Data Admin Concepts & Database Management

Lab 10 – Using Your Database

Table of Contents

Overview

This lab is the tenth of ten labs in which we will build a database using the systematic approach covered in the asynchronous material. Each successive lab will build upon the one before and can be a useful guide for building your own database projects.

This lab has two parts. In part one, we will build a basic graphical user interface (GUI) for our database using Microsoft Access. In part two, we will use RStudio and R to read data from the database and prepare some basic charts.

Read this lab document once through before beginning.

Learning Objectives

In this lab you will

- Demonstrate proficiency in using software to connect to a database server and manipulate the database contents.

Lab Goals

In this lab, we will use Microsoft Access, RStudio, and Open Database Connectivity (ODBC) to connect to our VidCast database.

TIP: *Appendix B of this lab lists a few resources for connecting to a database using other tools such as Python and Java.*

What You Will Need to Begin

- This document
- An active Internet connection (if using iSchool Remote lab)
- A blank Word (or similar) document into which you can place your answers. Please include your name, the current date, and the lab number on this document. Please also number your responses, indicating which part and question of the lab to which the answer pertains. Word docx format is preferred. If using another word processing application, please convert the document to pdf before submitting your work to ensure your instructor can open the file.
- To have completed Lab 08 – Database Programming
- Understanding of database tables and have reviewed the asynchronous material through Week 9
- One of the following means of accessing a SQL Server installation
    - A connection to the iSchool Remote Lab (https://remotelab.ischool.syr.edu)
    - A local installation of SQL Server (see Developer edition here https://www.microsoft.com/en-us/sql-server/sql-server-downloads-free-trial)
    - Regardless of how you access SQL Server, you will need to use SQL Server Management Studio to do so.
- An installation of Microsoft Access. This lab uses the latest version of Access 365, but the tools presented are available in most prior versions. If you don't already have a copy of Access, you can obtain a license in the following ways:
    - Install Office 365 using your license available from your SUmail portal account. For more information, visit this link:

  https://answers.syr.edu/display/software/Office+365+ProPlus+for+Staff+and+Faculty+through+Microsoft+Student+Advantage
  - Download and install the latest version of Access from the Microsoft Imagine service accessible at https://my.ischool.syr.edu/
  - If you would prefer to not install Access, or are not running a compatible operating system, you can use the Access installation provided on the iSchool Remote Lab. For help connecting to the Remote Lab, visit this page (netid login required): https://answers.syr.edu/pages/viewpage.action?pageId=9666607
- An installation of RStudio and R
  - To install RStudio and R on your own computer, first download and install R from https://cran.rstudio.com/
  - To use RStudio, obtain a copy at https://www.rstudio.com/products/rstudio/#Desktop. Be sure to download the open source edition. You do not need a commercial license to complete this lab.
  - If you prefer, you can use the RStudio installation provided on the iSchool remote lab.
  - **NOTE**: If you have been using your own computer for the labs thus far, it is advisable to download and install a copy of R and RStudio on your own computer, otherwise you will have to copy your database from your computer to the Remote Lab server using the Generate Scripts task of your database.

Part 1 - Creating a Graphical User Interface (GUI)

Setup

So far, we have been using SQL Server Management Studio to interact with our database objects and data. It is unreasonable for us to expect that our end users will exclusively use SSMS. In this part, we will prototype an interface that allows us to manage vc_User and vc_UserLogin records.

Formatting Note

Look for the "To Do" icon to point out sections of the lab you will need to do to complete the tasks.

Open Database Connectivity (ODBC)

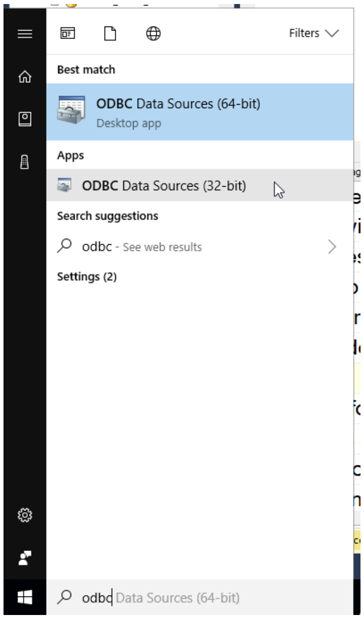
Throughout this lab, we will use ODBC to connect to your database. In the general case, RDBMS vendors provide the means to directly connect to their systems from a variety of tools, but for our purposes, ODBC will suffice.

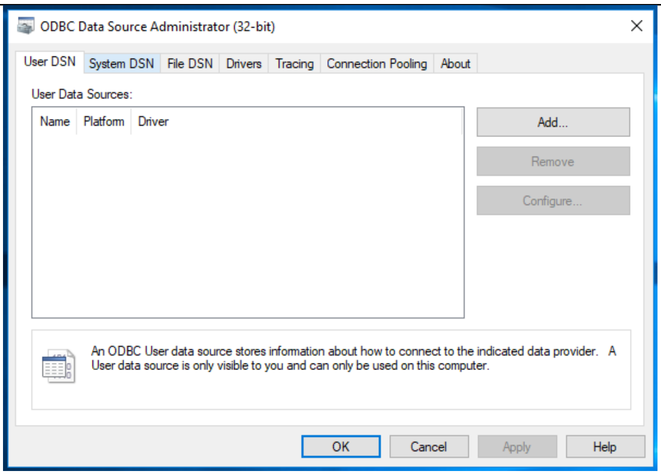
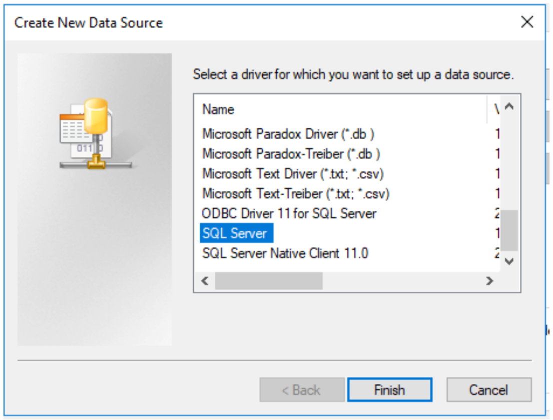
32-bit or 64-bit?

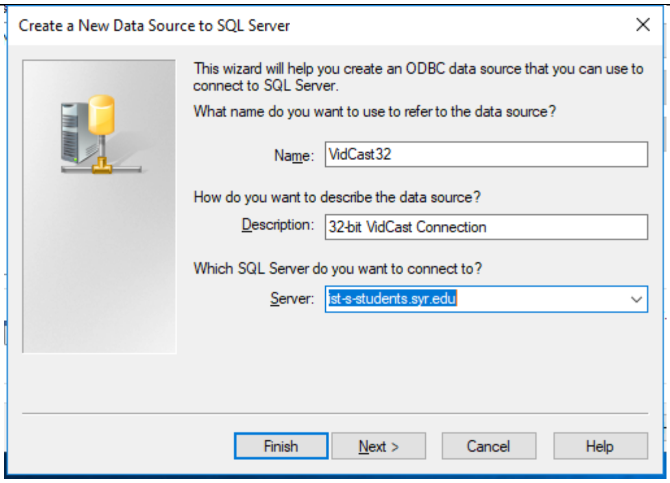
Windows 10 ships with two separate ODBC Administrators: one for 32-bit connections and one for 64-bit connections. The choice to use 32 or 64-bit drivers depends on the tool you'll be using to connect. For part 1, we assume 32-bit Access is installed and will create a 32-bit ODBC data source name (DSN) accordingly. Part 2 assumes a 64-bit installation of R and, therefore, uses a 64-bit DSN. Both methods are shown in this lab as well as the method for determining which architecture to employ.

Creating a Data Source Name (DSN) – 32-bit

Use the following steps to create a 32-bit DSN. These instructions are valid for either your own computer or the Remote Lab.

Step #	Action
1	**In Windows, click the Window button and begin typing "ODBC" without the quotes. You should see the ODBC Administrator in the Start Menu.** **Select the 32-bit option by clicking on it.**  Filters Best match ODBC Data Sources (64-bit) Desktop app Apps ODBC Data Sources (32-bit) Search suggestions odbc - See web results Settings (2) odbc Data Sources (64-bit)
2	**In the ODBC Data Source Administrator applet, ensure the User DSN tab is active. This makes the DSN only available to the user currently logged-in. By contrast, the System DSN option would create a DSN that any user on the computer could use. Remote Lab users do not have permission to create System DSNs, so this lab will exclusively work with User DSNs.**

	 ODBC Data Source Administrator (32-bit) User DSN System DSN File DSN Drivers Tracing Connection Pooling About User Data Sources: Name Platform Driver Add... Remove Configure... OK Cancel Apply Help An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you and can only be used on this computer. **Click Add.**
3	**In the Create New Data Source dialog, scroll the driver list to the bottom and pick SQL Server.**  Create New Data Source Name < Back Finish Cancel Select a driver for which you want to set up a data source. Microsoft Paradox Driver (*.db) Microsoft Paradox-Treiber (*.db) Microsoft Text Driver (*.txt; *.csv) Microsoft Text-Treiber (*.txt; *.csv) ODBC Driver 11 for SQL Server SQL Server SQL Server Native Client 11.0 **Click Finish.**
4	**Use the following values to make your connection:** ● **Name: VidCast32** ● **Description: 32-bit VidCast connection** ● **Server:** ○ **If Remote Lab: ist-s-students.syr.edu** ○ **If Your own computer: . (that's just a dot – can also use localhost)**



Create a New Data Source to SQL Server

This wizard will help you create an ODBC data source that you can use to connect to SQL Server.

What name do you want to use to refer to the data source?

Name: VidCast32

How do you want to describe the data source?

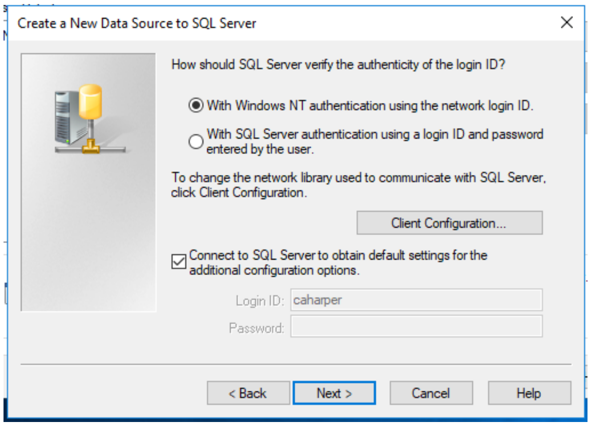
Description: 32-bit VidCast Connection

Which SQL Server do you want to connect to?

Server: ist-s-students.syr.edu

Click Next.

5 **On the next screen, ensure "With Windows NT…" is selected and click Next.**



Create a New Data Source to SQL Server

How should SQL Server verify the authenticity of the login ID?

● With Windows NT authentication using the network login ID.

○ With SQL Server authentication using a login ID and password entered by the user.

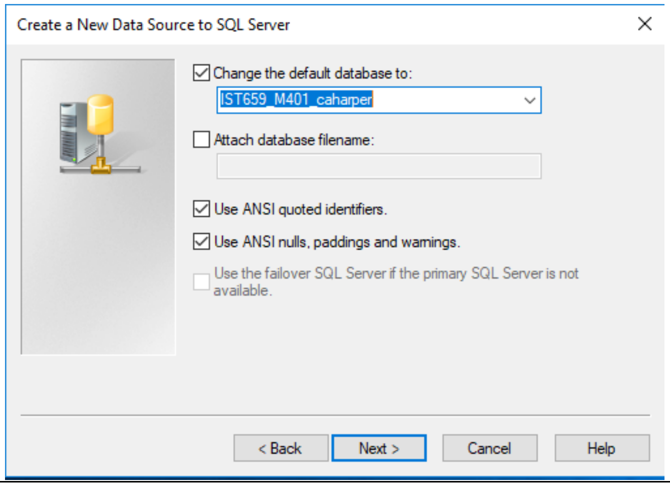
To change the network library used to communicate with SQL Server, click Client Configuration.

☑ Connect to SQL Server to obtain default settings for the additional configuration options.

Login ID: caharper

Password:

6 **On the next screen, check "Change default database to:" and select the correct database from the drop down.**



Create a New Data Source to SQL Server

☑ Change the default database to:

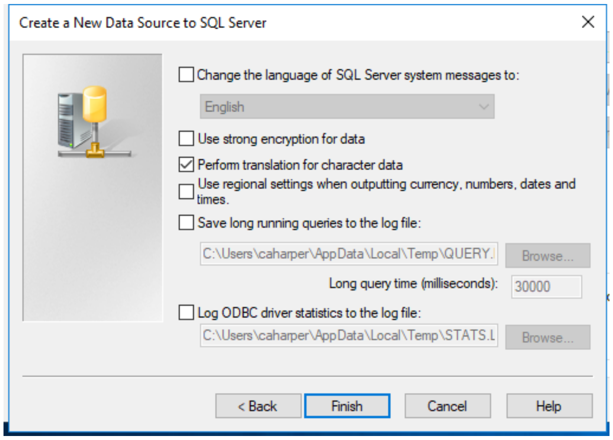
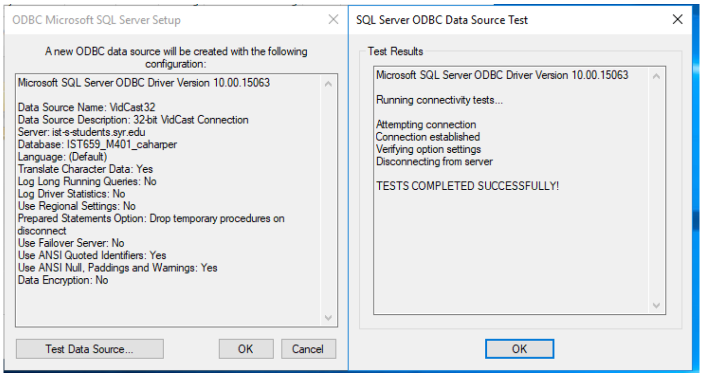
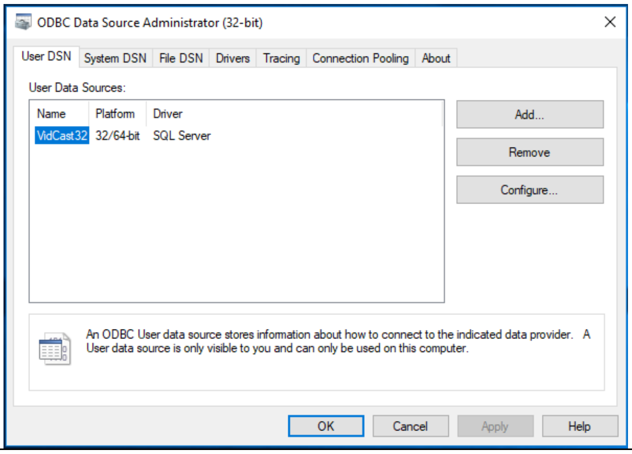
IST659_M401_caharper

☐ Attach database filename:

☑ Use ANSI quoted identifiers.

☑ Use ANSI nulls, paddings and warnings.

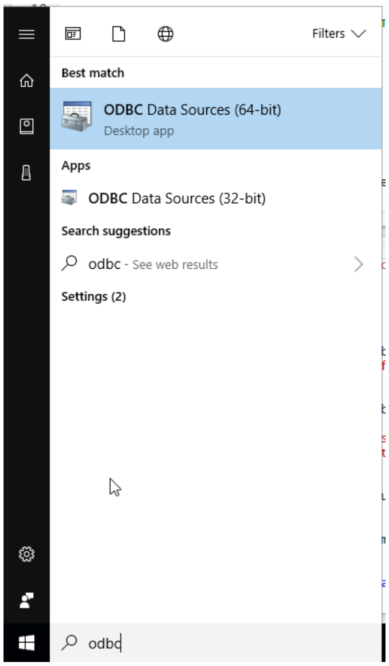
☐ Use the failover SQL Server if the primary SQL Server is not available.

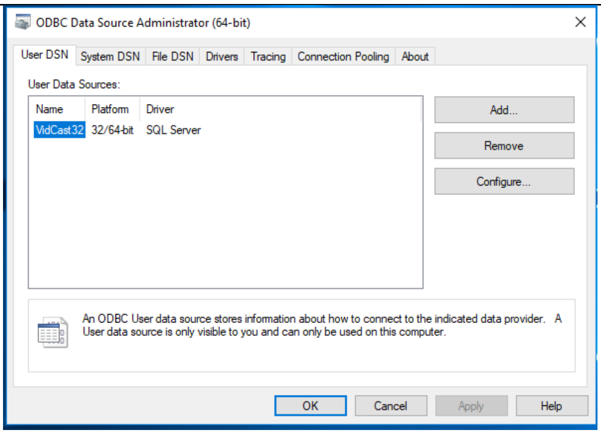
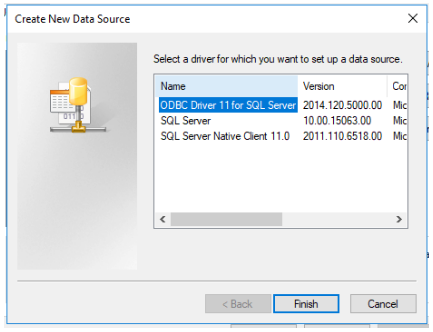
	Click Next.
7	**On the next screen, accept all the defaults and click Finish.**  Create a New Data Source to SQL Server ☐ Change the language of SQL Server system messages to: English ☐ Use strong encryption for data ☑ Perform translation for character data ☐ Use regional settings when outputting currency, numbers, dates and times. ☐ Save long running queries to the log file: C:\Users\caharper\AppData\Local\Temp\QUERY.L Long query time (milliseconds): 30000 ☐ Log ODBC driver statistics to the log file: C:\Users\caharper\AppData\Local\Temp\STATS.L
8	**On the next screen, click Test Connection. If you receive the message "TESTS COMPLETED SUCCESSFULLY", click OK in both dialogs.** **If you get an error, you'll have to resolve the error before proceeding.**  ODBC Microsoft SQL Server Setup A new ODBC data source will be created with the following configuration: Microsoft SQL Server ODBC Driver Version 10.00.15063 Data Source Name: VidCast32 Data Source Description: 32-bit VidCast Connection Server: ist-s-students.syr.edu Database: IST659_M401_caharper Language: (Default) Translate Character Data: Yes Log Long Running Queries: No Log Driver Statistics: No Use Regional Settings: No Prepared Statements Option: Drop temporary procedures on disconnect Use Failover Server: No Use ANSI Quoted Identifiers: Yes Use ANSI Null, Paddings and Warnings: Yes Data Encryption: No SQL Server ODBC Data Source Test Test Results Microsoft SQL Server ODBC Driver Version 10.00.15063 Running connectivity tests... Attempting connection Connection established Verifying option settings Disconnecting from server TESTS COMPLETED SUCCESSFULLY!
10	**Verify your DSN shows in the list and then click OK.**  ODBC Data Source Administrator (32-bit) User DSN System DSN File DSN Drivers Tracing Connection Pooling About User Data Sources: Name — Platform — Driver VidCast32 — 32/64-bit — SQL Server An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you and can only be used on this computer. Add... Remove Configure... OK Cancel Apply Help

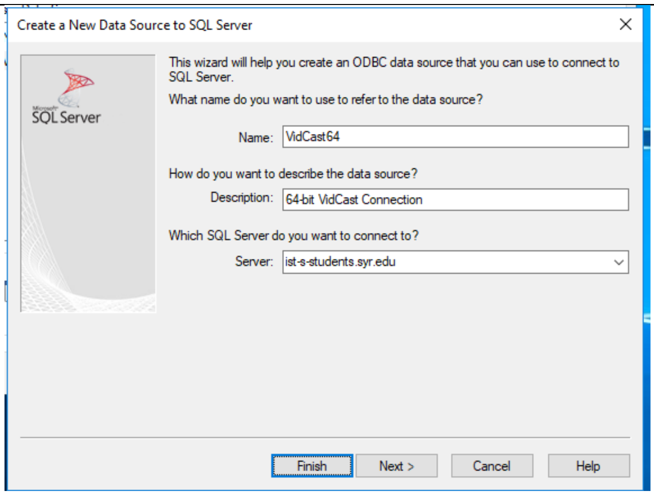
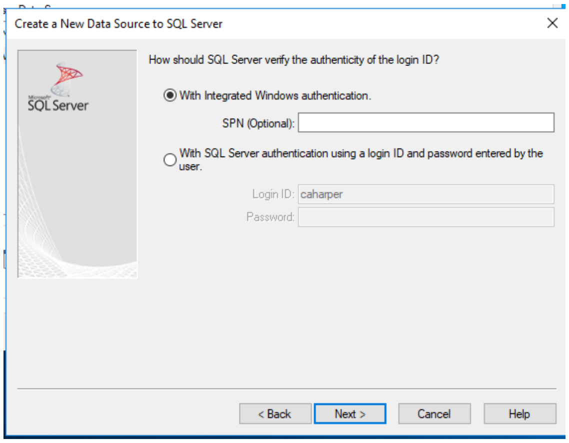
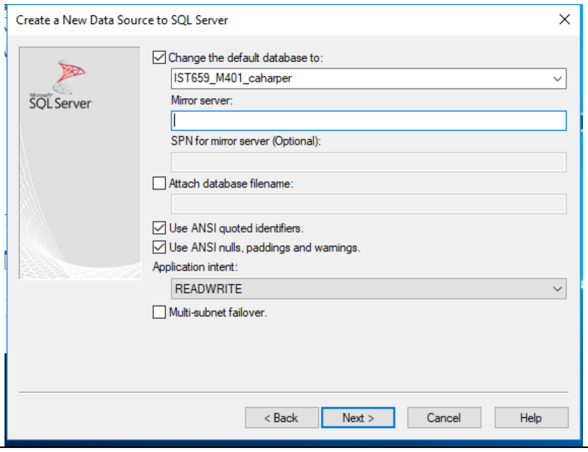
Creating a Data Source Name (DSN) – 64-bit

Creating a 64-bit DSN is the same process as creating a 32-bit with some subtle differences. The steps below may look like the 32-bit set up, but there are some differences.

Use the following steps to create a 64-bit DSN. These instructions are valid for either your own computer or the Remote Lab.

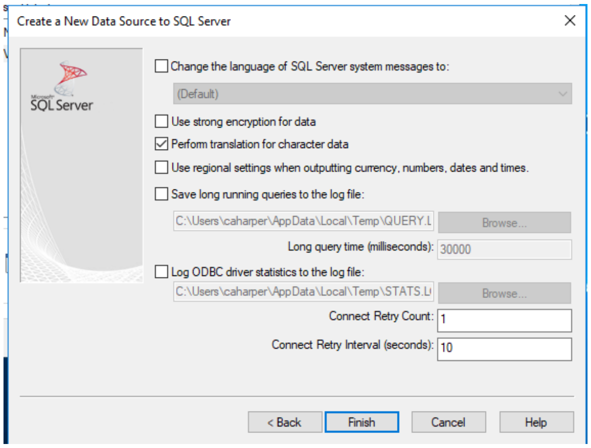
Step #	Action
1	**In Windows, click the Window button and begin typing "ODBC" without the quotes. You should see the ODBC Administrator in the Start Menu.** **Select the 64-bit option by clicking on it.**  Filters Best match ODBC Data Sources (64-bit) Desktop app Apps ODBC Data Sources (32-bit) Search suggestions odbc - See web results Settings (2) odbc
2	**In the ODBC Data Source Administrator applet, ensure the User DSN tab is active. This makes the DSN only available to the user currently logged-in. By contrast, the System DSN option would create a DSN that any user on the computer could use. Remote Lab users do not have permission to create System DSNs, so this lab will exclusively work with User DSNs.** **(Your 32-bit DSN should show in the list. That's okay.)**

	 ODBC Data Source Administrator (64-bit) User DSN System DSN File DSN Drivers Tracing Connection Pooling About Add... Remove Configure... OK Cancel Apply Help User Data Sources: Name Platform Driver VidCast32 32/64-bit SQL Server An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you and can only be used on this computer. **Click Add.**
3	**In the Create New Data Source dialog, pick ODBC Driver 11 for SQL Server.**  Select a driver for which you want to set up a data source. Name Version Cor ODBC Driver 11 for SQL Server 2014.120.5000.00 Mic SQL Server 10.00.15063.00 Mic SQL Server Native Client 11.0 2011.110.6518.00 Mic **Click Finish.**
4	**Use the following values to make your connection:** • **Name: VidCast64** • **Description: 64-bit VidCast connection** • **Server:** ○ **If Remote Lab: ist-s-students.syr.edu** ○ **If Your own computer: . (that's just a dot – can also use localhost)**

	 Create a New Data Source to SQL Server This wizard will help you create an ODBC data source that you can use to connect to SQL Server. What name do you want to use to refer to the data source? Name: VidCast64 How do you want to describe the data source? Description: 64-bit VidCast Connection Which SQL Server do you want to connect to? Server: ist-s-students.syr.edu **Click Next.**
5	**On the next screen, ensure "With Integrated Windows authentication" is selected and click Next.**  Create a New Data Source to SQL Server How should SQL Server verify the authenticity of the login ID? ● With Integrated Windows authentication. SPN (Optional): ○ With SQL Server authentication using a login ID and password entered by the user. Login ID: caharper Password:
6	**On the next screen, check "Change default database to:" and select the correct database from the drop down.**  Create a New Data Source to SQL Server ☑ Change the default database to: IST659_M401_caharper Mirror server: SPN for mirror server (Optional): □ Attach database filename: ☑ Use ANSI quoted identifiers. ☑ Use ANSI nulls, paddings and warnings. Application intent: READWRITE □ Multi-subnet failover.

Click Next.

7 **On the next screen, accept all the defaults and click Finish.**

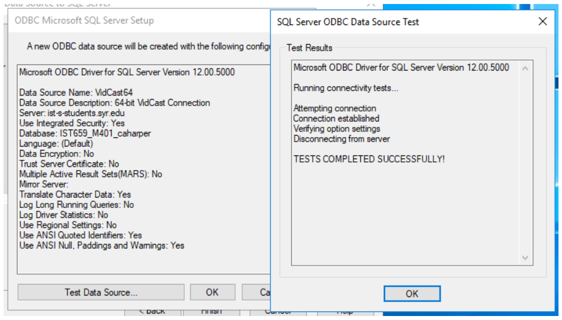


Create a New Data Source to SQL Server

☐ Change the language of SQL Server system messages to:
(Default)
☐ Use strong encryption for data
☑ Perform translation for character data
☐ Use regional settings when outputting currency, numbers, dates and times.
☐ Save long running queries to the log file:
C:\Users\caharper\AppData\Local\Temp\QUERY.L
Long query time (milliseconds): 30000
☐ Log ODBC driver statistics to the log file:
C:\Users\caharper\AppData\Local\Temp\STATS.L
Connect Retry Count: 1
Connect Retry Interval (seconds): 10

8 **On the next screen, click Test Connection. If you receive the message "TESTS COMPLETED SUCCESSFULLY", click OK in both dialogs.**

If you get an error, you'll have to resolve the error before proceeding.



ODBC Microsoft SQL Server Setup

A new ODBC data source will be created with the following configu

```
Microsoft ODBC Driver for SQL Server Version 12.00.5000

Data Source Name: VidCast64
Data Source Description: 64-bit VidCast Connection
Server: ist-s-students.syr.edu
Use Integrated Security: Yes
Database: IST659_M401_caharper
Language: (Default)
Data Encryption: No
Trust Server Certificate: No
Multiple Active Result Sets(MARS): No
Mirror Server:
Translate Character Data: Yes
Log Long Running Queries: No
Log Driver Statistics: No
Use Regional Settings: No
Use ANSI Quoted Identifiers: Yes
Use ANSI Null, Paddings and Warnings: Yes
```

SQL Server ODBC Data Source Test

Test Results

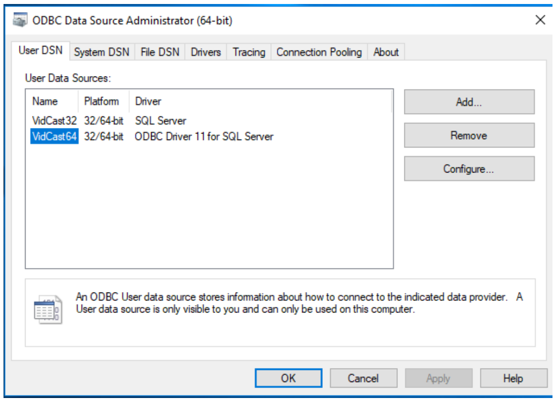
```
Microsoft ODBC Driver for SQL Server Version 12.00.5000

Running connectivity tests...

Attempting connection
Connection established
Verifying option settings
Disconnecting from server

TESTS COMPLETED SUCCESSFULLY!
```

10 **Verify your DSN shows in the list and then click OK.**



ODBC Data Source Administrator (64-bit)

User Data Sources:

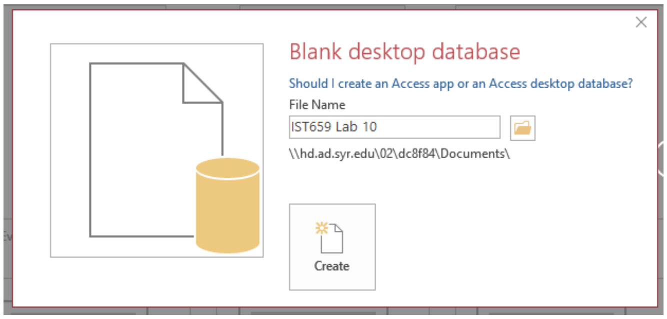
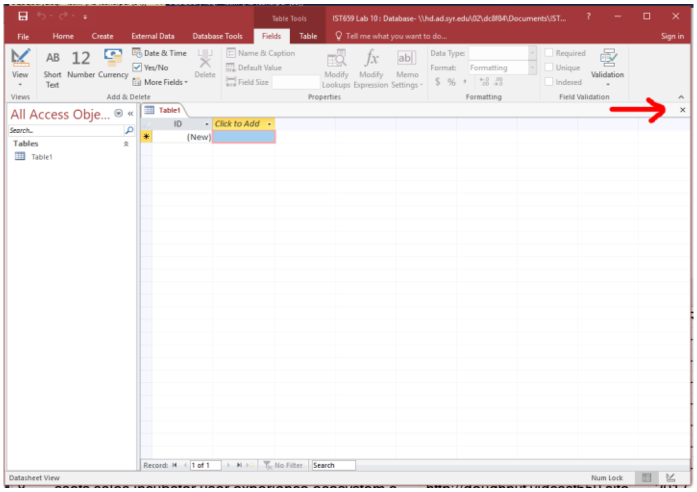
Name	Platform	Driver
VidCast32	32/64-bit	SQL Server
VidCast64	32/64-bit	ODBC Driver 11 for SQL Server

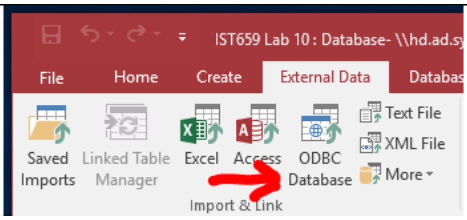
An ODBC User data source stores information about how to connect to the indicated data provider. A User data source is only visible to you and can only be used on this computer.

Using Microsoft Access to Connect to SQL Server

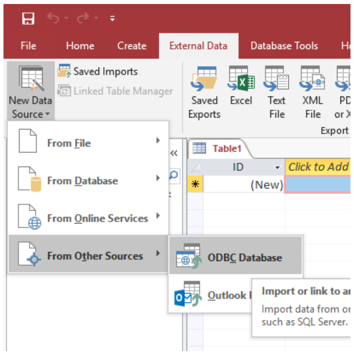
Now that we have created the DSNs, we can tell Access to use one of them to link to our data.

Use the following steps to create an Access database and connect to SQL Server:

Step #	Action
1	**Open Microsoft Access and select Blank Desktop Database from the list of templates provided. Name your database IST659 Lab 10 or something similar.** **Pay attention to where the file is going to save. The default is your Documents folder. Feel free to browse for an alternate location if you prefer.**  Blank desktop database Should I create an Access app or an Access desktop database? File Name: IST659 Lab 10 \\hd.ad.syr.edu\02\dc8f84\Documents\ **Click Create**
2	**We do not need the blank table that opens, so close the tab by clicking the X in the tab strip.**  IST659 Lab 10 : Database- \\hd.ad.syr.edu\02\dc8f84\Documents\IST... File Home Create External Data Database Tools Fields Table Tell me what you want to do... All Access Obje... Tables Table1 ID Click to Add (New) Datasheet View
3	**To begin the process of linking to the SQL Server database, click the External Data Ribbon and click the ODBC Database button.**

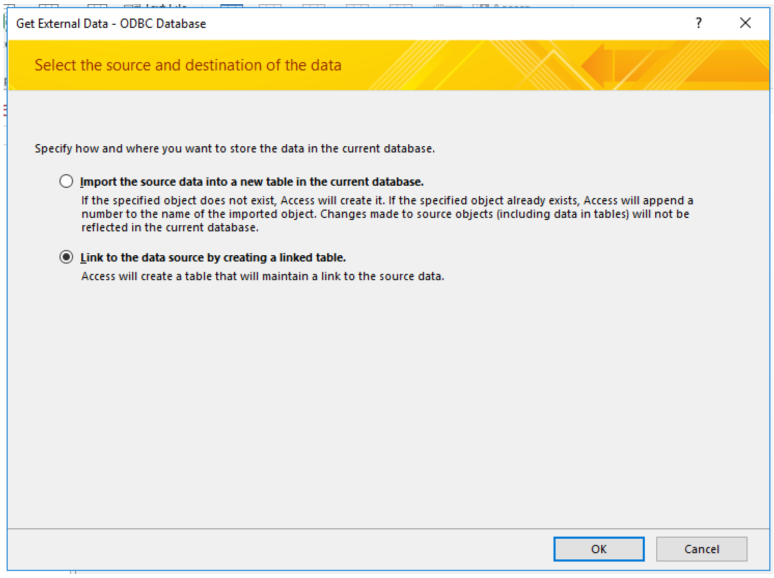


Note: If you are using Access 365, this option looks like the image below:



4 **In the dialog, select “Link to the data source…” to create a dynamic link to the database.**

The other option does a one-time import and will not save changes back to the server.



Get External Data - ODBC Database

Select the source and destination of the data

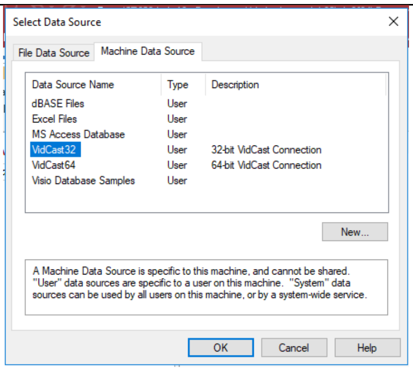
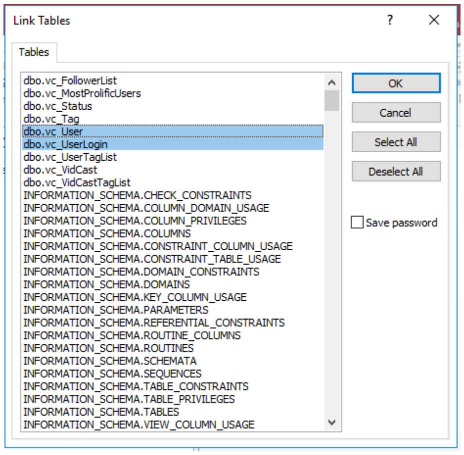
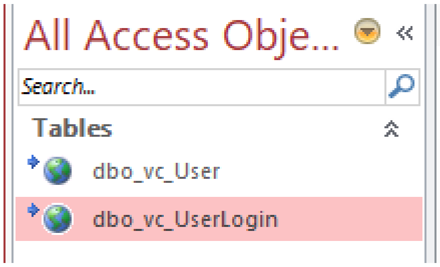
Specify how and where you want to store the data in the current database.

○ **Import the source data into a new table in the current database.**
If the specified object does not exist, Access will create it. If the specified object already exists, Access will append a number to the name of the imported object. Changes made to source objects (including data in tables) will not be reflected in the current database.

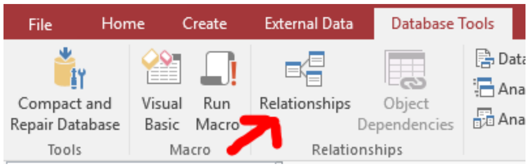
● **Link to the data source by creating a linked table.**
Access will create a table that will maintain a link to the source data.

Click OK

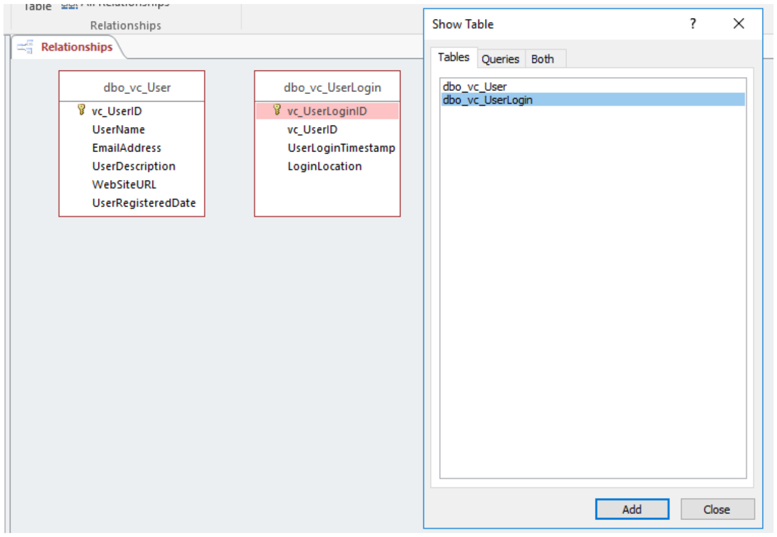
5 **In the Select Data Source dialog, select the Machine Data Source tab and select the VidCast32 entry.**

	 Select Data Source File Data Source Machine Data Source New... OK Cancel Help Data Source Name / Type / Description dBASE Files / User Excel Files / User MS Access Database / User VidCast32 / User / 32-bit VidCast Connection VidCast64 / User / 64-bit VidCast Connection Visio Database Samples / User A Machine Data Source is specific to this machine, and cannot be shared. "User" data sources are specific to a user on this machine. "System" data sources can be used by all users on this machine, or by a system-wide service. **Click OK.**
6	**In the Link Tables dialog, click once on dbo.vc_User and once on dbo.vc_UserLogin. (Take note that all our tables are here! We're only going to use two tables for now).**  Link Tables ? Tables OK Cancel Select All Deselect All ☐ Save password dbo.vc_FollowerList dbo.vc_MostProlificUsers dbo.vc_Status dbo.vc_Tag dbo.vc_User dbo.vc_UserLogin dbo.vc_UserTagList dbo.vc_VidCast dbo.vc_VidCastTagList INFORMATION_SCHEMA.CHECK_CONSTRAINTS INFORMATION_SCHEMA.COLUMN_DOMAIN_USAGE INFORMATION_SCHEMA.COLUMN_PRIVILEGES INFORMATION_SCHEMA.COLUMNS INFORMATION_SCHEMA.CONSTRAINT_COLUMN_USAGE INFORMATION_SCHEMA.CONSTRAINT_TABLE_USAGE INFORMATION_SCHEMA.DOMAIN_CONSTRAINTS INFORMATION_SCHEMA.DOMAINS INFORMATION_SCHEMA.KEY_COLUMN_USAGE INFORMATION_SCHEMA.PARAMETERS INFORMATION_SCHEMA.REFERENTIAL_CONSTRAINTS INFORMATION_SCHEMA.ROUTINE_COLUMNS INFORMATION_SCHEMA.ROUTINES INFORMATION_SCHEMA.SCHEMATA INFORMATION_SCHEMA.SEQUENCES INFORMATION_SCHEMA.TABLE_CONSTRAINTS INFORMATION_SCHEMA.TABLE_PRIVILEGES INFORMATION_SCHEMA.TABLES INFORMATION_SCHEMA.VIEW_COLUMN_USAGE **Click OK.**
7	**The tables are now linked and show in the All Access Objects pane with globe icons. This icon tells us this is an external data source.**  All Access Obje... Search... Tables dbo_vc_User dbo_vc_UserLogin
8	**Although we took care to create our foreign key constraints, we need to tell Access about them to make a few things easier with the Form Wizard.**

To create the relationship in Access, click the Database Tools ribbon and then click the Relationships button.



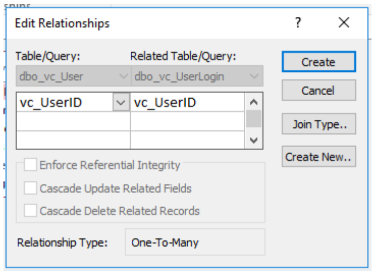
9 **With dbo.vc_User selected in the Show Table dialog, click Add once. Then click once on dbo.vc_UserLogin. Then click Add one more time.**



Click Close to close the Show Table dialog.

10 **When creating a relationship in Access, you drag the column from the one side of the relationship (in this case dbo_vc_User.vc_UserID) and drop on the column on the many side of the relationship (dbo_vc_UserLogin.vc_UserID)**

When you do, the Edit Relationships dialog shows. In the Relationship Type, it should say One-To-Many. If the wrong column was dragged and dropped, or if the correct column was dropped on the wrong foreign key column, it says “Indeterminate”. If you see Indeterminate, click Cancel and try again.


Edit Relationships

Table/Query: dbo_vc_User
Related Table/Query: dbo_vc_UserLogin

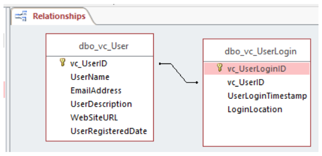
vc_UserID → vc_UserID

□ Enforce Referential Integrity
□ Cascade Update Related Fields
□ Cascade Delete Related Records

Relationship Type: One-To-Many

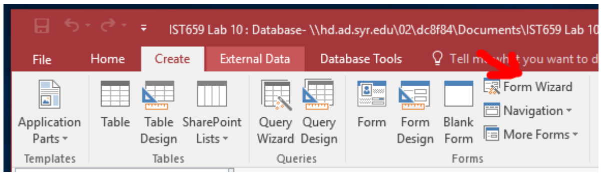
Click Create.

You should now see a line connecting the two tables.

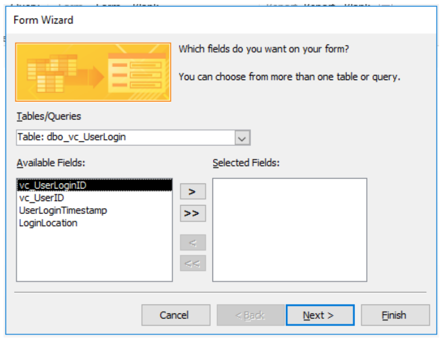


11 **Close the Relationships tab by clicking the X in the tab strip. When prompted to save your layout, click Yes.**

12 **Create a new form by clicking the Create ribbon and then the Form Wizard button.**



The form wizard appears.


Form Wizard

Which fields do you want on your form?

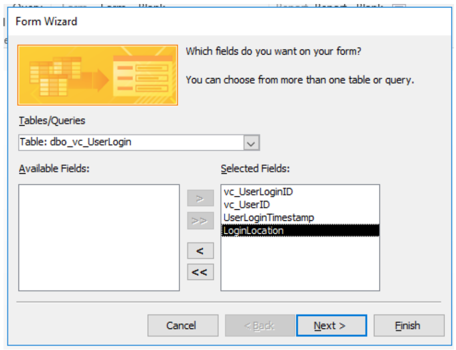
You can choose from more than one table or query.

Tables/Queries: Table: dbo_vc_UserLogin

Available Fields: vc_UserLoginID, vc_UserID, UserLoginTimestamp, LoginLocation

Selected Fields:

13 **With Table: dbo_vc_UserLogin selected in the Tables/Queries drop down, click the Add all button >> to add all the columns from the Available Fields list to the Selected Fields list.**


Form Wizard

Which fields do you want on your form?

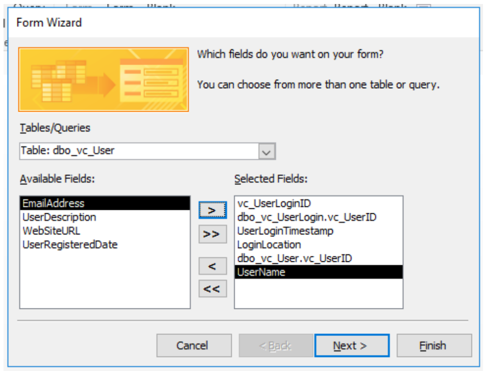
You can choose from more than one table or query.

Tables/Queries: Table: dbo_vc_UserLogin

Available Fields:

Selected Fields: vc_UserLoginID, vc_UserID, UserLoginTimestamp, LoginLocation

14 **Change the Tables/Queries drop down to Table: dbo_vc_User and use the > button to add the vc_UserID and UserName columns from the Available Fields list to the Selected Fields list.**



Form Wizard

Which fields do you want on your form?

You can choose from more than one table or query.

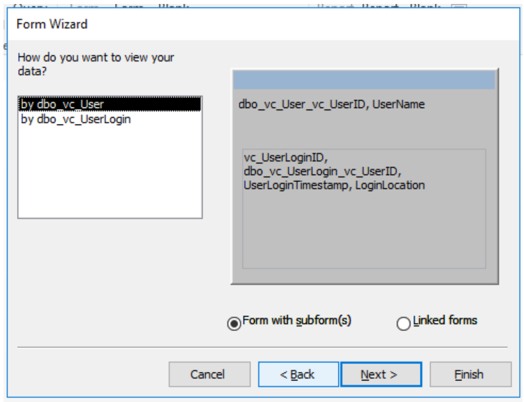
Tables/Queries: Table: dbo_vc_User

Available Fields:
- EmailAddress
- UserDescription
- WebSiteURL
- UserRegisteredDate

Selected Fields:
- vc_UserLoginID
- dbo_vc_UserLogin.vc_UserID
- UserLoginTimestamp
- LoginLocation
- dbo_vc_User.vc_UserID
- UserName

Click Next.

15 **On the next screen, select "by dbo_vc_User" from the list and ensure the "Form with subform(s)" option is selected.**



Form Wizard

How do you want to view your data?

- by dbo_vc_User
- by dbo_vc_UserLogin

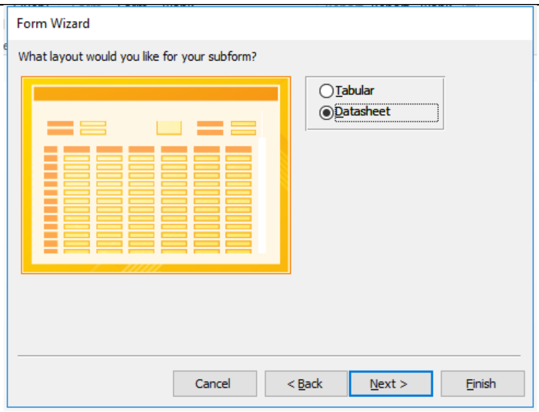
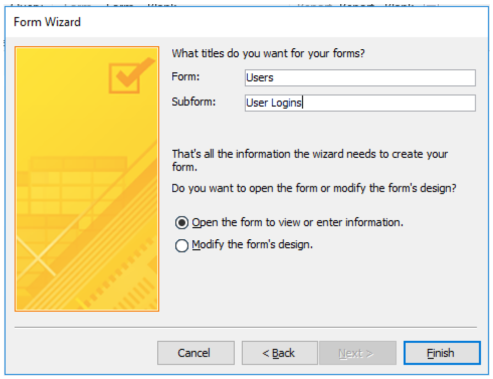
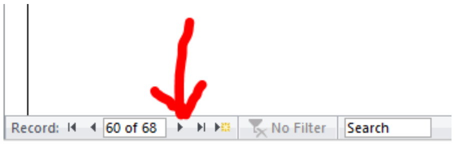
dbo_vc_User_vc_UserID, UserName

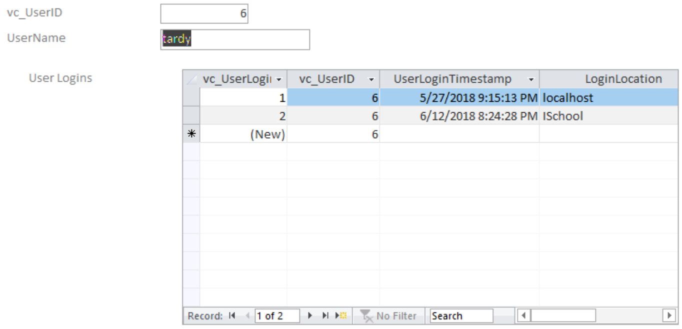
vc_UserLoginID, dbo_vc_UserLogin_vc_UserID, UserLoginTimestamp, LoginLocation

● Form with subform(s) ○ Linked forms

Click Next.

16 **On the next screen, ensure Datasheet is selected. This will show the login info for each user in a spreadsheet-like format. You may opt to use Tabular in your own projects if you feel that is a better look for your application.**

	 Form Wizard What layout would you like for your subform? ○ Tabular ● Datasheet **Click Next.**
17	**On the final screen, name the Form "Users" and the Subform "User Logins".** **Ensure "Open the form…" is selected.**  Form Wizard What titles do you want for your forms? Form: Users Subform: User Logins That's all the information the wizard needs to create your form. Do you want to open the form or modify the form's design? ● Open the form to view or enter information. ○ Modify the form's design. **Click Finish.**
18	**When the form loads, use the navigation buttons at the bottom of the main form to navigate to the user with UserName "tardy". Tardy should be the 60th user in the list (but that may be different depending on your database).**  Record: 60 of 68 **The UserLogin form now shows the logins you created in previous labs!**



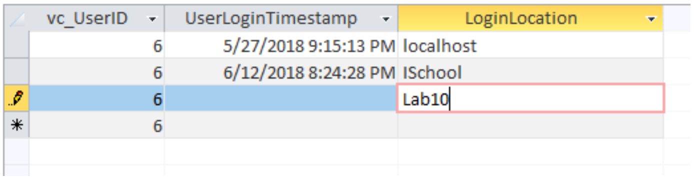
vc_UserID 6

UserName tardy

User Logins

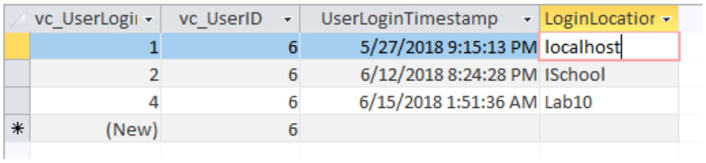
	vc_UserLogi	vc_UserID	UserLoginTimestamp	LoginLocation
	1	6	5/27/2018 9:15:13 PM	localhost
	2	6	6/12/2018 8:24:28 PM	ISchool
✳	(New)	6		

19 **On the line of the subform that has "(New)" in the vc_UserLoginID, type "Lab 10" without the quotes under LoginLocation. Notice that as you type the record selector at the left of the line now shows a pencil. This tells us that this record is "dirty". Dirty is a term that means this record has changed but has not been saved to the database.**



	vc_UserID	UserLoginTimestamp	LoginLocation
	6	5/27/2018 9:15:13 PM	localhost
	6	6/12/2018 8:24:28 PM	ISchool
✎	6		Lab10
✳	6		

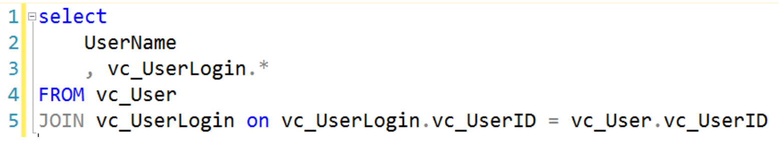
To save a changed record in Access, one merely need select another record. Click the LoginLocation that says localhost.



	vc_UserLogi	vc_UserID	UserLoginTimestamp	LoginLocatior
	1	6	5/27/2018 9:15:13 PM	localhost
	2	6	6/12/2018 8:24:28 PM	ISchool
	4	6	6/15/2018 1:51:36 AM	Lab10
✳	(New)	6		

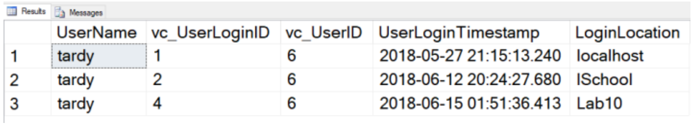
20 **Open SQL Server Management Studio and open a new query window for your database.**

Execute the following code against your database:



```
select
    UserName
    , vc_UserLogin.*
FROM vc_User
JOIN vc_UserLogin on vc_UserLogin.vc_UserID = vc_User.vc_UserID
```

Your results should look like this:

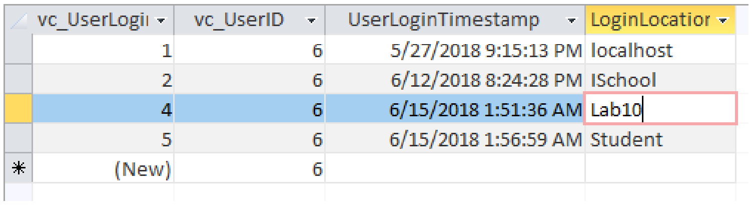


	UserName	vc_UserLoginID	vc_UserID	UserLoginTimestamp	LoginLocation
1	tardy	1	6	2018-05-27 21:15:13.240	localhost
2	tardy	2	6	2018-06-12 20:24:27.680	ISchool
3	tardy	4	6	2018-06-15 01:51:36.413	Lab10

Note: Your list may be different in part. Not to worry. If your lab 10 entry is there, all is well.

Keep SSMS and this query open. We'll be coming back to it.

21 **Return to Access and add another UserLogin record by typing your last name in the LoginLocation cell.**

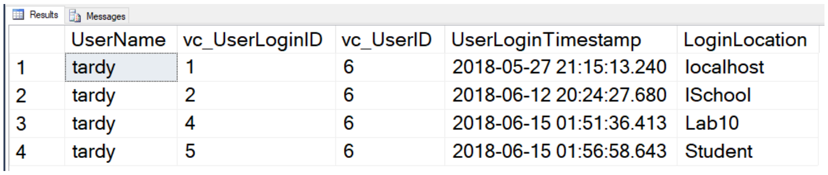


	vc_UserLogi	vc_UserID	UserLoginTimestamp	LoginLocatior
	1	6	5/27/2018 9:15:13 PM	localhost
	2	6	6/12/2018 8:24:28 PM	ISchool
	4	6	6/15/2018 1:51:36 AM	Lab10
	5	6	6/15/2018 1:56:59 AM	Student
✱	(New)	6		

Remember to click another row in the subform to save the record to the server.

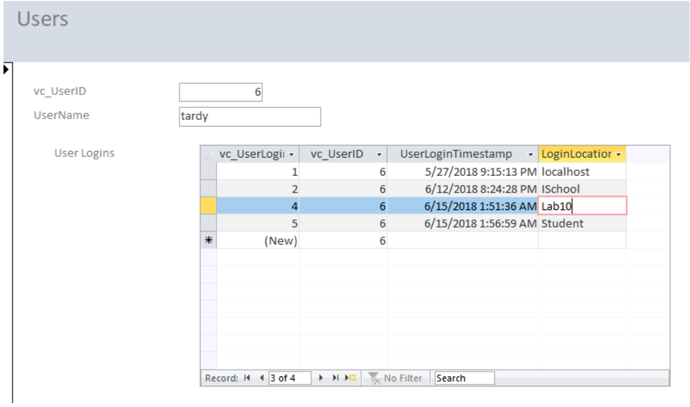
22 **Return to SSMS and execute the SELECT statement from step 20 again.**

Your results should look like this:



	UserName	vc_UserLoginID	vc_UserID	UserLoginTimestamp	LoginLocation
1	tardy	1	6	2018-05-27 21:15:13.240	localhost
2	tardy	2	6	2018-06-12 20:24:27.680	ISchool
3	tardy	4	6	2018-06-15 01:51:36.413	Lab10
4	tardy	5	6	2018-06-15 01:56:58.643	Student

23 **On your answers document, paste a screenshot of your users form with the UserLogin subform. Ensure the record for UserName tardy is on screen to show your new UserLogin records.**



Users

vc_UserID: 6

UserName: tardy

User Logins

	vc_UserLogi	vc_UserID	UserLoginTimestamp	LoginLocatior
	1	6	5/27/2018 9:15:13 PM	localhost
	2	6	6/12/2018 8:24:28 PM	ISchool
	4	6	6/15/2018 1:51:36 AM	Lab10
	5	6	6/15/2018 1:56:59 AM	Student
✱	(New)	6		

24 **Also, on your answers document, answer the following questions:**

1. **If we only provided a value for LoginLocation, how did the software know what to use for vc_UserLoginID and UserLoginTimestamp?**

	2. **Based on what the behavior we saw and the form/subform relationship, how did the software know which vc_UserID to use for the vc_UserLogin record?** 3. **What are some pros and cons (at least 2 of each) to using Access to manage SQL Server data?**
25	**After updating your answers doc, it is okay to close Access and SSMS.**

Part 2 – Using R to Connect to SQL Server

R is a powerful open source tool used by people in numerous fields of study including Data Science. At the iSchool, R is used extensively in Data Science classes.

This lab is intended to familiarize you with the R environment but is by no means a complete R tutorial. Instead, this exercise is intended to demonstrate how general-purpose programming tools connect to and consume data from relational database management systems.

In the general case, an application that needs data from data storage services such as an RDBMS adhere to the following logical flow:

1. Connect to the server and authenticate the user, creating a connection variable of some kind.
2. Prepare the SQL request to send to the server.
3. Send the SQL request prepared in step 2 to the server over the connection established in step 1. The server receives the request, validates its legitimacy, and performs the requested action. The server then sends the results back to the calling program, which stores it in some locally accessible data structure (usually a variable of some kind)
4. The program does whatever work it needs to do on the data.
5. The program outputs its results, either to the screen or to local storage.
6. The program repeats steps 2 through 5 as necessary.
7. The program closes its connection to the database.
8. The program ends.

As with any software, exceptions can happen at any step of the above process and they should be handled, but for now we will step through the successful program flow.

The preceding steps can be applied to R as well as Python, C/C++, PHP, Java, and any number of other languages.

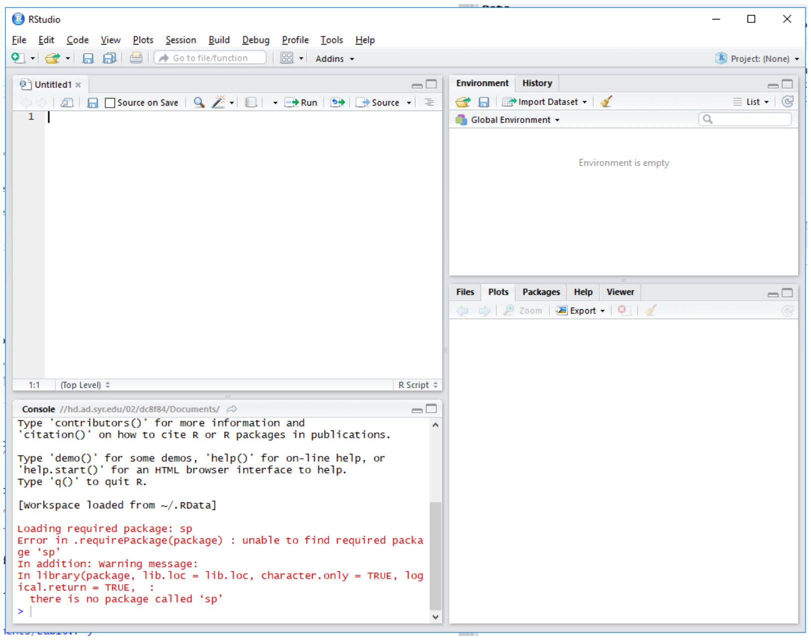
In Part 1, we used Access to read and manipulate data. Access abstracted many of the above steps, simplifying the process for better or worse. We didn't have to right any code to do any of the communication, we only had to flip switches and levers and all of that was handled. In R, we will code (most of) those steps by hand.

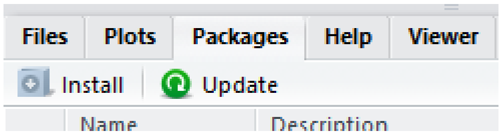
Getting Started with RStudio

Let’s dive right in.

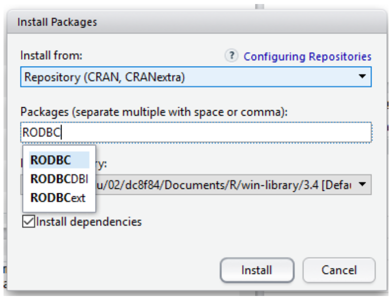
Use the following steps to use RStudio to connect to SQL Server:

Step #	Action
1	**Open RStudio from the start menu.**

2	**Click the File Menu, New File, R Script. A blank R Script file will appear.**
	
	 RStudio
	File Edit Code View Plots Session Build Debug Profile Tools Help
	Go to file/function
	Addins
	Project: (None)
	Untitled1
	Source on Save
	Run
	Source
	1:1
	(Top Level)
	R Script
	Environment History
	Import Dataset
	List
	Global Environment
	Environment is empty
	Files Plots Packages Help Viewer
	Zoom
	Export
	Console //hd.ad.syr.edu/02/dc8f84/Documents/
	`Type 'contributors()' for more information and`
	`'citation()' on how to cite R or R packages in publications.`
	
	`Type 'demo()' for some demos, 'help()' for on-line help, or`
	`'help.start()' for an HTML browser interface to help.`
	`Type 'q()' to quit R.`
	
	`[Workspace loaded from ~/.RData]`
	
	`Loading required package: sp`
	`Error in .requirePackage(package) : unable to find required packa`
	`ge ‘sp’`
	`In addition: Warning message:`
	`In library(package, lib.loc = lib.loc, character.only = TRUE, log`
	`ical.return = TRUE,  :`
	`  there is no package called ‘sp’`
	`>`
	
	RStudio arranges our workspace into quadrants. The upper left is the script file we’ll be working with. Lower left is the console. At this point, it’s worth noting that you can code line by line directly into the console, but that doesn’t net repeatable results, so we’ll use the script approach.
	
	Upper right is a list of variables and objects we have created this session. Lower right is a kitchen sink of tabs we’ll need. Important to us are the Plots and Packages tabs. Take a moment to locate them now.

3	**Because we are using ODBC to connect to SQL Server, we will need to tell R to load up a package called RODBC.** **From the Packages tab, click Install.**  Files Plots Packages Help Viewer Install Update Name Description

The Install Packages dialog appears. Type RODBC in the Packages box:

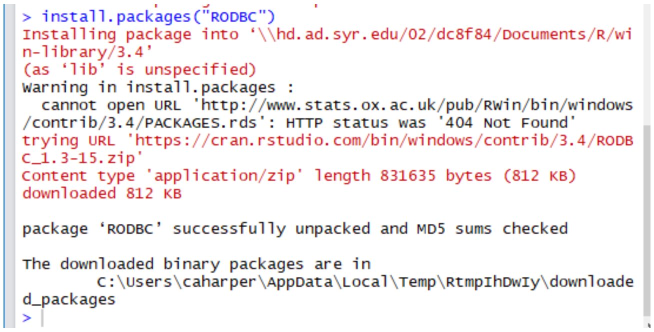


Install from: Repository (CRAN, CRANextra)

Packages (separate multiple with space or comma): RODBC

☑ Install dependencies

Then click Install. R will go through some gyrations to install this package. You will see activity in the console.



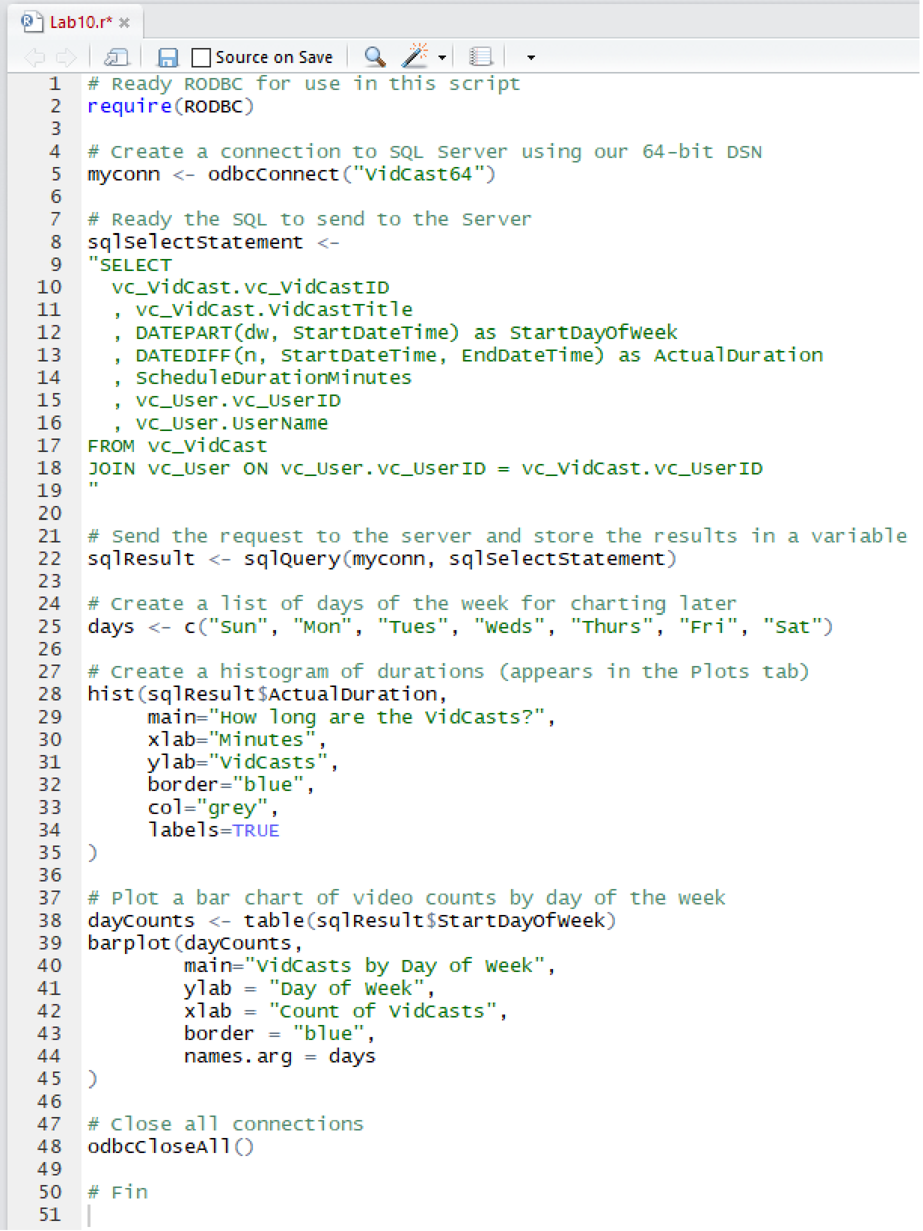
```
> install.packages("RODBC")
Installing package into ‘\\hd.ad.syr.edu/02/dc8f84/Documents/R/wi
n-library/3.4’
(as ‘lib’ is unspecified)
Warning in install.packages :
  cannot open URL 'http://www.stats.ox.ac.uk/pub/RWin/bin/windows
/contrib/3.4/PACKAGES.rds': HTTP status was '404 Not Found'
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.4/RODB
C_1.3-15.zip'
Content type 'application/zip' length 831635 bytes (812 KB)
downloaded 812 KB

package ‘RODBC’ successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\caharper\AppData\Local\Temp\RtmpIhDwIy\downloade
d_packages
>
```

Look for the phrase “package ‘RODBC’ successfully unpacked and MD5 sums checked” to confirm the installation was successful.

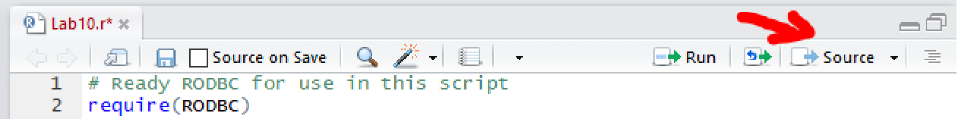
4 **In the script pane (upper left of the four), enter the following code. Save your file often (especially if working in the remote lab!)**



```
 1  # Ready RODBC for use in this script
 2  require(RODBC)
 3
 4  # Create a connection to SQL Server using our 64-bit DSN
 5  myconn <- odbcConnect("VidCast64")
 6
 7  # Ready the SQL to send to the Server
 8  sqlSelectStatement <-
 9  "SELECT
10    vc_VidCast.vc_VidCastID
11    , vc_VidCast.VidCastTitle
12    , DATEPART(dw, StartDateTime) as StartDayOfWeek
13    , DATEDIFF(n, StartDateTime, EndDateTime) as ActualDuration
14    , ScheduleDurationMinutes
15    , vc_User.vc_UserID
16    , vc_User.UserName
17  FROM vc_VidCast
18  JOIN vc_User ON vc_User.vc_UserID = vc_VidCast.vc_UserID
19  "
20
21  # Send the request to the server and store the results in a variable
22  sqlResult <- sqlQuery(myconn, sqlSelectStatement)
23
24  # Create a list of days of the week for charting later
25  days <- c("Sun", "Mon", "Tues", "Weds", "Thurs", "Fri", "Sat")
26
27  # Create a histogram of durations (appears in the Plots tab)
28  hist(sqlResult$ActualDuration,
29       main="How long are the VidCasts?",
30       xlab="Minutes",
31       ylab="VidCasts",
32       border="blue",
33       col="grey",
34       labels=TRUE
35  )
36
37  # Plot a bar chart of video counts by day of the week
38  dayCounts <- table(sqlResult$StartDayOfWeek)
39  barplot(dayCounts,
40          main="VidCasts by Day of Week",
41          ylab = "Day of Week",
42          xlab = "Count of VidCasts",
43          border = "blue",
44          names.arg = days
45  )
46
47  # Close all connections
48  odbcCloseAll()
49
50  # Fin
51
```

5	**After typing the code, return to lines 29 and 40. Add your netid to the string identifiers on these lines. For instance, if your netid is jpstudent, your line 29 should say:**
	`main="jpstudent How long are the VidCasts?",`
	
	And your line 40 should say:
	`main="jpstudent VidCasts by Day of Week",`
	
	Save your work.

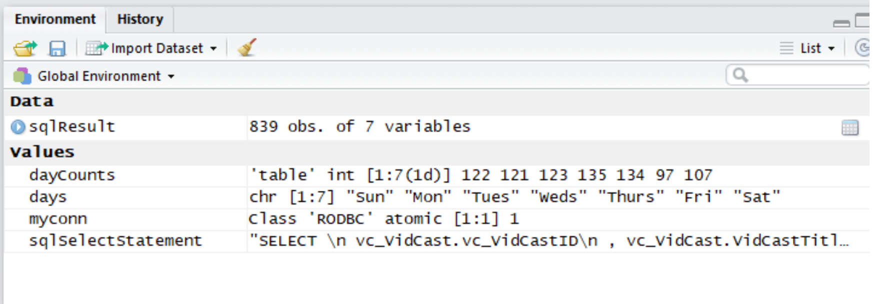
6	**In the toolbar above the script pane, click the Source button. Things should happen.**



```
# Ready RODBC for use in this script
require(RODBC)
```

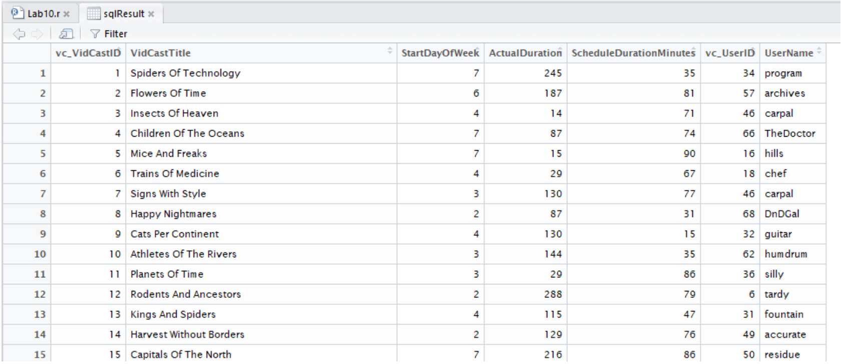
7 **Clicking the Source button executed your script one line at a time. You can check the console to see any messages sent your way as a result.**

The environment pane (upper right) now lists all the variables and data objects created and populated during the execution of your script.



```
Data
sqlResult            839 obs. of 7 variables
Values
dayCounts            'table' int [1:7(1d)] 122 121 123 135 134 97 107
days                 chr [1:7] "Sun" "Mon" "Tues" "Weds" "Thurs" "Fri" "Sat"
myconn               Class 'RODBC' atomic [1:1] 1
sqlSelectStatement   "SELECT \n vc_VidCast.vc_VidCastID\n , vc_VidCast.VidCastTitl…
```

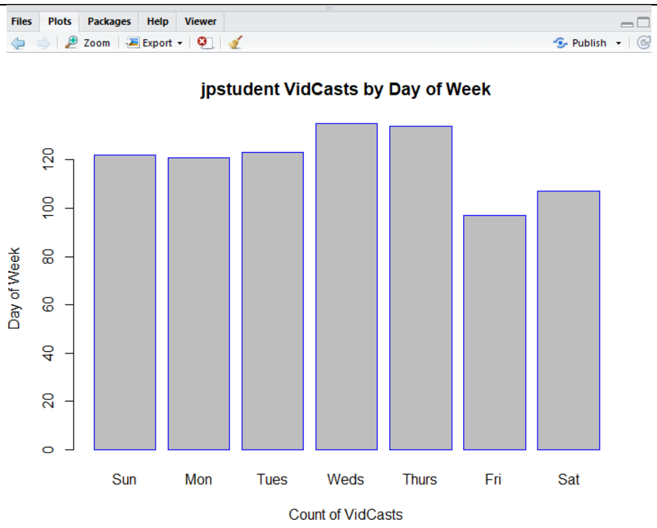
8 **In the Environment tab, click the sqlResult data item. This will open a grid view of the records.**



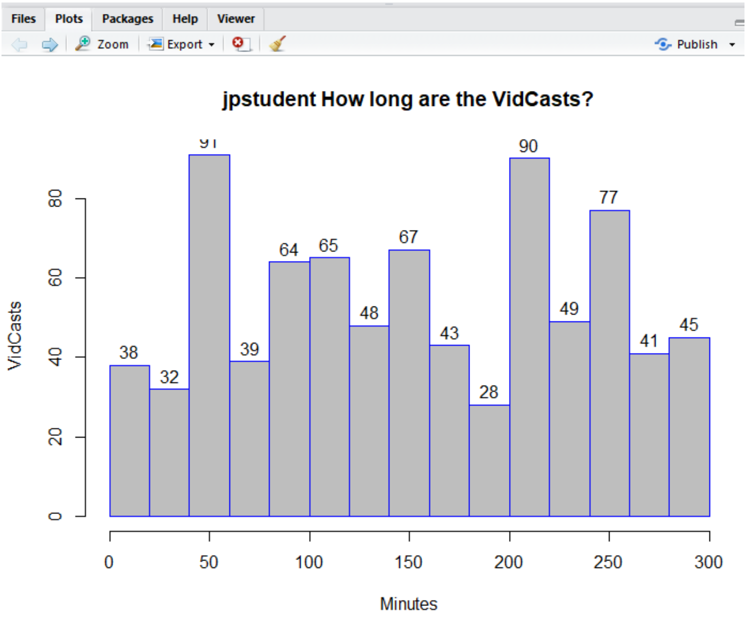
	vc_VidCastID	VidCastTitle	StartDayOfWeek	ActualDuration	ScheduleDurationMinutes	vc_UserID	UserName
1	1	Spiders Of Technology	7	245	35	34	program
2	2	Flowers Of Time	6	187	81	57	archives
3	3	Insects Of Heaven	4	14	71	46	carpal
4	4	Children Of The Oceans	7	87	74	66	TheDoctor
5	5	Mice And Freaks	7	15	90	16	hills
6	6	Trains Of Medicine	4	29	67	18	chef
7	7	Signs With Style	3	130	77	46	carpal
8	8	Happy Nightmares	2	87	31	68	DnDGal
9	9	Cats Per Continent	4	130	15	32	guitar
10	10	Athletes Of The Rivers	3	144	35	62	humdrum
11	11	Planets Of Time	3	29	86	36	silly
12	12	Rodents And Ancestors	2	288	79	6	tardy
13	13	Kings And Spiders	4	115	47	31	fountain
14	14	Harvest Without Borders	2	129	76	49	accurate
15	15	Capitals Of The North	7	216	86	50	residue

9 **In your answers document, create a Part 2 heading. Below it, paste a screenshot of your sqlResult data object (Like step 8 above. Not all data need be shown).**

10 **The Plots tab in the lower right shows a bar chart like this:**

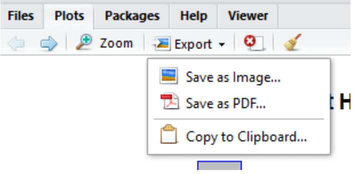


We actually created two plots. In the toolbar of the Plots tab, click the back arrow ⇦ to show the first plot, a histogram of VidCast durations. It should look like this:

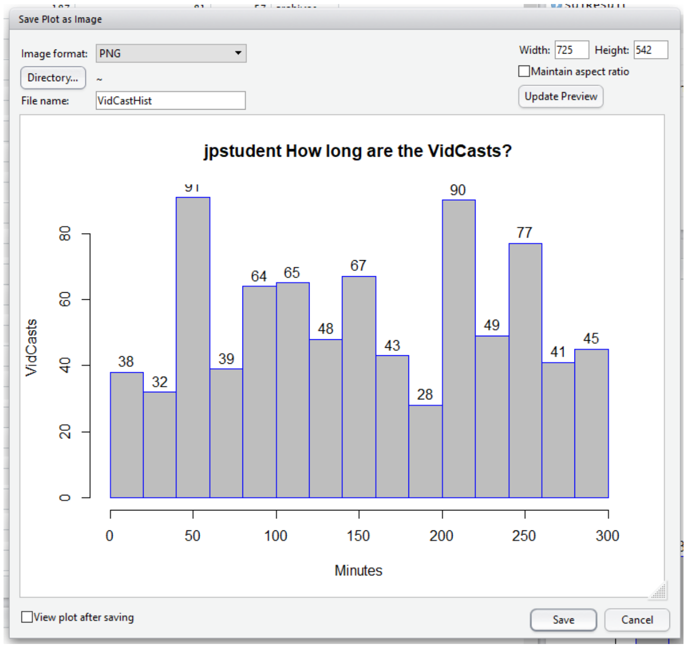


11 **A common workflow with the basic R plotting tools is to generate plots like we did here, export them to an image file, import those files into a more robust image editor such as InkScape or Adobe Illustrator where they can be edited to be more visually appealing as vector graphics.**

For now, we will export them and leave them in their current state. In the toolbar of the Plot tab, click Export, then Save as Image.



In the Save Plot as Image dialog, change the File name to VidCastHist



Save Plot as Image

Image format: PNG

Directory... ~

File name: VidCastHist

Width: 725 Height: 542

☐ Maintain aspect ratio

Update Preview

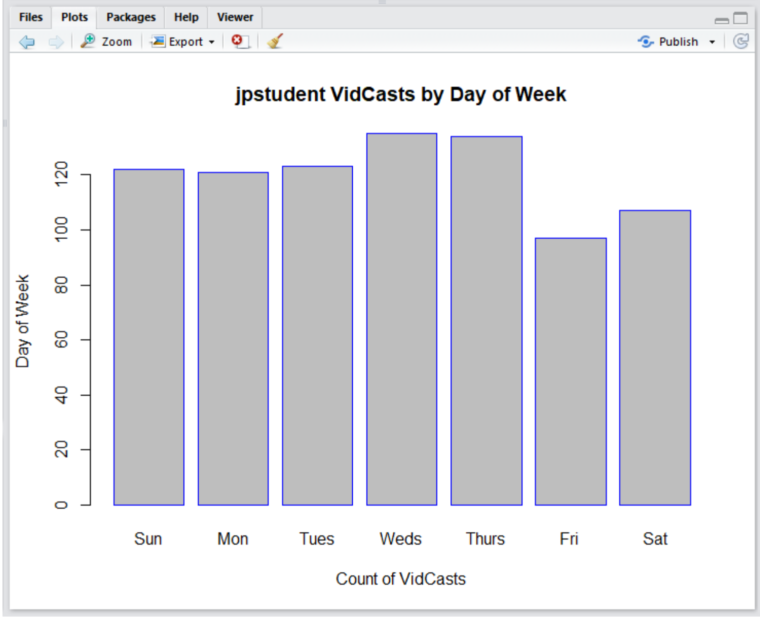
☐ View plot after saving

Save Cancel

OPTIONAL: By default, RStudio saves the plot to your Documents folder. If you would like to change this location, click the Directory button and browse to the folder in which you would like to save the image. Remember where you saved it because you'll need it later.

Click Save, accepting all other defaults.

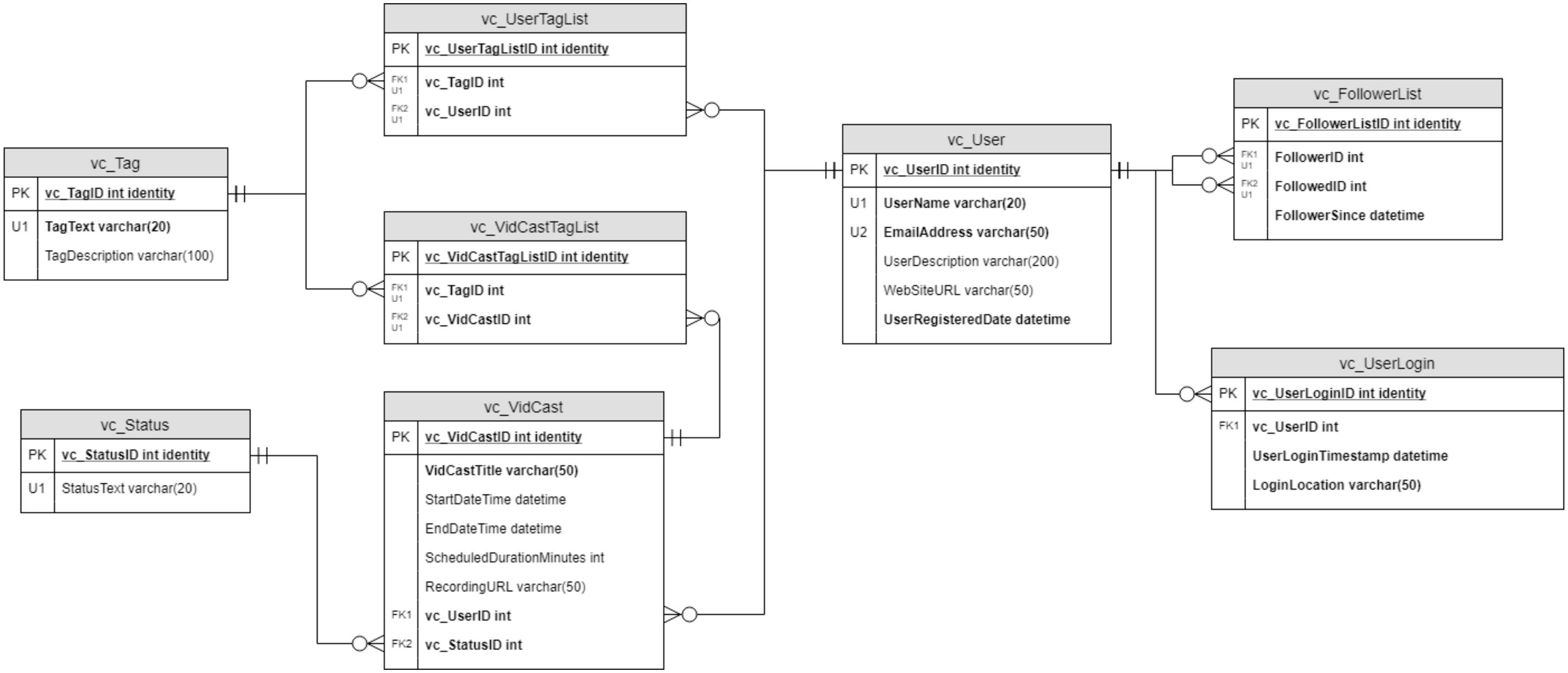
12 **Return to the bar chart by clicking the forward ➪ button in the Plots toolbar.**

	 Files Plots Packages Help Viewer Zoom Export Publish jpstudent VidCasts by Day of Week Day of Week 0 20 40 60 80 100 120 Sun Mon Tues Weds Thurs Fri Sat Count of VidCasts **Using the same steps as before, export this plot, giving it a File name of VidCastCount**
13	**In your answers document, import the two plots you saved in steps 11 and 12.**
14	**After your plots in your answers document, answer the following questions:** **1. In reference to the script in step 4, what does the SQL code on line 12 do?** **2. What is one way to simplify lines 8 through 19? (think lab 8…)**
15	**When you've finished to this step, you can close RStudio.**

What to Submit

After completing Part 2, save your answers document and submit it to the appropriate section on the LMS.

Appendix A – VidCast Logical Model Diagram



For the full diagram, see https://drive.google.com/file/d/1KRqkSvQABuTMXqYAzojTCt9etTSR8Vea/view?usp=sharing

Appendix B – Connecting to Databases in Other Environments

Below is a list of some starting points for using other languages and tools to create data aware applications.

Language	Where used	Link(s)
PHP	General purpose web server scripting language, widely used to serve dynamic web sites	Creating an API: https://www.codeofaninja.com/2017/02/create-simple-rest-api-in-php.html (See also their React tutorial that uses the API built in this REST tutorial) PHP Tutorial: https://www.w3schools.com/php/default.asp *amp Server: http://ampps.com/
Java	Primarily middleware and server-side applications, Android Apps	The Java Tutorial (JDBC Thread): https://docs.oracle.com/javase/tutorial/jdbc/overview/index.html
C/C++	Literally everywhere	MySQL C++ Connector: https://dev.mysql.com/doc/connector-cpp/8.0/en/ Microsoft's Developer Docs for using C/C++ over ODBC: https://docs.microsoft.com/en-us/azure/sql-database/sql-database-develop-cplusplus-simple
Python	Many data science applications, increased use in server-side scripting and client applications. Often used to scrape web data and load to normalized data stores	PyODBC (Python ODBC Bridge): https://github.com/mkleehammer/pyodbc/wiki PyMSSQL (Python SQL Server Connections): http://www.pymssql.org/en/stable/ PyMySQL (Python MySQL connector): https://pymysql.readthedocs.io/en/latest/ Justin Clark's AlphaVantage Stock Database System: https://github.com/justinclarkhome/AlphaVantageStockDB

Excel	Spreadsheets and such.	PowerQuery: https://support.office.com/en-us/article/connect-a-sql-server-database-to-your-workbook-power-query-22c39d8d-5b60-4d7e-9d4b-ce6680d43bad Using MS Query (Old school but very powerful) https://analystcave.com/create-microsoft-query-excel-excel-query/