

Chapter 5

Exploring Navicat and Sequel Pro

Skills you will learn: Features of the basic user interfaces of the Navicat and Sequel Pro front end programs for MySQL.

Unlike Microsoft Access, Navicat and Sequel Pro are not stand-alone database applications, but instead clients that allow you to run commands on the MySQL server. For a fuller discussion of the differences between desktop databases and server databases, see Chapter 5 of *The Data Journalist*.

This tutorial is primarily intended to assist with using Navicat to connect to a MySQL server installed locally on your own computer, and you may need to obtain additional information to connect to a remote server, such as login credentials. You will also need permission to access the remote server and query databases it hosts.

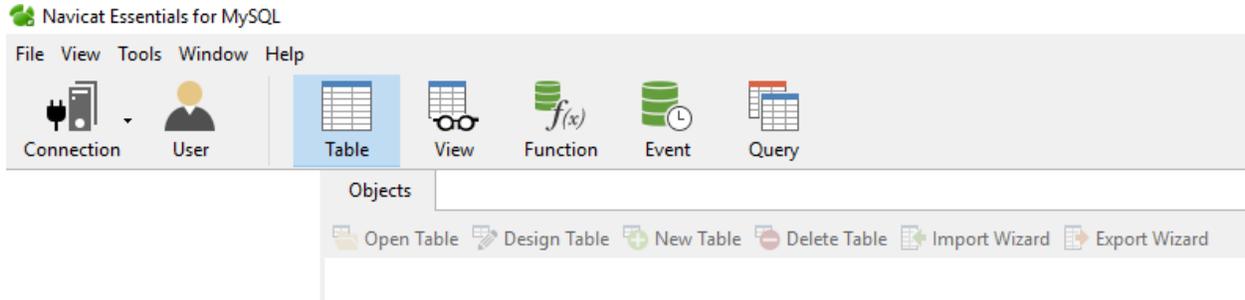
The current version of Navicat as of August 2017 is version 12. Most of the images seen here are from version 11, but the material has been checked for compatibility with version 12 and any small differences have been noted in the text.

Navicat

Navicat is a well-developed and refined front end for MySQL, available in both Windows and Mac versions.

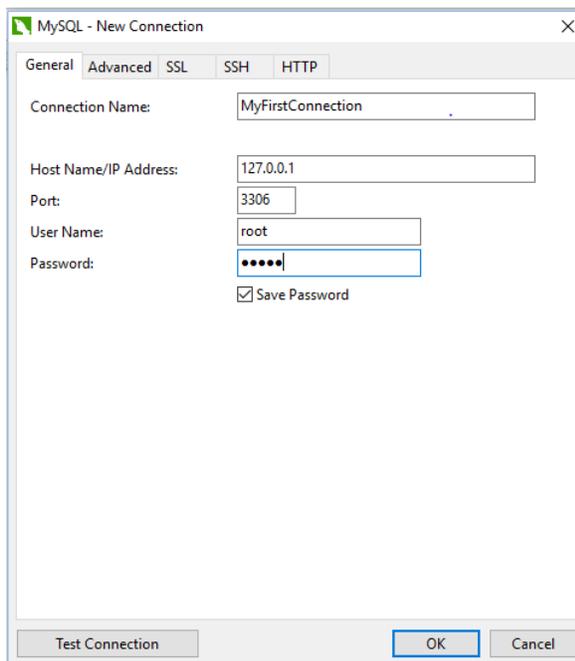
There are several licence levels of Navicat, to connect to MySQL and other server databases. For the purposes of this tutorial, we will focus on the most basic and affordable of these, Navicat Essentials. You can see the complete range of features and capabilities in different versions at www.navicat.com.

When you first open Navicat Essentials, you will see a screen something like this.



The first step you will need to take is to make a connection to the MySQL database. This tutorial assumes you have already installed MySQL on your computer. If you plan to install it on your own computer and have not yet done so, please see the tutorial **Installing and Configuring the MySQL Server**.

To initiate a connection, click on the Connection icon and choose MySQL. Fill out the connection dialogue (General tab) with the settings seen here, replacing the password with the password you set for the MySQL root user. If you didn't set a password, leave the password field blank (we suggest setting a password for security reasons).



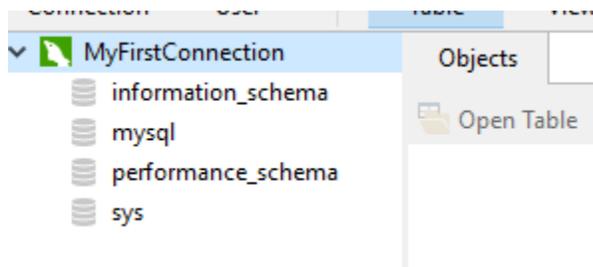
MySQL, as a matter of convention, uses port 3306. The host name can also be entered as “localhost” but entering the IP address of the local machine, 127.0.0.1, will avoid possible connection problems in some configurations. If you are logging into a remote server, you will need the IP address of that server and any instructions from the server provider on logging in.

If you are logging on as a user other than root, change the User Name setting accordingly.

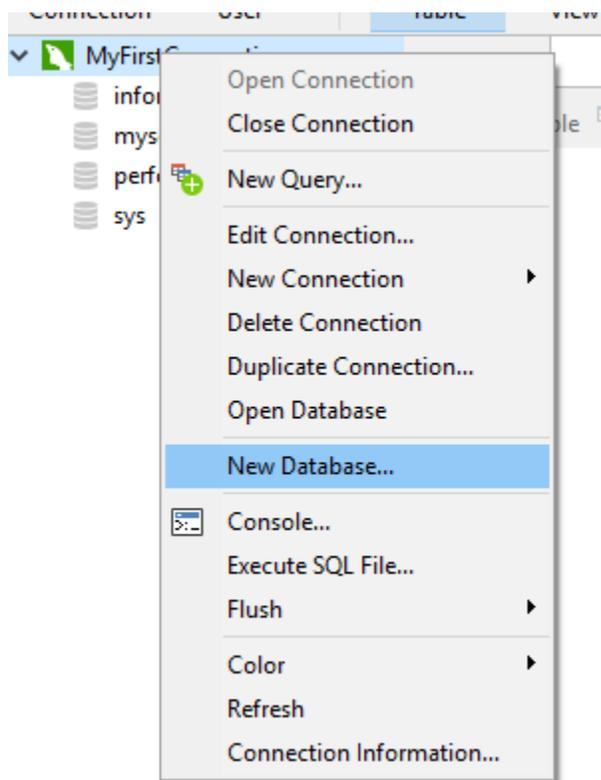
To ensure the connection works, click on Test Connection. If it doesn't work, it may mean you have set the configuration options incorrectly, or the server may not be running.

When you are satisfied everything is correct, click OK.

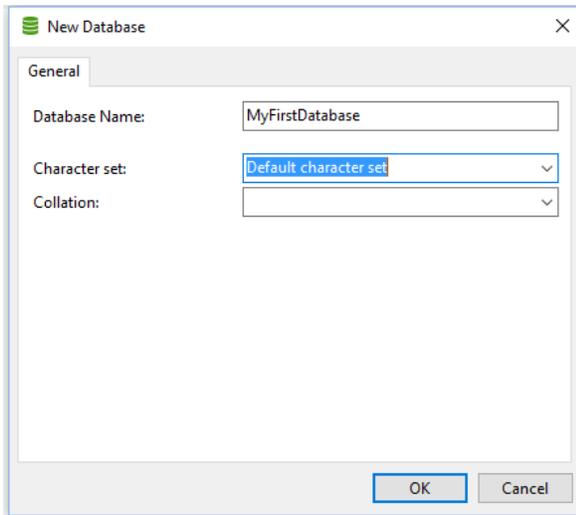
If you double click on the name of your connection, in the table of contents area (in Navicat parlance, called the navigation pane), you will see all of the existing databases on the server. If you have not yet created your own, it will look like the image below, and you will see only the system databases. These should not be altered.



You should probably create a new database for each new project. To create a database, right click on the connection name and choose New Database.



That will bring up the New Database dialogue.

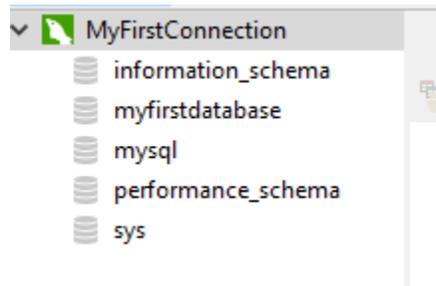


Note: If you have version 12 of Navicat, the character set field may come up as empty to indicate use of the default character set; leaving it empty is the equivalent of entering “Default character set” in version 11.)

You will need to give your database a name. It makes sense that the name be related to the project.

The default character set is Latin1. This will be suitable for most applications using western languages, but if you want, you can choose one of the character sets in the dropdown list, such as Unicode UTF-8 or a language-specific character set such as traditional Chinese, Arabic or Russian, depending on the characters you expect to exist in your data.

When you are done, click OK, and your new database will appear in the table of contents.



To select the database for use, double click on its name.

Please see the tutorial **Making Tables and Importing Data into MySQL** to learn how to create new tables using Navicat.

Once you have created tables, they will appear below the Tables header in the table of contents, if you click on the expansion arrow. They will also appear under the Objects tab when the table icon is selected. They are now ready for analysis.

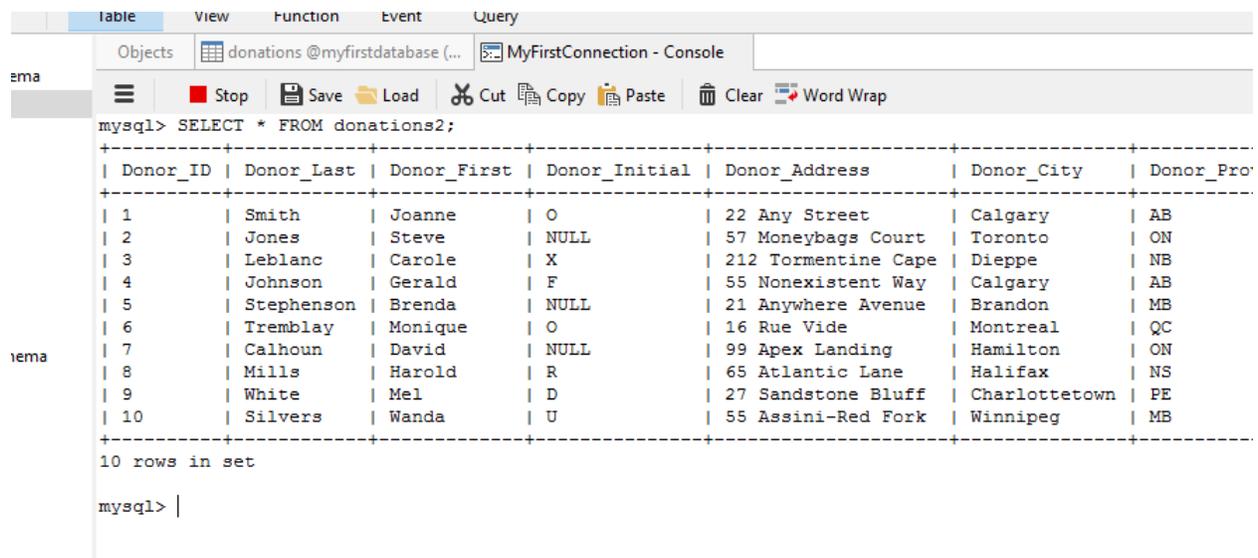
Most of the analysis you will do involves writing queries. The companion tutorial, **Getting Started with Queries in MySQL**, covers this important functionality in Navicat and Sequel Pro.

There is one other really useful feature in Navicat, and that is the ability to switch directly to the MySQL command prompt, for entering SQL queries and other commands such as LOAD DATA INFILE without having to open the command line utility.

Right click on the database name, and choose Console, or go to the Tools menu and choose Console. A console window will open, with functionality that duplicates that of the command line tool, complete with the MySQL> prompt.

You can save commands as well as open previously saved commands.

Here, the console is pictured with a query completed.



The screenshot shows the Navicat MySQL console window. The title bar reads "MyFirstConnection - Console". The menu bar includes "Objects", "donations @myfirstdatabase (...)", and "MyFirstConnection - Console". The toolbar contains icons for "Stop", "Save", "Load", "Cut", "Copy", "Paste", "Clear", and "Word Wrap". The console text shows the command "mysql> SELECT * FROM donations2;" followed by a table of 10 rows. The table has columns: Donor_ID, Donor_Last, Donor_First, Donor_Initial, Donor_Address, Donor_City, and Donor_Pro. Below the table, it says "10 rows in set" and "mysql> |".

Donor_ID	Donor_Last	Donor_First	Donor_Initial	Donor_Address	Donor_City	Donor_Pro
1	Smith	Joanne	O	22 Any Street	Calgary	AB
2	Jones	Steve	NULL	57 Moneybags Court	Toronto	ON
3	Leblanc	Carole	X	212 Tormentine Cape	Dieppe	NB
4	Johnson	Gerald	F	55 Nonexistent Way	Calgary	AB
5	Stephenson	Brenda	NULL	21 Anywhere Avenue	Brandon	MB
6	Tremblay	Monique	O	16 Rue Vide	Montreal	QC
7	Calhoun	David	NULL	99 Apex Landing	Hamilton	ON
8	Mills	Harold	R	65 Atlantic Lane	Halifax	NS
9	White	Mel	D	27 Sandstone Bluff	Charlottetown	PE
10	Silvers	Wanda	U	55 Assini-Red Fork	Winnipeg	MB

If you wish to see a history of commands you have run since opening the connection, click on Tools>History Log.

There are a number of options that can be set from Navicat.

Options is an item under the Tools tab.

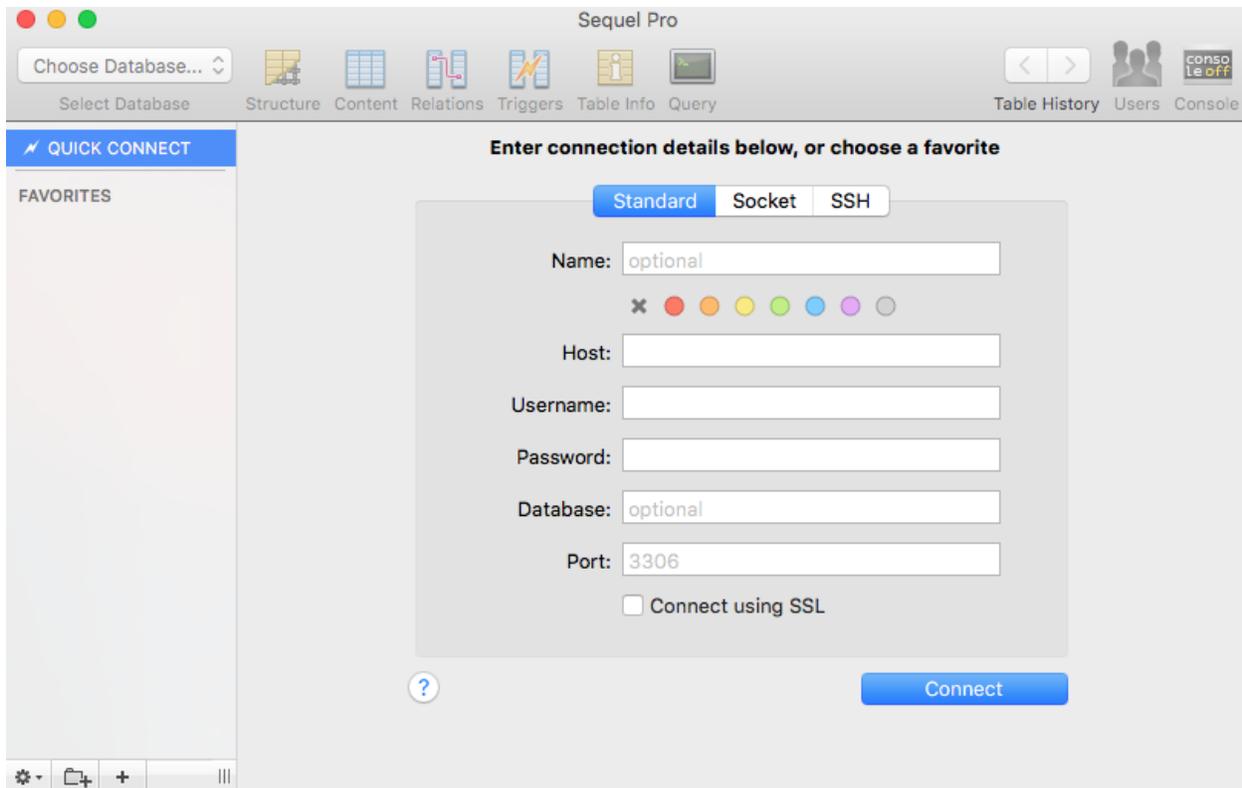
Many of the choices deal with how the interface looks, and we'll leave those to you to explore. However, there are a few worth noting.

If you open Editor under the Appearance tab you can control whether you see line numbers in SQL queries. Line numbers make queries easier to read and are the default.

Under Data and Grid you can enable or disable a limit on the number of records shown in a table or query. The default is 1,000.

Sequel Pro

When you first open Sequel Pro, you are presented with a screen that will allow you to make a connection to the MySQL database.

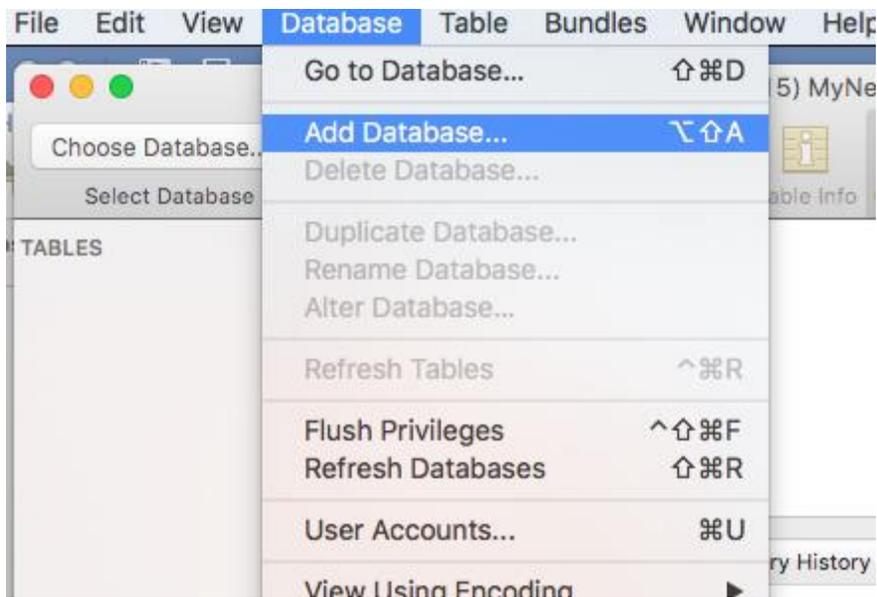


In the standard tab, enter a name for the connection. This can be anything you like. The host should be entered as 127.0.0.1, the username will be root, unless you are connecting as other than the root user, and the password is whatever password you have set for the root user. If no password is set, leave this field blank. The port can be left as the default. Generally speaking, on a Mac, which uses an operating system based on Unix, you can also connect using the socket tab.

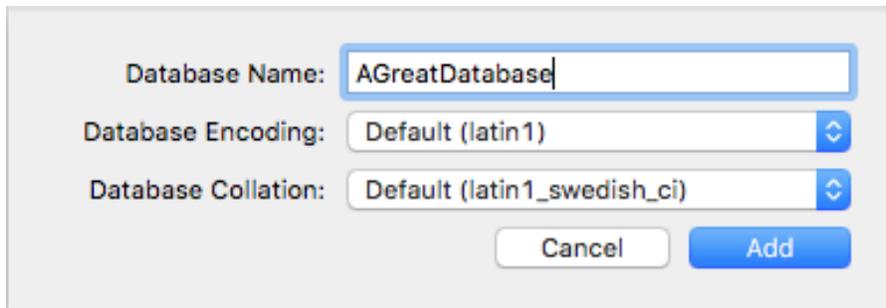
When you have entered the necessary details, click on test connection. If the parameters are entered correctly and the MySQL server is running, you should get the message "connection succeeded."

Click Connect to complete the connection.

To create a database, choose **Add Database** from the Database menu.



Next, give your database a name and set the encoding and collation. The encoding is the character set used to represent characters; in this case the default latin1 is used. The collation determines how characters are compared to each other in comparison operations such as =, > and <>. The ci collations are case insensitive, which is the way journalists would likely prefer it. The default collation should be fine.



When you are done, click Add and the database will be created and a connection established to it.

Taking a look at another database, already created, we can see that tables are listed along the left side of the user interface, in the Tables area. If you click on a table, you can make it active, and greyed out icons across the top of the screen will become available for use.

Donor_ID	Donor_Last	Donor_First	Donor_Initial	Donor_Address	Donor_City	Donor_Prov
1	Smith	Joanne	O	22 Any Street	Calgary	AB
2	Jones	Steve		57 Moneybags Court	Toronto	ON
3	Leblanc	Carole	X	212 Tormentine Cape	Dieppe	NB
4	Johnson	Gerald	F	55 Nonexistent Way	Calgary	AB
5	Stephenson	Brenda		21 Anywhere Avenue	Brandon	MB
6	Tremblay	Monique	O	16 Rue Vide	Montreal	QC
7	Calhoun	David		99 Apex Landing	Hamilton	ON
8	Mills	Harold	R	65 Atlantic Lane	Halifax	NS
9	White	Mel	D	27 Sandstone Bluff	Charlottetown	PE
10	Silvers	Wanda	U	55 Assini-Red Fork	Winnipeg	MB

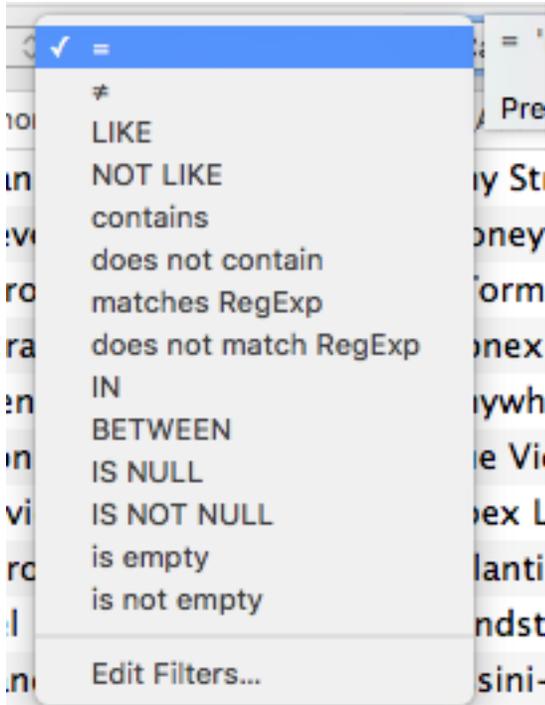
TABLE INFORMATION

- created: 2016-0...
- engine: InnoDB
- rows: 10
- size: 16.0 KiB
- encoding: latin1

In the Content tab, you can view the table fields and records. There is also a Search functionality above the table that allows you to filter records based on criteria that you set.

Donor_Last	Donor_First	Donor_Initial	Donor_Address	Donor_City	Donor_Province	Donor_Age	Dor
Smith	Joanne	O	22 Any Street	Calgary	AB	77	F
Jones	Steve		57 Moneybags Court	Toronto	ON	21	M
Leblanc	Carole	X	212 Tormentine Cape	Dieppe	NB	34	
Johnson	Gerald	F	55 Nonexistent Way	Calgary	AB	55	M

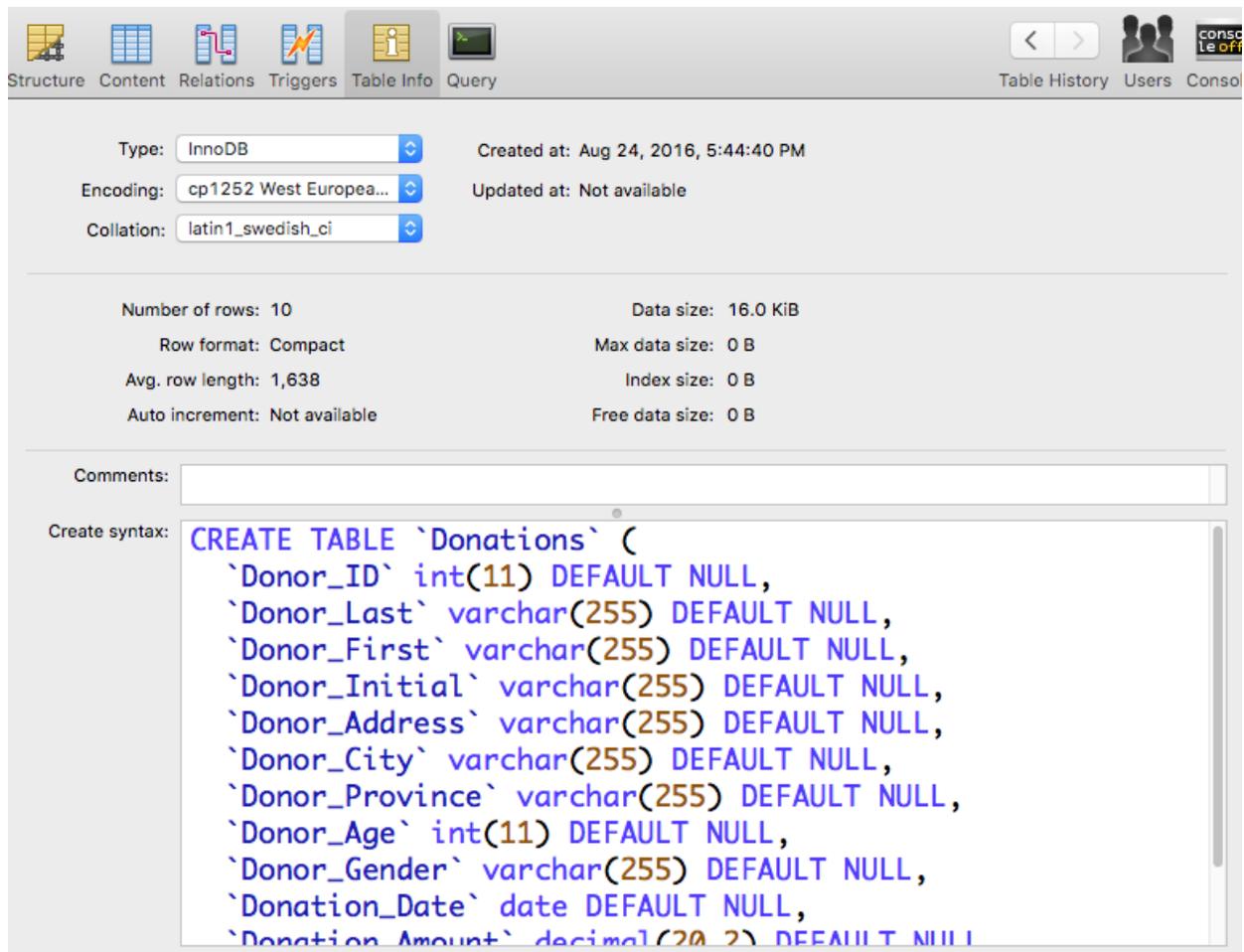
Here a filter has been entered to see only records that have Calgary in the Donor_City field. Clicking **Filter** activates it. Clicking the small X on the right side of the filter window will remove the filter. There is a range of possible operators available in a dropdown list.



The Structure tab shows the fields and data types. You can make changes to the table structure, including the data types and field names, here.

Field	Type	Length	Unsigned	Zerofill	Binary	Allow Null	Key	Default	Extra	Encoding	Collation	Comm..
Donor_ID	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None			
Donor_Last	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None	cp1252	latin1_swe	
Donor_First	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None	cp1252	latin1_swe	
Donor_Initial	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None	cp1252	latin1_swe	
Donor_Addr...	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None	cp1252	latin1_swe	
Donor_City	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None	cp1252	latin1_swe	
Donor_Prov...	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None	cp1252	latin1_swe	
Donor_Age	INT	11	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None			
Donor_Gen...	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None	cp1252	latin1_swe	
Donation_D...	DATE		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None			
Donation_A...	DECIMAL	20,2	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None			
PartyCode	VARCHAR	255	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		NULL	None	cp1252	latin1_swe	

The Table Info tab does as its name suggests; it provides metadata about the table, including the CREATE TABLE statement that was used to create it.



The query icon is used to open the query window, in which SQL queries are written. See the tutorial **Getting Started with Queries in MySQL** for a full discussion of writing queries using Sequel Pro.

Sequel Pro allows you to control many aspects of your MySQL database and database objects. Many of the functions are beyond the scope of this tutorial. However, we will review some of the most important tools.

Let's first look at the File menu:

File	Edit	View	Database	Tab
New Connection Window				⌘N
New Connection Tab				⌘T
Add To Favorites				⇧⌘A
Open...				⌘O
Open Recent				▶
Save Connection				⌘S
Save Session				⌘⇧⌘S
Save Query				⌘⇧S
Import...				⇧⌘I
Export...				⇧⌘E
Close Window				⇧⌘W
Close Tab				⌘W
Print...				⌘P

The New Connection Window and New Connection tab options allow you to create a new connection to a MySQL server in either a window or tab. You are unlikely to do this in the normal course of a journalist's work..

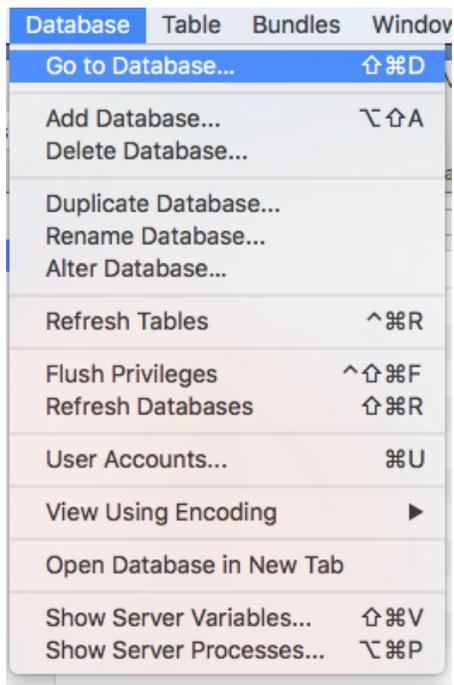
The Open option allows you to open an existing query, while Save Query allows you to save the text of a query.

Import allows you to import external data, or a MySQL dump file.

The View menu largely duplicates the large icons on the main toolbar.

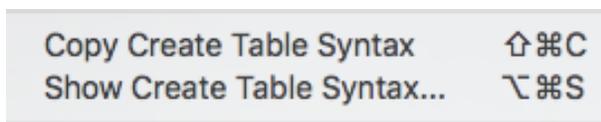
View	Database	Table	Bund
Show Tab Bar			⇧⌘T
Table Structure			⌘1
Table Content			⌘2
Table Relations			⌘3
Table Triggers			⌘4
Table Info			⌘5
Custom Query			⌘6
Back In History			⌘⇧←
Forward In History			⌘⇧→
Show Console			⇧⌘K
Clear Console			⌘K
Show Navigator			⌘⇧N
Display Binary Data as Hex			
Hide Toolbar			⌘⇧T
Customize Toolbar...			
Enter Full Screen			⌘⇧F

The Database menu provides options for administering the MySQL database.



Here, you can add and delete databases, duplicate databases, rename them, or alter their attributes. You can also refresh the view of tables and databases within the user input, and open a new database in a separate tab. You can also create new user accounts in MySQL.

The Table menu provides two highly useful options (we're not showing the whole menu here).



The Copy Create Table Syntax allows you to copy the SQL command that would recreate the table. The Show Create Table Syntax option simply allows you to view it.

Here is an example for the Political Donations database we've been using.

Create syntax for TABLE 'Donations'

```
1 CREATE TABLE `Donations` (  
2   `Donor_ID` int(11) DEFAULT NULL,  
3   `Donor_Last` varchar(255) DEFAULT NULL,  
4   `Donor_First` varchar(255) DEFAULT NULL,  
5   `Donor_Initial` varchar(255) DEFAULT NULL,  
6   `Donor_Address` varchar(255) DEFAULT NULL,  
7   `Donor_City` varchar(255) DEFAULT NULL,  
8   `Donor_Province` varchar(255) DEFAULT NULL,  
9   `Donor_Age` int(11) DEFAULT NULL,  
10  `Donor_Gender` varchar(255) DEFAULT NULL,  
11  `Donation_Date` date DEFAULT NULL,  
12  `Donation_Amount` decimal(20,2) DEFAULT NULL,  
13  `PartyCode` varchar(255) DEFAULT NULL  
14 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

Save As...

Copy

Having access to the syntax is especially useful for either recreating a similar table, or quickly creating a table with different field names or data types.

Before we conclude, it's worth addressing Bundles. Bundles are a bit like add-ons in some other applications; extra code that provides additional functionality. They are not particularly well supported, but there is limited information available on this GitHub page:

<https://github.com/sequelpro/Bundles>

There is one bundle, however, that is built in to the distribution of Sequel Pro. It allows you to convert any table or query display to JSON.

Right click on the table or query result, and choose **BUNDLES>Copy>Copy as JSON**.

Donor_ID	Donor_Last	Donor_First	Donor_Initial	Donor_Address	Donor_City	Donor_Province
1	Smith	Joanne	O	22 Any Street	Calgary	AB
2	Jones			57 Moneybags Court	Toronto	ON
3	Leblanc			212 Tormentine Cape	Dieppe	NB
4	Johnson			55 Nonexistent Way	Calgary	AB
5	Stephens			21 Anywhere Avenue	Brandon	MB
6	Tremblay			16 Rue Vide	Montreal	QC
7	Calhoun			99 Apex Landing	Hamilton	ON
8	Mills			65 Atlantic Lane	Halifax	NS
9	White					PE
10	Silvers	Wanda	U	55 Assini-Red Fork	Winnipeg	MB

The JSON will be copied to the clipboard. From there, you can save it in a plain text file with a JSON extension. Here's one row of the above copied using the Copy as JSON bundle.

```
{
  "data":
  [
    {
      "Donor_ID": 1,
      "Donor_Last": "Smith",
      "Donor_First": "Joanne",
      "Donor_Initial": "O",
      "Donor_Address": "22 Any Street",
      "Donor_City": "Calgary",
      "Donor_Province": "AB",
      "Donor_Age": 77,
      "Donor_Gender": "F",
      "Donation_Date": "2016-08-21",
      "Donation_Amount": 1500,
      "PartyCode": "Red"
    }
  ]
}
```