

# iSeries Navigator Does SQL and More

submitted by [Steve Kontos](#).

To run SQL statements (or scripts) in iSeries Navigator, do the following:

Expand a configured connection by clicking the little plus (+) sign.

Expand the branch called **Databases**.

*Right Click* on the first entry

Select **Run SQL Scripts** from the context menu.

To change the syntax for entering SQL statements from `library.table` to `library/table`, or vice versa:

Click **Connection** in the iSeries Navigator script window

Click **JDBC setup**

Click the **Format** tab

Select the **Naming Convention** from the drop down list

## Running Control Language Commands In the Script Window

iSeries Navigator SQL has a facility `CL:` to run Control Language commands. For example:

```
CL:DSPOBJD OBJ(STEVEK/*ALL) OBJTYPE(*ALL) OUTPUT(*OUTFILE) OUTFILE(QTEMP/DSPOBJD);
```

The line above outputs a list of objects in library STEVEK.

Now show the 10 biggest objects in library STEVEK:

```
select * from qtemp/dspobjd order by odobsz desc fetch first 10 rows only;
```

The `CL:` facility in iSeries Nav does not extend to interactive commands. Trying to run, say,

```
CL:call qcmd
```

will result in an error.

The two lines below form a simple script which can be saved and reused.

```
CL:DSPOBJD OBJ(STEVEK/*ALL) OBJTYPE(*ALL) OUTPUT(*OUTFILE) OUTFILE(QTEMP/DSPOBJD);  
select * from qtemp/dspobjd order by odobsz desc fetch first 10 rows only;
```

To run a script, all SQL statements need to be terminated with a semi-colon. `Ctrl-R` will run the script. You can also select **Run --> All** from the menu bar.

To save a script that you have created, click **File** and then **Save** and save your script.

In addition to executing `CL` commands via the script facility, any SQL statement (or

statements) can be saved as a script.

## Creating Work Files

```
/* Delete the table */  
  
drop table stevek/histwork;  
  
/* Create the table from a SELECT statement */  
  
create table stevek/histwork as  
(select * from proddta/f42119 left join proddta/f5542119  
on sdkcoo=s1kcoo and sddoco=s1doco and sddcto=s1dcto and sdlnid=s1lnid  
where sddrqj between 106000 and 106999)  
with data;
```

The example above creates a workfile of sales orders for the year-to-date for 2006 (note that `sddrqj` is a Julian date field).

## Creating Permanent Files

You can document the creation of a database via the scripting facility in iSeries Nav.

First create a table called COMPANY, keyed by company number.

```
CREATE TABLE company  
(  
company_company for column compcomp NUMERIC (3, 0) CONSTRAINT pk_compcomp PRIMARY KEY,  
company_name for column compname VARCHAR (50) NOT NULL,  
company_status for column compstat CHAR (1) NOT NULL DEFAULT 'A'  
);
```

Specifying `NOT NULL` is not necessary, but is something you might want to do if you develop with RPG. RPG can handle nulls, but not without a little help on compilation, or a lot of help in the calculation specifications.

Now create another table PROJECT, keyed by project code, explicitly specifying the `NOT NULL` values:

```
CREATE TABLE PROJECT (  
PROJECT_COMPANY FOR COLUMN PROJCOMP NUMERIC(3, 0) NOT NULL DEFAULT 0 ,  
PROJECT_CODE FOR COLUMN PROJCODE CHAR(10) NOT NULL DEFAULT ' ' ,  
PROJECT_NAME FOR COLUMN PROJNAME VARCHAR(50) NOT NULL DEFAULT ' ' ,  
PROJECT_STATUS FOR COLUMN PROJSTATUS CHAR(1) NOT NULL DEFAULT 'A' ,  
CONSTRAINT PK_PROJCODE PRIMARY KEY( PROJECT_CODE ) ) ;
```

Finally, relate the company column in the PROJECT table to its parent:

```
ALTER TABLE PROJECT  
ADD CONSTRAINT FK_PROJCOMP  
FOREIGN KEY( PROJECT_COMPANY )  
REFERENCES COMPANY ( COMPANY_COMPANY )  
ON DELETE NO ACTION  
ON UPDATE NO ACTION ;
```

Now, you can start inserting records.

```
INSERT INTO company VALUES(1, 'TEST COMPANY', default);  
INSERT INTO project VALUES(1, 'STEVEK001', ' Steve K Project #001 Test', default);
```

The things you can do with CREATE and ALTER are too numerous to list, but if you have never created or modified files outside of DDS, you might want to research this a little further. See the reference section at the end of this document.

Whether it is creating files, rebuilding indexes, running cleanup utilities, or whatever you can think of – the SQL scripting facility in iSeries Navigator is another tool available to you.

## **Reference**

DB2 UDB for iSeries SQL Programming Concepts V5R2

<http://publib.boulder.ibm.com/series/v5r2/ic2924/index.htm?info/sqlp/rbafymstdbs.htm>

DB2 Universal Database for iSeries SQL Reference

<http://publib.boulder.ibm.com/series/v5r2/ic2924/index.htm?info/db2/rbafzmst02.htm>

“SQL Bible”. Alex Kriegel, Boris M. Trkhnov. Wiley Press, 2003.

<http://www.wiley.com/WileyCDA/WileyTitle/productCd-0764525840.html>