TUTORIAL FOR IMPORTING OTTAWA FIRE HYDRANT PARKING VIOLATION DATA INTO MYSQL

We have spent the first part of the course learning Excel: importing files, cleaning, sorting, filtering, pivot tables and exporting filtered datasets into new files.

Now it's time to graduate to MySQL, which has the advantage of being able to work with multiple tables with millions of records.

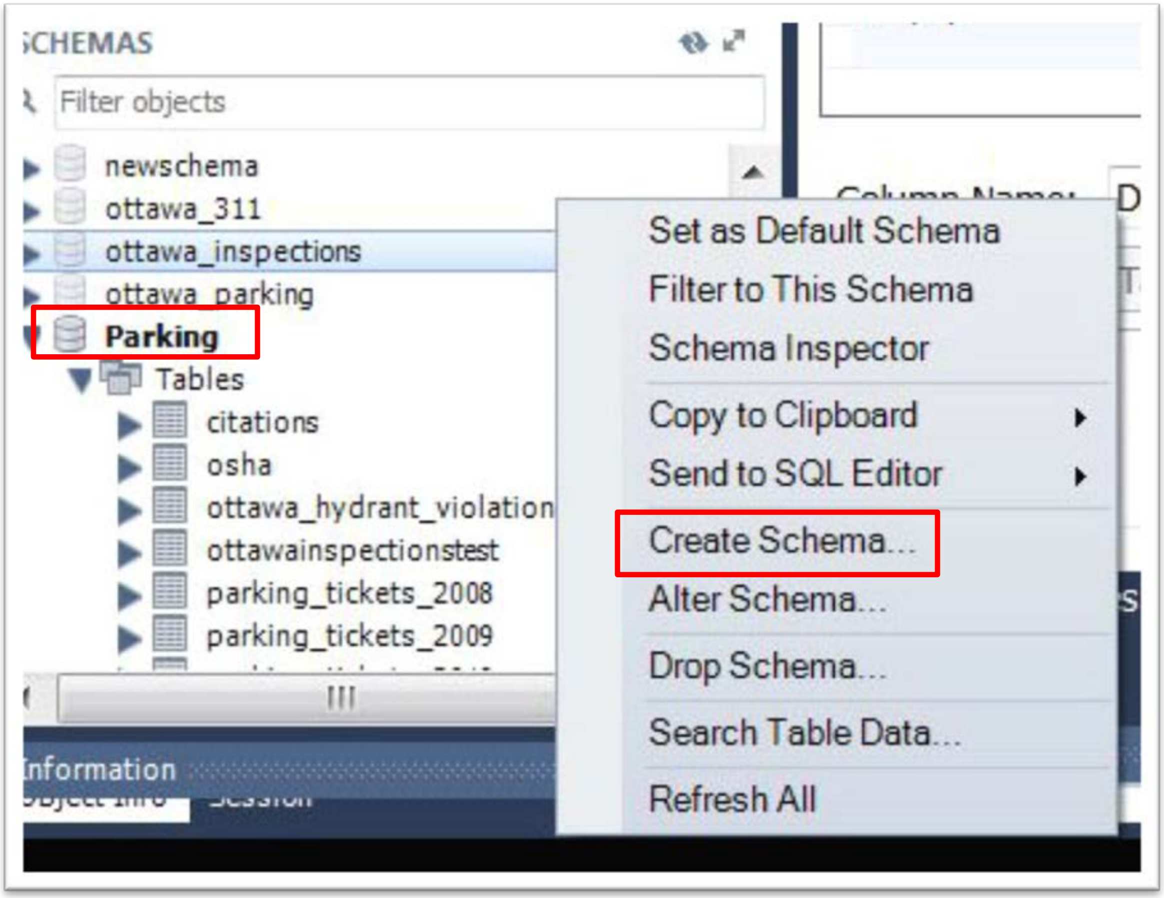
We have already successfully downloaded MySQL onto your computers by either using the tutorial for Windows or Macs, or MAMP pro for Windows.

Now, we will import the dataset that contains the 2008 violations for parking too close to fire hydrants that formed the basis of Steve Rennie's story. (NOTE: I have also uploaded the Excel workbook that contains worksheets for the years 2008-2014 year-to-date, as well as a worksheet that explains the categories. We will be using this workbook in the future. Please feel free to download this Excel workbook as well. It's located under the syllabus' Sept 30 section. However, for this tutorial, we are **ONLY** using the 2008 dataset only.)

So let's get started.

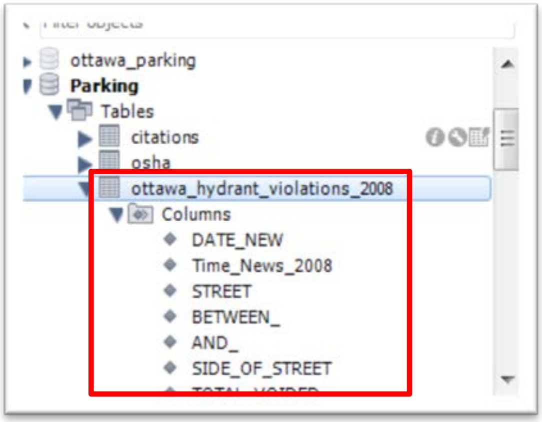
1. Download the city of Ottawa’s 2008 fire hydrant parking violation dataset by going to the Sept.30 link of our syllabus, and right-clicking on the “2008 Fire Hydrant Parking Data” link to save as a csv file.
2. Save the file to a folder set up for this exercise.
3. Make a copy and work from the backup.
4. You’ll notice that this is **ONLY** violations from 2008. We will be adding subsequent years to this dataset. This dataset represents a starting point.
5. Make sure that the backup version for this tutorial is on your C-drive.
6. Open the file. You’ll notice that the dates in the date columns are ordered year, month, day. This is the order the dates must be in. Otherwise, MySQL will import the dates as zeros.
7. Open MySQL Workbench.
8. We must create a “SCHEMA”.

9. Under the “SCHEMAS” section on the left



10. Select Create Schema, which is really a database that contains a collection of tables. Think of the tables as giant filing cabinets that contain separate files. Those files are the smaller, individual tables that will contain data you import. So in the example above, the Schema is “Parking”. The filing cabinet is “Tables”. The files are the tables. And if you click on the arrows next to each of those tables,

MySQL produces the individual columns.

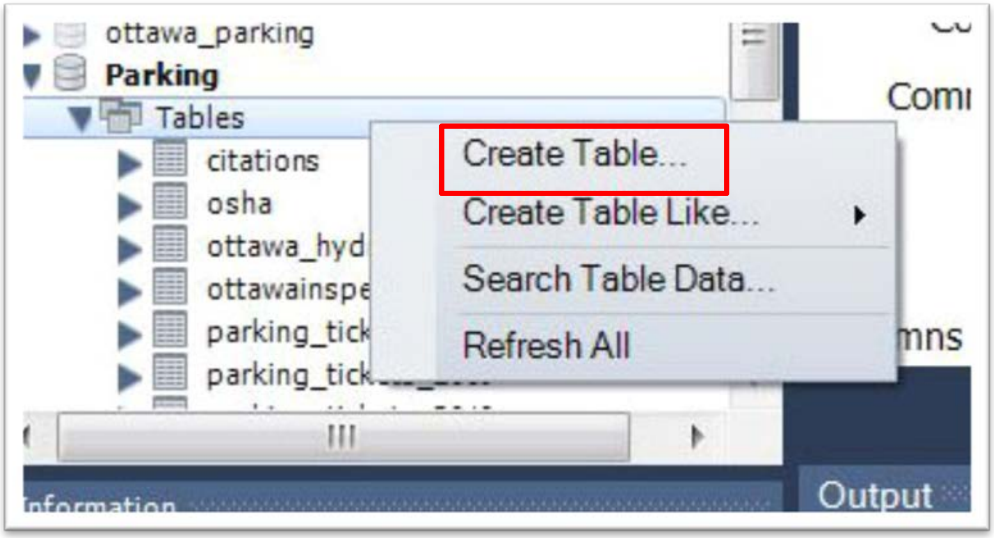


- ottawa_parking
- **Parking**
  - Tables
    - citations
    - osha
    - ottawa_hydrant_violations_2008
      - Columns
        - DATE_NEW
        - Time_News_2008
        - STREET
        - BETWEEN_
        - AND_
        - SIDE_OF_STREET

11. Unlike Excel, MySQL does not allow you to import files directly. This means that you have to create a table that you can see in the screenshot above that contains the columns.

12. Before you begin creating a table in MySQL, open your csv file if you haven’t done so already. Having it open allows you to count the characters in cells in columns with lots of text, and copy and paste the column names into MySQL. This is where it helps to have two screens. (NOTE: because this dataset is small – 5,466 records – we can open it in Excel. If the file had millions of records, we would have to open it in a text editor such as TextWrangler for Macs of Notepad++ for machines using windows.)

13. To begin creating the table, right-click on “Tables” to obtain a short-cut menu and chose Create Table.



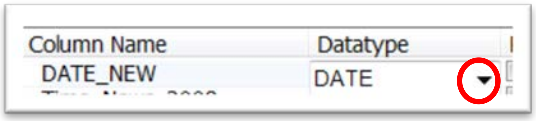
14. Once you have a table, give it a name in the “Table Name” section.

15. This is where we will create the table by adding the table name in the “Column Name” section

16. Next we must define the “Datatype” for each column we create. This is MySQL’s way of ensuring the data is properly formatted to match the data in your csv file that you have open. Pages 185-186 of Computer-Assisted Reporting explain the data types, except for one that we will use in this tutorial called “FLOAT”, which is a more flexible format for storing numbers. (NOTE: if in the future, you would like to make the process of importing data into MySQL easier, there is a product called NAVICAT, which bypasses some of the steps we are going to take. It’s a decent option, although one that comes with a modest

price tag. However, there is a lot to be said for learning how to do things manually, which leads to a better understanding of importing data into MySQL. Such an improved understanding also makes it easier to troubleshoot, which is always the frustrating part of working with data.)

17. Once you've names the column, you must choose a data type. You can obtain a drop-down menu for the various data types discussed in the textbook, by double-clicking on the cell in the "Datatype" column to obtain an arrow.



Column Name	Datatype
DATE_NEW	DATE

18. Clicking on the arrow, produces a menu which provides a list of options.

19. We've chosen "DATE" for the first column for obvious reasons. (NOTE: In order for MySQL to import date columns properly, the order must be year, month, day. Please see the "TECH TIP" note on page 189 of our textbook.)

20. As is explained in the textbook, the VARCHAR is a variable character field used to import columns that can hold letters or numbers. Clicking on the VARCHAR option from the drop-down menu produces the term, plus two empty brackets. Click inside the brackets to specify the number of characters you want. Because the length will

vary, you must choose the maximum length. So say the number of characters varied from 10 to 60. You would choose a VARCHAR of 60, or 65 just be on the safe side. This involves counting the number of characters in the column that contains the greatest numbers, which is easy because you have the csv file open!! This means doing a manual count. (NOTE: if the column has long text strings that make counting tricky, you could automate the process by creating a new column in Excel and using the LEN function ("=LEN(cell reference") which produces the character length for that cell, You could then copy the formula, sort in descending order to easily obtain the maximum and minimum values in the range. Once you have the range of character lengths, just delete the column.)

21. For the columns that contain the fine amounts, we will use the "FLOAT" option, which is a bit more flexible than double, a format that allows decimal places.
22. And you will also have to add an extra column at the end that will cut down on the "truncation" errors that occur. I've called that column "Empty". However, the exact name doesn't matter.

23. Your table should look like this:

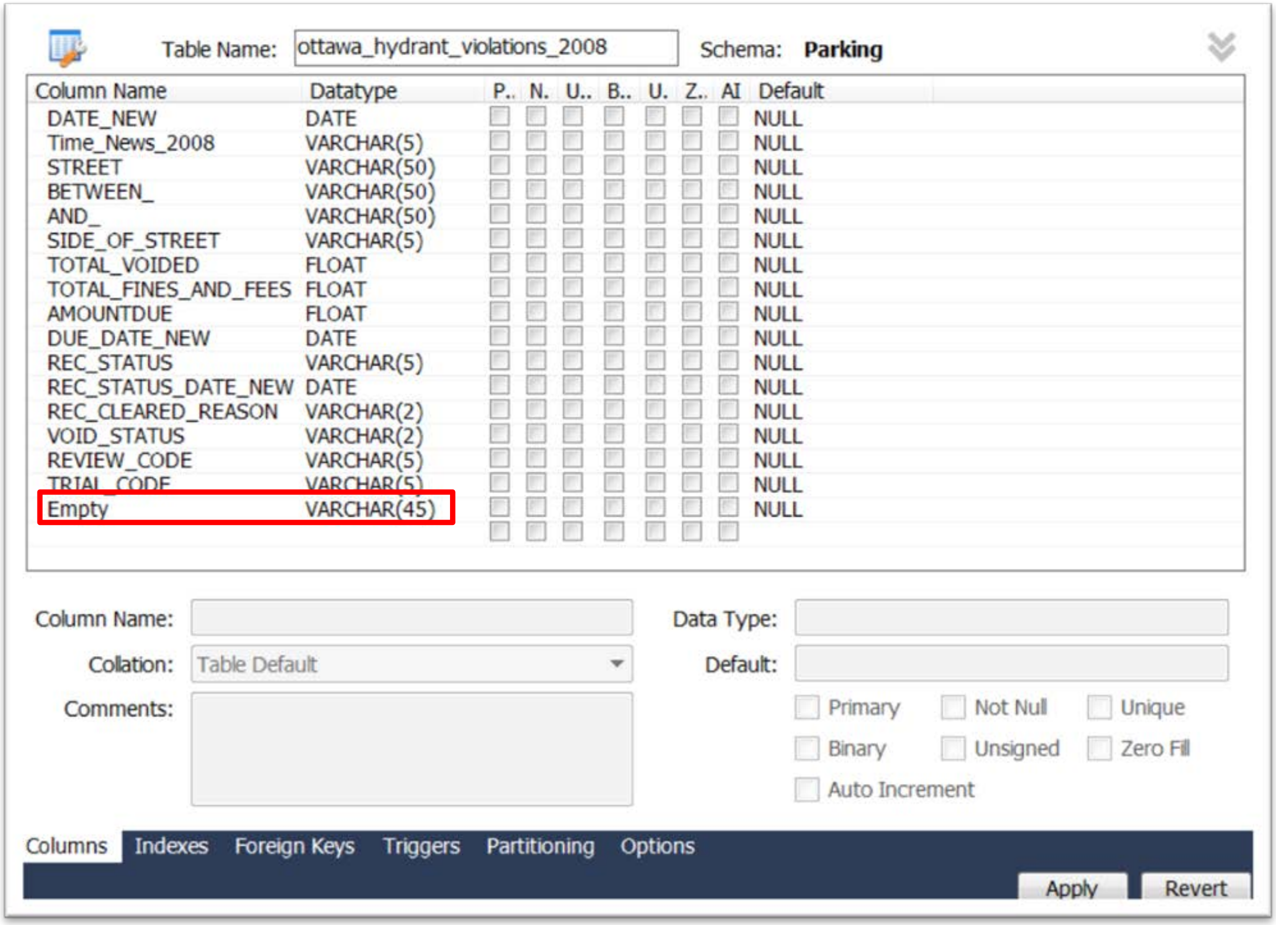


Table Name: ottawa_hydrant_violations_2008 Schema: **Parking**

Column Name	Datatype	P..	N.	U..	B..	U.	Z..	AI	Default
DATE_NEW	DATE	□	□	□	□	□	□	□	NULL
Time_News_2008	VARCHAR(5)	□	□	□	□	□	□	□	NULL
STREET	VARCHAR(50)	□	□	□	□	□	□	□	NULL
BETWEEN_	VARCHAR(50)	□	□	□	□	□	□	□	NULL
AND_	VARCHAR(50)	□	□	□	□	□	□	□	NULL
SIDE_OF_STREET	VARCHAR(5)	□	□	□	□	□	□	□	NULL
TOTAL_VOIDED	FLOAT	□	□	□	□	□	□	□	NULL
TOTAL_FINES_AND_FEES	FLOAT	□	□	□	□	□	□	□	NULL
AMOUNTDUE	FLOAT	□	□	□	□	□	□	□	NULL
DUE_DATE_NEW	DATE	□	□	□	□	□	□	□	NULL
REC_STATUS	VARCHAR(5)	□	□	□	□	□	□	□	NULL
REC_STATUS_DATE_NEW	DATE	□	□	□	□	□	□	□	NULL
REC_CLEARED_REASON	VARCHAR(2)	□	□	□	□	□	□	□	NULL
VOID_STATUS	VARCHAR(2)	□	□	□	□	□	□	□	NULL
REVIEW_CODE	VARCHAR(5)	□	□	□	□	□	□	□	NULL
TRIAL_CODE	VARCHAR(5)	□	□	□	□	□	□	□	NULL
Empty	VARCHAR(45)	□	□	□	□	□	□	□	NULL
		□	□	□	□	□	□	□	

Column Name: Data Type:

Collation: Table Default Default:

Comments:

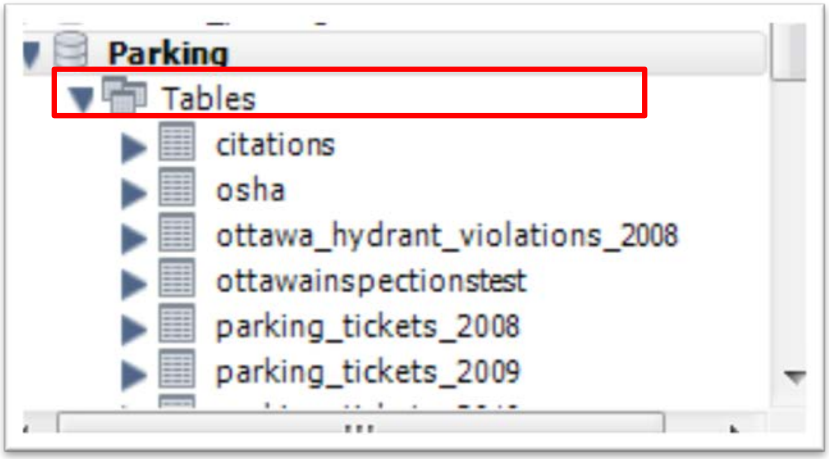
□ Primary □ Not Null □ Unique

□ Binary □ Unsigned □ Zero Fill

□ Auto Increment

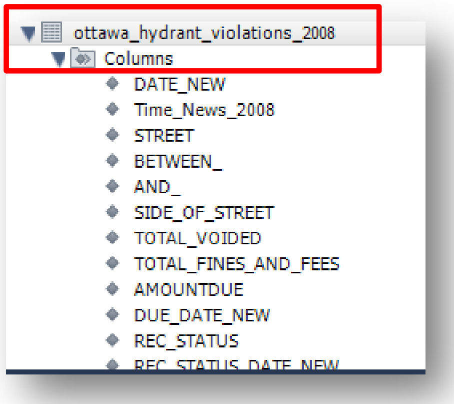
24. Notice that the first column – DATE_NEW -- is NOT a "primary" field. When creating the table, MySQL assumes that the values in the first column constitute a primary field. We don't want the dates to be a primary field, so it's crucial that you de-select primary key option. (NOTE: Otherwise, MySQL will only return about 365 results, instead of 5,465 rows). The NULL value is simply an unknown value in any database. If you're satisfied with the result, select the "Apply" tab to begin the process of saving the table.

25. To see the new table, right-click in the "Schema" that you've created, choose the "Refresh All" option from the shortcut menu. Click on the "Tables" icon to see your new table or file.



Parking
Tables
citations
osha
ottawa_hydrant_violations_2008
ottawainspectionstest
parking_tickets_2008
parking_tickets_2009

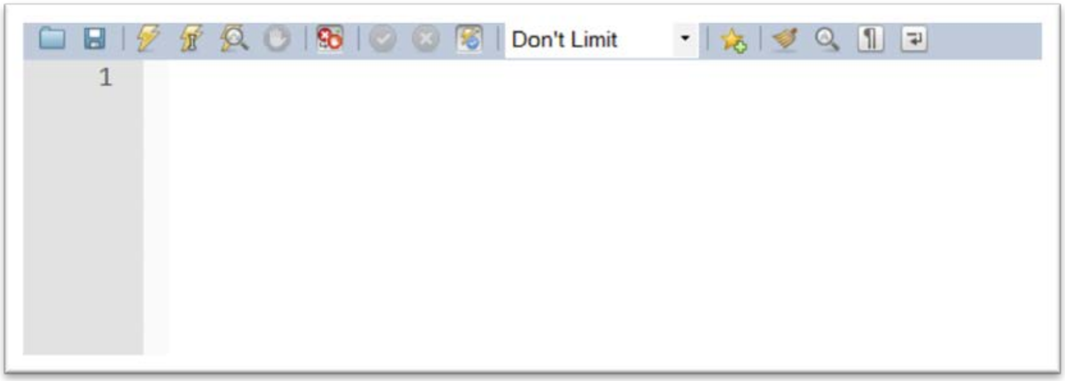
26. To see the columns in your table, click on the arrow to the left of the table you've just created.



ottawa_hydrant_violations_2008
Columns
DATE_NEW
Time_News_2008
STREET
BETWEEN_
AND_
SIDE_OF_STREET
TOTAL_VOIDED
TOTAL_FINES_AND_FEES
AMOUNTDUE
DUE_DATE_NEW
REC_STATUS
REC_STATUS_DATE_NEW

27. Now we're ready to import the file. To do so, we must go to a new "query" tab, which you can obtain by selecting the "New Query Tab" option from the "File" section of your menu.

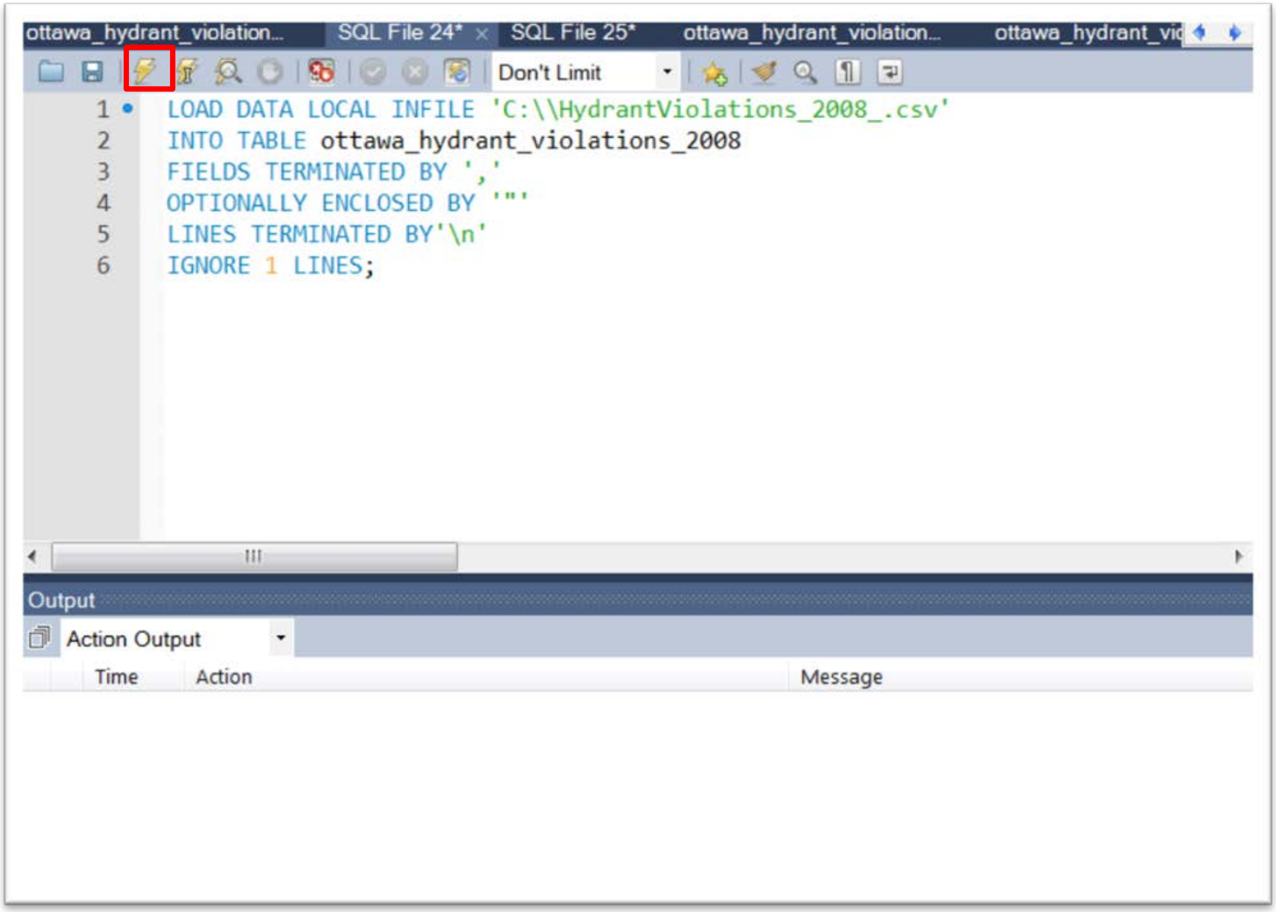
28. To obtain a new query tab, you could also select "SQL" icon under the menu's "File" section. Either option you choose, your blank query tab should look like this: (NOTE: in Computer-Assisted reporting, we use the MySQL Query Browser. Though the interface is slightly different, the concepts are the same as Workbench.)



29. You can also adjust the width of the panels on either side of the query tab. We will use the "LoadDataInFile" command described on pages 187-188 of the textbook.

30. For this tutorial, the files will differ, but the concept and the order of the commands is similar. It should look

like this:



```
1 LOAD DATA LOCAL INFILE 'C:\\HydrantViolations_2008_.csv'
2 INTO TABLE ottawa_hydrant_violations_2008
3 FIELDS TERMINATED BY ','
4 OPTIONALLY ENCLOSED BY '"'
5 LINES TERMINATED BY'\n'
6 IGNORE 1 LINES;
```

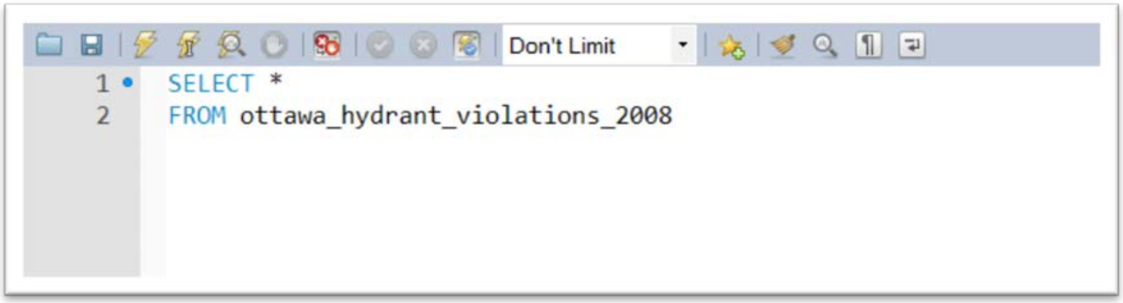
31. So let's break it down. Line 1: You have to "load" the file. Line 2: it must be loaded "into" a table. Line 3: this part of the query specifies that the information in each column is separated or terminated by a comma "," – not a tab "\t" referenced in the textbook's example, which references a different data set. Line 4: The text in each column is enclosed by quotation marks. Line 5: that each line (or record or row) is "terminated" by a carriage return ('\n'). Line 6: Because we have already created

column heads in our table, we are telling MySQL to “ignore” the first line in the table we are about to import.

32. To run the query, select the yellow lighten bolt icon, highlighted in the screen shot above, at the top left-hand side of the menu that contains the various icons.

33. You will get some “truncation” errors. Please ignore them for now, as they will not affect the data you are about to import.

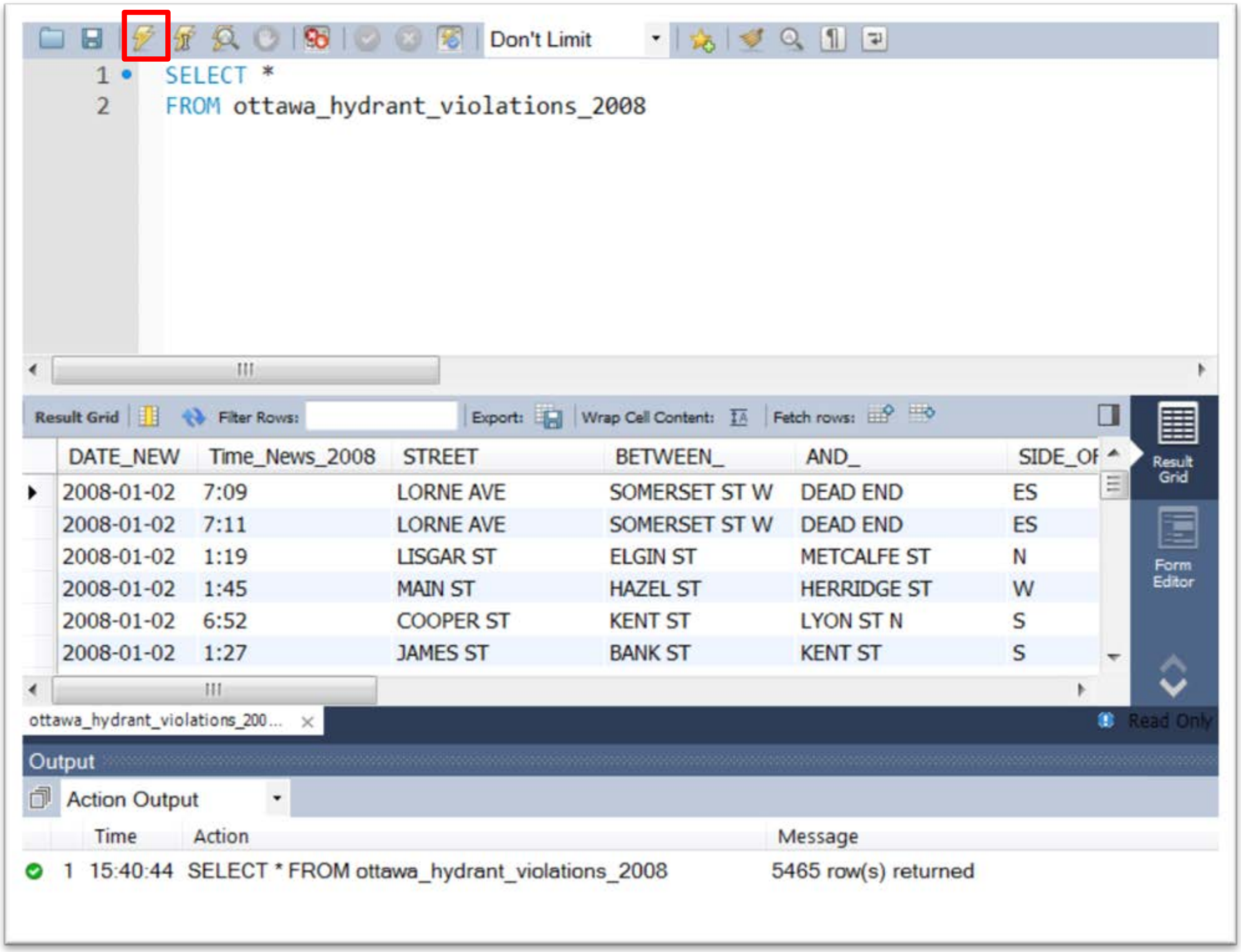
34. To see the data in your new table, open a query tab, and use a “SELECT” statement described on pages 190-192 of Computer-Assisted Reporting. Your query should look like this:



```
SELECT *
FROM ottawa_hydrant_violations_2008
```

35. Okay, before running the query, let’s break it down again. The asterisk to the right of “SELECT” tells MySQL that you want to select all the columns. The FROM tells MySQL where to do.

36. Let’s run the query.

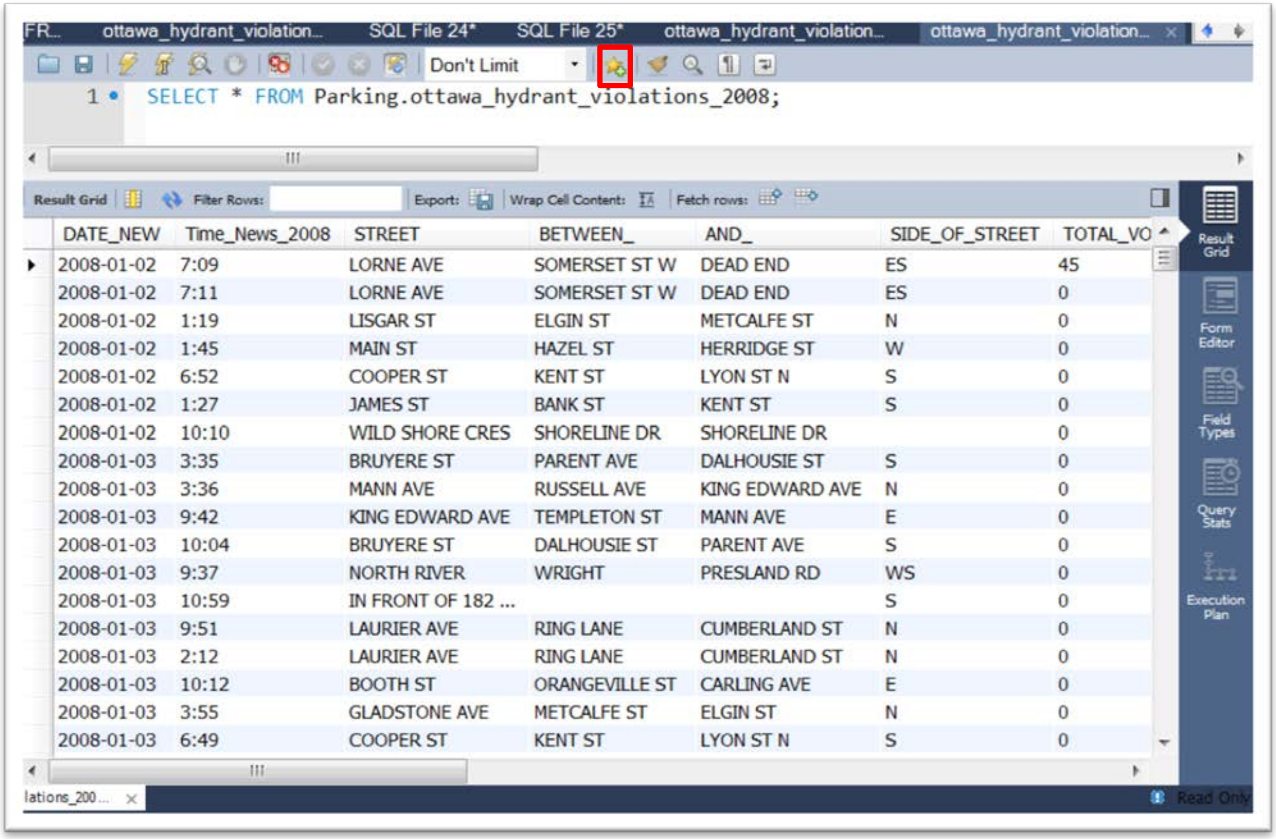


```
SELECT *
FROM ottawa_hydrant_violations_2008
```

DATE_NEW	Time_News_2008	STREET	BETWEEN_	AND_	SIDE_OF
2008-01-02	7:09	LORNE AVE	SOMERSET ST W	DEAD END	ES
2008-01-02	7:11	LORNE AVE	SOMERSET ST W	DEAD END	ES
2008-01-02	1:19	LISGAR ST	ELGIN ST	METCALFE ST	N
2008-01-02	1:45	MAIN ST	HAZEL ST	HERRIDGE ST	W
2008-01-02	6:52	COOPER ST	KENT ST	LYON ST N	S
2008-01-02	1:27	JAMES ST	BANK ST	KENT ST	S

```
     Time      Action                                        Message
1    15:40:44  SELECT * FROM ottawa_hydrant_violations_2008  5465 row(s) returned
```

37. To see the table, you can adjust the height of the panels below the query space.



```
1 • SELECT * FROM Parking.ottawa_hydrant_violations_2008;
```

DATE_NEW	Time_News_2008	STREET	BETWEEN_	AND_	SIDE_OF_STREET	TOTAL_VO
2008-01-02	7:09	LORNE AVE	SOMERSET ST W	DEAD END	ES	45
2008-01-02	7:11	LORNE AVE	SOMERSET ST W	DEAD END	ES	0
2008-01-02	1:19	LISGAR ST	ELGIN ST	METCALFE ST	N	0
2008-01-02	1:45	MAIN ST	HAZEL ST	HERRIDGE ST	W	0
2008-01-02	6:52	COOPER ST	KENT ST	LYON ST N	S	0
2008-01-02	1:27	JAMES ST	BANK ST	KENT ST	S	0
2008-01-02	10:10	WILD SHORE CRES	SHORELINE DR	SHORELINE DR		0
2008-01-03	3:35	BRUYERE ST	PARENT AVE	DALHOUSIE ST	S	0
2008-01-03	3:36	MANN AVE	RUSSELL AVE	KING EDWARD AVE	N	0
2008-01-03	9:42	KING EDWARD AVE	TEMPLETON ST	MANN AVE	E	0
2008-01-03	10:04	BRUYERE ST	DALHOUSIE ST	PARENT AVE	S	0
2008-01-03	9:37	NORTH RIVER	WRIGHT	PRESLAND RD	WS	0
2008-01-03	10:59	IN FRONT OF 182 ...			S	0
2008-01-03	9:51	LAURIER AVE	RING LANE	CUMBERLAND ST	N	0
2008-01-03	2:12	LAURIER AVE	RING LANE	CUMBERLAND ST	N	0
2008-01-03	10:12	BOOTH ST	ORANGEVILLE ST	CARLING AVE	E	0
2008-01-03	3:55	GLADSTONE AVE	METCALFE ST	ELGIN ST	N	0
2008-01-03	6:49	COOPER ST	KENT ST	LYON ST N	S	0

38. We can see that it has returned the same number of rows contained in the original file (NOTE: You should always toggle back and forth between the original csv file that is open in Excel, or your text editor to make sure you got everything.)

39. The “Snippets” icon allows you so save your queries, which you can do by clicking on the yellow star icon with a green “+” sign attached to the bottom, which is highlighted in the red square in the screenshot above. You’ll get a dialog box that will allow you to name the

query, which will then be deposited to the right of the table. You can also save the query by clicking on the “Save Script” option in the “File” portion of the menu above. Be sure to designate the area on your hard drive where you want to save the script.

40. Now that you’ve downloaded a file, you can experiment with some of the queries described in our textbook.

41. We will be continuing this exercise in class with subsequent years. So it’s important that you accomplish the tasks set out in this tutorial.

42. For more detailed information about MySQL, you can visit: http://dev.mysql.com/doc/refman/5.5/en/index.html