Learn SQL
Ajay Askoolum

Learn SQL

I invite you to add another skill, namely, Structured Query Language (SQL), not only to extend the scope of APL applications but also to add flexibility to the way APL handles large volumes of data.

This will empower you to separate your application data, from a proprietary format, namely, component files, to an industry standard format, namely, a relational database. This will allow you to share—acquire and transfer—data with considerable ease and independently of APL.

All the resources you will need to learn SQL is available as free downloads. You will need the following:

1. A Relational Database Management System (RDBMS)
2. A graphical user interface to interact with the RDBMS: SQL Server Management Studio (SSMS).
3. Learn SQL; for this, you will need sample database(s) and a tutorial linked to the sample database(s) and covering the SQL spectrum or domain.
4. A means of acquiring and transferring data between APL+Win and the RDBMS. The drivers and providers are installed automatically with the RDBMS enabling APL+Win to use ActiveX Data Object (ADO) or newer technology with the C# Script Engine.

The key incentive is that you can acquire SQL skills at your own pace.

Objective

Although there are several industry-wide standards, SQL has several dialects. In this guideline, you will start off with transact SQL or TSQL. You can learn SQL on an *as needs arise* basis. The world-wide web is a rich source for SQL tutorials, worked examples, and a platform for seeking solutions.

The APL+Win forum http://forum.apl2000.com/viewforum.php?f=17 is a good place to raise issues for APL+Win specific responses.

RDBMS

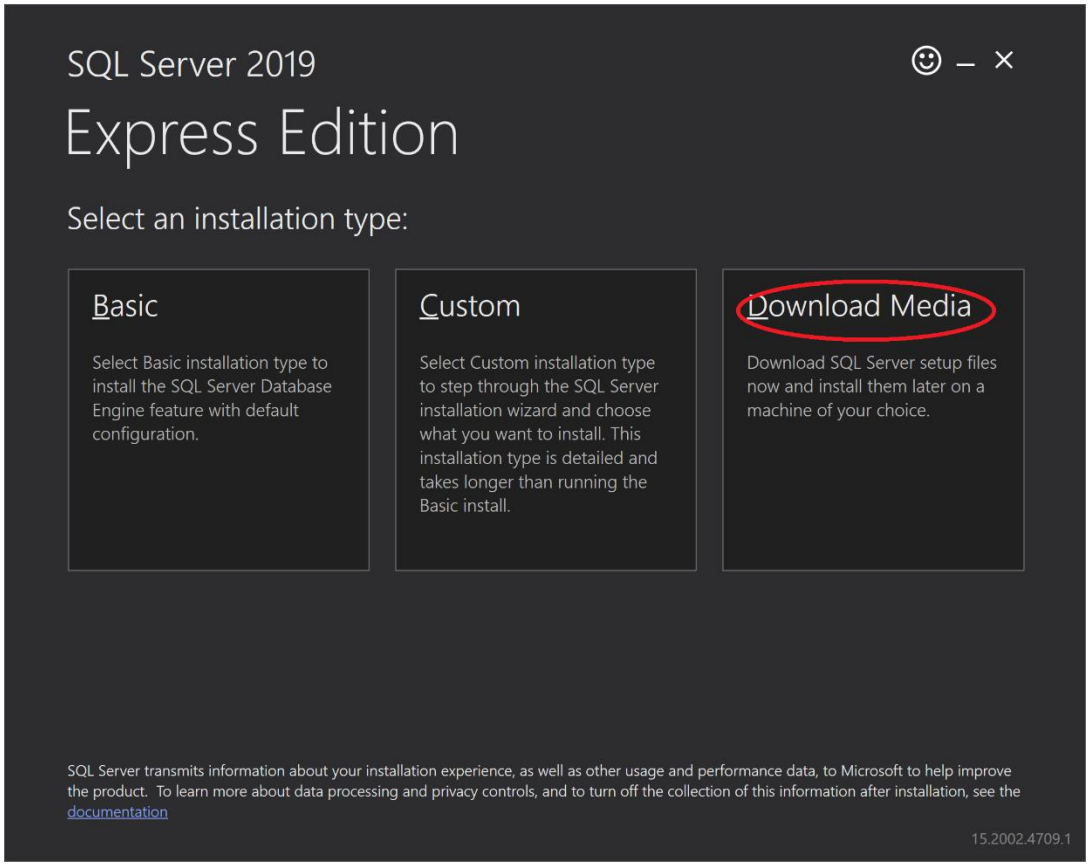
Microsoft offer a free and upward compatible Relational Database Management System (RDBMS) with a minimal footprint, LocalDB at the following link:

https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/sql-server-express-localdb?view=sql-server-ver15

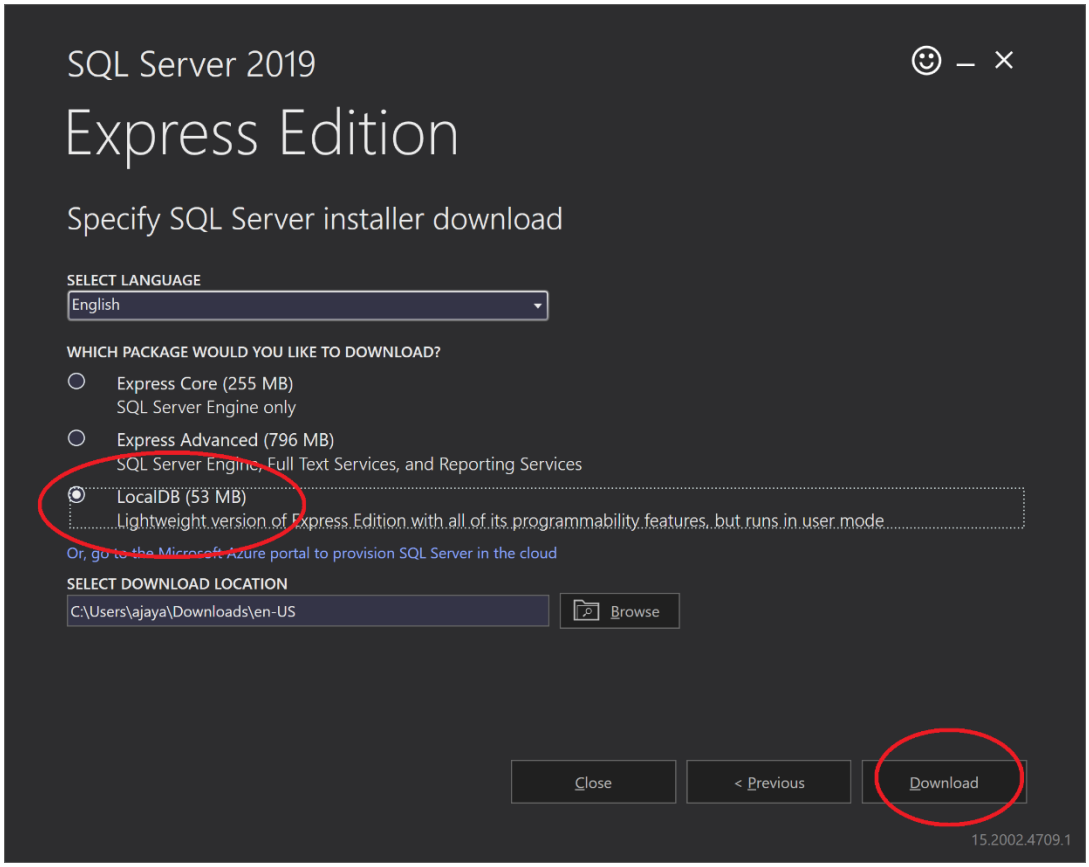
At that page, click on the **SQL Server Express 2019** link which resolves to the following download link:

https://go.microsoft.com/fwlink/?LinkID=866658

On the dialogue, shown below, select Download Media.

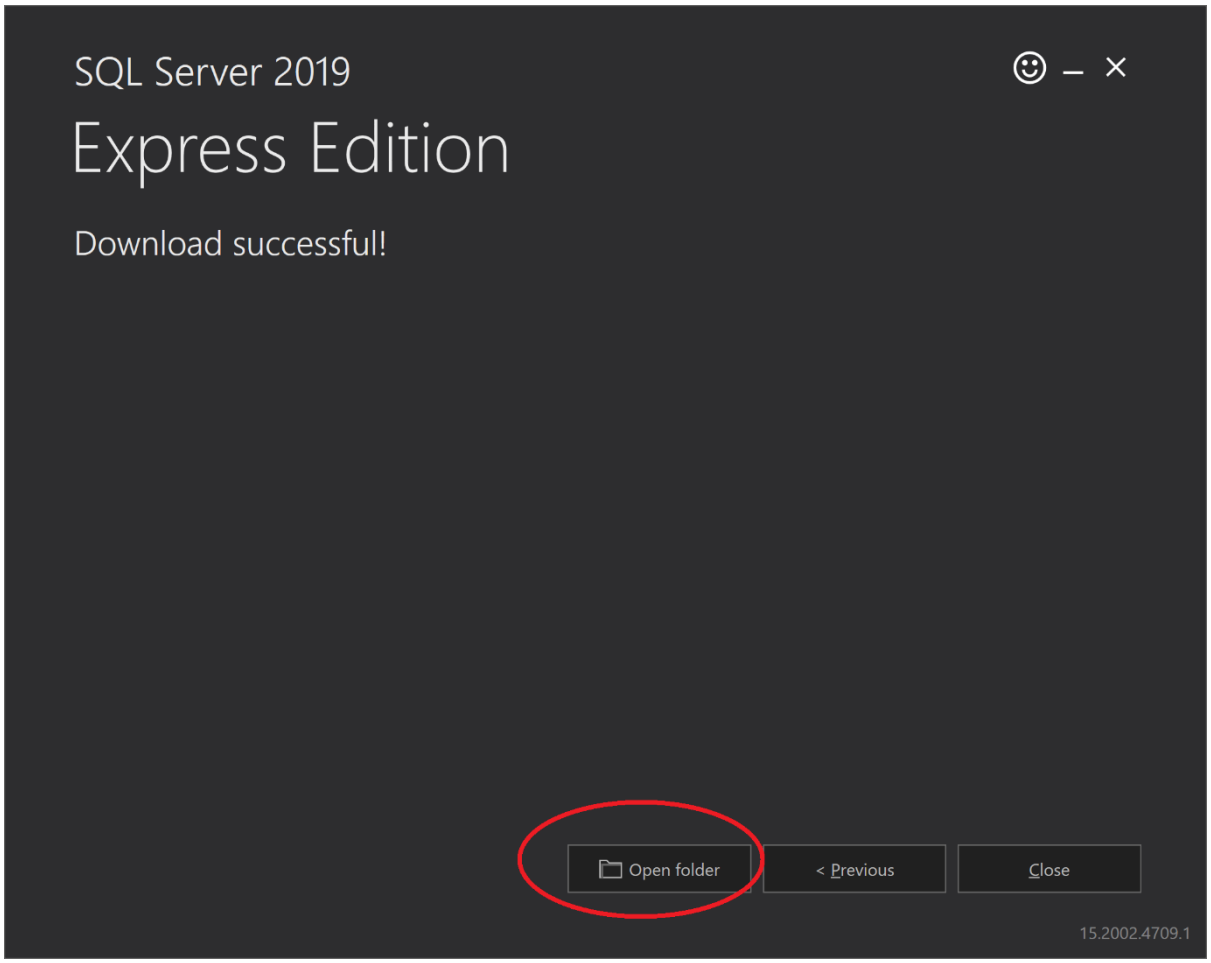


This will download **SQL2019-SSEI-Expr.exe** to your computer. This will launch another dialogue, shown below:



Select LocalDB and click the **Download** button. This will download **SqlLocalDB.msi** to your computer.

When the download completes, another dialogue, see below, appears; click the **Open Folder** button



Double-click on **SqlLocalDB.msi** to install LocalDB. This will install LocalDB.

RDBMS Administration

RDBMS require administration. LocalDB requires some degree of administration.

You need to locate **SqlLocalDB.exe** and set up a shortcut to open the command prompt at that location. If you did not override the default location option, you need to locate it will be C:\Program Files\Microsoft SQL Server\150\Tools\Binn. As its target, your shortcut will have the following:

C:\Windows\System32\cmd.exe "C:\Program Files\Microsoft SQL Server\150\Tools\Binn\SqlLocalDB.exe"

With the command prompt, running **SQLLocalDB.EXE** with parameter **/?** provides basic syntax guidelines and indicates the extent of the minimal administrative tasks.:

```
Microsoft (R) SQL Server Express LocalDB Command Line Tool
Version 15.0.2000.5
Copyright (c) Microsoft Corporation.  All rights reserved.
Usage: SqlLocalDB operation [parameters...]

Operations:
 -?                                      Prints this information
 create|c ["instance name" [version-    Creates a new LocalDB instance with
 number] [-s]]                          a specified name and version
 -s                                     starts the new LocalDB instance
                                        after it's created
```

```
delete|d ["instance name"]              Deletes the LocalDB instance with
                                        the specified name

start|s ["instance name"]               Starts the LocalDB instance with
                                        the specified name

stop|p ["instance name" [-i|-k]]        Stops the LocalDB instance with
                                        the specified name, after current
                                        queries finish

-i                                      request LocalDB instance shutdown
                                        with NOWAIT option
-k                                      kills LocalDB instance process
                                        without contacting it

share|h ["owner SID or account"]        Shares the specified private
"private name" "shared name"            instance using the specified shared
                                        name.    If the user SID or account
                                        name is omitted, it defaults to
                                        current user.
unshare|u ["shared name"]               Stops the sharing of the specified
                                        shared LocalDB instance.
info|i                                  If the [version-number] parameter
                                        is omitted, it defaults to the
                                        latest LocalDB version installed in
                                        the system.

                                        Lists all existing LocalDB
                                        instances owned by the current user
                                        and all shared LocalDB instances.
info|i "instance name"                   Prints the information about the
                                        specified LocalDB instance

versions|v                              Lists all LocalDB versions
                                        installed on the computer.
trace|t on|off                          Turns tracing on and off
```

```
SqlLocalDB treats spaces as delimiters. It is necessary to surround
instance names that contain spaces and special characters with quotes.

For example:

   SqlLocalDB create "My LocalDB Instance"

The instance name can sometimes be omitted, as indicated above, or

specified as "". In this case, the reference is to the default LocalDB

instance "MSSQLLocalDB".
```

Further details on SQLLocalDB.EXE are found at:

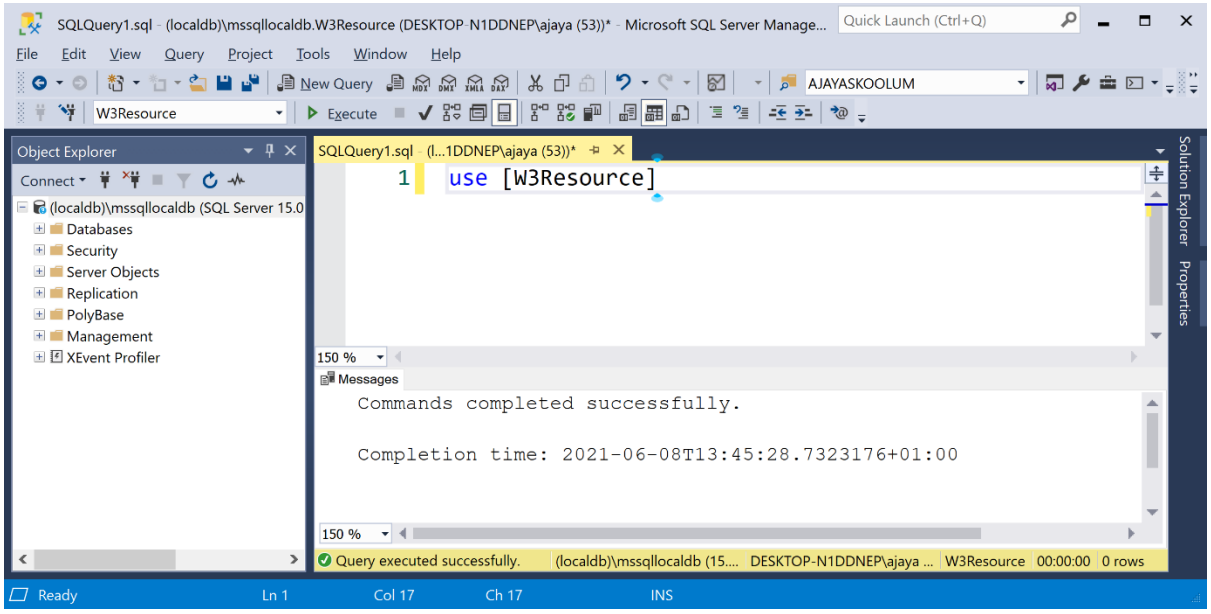
https://docs.microsoft.com/en-us/sql/tools/sqllocaldb-utility?view=sql-server-ver15

SSMS

SQL Server Management Studio provides a graphical interface for managing LocalDB. You can download SSMS **SSMS-Setup-ENU.exe** from:

https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15

SSMS is an intuitive and feature-rich graphical user interface for creating and testing SQL script[1]s for managing data.



Much like the APL interactive session, you enter SQL scripts in one window, click the Execute icon and see the results in another window.

Getting started with SQL

There are innumerable SQL tutorials available on the world-wide web. The following provides a tutorial in a simple and easy to understand manner:

https://www.w3resource.com/sql/tutorials.php

Another resource can be found at https://www.w3schools.com/sql/default.asp ; this allows hands-on learning as it allows SQL execution in the browser.

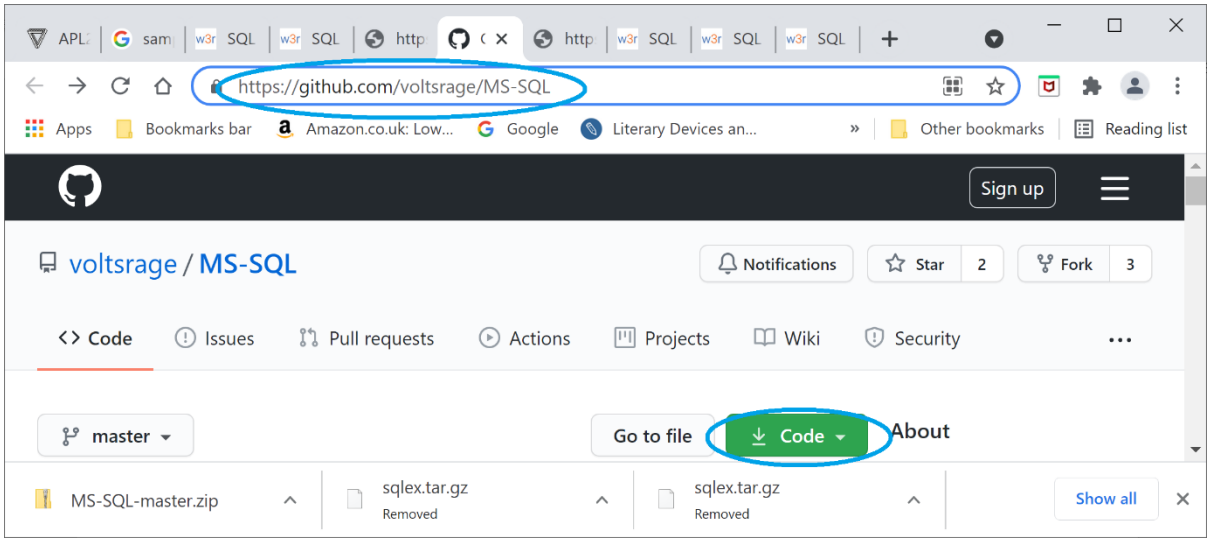
Sample database

In order to practice SQL, you need a database to test your code against since it will be impossible to create your own data without knowledge of SQL. The tutorial at https://www.w3resource.com/sql/tutorials.php is based on a PostgreSQL RDBMS.

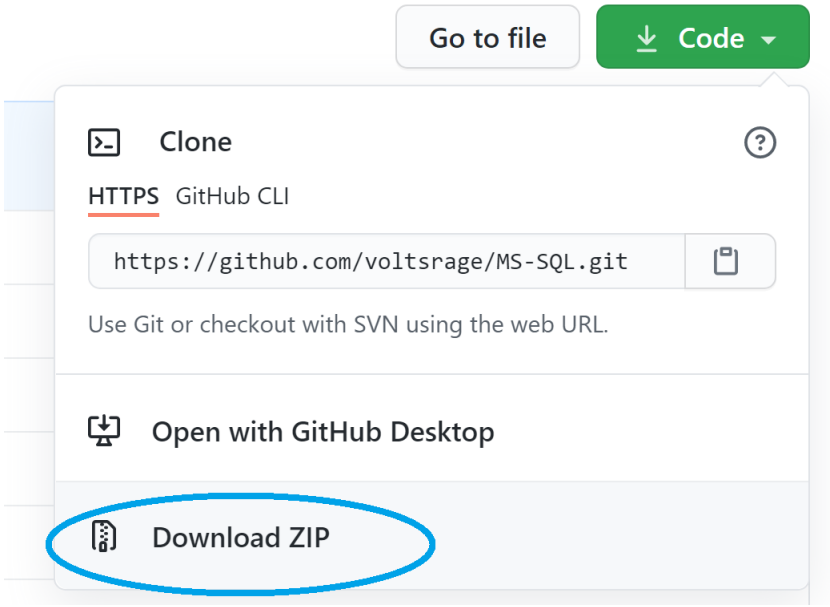
Fortunately, https://github.com/voltsrage/ms-sql provides the scripts for creating the SQL Server version of the PostgreSQL database used by the https://www.w3resource.com/sql/tutorials.php tutorial.

Navigate to https://github.com/voltsrage/ms-sql and click on **Code**—see below—to download scripts for all the SQL Server objects:

[1] For APL, you will need scripts that you have already tested elsewhere as APL does not have facilities for debugging SQL statements..



This provides an option to download a ZIP archive of the scripts, **MS-SQL-master.zip** by clicking on **Download Zip**.



The zip file contains this particular script, **W3ResourceDB.sql**. Launch SSMS, open **W3ResourceDB.sql** and simply click Execute to create the sample database used in the tutorial.

The script is using SQL Server 2017. Above, you have downloaded LOCALDB 2019. You may make the following changes to **W3ResourceDB.sql** before executing it.

First, on Lines[]7 and Line[11], change

C:\Program Files\Microsoft SQL Server\MSSQL14.MSSQLSERVER\MSSQL\DATA

to point to a location of your choice. The location you specify must exist.

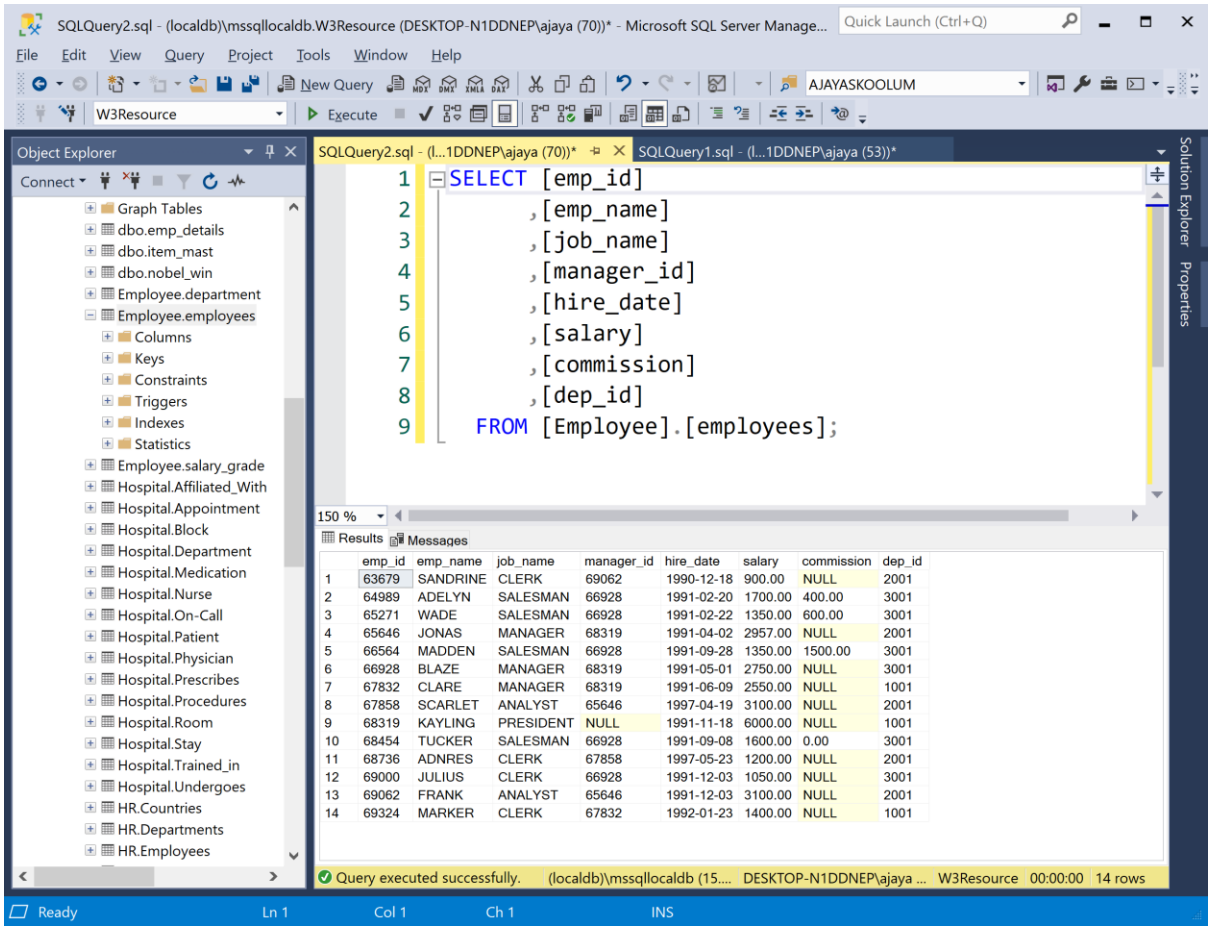
Second, on line [11] , change 140 to 150

ALTER DATABASE [W3Resource] SET COMPATIBILITY_LEVEL = 140

Getting data with APL+Win

There are several ways for returning RDBMS data to the APL workspace.

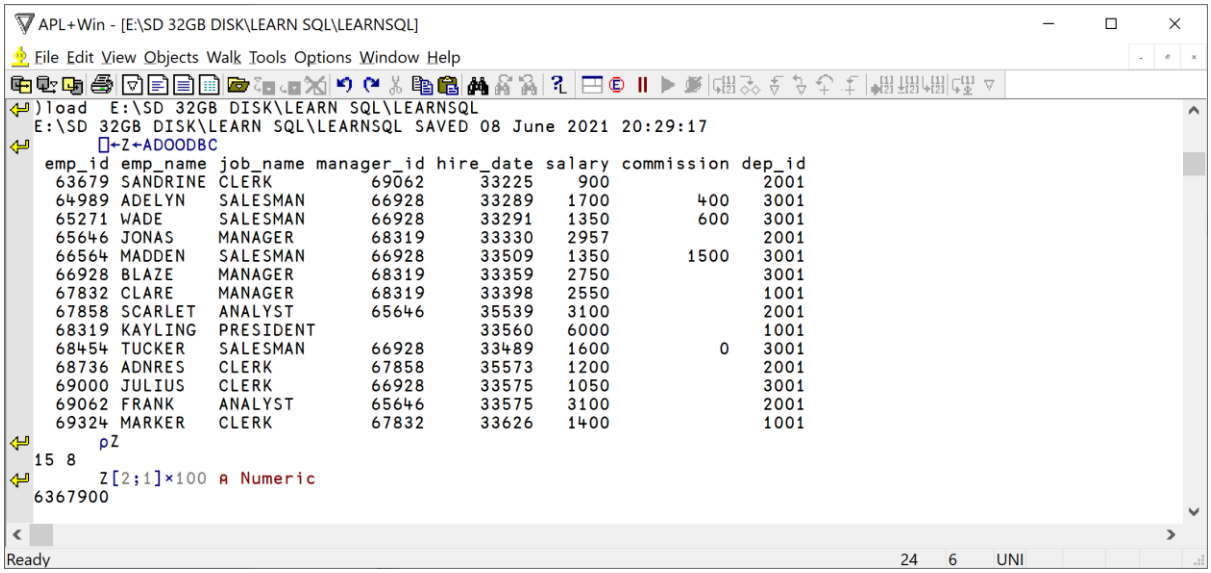
For reference, See below for a view of one table in the W3Resource database.



Using ODBC

The following function retrieves the same information using an ODBC connection:

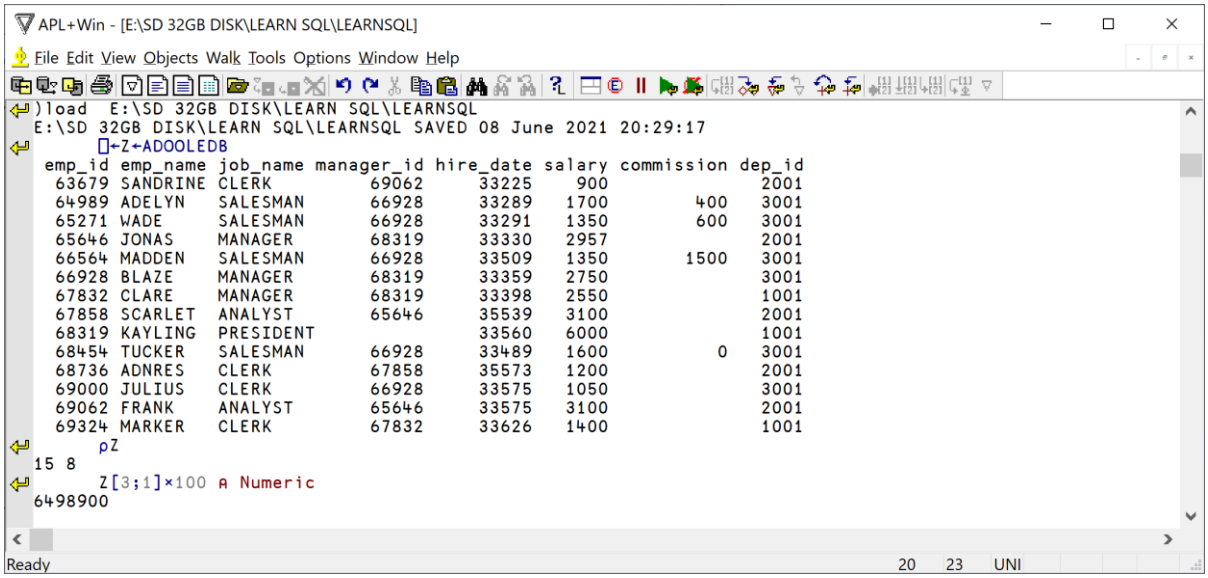
```
    ∇ Z←ADOODBC;ConnectionString;sql;⎕wself
[1]   ⍝ Ajay Askoolum
[2]   ConnectionString←'Driver={ODBC Driver 17 for SQL
      Server};Server=(localdb)\MSSQLLocaldb;Database=W3Resource;Tr
      usted_Connection=yes;'
[3]   sql←'select * from [Employee].[employees];'
[4]   ⎕wself←'ado' ⎕wi 'Create' 'ADODB.Recordset'
[5]   ⎕wi 'XOpen' sql ConnectionString
[6]   Z←((⊂⎕wself) ⎕wi ¨(⊂⊂'Fields.Item().Name'),¨⊂¨(⍳
      ⎕io)+⍳⎕wi 'xFields.Count'),[⎕io]⎕wi 'XGetRows'
[7]   ⎕wself ⎕wi 'Delete'
    ∇
```



Using OLEDB

The following function retrieves the same information using an OLEDB connection:

```
    ∇ Z←ADOOLEDB;ConnectionString;sql;⎕wself
[1]   ⍝ Ajay Askoolum
[2]   ConnectionString←'Provider=MSOLEDBSQL.1;Integrated
Security=SSPI;Persist Security Info=False;User ID="";Initial
Catalog=W3Resource;Data Source=(localdb)\mssqllocaldb;Initial File Na
      me="";Server SPN="";Authentication="";Access Token=""'
[3]   sql←'select * from [Employee].[employees];'
[4]   ⎕wself←'ado' ⎕wi 'Create' 'ADODB.Recordset'
[5]   ⎕wi 'XOpen' sql ConnectionString
[6]   Z←((⊂⎕wself) ⎕wi ¨(⊂⊂'Fields.Item().Name'),¨⊂¨(-⎕io)+⍳⎕wi
'xFields.Count'),[⎕io]⎕wi 'XGetRows'
[7]   ⎕wself ⎕wi 'Delete'
    ∇
```



Using C# Script Engine

Alternative methods include the C# Script Engine if you are adept at C# or the ActiveX server mentioned here: http://forum.apl2000.com/viewtopic.php?f=21&t=849

The C# code in a literal array of rank 2 and names CSCode is[2]:

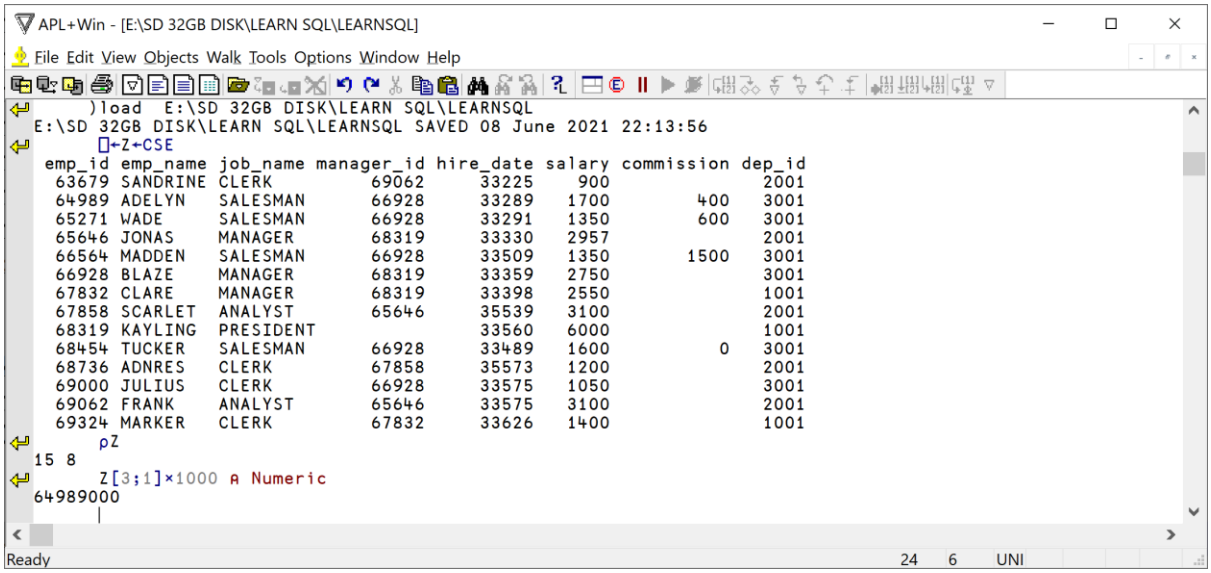

```
 1 static object LocalDB(string DBName, string SQL)
 2 {
 3     ArrayList myArray = new ArrayList();
 4     string connectionString = String.Format(@"Data
Source=np:\\.\pipe\LOCALDB#86B005C8\tsql\query;Initial
Catalog={0};Integrated Security=true;", DBName);
 5     using (SqlConnection connection = new
SqlConnection(connectionString))
 6     {
 7         using (SqlCommand command = new SqlCommand(SQL, connection))
 8         {
 9             try
10             {
11                 connection.Open();
12                 SqlDataReader reader = command.ExecuteReader();
13                 string[] columns = new string[reader.FieldCount];
14                 for (int i = 0; i< reader.FieldCount; i++)
15                     {
16                       columns[i]= reader.GetName(i);
17                     }
18                  myArray.Add(columns);
19                 while (reader.Read())
20                 {
21                     var values = new Object[reader.FieldCount];
22                     reader.GetValues(values);
23                     myArray.Add(values);
24                 }
25                 reader.Close();
26             }
27             catch (Exception ex)
28             {
29                 string error = ex.Message;
30             }
31         }
32         return (object) myArray.ToArray();
33     }
34 }
```

The C# Script Engine code is as follows:

```
  ∇ Z←CSE
[1]  ⍝ Ajay Askoolum
[2]  □cself←'c' □cse 'Init' 'System' 'System.Data' 'System.Collections'
[3]  ←□cse 'ExecStmt' 'using System;using System.Collections;using System.Data;using System.Data.SqlClient;'
[4]  ←□cse 'Exec' CSCode
[5]  Z←⊃□cse 'GetValue' 'LocalDB("{0}","{1}")' "W3Resource"  "SELECT * FROM [EMPLOYEE].[EMPLOYEES];"
  ∇
```

[2] Examine the code in the workspace; here I've added line numbering as an aid to readability—highlight continuation lines—but C# code does not have line numbers.

See the screenshot below for the results:



Warning

I can guarantee that the function CSE will not work for you. With equal conviction, I can state that I have not contrived to produce the screenshot by subterfuge.

This elusive problem was raised in http://forum.apl2000.com/viewtopic.php?f=21&t=849#p3329. At the time, the conclusion was that there were insurmountable problems—see

http://forum.apl2000.com/viewtopic.php?f=21&t=849#p3331**.**

So, what is the explanation?

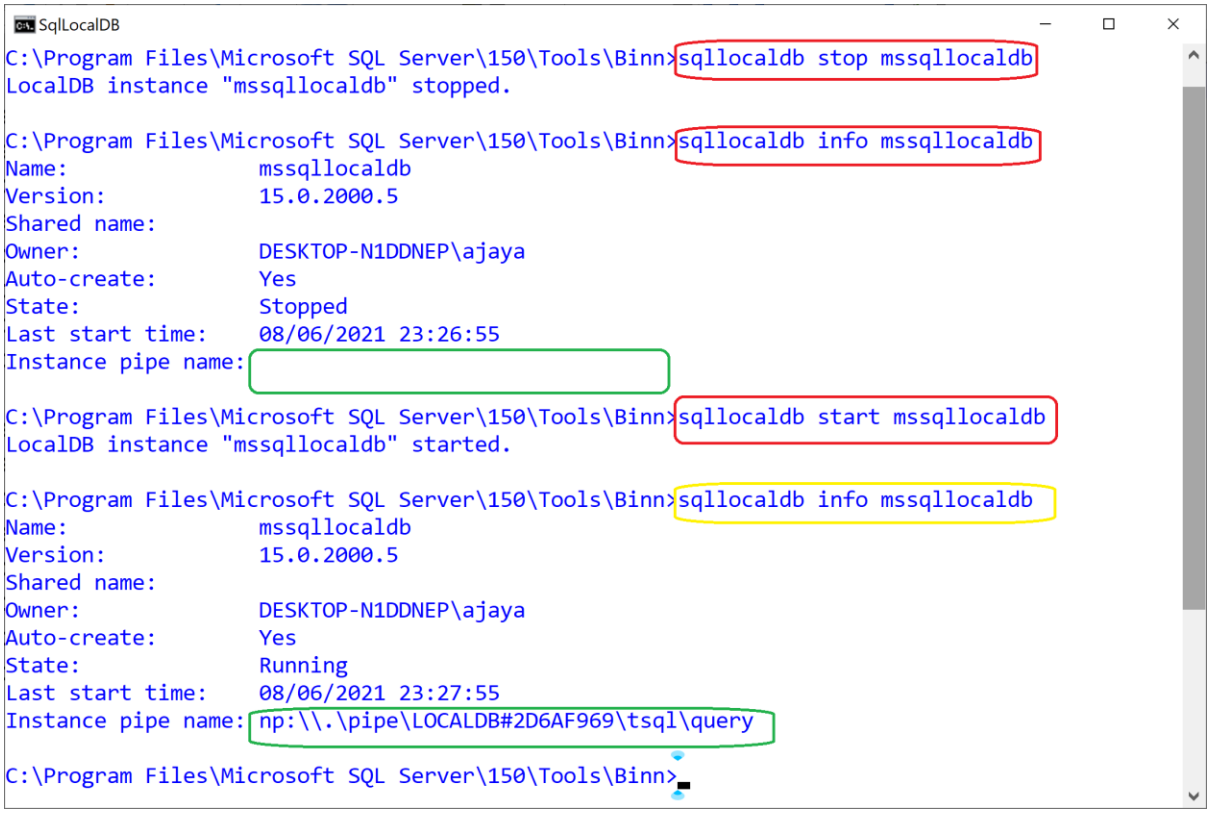
LocalDB acts like an embedded database; it must be on the same computer as the application and its state can be on or off. There is no server. The library System.Data.SqlClient requires a connection string that starts with the specification of a *Data Source* or a server name.

In the screenshot below, I am using SQLLocalDB.EXE to query and alter the state of LocalDB; I have enclosed the commands issued in red rectangles. Note the information enclosed in green rectangles, that is, the *Instance pipe name*.

In the C# code above—held in variable CSCode in the workspace—I've highlighted the *Instance pipe name; y*ou will note that the *Instance pipe name* shown in the C# code above is different from the *Instance pipe name* in the screenshot below. That is the heart of the problem.: *Instance pipe name* is empty when LocalDB has stopped and volatile when it is running.

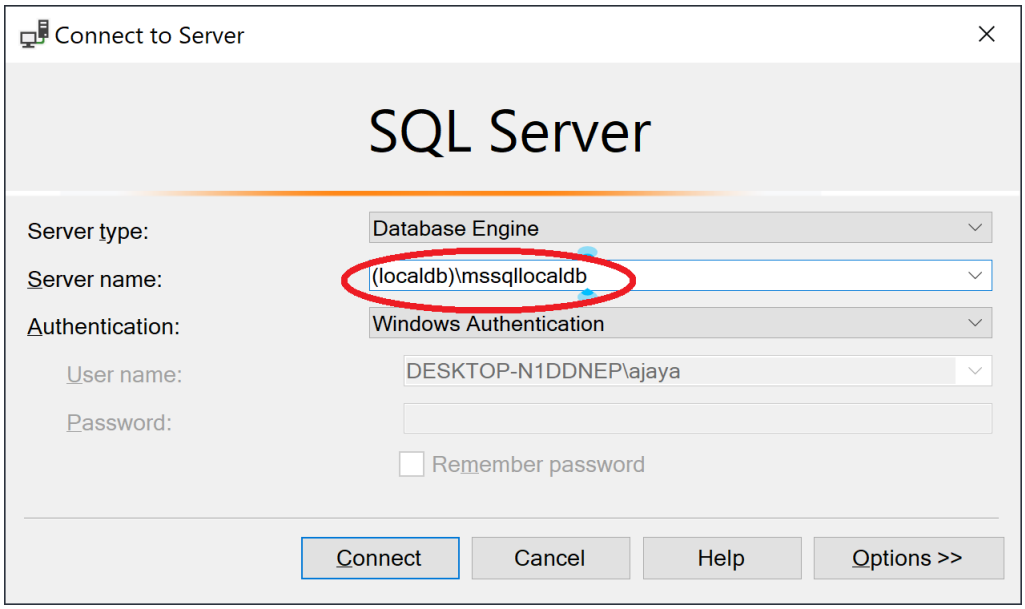
Therefore, it is impossible to embed the *Data Source* specification in code as it is volatile. That is why the CSE function will not run for you; a workaround is for you to repeat the steps in the screenshot below and to replace the *Instance pipe name* in the C# code by the one that you see in your command prompt session.

It is quite messy to automate the process for programmatically retrieving the *Instance pipe name* in real time with APL+Win; therefore, I'll omit further discussion of the issues especially since I've provided a workaround.



LocalDB is ambivalent

In some instances, you can use (localdb)\MSSQLLocalDB as the 'server' name, for example in a Visual Studio data project or with SSMS in code you need to specify the *Instance pipe name*.



Finally

As I mentioned, SQL is a valuable skill that you can acquire at your own pace. It will enable you to separate the data from your application into an independent medium, namely, SQL server or another.

Google searches yield relevant results for almost all SQL related queries. However, the APL+Win forum is a good place to start for raising issues if you want 100% relevant responses.

In your quest to learn SQL, you will want to be acquainted with SQL conventions both for syntax— https://docs.microsoft.com/en-us/sql/t-sql/language-elements/transact-sql-syntax-conventions-transact-sql?view=sql-server-ver15 --and scripts— https://www.c-sharpcorner.com/UploadFile/f0b2ed/what-is-naming-convention/ -- sooner rather than later.

Remember that the ability to share and acquire APL application data *independently* of APL empowers APL.

Ajay Askoolum
June 2021