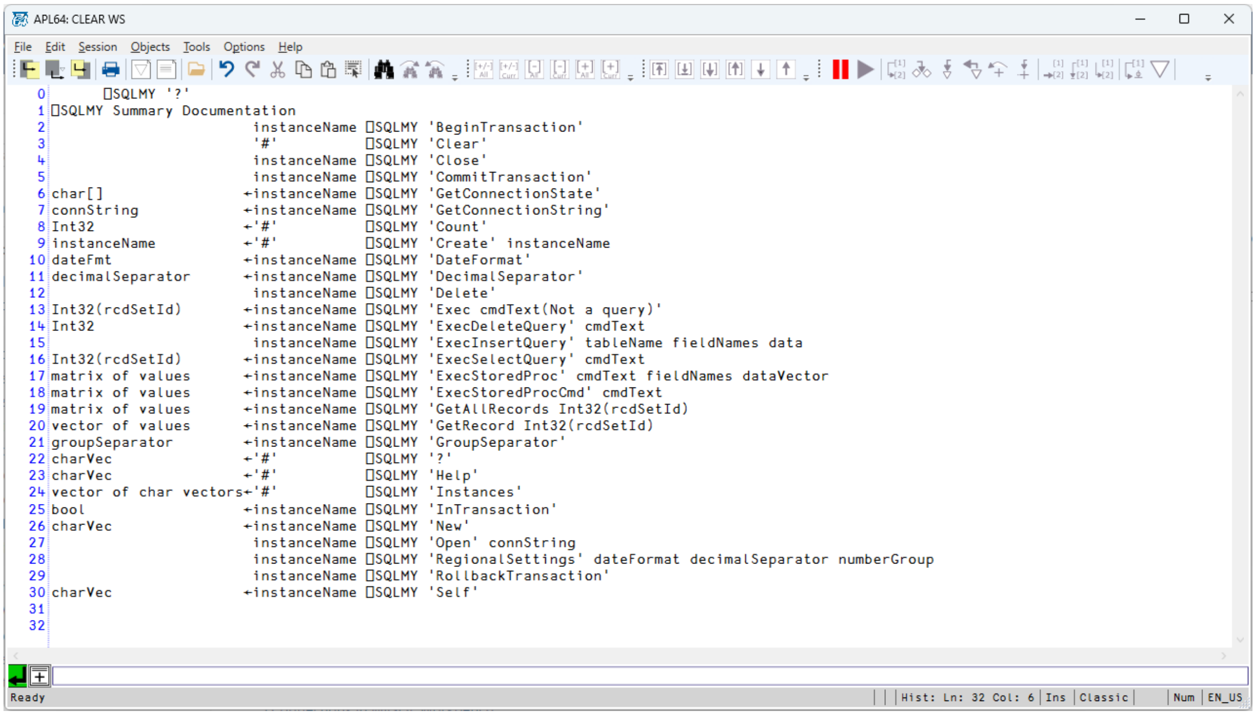
Using ⎕SQLMY

Contents

Overview

The APL64 ⎕SQLMY system function provides a structured query language (SQL)-based interface to a MySql database.

The ⎕SQLMY system function employs the Oracle MySql.Data Nuget package for a MySql .Net client in APL64. The object model for ⎕SQLMY includes methods and properties to easily create a secure, high-performance application system data interface.



```
      ⎕SQLMY '?'
⎕SQLMY Summary Documentation
                       instanceName ⎕SQLMY 'BeginTransaction'
                       '#'          ⎕SQLMY 'Clear'
                       instanceName ⎕SQLMY 'Close'
                       instanceName ⎕SQLMY 'CommitTransaction'
char[]                ←instanceName ⎕SQLMY 'GetConnectionState'
connString            ←instanceName ⎕SQLMY 'GetConnectionString'
Int32                 ←'#'          ⎕SQLMY 'Count'
instanceName          ←'#'          ⎕SQLMY 'Create' instanceName
dateFmt               ←instanceName ⎕SQLMY 'DateFormat'
decimalSeparator      ←instanceName ⎕SQLMY 'DecimalSeparator'
                       instanceName ⎕SQLMY 'Delete'
Int32(rcdSetId)       ←instanceName ⎕SQLMY 'Exec cmdText(Not a query)'
Int32                 ←instanceName ⎕SQLMY 'ExecDeleteQuery' cmdText
                       instanceName ⎕SQLMY 'ExecInsertQuery' tableName fieldNames data
Int32(rcdSetId)       ←instanceName ⎕SQLMY 'ExecSelectQuery' cmdText
matrix of values      ←instanceName ⎕SQLMY 'ExecStoredProc' cmdText fieldNames dataVector
matrix of values      ←instanceName ⎕SQLMY 'ExecStoredProcCmd' cmdText
matrix of values      ←instanceName ⎕SQLMY 'GetAllRecords Int32(rcdSetId)
vector of values      ←instanceName ⎕SQLMY 'GetRecord Int32(rcdSetId)
groupSeparator        ←instanceName ⎕SQLMY 'GroupSeparator'
charVec               ←'#'          ⎕SQLMY '?'
charVec               ←'#'          ⎕SQLMY 'Help'
vector of char vectors←'#'          ⎕SQLMY 'Instances'
bool                  ←instanceName ⎕SQLMY 'InTransaction'
charVec               ←instanceName ⎕SQLMY 'New'
                       instanceName ⎕SQLMY 'Open' connString
                       instanceName ⎕SQLMY 'RegionalSettings' dateFormat decimalSeparator numberGroup
                       instanceName ⎕SQLMY 'RollbackTransaction'
charVec               ←instanceName ⎕SQLMY 'Self'
```

The MySql Community Edition database is available as a no-cost option.

Prerequisites

To use ⎕SQLMY, the MySql database engine must be installed and running on the target workstation or on a server to which the target workstation has access.

A Database Must Exist in the Running Instance of MySql:

A database must exist on the target workstation or on a server to which the target workstation has access. The MySql Workbench is an excellent tool to establish this instance.

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.

The examples in the remainder of this document are based on the sample database, tables and values illustrated here. The database name (schema) is 'schema1' and the table in the database is 'tbone' which can be created using the MySql Workbench:

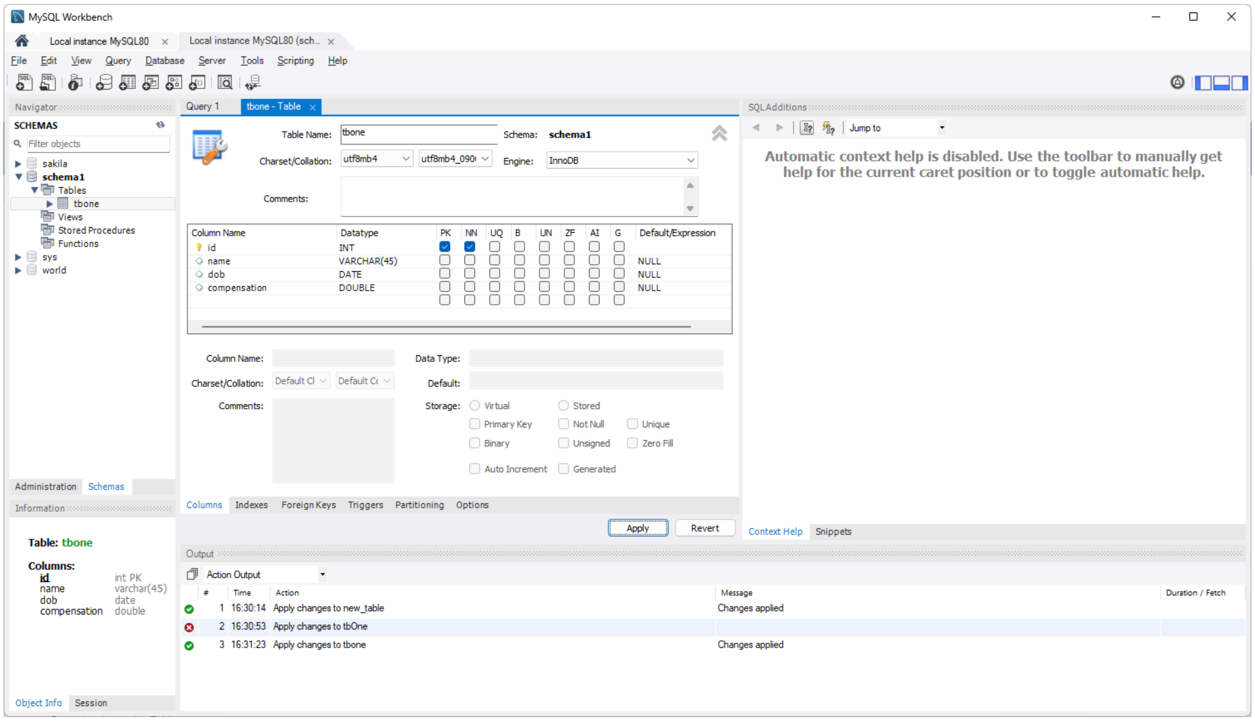


Table Name: tbone Schema: **schema1**

Charset/Collation: utf8mb4 utf8mb4_090 Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	☑	☑	□	□	□	□	□	□	
name	VARCHAR(45)	□	□	□	□	□	□	□	□	NULL
dob	DATE	□	□	□	□	□	□	□	□	NULL
compensation	DOUBLE	□	□	□	□	□	□	□	□	NULL
		□	□	□	□	□	□	□	□	

Column Name: Data Type:

Charset/Collation: Default Cl Default Cc Default:

Comments: Storage: ○ Virtual ○ Stored

□ Primary Key □ Not Null □ Unique

□ Binary □ Unsigned □ Zero Fill

□ Auto Increment □ Generated

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Table: tbone

Columns:

id	int PK
name	varchar(45)
dob	date
compensation	double

	#	Time	Action	Message	Duration / Fetch
✓	1	16:30:14	Apply changes to new_table	Changes applied	
✗	2	16:30:53	Apply changes to tbOne		
✓	3	16:31:23	Apply changes to tbone	Changes applied	

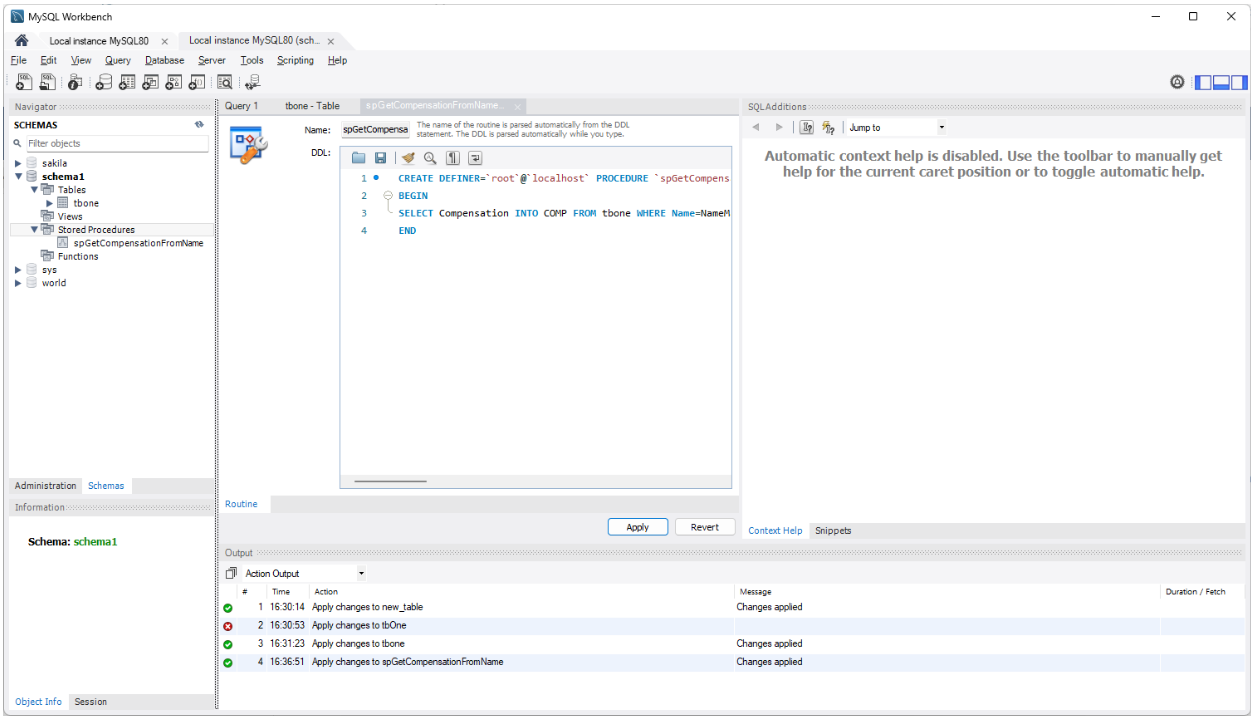
Insert the tbone table sample data:

```
insert into tbone values (1,'Joe','1960-01-01',50000);
insert into tbone values (2,'Jim','1961-01-01',60000);
```

Create the sample stored procedure with arguments and results:

```
CREATE PROCEDURE `spGetCompensationFromName`
(IN NameMatch VARCHAR(45), OUT COMP DOUBLE)
BEGIN
```

```
SELECT Compensation INTO COMP FROM tbone WHERE Name=NameMatch;
END
```

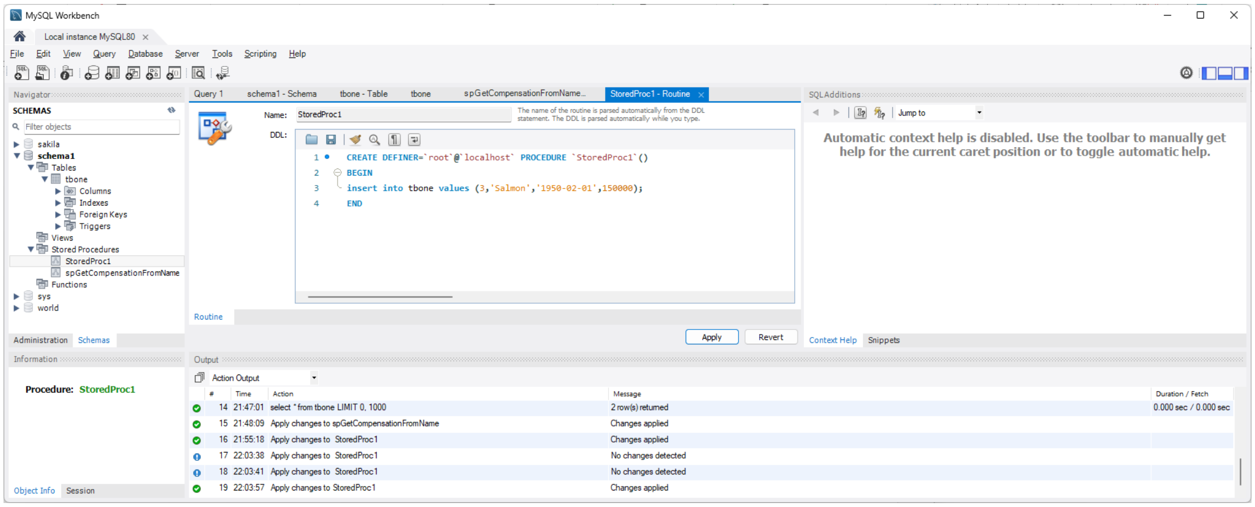

MySQL Workbench
Local instance MySQL80
Local instance MySQL80 (sch...
File Edit View Query Database Server Tools Scripting Help
Navigator
SCHEMAS
Filter objects
sakila
schema1
Tables
tbone
Views
Stored Procedures
spGetCompensationFromName
Functions
sys
world
Administration Schemas
Information
Schema: schema1
Object Info Session
Query 1
tbone - Table
spGetCompensationFromName...
Name: spGetCompensa
The name of the routine is parsed automatically from the DDL statement. The DDL is parsed automatically while you type.
DDL:
Routine
Apply
Revert
SQLAdditions
Jump to
Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.
Context Help Snippets
Output
Action Output

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `spGetCompens
BEGIN
SELECT Compensation INTO COMP FROM tbone WHERE Name=NameM
END
```

	#	Time	Action	Message	Duration / Fetch
✓	1	16:30:14	Apply changes to new_table	Changes applied	
✗	2	16:30:53	Apply changes to tbOne		
✓	3	16:31:23	Apply changes to tbone	Changes applied	
✓	4	16:36:51	Apply changes to spGetCompensationFromName	Changes applied	

Create the sample stored procedure without arguments or results:

```
CREATE PROCEDURE ` StoredProc1`
()
BEGIN
insert into tbone values (3,'Salmon','1950-02-01',150000);
END
```



```
CREATE DEFINER=`root`@`localhost` PROCEDURE `StoredProc1`()
BEGIN
insert into tbone values (3,'Salmon','1950-02-01',150000);
END
```

	#	Time	Action	Message	Duration / Fetch
✓	14	21:47:01	select * from tbone LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
✓	15	21:48:09	Apply changes to spGetCompensationFromName	Changes applied	
✓	16	21:55:18	Apply changes to StoredProc1	Changes applied	
ℹ	17	22:03:38	Apply changes to StoredProc1	No changes detected	
ℹ	18	22:03:41	Apply changes to StoredProc1	No changes detected	
✓	19	22:03:57	Apply changes to StoredProc1	Changes applied	

Actions

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

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

The name of an ⎕SQLMY instance can be specified by:

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

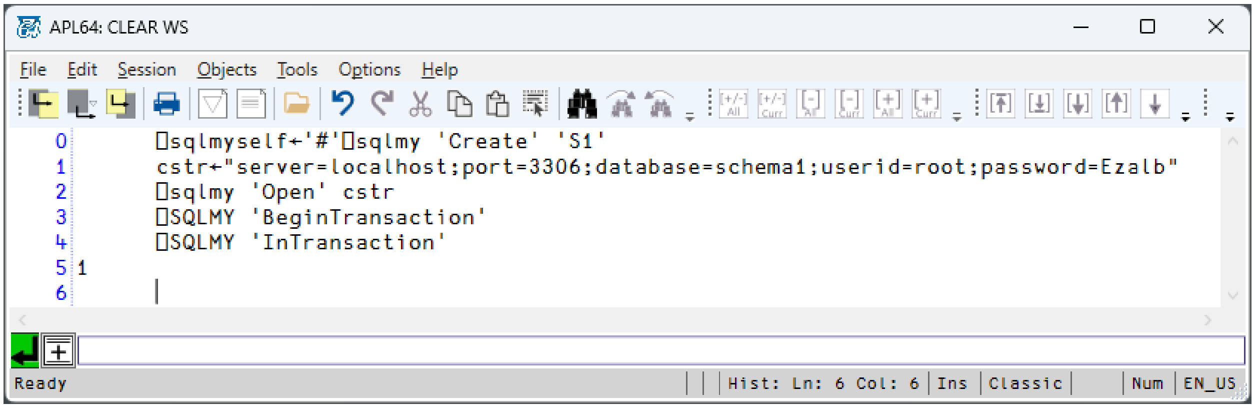
⎕SQLMY action names are not case sensitive. ⎕SQLMY instance names are case sensitive.

BeginTransaction

This action applies to an ⎕SQLMY instance. The BeginTransaction action, for the MySql instance named in the left argument, indicates that subsequent ⎕SQLMY actions will be incorporated into an SQL transaction. An SQL transaction may contain multiple ⎕SQLMY 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.

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'BeginTransaction'
⎕SQLMY 'InTransaction'
```

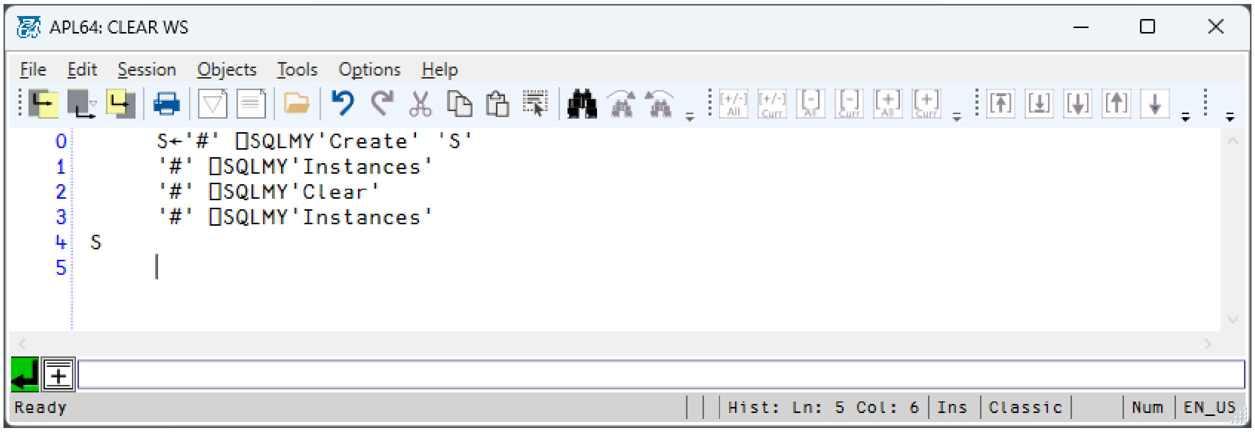

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 6 Col: 6 Ins Classic Num EN_US

```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'BeginTransaction'
      ⎕SQLMY 'InTransaction'
1
```

Clear

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

```
S←'#' ⎕SQLMY'Create' 'S'
'#' ⎕SQLMY'Instances'
'#' ⎕SQLMY'Clear'
'#' ⎕SQLMY'Instances'
```

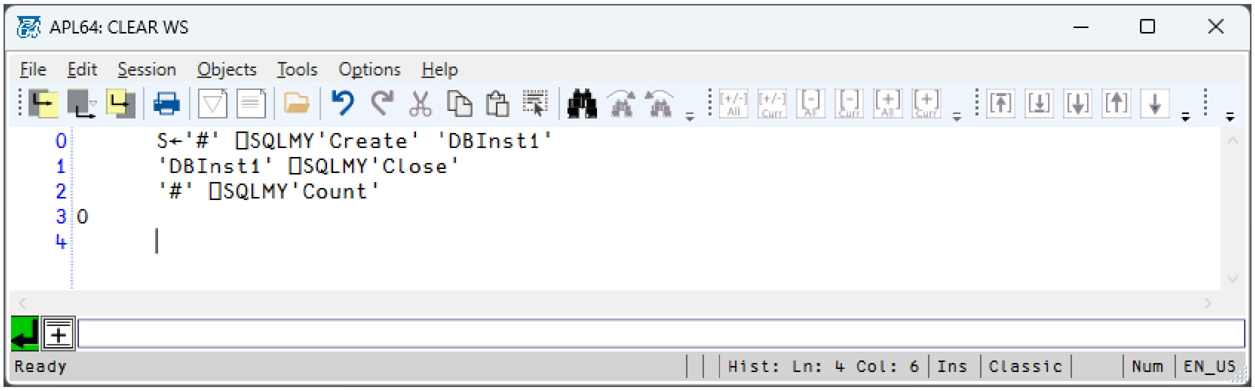


```
      S←'#' ⎕SQLMY'Create' 'S'
      '#' ⎕SQLMY'Instances'
      '#' ⎕SQLMY'Clear'
      '#' ⎕SQLMY'Instances'
 S
```

Close

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

```
S←'#' ⎕SQLMY'Create' 'DBInst1'
'DBInst1' ⎕SQLMY'Close'
 '#' ⎕SQLMY'Count'
```



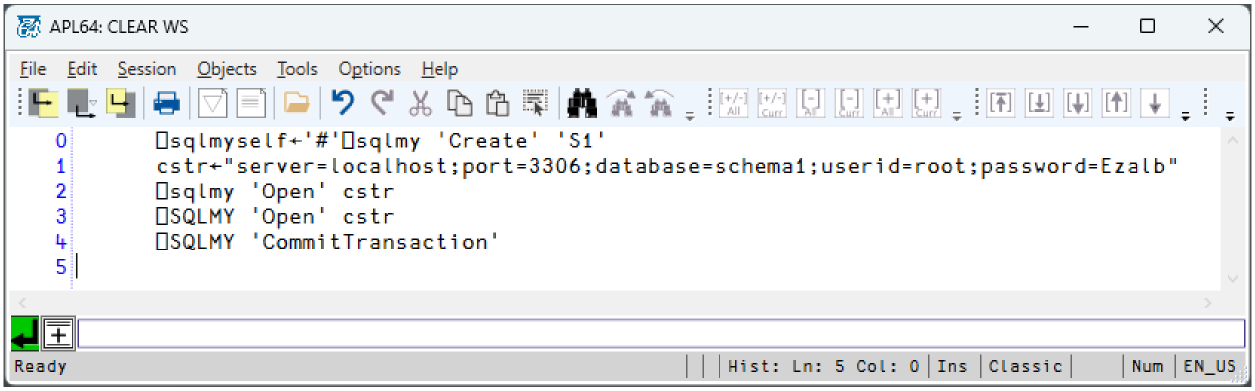
```
      S←'#' ⎕SQLMY'Create' 'DBInst1'
      'DBInst1' ⎕SQLMY'Close'
      '#' ⎕SQLMY'Count'
0
```

CommitTransaction

This action applies to an ⎕SQLMY instance. The CommitTransaction action causes the effects of the current transaction to be applied to the connected MySql 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 MySql database. The CommitTransaction action has no result.

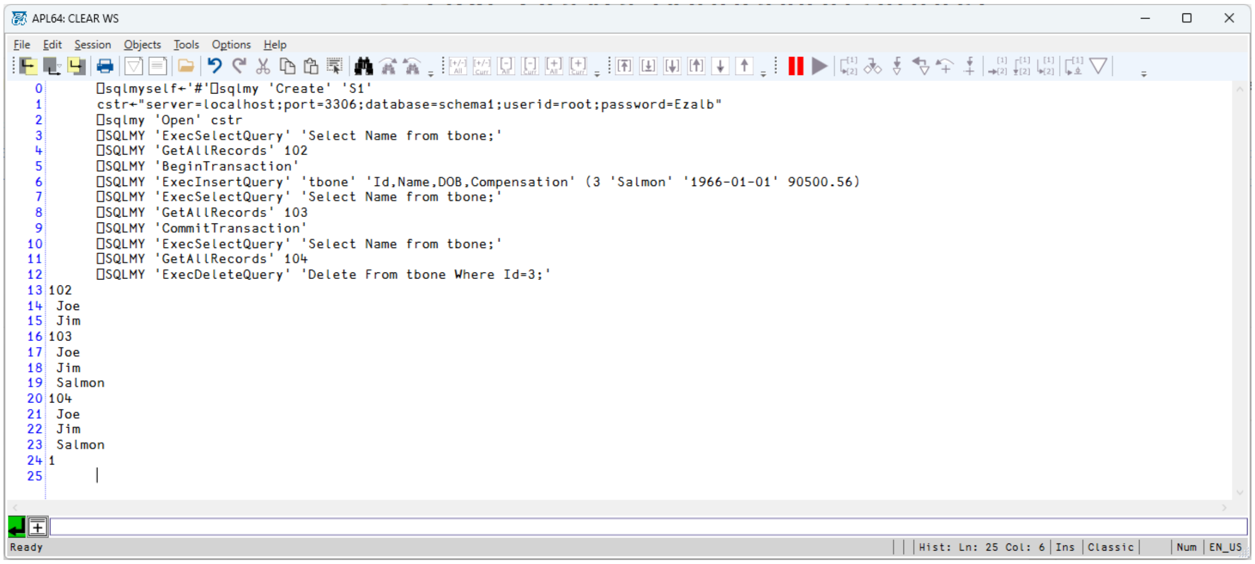
```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕SQLMY 'Open' cstr
⎕SQLMY 'CommitTransaction'
```



```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'Open' cstr
      ⎕SQLMY 'CommitTransaction'
```

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
⎕SQLMY 'GetAllRecords' 102
⎕SQLMY 'BeginTransaction'
⎕SQLMY 'ExecInsertQuery' 'tbone' 'Id,Name,DOB,Compensation' (3 'Salmon' '1966-01-01' 90500.56)
⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
⎕SQLMY 'GetAllRecords' 103
⎕SQLMY 'CommitTransaction'
⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
⎕SQLMY 'GetAllRecords' 104
⎕SQLMY '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:



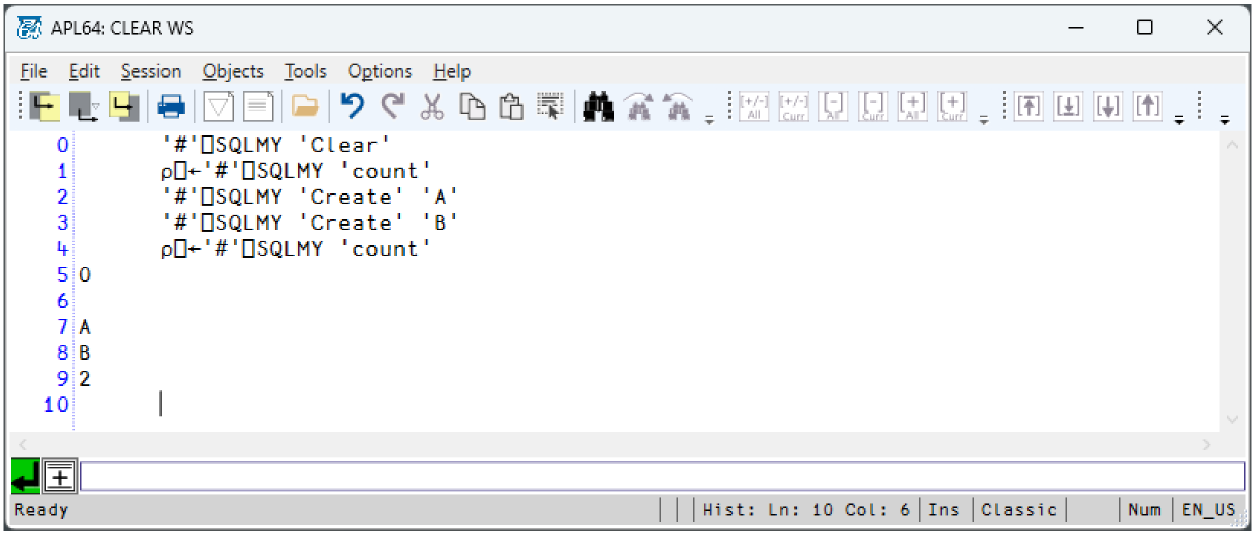
```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
      ⎕SQLMY 'GetAllRecords' 102
      ⎕SQLMY 'BeginTransaction'
      ⎕SQLMY 'ExecInsertQuery' 'tbone' 'Id,Name,DOB,Compensation' (3 'Salmon' '1966-01-01' 90500.56)
      ⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
      ⎕SQLMY 'GetAllRecords' 103
      ⎕SQLMY 'CommitTransaction'
      ⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
      ⎕SQLMY 'GetAllRecords' 104
      ⎕SQLMY 'ExecDeleteQuery' 'Delete From tbone Where Id=3;'
102
 Joe
 Jim
103
 Joe
 Jim
 Salmon
104
 Joe
 Jim
 Salmon
1
```

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 25 Col: 6 Ins Classic Num EN_US

Count

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

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



```
      '#'⎕SQLMY 'Clear'
      ρ⎕←'#'⎕SQLMY 'count'
      '#'⎕SQLMY 'Create' 'A'
      '#'⎕SQLMY 'Create' 'B'
      ρ⎕←'#'⎕SQLMY 'count'
0

A
B
2
```

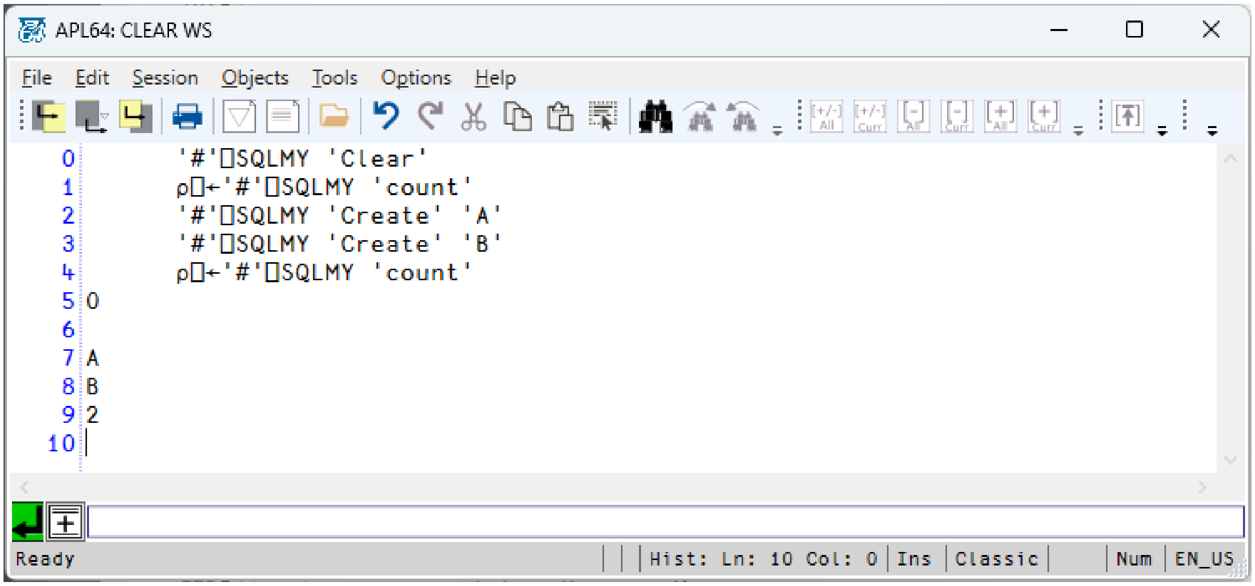
APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 10 Col: 6 Ins Classic Num EN_US

Create

This action is performed on the ⎕SQLMY object. The Create action will create an ⎕SQLMY instance with a user-provided name in the right argument. Multiple ⎕SQLMY instances are possible in the same APL64 instance, so that multiple MySql 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 MySql database is generally used once in an APL64 instance. The Create action does not open a connection to an MySql database, use the Open action for that purpose. The result of a successful 'Create' action is a text vector containing the □SQLMY instance name.

```
'#'□SQLMY 'Clear'
'ρ□←'#'□SQLMY 'count'
'#'□SQLMY 'Create' 'A'
'#'□SQLMY 'Create' 'B'
'#'□SQLMY 'Instances'
'ρ□←'#'□SQLMY 'count'
```



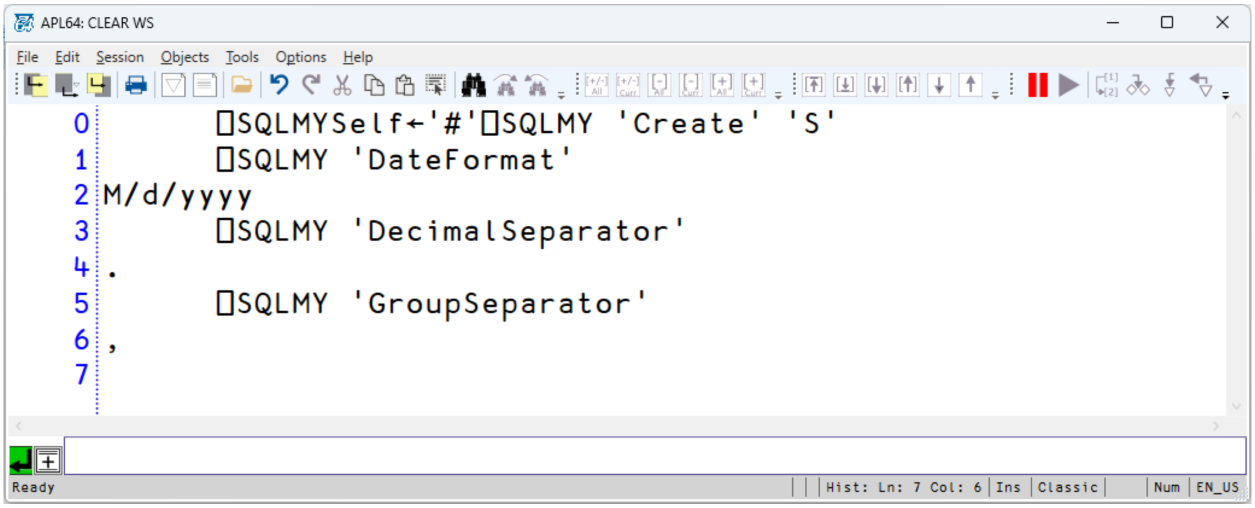
```
      '#'⎕SQLMY 'Clear'
      ρ⎕←'#'⎕SQLMY 'count'
      '#'⎕SQLMY 'Create' 'A'
      '#'⎕SQLMY 'Create' 'B'
      ρ⎕←'#'⎕SQLMY 'count'
0

A
B
2
```

DateFormat

This action returns the current .Net Short Date Pattern text applicable to the specified □SQLMY instance.

```
□SQLMYSelf←'#'□SQLMY 'Create' 'S'
□SQLMY 'DateFormat'
□SQLMY 'DecimalSeparator'
□SQLMY 'GroupSeparator'
```

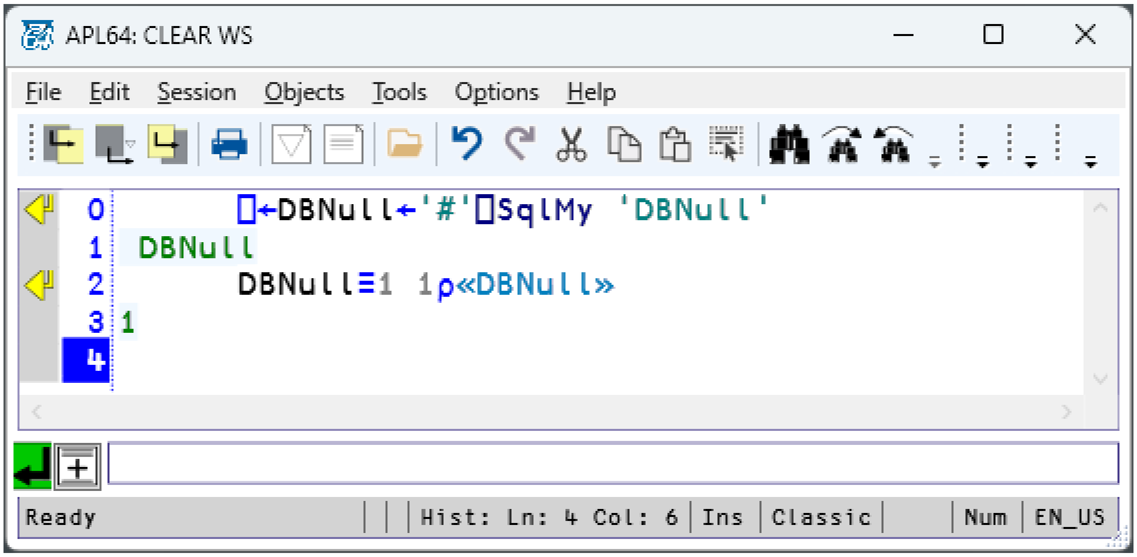

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 7 Col: 6 Ins Classic Num EN_US

```
      ⎕SQLMYSelf←'#'⎕SQLMY 'Create' 'S'
      ⎕SQLMY 'DateFormat'
M/d/yyyy
      ⎕SQLMY 'DecimalSeparator'
.
      ⎕SQLMY 'GroupSeparator'
,
```

DBNull

This action returns a value which may be used to specify null values for the ExecInsertQuery and ExecStoredProc ⎕SQLMy 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←'#'⎕SqlMy 'DBNull'
DBNull≡1 1⍴«DBNull»
```

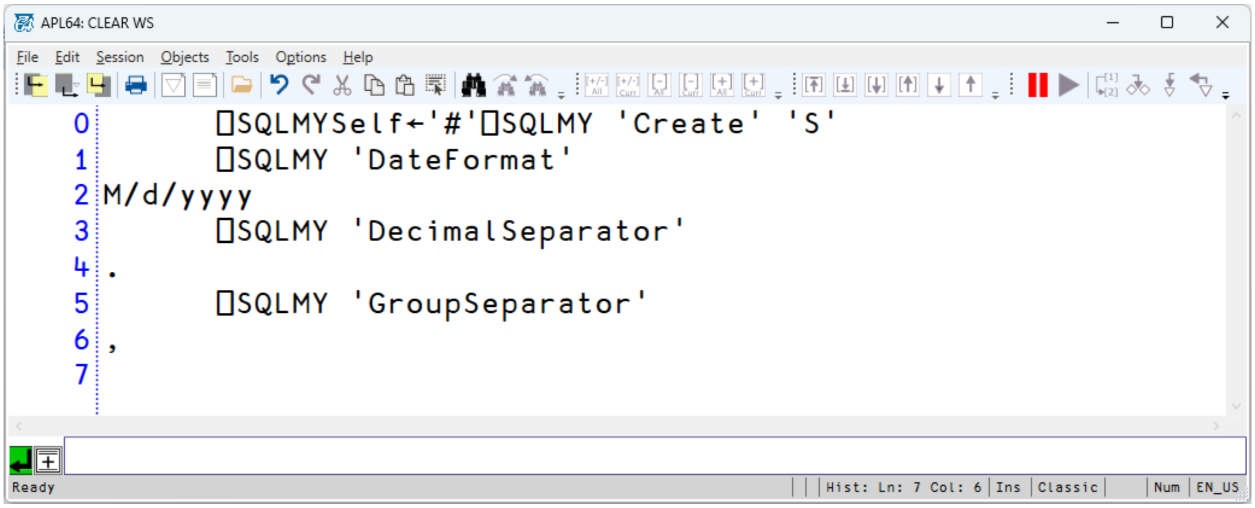

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 4 Col: 6 Ins Classic Num EN_US

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

DecimalSeparator

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

```
⎕SQLMYSelf←'#'⎕SQLMY 'Create' 'S'
⎕SQLMY 'DateFormat'
⎕SQLMY 'DecimalSeparator'
⎕SQLMY 'GroupSeparator'
```

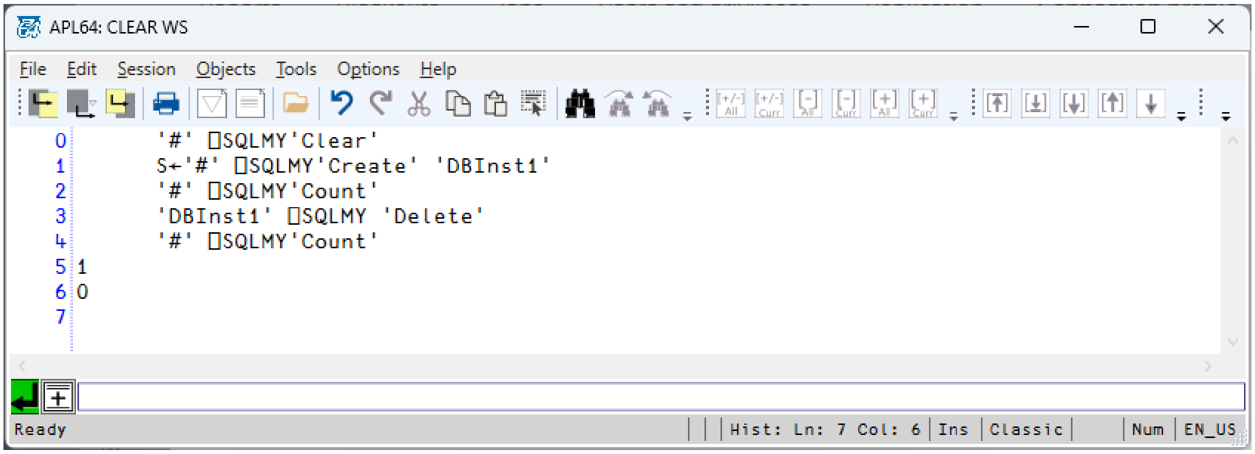

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 7 Col: 6 Ins Classic Num EN_US

```
      ⎕SQLMYSelf←'#'⎕SQLMY 'Create' 'S'
      ⎕SQLMY 'DateFormat'
M/d/yyyy
      ⎕SQLMY 'DecimalSeparator'
.
      ⎕SQLMY 'GroupSeparator'
,
```

Delete

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

```
'#' ⎕SQLMY'Clear'
S←'#' ⎕SQLMY'Create' 'DBInst1'
 '#' ⎕SQLMY'Count'
'DBInst1' ⎕SQLMY 'Delete'
 '#' ⎕SQLMY'Count'
```


APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 7 Col: 6 Ins Classic Num EN_US

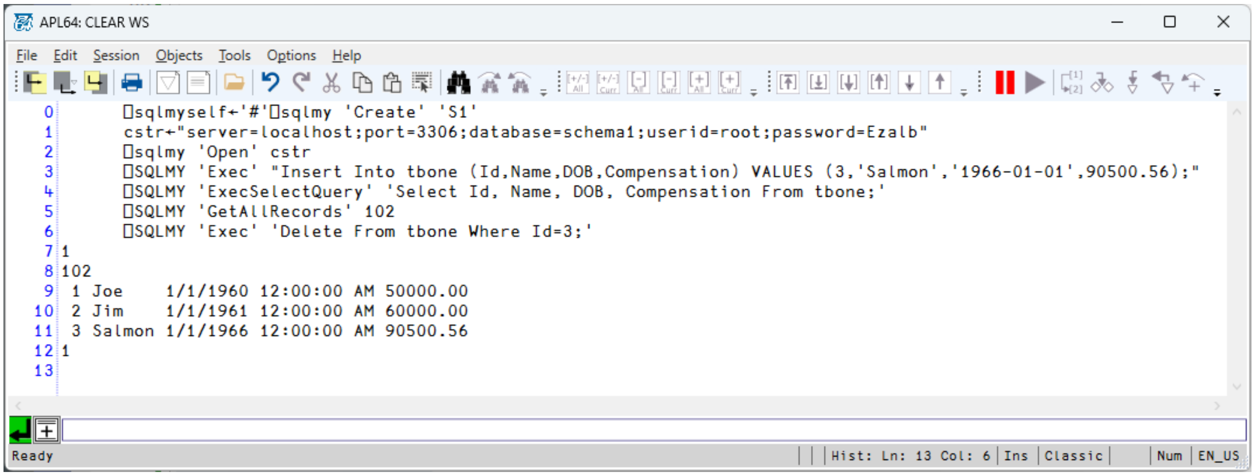
```
      '#' ⎕SQLMY'Clear'
      S←'#' ⎕SQLMY'Create' 'DBInst1'
      '#' ⎕SQLMY'Count'
      'DBInst1' ⎕SQLMY 'Delete'
      '#' ⎕SQLMY'Count'
1
0
```

Exec

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

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
```

```
⎕sqlmy 'Open' cstr
⎕SQLMY 'Exec' "Insert Into tbone (Id,Name,DOB,Compensation) VALUES (3,'Salmon','1966-01-01',90500.56);"
⎕SQLMY 'ExecSelectQuery' 'Select Id, Name, DOB, Compensation From tbone;'
⎕SQLMY 'GetAllRecords' 102
⎕SQLMY 'Exec' 'Delete From tbone Where Id=3;'
```

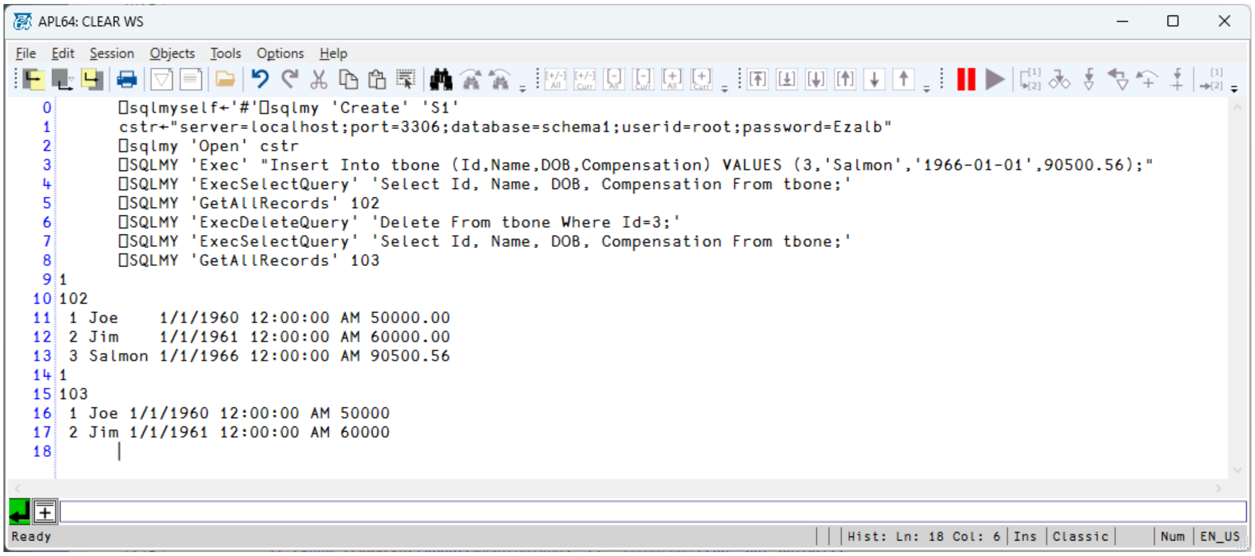


```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'Exec' "Insert Into tbone (Id,Name,DOB,Compensation) VALUES (3,'Salmon','1966-01-01',90500.56);"
      ⎕SQLMY 'ExecSelectQuery' 'Select Id, Name, DOB, Compensation From tbone;'
      ⎕SQLMY 'GetAllRecords' 102
      ⎕SQLMY 'Exec' 'Delete From tbone Where Id=3;'
1
102
 1 Joe    1/1/1960 12:00:00 AM 50000.00
 2 Jim    1/1/1961 12:00:00 AM 60000.00
 3 Salmon 1/1/1966 12:00:00 AM 90500.56
1
```

ExecDeleteQuery

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

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'Exec' "Insert Into tbone (Id,Name,DOB,Compensation) VALUES (3,'Salmon','1966-01-01',90500.56);"
⎕SQLMY 'ExecSelectQuery' 'Select Id, Name, DOB, Compensation From tbone;'
⎕SQLMY 'GetAllRecords' 102
⎕SQLMY 'ExecDeleteQuery' 'Delete From tbone Where Id=3;'
⎕SQLMY 'ExecSelectQuery' 'Select Id, Name, DOB, Compensation From tbone;'
⎕SQLMY 'GetAllRecords' 103
```



```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'Exec' "Insert Into tbone (Id,Name,DOB,Compensation) VALUES (3,'Salmon','1966-01-01',90500.56);"
      ⎕SQLMY 'ExecSelectQuery' 'Select Id, Name, DOB, Compensation From tbone;'
      ⎕SQLMY 'GetAllRecords' 102
      ⎕SQLMY 'ExecDeleteQuery' 'Delete From tbone Where Id=3;'
      ⎕SQLMY 'ExecSelectQuery' 'Select Id, Name, DOB, Compensation From tbone;'
      ⎕SQLMY 'GetAllRecords' 103
1
102
 1 Joe    1/1/1960 12:00:00 AM 50000.00
 2 Jim    1/1/1961 12:00:00 AM 60000.00
 3 Salmon 1/1/1966 12:00:00 AM 90500.56
1
103
 1 Joe 1/1/1960 12:00:00 AM 50000
 2 Jim 1/1/1961 12:00:00 AM 60000
      |
```

ExecInsertQuery

This action applies to an ⎕SQLMY instance.

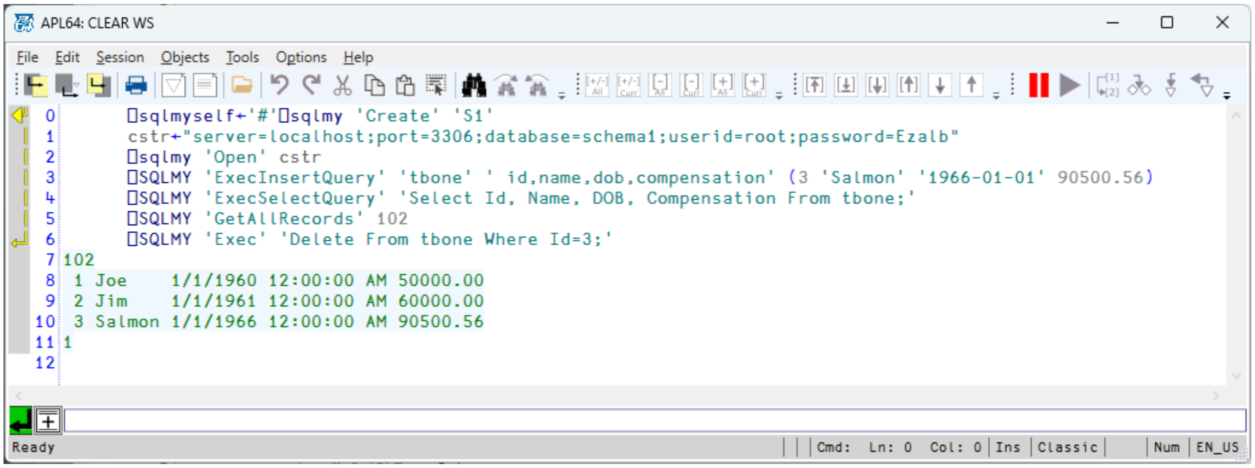
Right arguments:

1. MySql table name
2. MySql column names: Comma delimited text with selected MySql table column names, or '*' to indicate all MySql 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 MySql column names and APL64 data matrix columns must be conformable.

Specify all columns by name:

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'ExecInsertQuery' 'tbone' ' id,name,dob,compensation' (3 'Salmon' '1966-01-01' 90500.56)
⎕SQLMY 'ExecSelectQuery' 'Select Id, Name, DOB, Compensation From tbone;'
⎕SQLMY 'GetAllRecords' 102
⎕SQLMY 'Exec' 'Delete From tbone Where Id=3;'
```

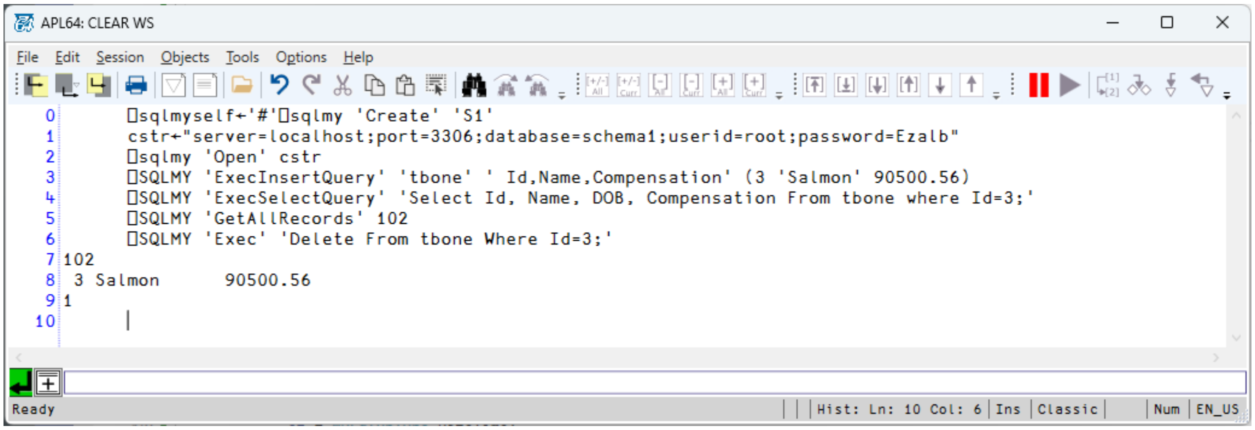


APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Cmd: Ln: 0 Col: 0 Ins Classic Num EN_US

```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'ExecInsertQuery' 'tbone' ' id,name,dob,compensation' (3 'Salmon' '1966-01-01' 90500.56)
      ⎕SQLMY 'ExecSelectQuery' 'Select Id, Name, DOB, Compensation From tbone;'
      ⎕SQLMY 'GetAllRecords' 102
      ⎕SQLMY 'Exec' 'Delete From tbone Where Id=3;'
102
 1 Joe    1/1/1960 12:00:00 AM 50000.00
 2 Jim    1/1/1961 12:00:00 AM 60000.00
 3 Salmon 1/1/1966 12:00:00 AM 90500.56
1
```

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:

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'ExecInsertQuery' 'tbone' ' Id,Name,Compensation' (3 'Salmon' 90500.56)
⎕SQLMY 'ExecSelectQuery' 'Select Id, Name, DOB, Compensation From tbone where Id=3;'
⎕SQLMY 'GetAllRecords' 102
⎕SQLMY 'Exec' 'Delete From tbone Where Id=3;'
```

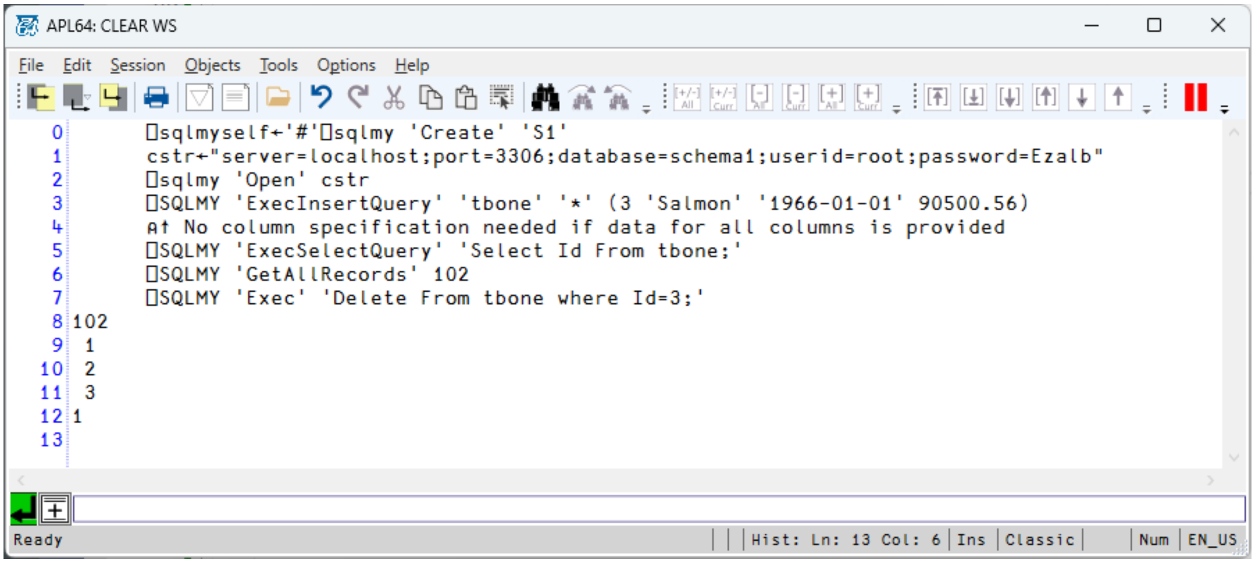


APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 10 Col: 6 Ins Classic Num EN_US

```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'ExecInsertQuery' 'tbone' ' Id,Name,Compensation' (3 'Salmon' 90500.56)
      ⎕SQLMY 'ExecSelectQuery' 'Select Id, Name, DOB, Compensation From tbone where Id=3;'
      ⎕SQLMY 'GetAllRecords' 102
      ⎕SQLMY 'Exec' 'Delete From tbone Where Id=3;'
102
 3 Salmon       90500.56
1
```

Specify all columns to be inserted with '*':

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'ExecInsertQuery' 'tbone' '*' (3 'Salmon' '1966-01-01' 90500.56)
⍝↑ No column specification needed if data for all columns is provided
⎕SQLMY 'ExecSelectQuery' 'Select Id From tbone;'
⎕SQLMY 'GetAllRecords' 102
⎕SQLMY 'Exec' 'Delete From tbone where Id=3;'
```



```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'ExecInsertQuery' 'tbone' '*' (3 'Salmon' '1966-01-01' 90500.56)
      ⍝ No column specification needed if data for all columns is provided
      ⎕SQLMY 'ExecSelectQuery' 'Select Id From tbone;'
      ⎕SQLMY 'GetAllRecords' 102
      ⎕SQLMY 'Exec' 'Delete From tbone where Id=3;'
102
  1
  2
  3
1
```

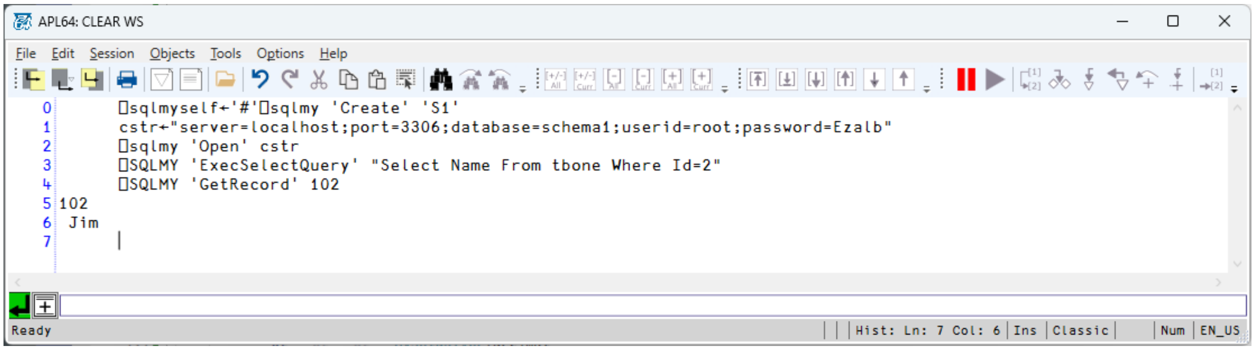
ExecSelectQuery

The ExecSelectQuery action applies to an ⎕sqlmy instance. The action argument provides the SQL Select statement. Depending on the MySql 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.

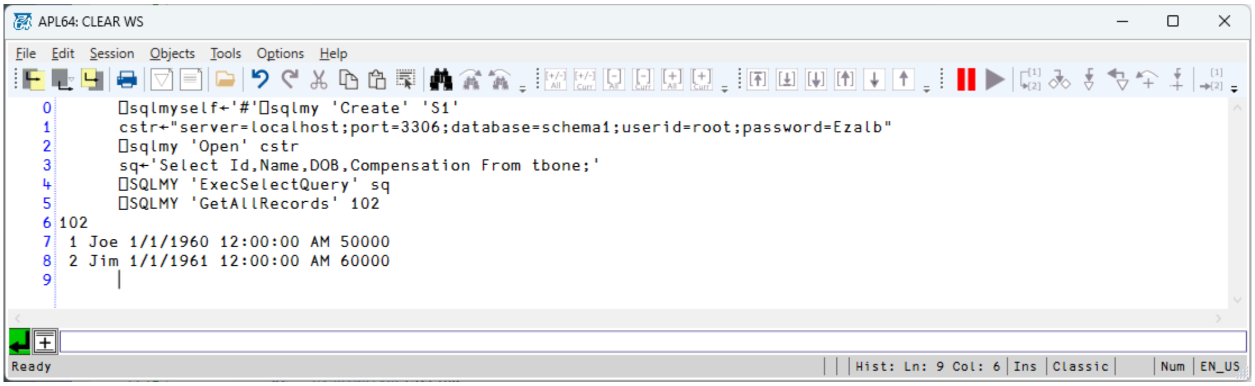
The SQL-format query text argument to the ⎕SQLMy ExecSelectQuery method, should yield only on record set. A query of the form 'query1;…;queryN', is not supported in APL64 and must be separated into individual queries. As an alternative to separate queries, a stored procedure can be created and executed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

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'ExecSelectQuery' "Select Name From tbone Where Id=2"
⎕SQLMY 'GetRecord' 102
```



```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'ExecSelectQuery' "Select Name From tbone Where Id=2"
      ⎕SQLMY 'GetRecord' 102
102
 Jim
```

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
sq←'Select Id,Name,DOB,Compensation From tbone;'
⎕SQLMY 'ExecSelectQuery' sq
⎕SQLMY 'GetAllRecords' 102
```



APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 9 Col: 6 Ins Classic Num EN_US

```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      sq←'Select Id,Name,DOB,Compensation From tbone;'
      ⎕SQLMY 'ExecSelectQuery' sq
      ⎕SQLMY 'GetAllRecords' 102
102
 1 Joe 1/1/1960 12:00:00 AM 50000
 2 Jim 1/1/1961 12:00:00 AM 60000
      |
```

ExecStoredProc

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

Syntax: MatrixOfValues←instanceName⎕SqlMy '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 MySql data types can be represented in APL64, so the data types of stored procedure parameters must be carefully considered. For MySql data types which have no representation in APL64, the stored procedure must cast the parameter values to APL64-compatible data types.

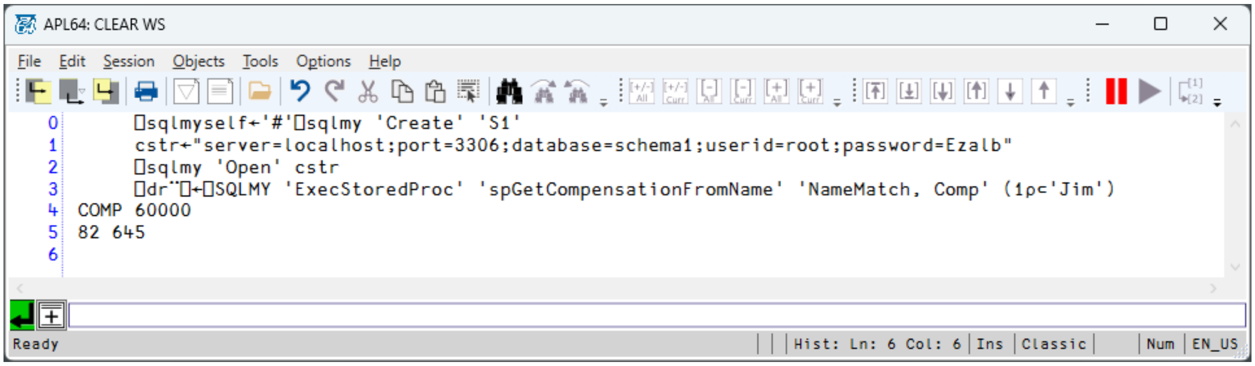
The APL64-compatible data types are Boolean, Double, Int32, Char, Varchar and String.

The result is an APL 2-column 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:

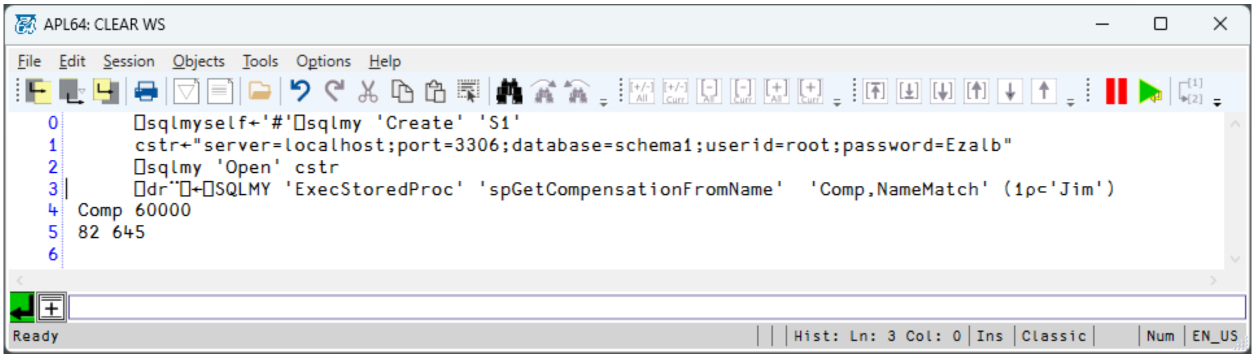
```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕dr¨⎕←⎕SQLMY 'ExecStoredProc' 'spGetCompensationFromName' 'NameMatch, Comp' (1ρ⊂'Jim')
```



```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕dr¨⎕←⎕SQLMY 'ExecStoredProc' 'spGetCompensationFromName' 'NameMatch, Comp' (1ρ⊂'Jim')
COMP 60000
82 645
```

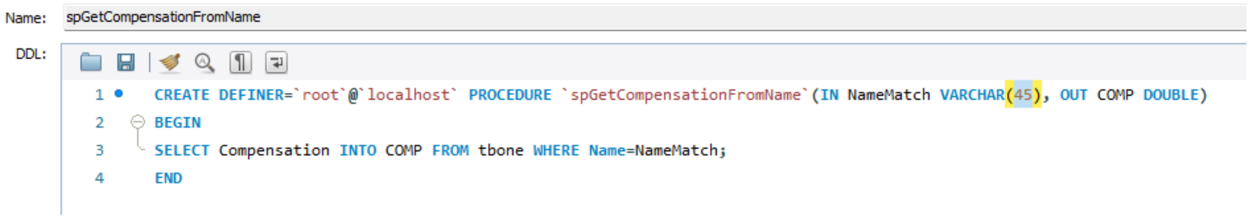
In and Out parameters with Out parameter specified before the In parameter:

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕dr¨⎕←⎕SQLMY 'ExecStoredProc' 'spGetCompensationFromName'  'Comp, NameMatch' (1ρ⊂'Jim')
```



```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕dr¨⎕←⎕SQLMY 'ExecStoredProc' 'spGetCompensationFromName'  'Comp,NameMatch' (1ρ⊂'Jim')
Comp 60000
82 645
```

The stored procedure definition:



Name: spGetCompensationFromName

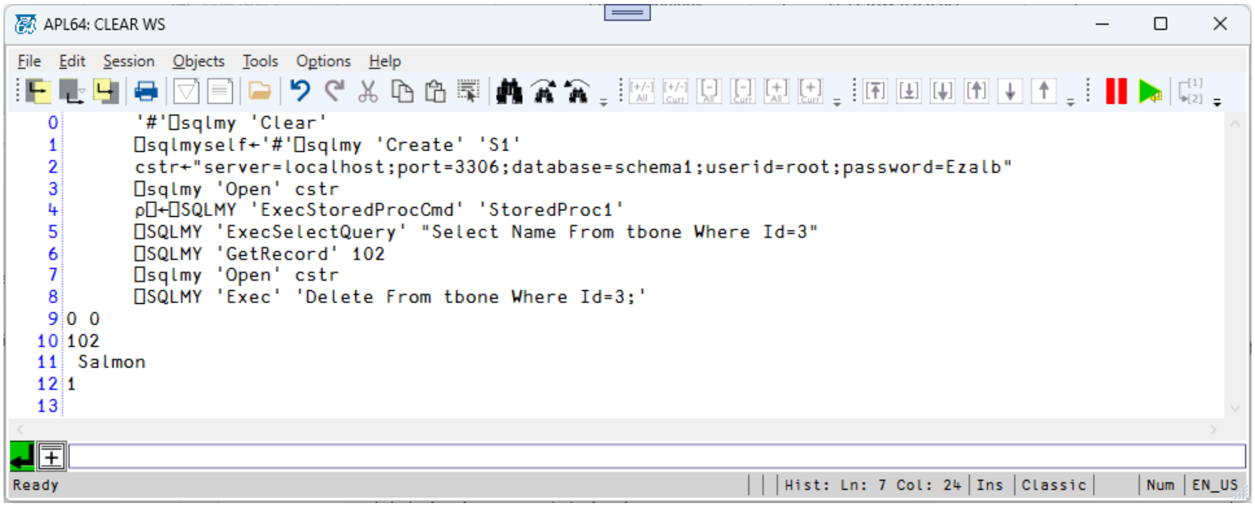
DDL:

```
CREATE DEFINER=`root`@`localhost` PROCEDURE `spGetCompensationFromName`(IN NameMatch VARCHAR(45), OUT COMP DOUBLE)
BEGIN
SELECT Compensation INTO COMP FROM tbone WHERE Name=NameMatch;
END
```

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.

```
'#'⎕sqlmy 'Clear'
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
ρ⎕←⎕SQLMY 'ExecStoredProcCmd' 'StoredProc1'
⎕SQLMY 'ExecSelectQuery' "Select Name From tbone Where Id=3"
⎕SQLMY 'GetRecord' 102
⎕sqlmy 'Open' cstr
⎕SQLMY 'Exec' 'Delete From tbone Where Id=3;'
```



```
      '#'⎕sqlmy 'Clear'
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ρ⎕←⎕SQLMY 'ExecStoredProcCmd' 'StoredProc1'
      ⎕SQLMY 'ExecSelectQuery' "Select Name From tbone Where Id=3"
      ⎕SQLMY 'GetRecord' 102
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'Exec' 'Delete From tbone Where Id=3;'
0 0
102
 Salmon
1
```

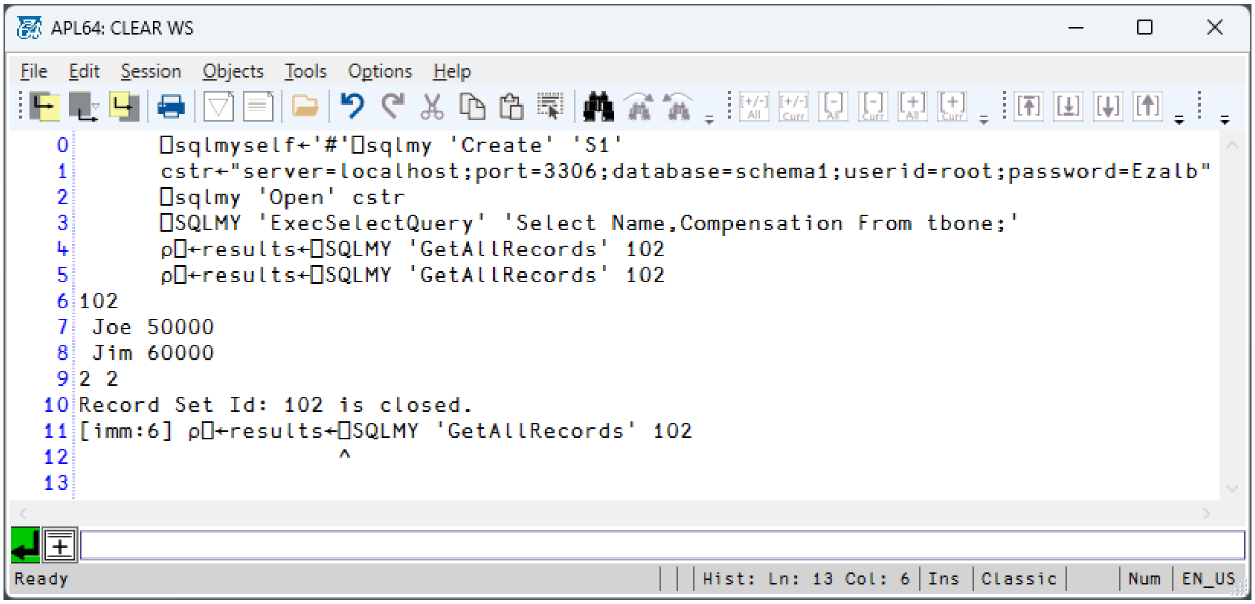
GetAllRecords

This action applies to an ⎕SQLMY 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.

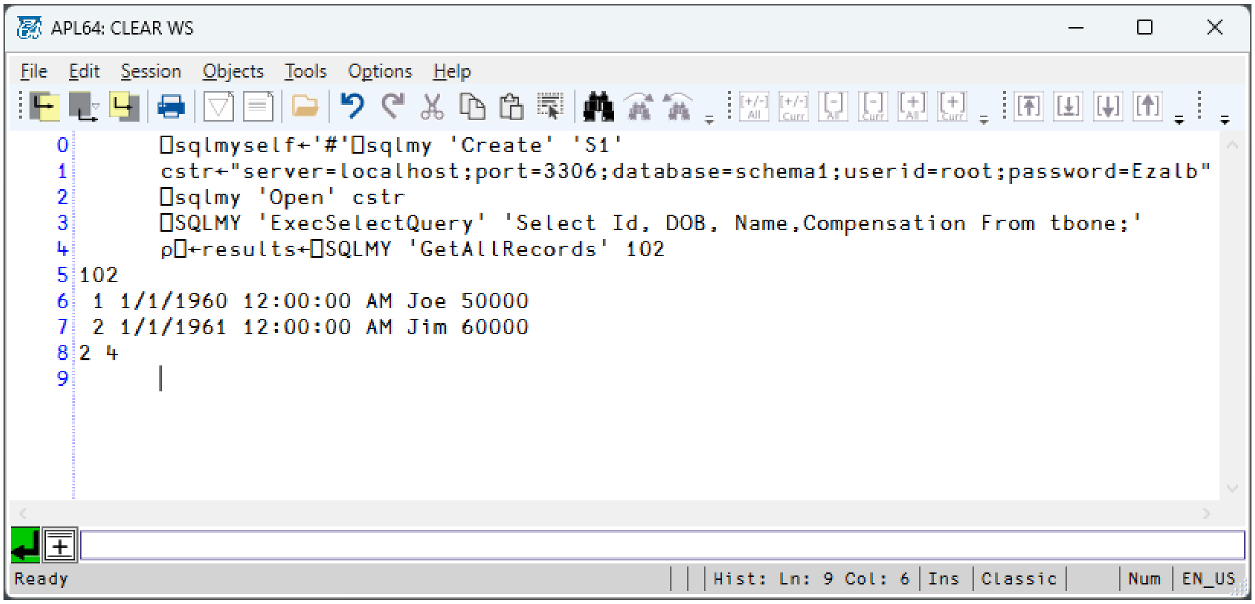
An exception will occur if there are no remaining records to be read.

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'ExecSelectQuery' 'Select Name,Compensation From tbone;'
ρ⎕←results←⎕SQLMY 'GetAllRecords' 102
ρ⎕←results←⎕SQLMY 'GetAllRecords' 102
```



```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'ExecSelectQuery' 'Select Name,Compensation From tbone;'
      ρ⎕←results←⎕SQLMY 'GetAllRecords' 102
      ρ⎕←results←⎕SQLMY 'GetAllRecords' 102
102
 Joe 50000
 Jim 60000
2 2
Record Set Id: 102 is closed.
[imm:6] ρ⎕←results←⎕SQLMY 'GetAllRecords' 102
                   ^
```

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'ExecSelectQuery' 'Select Id, DOB, Name,Compensation From tbone;'
ρ⎕←results←⎕SQLMY 'GetAllRecords' 102
```



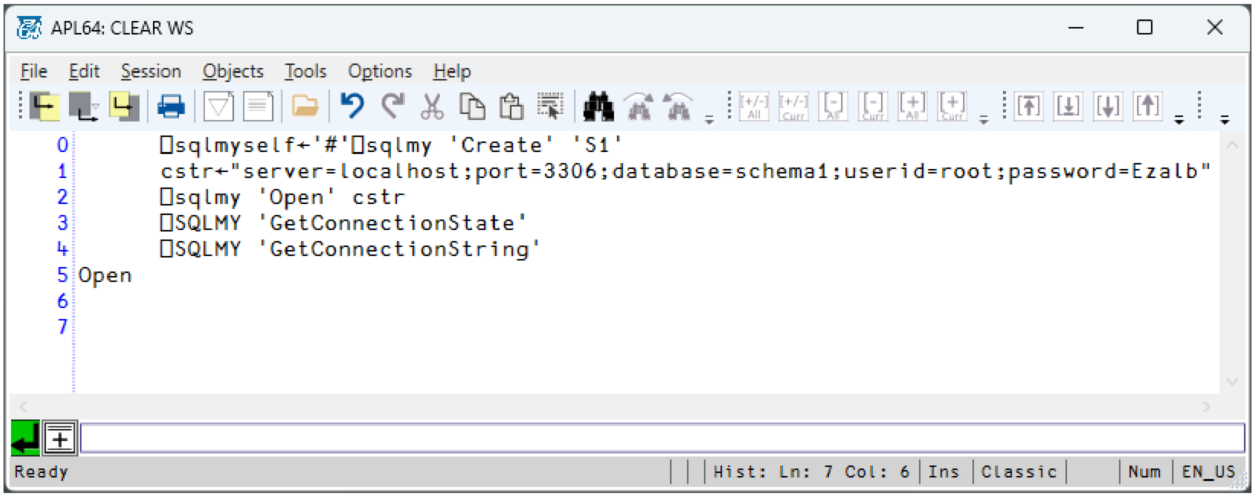
```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'ExecSelectQuery' 'Select Id, DOB, Name,Compensation From tbone;'
      ρ⎕←results←⎕SQLMY 'GetAllRecords' 102
102
 1 1/1/1960 12:00:00 AM Joe 50000
 2 1/1/1961 12:00:00 AM Jim 60000
2 4
```

GetConnectionState

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

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
```

```
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'GetConnectionState'
⎕SQLMY 'GetConnectionString'
```

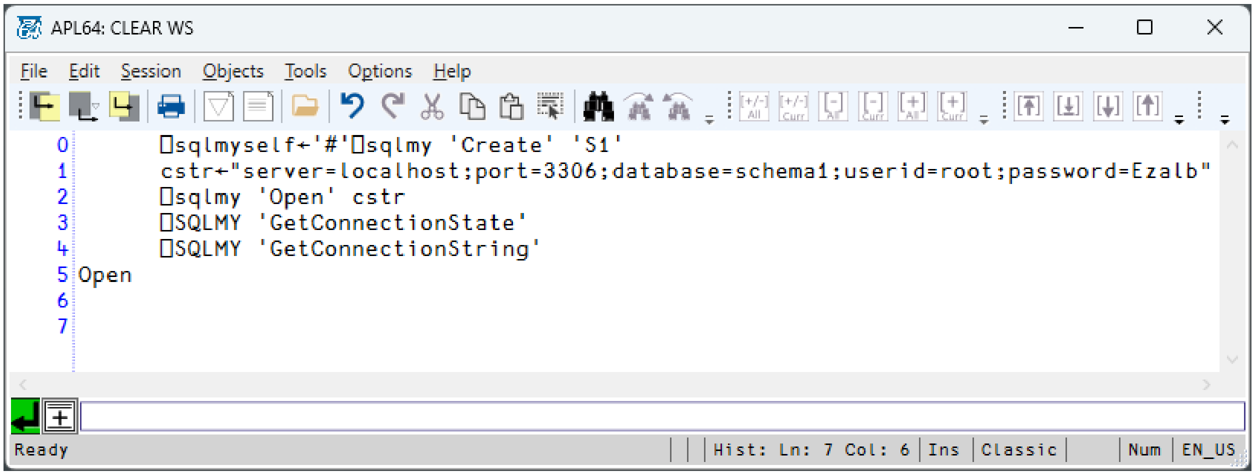


```
        ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
        cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
        ⎕sqlmy 'Open' cstr
        ⎕SQLMY 'GetConnectionState'
        ⎕SQLMY 'GetConnectionString'
Open
```

GetConnectionString

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

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'GetConnectionState'
⎕SQLMY 'GetConnectionString'
```



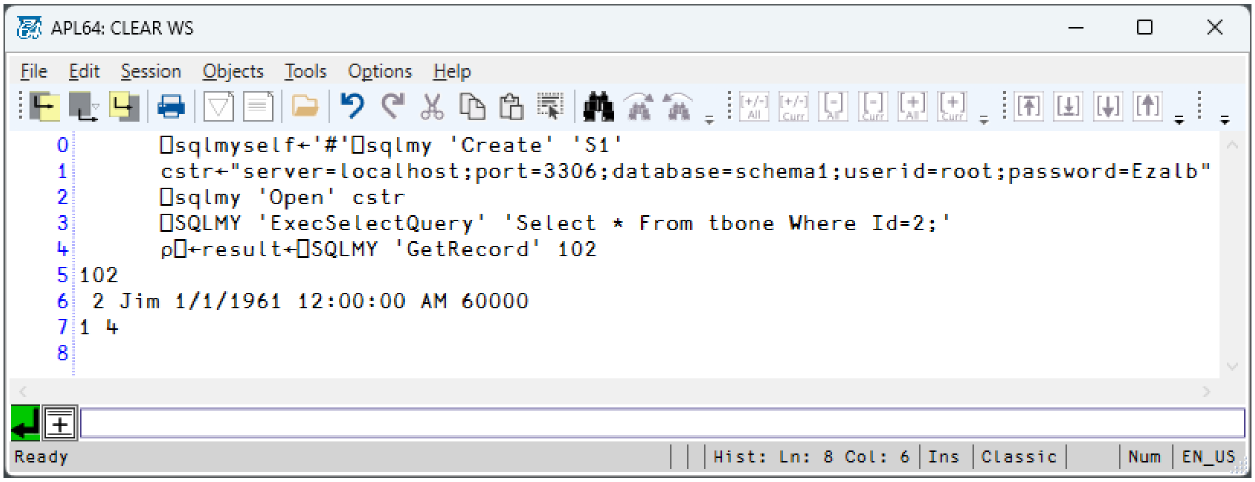
```
        ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
        cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
        ⎕sqlmy 'Open' cstr
        ⎕SQLMY 'GetConnectionState'
        ⎕SQLMY 'GetConnectionString'
Open
```

GetRecord

This action applies to an ⎕SQLMY instance. The right argument is the ⎕SQLMY 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. 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.

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'ExecSelectQuery' 'Select * From tbone Where Id=2;'
ρ⎕←result←⎕SQLMY 'GetRecord' 102
```

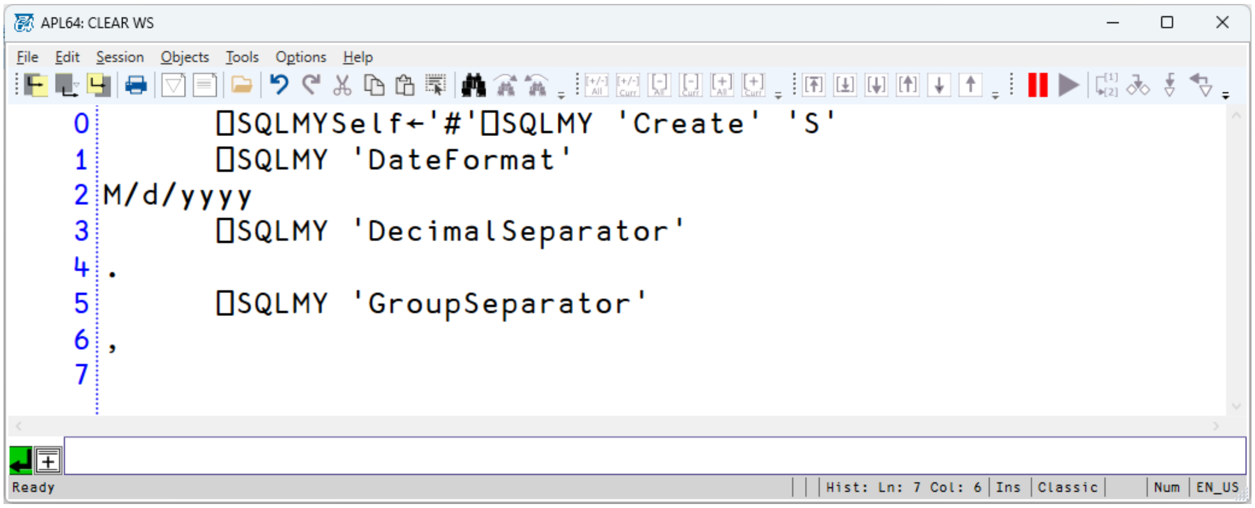


```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'ExecSelectQuery' 'Select * From tbone Where Id=2;'
      ρ⎕←result←⎕SQLMY 'GetRecord' 102
102
 2 Jim 1/1/1961 12:00:00 AM 60000
1 4
```

GroupSeparator

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

```
⎕SQLMYSelf←'#'⎕SQLMY 'Create' 'S'
⎕SQLMY 'DateFormat'
⎕SQLMY 'DecimalSeparator'
⎕SQLMY 'GroupSeparator'
```

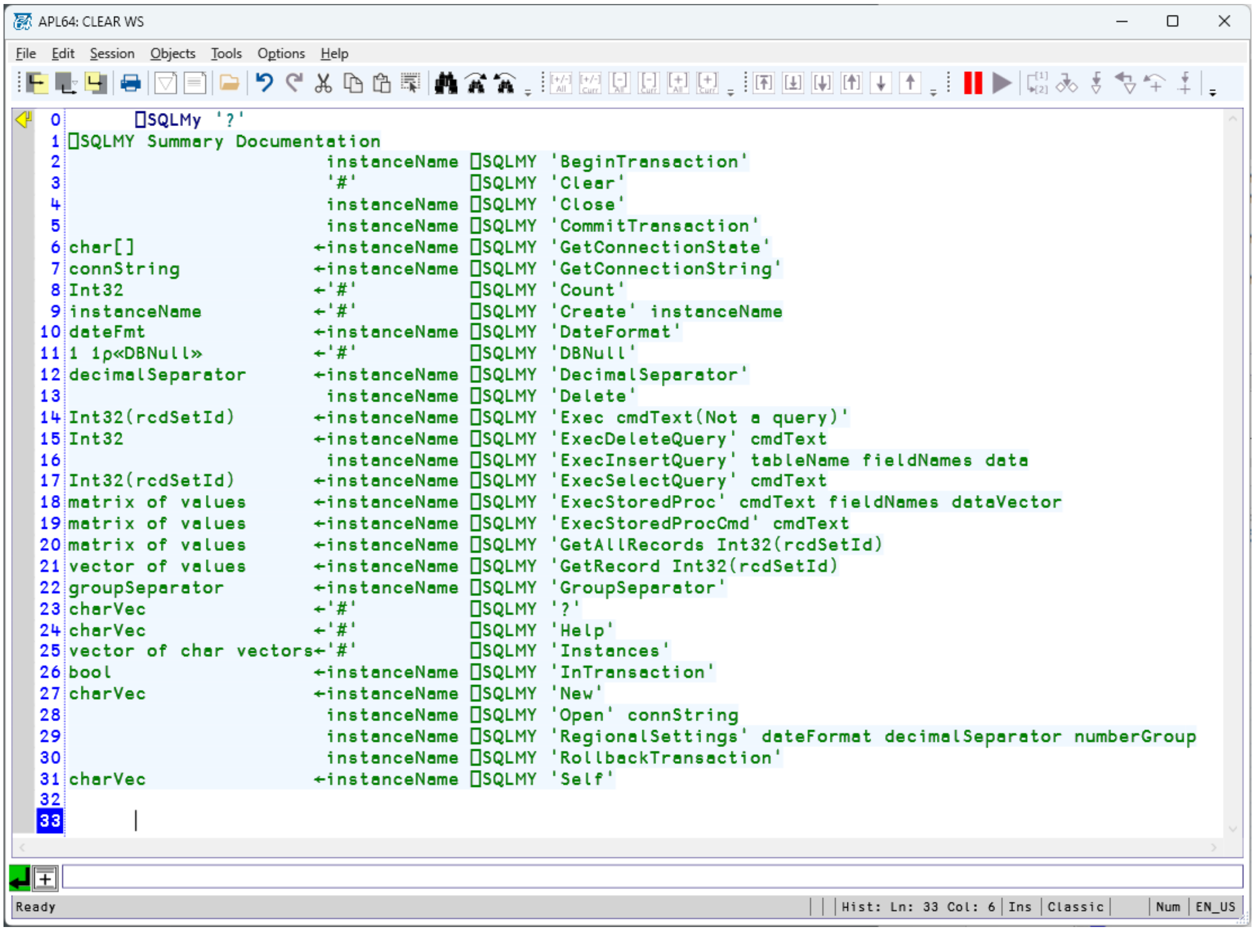


```
      ⎕SQLMYSelf←'#'⎕SQLMY 'Create' 'S'
      ⎕SQLMY 'DateFormat'
M/d/yyyy
      ⎕SQLMY 'DecimalSeparator'
.
      ⎕SQLMY 'GroupSeparator'
,

```

? or Help

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



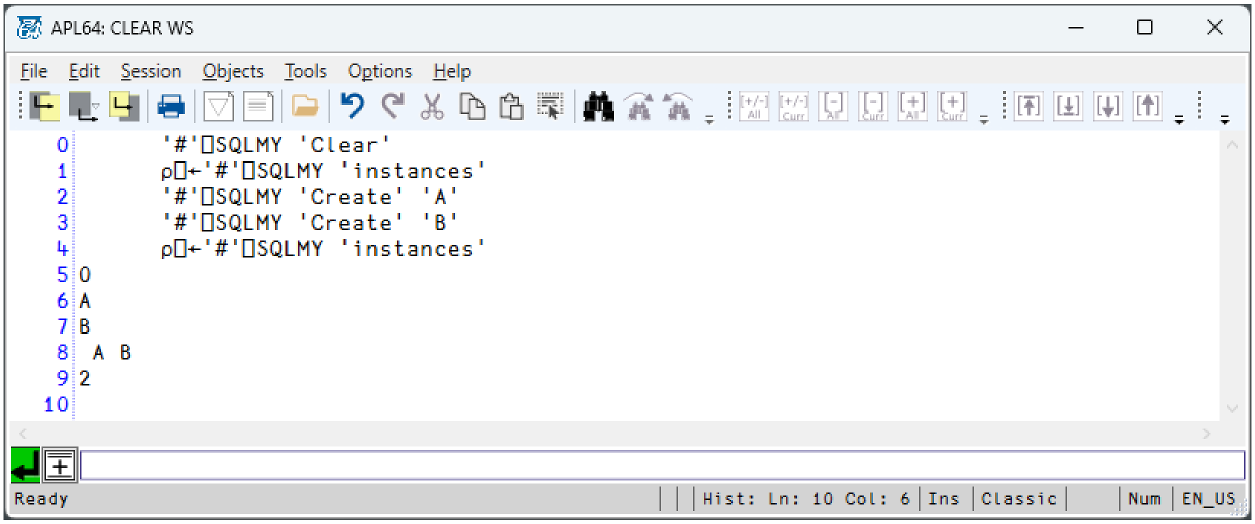
```
      ⎕SQLMy '?'
⎕SQLMY Summary Documentation
                       instanceName ⎕SQLMY 'BeginTransaction'
                       '#'          ⎕SQLMY 'Clear'
                       instanceName ⎕SQLMY 'Close'
                       instanceName ⎕SQLMY 'CommitTransaction'
char[]                ←instanceName ⎕SQLMY 'GetConnectionState'
connString            ←instanceName ⎕SQLMY 'GetConnectionString'
Int32                 ←'#'          ⎕SQLMY 'Count'
instanceName          ←'#'          ⎕SQLMY 'Create' instanceName
dateFmt               ←instanceName ⎕SQLMY 'DateFormat'
1 1ρ«DBNull»          ←'#'          ⎕SQLMY 'DBNull'
decimalSeparator      ←instanceName ⎕SQLMY 'DecimalSeparator'
                       instanceName ⎕SQLMY 'Delete'
Int32(rcdSetId)       ←instanceName ⎕SQLMY 'Exec cmdText(Not a query)'
Int32                 ←instanceName ⎕SQLMY 'ExecDeleteQuery' cmdText
                       instanceName ⎕SQLMY 'ExecInsertQuery' tableName fieldNames data
Int32(rcdSetId)       ←instanceName ⎕SQLMY 'ExecSelectQuery' cmdText
matrix of values      ←instanceName ⎕SQLMY 'ExecStoredProc' cmdText fieldNames dataVector
matrix of values      ←instanceName ⎕SQLMY 'ExecStoredProcCmd' cmdText
matrix of values      ←instanceName ⎕SQLMY 'GetAllRecords Int32(rcdSetId)
vector of values      ←instanceName ⎕SQLMY 'GetRecord Int32(rcdSetId)
groupSeparator        ←instanceName ⎕SQLMY 'GroupSeparator'
charVec               ←'#'          ⎕SQLMY '?'
charVec               ←'#'          ⎕SQLMY 'Help'
vector of char vectors←'#'          ⎕SQLMY 'Instances'
bool                  ←instanceName ⎕SQLMY 'InTransaction'
charVec               ←instanceName ⎕SQLMY 'New'
                       instanceName ⎕SQLMY 'Open' connString
                       instanceName ⎕SQLMY 'RegionalSettings' dateFormat decimalSeparator numberGroup
                       instanceName ⎕SQLMY 'RollbackTransaction'
charVec               ←instanceName ⎕SQLMY 'Self'

```

Instances

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

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

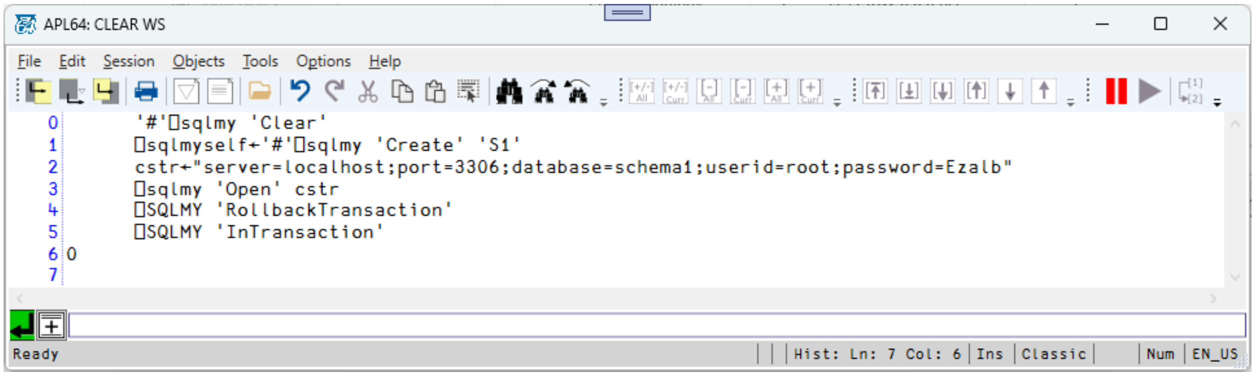


```
      '#'⎕SQLMY 'Clear'
      ρ⎕←'#'⎕SQLMY 'instances'
      '#'⎕SQLMY 'Create' 'A'
      '#'⎕SQLMY 'Create' 'B'
      ρ⎕←'#'⎕SQLMY 'instances'
0
A
B
 A  B
2
```

InTransaction

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

```
'#'⎕sqlmy 'Clear'
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'RollbackTransaction'
⎕SQLMY 'InTransaction'
```

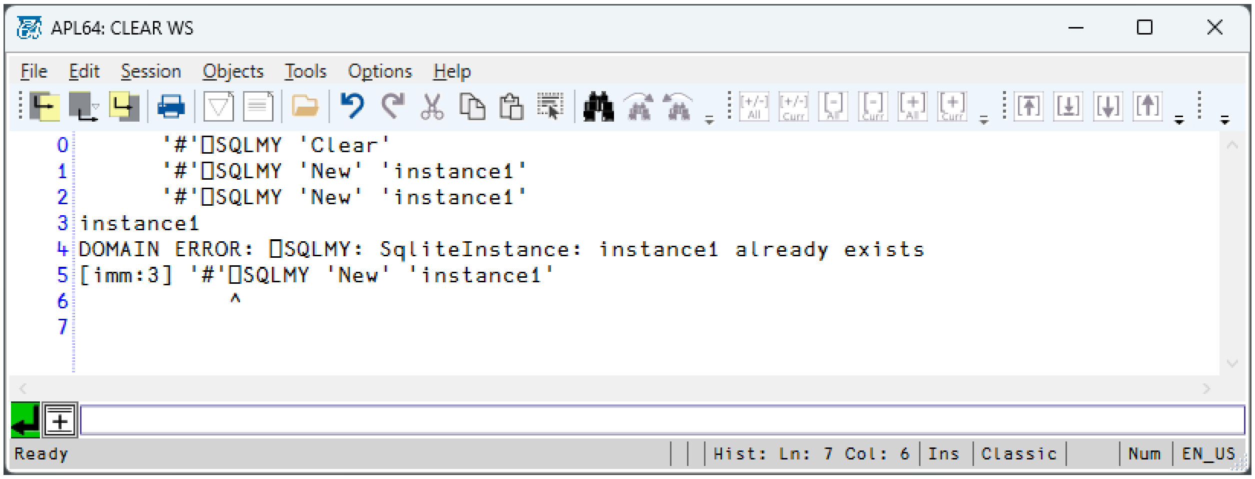


```
      '#'⎕sqlmy 'Clear'
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'RollbackTransaction'
      ⎕SQLMY 'InTransaction'
0
```

New

This action is performed on the ⎕SQLMY object. The New action will create an ⎕SQLMY instance with a user-provided name in the right argument. Multiple ⎕SQLMY instances are possible in the same APL64 instance, so that multiple MySql databases may be conveniently accessed. The New action will fail if the named instance already exists. The New action for a particular MySql database is generally used once in an APL64 instance. The New action does not open a connection to an MySql database, use the Open action for that purpose. The result of a successful 'New' action is a text vector containing the ⎕SQLMY instance name.

```
'#'⎕SQLMY 'Clear'
'#'⎕SQLMY        'New'        'instance1'
'#'⎕SQLMY 'New' 'instance1'
```

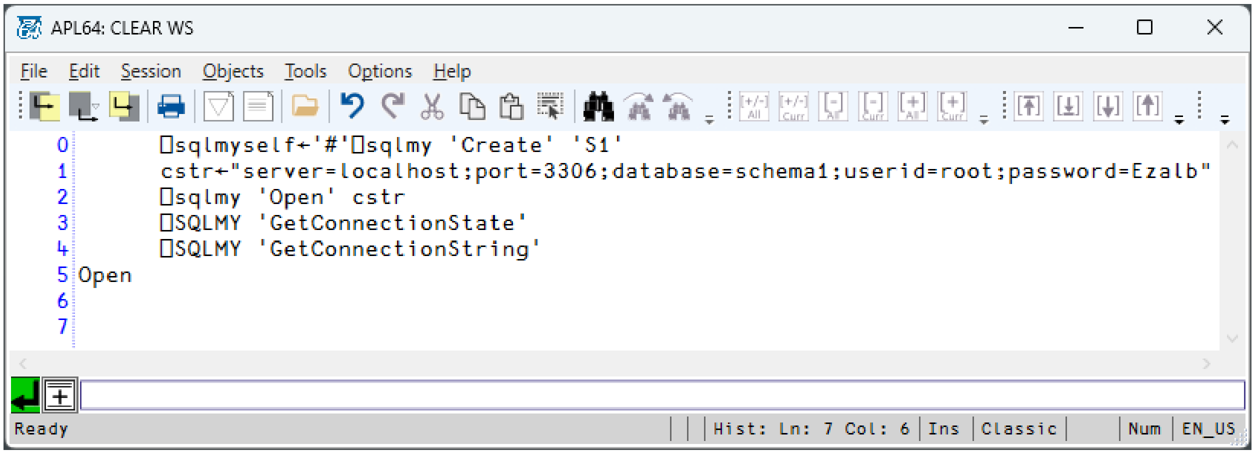


```
      '#'⎕SQLMY 'Clear'
      '#'⎕SQLMY 'New' 'instance1'
      '#'⎕SQLMY 'New' 'instance1'
instance1
DOMAIN ERROR: ⎕SQLMY: SqliteInstance: instance1 already exists
[imm:3] '#'⎕SQLMY 'New' 'instance1'
             ^
```

Open

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

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'GetConnectionState'
⎕SQLMY 'GetConnectionString'
```

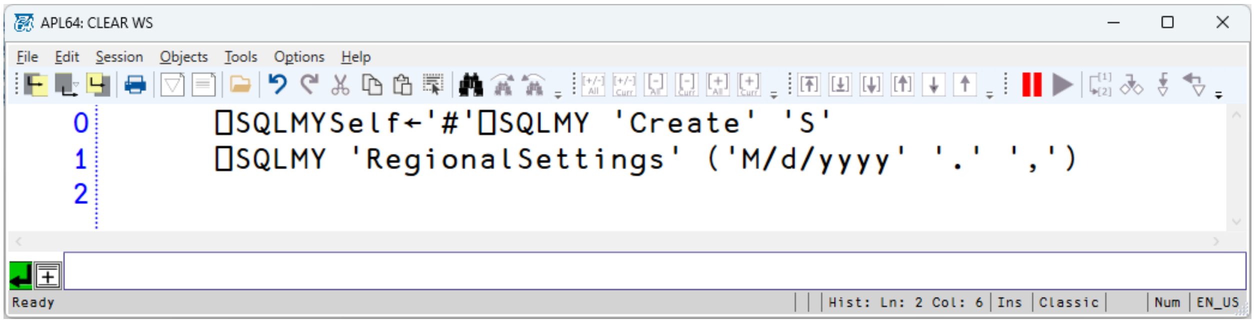


```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'GetConnectionState'
      ⎕SQLMY 'GetConnectionString'
Open
```

RegionalSettings

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

```
⎕SQLMYSelf←'#'⎕SQLMY 'Create' 'S'
⎕SQLMY 'RegionalSettings'  'M/d/yyyy'  '.'  ','
```

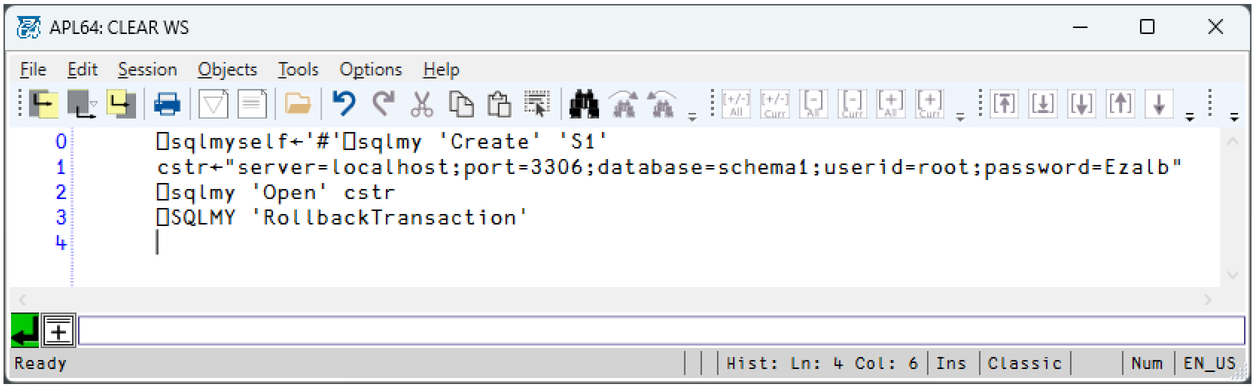


```
      ⎕SQLMYSelf←'#'⎕SQLMY 'Create' 'S'
      ⎕SQLMY 'RegionalSettings' ('M/d/yyyy' '.' ',')
```

RollbackTransaction

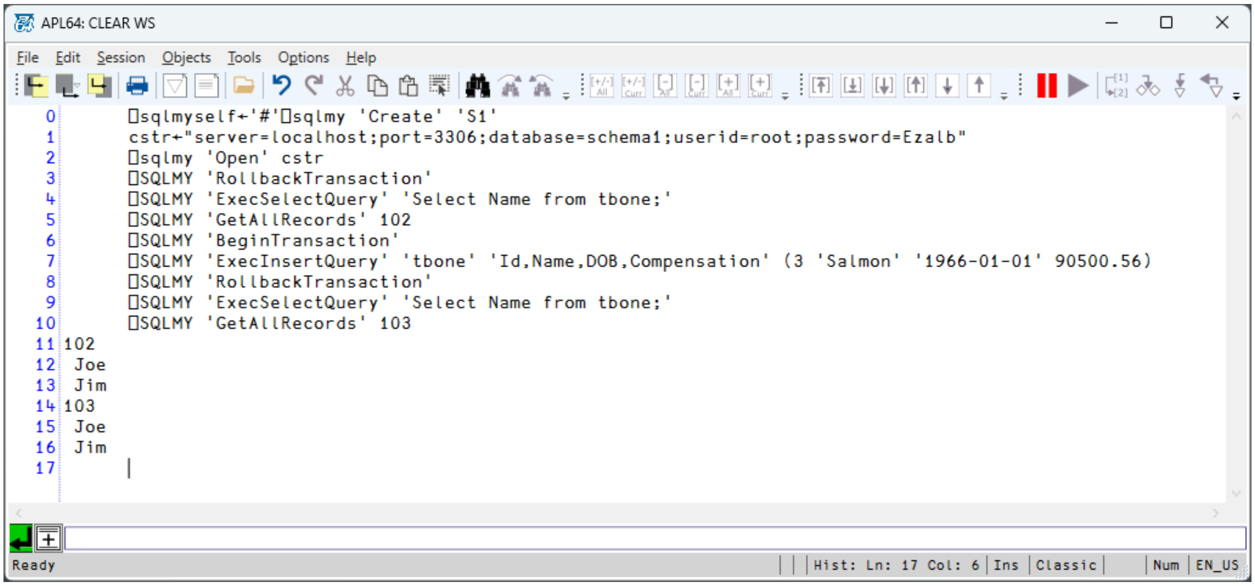
This action applies to an ⎕SQLMY instance. The RollbackTransaction action is used to cancel any pending SQL operations which are included in the current SQL transaction. The RollbackTransaction action has no result. The RollbackTransaction action has no effect if no SQL transaction is in progress.

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'RollbackTransaction'
```


APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 4 Col: 6 Ins Classic Num EN_US

```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'RollbackTransaction'
      
```

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'RollbackTransaction'
⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
⎕SQLMY 'GetAllRecords' 102
⎕SQLMY 'BeginTransaction'
⎕SQLMY 'ExecInsertQuery' 'tbone' 'Id,Name,DOB,Compensation' (3 'Salmon' '1966-01-01' 90500.56)
⎕SQLMY 'RollbackTransaction'
⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
⎕SQLMY 'GetAllRecords' 103
```

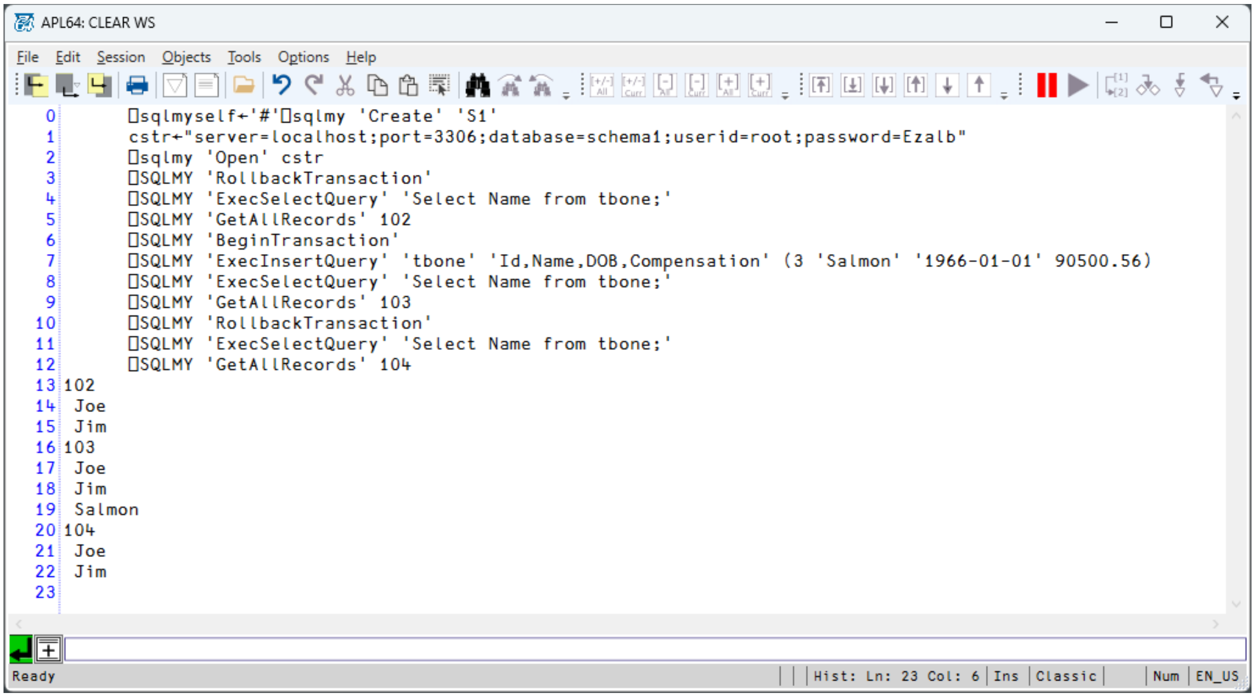

APL64: CLEAR WS
File Edit Session Objects Tools Options Help
Ready
Hist: Ln: 17 Col: 6 Ins Classic Num EN_US

```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'RollbackTransaction'
      ⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
      ⎕SQLMY 'GetAllRecords' 102
      ⎕SQLMY 'BeginTransaction'
      ⎕SQLMY 'ExecInsertQuery' 'tbone' 'Id,Name,DOB,Compensation' (3 'Salmon' '1966-01-01' 90500.56)
      ⎕SQLMY 'RollbackTransaction'
      ⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
      ⎕SQLMY 'GetAllRecords' 103
102
 Joe
 Jim
103
 Joe
 Jim
      
```

```
⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
⎕sqlmy 'Open' cstr
⎕SQLMY 'RollbackTransaction'
⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
⎕SQLMY 'GetAllRecords' 102
⎕SQLMY 'BeginTransaction'
```

```
⎕SQLMY 'ExecInsertQuery' 'tbone' 'Id,Name,DOB,Compensation' (3 'Salmon' '1966-01-01' 90500.56)
⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
⎕SQLMY 'GetAllRecords' 103
⎕SQLMY 'RollbackTransaction'
⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
⎕SQLMY '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:

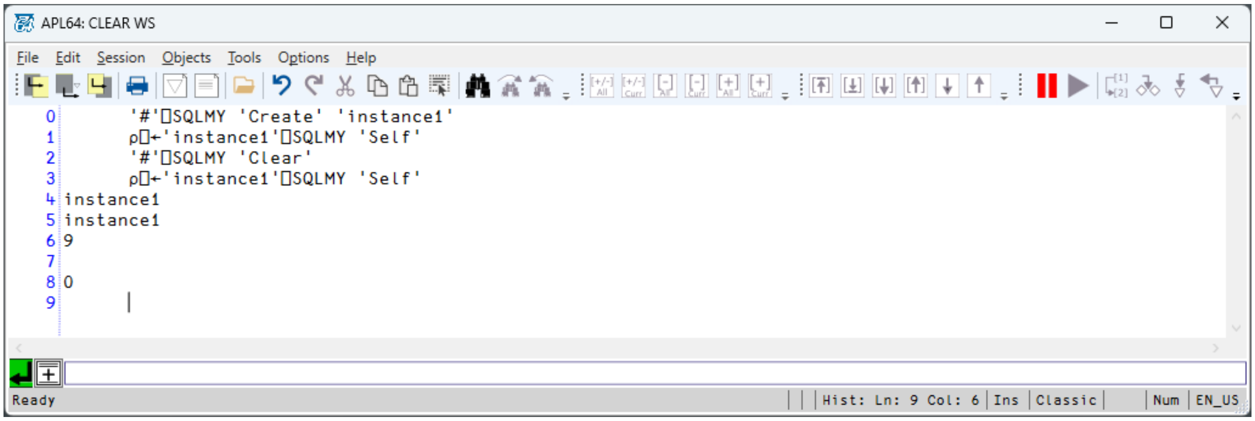


```
      ⎕sqlmyself←'#'⎕sqlmy 'Create' 'S1'
      cstr←"server=localhost;port=3306;database=schema1;userid=root;password=Ezalb"
      ⎕sqlmy 'Open' cstr
      ⎕SQLMY 'RollbackTransaction'
      ⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
      ⎕SQLMY 'GetAllRecords' 102
      ⎕SQLMY 'BeginTransaction'
      ⎕SQLMY 'ExecInsertQuery' 'tbone' 'Id,Name,DOB,Compensation' (3 'Salmon' '1966-01-01' 90500.56)
      ⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
      ⎕SQLMY 'GetAllRecords' 103
      ⎕SQLMY 'RollbackTransaction'
      ⎕SQLMY 'ExecSelectQuery' 'Select Name from tbone;'
      ⎕SQLMY 'GetAllRecords' 104
102
 Joe
 Jim
103
 Joe
 Jim
 Salmon
104
 Joe
 Jim
```

Self

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

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



```
      '#'⎕SQLMY 'Create' 'instance1'
      ρ⎕←'instance1'⎕SQLMY 'Self'
      '#'⎕SQLMY 'Clear'
      ρ⎕←'instance1'⎕SQLMY 'Self'
instance1
instance1
9

0
```

Learn Structured Query Language

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