

CASA0006/0009 Workshop 1 - Introduction to Databases

Authors: Steve Gray & Huanfa Chen

Updated: 2022/01/22

Welcome to workshop 1 of CASA0009 and CASA0006. Today we're going to learn how to connect to a database, how to create your first tables, import data into a table and then finally get some data back out of the database.

We will explore a database client that you'll use over the next few weeks of the course, MySQL Workbench, which is a free download if you want to use, the client on your personal computer or laptop.

Note that the user interface on your machine may slightly differ from the following snapshots (which are from the workbench 8.0 CE on Windows 10).

Installing MySQL client

You can download the client from <http://dev.mysql.com/downloads/workbench/>. If you are working on UCL cluster machines or Desktop@UCL, we have already set up the workbench so you don't need to download the software.

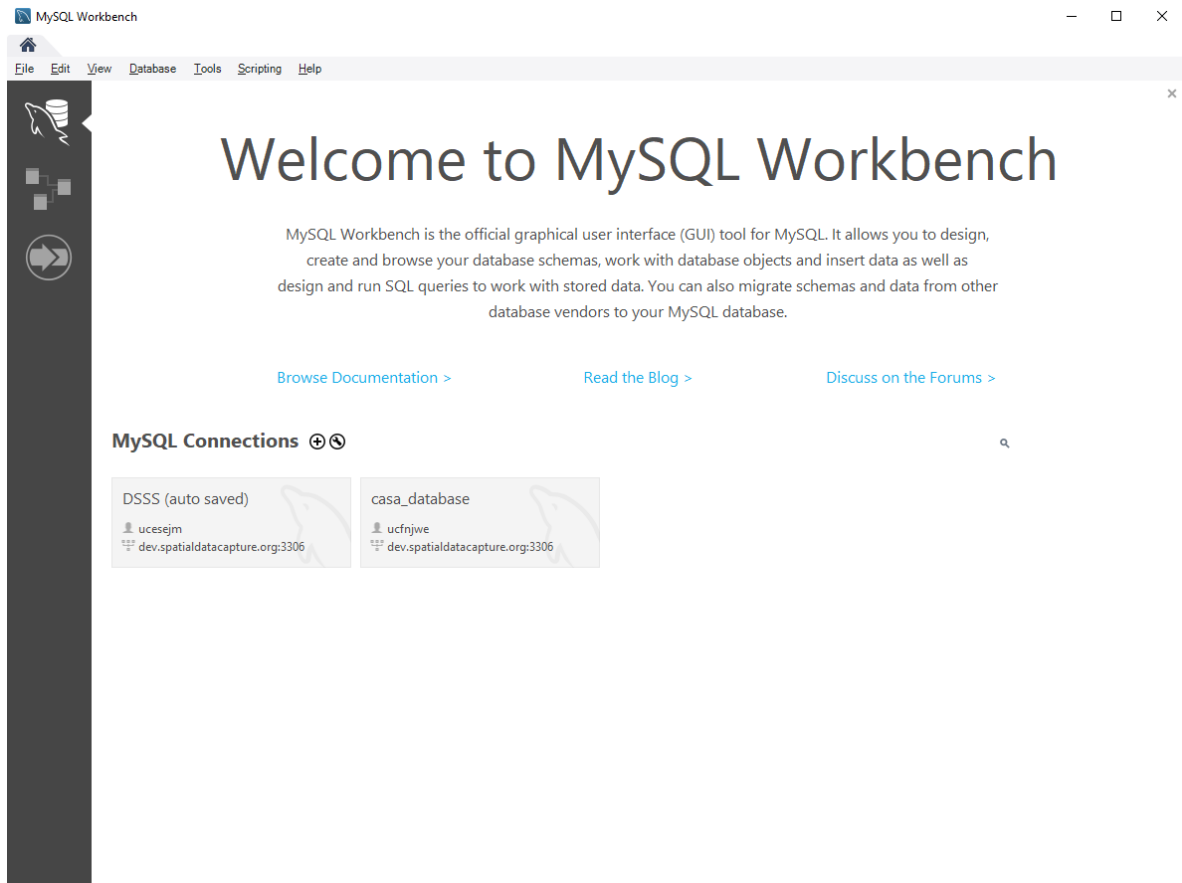
The latest version of MySQL workbench is 8.0.23. Note that this version may not work on some OS machines. If so, you can download an older version (e.g. 5.2.43) from <https://downloads.mysql.com/archives/workbench/>. The older versions also work for this module.

When you are installing the MySQL workbench, follow the default settings.

If you are outside the UCL network, there are two ways that you can connect to this MySQL server:

- Using the UCL VPN: <https://www.ucl.ac.uk/isd/services/get-connected/ucl-virtual-private-net-work-vpn>
- Using the Desktop@UCL Anywhere: <https://www.ucl.ac.uk/isd/services/computers/remote-access/desktopucl-anywhere>

Using MySQL workbench to log into database server

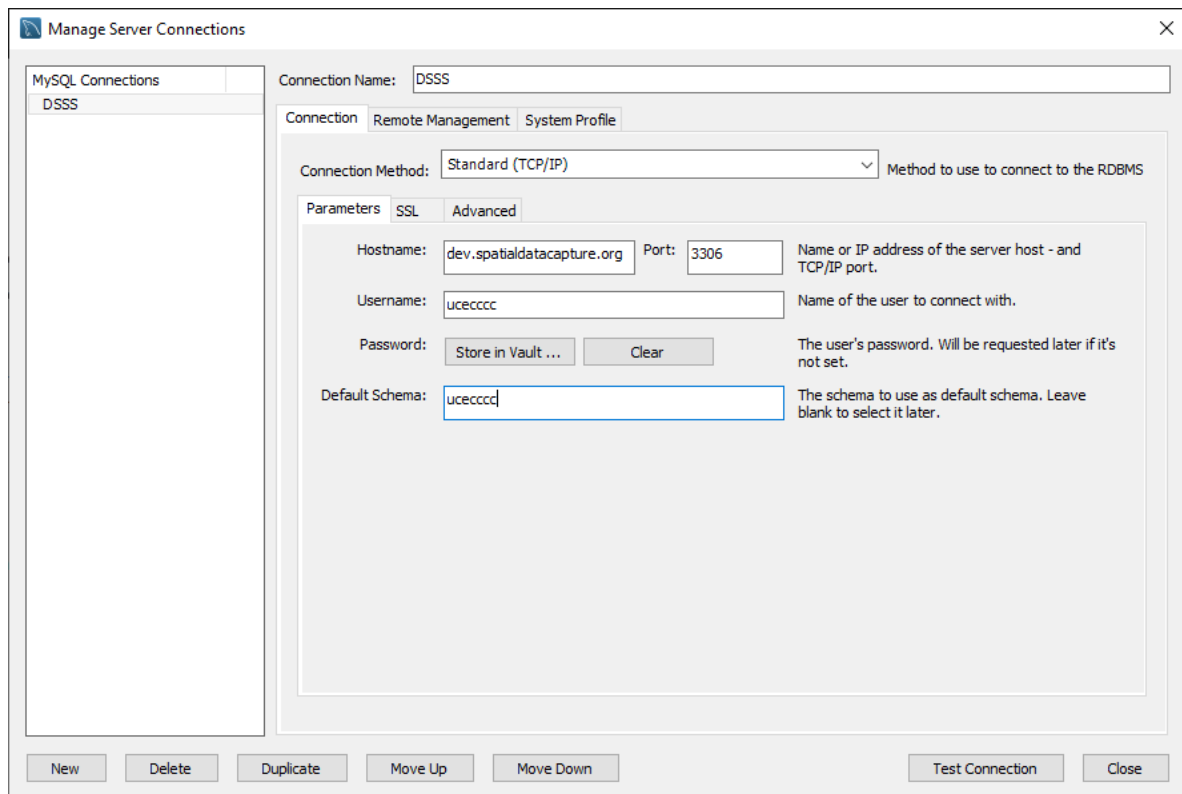


MySQL Workbench allows you to connect to any MySQL database that is running on a server. To do this you must first set up your connection to the machine. As mentioned in the lecture, you need to know the following in order to connect to the server:

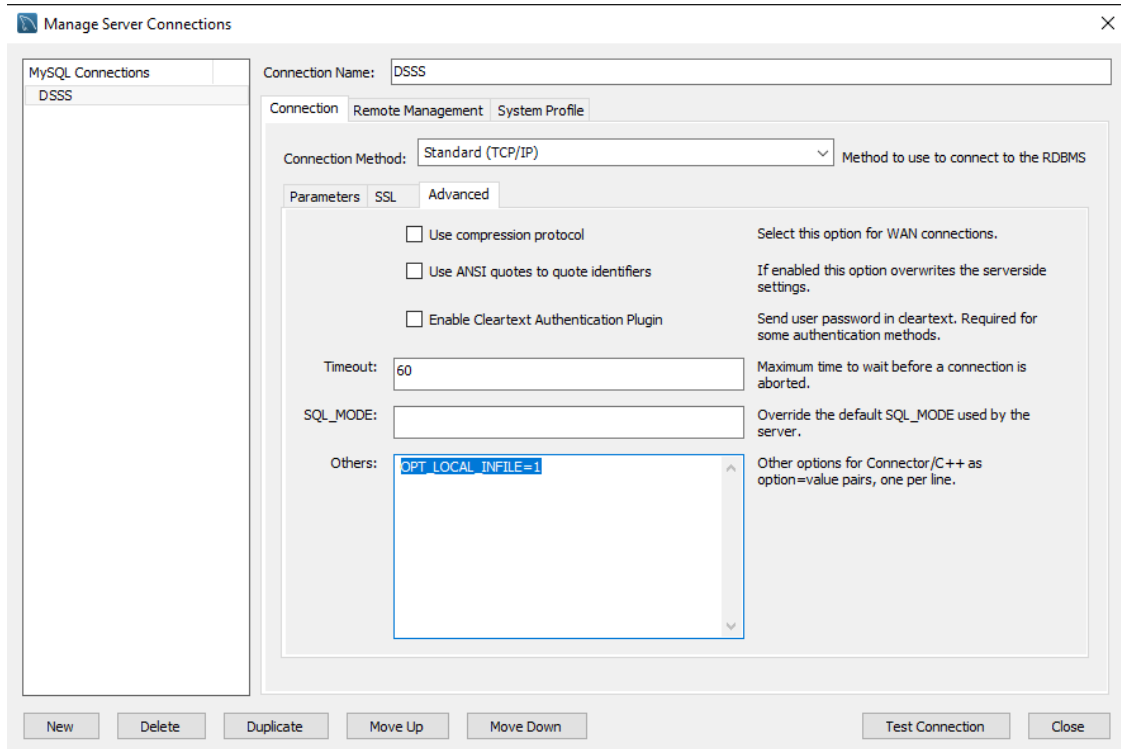
- The hostname (or IP Address) of the server and the port. This tells where the server is located.
- A username. The username and password are unique for everyone - keep it confidential.
- A password

Please follow the steps to create and store a connection to the server:

1. Click on the 'Database' on the menu, and then click on 'Manage Connections...'. You will see a window of 'Manage Server Connections'.
2. Click the button 'New' on the bottom left. Give your connection a name like 'MySQL_SDC'. Enter the hostname, username, password, and default schema (or database). These are on your pass card which you received at the start of the workshop. Here, the hostname is **dev.spatialdatacapture.org**, and the port is **3306**.



3. Change the setting of importing local files. Click the tab of 'Advanced'. In the text box of 'Others', enter 'OPT_LOCAL_INFILE=1'. This enables loading a local file to the database.



4. Testing your connection and make sure it succeeds.
5. Click 'Close' to exit. The Workbench will automatically save this connection.

Then, connect to the saved database

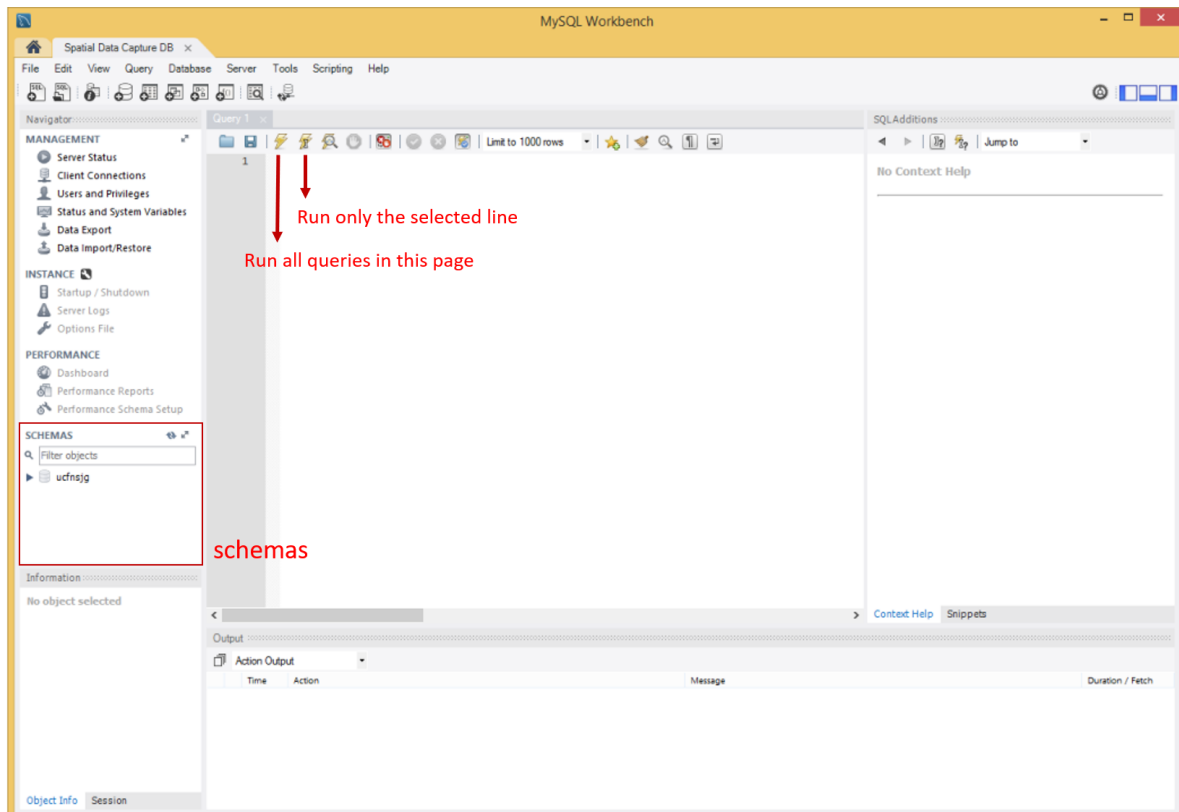
1. Click on the 'Database' on the menu, and then click on 'Connect to Database...'. You will see a window of 'Manage Server Connections'.
2. In the new window and in the list of 'Stored Connections', choose the one that you just created. Click 'OK'.

You can also use the shortcuts on the front page of MySQL workbench.

Creating your first table

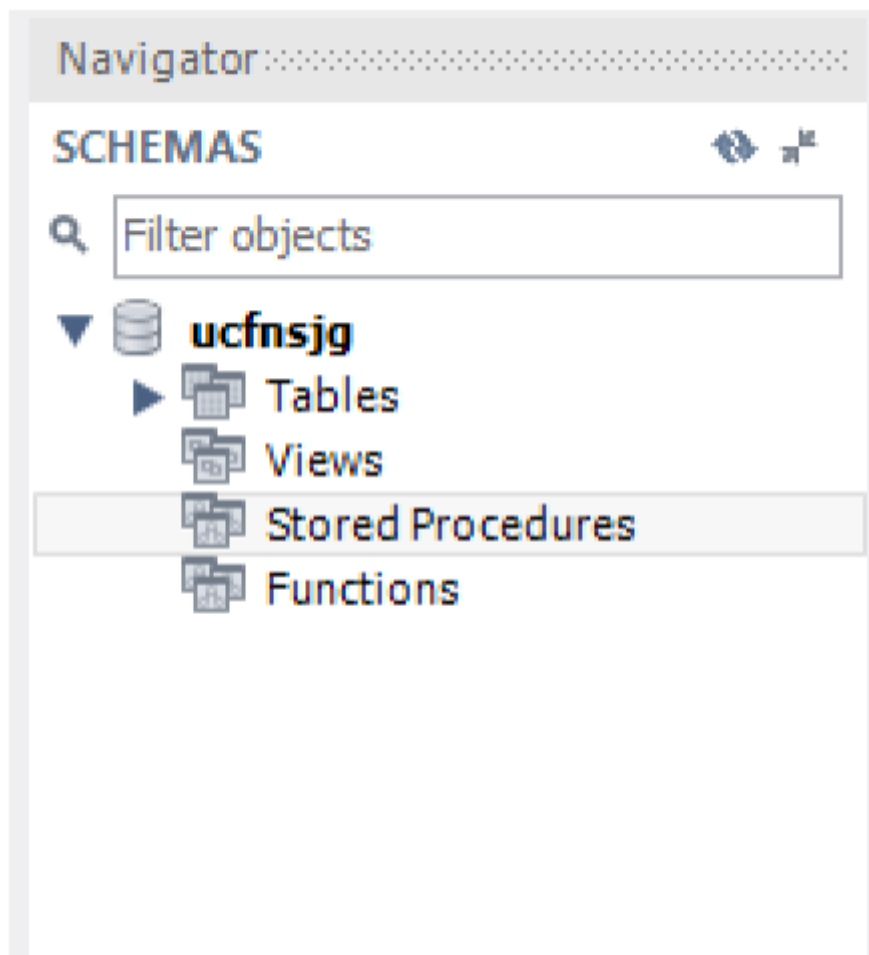
A database is useless without any data stored within it. Today we'll create 3 tables that have data from photos that users have uploaded to Flickr: the photo sharing website. In this task you'll learn how to create some tables to store data within. Remember a table is a unique location to store data within, like a book sitting on a shelf of a library.

When you have connected to the database server you'll be presented with the following screen. This is the heart of the database client. Have a look around and play with the interface to make yourself familiar with the client.



The main panel in the middle is where we enter our SQL queries to get our data from the database. The right panel shows contextual information to help us build our queries. Finally, the bottom panel is where our results appear when we execute a query on the database. There are two lightning buttons for running SQL queries, with one running all queries in the page and the other running only the selected line.

In the bottom left corner, a submenu with the title "Schemas". **Schema is just a fancy name for a database.** This is where your database is stored and you should be able to see your username next to the standard database symbol. Click the arrow to open up your database.



In this workbench, there are two ways to create tables: with the interface or with a query.

1. Creating a table with the interface

1. Inside your database – right click on Tables and select Create Table
2. Give your table a title: photos
3. Under column name double click and enter the data for each of columns (Make sure to change the Data type, otherwise creating the table won't work). Here, 'PK' and 'NN' mean 'Primary Key' and 'Not Null', respectively. They are constraints to this column.

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
pid	BIGINT(11)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
date_uploaded	BIGINT(11)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
download_url	VARCHAR(255)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Column Name:
 Collation:
 Comments:

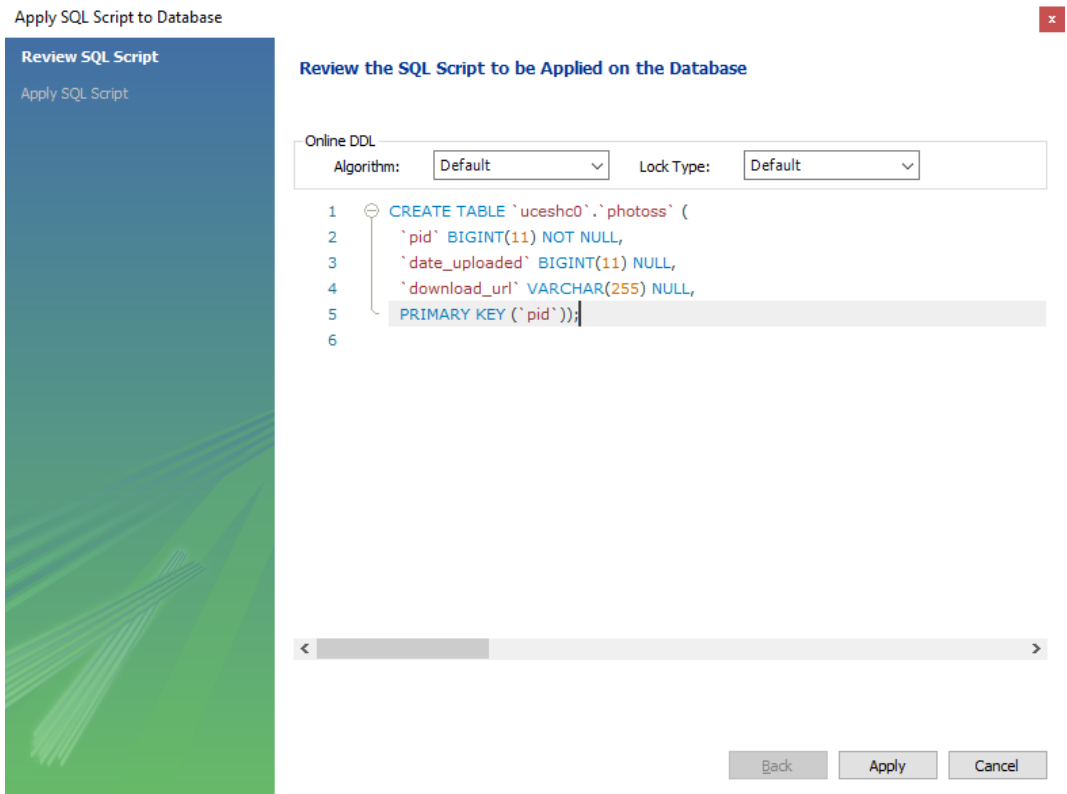
Data Type:
 Default:

☒ Primary Key
 ☒ Not Null
 ☐ Unique

☐ Binary
 ☐ Unsigned
 ☐ Zero Fill

☐ Auto Increment

4. Click Apply to create the table. The software will turn the table specifications in to SQL query, and then run the query. If the query is correct, click 'Apply' to continue.



2. Creating a table with a query

When you create a table with the user interface, it's creating the following SQL query and executing it, creating the table. To create the same photos table you can enter the following query in the Query browser and click the lightening bolt (*which lightening bolt should be used?*).

```
CREATE TABLE `photo` (  
  `pid` bigint(11) NOT NULL,  
  `date_uploaded` bigint(11) DEFAULT NULL,  
  `download_url` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`pid`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Some tips are:

- Always end a SQL query with ';'. Otherwise, you will see an error from MySQL.
- There is difference between ` (called backtick or backquote) and ' (apostrophe or single quote). If you use single quotes above, it won't work. In SQL, backticks are used to indicate database, table, and column names. If you are sure that the names are not reserved or used, you can write a query without using backticks (why not try removing the backticks in the above query?). Quotes (' or ") are used to delimit strings, and differentiate them from column names. Strings always need quotes.

3. Reflecting on the data types

In the query above, we have used two different data types, namely **bigint** and **varchar**. **bigint** is one of the integer types and is an eight-byte signed integer. More details about integer in MySQL are [here](#) and [here](#). The **20** in BIGINT(11) is simply a hint for the display width, and has nothing to do with the storage or range of this value.

varchar is one of the string data types, and 255 in varchar(255) means a length that indicates the maximum number of characters. **varchar(255)** can hold up to 255 characters.

Task: Can you create the following two tables with these column names and types?

Table name: **metadata**

Column Name	Type
pid	bigint(11)
description	varchar(25)
device	varchar(25)
tags	varchar(25)

Table name: **photo_locations**

Column Name	Type
pid	bigint(11)
lat	varchar(25)
lon	varchar(25)
coords	point

Importing data into a table

We now have to import data into our table. Open up a browser and download the 3 data files for this workshop: http://dev.spatialdatacapture.org/data/workshop_1/data.zip

Unzip the data into your user folder and make a note where the files have been created. You'll need the full path to the file in a minute.

To import the data into your table we have to run a query to read the CSV file and send it to the database for processing.

In the Query Browser enter the following command:

```
LOAD DATA LOCAL INFILE 'path/to/folder/flickr_2010_uk_JanFeb_photos.csv' INTO
TABLE photos FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY '\n'
IGNORE 1 LINES;
```

This command can be broken down into several parts:

1. `INFILE 'path/to/folder/flickr_2010_uk_JanFeb_photos.csv'`: tell where the local file is;
2. `INTO TABLE photos`: what table to load the data into
3. `FIELDS TERMINATED BY ','`: the formatting of this local file. Its columns are separated by comma.
4. `ENCLOSED BY '"'`: the formatting of this local file. The input values are enclosed within quotation marks.
5. `LINES TERMINATED BY '\n'`: the formatting of this local file. Each line is terminated by newline.
6. `IGNORE 1 LINES`: the first line (i.e. the header) in the local file is ignored.

Reference: <https://dev.mysql.com/doc/refman/8.0/en/load-data.html>

When you're ready to run the command, click the lightening bolt to execute the query.

Loading the location data

The table 'photo_locations' contains a different type of data. The POINT data type is a special geometric type that allows the user to make spatial queries within the database. If we just tried to load the data straight from the CSV file then various errors would occur.

To fix this we need to convert the text of the POINT type data in the file and insert it in the table row by row. To do this we call **LOAD DATA LOCAL** like this:

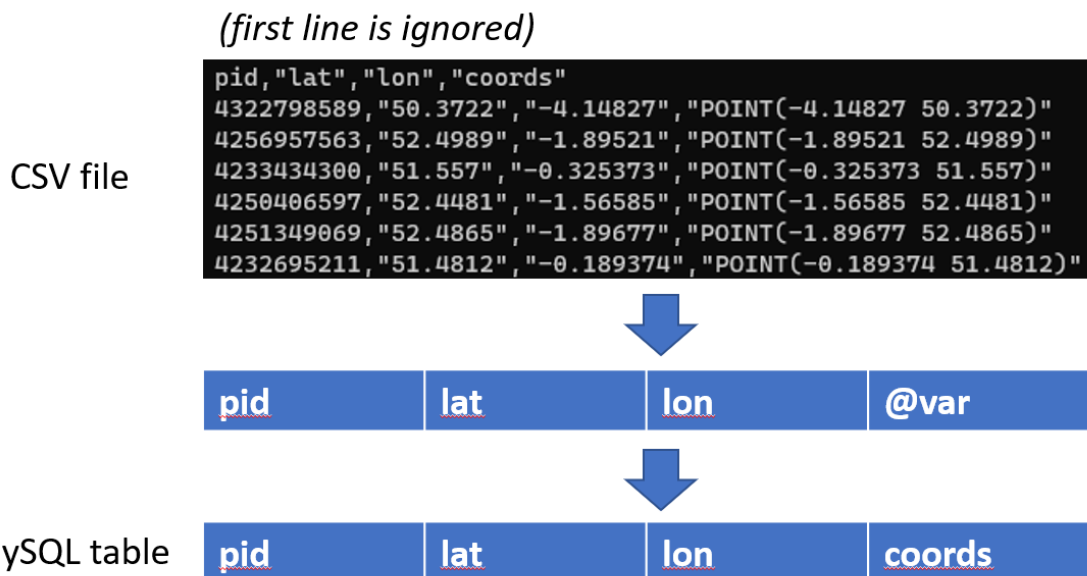
```
LOAD DATA LOCAL INFILE
'path/to/folder/flickr_2010_uk_JanFeb_photo_locations.csv' INTO TABLE
photo_locations FIELDS TERMINATED BY ',' ENCLOSED BY '"' LINES TERMINATED BY
'\n' IGNORE 1 LINES (pid, lat, lon, @var4) SET coords = ST_GeomFromText(@var4);
```

There is something new about this command. In LOAD DATA LOCAL, by default, all columns are loaded. When no column list is provided at the end of the statement, input lines are expected to contain a field for each table column.

If you want to load only some of a table's columns, specify a column list, such as (pid, lat, lon, @var4). This means that the first three columns in this local file will be imported as the columns of pid, lat, and lon into this table, respectively. Then, the four column of the local file is called @var, and then @var4 is transformed into **coords** by ST_GeomFromText function and stored as coords in the MySQL table. Note that coords is of the **point** type of spatial data.

This query is illustrated as below.

Reference: <https://dev.mysql.com/doc/refman/8.0/en/load-data.html>



Querying data in a database

Now you have all the data imported into your database we can start looking at the data. To look at the data that you have imported into the photos table, run the following query:


```
SELECT * FROM photos;
```

This will select all the rows in the photos table. These are real photos online right now. Right click on a row to copy the URL of the photo and paste into a browser to look at the photo. Have a look around, they're over 37,000 photos in your database.

Repeat Task 3 and 4 to import the data into the other two tables and have a look around the data within the tables. Before leave you should have data in all 3 of your tables.

Hint: To retrieve all the data from the other tables, run the following commands:

```
SELECT * FROM photo_locations;  
SELECT * FROM metadata;
```

Some Extra Fun

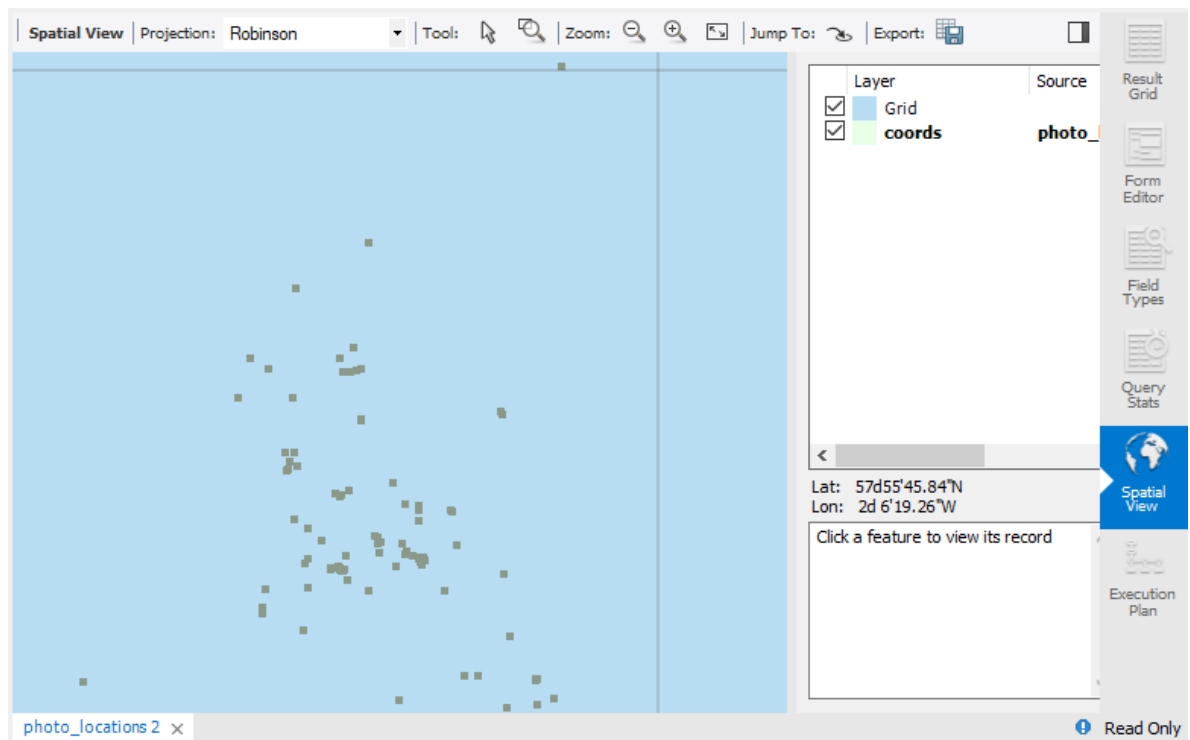
When you select all the data from photo_locations you'll see all the data contained in the table, but you'll also see that the coords column say BLOB for every value. Try viewing the data with the Spatial Viewer and see what it looks like. You may have to zoom out to see the full effect.

The reason the viewer says BLOB rather than displaying the POINT text in the data is that MySQL Workbench doesn't know how to display point data as text, as our last command forced the database to change the text into binary data via the **ST_GeomFromText** function!

Run this query `select * from photo_location`, and you will see this in the result grid window:

pid	lat	lon	coords
4288994900	51.4977	-0.174633	BLOB
4258151308	51.4601	-0.063879	BLOB
4294244938	50.6129	-2.13315	BLOB
4320526912	52.4497	-1.82047	BLOB
4381787818	55.862	-4.23496	BLOB
4379375737	55.862	-4.23496	BLOB
4388944367	55.862	-4.23496	BLOB
4386670351	55.862	-4.23496	BLOB
4386679783	55.862	-4.23496	BLOB
4388962103	55.862	-4.23496	BLOB
4381775362	55.862	-4.23496	BLOB
4333740000	52.6787	0.947699	BLOB
4333754914	52.6787	0.947699	BLOB
4235547772	51.5487	-0.061261	BLOB
4234769879	51.5487	-0.061261	BLOB
4234785779	51.5487	-0.061261	BLOB
4235550734	51.5487	-0.061261	BLOB
4253415817	51.6643	-0.364694	BLOB
4277934005	51.519	-0.1265	BLOB
4278684178	51.519	-0.1265	BLOB
4334807867	52.9046	-2.03958	BLOB
4316726701	51.8158	-0.203837	BLOB
4387819167	50.3672	-4.14157	BLOB
4362554571	52.5294	-1.89085	BLOB
4307555570	51.0772	1.50787	BLOB

If you click the option of 'Spatial View', you will be able to see the map of points:



The End

Well done! If you've completed all the tasks above then you're finished. Go and enjoy the rest of your day! You should now have a better understanding of databases and how they work. Next week you'll learn more about SQL and how to use powerful queries to interrogate the data further.

See you all next week.