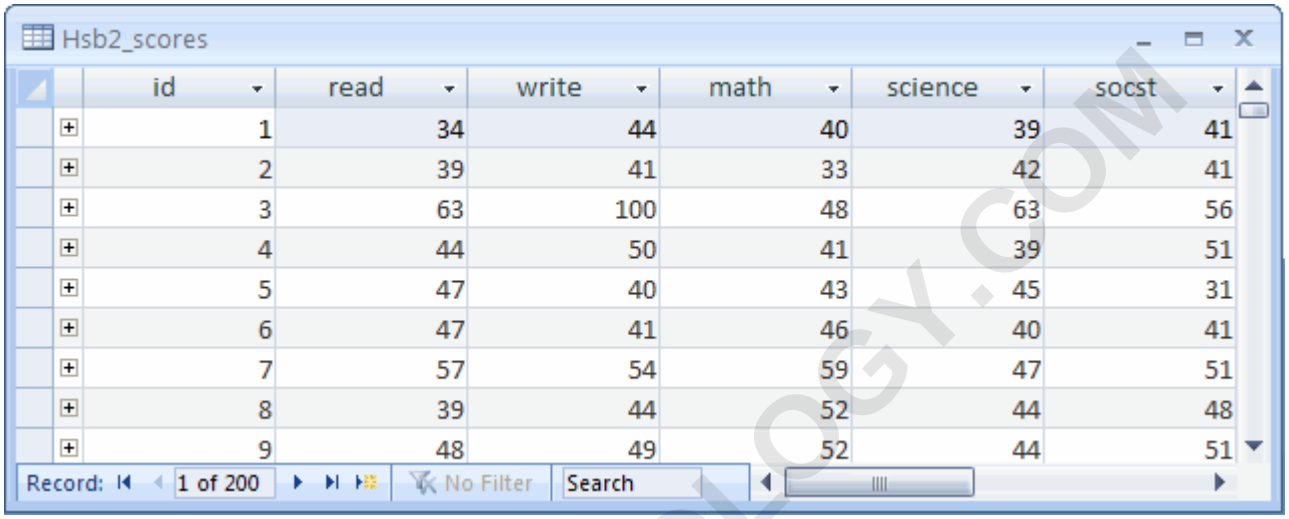
Altering a table: We can make changes to the data in this table in Stata and then export the
altered dataset back to its database. We do this with odbc insert,
listing all of the variables to send to the database. We then indicate the
table and Data Source and, finally, say overwrite to replace the data
currently stored in the given table.

```
replace write = 100 if write >=60
odbc insert id read write math science socst, table(Hsb2_scores) dsn("testdb") overwrite
```

We can look at the updated table in Access to see the

changes. For

id = 3, the write score had previously been 65 and it is now 100.

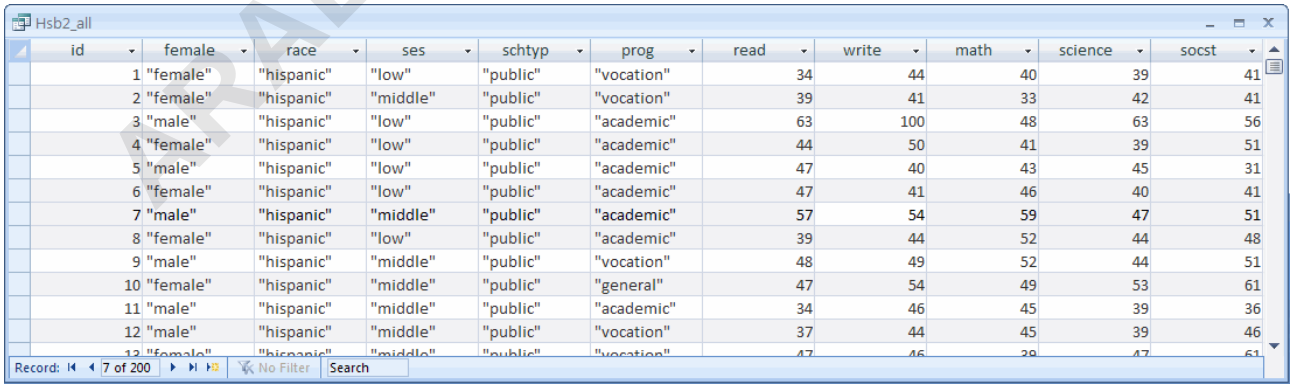


Hsb2_scores

id	read	write	math	science	socst
1	34	44	40	39	41
2	39	41	33	42	41
3	63	100	48	63	56
4	44	50	41	39	51
5	47	40	43	45	31
6	47	41	46	40	41
7	57	54	59	47	51
8	39	44	52	44	48
9	48	49	52	44	51

Record: 1 of 200 No Filter Search

Not only has this table been updated, but the query linking this table to the demographic information also reflects the changes.



Hsb2_all

id	female	race	ses	schtyp	prog	read	write	math	science	socst
1	"female"	"hispanic"	"low"	"public"	"vocation"	34	44	40	39	41
2	"female"	"hispanic"	"middle"	"public"	"vocation"	39	41	33	42	41
3	"male"	"hispanic"	"low"	"public"	"academic"	63	100	48	63	56
4	"female"	"hispanic"	"low"	"public"	"academic"	44	50	41	39	51
5	"male"	"hispanic"	"low"	"public"	"academic"	47	40	43	45	31
6	"female"	"hispanic"	"low"	"public"	"academic"	47	41	46	40	41
7	"male"	"hispanic"	"middle"	"public"	"academic"	57	54	59	47	51
8	"female"	"hispanic"	"low"	"public"	"academic"	39	44	52	44	48
9	"male"	"hispanic"	"middle"	"public"	"vocation"	48	49	52	44	51
10	"female"	"hispanic"	"middle"	"public"	"general"	47	54	49	53	61
11	"male"	"hispanic"	"middle"	"public"	"academic"	34	46	45	39	36
12	"male"	"hispanic"	"middle"	"public"	"vocation"	37	44	45	39	46

Record: 7 of 200 No Filter Search

Creating a new table: Similarly, we can make changes

and then create a
new table in the database.

```
clear
odbc load, table("Hsb2_scores") dsn("testdb")
gen totscore = read + write + math + science + socst
odbc insert id totscore, table(Hsb2_totals) dsn("testdb")
create
```

We can see this new table in our Access database.