Using □SQLDB2 For IBM DB2

Contents

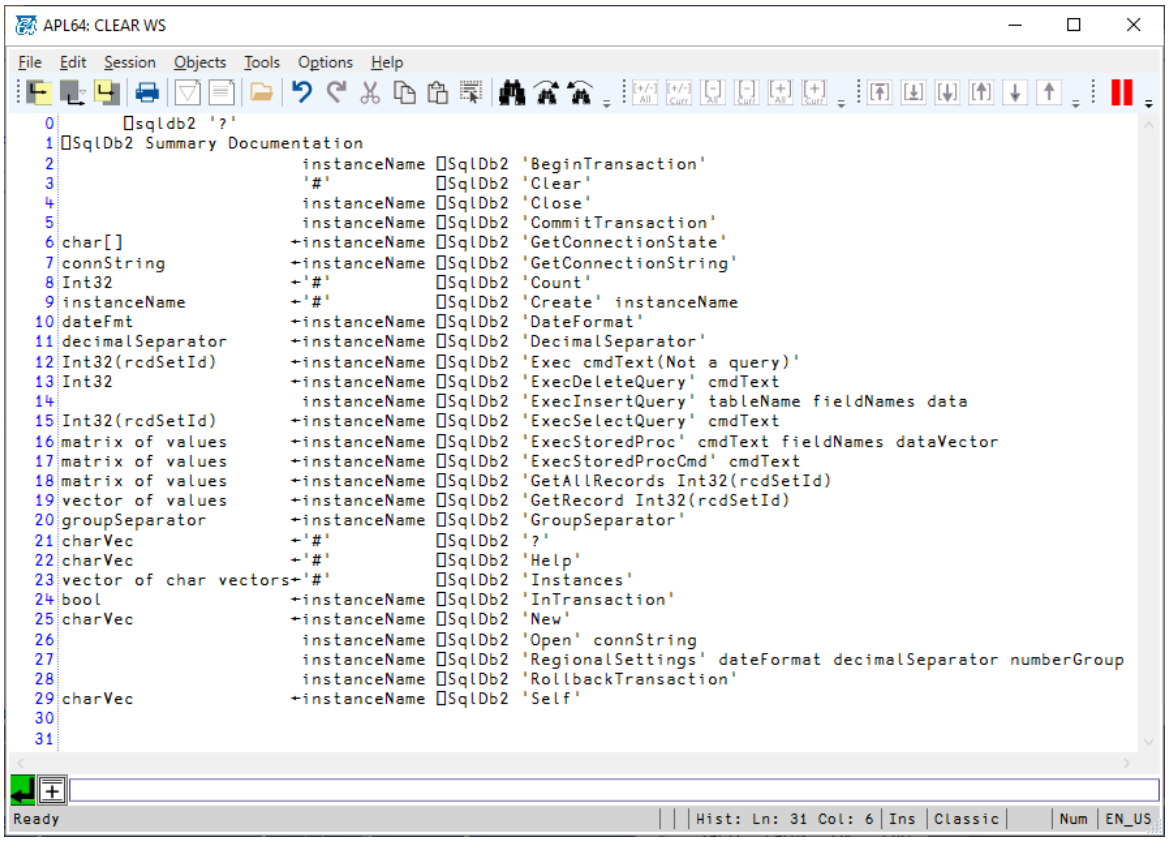
Overview

The APL64 □SQLDB2 system function provides a structured query language (SQL)-based interface to an IBM DB2 database.

The Tool employs the IBM.Data.DB2 .Net assembly so APL64 executable statements may be used to Select, Delete and Insert components of an IBM DB2 database.

With the (no-cost) availability of the IBM DB2 Community Edition, scalable-to-enterprise-level database software is more affordable and the logical choice for professionally-designed application systems.

The object model for □SQLDB2 includes methods and properties to easily create a secure, high-performance application system data interface.



Prerequisites

To use ⎕SQLDB2, the target workstation must have appropriate access to an existing IBM DB2 database. That IBM DB2 database software may be locally installed, installed on a local server or cloud based.

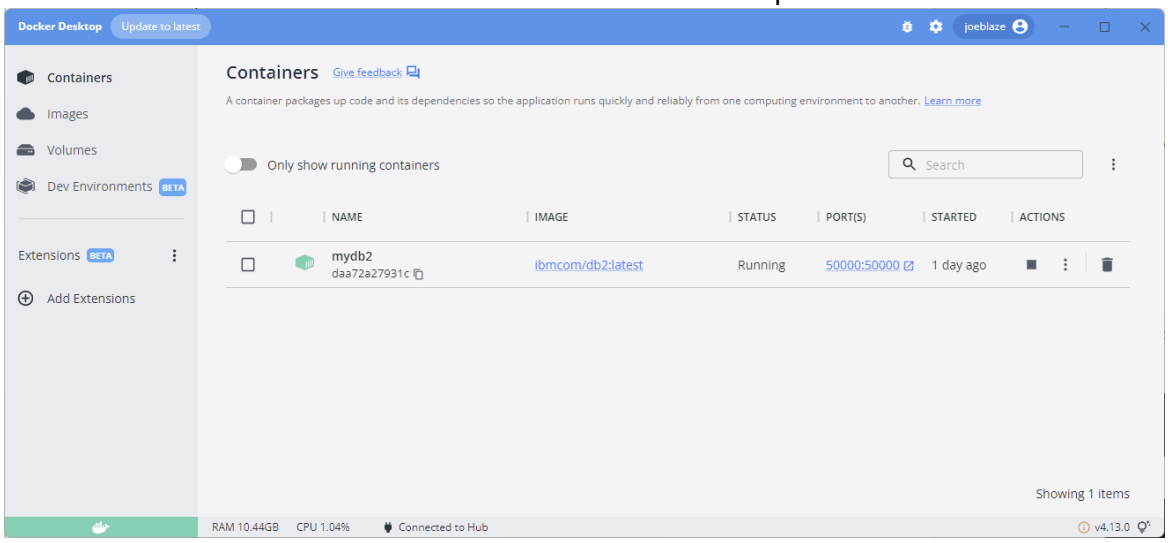
The examples in this document assume that a sample IBMDB2 database, named 'TESTDB' is installed, running and accessible to the target workstation. The examples in this document also assume that this sample IBM DB2 database has a table, named 'tbOne' exists with sample columns and sample data.

The examples in this document are for demonstration purposes, so they may not incorporate enterprise-level security measures, use a local workstation deployment not amenable to sharing data, and use a simplified connection string. For simplicity, some examples do not include all APL64 executable statements which may be necessary to reproduce the illustrated output. Some of the output from the examples was generated from multi-row APL64 executable statements.

Installation of IBM DB2 Community Edition

Full installation details of an IBM DB2 database are beyond the scope of this document. Review the IBM DB2 documentation for installing a test version of DB2. This blog post can also be helpful. The steps used for local installation of IBM DB2 on a Windows workstation include:

- Obtain an IBM id
- Install Docker, a container maintenance tool which simplifies the installation of a DB2 container pre-prepared by IBM
- Install the IBM DB2 container using a Docker command from the Docker terminal.
- Run the IBM DB2 Docker container from the Docker Desktop:

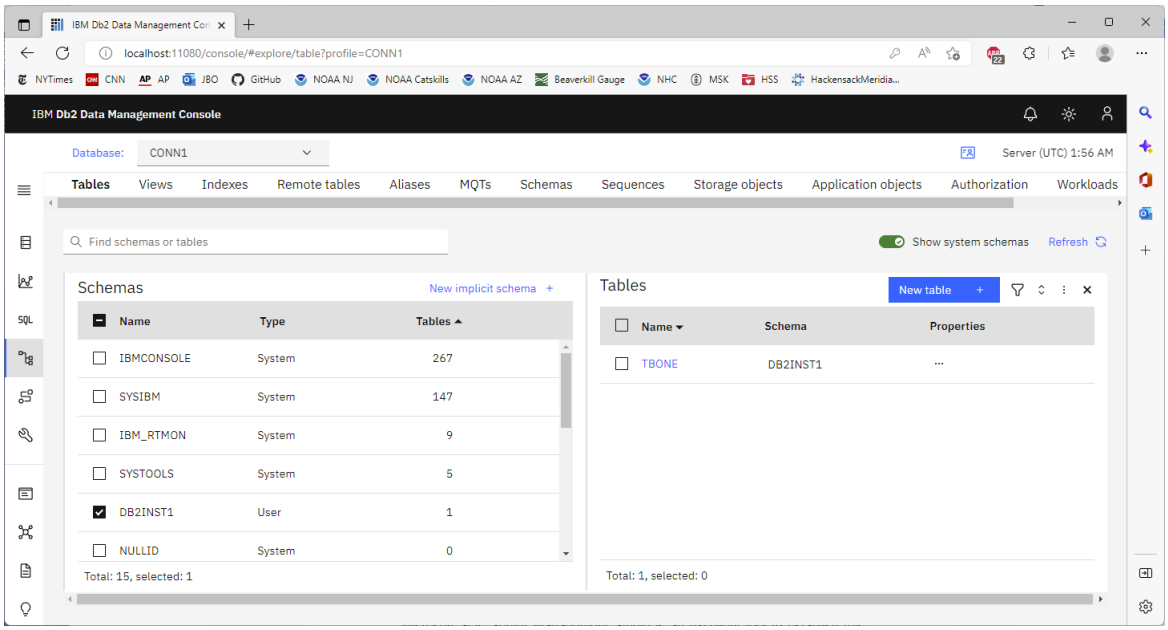


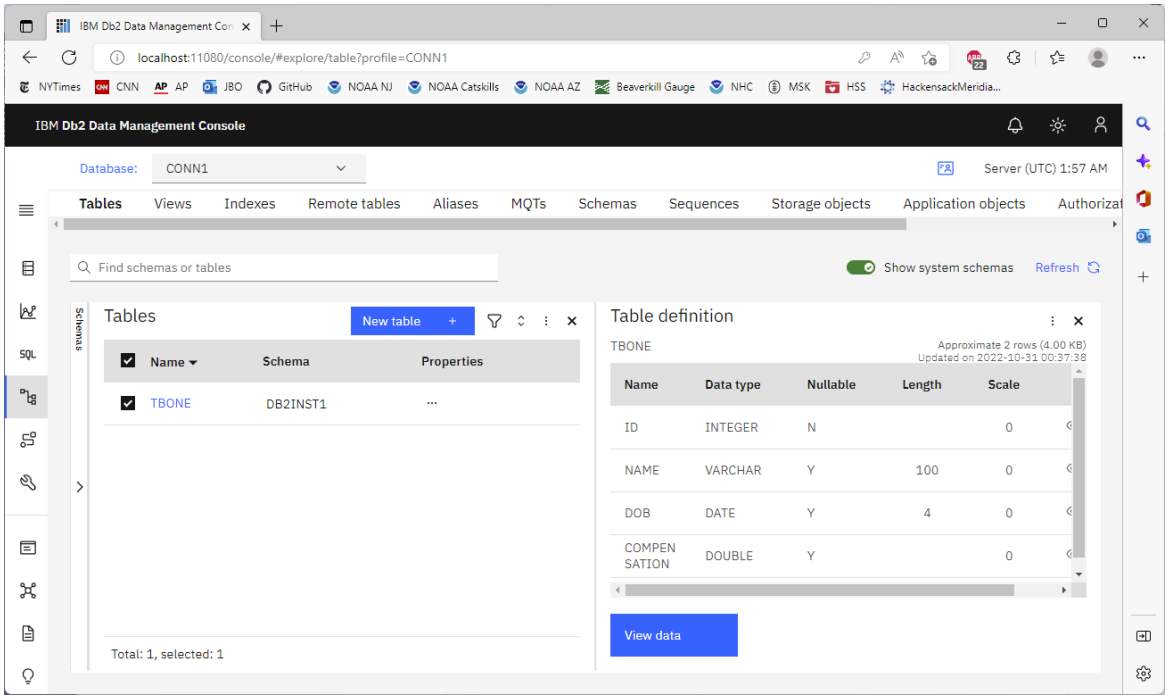
- Install the IBM Db2 Data Management Console and use it to:
  - Configure the DB2 repository. Use the Console to create the textdb DB2 database used for the examples in this document.
  - Create a connection profile with the TESTDB database
  - Create the tbOne table in the TESTDB database by executing an SQL statement in the console:

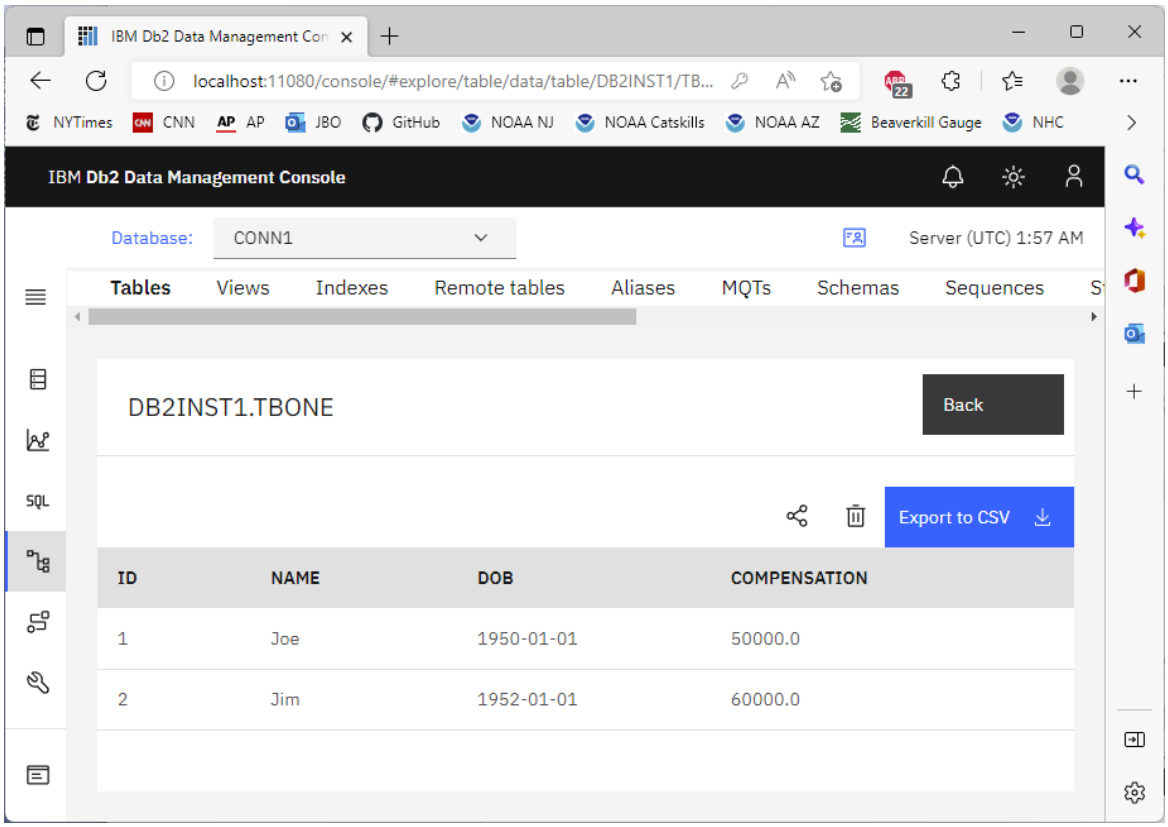
```
CREATE TABLE tbOne (Id Int Not Null Primary Key, Name VARCHAR(100), DOB Date, Compensation Float);
```

  - Enter the sample data into the tbOne table by executing SQL statements in the console:

```
INSERT INTO tbOne (Id, Name, DOB, Compensation) VALUES(1,'Joe', '1950-01-01',50000);
INSERT INTO tbOne (Id, Name, DOB, Compensation) VALUES(2,'Jim', '1952-01-01',60000);
```







A sample stored procedure may be added, if desired.

□SQLDB2 Actions

sqldb2 actions operate on either the □sqldb2 object or an □sqldb2 instance. The □sqldb2 object is a container for all □sqldb2 instances in an APL64 instance. An □sqldb2 instance is associated with a specific Sqldb2 database accessible to the target workstation user.

The left argument of the APL64 □sqldb2 system function determines if the □sqldb2 action applies to the □sqldb2 object ('#' left argument) or □sqldb2 instance (name of □sqldb2 instance).

The name of an □sqldb2 instance can be specified by:

- An APL64 text expression, e.g. 'myinstance', «myinstance»
- An APL64 variable with value equal to an □sqldb2 instance name
- Elided, if the □Sqldb2Self value is an □sqldb2 instance name

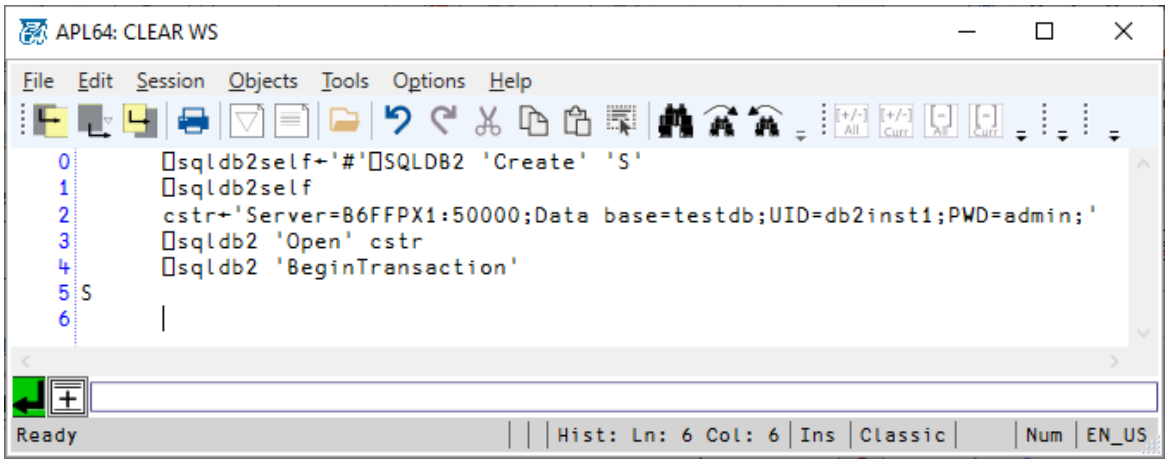
□sqldb2 action names are not case sensitive. □sqldb2 instance names are case sensitive.

BeginTransaction

This action applies to an ⎕sqldb2 instance. The BeginTransaction action, for the Sqldb2 instance named in the left argument, indicates that subsequent ⎕sqldb2 actions will be incorporated into an SQL transaction. An SQL transaction may contain multiple ⎕sqldb2 actions. An SQL transaction can either be committed or rolled back. The BeginTransaction action has no result.

SQL transactions are used to combine multiple SQL statements, possibly affecting multiple tables in an SQL database, so that they are either all performed or none are performed.

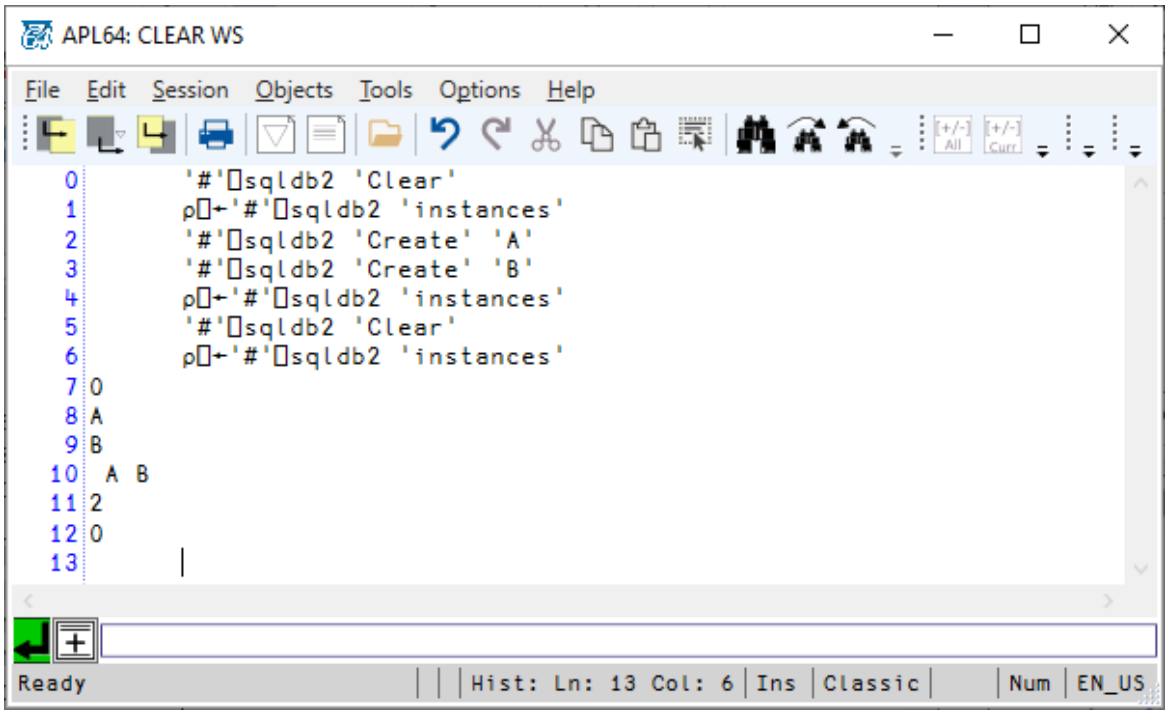
```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕sqldb2 'BeginTransaction'
```



Clear

This action is performed on the ⎕sqldb2 object. The Clear action closes the Sqldb2 database connection associated with any ⎕sqldb2 instance and deletes all ⎕sqldb2 instances in the current APL64 instance. The Clear action has no result.

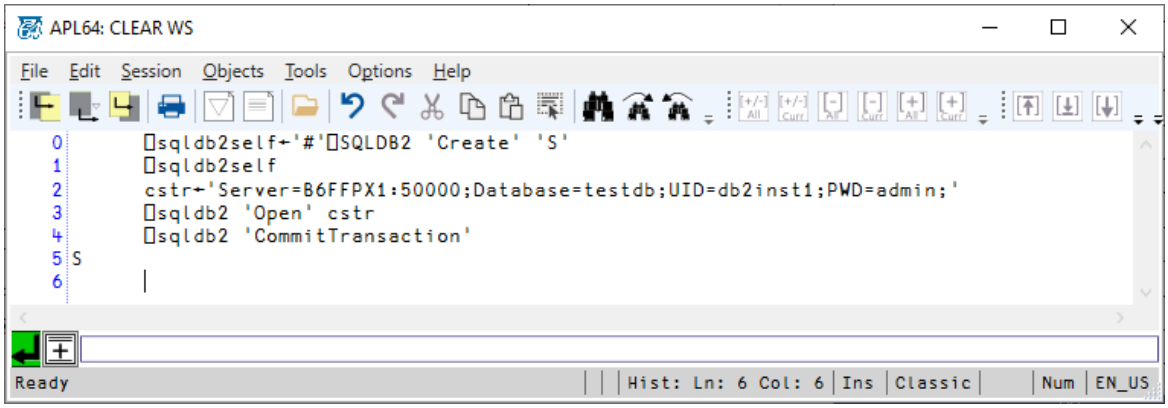
```
'#'⎕sqldb2 'Clear'
ρ⎕←'#'⎕sqldb2                    'instances'
'#'⎕sqldb2 'Create' 'A'
'#'⎕sqldb2 'Create' 'B'
ρ⎕←'#'⎕sqldb2                    'instances'
'#'⎕sqldb2 'Clear'
ρ⎕←'#'⎕sqldb2 'instances'
```



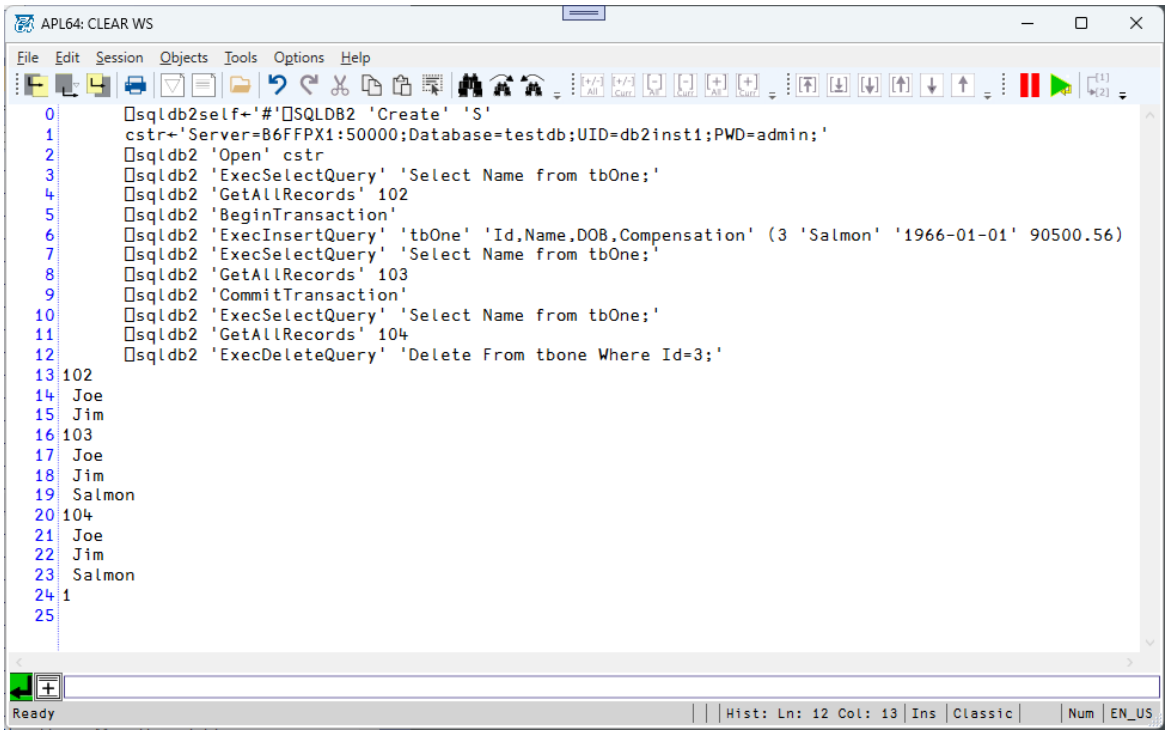
CommitTransaction

This action applies to an ⎕sqldb2 instance. The CommitTransaction action causes the effects of the current transaction to be applied to the connected Sqldb2 database. The CommitTransaction action has no effect if the BeginTransaction action has not occurred or if the RollbackTransaction action has occurred. Prior to committal or rollback of a transaction, the effects of the existing SQL statements in the transaction are pending effects in the Sqldb2 database. The CommitTransaction action has no result.

```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕sqldb2 'CommitTransaction'
```



```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕sqldb2 'ExecSelectQuery' 'Select Name from tbOne;'
⎕sqldb2 'GetAllRecords' 102
⎕sqldb2 'BeginTransaction'
⎕sqldb2 'ExecInsertQuery' 'tbOne' 'Id,Name,DOB,Compensation' (3 'Salmon' '1966-01-01' 90500.56)
⎕sqldb2 'ExecSelectQuery' 'Select Name from tbOne;'
⎕sqldb2 'GetAllRecords' 103
⎕sqldb2 'CommitTransaction'
⎕sqldb2 'ExecSelectQuery' 'Select Name from tbOne;'
⎕sqldb2 'GetAllRecords' 104
⎕sqldb2 'ExecDeleteQuery' 'Delete From tbone Where Id=3;'
```



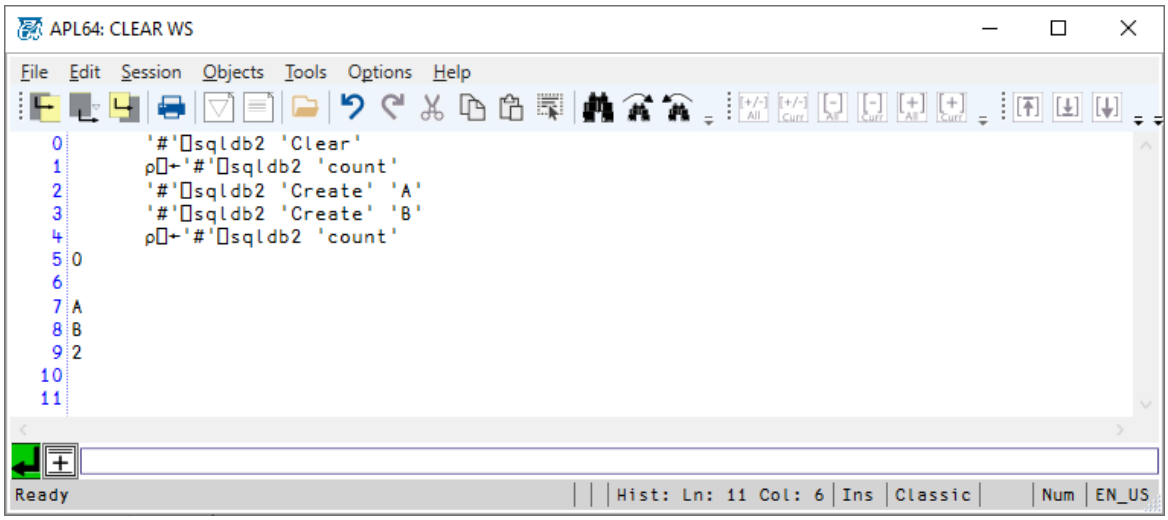
Note that the records in the #103 data set include the pending effects of the ExecInsertQuery action and the records in the #104 data set include the effects of the committed ExecInsertQuery action:

Count

This action is performed on the ⎕sqldb2 object. The Count action returns the number of ⎕sqldb2 instances which exist in the APL64 instance.

```
'#'⎕sqldb2 'Clear'
```

```
ρ⎕←'#'⎕sqldb2 'count'
'#'⎕sqldb2 'Create' 'A'
'#'⎕sqldb2 'Create' 'B'
ρ⎕←'#'⎕sqldb2 'count'
```

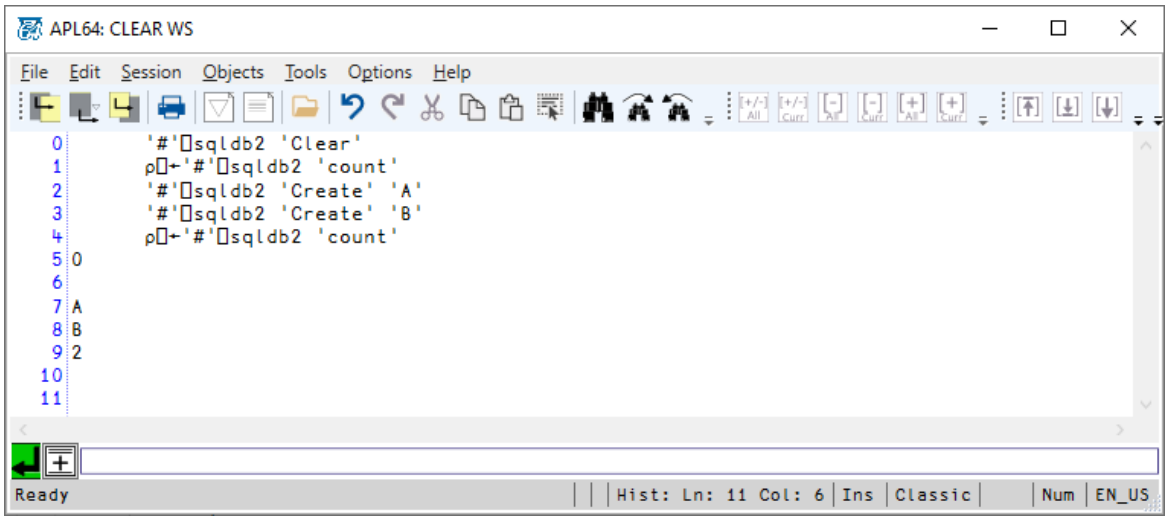


Create

This action is performed on the ⎕sqldb2 object. The Create action will create an ⎕Sqldb2 instance with a user-provided name in the right argument. Multiple ⎕Sqldb2 instances are possible in the same APL64 instance, so that multiple Sqldb2 databases may be conveniently accessed.

The Create action will not fail if the named instance already exists. The Create action will close and delete a pre-existing instance with the same name. The Create action for a particular Sqldb2 database is generally used once in an APL64 instance. The Create action does not open a connection to an Sqldb2 database, use the Open action for that purpose. The result of a successful 'Create' action is a text vector containing the ⎕Sqldb2 instance name.

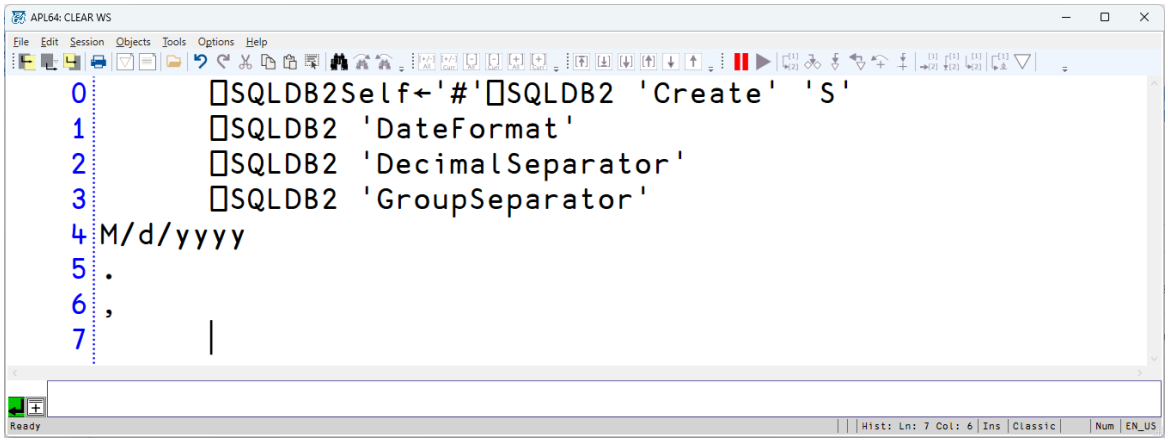
```
'#'⎕sqldb2 'Clear'
ρ⎕←'#'⎕sqldb2 'count'
'#'⎕sqldb2 'Create' 'A'
'#'⎕sqldb2 'Create' 'B'
ρ⎕←'#'⎕sqldb2 'count'
```



DateFormat

This action returns the current .Net Short Date Pattern text applicable to the specified ⎕SQLDB2 instance.

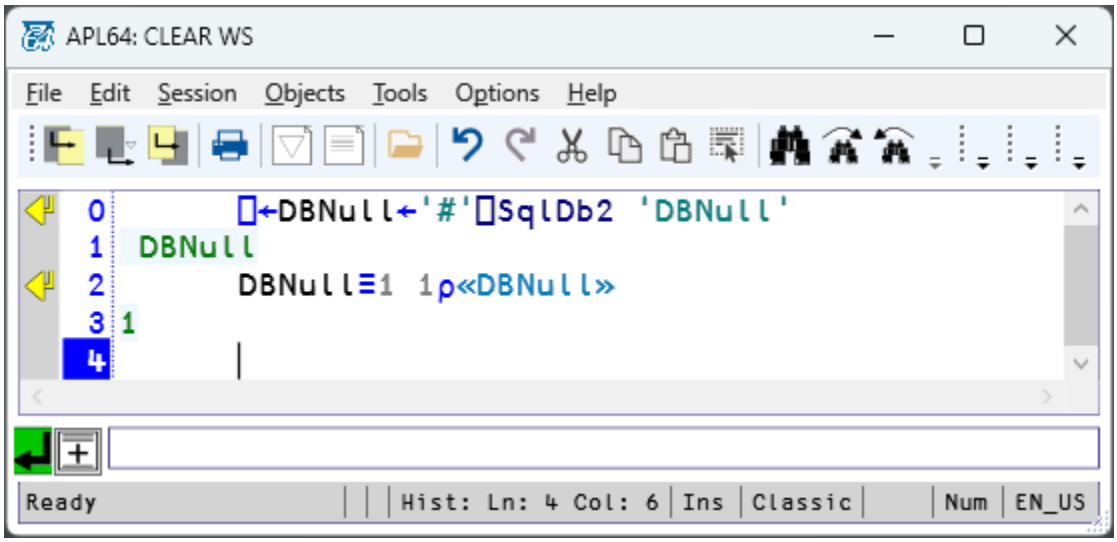
```
⎕SQLDB2Self←'#'⎕SQLDB2 'Create' 'S'
⎕SQLDB2 'DateFormat'
⎕SQLDB2 'DecimalSeparator'
⎕SQLDB2 'GroupSeparator'
```



DBNull

This action returns a value which may be used to specify null values for the ExecInsertQuery and ExecStoredProc ⎕SQLDb2 instance actions. This value will also be used as the APL64 value associated with a DBNull value returned from a GetRecord, GetAllRecords, or ExecStoredProc action.

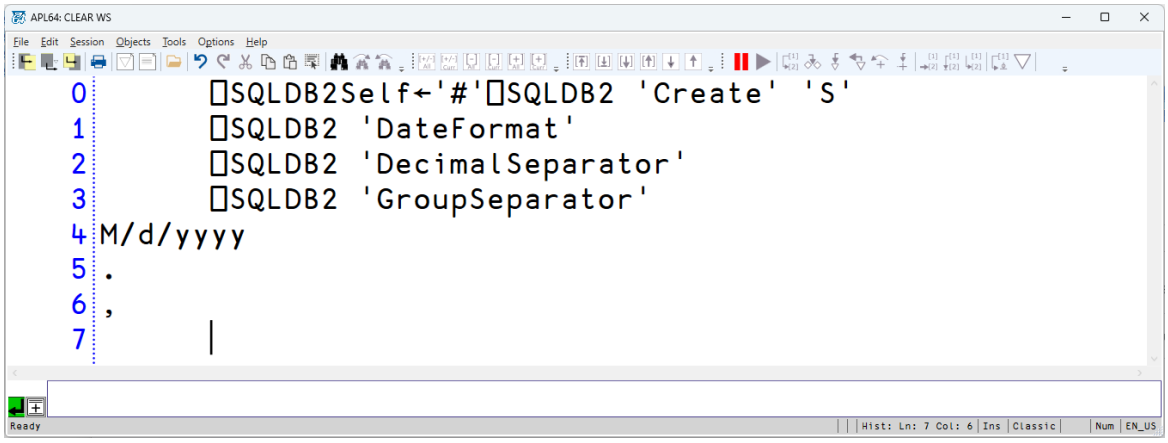
```
⎕←DBNull←'#'⎕SqlDb2 'DBNull'
DBNull≡1 1⍴«DBNull»
```



DecimalSeparator

This action returns the current .Net Number Decimal Separator text applicable to the specified ⎕SQLDB2 instance.

```
⎕SQLDB2Self←'#'⎕SQLDB2 'Create' 'S'
⎕SQLDB2 'DateFormat'
⎕SQLDB2 'DecimalSeparator'
⎕SQLDB2 'GroupSeparator'⎕DR ⎕←s ⎕SQLDB2 'DecimalSeparator'
```

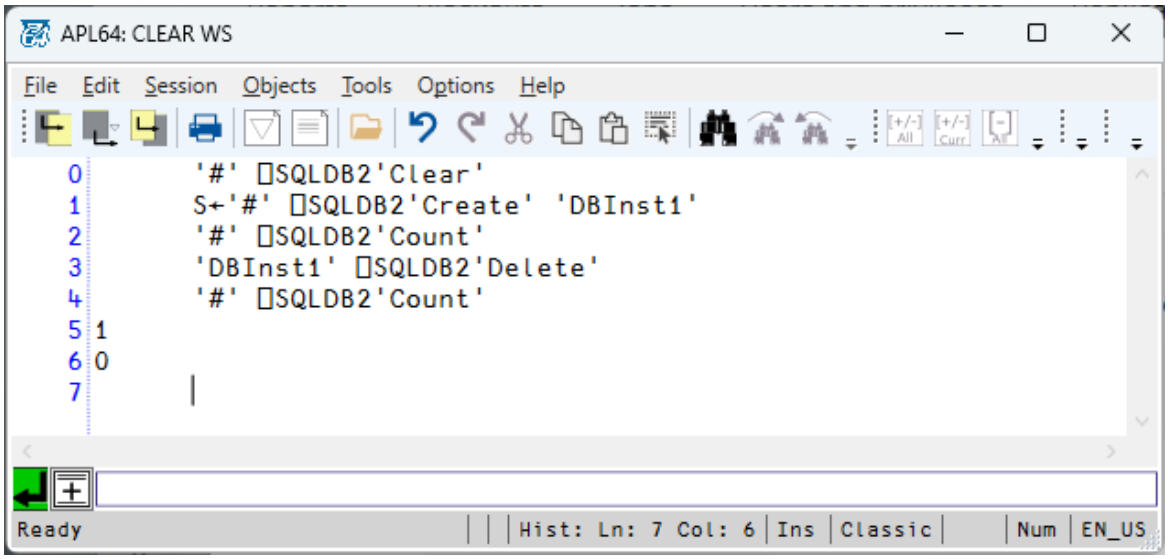


Delete

The Delete action is performed on an ⎕SQLDB2 instance. The close action will close an existing connection and delete the associated ⎕SQLDB2 instance.

```
'#' ⎕SQLDB2'Clear'
S←'#' ⎕SQLDB2'Create' 'DBInst1'
 '#' ⎕SQLDB2'Count'
```

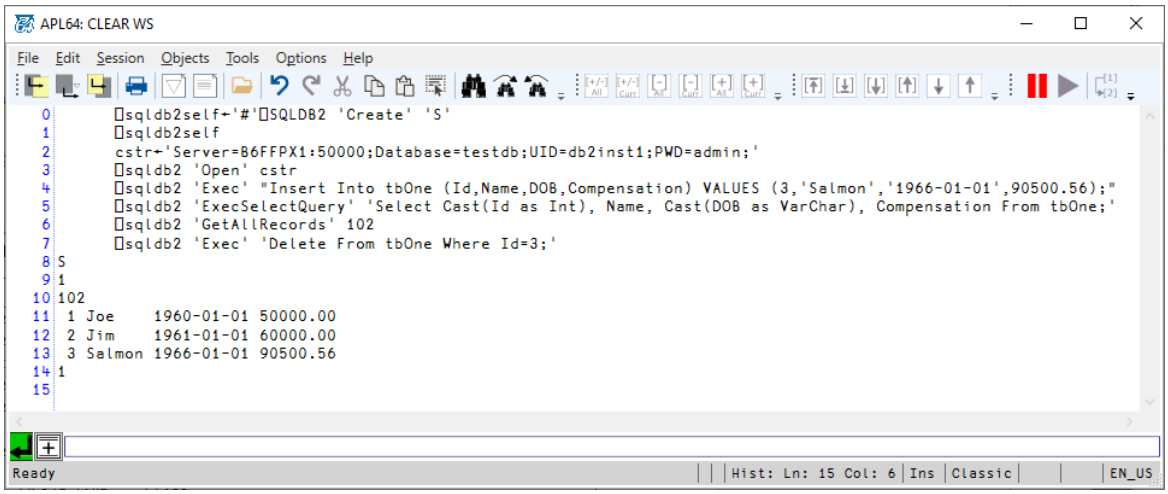
```
'DBInst1' ⎕SQLDB2'Delete'
'#' ⎕SQLDB2'Count'
```



Exec

This action applies to an ⎕sqldb2 instance. The ⎕sqldb2 Exec action returns the number of database rows, if any, affected by the executed SQL statement provided by the right argument. The ⎕sqldb2 Exec action cannot be used to return results of a Select SQL statement.

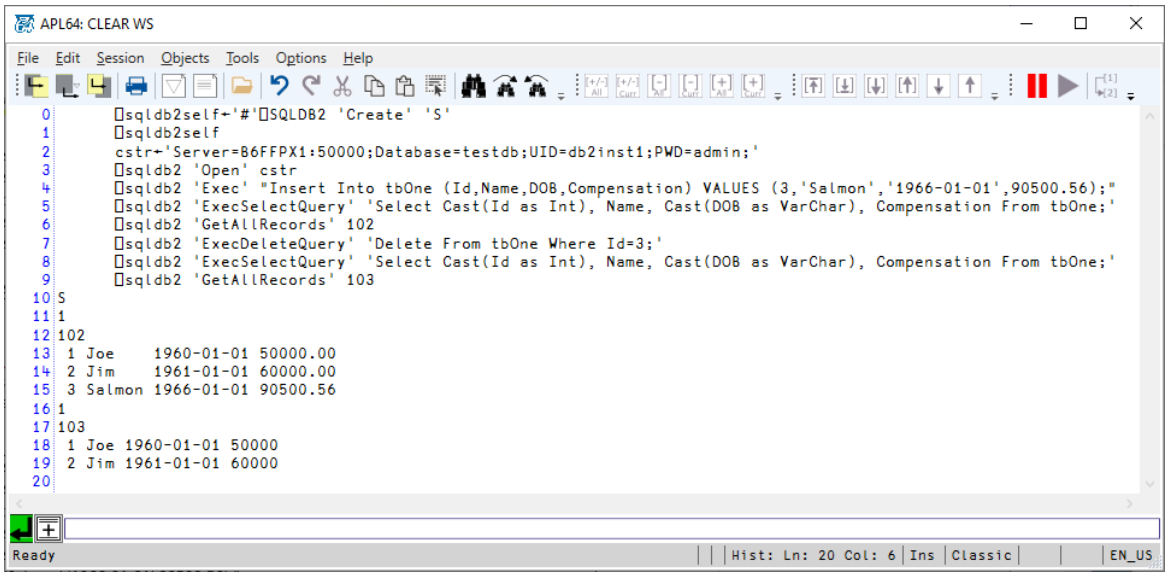
```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕sqldb2 'Exec' "Insert Into tbOne (Id,Name,DOB,Compensation) VALUES (3,'Salmon','1966-01-01',90500.56);"
⎕sqldb2 'ExecSelectQuery' 'Select Cast(Id as Int), Name, Cast(DOB as VarChar), Compensation From tbOne;'
⎕sqldb2 'GetAllRecords' 102
⎕sqldb2 'Exec' 'Delete From tbOne Where Id=3;'
```



ExecDeleteQuery

This action applies to an ⎕sqldb2 instance. The right argument is the text of the SQL Delete statement. The result of the ⎕sqldb2 ExecDeleteQuery is the number of records affected, if any.

```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕sqldb2 'Exec' "Insert Into tbOne (Id,Name,DOB,Compensation) VALUES (3,'Salmon','1966-01-01',90500.56);"
⎕sqldb2 'ExecSelectQuery' 'Select Cast(Id as Int), Name, Cast(DOB as VarChar), Compensation From tbOne;'
⎕sqldb2 'GetAllRecords' 102
⎕sqldb2 'ExecDeleteQuery' 'Delete From tbOne Where Id=3;'
⎕sqldb2 'ExecSelectQuery' 'Select Cast(Id as Int), Name, Cast(DOB as VarChar), Compensation From tbOne;'
⎕sqldb2 'GetAllRecords' 103
```



ExecInsertQuery

This action applies to an ⎕sqldb2 instance.

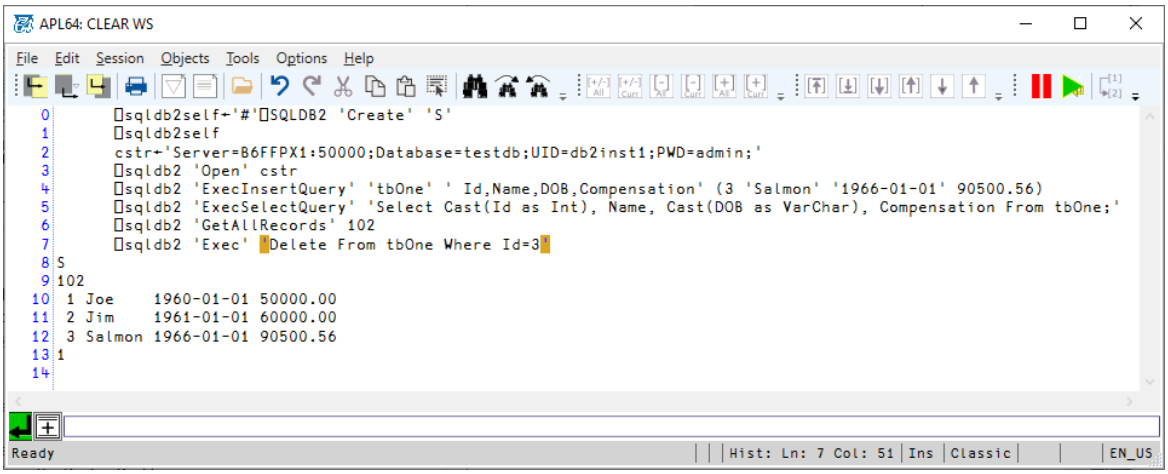
Right arguments:

1. Sqldb2 table name
2. Sqldb2 column names: Comma delimited text with selected Sqldb2 table column names, or ' ' or '*' to indicate all Sqldb2 table column names.
3. APL64 data: A matrix containing the APL64 values to be inserted. One row for each record to be inserted.

The order of the Sqldb2 column names and APL64 data matrix columns must be conformable.

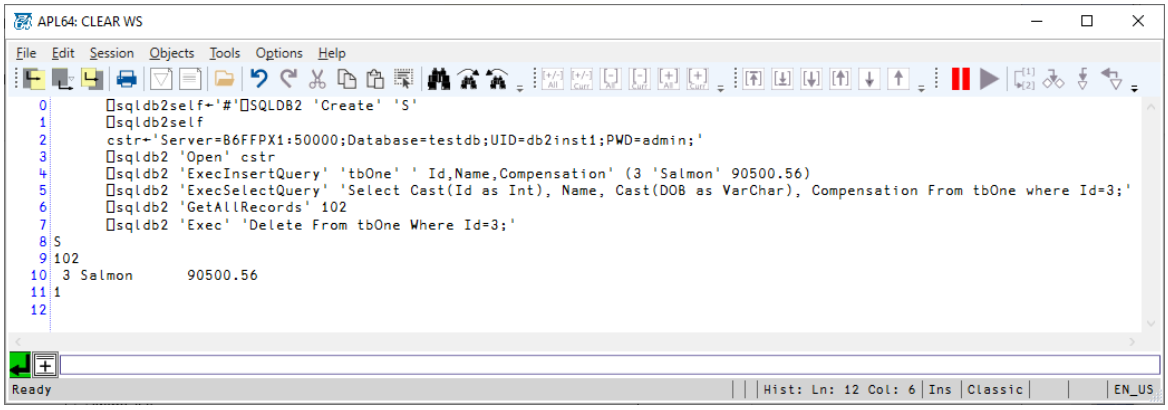
Specify all columns by name:

```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕sqldb2 'ExecInsertQuery' 'tbOne' ' Id,Name,DOB,Compensation' (3 'Salmon' '1966-01-01' 90500.56)
⎕sqldb2 'ExecSelectQuery' 'Select Cast(Id as Int), Name, Cast(DOB as VarChar), Compensation From tbOne;'
⎕sqldb2 'GetAllRecords' 102
⎕sqldb2 'Exec' 'Delete From tbOne Where Id=3;'
```



If only certain (nullable) column values are to be inserted, the required values for the non-nullable columns must also be inserted. Database null values are converted to APL64: 0 0ρ0:

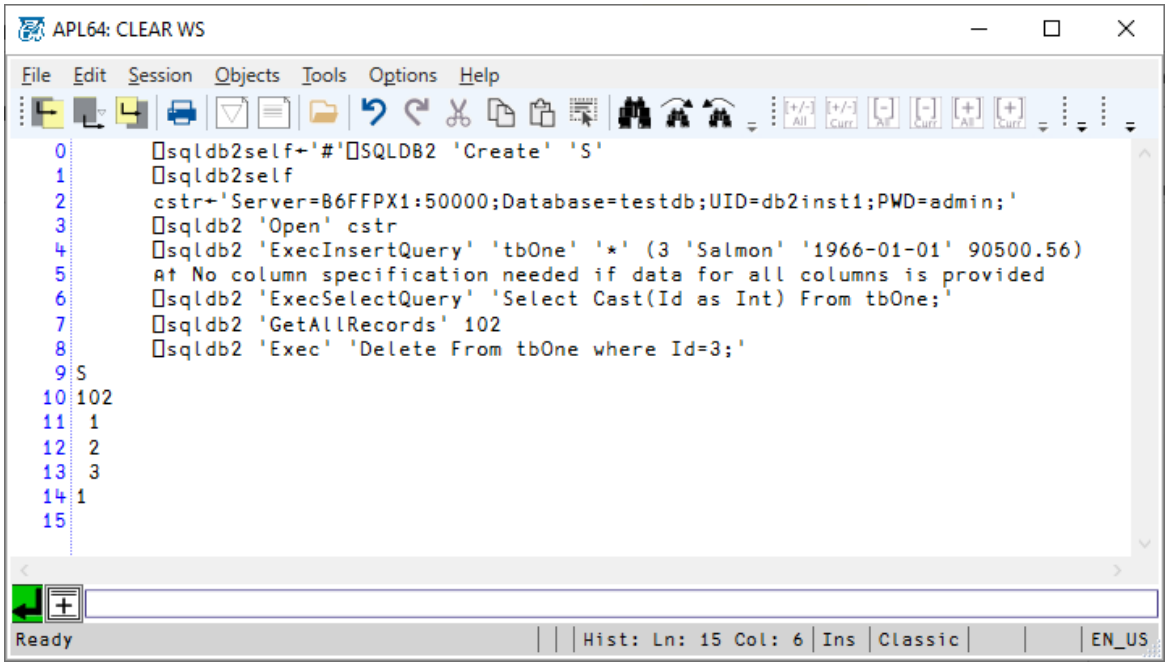
```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕sqldb2 'ExecInsertQuery' 'tbOne' ' Id,Name,Compensation' (3 'Salmon' 90500.56)
⎕sqldb2 'ExecSelectQuery' 'Select Cast(Id as Int), Name, Cast(DOB as VarChar), Compensation From tbOne where Id=3;'
⎕sqldb2 'GetAllRecords' 102
⎕sqldb2 'Exec' 'Delete From tbOne Where Id=3;'
```



Specify all columns to be inserted with '*':

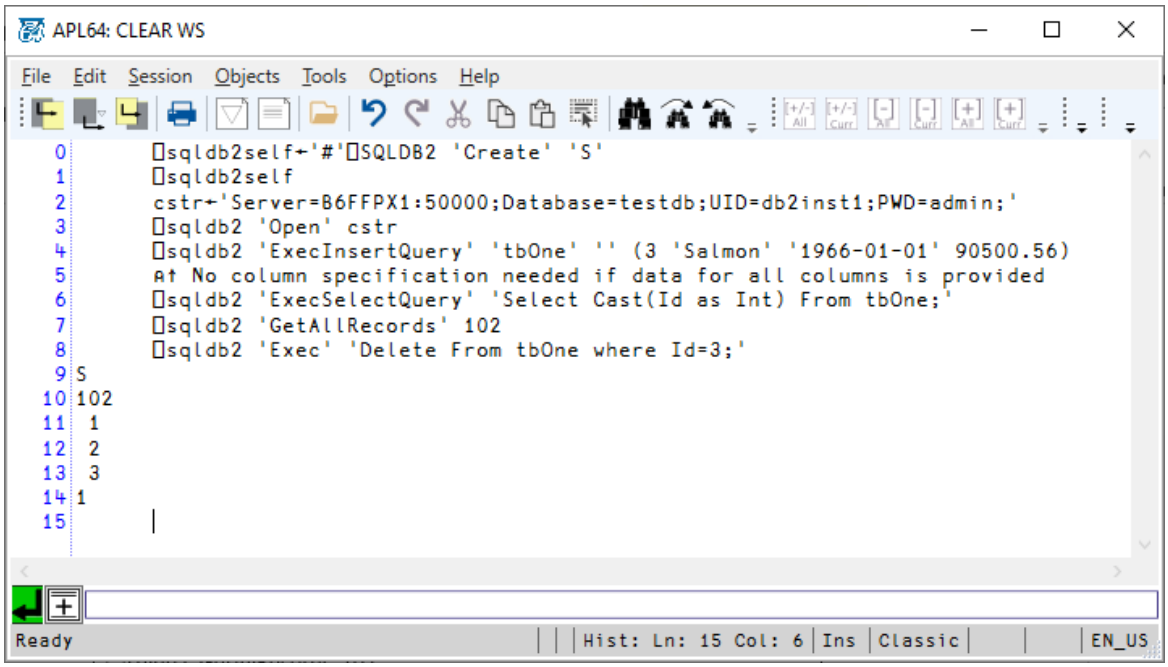
```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕sqldb2 'ExecInsertQuery' 'tbOne' '*' (3 'Salmon' '1966-01-01' 90500.56)
⍝↑ No column specification needed if data for all columns is provided
```

```
sqldb2 'ExecSelectQuery' 'Select Cast(Id as Int) From tbOne;'
sqldb2 'GetAllRecords' 102
sqldb2 'Exec' 'Delete From tbOne where Id=3;'
```



No columns ('') need to be specified for the ExecInsertQuery, if the data to be inserted includes all columns:

```
sqldb2self←'#'SQLDB2 'Create' 'S'
sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
sqldb2 'Open' cstr
sqldb2 'ExecInsertQuery' 'tbOne' '' (3 'Salmon' '1966-01-01' 90500.56)
⍝↑ No column specification needed if data for all columns is provided
sqldb2 'ExecSelectQuery' 'Select Cast(Id as Int) From tbOne;'
sqldb2 'GetAllRecords' 102
sqldb2 'Exec' 'Delete From tbOne where Id=3;'
```



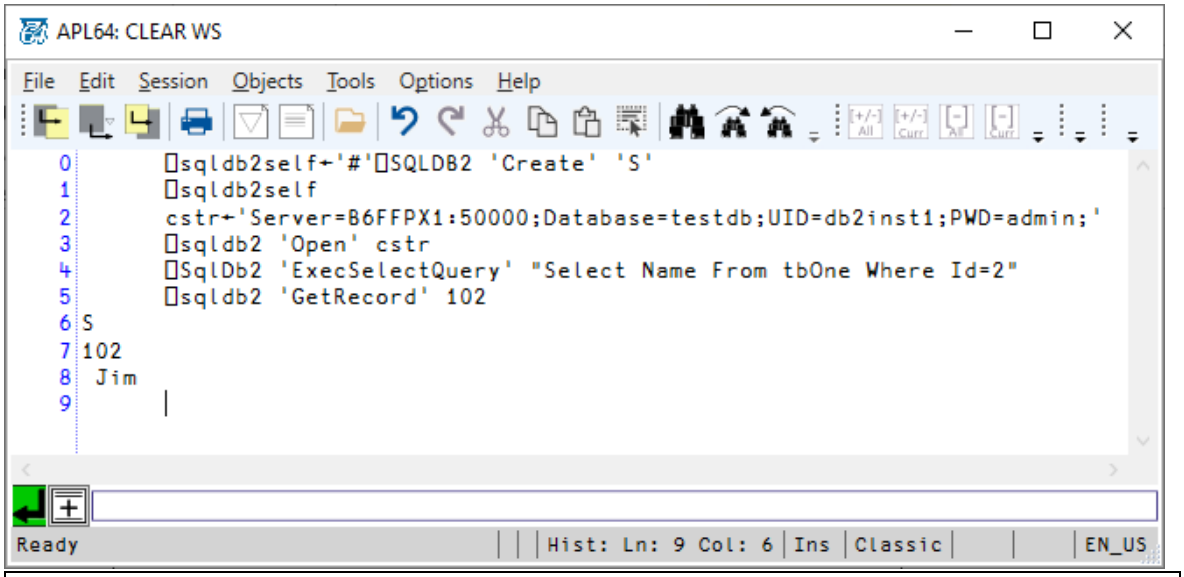
ExecSelectQuery

The ExecSelectQuery action applies to an ⎕SQLDB instance. The action argument provides the SQL Select statement. Depending on the SqlDb data type of a column selected, it may be necessary to cast that data column values to a data type which has a representation in APL64.

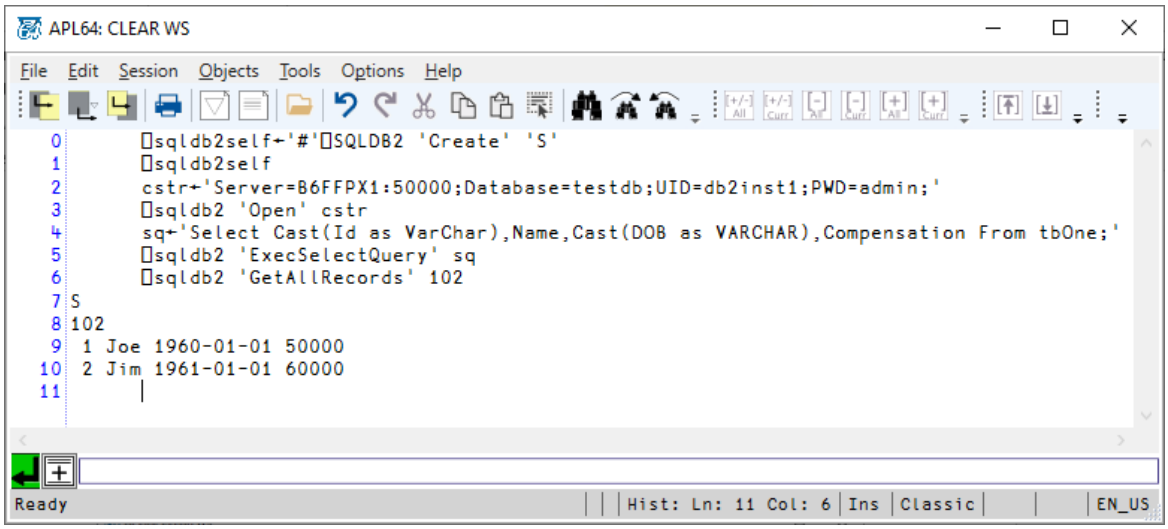
The ExecSelectQuery does not return the result of the SQL Select statement. The result is an integer indicating the record set containing the result of the SQL Select statement. Use the 'GetRecord' or 'GetAllRecords' actions to access the specified record set.

The SQL-format query text argument to the ⎕SQLDB ExecSelectQuery method, should yield only on record set. A query of the form 'query1;…;queryN', is not supported in APL64 and must be separated into individual queries. As an alternative to separate queries, a stored procedure can be created and executed using the ⎕SQLDB ExecStoredProc method.

```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕SqlDb2 'ExecSelectQuery' "Select Name From tbOne Where Id=2"
⎕sqldb2 'GetRecord' 102
```




```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
sq←'Select Cast(Id as VarChar),Name,Cast(DOB as VARCHAR),Compensation From
tbOne;'
⎕sqldb2 'ExecSelectQuery' sq
⎕sqldb2 'GetAllRecords' 102
```



ExecStoredProc

This action can be used to run a stored procedure which is input, output and input/output parameters.

Syntax:
MatrixOfValues←instanceName⎕SqlDb2 'ExecStoredProc' procName paramNames dataVector

The action arguments are:

- An APL text vector (procName) containing the stored procedure name
- Comma-delimited APL text (paramNames) containing the stored procedure parameter names. Prefixed and suffixed space characters will be trimmed from each parameter name.
- An APL rank-1 vector (dataVector) of 'In' and 'InOut' parameter values. The order of the parameter values and the order of the 'In' and 'InOut' parameter names must be conformable. There should be no 'place holder' element in the dataVector for an Out parameter.

The parameter specifications, direction(In, InOut, Out) and data type, are determined by APL64 by querying the applicable database system tables.

Not all IBM Db2 data types can be represented in APL64, so the data types of stored procedure parameters must be carefully considered. For IBM Db2 data types which have no representation in APL64, the stored procedure can cast the parameter values to APL64-compatible data types.

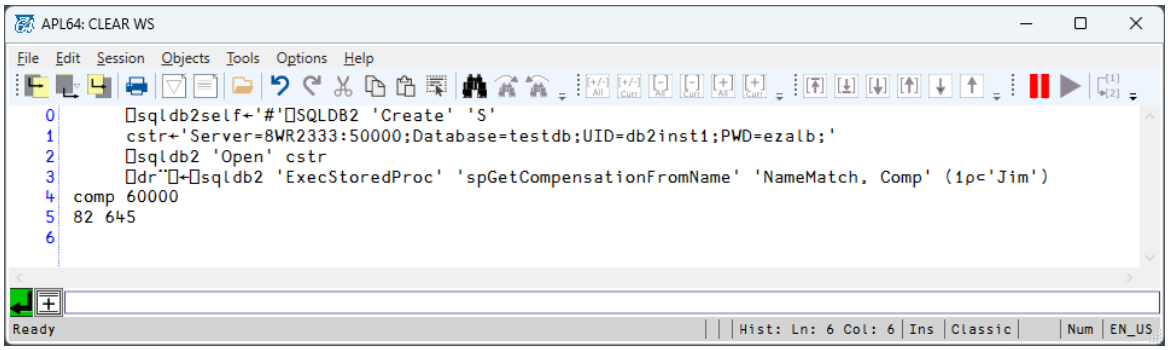
The APL64-compatible data types are Boolean, Double, Int32, Char, Varchar and String.

The result is a 2-column APL64 matrix with one row for each out and inout parameter:

- Column #1: Parameter name
- Column #2: Parameter value

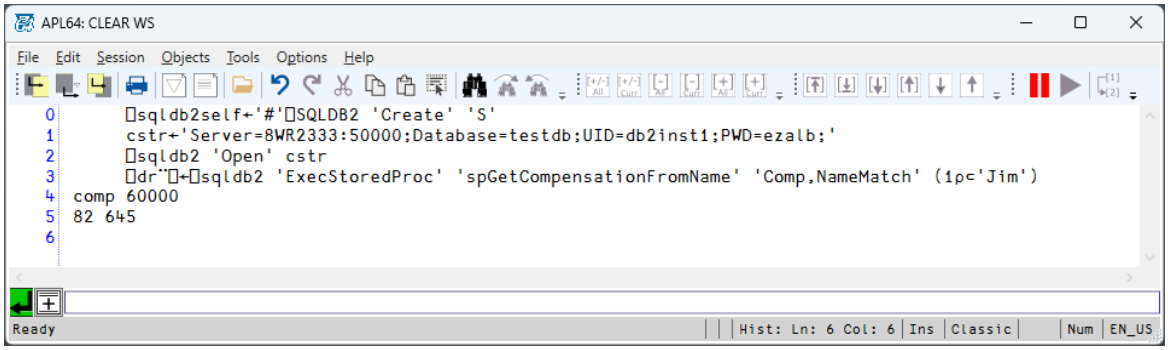
In and Out parameters with the In parameter specified before the Out parameter.:

```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
cstr←'Server=8WR2333:50000;Database=testdb;UID=db2inst1;PWD=ezalb;'
⎕sqldb2 'Open' cstr
⎕dr¨⎕←⎕sqldb2 'ExecStoredProc' 'spGetCompensationFromName' 'NameMatch, Comp' (1⍴⊂'Jim')
```



Different order of fieldNames and associated dataVector values:

```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
cstr←'Server=8WR2333:50000;Database=testdb;UID=db2inst1;PWD=ezalb;'
⎕sqldb2 'Open' cstr
⎕dr¨⎕←⎕sqldb2 'ExecStoredProc' 'spGetCompensationFromName'
'Comp,NameMatch' (1⍴⊂'Jim')
```



The stored procedure definition:

SPGETCOMPENSATIONFROMNAME

```
CREATE OR REPLACE PROCEDURE DB2INST1.spGetCompensationFromName
(IN NameMatch VARCHAR(100), OUT COMP DOUBLE)
LANGUAGE SQL
P1:
BEGIN
SELECT Compensation INTO COMP FROM DB2INST1.TBONE WHERE Name=NameMatc
h;
END P1
```

ExecStoredProcCmd

This action can be used to run a stored procedure which has no input or output parameters. The single action argument is an APL text vector (cmdText) containing the stored procedure name. The result is an empty, 2-column APL matrix.

```
'#'⎕SqlDb2 'Clear'
⎕sqldb2self←'#'⎕SqlDb2 'Create' 'S1'
cstr←'Server=8WR2333:50000;Database=testdb;UID=db2inst1;PWD=ezalb;'
⎕SqlDb2 'Open' cstr
ρ⎕←⎕SqlDb2 'ExecStoredProcCmd' 'StoredProc1'
⍝↑ Inserts a new record for 'Salmon' in the tbone table
⎕SqlDb2 'ExecSelectQuery' "Select Name From tbone Where Id=3"
⎕SqlDb2 'GetRecord' 102
⎕SqlDb2 'Open' cstr
⎕SqlDb2 'Exec' 'Delete From tbone Where Id=3;'
```

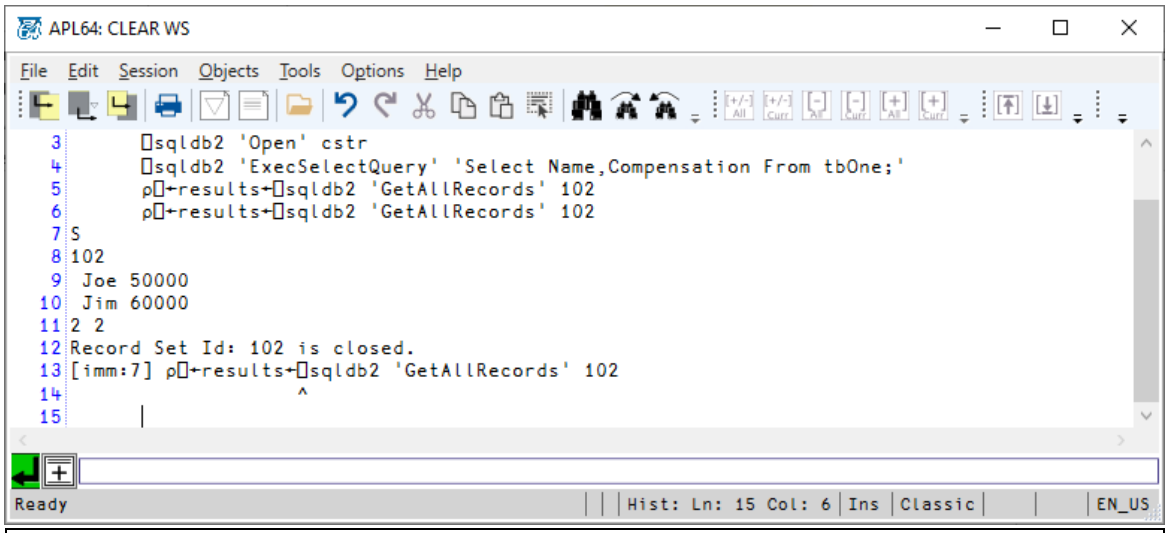
GetAllRecords

This action applies to an ⎕sqldb2 instance. The 'GetAllRecords' action will get the remaining records which have not already been read from the specified record set. This action closes the specified record set, because all records have been read.

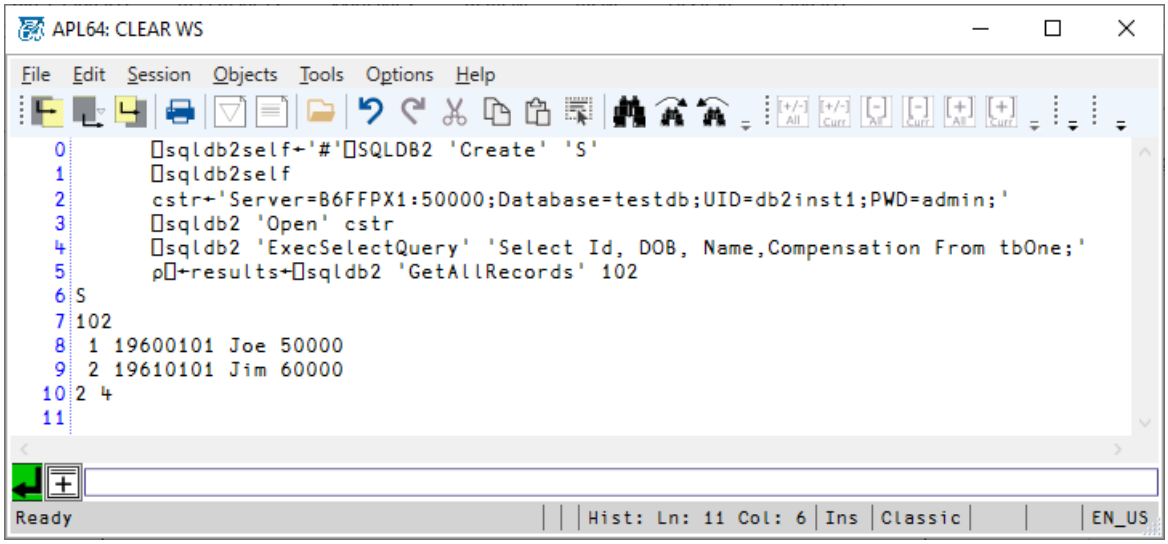
An exception may occur if the data in the record set has no representation in APL64. In this case the query which created the record set must be modified, e.g. Cast(), to make the returned data conformable with APL64.

An exception will occur if there are no remaining records to be read.

```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕sqldb2 'ExecSelectQuery' 'Select Name,Compensation From tbOne;'
ρ⎕←results←⎕sqldb2 'GetAllRecords' 102
ρ⎕←results←⎕sqldb2 'GetAllRecords' 102
```




```
sqldb2self←'#'SQLDB2 'Create' 'S'
sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
sqldb2 'Open' cstr
sqldb2 'ExecSelectQuery' 'Select Id, DOB, Name,Compensation From tbOne;'
ρ←results←sqldb2 'GetAllRecords' 102
```

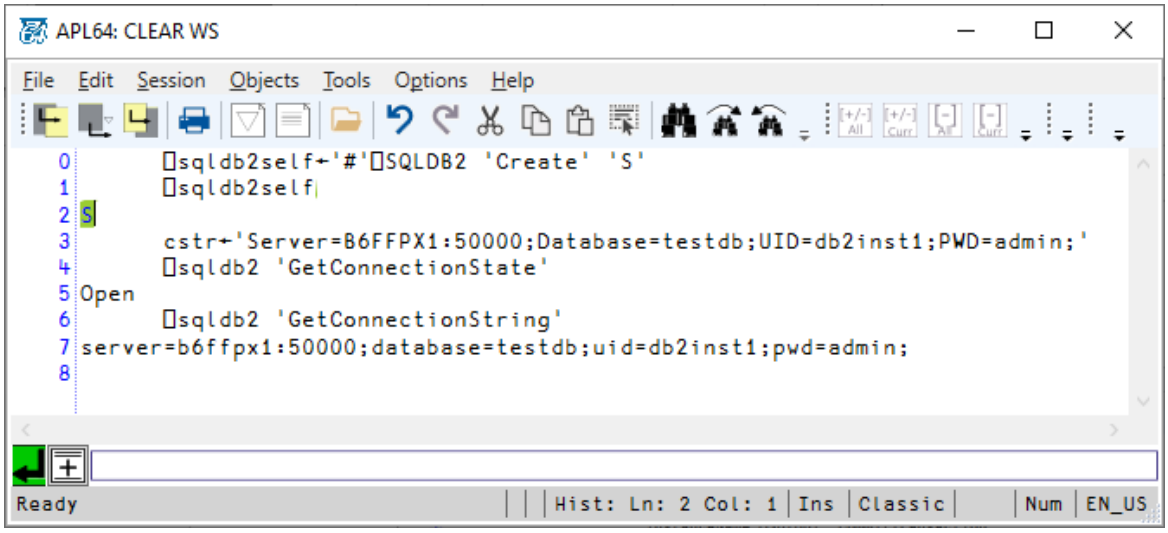


GetConnectionState

This action applies to an sqldb2 instance. The result of the GetConnectionState action is a text vector indicating the connection state of the specified Sqldb2 instance to an Sqldb2 database.

```
sqldb2self←'#'SQLDB2 'Create' 'S'
sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
```

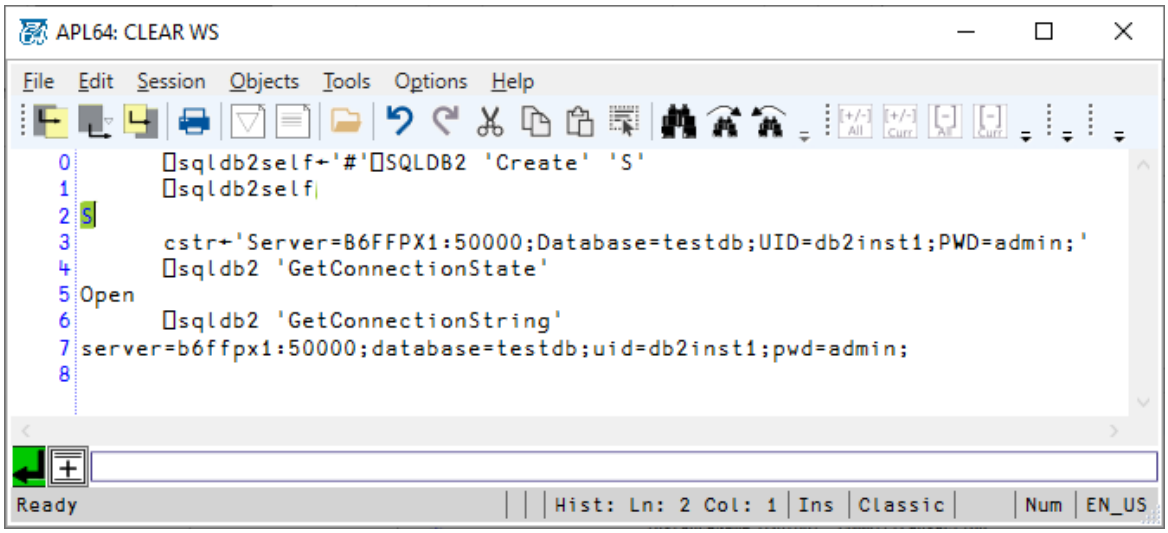
```
⎕sqldb2 'Open' cstr
⎕sqldb2 'GetConnectionState'
⎕sqldb2 'GetConnectionString'
```



GetConnectionString

This action applies to an ⎕sqldb2 instance. The result of the GetConnectionString is the Sqldb2 connection string, if any, associated with the specified ⎕Sqldb2 instance. Use the Open action to set the connection string.

```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕sqldb2 'GetConnectionState'
⎕sqldb2 'GetConnectionString'
```

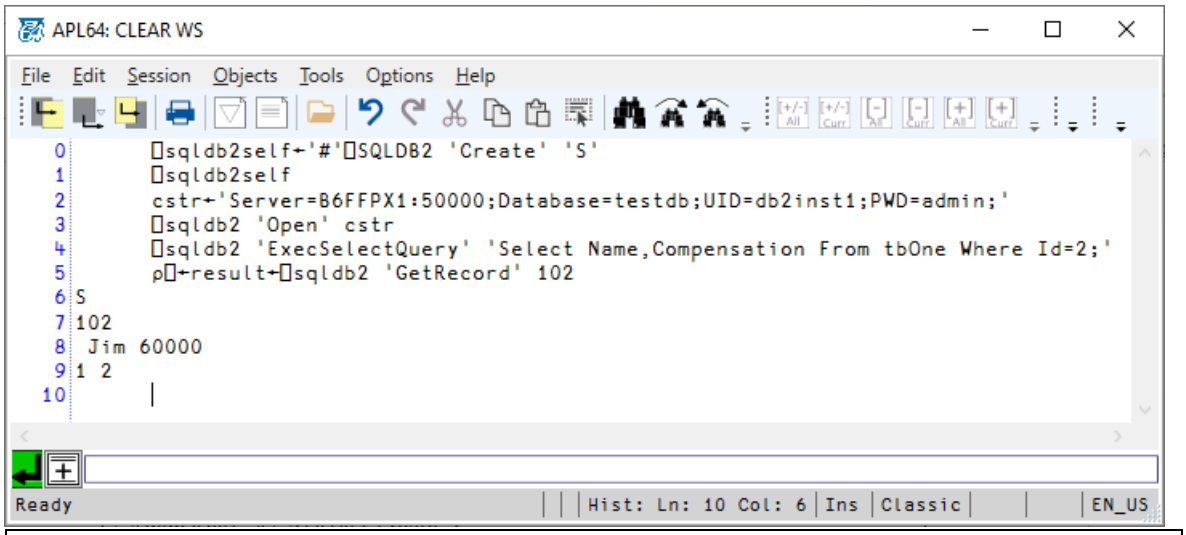


GetRecord

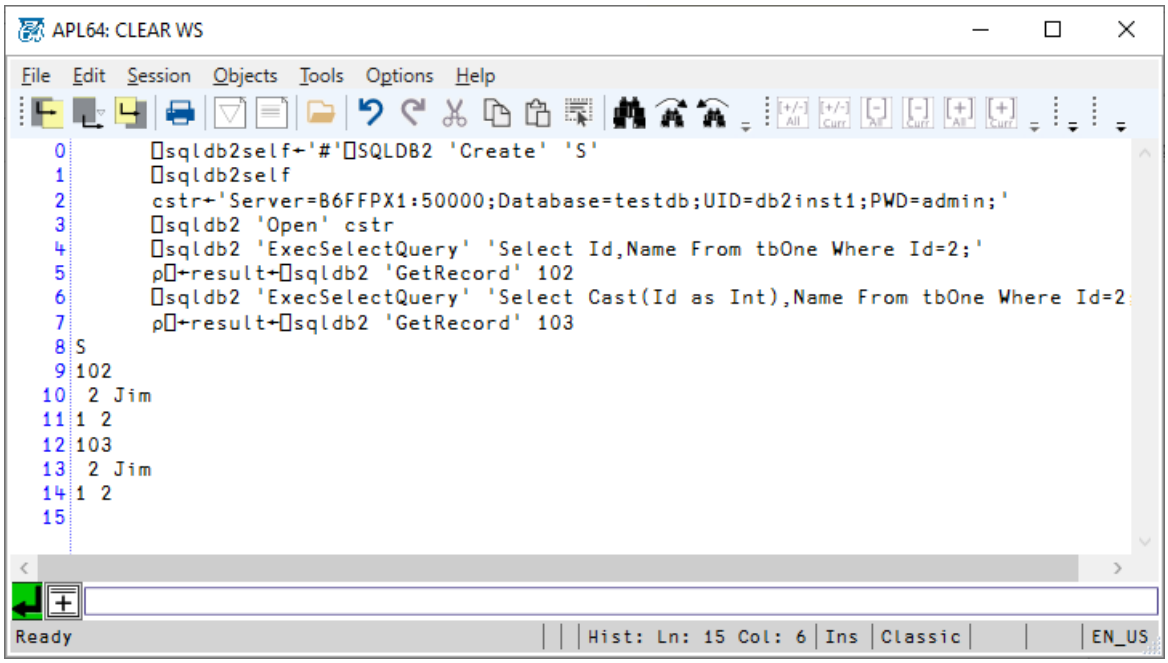
This action applies to an ⎕sqldb2 instance. The right argument is the ⎕sqldb2 record set number. The GetRecord action will read the next available record, if any, from the specified record set and return its values to the APL64 instance.

An exception may occur if the data in the record set has no representation in APL64. In this case, the query which created the record set must be modified to make the returned data conformable with APL64. This exception does not occur for the sample table used in this document, because all column types are conformable as APL64 data types. If such an exception does occur, the Cast() method of the database can generally convert the source data in the database record to an APL64 data type.

```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕sqldb2 'ExecSelectQuery' 'Select Name,Compensation From tbOne Where Id=2;'
⍴⎕←result←⎕sqldb2 'GetRecord' 102
```



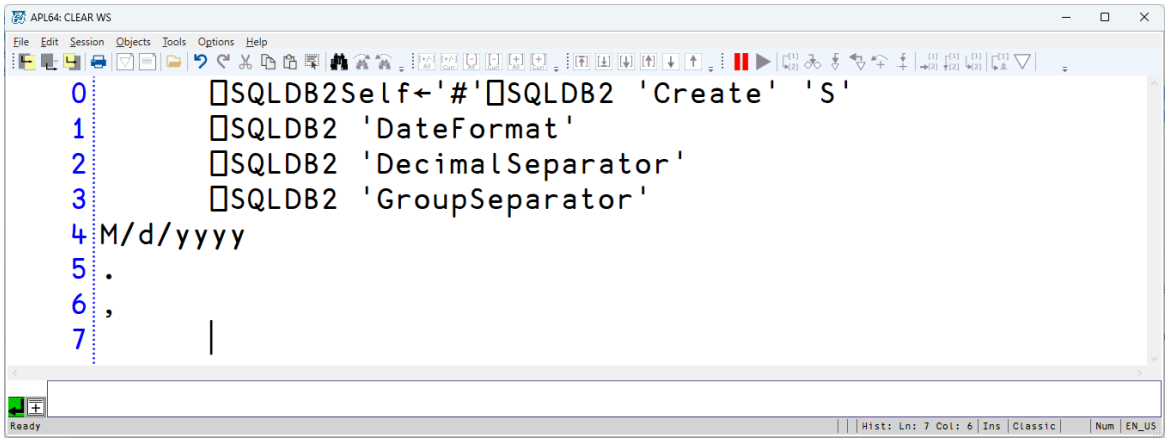

```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕sqldb2 'ExecSelectQuery' 'Select Id,Name From tbOne Where Id=2;'
ρ⎕←result←⎕sqldb2 'GetRecord' 102
⎕sqldb2 'ExecSelectQuery' 'Select Cast(Id as Int),Name From tbOne Where Id=2;'
ρ⎕←result←⎕sqldb2 'GetRecord' 103
```



GroupSeparator

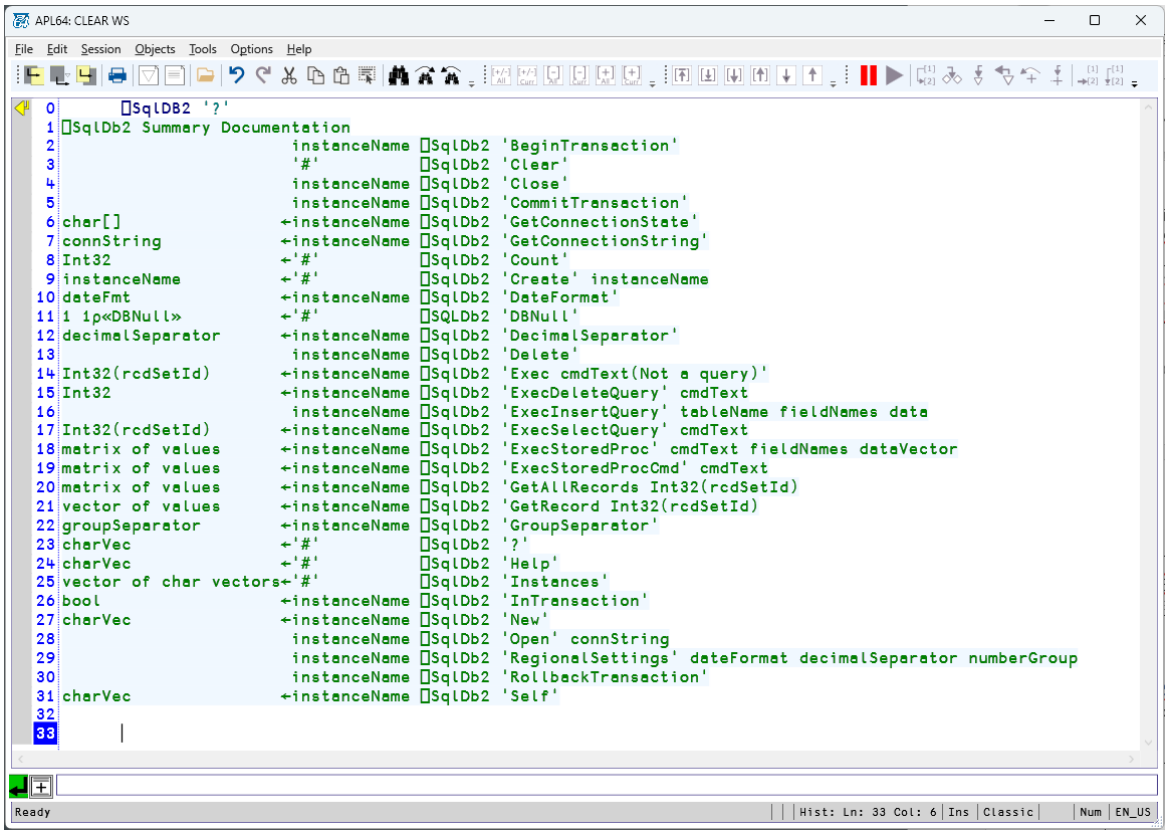
This action returns the .Net Number Group Separator text applicable to the specified ⎕SQLDB2 instance.

```
⎕SQLDB2Self←'#'⎕SQLDB2 'Create' 'S'
⎕SQLDB2 'DateFormat'
⎕SQLDB2 'DecimalSeparator'
⎕SQLDB2 'GroupSeparator'
```



? or Help

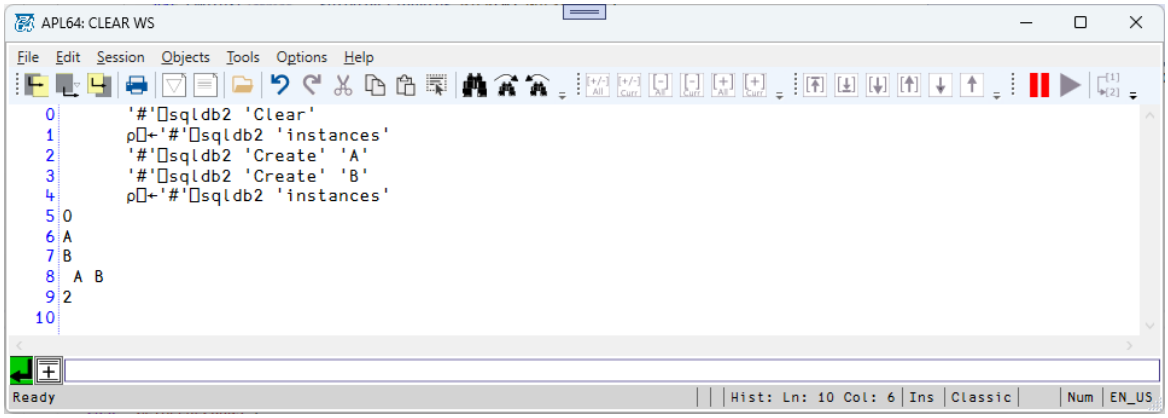
This action is performed on the ⎕sqldb2 object. The ? and Help actions will return a text array containing a summary of the ⎕sqldb2 syntax.



Instances

This action is performed on the ⎕sqldb2 object. The result of the Instances action is the number of existing ⎕sqldb2 instances in the APL64 instance.

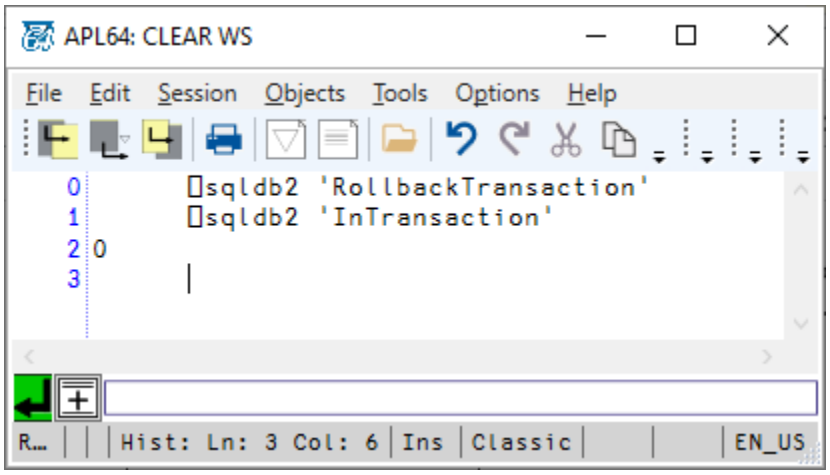
```
'#'⎕sqldb2 'Clear'
⍴⎕←'#'⎕sqldb2 'instances'
'#'⎕sqldb2 'Create' 'A'
'#'⎕sqldb2 'Create' 'B'
⍴⎕←'#'⎕sqldb2 'instances'
```



InTransaction

This action applies to an ⎕sqldb2 instance. The Boolean result of the InTransaction action indicates if a transaction block is in existence for the specified ⎕sqldb2 instance.

```
⎕sqldb2 'RollbackTransaction'
⎕sqldb2 'InTransaction'
```

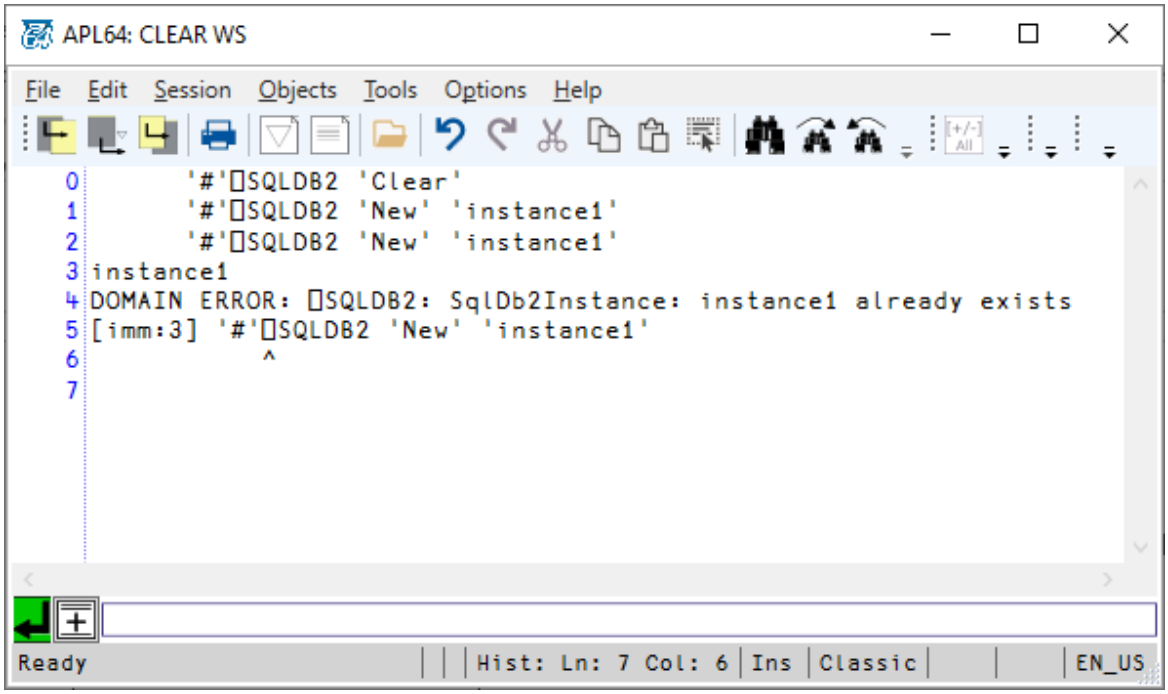


New

This action is performed on the ⎕sqldb2 object. The New action will create an ⎕Sqldb2 instance with a user-provided name in the right argument. Multiple ⎕Sqldb2 instances

are possible in the same APL64 instance, so that multiple Sqldb2 databases may be conveniently accessed. The New action will fail if the named instance already exists. The New action for a particular Sqldb2 database is generally used once in an APL64 instance. The New action does not open a connection to an Sqldb2 database, use the Open action for that purpose. The result of a successful 'New' action is a text vector containing the ⎕Sqldb2 instance name.

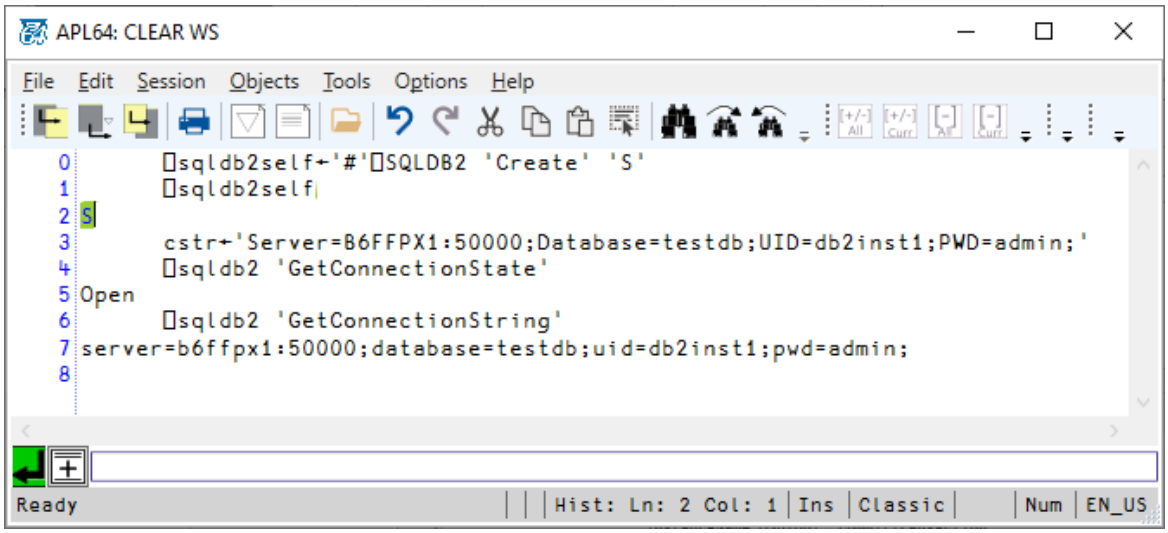
```
'#'⎕SQLDB2                              'Clear'
'#'⎕SQLDB2                  'New'       'instance1'
'#'⎕SQLDB2 'New' 'instance1'
```



Open

This action applies to an ⎕sqldb2 instance. The Open action requires the specification of a connection string as the right argument.

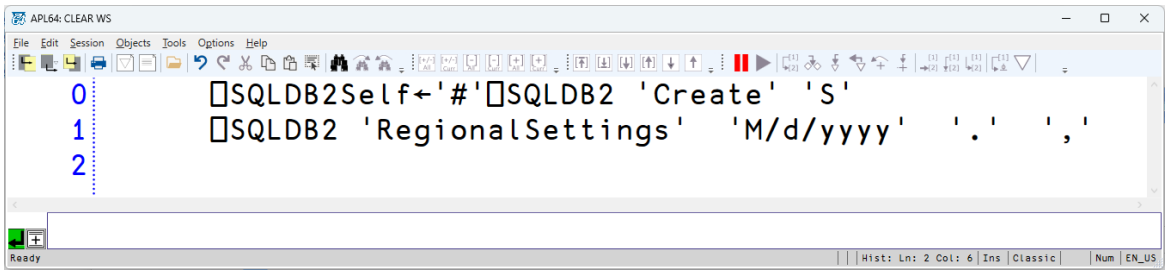
```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕sqldb2 'GetConnectionState'
⎕sqldb2 'GetConnectionString'
```



RegionalSettings

- This action can be used to set the regional values for the specified ⎕SQLDB2 instance:
- ShortDatePattern
- CurrencyDecimalSeparator, NumberDecimalSeparator & PercentDecimalSeparator
- CurrencyGroupSeparator, NumberGroupSeparator & PercentGroupSeparator

```
⎕SQLDB2Self←'#'⎕SQLDB2 'Create' 'S'
⎕SQLDB2 'RegionalSettings'  'M/d/yyyy'  '.'  ','
```

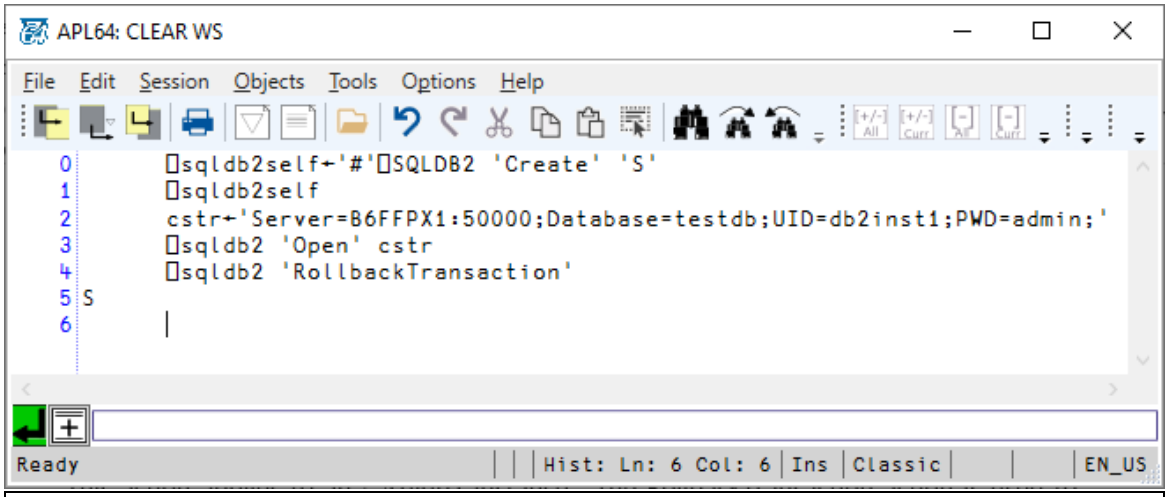


RollbackTransaction

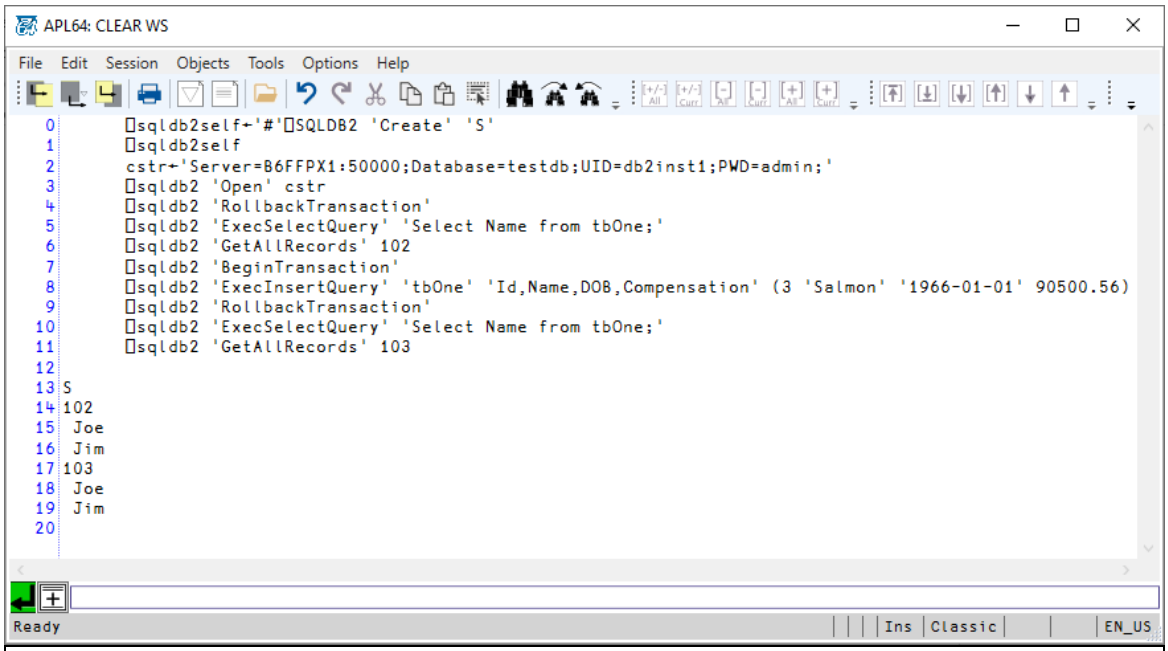
This action applies to an ⎕sqldb2 instance. The RollbackTransaction action is used to cancel any pending SQL operations which are included in the current SQL transaction. The RollbackTransaction action has no result. The RollbackTransaction action has no effect if no SQL transaction is in progress.

```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
```

```
sqldb 'RollbackTransaction'
```

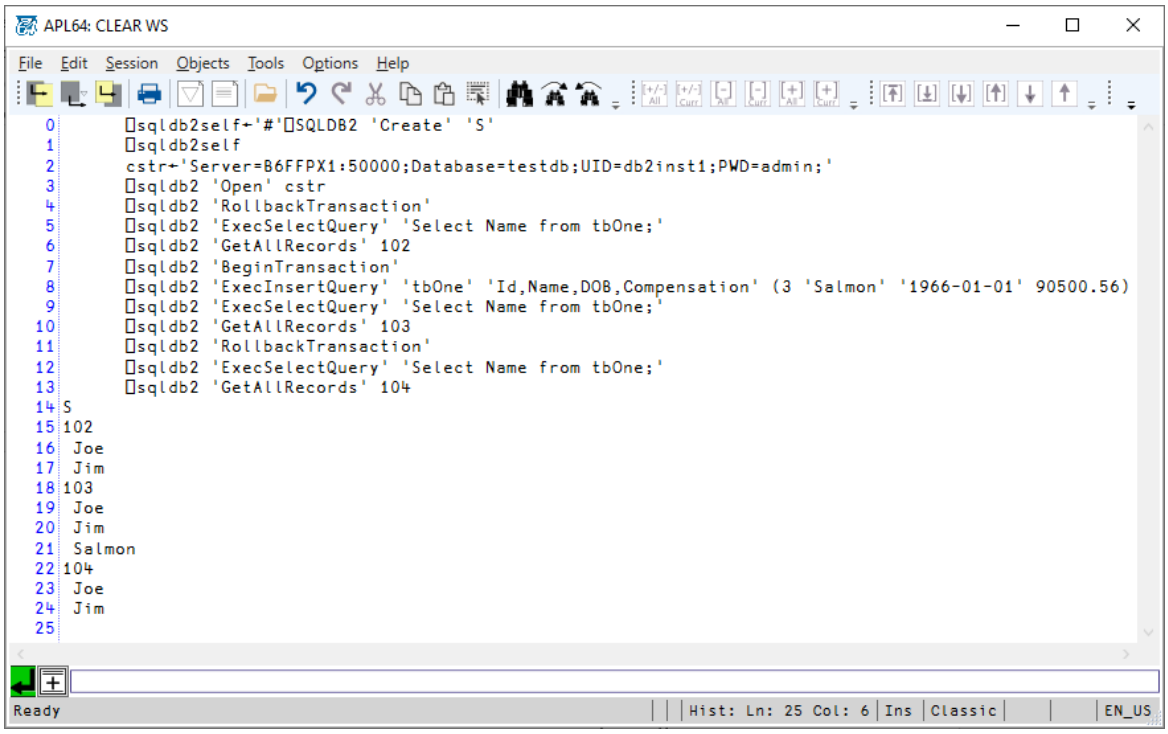



```
sqldb2self←'#'SQLDB2 'Create' 'S'
sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
sqldb2 'Open' cstr
sqldb2 'RollbackTransaction'
sqldb2 'ExecSelectQuery' 'Select Name from tbOne;'
sqldb2 'GetAllRecords' 102
sqldb2 'BeginTransaction'
sqldb2 'ExecInsertQuery' 'tbOne' 'Id,Name,DOB,Compensation' (3 'Salmon' '1966-01-01' 90500.56)
sqldb2 'RollbackTransaction'
sqldb2 'ExecSelectQuery' 'Select Name from tbOne;'
sqldb2 'GetAllRecords' 103
```




```
⎕sqldb2self←'#'⎕SQLDB2 'Create' 'S'
⎕sqldb2self
cstr←'Server=B6FFPX1:50000;Database=testdb;UID=db2inst1;PWD=admin;'
⎕sqldb2 'Open' cstr
⎕sqldb2 'RollbackTransaction'
⎕sqldb2 'ExecSelectQuery' 'Select Name from tbOne;'
⎕sqldb2 'GetAllRecords' 102
⎕sqldb2 'BeginTransaction'
⎕sqldb2 'ExecInsertQuery' 'tbOne' 'Id,Name,DOB,Compensation' (3 'Salmon' '1966-01-01' 90500.56)
⎕sqldb2 'ExecSelectQuery' 'Select Name from tbOne;'
⎕sqldb2 'GetAllRecords' 103
⎕sqldb2 'RollbackTransaction'
⎕sqldb2 'ExecSelectQuery' 'Select Name from tbOne;'
⎕sqldb2 'GetAllRecords' 104
```

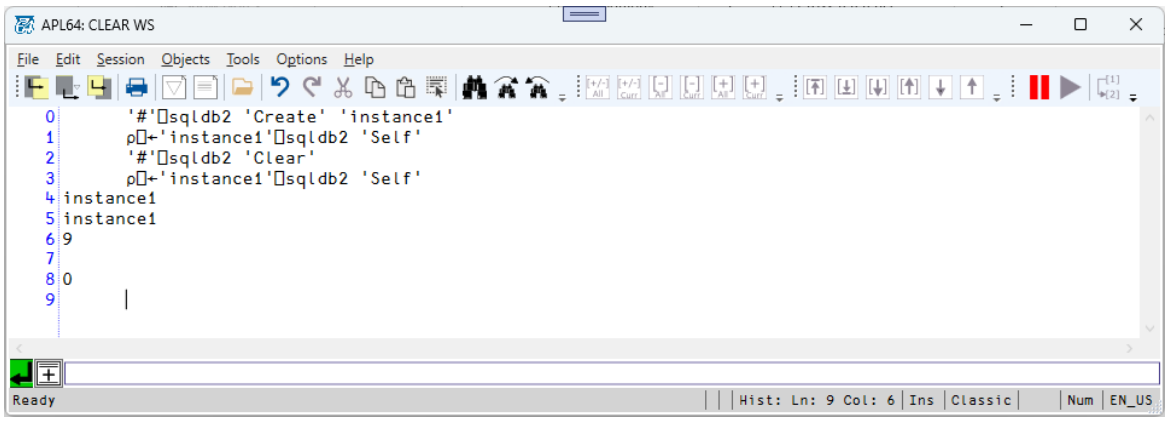
Note that the records in the #103 data set include the pending effects of the ExecInsertQuery action, but the records in the #104 data set do not include the effects of the rollbacked ExecInsertQuery action:



Self

This action is performed on the ⎕sqldb2 object. The right argument specifies an ⎕sqldb2 instance name for a potentially existing ⎕sqldb2 instance. If the ⎕sqldb2 instance exists in the APL64 instance, the result of the Self action is the ⎕sqldb2 instance name.

```
'#'⎕sqldb2 'Create' 'instance1'
ρ⎕←'instance1'⎕sqldb2                    'Self'
'#'⎕sqldb2 'Clear'
ρ⎕←'instance1'⎕sqldb2 'Self'
```



Learn Structured Query Language

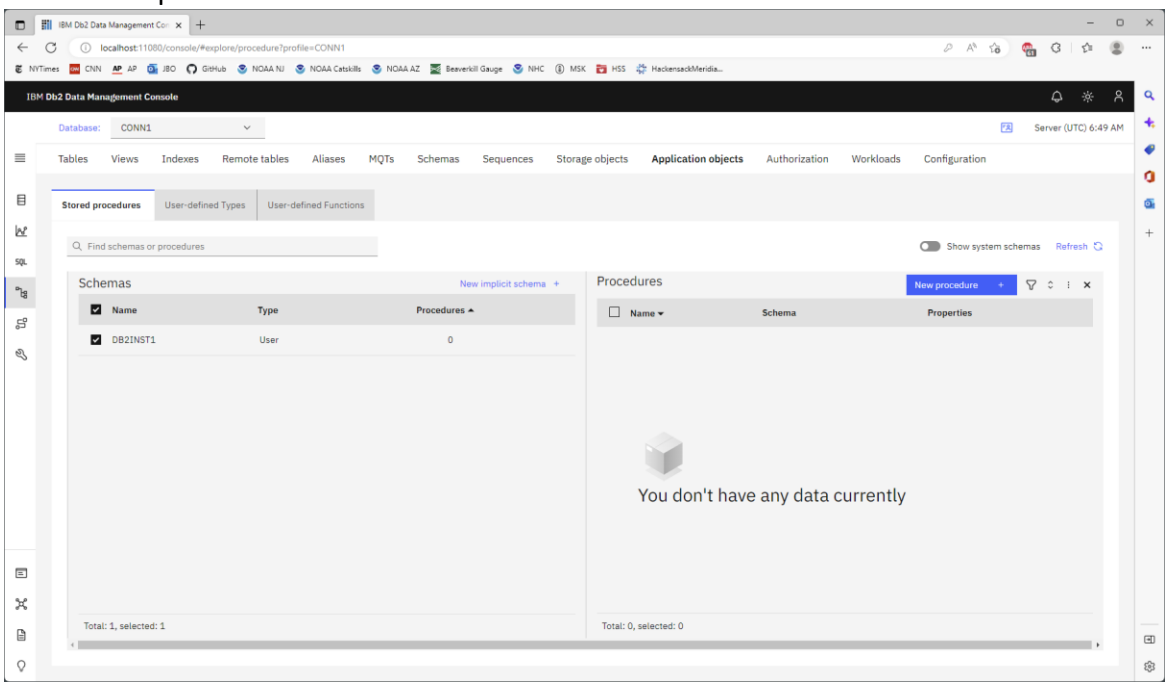
http://www.w3schools.com/sql/default.asp

Adding Sample Stored Procedure

[Sample IBM Db2 Stored Procedures in SQL]

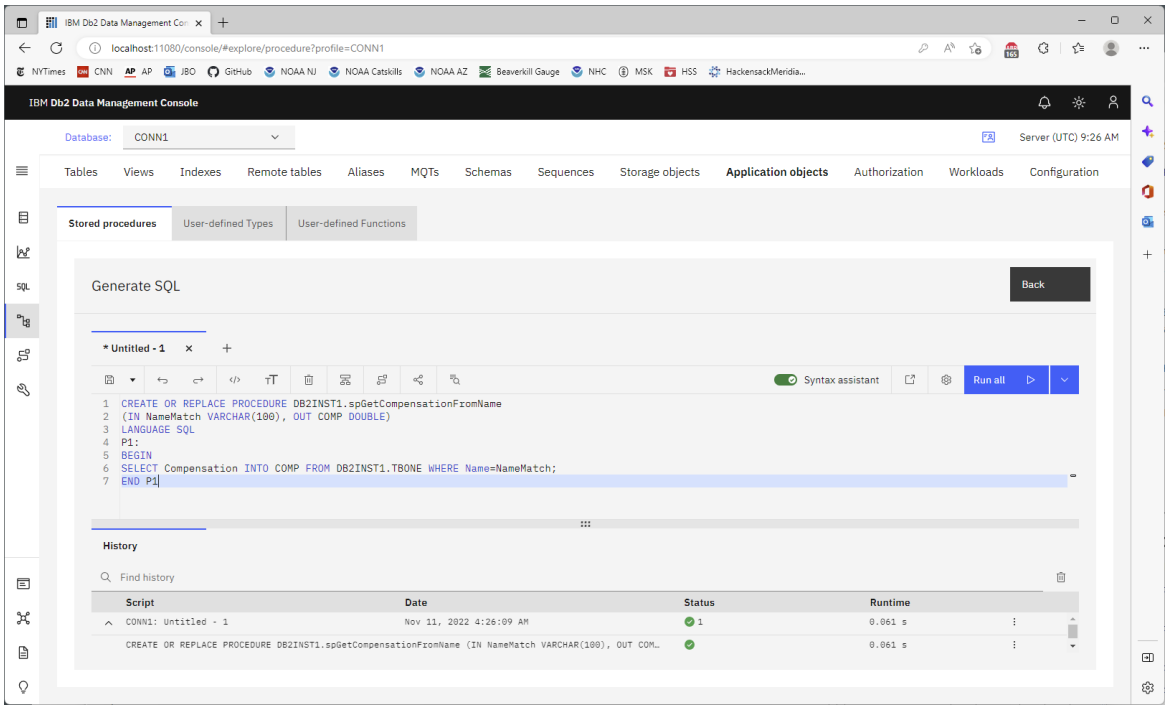
[More IBM Db2 Stored Procedure Samples]

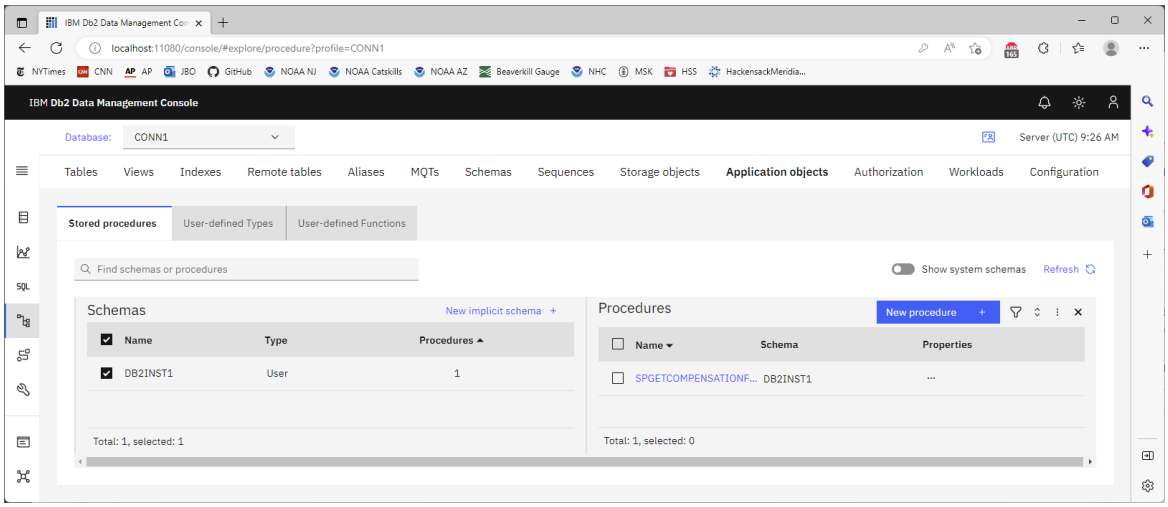
In the IBM Db2 Management Console select the ‘Application objects’ menu item and click the ‘Stored procedures’ button:



Click the ‘New procedure’ button, complete the new stored procedure template and click the ‘Run all’ button to create the stored procedure:

```
CREATE OR REPLACE PROCEDURE DB2INST1.spGetCompensationFromName
(IN NameMatch VARCHAR(100), OUT COMP DOUBLE)
LANGUAGE SQL
P1:
BEGIN
SELECT Compensation INTO COMP FROM DB2INST1.TBONE WHERE Name=NameMatch;
END P1
```

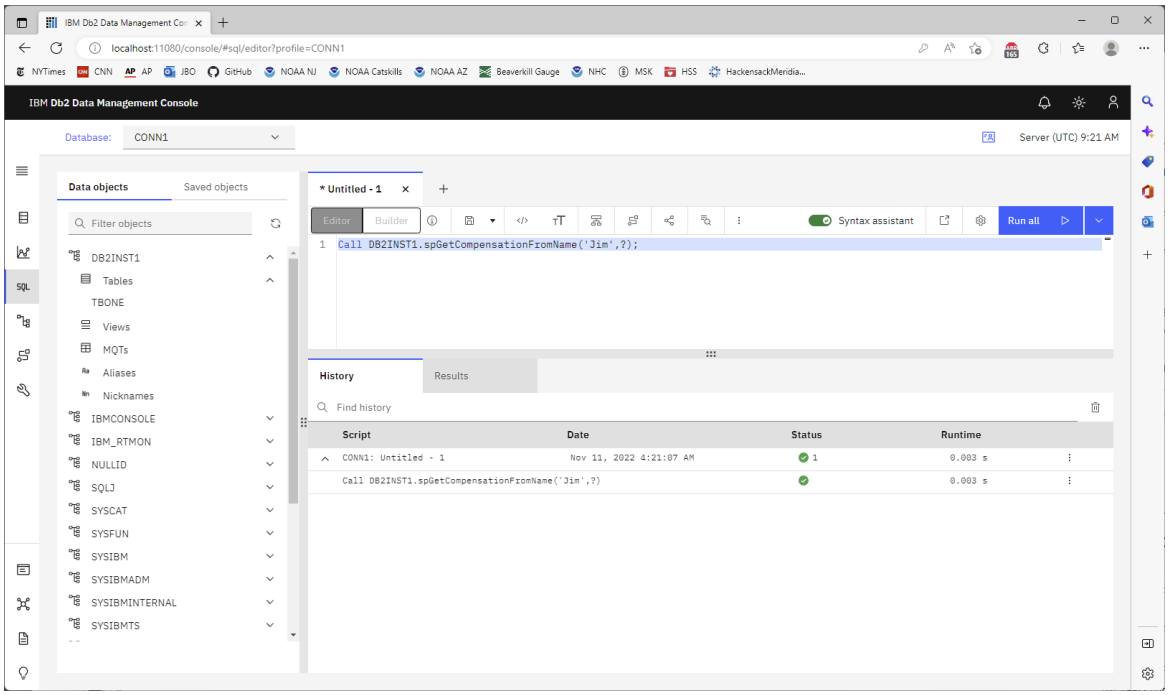




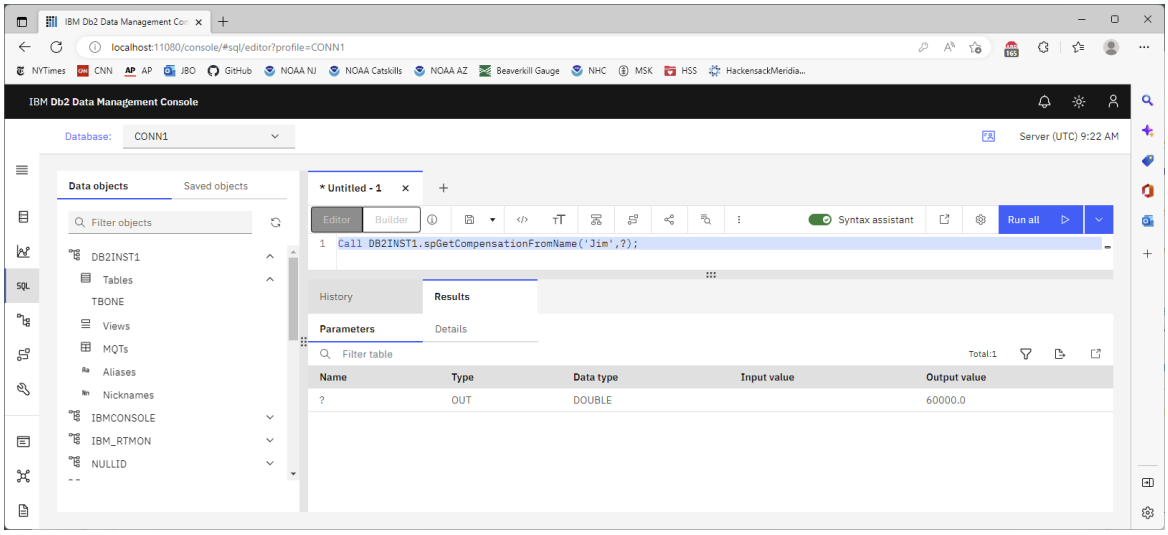
Check the stored procedure runs properly:

Enter the SQL statement and click the ‘Run all’ button:

```
Call DB2INST1.spGetCompensationFromName('Jim',?);
```



Check the results set:



A stored procedure with no arguments or results is used for the 'ExecStoredProcCmd' action example:

```
CREATE OR REPLACE PROCEDURE DB2INST1.spGetCompensationFromName
(IN NameMatch VARCHAR(100), OUT COMP DOUBLE)
LANGUAGE SQL
P1:
BEGIN
Insert into tbone values (3,'Salmon','1950-02-01',150000);
END P1
```