

MySQL data connection setup in Tableau

This document will detail the steps necessary for connecting to a Jamf Pro MySQL database as a Tableau data source.

One of the easier ways to get Jamf Pro data is to connect directly to Jamf Pro's MySQL database. While the process for doing this is fairly straight forward, some of the steps are particular and should be done in a specific way. For example, allowing access to the MySQL database in a relatively secure fashion.

Even after a data connection is established and verified, how the data interacts isn't always obvious. For example, in my demo visualizations I'm using the `jamfsoftware.computers_denormalized` and `jamfsoftware.mobile_devices_denormalized` tables. With these I'm generating something resembling a general inventory visualization. While many of the fields in the two tables are similar, the data in them is actually quite different. The two tables can't be easily joined to produce a unified visualization. Luckily, there are ways to produce something very close to a unified visualization.

Preparing MySQL

The first thing we have to do is prepare MySQL to both allow external connection in general and a specific user connection to gather data from MySQL. By default the Jamf Pro MySQL instance does not allow external connection or have any other users than root and those used to connect to Jamf Pro.

▼ [Allowing connections to MySQL](#)

By default, external connections are not possible to a Jamf Pro MySQL instance. The default IP of the MySQL server is '127.0.0.1', so external requests aren't even seen. We must change the MySQL configuration file and restart so external connections are possible.

```
> cd /etc/mysql/mysql.conf.d
> sudo vi mysqld.cnf
```

The exact location of the 'mysqld.cnf' file that needs editing may vary, but it should be located somewhere in '/etc/mysql'.

Comment out 'bind-address'.

```
43 #opener
44 bind-address            = 127.0.0.1
45 #

to

43 #opener
44 #bind-address          = 127.0.0.1
45 #
```

Restart mysql.

```
> sudo service mysql restart
```

▼ [Create limited MySQL user](#)

We should now create a MySQL user specifically for accessing MySQL from Tableau. There are a couple reasons for doing this. The first is to limit access to the databases and tables in MySQL. Tableau should only need access to the 'jamfsoftware' database and should only need 'read' (SELECT) access to tables within that database. Another reason is logging usage and access from Tableau. If all requests from

Tableau come from specific users, it is easier to track any impact Tableau usage may have on Jamf Pro.

First log in to MySQL. Where <admin> is a MySQL admin user or 'root'.

```
> mysql -u <admin> -p
```

Now create a new user. <host> can be an IP address or a [FQDN](#) .

```
: CREATE USER '<username>'@'<host>' IDENTIFIED BY '<password>';
```

examples

```
: CREATE USER 'tableau'@'foo.bar.org' IDENTIFIED BY 'alpha123';
```

```
: CREATE USER 'tableau'@'10.20.30.40' IDENTIFIED BY 'alpha123';
```

If SSL certificates are in use, the user can be created requiring their use for connections.

```
: CREATE USER '<username>'@'<host>' IDENTIFIED BY '<password>' REQUIRE  
SSL;
```

Give the user access to specific databases and tables within MySQL.

```
: GRANT SELECT ON jamfsoftware.<table> TO '<username>'@'<host>';
```

examples:

```
: GRANT SELECT ON jamfsoftware.computers_denormalized TO  
'tableau'@'foo.bar.org';
```

```
: GRANT SELECT ON jamfsoftware.mobile_devices_denormalized TO  
'tableau'@'foo.bar.org';
```

```
: GRANT SELECT ON jamfsoftware.buildings TO 'tableau'@'foo.bar.org';
```

```
: GRANT SELECT ON jamfsoftware.departments TO 'tableau'@'foo.bar.org';
```

Note: While it is possible to grant a user access to all tables within a database with wildcards, doing so carries some risk. The user 'tableau' could be granted SELECT access to all tables and fields within the 'jamfsoftware' database. However, this includes the Jamf Pro 'users' table and all the user subtables. When possible, MySQL users should only be granted the access required to produce the desired Tableau visualizations.

▼ [Create SSL cert for MySQL connection \(ver. 5.6 and before, optional\)](#)

[MySQL Documentation : creating ssl files](#)

Connecting to a MySQL database remotely can be done several ways. Tableau uses a standard connection to the default MySQL port 3306.

While this method works well, it is also insecure. Credentials to log in to MySQL are passed via an unencrypted connection and could be snooped. Luckily, there is a way to connect to MySQL via a secure encrypted connection.

Care should be taken protecting the certs, especially the private keys. With the private keys, anyone could generate client certs which could allow access to MySQL server.

This is the older, more complicated method for generating SSL certs for MySQL. With version 5.7 and newer there is a simpler method.

The steps below assume no certs are currently in use for MySQL.

First, create a certificate authority certificate.

```
> cd /etc/mysql
> mkdir certs && cd certs

> openssl genrsa 2048 > ca-key.pem
> openssl req -new -x509 -nodes -days 3600 -key ca-key.pem -out ca.pem
```

Follow the instructions presented

```
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

```
-----
```

```
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:MN
Locality Name (eg, city) []:Frostbite Falls
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Mooseworks
Organizational Unit Name (eg, section) []:hats
Common Name (e.g. server FQDN or YOUR name) []:hats.moosensqurl.net
Email Address []:help@moosensqurl.net
```

Create server certificate. These are the certs that are used by the server to establish a secure connection.

Sign and add a passphrase for server certificate. 'server-cert.pem' is the public key. 'server-key.pem' is the private key.

```

> openssl req -newkey rsa:2048 -days 3600 -nodes -keyout
server-key.pem -out server-req.pem
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:MN
Locality Name (eg, city) []:Frostbite Falls
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Mooseworks
Organizational Unit Name (eg, section) []:hats
Common Name (e.g. server FQDN or YOUR name) []:hats.moosensqurl.net
Email Address []:help@moosensqurl.net

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:rocky001BullW
An optional company name []:Mooseworks

> openssl rsa -in server-key.pem -out server-key.pem
> openssl x509 -req -in server-req.pem -days 3600 -CA ca.pem -CAkey
ca-key.pem -set_serial 00001 -out server-cert.pem

```

Create client certificates. These are the certs that will be used remotely to log in to MySQL securely.

The client challenge password should be different from server challenge password. There could be multiple client certs for a single server cert.

```

> openssl req -newkey rsa:2048 -days 3600 -nodes -keyout
client-key.pem -out client-req.pem

-----
You are about to be asked to enter information that will be
incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or
a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:MN
Locality Name (eg, city) []:Frostbite Falls
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Mooseworks
Organizational Unit Name (eg, section) []:hats
Common Name (e.g. server FQDN or YOUR name) []:hats.moosensqurl.net
Email Address []:help@moosensqurl.net

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:boris991Natasha&
An optional company name []:Mooseworks

> openssl rsa -in client-key.pem -out client-key.pem
> openssl x509 -req -in client-req.pem -days 3600 -CA ca.pem -CAkey
ca-key.pem -set_serial 00001 -out client-cert.pem

```

Verify new certificates against certificate authority.

```

> openssl verify -CAfile ca.pem server-cert.pem client-cert.pem
server-cert.pem: OK
client-cert.pem: OK

```

Copy certs to MySQL data directory.

▼ [Find MySQL data directory](#)

The MySQL data directory can be found with a command to mysql.

```
> mysql -u <admin user> -p -e 'SHOW VARIABLES WHERE Variable_Name LIKE "%dir"'
```

```
> cd /etc/mysql/certs

## if certs file doesn't exist is data directory
> sudo mkdir /var/lib/mysql/certs
> sudo chown mysql:mysql /var/lib/mysql/certs
> sudo cp ca.pem server-cert.pem server-key.pem /var/lib/mysql/certs

## change ownership of certs so MySQL can access them
> sudo chown -R mysql:mysql /var/lib/mysql/certs
```

Now configure MySQL to use the new certs. The location and name of the MySQL configuration file can vary by individual installation.

MySQL : Using secure connections

▼ Find 'mysqld.cnf' files

> locate mysqld.cnf

```
> cd /etc/mysql/mysql.conf.d
> sudo vi mysqld.cnf
```

'mysqld.cnf'

```
[mysqld]
...
# mysql ssl certs
ssl-ca=/var/lib/mysql/certs/ca.pem
ssl-cert=/var/lib/mysql/certs/server-cert.pem
ssl-key=/var/lib/mysql/certs/server-key.pem
```

Restart mysql.

```
> sudo service mysql restart
```

Copy client cert, client private key, and certificate authority cert to remote system. Once copied, these certs can be used to establish a secure connection to MySQL.

```
### on remote system
> scp <user>@<mysql host>:client-cert.pem .
> scp <user>@<mysql host>:client-key.pem .
> scp <user>@<mysql host>:ca.pem .
```

✓ Create SSL cert for MySQL connection (ver 5.7 and after, optional)

With version 5.7 of MySQL there is a much simpler method for generating SSL certs for use with MySQL. A single command will generate all required certificates in a specified directory. Care should be taken protecting the certs, especially the private keys. With the private keys, anyone could generate client certs which could allow access to MySQL server.

Generate certs.

```
> sudo mysql_ssl_rsa_setup --datadir=<mysql data directory> --verbose
```

```
## if certs file doesn't exist is data directory
```

```
> sudo mkdir /var/lib/mysql/certs
> sudo chown mysql:mysql /var/lib/mysql/certs
```

example:

```
> sudo mysql_ssl_rsa_setup --datadir=/var/lib/mysql/certs --verbose
```

```
2017-03-10 18:33:12 [NOTE]      Destination directory:
/var/lib/mysql/certs
2017-03-10 18:33:12 [NOTE]      Executing : openssl version
OpenSSL 1.0.2g  1 Mar 2016
2017-03-10 18:33:12 [NOTE]      Executing : openssl req -newkey rsa:2048
-days 3650 -nodes -keyout ca-key.pem -subj
/CN=MySQL_Server_5.7.17_Auto_Generated_CA_Certificate -out ca-req.pem
&& openssl rsa -in ca-key.pem -out ca-key.pem
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'ca-key.pem'
-----
writing RSA key
2017-03-10 18:33:12 [NOTE]      Executing : openssl x509 -sha256 -days
3650 -set_serial 1 -req -in ca-req.pem -signkey ca-key.pem -out ca.pem
Signature ok
subject=/CN=MySQL_Server_5.7.17_Auto_Generated_CA_Certificate
Getting Private key
2017-03-10 18:33:12 [NOTE]      Executing : openssl req -newkey rsa:2048
-days 3650 -nodes -keyout server-key.pem -subj
/CN=MySQL_Server_5.7.17_Auto_Generated_Server_Certificate -out
```

```

server-req.pem && openssl rsa -in server-key.pem -out server-key.pem
Generating a 2048 bit RSA private key
.....+++
.....+++
writing new private key to 'server-key.pem'
-----
writing RSA key
2017-03-10 18:33:12 [NOTE]      Executing : openssl x509 -sha256 -days
3650 -set_serial 2 -req -in server-req.pem -CA ca.pem -CAkey
ca-key.pem -out server-cert.pem
Signature ok
subject=/CN=MySQL_Server_5.7.17_Auto_Generated_Server_Certificate
Getting CA Private Key
2017-03-10 18:33:12 [NOTE]      Executing : openssl req -newkey rsa:2048
-days 3650 -nodes -keyout client-key.pem -subj
/CN=MySQL_Server_5.7.17_Auto_Generated_Client_Certificate -out
client-req.pem && openssl rsa -in client-key.pem -out client-key.pem
Generating a 2048 bit RSA private key
.....
.....+++
.....
.....+++
writing new private key to 'client-key.pem'
-----
writing RSA key
2017-03-10 18:33:12 [NOTE]      Executing : openssl x509 -sha256 -days
3650 -set_serial 3 -req -in client-req.pem -CA ca.pem -CAkey
ca-key.pem -out client-cert.pem
Signature ok
subject=/CN=MySQL_Server_5.7.17_Auto_Generated_Client_Certificate
Getting CA Private Key
2017-03-10 18:33:12 [NOTE]      Executing : openssl verify -CAfile
ca.pem server-cert.pem client-cert.pem
server-cert.pem: OK
client-cert.pem: OK
2017-03-10 18:33:12 [NOTE]      Executing : openssl genrsa -out
private_key.pem 2048
Generating RSA private key, 2048 bit long modulus
.....+++
.....+++
e is 65537 (0x10001)
2017-03-10 18:33:12 [NOTE]      Executing : openssl rsa -in

```



```
private_key.pem -pubout -out public_key.pem
writing RSA key
2017-03-10 18:33:12 [NOTE]      Success!
```

Change ownership of certs so MySQL can access them.

```
> sudo chown -R mysql:mysql /var/lib/mysql/certs
```

Copy client cert, client private key, and certificate authority cert to remote system. Once copied, these certs can be used to establish a secure connection to MySQL.

```
### on remote system
> scp <user>@<mysql host>:client-cert.pem .
> scp <user>@<mysql host>:client-key.pem .
> scp <user>@<mysql host>:ca.pem .
```

✓ Testing the connection

There are several tools for testing a MySQL connection to verify it functions. One could just try out the connection with Tableau. Tableau Desktop provides some logging about connections and information on what happens on failure. However, having an interactive connection tool can be very useful.

One useful tool in this regard is 'Sequel Pro'. Available for macOS, 'Sequel Pro' allows creation of database connections, browsing through available databases, even running SQL commands against tables.

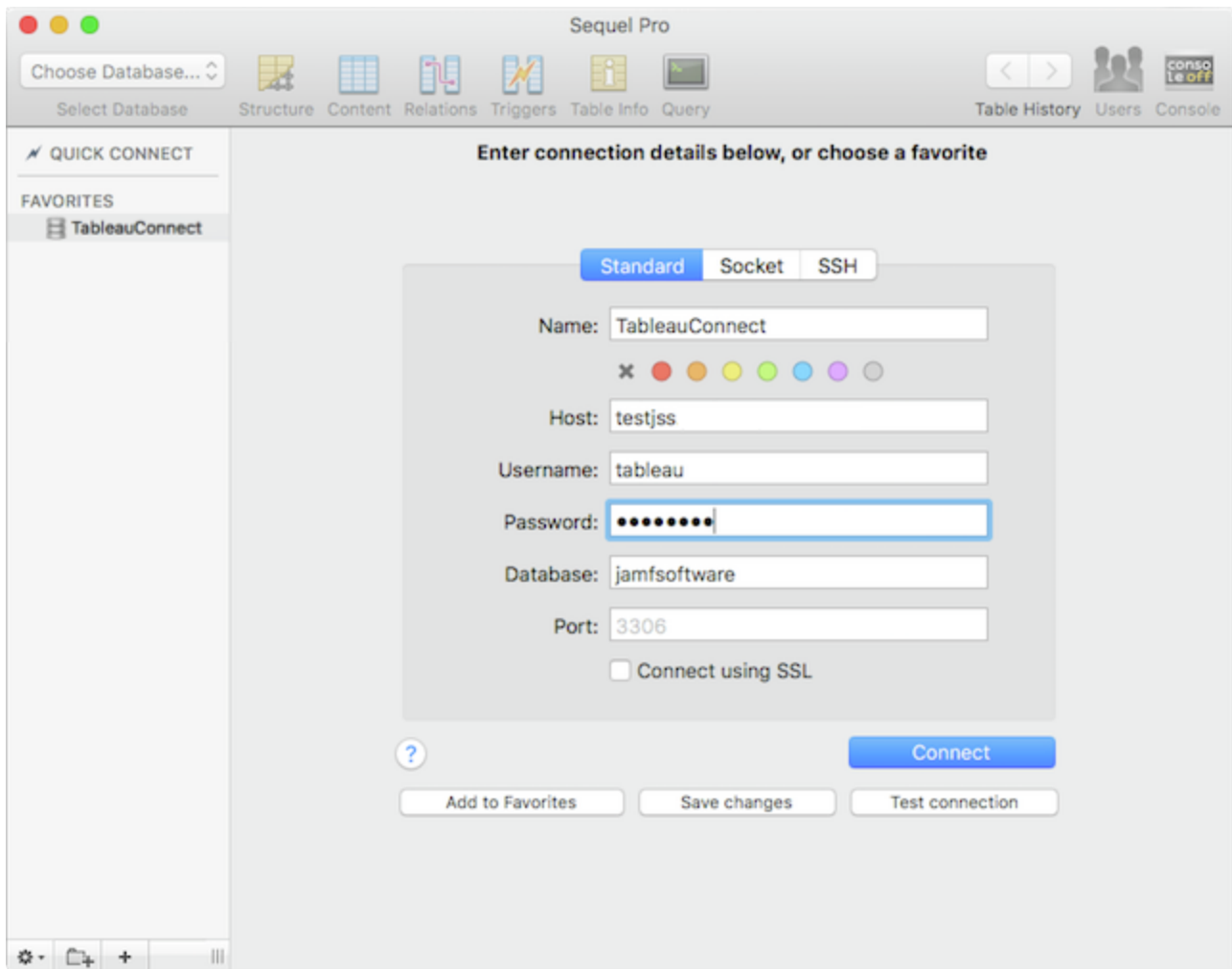
Sequel Pro

A more cross-platform solution would be MySQL's own tool, 'MySQL Workbench'. 'MySQL Workbench' has more functionality than 'Sequel Pro', but is often more complex to use.

MySQL Workbench

To demonstrate how to test a MySQL connection using 'Sequel Pro'.

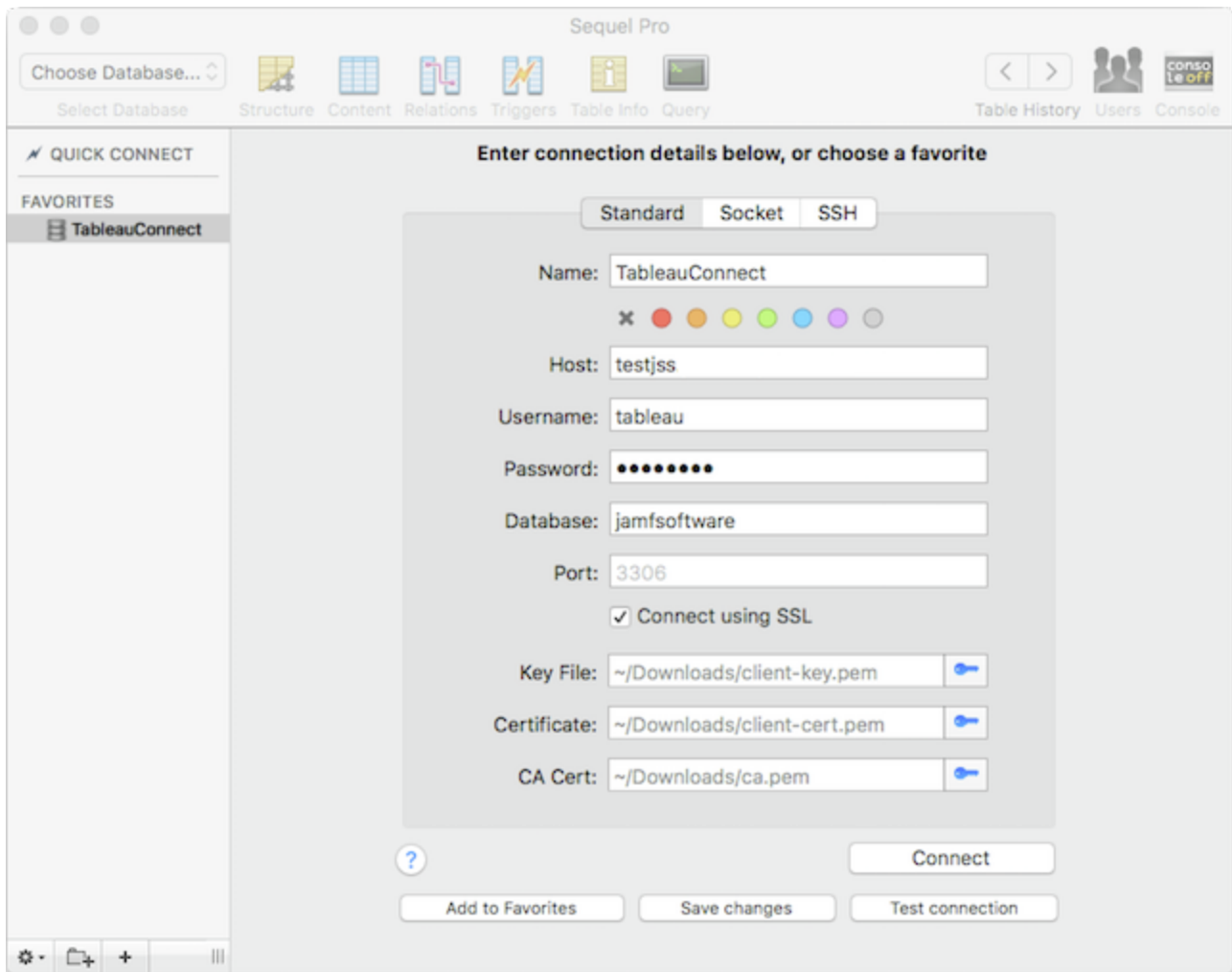
First look at the basic connection screen.



On this screen:

- **'Name'** is the label for the connection.
- **'Host'** is the fully qualified host name of the MySQL server.
- **'Username'** is the username used to log in to MySQL.
- **'Password'** is the password used to log in to MySQL.
- **'Database'** is an optional field. If a valid database name is entered here, when connecting to MySQL that database will automatically be opened. If left blank, a list of valid databases is shown on connection.
- **'Port'** is auto-populated with the default port for MySQL connections. It can be changed, but generally isn't.
- The **'Connect using SSL'** checkbox isn't required. However, connecting to MySQL via SSL is a good idea.

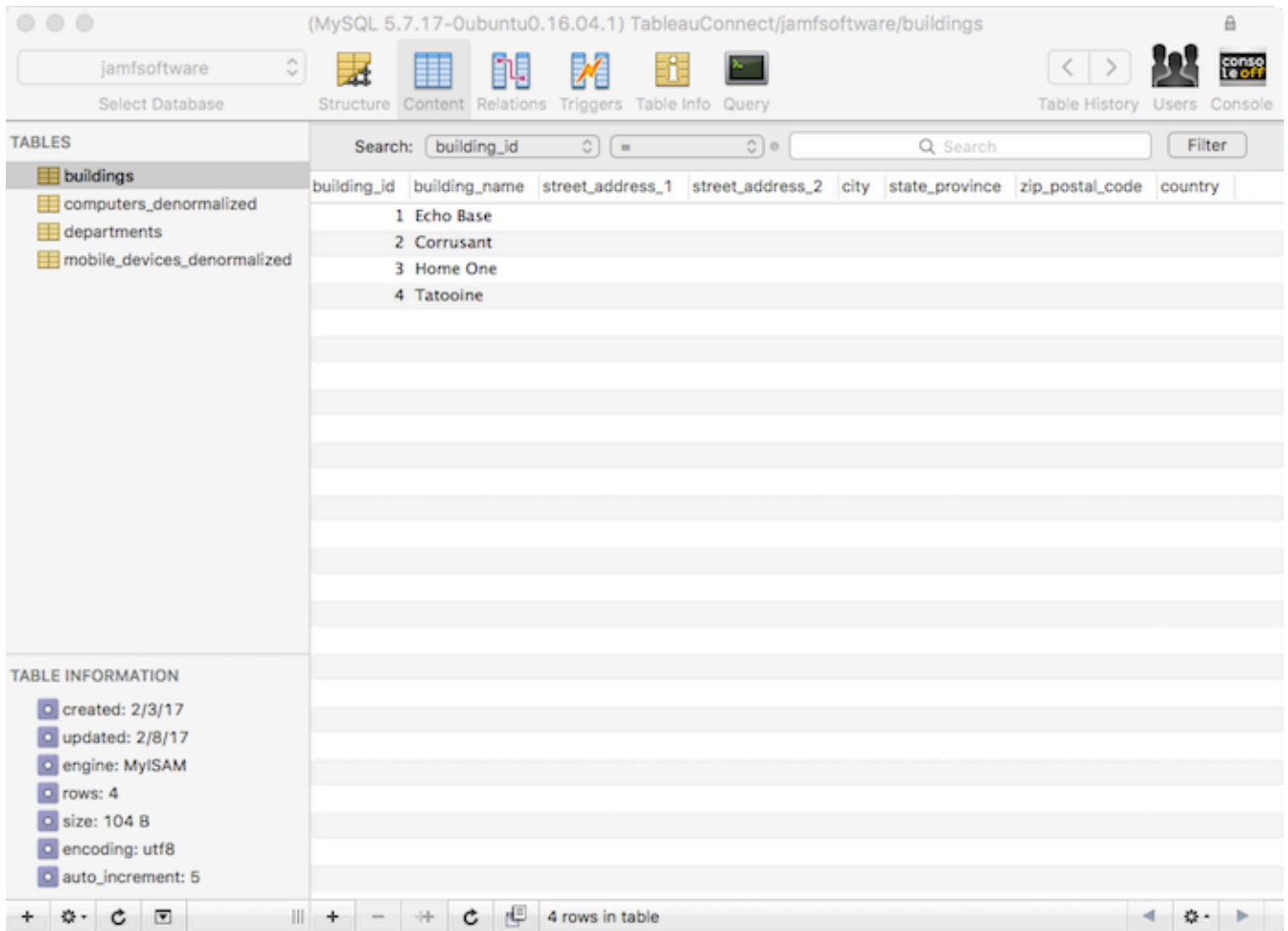
Using the certificates generated above, a secure connection to MySQL can be made.



Three certificates or encryption keys are needed by 'Sequel Pro' to establish a secure connection.

- The client private key file: **client-key.pem**
- The client certificate file: **client-cert.pem**
- The public certificate authority file: **ca.pem**

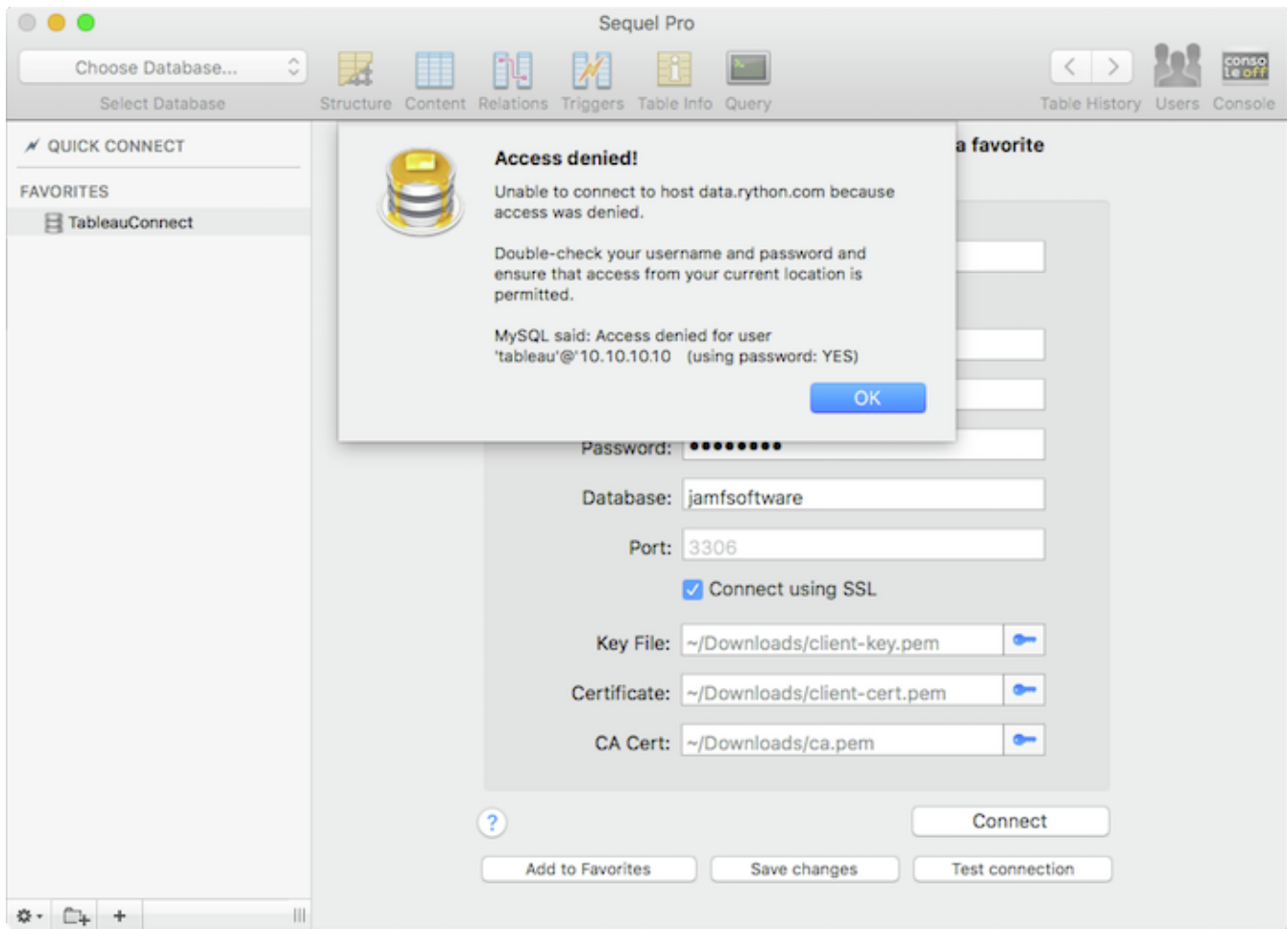
If a connection is successful, 'Sequel Pro' will show the databases and tables this user has access to. If the connection is set to automatically use a database, the available tables for that database will be shown.



General areas on this screen:

- **'Select Database'** shows this connection's available databases.
- **'Tables'** shows the database tables available to this connection.
- **'Table Information'** show some basic information for the selected table.
- The main panel can show multiple kinds of information about a table. In this case the **'Content'** of the database table is show.

If the connection fails, a message like the following will be displayed.



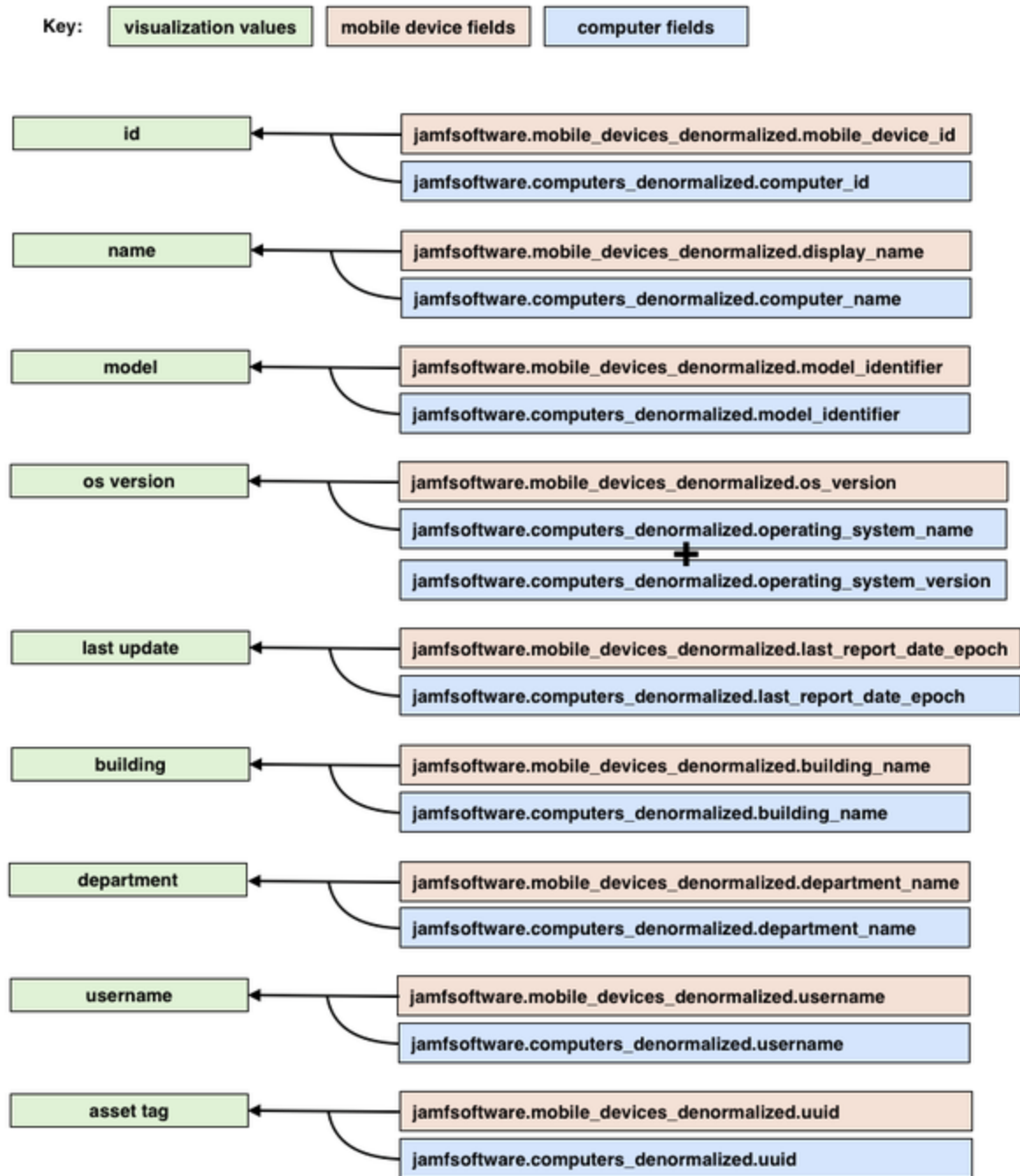
Connection failure can happen for many reasons. Some of them are:

- The username or password may be incorrect.
- The MySQL server may be down.
- The user's permissions may not allow a connection from the client system.
- The certificates may be invalid for the server system.

▼ [Trace database fields \(optional\)](#)

While it isn't required, it can be useful to trace through or map values used in a Tableau visualization. It's not always obvious where the data we are looking for resides in the Jamf Pro MySQL database.

Using a general inventory report as an example, determine where in the MySQL database the values we want reside. Tracing the fields is also useful for determining how to restrict access for a MySQL user.



Using Tableau

Once the Jamf Pro MySQL data source is set up, data can be imported to Tableau to create visualizations. For purposes of this example, we will create a general inventory visualization. This example will be using Tableau Desktop 10.2

[Tableau Documentation : Connecting to MySQL](#)

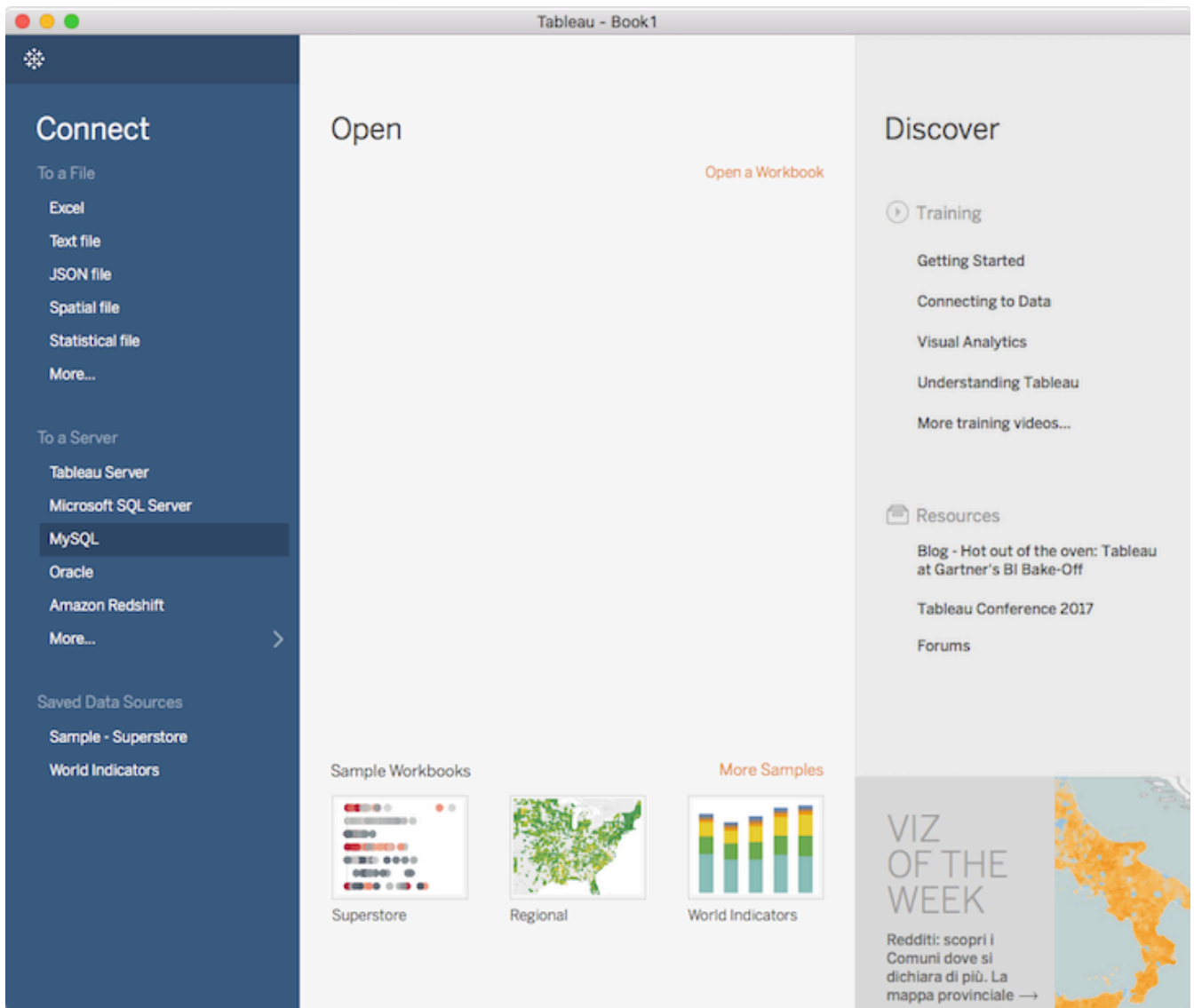
Tableau MySQL drivers

With Tableau Desktop 10.2, the MySQL database driver is included by default. With previous versions of Tableau Desktop (8.2 - 10.1), the drivers must be downloaded from Tableau and installed separately. The database drivers must be installed before a connection to a MySQL database can be made.

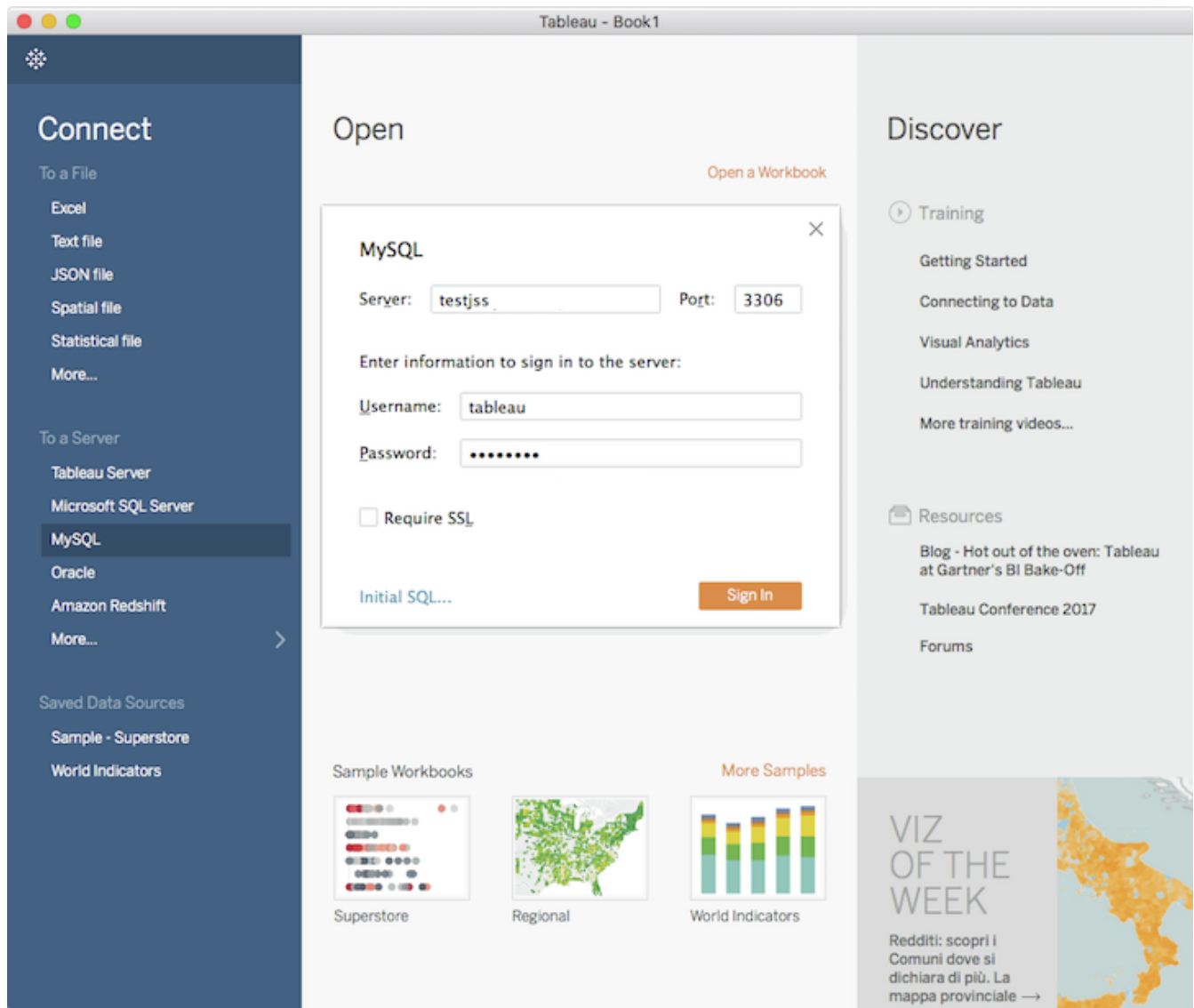
[Tableau MySQL drivers.](#)

▼ [Add MySQL data connection](#)

Open Tableau Desktop and select **'MySQL'** from the list of available data connections.



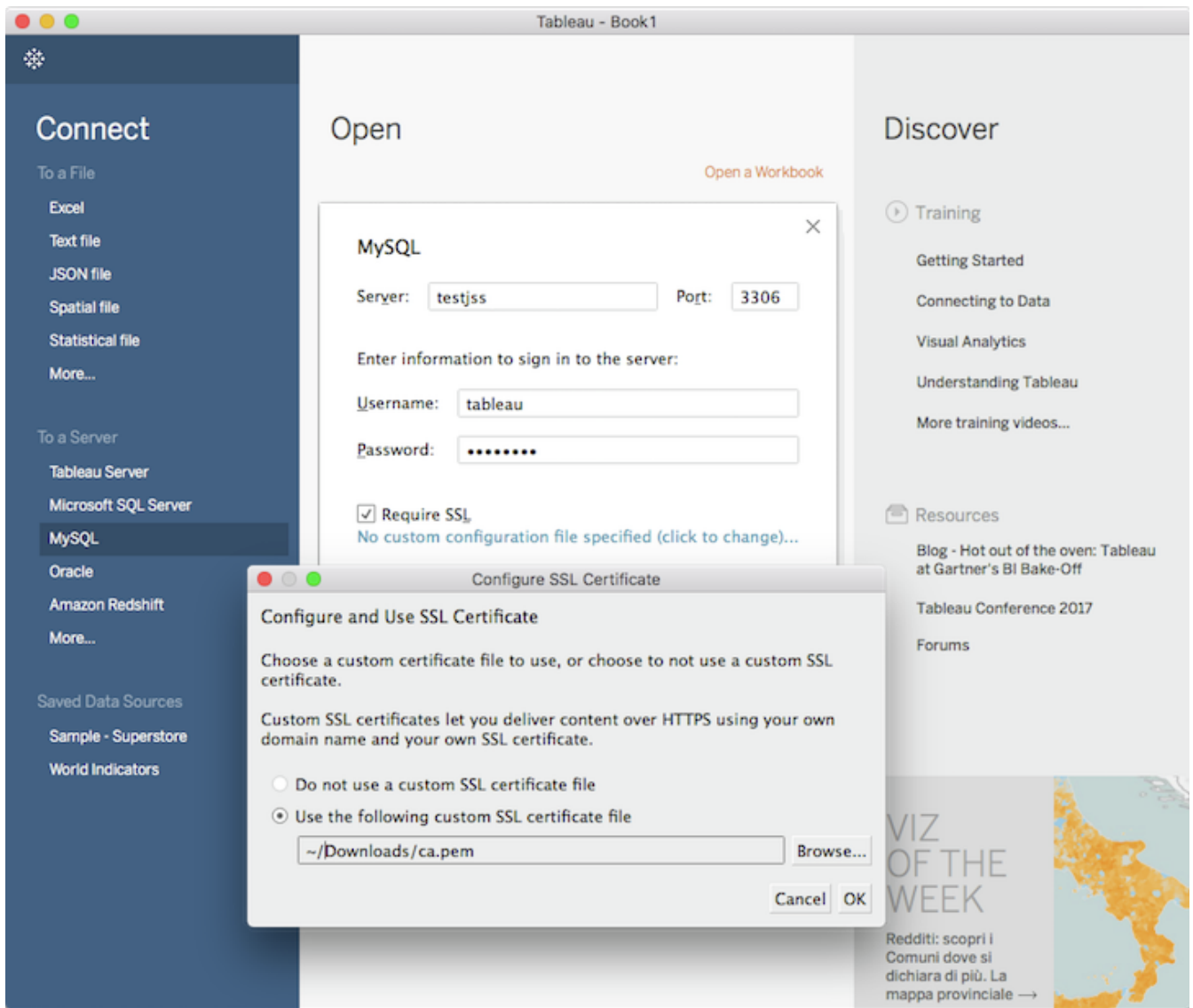
Connect to the Jamf Pro MySQL server using a valid username and password.



If an SSL cert has been set up for this client, use it when connecting to Jamf Pro MySQL server.

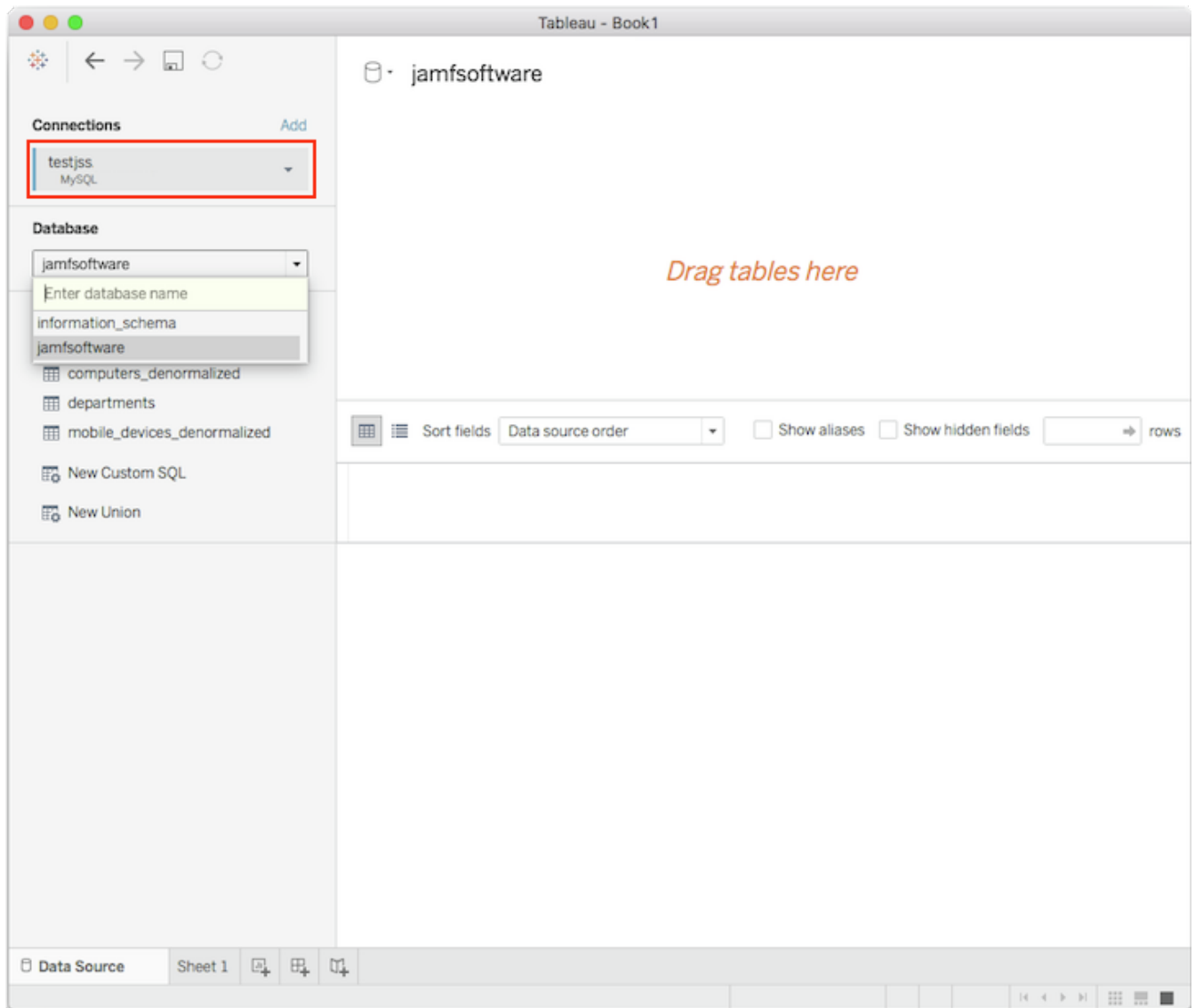
Select the '**Require SSL**' checkbox in the connection dialog. Then click on the '**.. (click to change)...**' link.

Select the 'Use the following custom SSL certificate file' option and browse to the location of the certificate authority cert generated earlier (ca.pem). Accept all the changes.

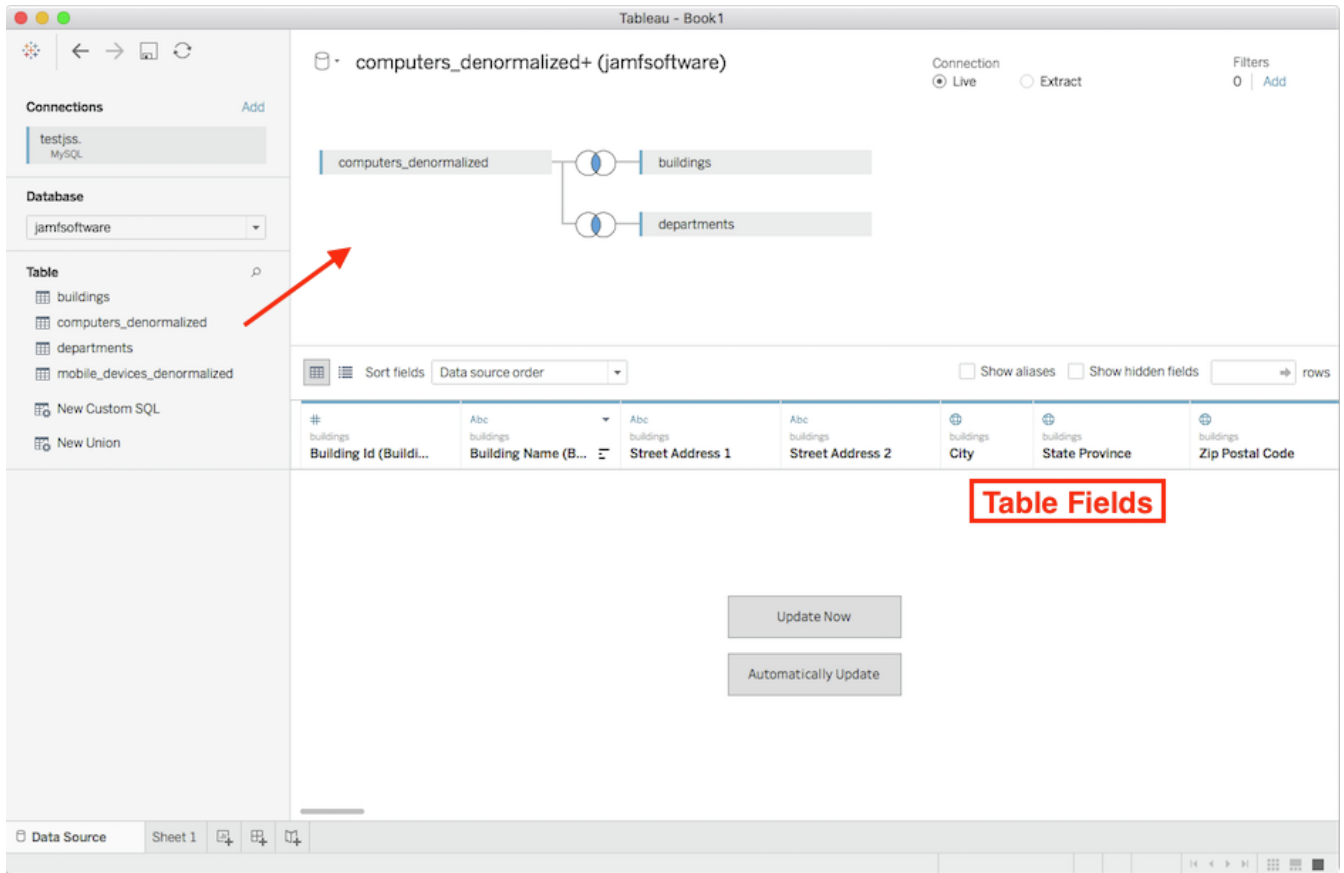


▼ Create data sources

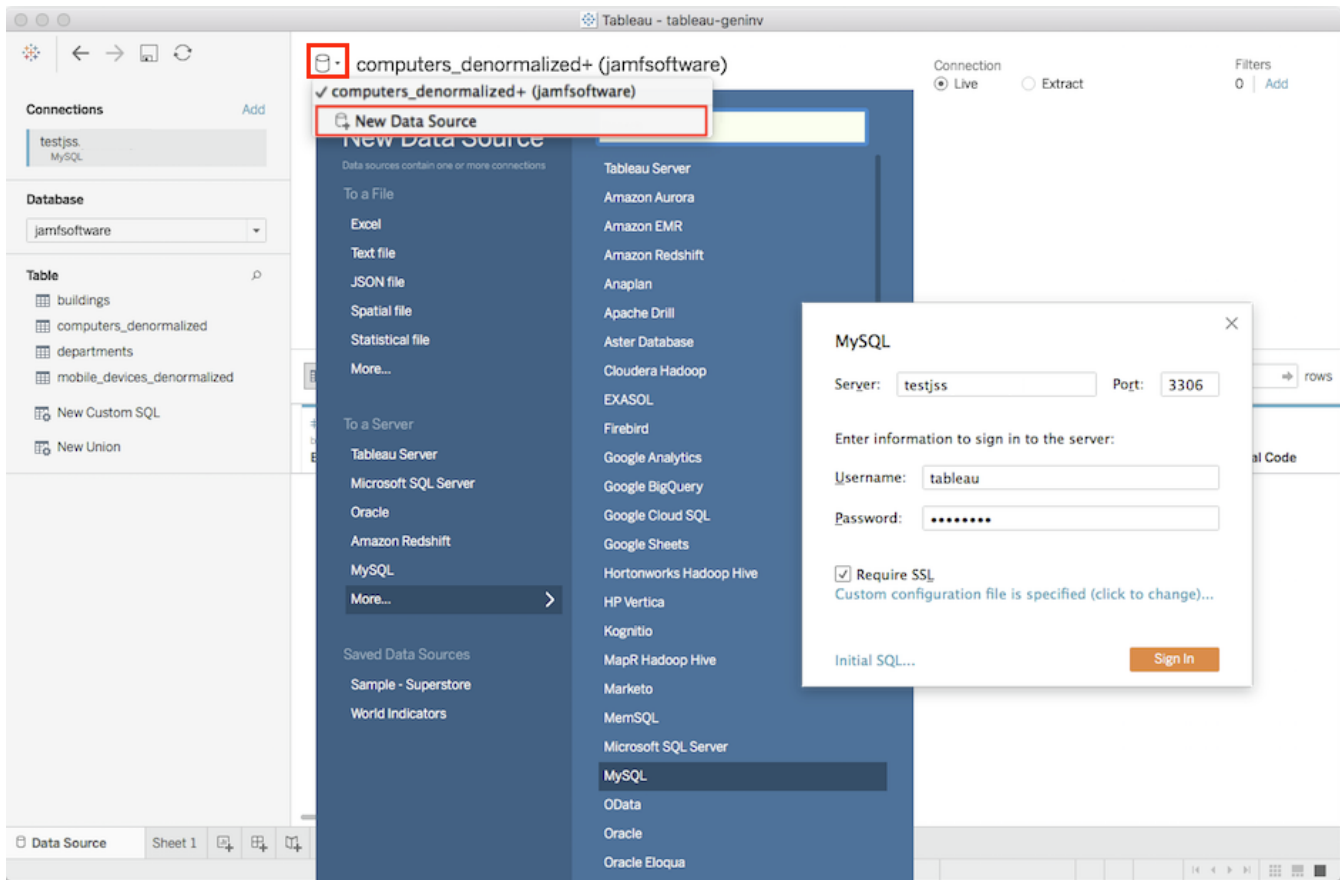
Once Tableau successfully connects to a Jamf Pro MySQL server, select the database to use. In this case, the database should be 'jamfsoftware'.



Drag tables from the column on the left to the data source area in the top panel. First drag the '**computers_denormalized**' table, then the '**buildings**' and '**departments**' tables. Tableau will automatically create joins between the tables. This creates a data source for the computer general inventory visualization. All the fields from the joined tables are shown in the panel below.



A data source for the mobile devices general inventory visualization also needs to be created. Select the database icon at the top of the screen. From the dropdown, select 'New Data Source' to create a new data source. Setting up the new data source is very much like setting up the first one. We need to connect to MySQL with a username, password, and SSL cert.



Once the new connection is established, set up the new data source for mobile devices like the first one. First drag 'mobile_devices_denormalized' table to the upper panel, then the 'buildings' and 'departments' tables.

Tableau - tableau-geninv

mobile_devices_denormalized+ (jamfsoftware)

Connection: ☒ Live ☐ Extract Filters: 0 | [Add](#)

Connections: [Add](#)

testjss
MySQL

Database: jamfsoftware

Table:

- buildings
- computers_denormalized
- departments
- mobile_devices_denormalized
- New Custom SQL
- New Union

mobile_devices_denormalized buildings departments

Sort fields: Data source order ☐ Show aliases ☐ Show hidden fields rows

#	buildings	buildings	buildings	buildings	buildings	buildings
	Building Id (Build...	Building Name (B...	Street Address 1	Street Address 2	City	State Province

Update Now

Automatically Update

Data Source Sheet 1

Now two data sources are available for visualizations. Clicking the '**Update Now**' will populate the fields in the lower panel with values.

Tableau - tableau-geninv

Connections: testjss, MySQL

Database: jamfsoftware

Table: buildings, computers_denormalized, departments, mobile_devices_denormalized, New Custom SQL, New Union

mobile_devices_denormalized+ (jamfsoftware)

computers_denormalized+ (jamfsoftware)

✓ mobile_devices_denormalized+ (jamfsoftware)

New Data Source

mobile_devices_denormalized

buildings

departments

Sort fields: Data source order

Show aliases Show hidden fields rows

#	buildings	buildings	buildings	buildings	buildings	buildings	buildings
	Building Id (Build...	Building Name (B...	Street Address 1	Street Address 2	City	State Province	Zip Postal Code

Update Now

Automatically Update

Data Source Sheet 1

Tableau - tableau-geninv

Connections: testjss, MySQL

Database: jamfsoftware

Table: buildings, computers_denormalized, departments, mobile_devices_denormalized, New Custom SQL, New Union

computers_denormalized+ (jamfsoftware)

computers_denormalized

buildings

departments

Sort fields: Data source order

Show aliases Show hidden fields 1,000 rows

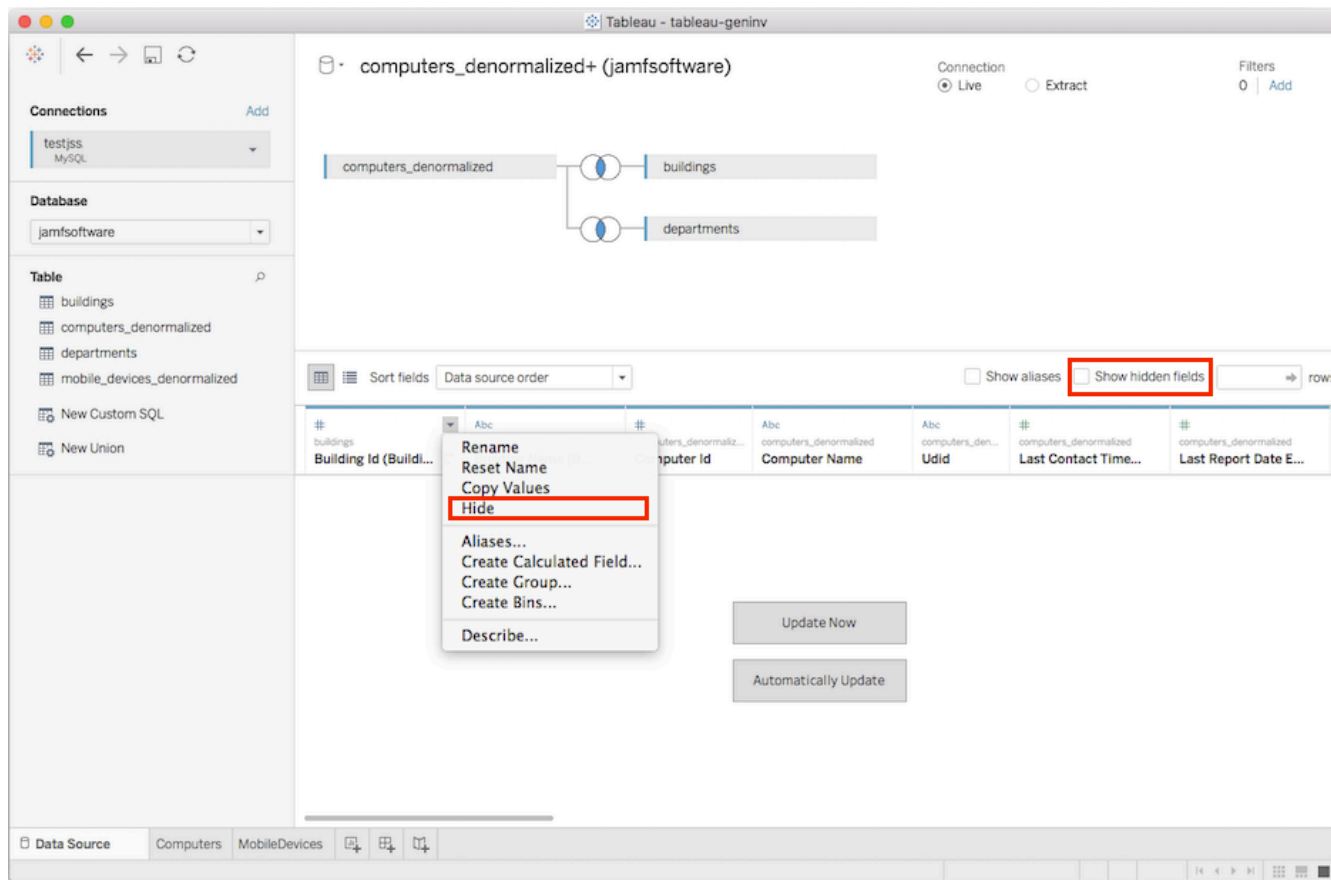
Abc computers_denormalized	Abc computers_denormalized	Abc computers_denormalized	Abc computers_denormalized	Abc computers_denormalized	# computers_denormalized	# computers_denormalized
Model Identifier	Mac Address	Alt Mac Address	Operating System ...	Operating System ...	Warranty Date Ep...	Last Patch Up
	DE:EB:7D:2F:E0:42	02:A7:2E:AB:8B:14	MacOs	1.0	1,424,534,563,000	
	54:B0:C1:C9:58:06	5E:80:07:F1:BA:C9	MacOs	1.0	1,440,259,363,000	
iMac11,1	3C:C6:C3:BF:6F:4A	08:72:24:3D:45:95	OSX	10.10	1,479,139,363,000	
MacBookPro10,1	B8:9C:6E:24:84:47	7E:47:03:88:DC:90	OSX	10.11	1,464,969,763,000	
iMac11,1	68:E6:C9:83:4A:5F	84:F7:1F:B4:D0:3C	OSX	10.10	1,431,360,163,000	
MacBookPro10,1	B4:11:A1:B1:DA:B7	AA:D8:67:BD:23:A8	macOS	10.12	1,433,779,363,000	
iMac11,1	1A:3B:41:FS:31:2B	70:59:3C:75:EF:B5	OSX	10.10	1,453,651,363,000	
MacBookPro13,3	D6:C9:09:AE:13:86	FE:FC:F2:54:CD:F5	OSX	10.11	1,432,137,763,000	
MacPro6,2	90:CB:A7:19:6B:75	86:98:00:41:2B:9D	macOS	10.12	1,406,563,363,000	
MacPro6,2	20:5F:44:26:E1:73	9C:CE:86:E1:4F:58	OSX	10.11	1,464,883,363,000	
MacBookPro13,3	DA:22:70:F1:47:49	EC:B4:5F:04:EF:F2	macOS	10.12	1,461,081,763,000	

Data Source Sheet 1

▼ Hide unused database fields (optional)

When building a data source **all** the fields from the primary table and the joined fields are included. This can make for a very long list of fields to choose from when creating a visualization. Luckily, fields can be hidden from view.

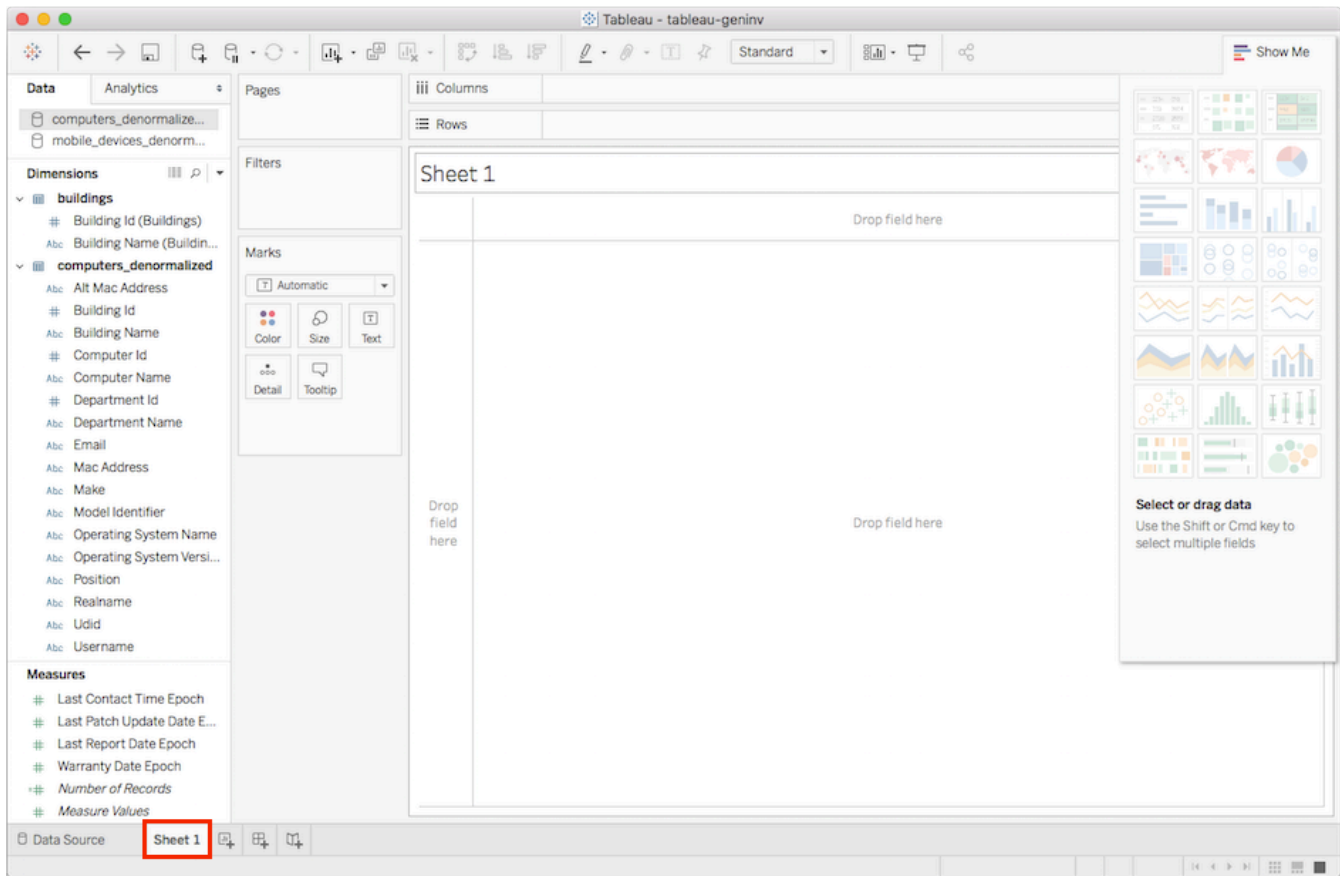
If a field is accidentally hidden, toggle the 'Show hidden fields' checkbox. Hidden fields will be show greyed out and can be toggled back to 'shown'.



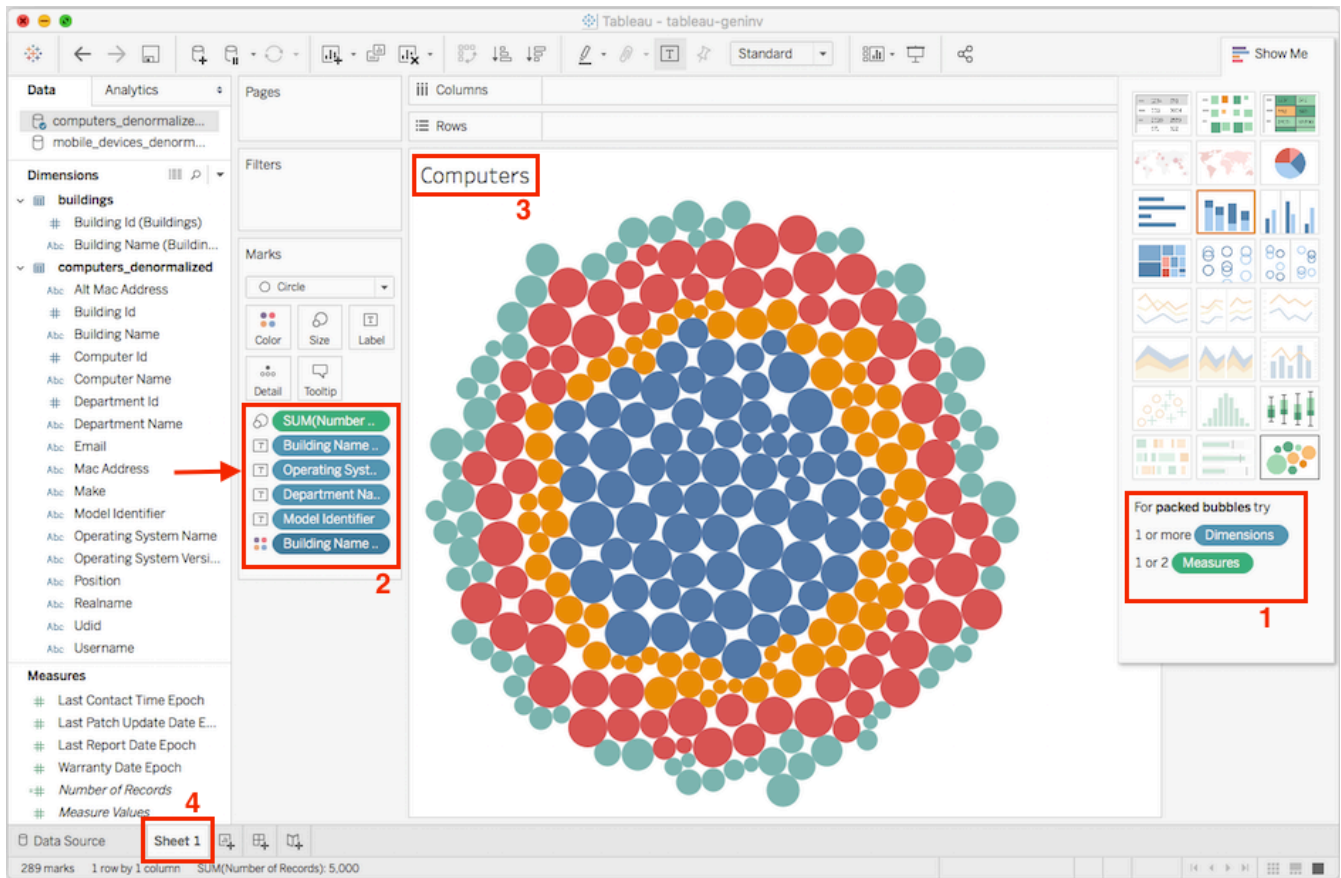
✓ Create worksheets

Worksheets are where data is combined in various ways to create visualization of that data. Here we are going to create a basic general inventory visualization for all computers and mobile devices in the Jamf Pro.

Select 'Sheet 1' from the bottom of the screen.

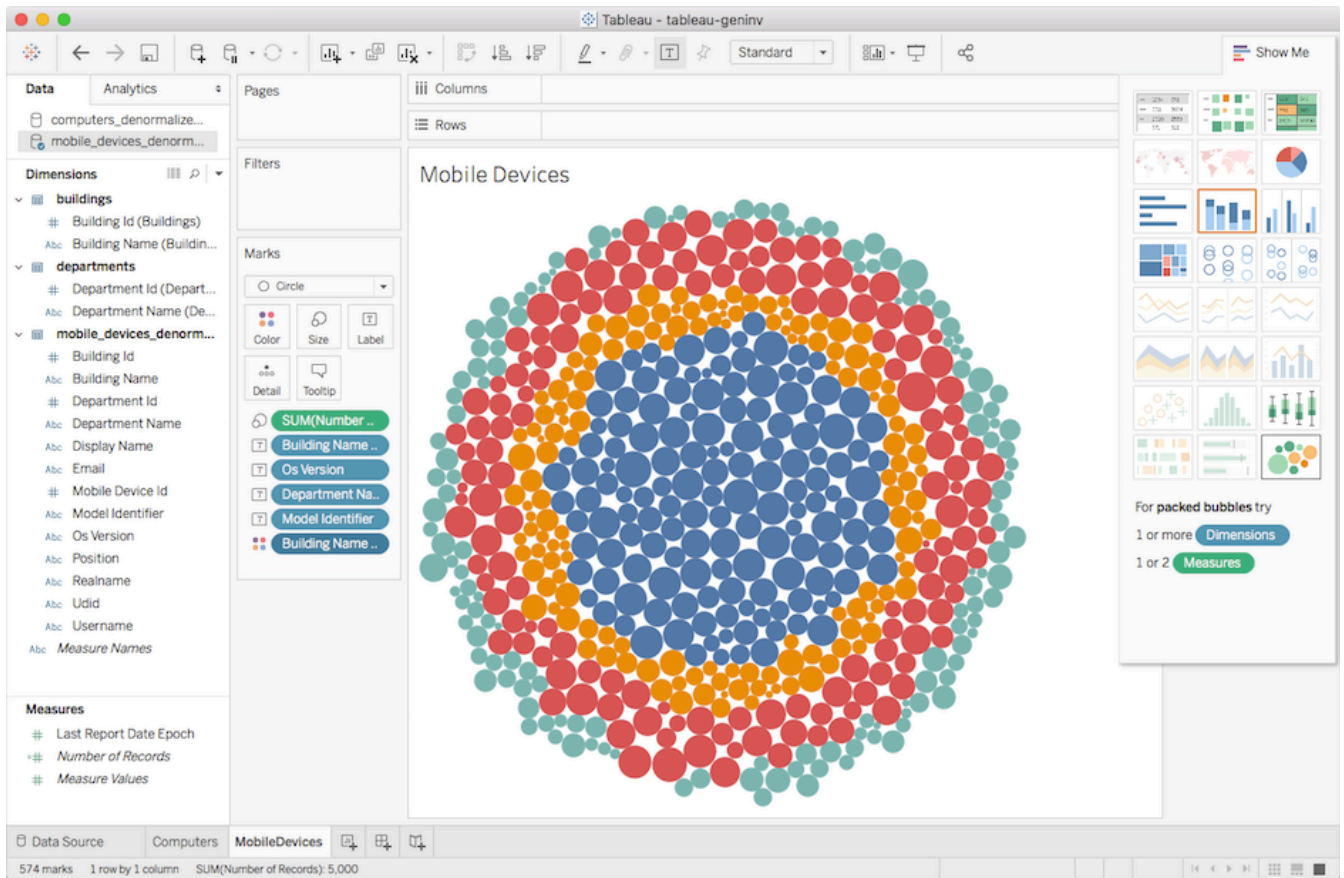


A quick rundown of how visualizations are created in Tableau.

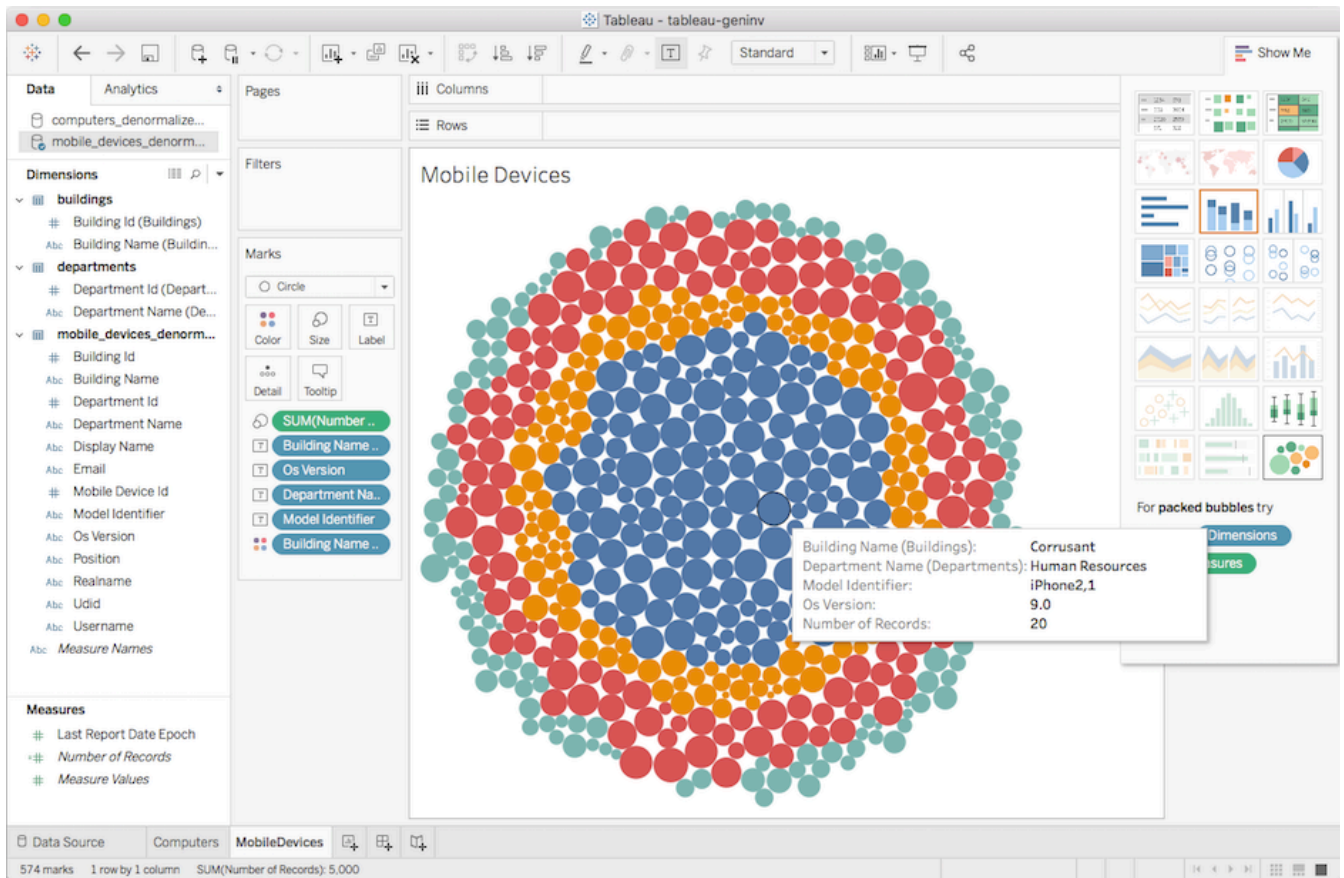


1. Each graph or visualization requires certain kinds of associations or data. For this example we're going to use the 'packed bubbles' visualization. This requires one or two '**Measures**' and 1 or more '**Dimensions**'. These are both listed in the left most panel.
2. '**Measures**' and '**Dimensions**' are dragged from the left most panel to display areas on the main panel. These are '**Row**', '**Columns**', or '**Marks**' depending on the type of graph or visualization. The 'packed bubbles' visualization uses all Marks. Drag the 'Number of Records' measure to the Marks box and right click it to change it's cast to 'size'. This will determine the size of the bubbles. Then drag the database table fields 'Building Name', 'Operating System', 'Department Name', and 'Model Identifier' over. Cast them as 'label'. When hovering over a bubble, these values will be displayed. Lastly, drag the database table field 'Building Name' over and cast it as 'color'. This alters the color of the bubble depending on the value of 'Building Name'.
3. The display are name can be renamed to 'Computers'.
4. The sheet name can also be renamed to 'Computers'.

Repeat these steps for Mobile Devices. Some of the field names are slightly different, but the field value mapping done earlier can help with determining what should go where.



Hovering over a bubble shows the text field values.



Save and re-open workbook

Save the workbook with the two general inventory visualizations worksheet.

Tableau does not maintain a persistent connection with a database or other data source. Nor does it store data locally. Each time a workbook is re-opened it must pull new data from it's data sources. For a database data connection, this mean re-authenticating with the database each time a workbook is opened.

