CS3860 Lab 1 - Database

MySQL install

Objective: Understand database management system installation, and gain a high level overview of database creation, defining table schemas, and the pragmatics of how to create tables, import data, and define primary keys and indexes.

Record your observations and submit the output of your queries in your lab report for each major step (denoted with [major step] below).

Deliverables
Complete a lab report containing the following:

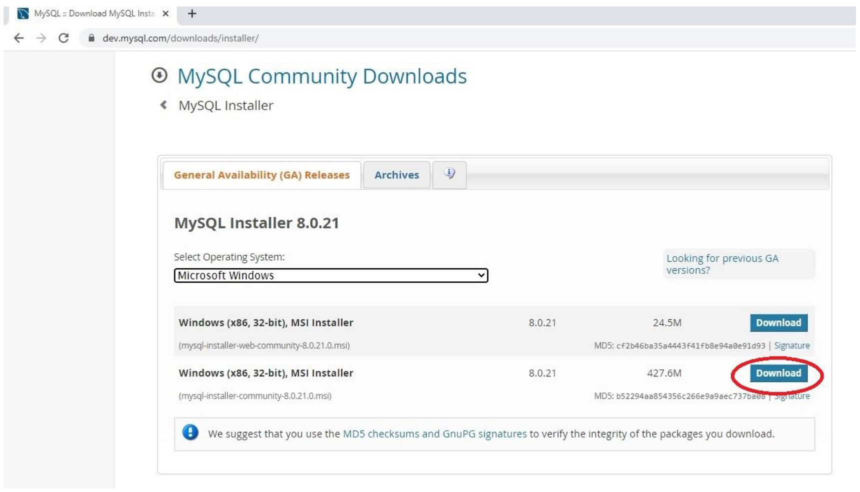
- What you've learned from the lab
- How it could be improved
- Observations and query results for each major step of the lab.
- ERD (Entity Relation Diagram)
- Submit to Canvas in PDF format!

Notes:

- ***When installing MySQL, it is important to create and remember your administrator login and password***.
- *Walk Through the Installation Process. Make sure you include options for command line, database administration, and automatic startup. Set and record your administrative password. Include command line tools.*

Download and install MySQL Installer

MySQL Community server(for all platforms): https://dev.mysql.com/downloads/installer/



MySQL :: Download MySQL Insta
dev.mysql.com/downloads/installer/

MySQL Community Downloads
MySQL Installer

General Availability (GA) Releases | Archives

MySQL Installer 8.0.21

Select Operating System: Microsoft Windows

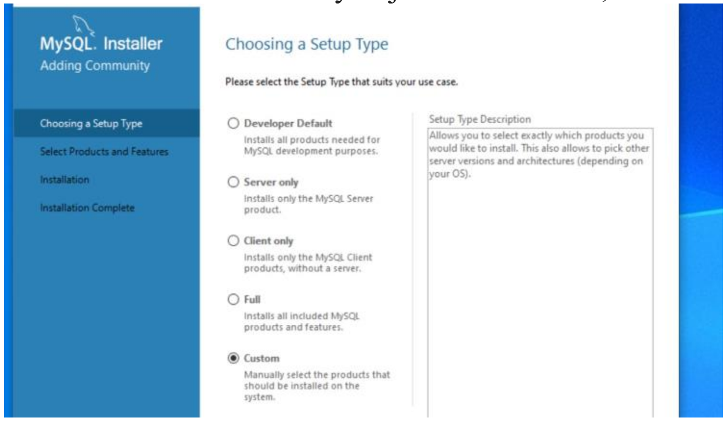
Looking for previous GA versions?

Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.21.0.msi)	8.0.21	24.5M MD5: cf2b46ba35a4443f41fb8e94a0e91d93 | Signature	Download
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-8.0.21.0.msi)	8.0.21	427.6M MD5: b52294aa854356c266e9a9aec737ba08 | Signature	Download

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

On the next screen, if asked to login/sign up you may simply click “No thanks, just start my download” link.

Double click the installer you just downloaded, choose the “Custom” setup type.



Choosing a Setup Type

Please select the Setup Type that suits your use case.

Developer Default
Installs all products needed for MySQL development purposes.

Server only
Installs only the MySQL Server product.

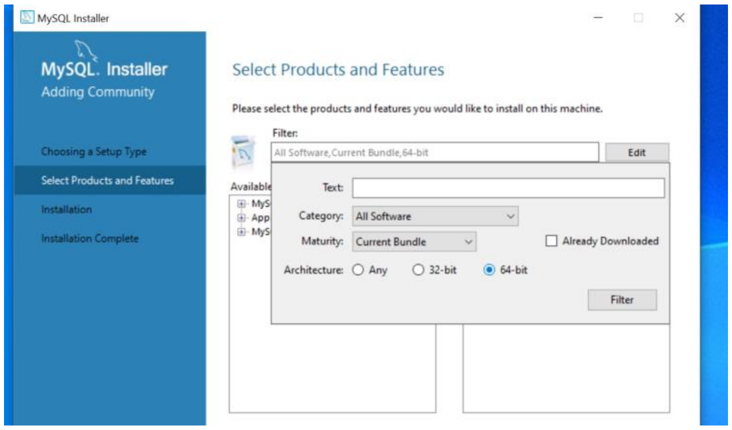
Client only
Installs only the MySQL Client products, without a server.

Full
Installs all included MySQL products and features.

[✓ Custom]
Manually select the products that should be installed on the system.

Setup Type Description
Allows you to select exactly which products you would like to install. This also allows to pick other server versions and architectures (depending on your OS).

You may filter to install products that are 64-bit.



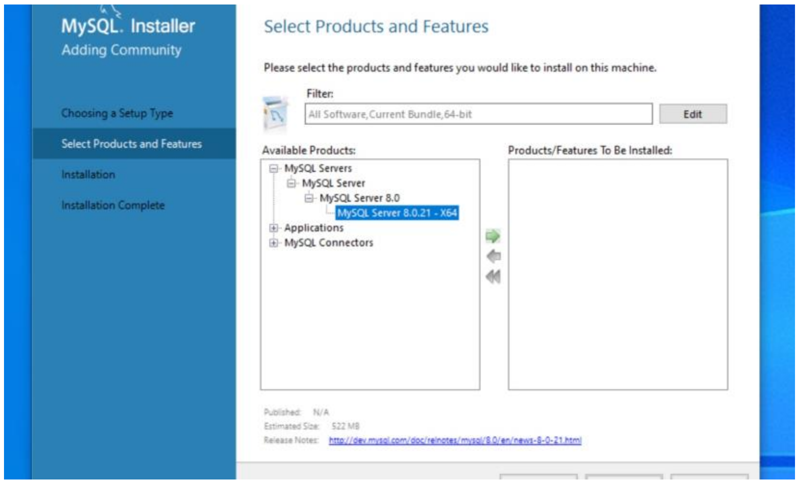
Select Products and Features

Please select the products and features you would like to install on this machine.

Filter: All Software,Current Bundle,64-bit

Text:
Category: All Software
Maturity: Current Bundle
Already Downloaded
Architecture: Any 32-bit [✓ 64-bit]

Select the products to install by clicking on each one as shown in following screenshot, and then clicking the green arrow to move them over to the “to be installed” column



Select Products and Features

Please select the products and features you would like to install on this machine.

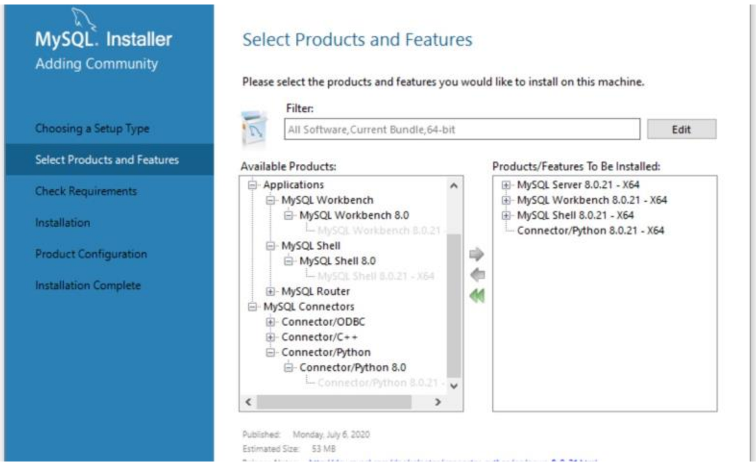
Filter: All Software,Current Bundle,64-bit

Available Products:
MySQL Servers
MySQL Server
MySQL Server 8.0
MySQL Server 8.0.21 - X64
Applications
MySQL Connectors

Products/Features To Be Installed:

Published: N/A
Estimated Size: 522 MB
Release Notes: http://dev.mysql.com/doc/relnotes/mysql/8.0/en/news-8-0-21.html

Please select Mysql Sever, Applications → Workbench, Mysql shell and python connector as shown below. **Make sure to match the 4 items shown in the right column before clicking next. Note that if you do not have python installed, it won't let you install the python connector. You can get that here (64bit windows): https://www.python.org/ftp/python/3.7.9/python-3.7.9-amd64.exe



Select Products and Features

Please select the products and features you would like to install on this machine.

Filter: All Software,Current Bundle,64-bit

Available Products:

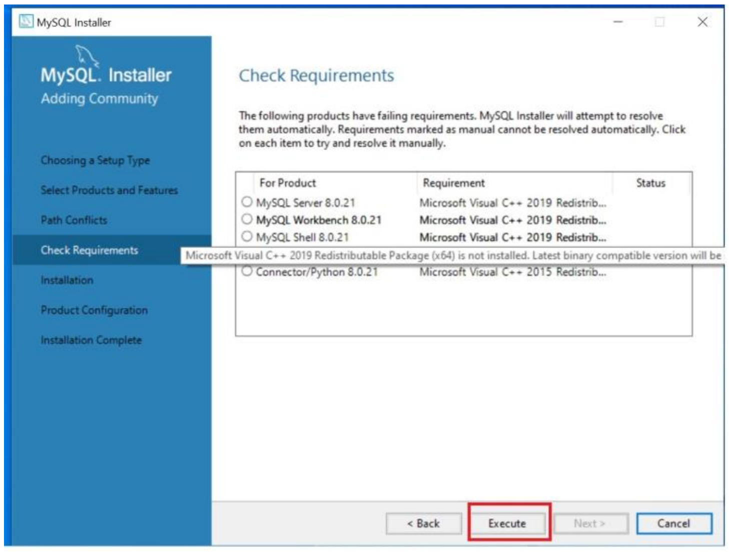
```
Applications
  MySQL Workbench
    MySQL Workbench 8.0
      MySQL Workbench 8.0.21
  MySQL Shell
    MySQL Shell 8.0
      MySQL Shell 8.0.21 - X64
  MySQL Router
MySQL Connectors
  Connector/ODBC
  Connector/C++
  Connector/Python
    Connector/Python 8.0
      Connector/Python 8.0.21
```

Products/Features To Be Installed:

```
MySQL Server 8.0.21 - X64
MySQL Workbench 8.0.21 - X64
MySQL Shell 8.0.21 - X64
Connector/Python 8.0.21 - X64
```

Published: Monday, July 6, 2020
Estimated Size: 53 MB

In the next step, if you will find some requirements are not met for few mysql products. But mysql installer will try to download these requirements and install them for you. So, click execute on the below screen.



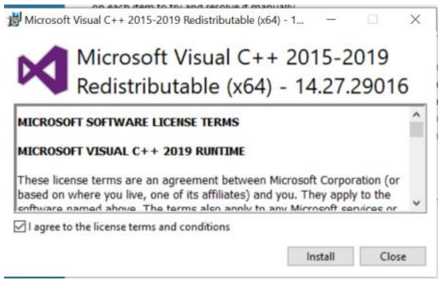
Check Requirements

The following products have failing requirements. MySQL Installer will attempt to resolve them automatically. Requirements marked as manual cannot be resolved automatically. Click on each item to try and resolve it manually.

For Product	Requirement	Status
MySQL Server 8.0.21	Microsoft Visual C++ 2019 Redistrib...	
MySQL Workbench 8.0.21	Microsoft Visual C++ 2019 Redistrib...	
MySQL Shell 8.0.21	Microsoft Visual C++ 2019 Redistrib...	
Connector/Python 8.0.21	Microsoft Visual C++ 2015 Redistrib...	

Microsoft Visual C++ 2019 Redistributable Package (x64) is not installed. Latest binary compatible version will be

You will see that mysql tries to install Visual C++ (needed for Workbench), so check "I agree" and click install in the below screen.


Microsoft Visual C++ 2015-2019 Redistributable (x64) - 1...

Microsoft Visual C++ 2015-2019 Redistributable (x64) - 14.27.29016

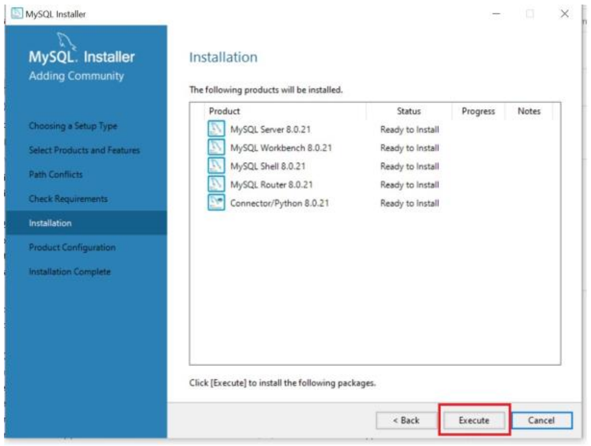
MICROSOFT SOFTWARE LICENSE TERMS

MICROSOFT VISUAL C++ 2019 RUNTIME

These license terms are an agreement between Microsoft Corporation (or based on where you live, one of its affiliates) and you. They apply to the software named above. The terms also apply to any Microsoft services or

☑ I agree to the license terms and conditions

Now, since all requirements are met, you are ready to install all Mysql products, so click execute on below screen.


MySQL Installer

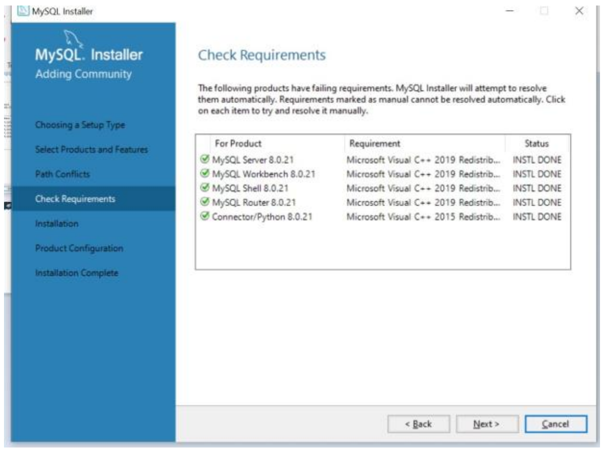
Installation

The following products will be installed.

Product	Status	Progress	Notes
MySQL Server 8.0.21	Ready to Install		
MySQL Workbench 8.0.21	Ready to Install		
MySQL Shell 8.0.21	Ready to Install		
MySQL Router 8.0.21	Ready to Install		
Connector/Python 8.0.21	Ready to Install		

Click [Execute] to install the following packages.

Upon successful installation you will see the following screen.


MySQL Installer

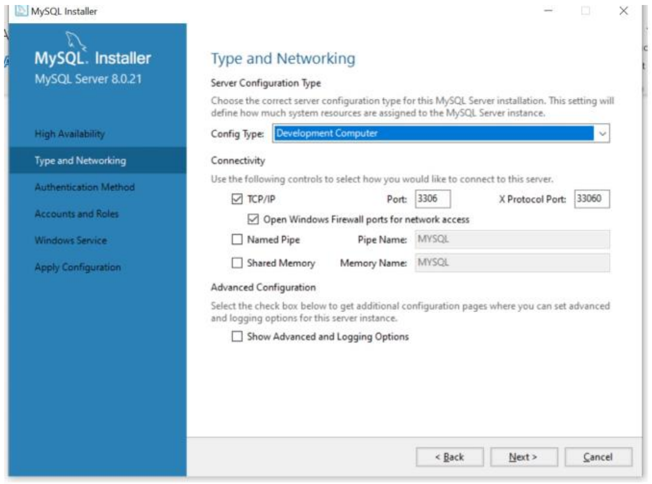
Check Requirements

The following products have failing requirements. MySQL Installer will attempt to resolve them automatically. Requirements marked as manual cannot be resolved automatically. Click on each item to try and resolve it manually.

For Product	Requirement	Status
✓ MySQL Server 8.0.21	Microsoft Visual C++ 2019 Redistrib...	INSTL DONE
✓ MySQL Workbench 8.0.21	Microsoft Visual C++ 2019 Redistrib...	INSTL DONE
✓ MySQL Shell 8.0.21	Microsoft Visual C++ 2019 Redistrib...	INSTL DONE
✓ MySQL Router 8.0.21	Microsoft Visual C++ 2019 Redistrib...	INSTL DONE
✓ Connector/Python 8.0.21	Microsoft Visual C++ 2015 Redistrib...	INSTL DONE

Click Next, you will see the product configuration screen, click Next again.
On “High Availability” screen, select Standalone Mysql Server (you do not need *Innodb* cluster server or router)

In the "Type and Networking" screen (shown below), select "Development Computer", the default port is 3306, do not change anything on this screen, click Next


MySQL Installer

MySQL. Installer
MySQL Server 8.0.21

High Availability
Type and Networking
Authentication Method
Accounts and Roles
Windows Service
Apply Configuration

Type and Networking

Server Configuration Type

Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.

Config Type: Development Computer

Connectivity

Use the following controls to select how you would like to connect to this server.

- [✓] TCP/IP Port: 3306 X Protocol Port: 33060
  - [✓] Open Windows Firewall ports for network access
- [ ] Named Pipe Pipe Name: MYSQL
- [ ] Shared Memory Memory Name: MYSQL

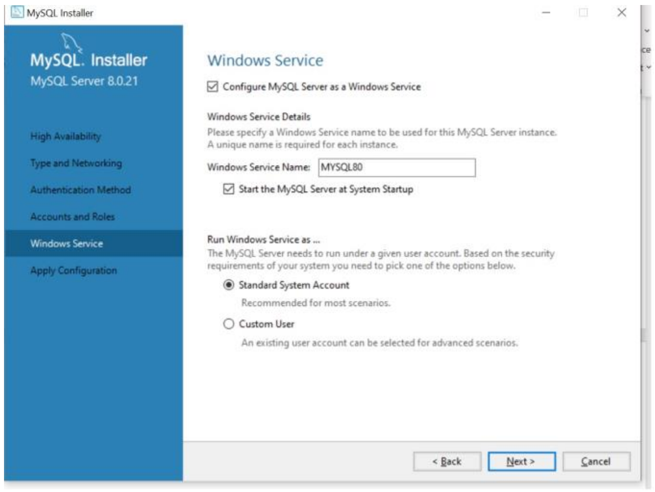
Advanced Configuration

Select the check box below to get additional configuration pages where you can set advanced and logging options for this server instance.

- [ ] Show Advanced and Logging Options

< Back Next > Cancel

Next, choose "Use strong password Encryption for Authentication" option for Authentication Method screen.
Next, enter your password and **remember it.**
Now, set up mysql as windows service as shown below.


MySQL Installer

MySQL. Installer
MySQL Server 8.0.21

High Availability
Type and Networking
Authentication Method
Accounts and Roles
Windows Service
Apply Configuration

Windows Service

- [✓] Configure MySQL Server as a Windows Service

Windows Service Details

Please specify a Windows Service name to be used for this MySQL Server instance. A unique name is required for each instance.

Windows Service Name: MYSQL80

- [✓] Start the MySQL Server at System Startup

Run Windows Service as ...

The MySQL Server needs to run under a given user account. Based on the security requirements of your system you need to pick one of the options below.

- (●) Standard System Account
  Recommended for most scenarios.
- ( ) Custom User
  An existing user account can be selected for advanced scenarios.

< Back Next > Cancel

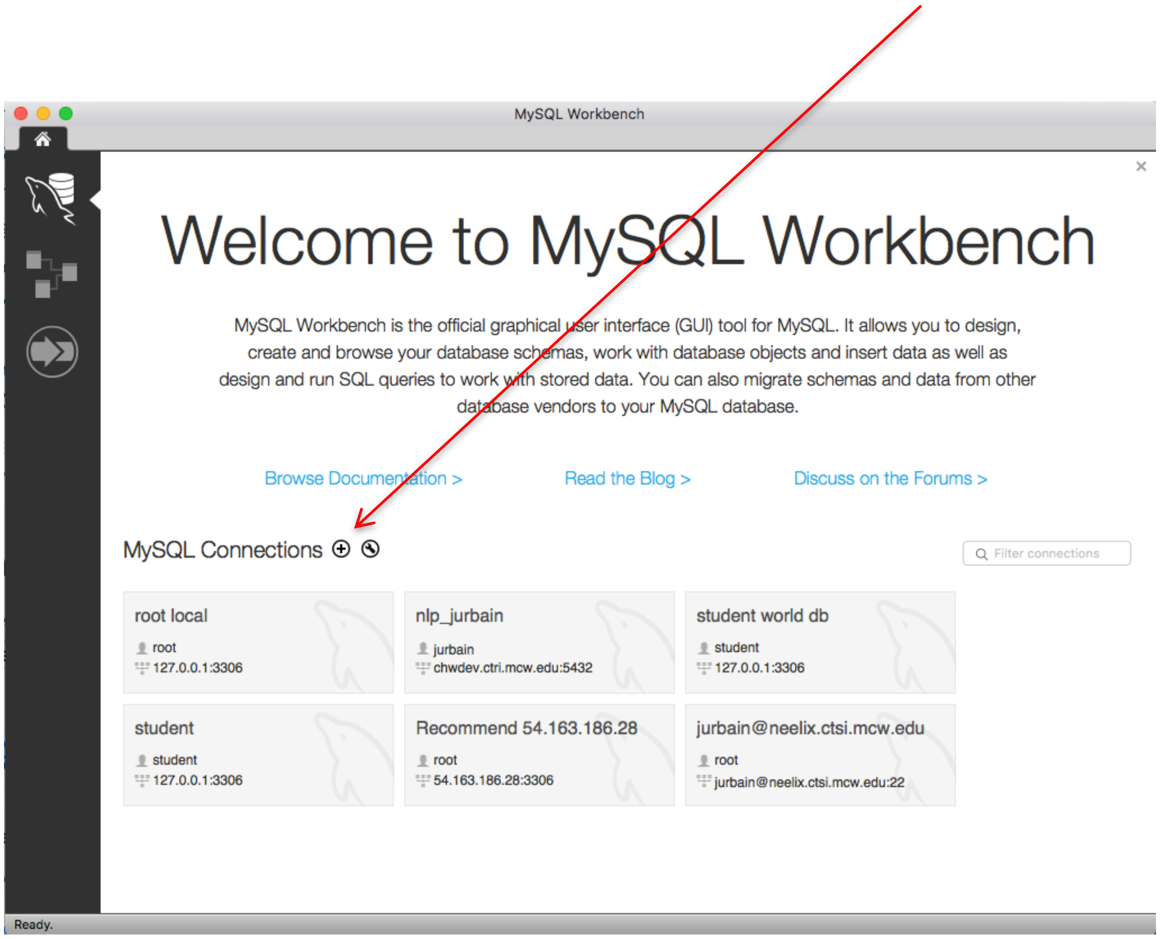
Finally, finish your installation by clicking execute on Apply Configuration screen.

Open MySQL Workbench.

You may perform the following steps to create a database and selecting a database schema by using either the *MySQL Workbench - SQL Development Tools* or the *command line*. If you use the *Query Browser* in the workbench to create a database, you can **do not need to create the database with the *command line*** (it only needs to be created once). However, you still need to look at the command line instruction section because we will be using the command line to load in a data file to populate our database.

Setting of a database connection and creating a database schema

Start *MySQL* Workbench. Under *SQL Development*, select ***Connection*** +.



MySQL Workbench

Welcome to MySQL Workbench

MySQL Workbench is the official graphical user interface (GUI) tool for MySQL. It allows you to design, create and browse your database schemas, work with database objects and insert data as well as design and run SQL queries to work with stored data. You can also migrate schemas and data from other database vendors to your MySQL database.

Browse Documentation > Read the Blog > Discuss on the Forums >

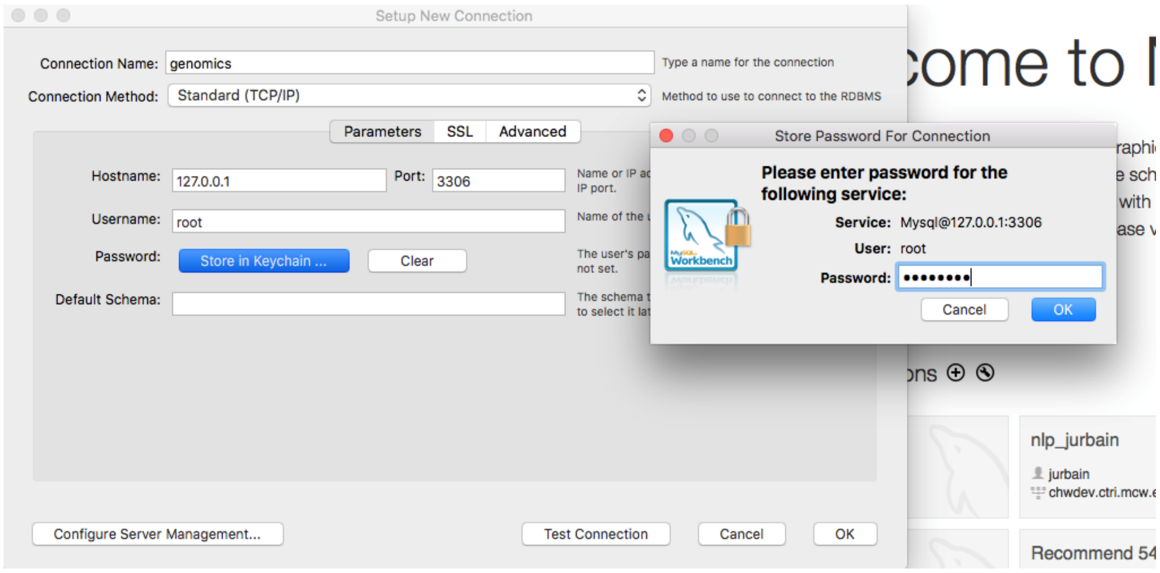
MySQL Connections ⊕ ⊗

Filter connections

Connection	User	Host
root local	root	127.0.0.1:3306
nlp_jurbain	jurbain	chwdev.ctri.mcw.edu:5432
student world db	student	127.0.0.1:3306
student	student	127.0.0.1:3306
Recommend 54.163.186.28	root	54.163.186.28:3306
jurbain@neelix.ctsi.mcw.edu	root	jurbain@neelix.ctsi.mcw.edu:22

Ready.

Provide a logical name for your ***connection***, enter your user name and password (store in vault). Leave *Default Schema* empty. Try *Test Connection*.


Setup New Connection

Connection Name: genomics — Type a name for the connection

Connection Method: Standard (TCP/IP) — Method to use to connect to the RDBMS

Parameters | SSL | Advanced

Hostname: 127.0.0.1 Port: 3306 — Name or IP ad... IP port.

Username: root — Name of the ...

Password: Store in Keychain ... Clear — The user's pa... not set.

Default Schema: — The schema t... to select it lat...

Configure Server Management... Test Connection Cancel OK

Store Password For Connection

Please enter password for the following service:

Service: Mysql@127.0.0.1:3306

User: root

Password: ••••••••

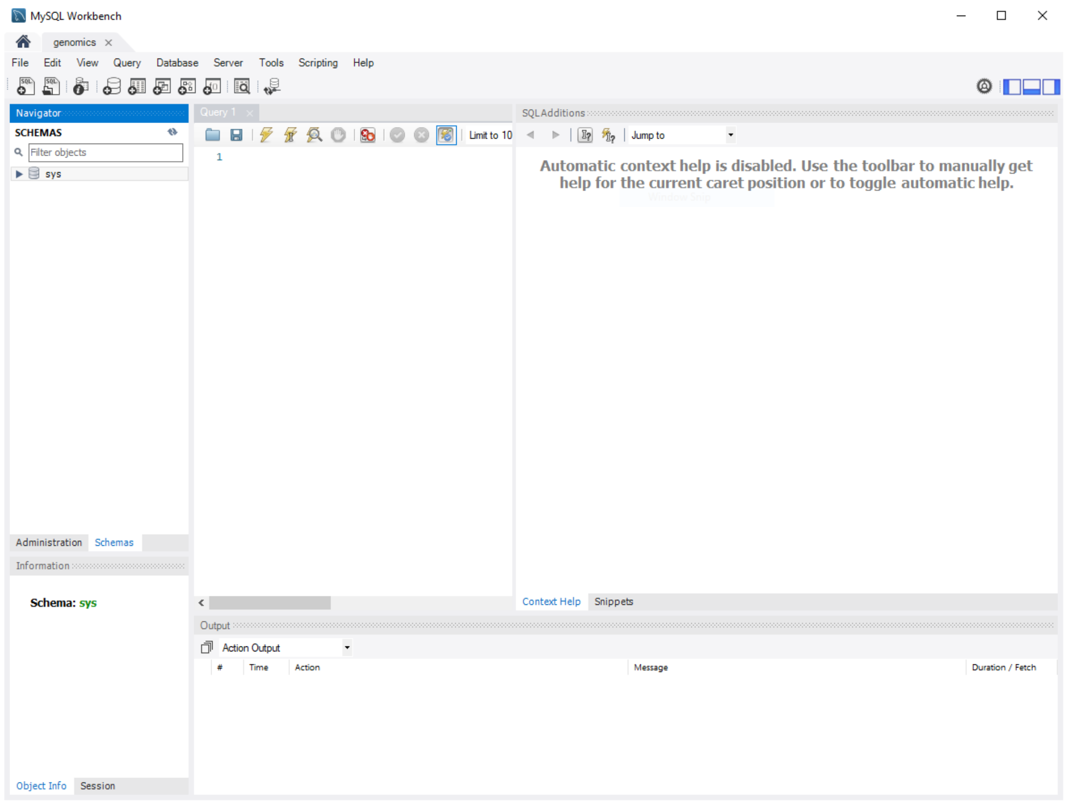
Cancel OK

Note: If you have trouble connecting, in the Control Panel, select Windows Firewall. Select the Exceptions tab, add MySQL or Port number: 3306 and name MySQL. If you still have trouble, check to make sure the MySQL service is running by "services.msc" from the windows command line prompt (Start->cmd).

Click *OK*. Then double click on the connection to open it.

In the future, you will just be able to click on your connections on the main screen.

The SQL Editor will now be displayed showing any existing databases on the schema tab (left about halfway down). The "sys" schema was created by default during installation for me).

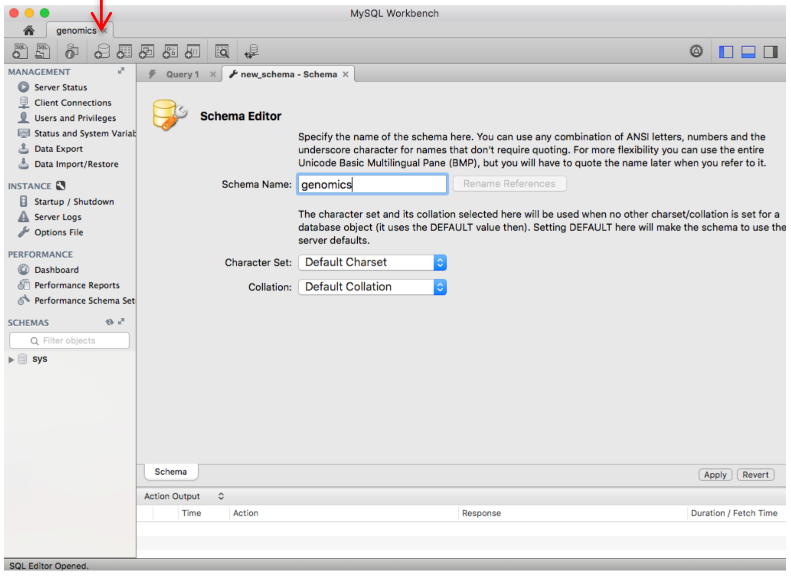


Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

Schema: sys

Select the can, or right-mouse-click on one of the existing databases in the *Object Browser* (left-hand panel) and select *Create New Schema* (the words schema and database are used interchangeably in this program). Name your new schema“*genomics*”. Select *Apply*, (And Apply again) then select Finish for *Execute SQL Statements*.

Note: The image below shows the “can” icon:



Schema Editor

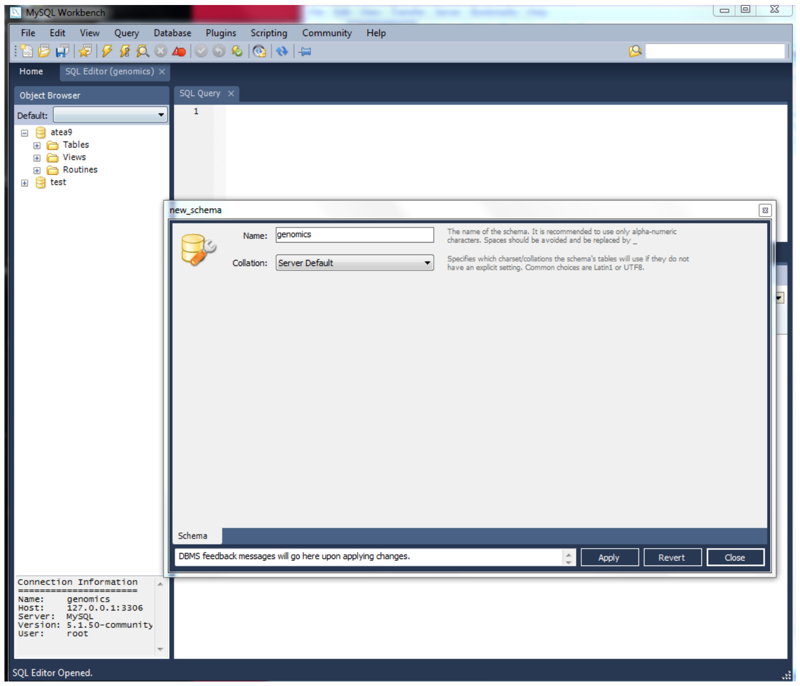
Specify the name of the schema here. You can use any combination of ANSI letters, numbers and the underscore character for names that don't require quoting. For more flexibility you can use the entire Unicode Basic Multilingual Pane (BMP), but you will have to quote the name later when you refer to it.

Schema Name: genomics

The character set and its collation selected here will be used when no other charset/collation is set for a database object (it uses the DEFAULT value then). Setting DEFAULT here will make the schema to use the server defaults.

Character Set: Default Charset

Collation: Default Collation



Name: genomics

The name of the schema. It is recommended to use only alpha-numeric characters. Spaces should be avoided and be replaced by _

Collation: Server Default

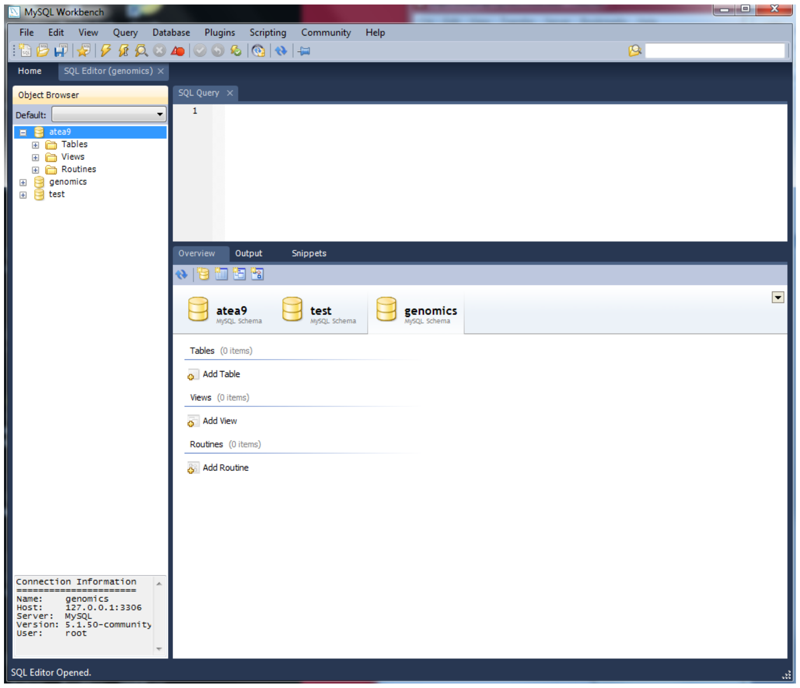
Specifies which charset/collations the schema's tables will use if they do not have an explicit setting. Common choices are Latin1 or UTF8.

DBMS feedback messages will go here upon applying changes.

```
Connection Information
======================
Name:    genomics
Host:    127.0.0.1:3306
Server:  MySQL
Version: 5.1.50-community
User:    root
```

SQL Editor Opened.

A new database schema *genomics* should be available. Select the *genomics* schema (double click on it to **bold** it – this sets it to the current database “in use”).



```
Connection Information
======================
Name:    genomics
Host:    127.0.0.1:3306
Server:  MySQL
Version: 5.1.50-community
User:    root
```

SQL Editor Opened.

The following 2 pages are the command line approach to creating a database, You should review these (even though you just created the genomics database with the GUI using the steps above).

Connecting to the MySQL Server for the First Time Using *Command line*

Two options:

Open the windows command line (Start->cmd) and navigate using the “cd” command to *C:\\Program Files\MySQL\MySQLServer 8.0\bin*

1. (harder) Use the mysql command to launch mysql

```
%>mysql [options] [database]
```

[database] is the name of your database. To list out the [options]:

```
%>mysql --help
```

For now, we just need to connect to the database server on your system:

```
%>mysql -u root -p --local-infile
```

root is the default login for MySQL. local-infile is an option to allow you to load local files. You will be prompted for the password.

Or 2 (easier). Open the “MySQL 8.0 Command Line Client” directly from the Start menu. You will be prompted for the password.

The following should appear:

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Blah blah blah
```

Set the following variable to allow loading local file (this variable must be set to 1 to enable loading data from a local file):

```
SET GLOBAL local_infile = 1;
```

If you want to, you can change the password to something other than the default or the value you set on install as follows:

```
mysql>SET PASSWORD FOR 'root'@'localhost' =PASSWORD('secret_password');
```

The 'root', which is the username, and 'localhost', which is the hostname, constitute a unique user in MySQL.

'localhost' is a name used to specify the local server. 'root'@'localhost' tells the MySQL server to set the password for a user named 'root' that will connect specifically from the local server (thus 'localhost').

If you changed your password, in order to test the new password, exit the MySQL terminal connection using the following command:

```
mysql>\q
```

This will return you to the system shell. Now log back into the monitor, this time using the following command:

```
%>mysql -u root -p
```

Doing so will result in a prompt for the root user password, as follows:

```
Enter password:
```

Creating and using a database (you likely already did this with the GUI)

```
mysql> SHOW DATABASES;
information_schema
mysql
test
```

If the `test` database exists, try to access it:

```
mysql> USE test
Database changed
```

Create the genomics database (only do this if you did not create it using the GUI above). In the mysql program:

mysql> create database *genomics*;

```
mysql> SHOW DATABASES;
information_schema
genomics
mysql
test
```

Next, you need to select the database to *use*. Select the genomics database (you can do this not matter which way you created it in the first place):

```
mysql> use genomics;
Database changed
```

Importing data into a database (use command line for actually importing the data. Either command line or GUI can be used to create the table where the data will go)

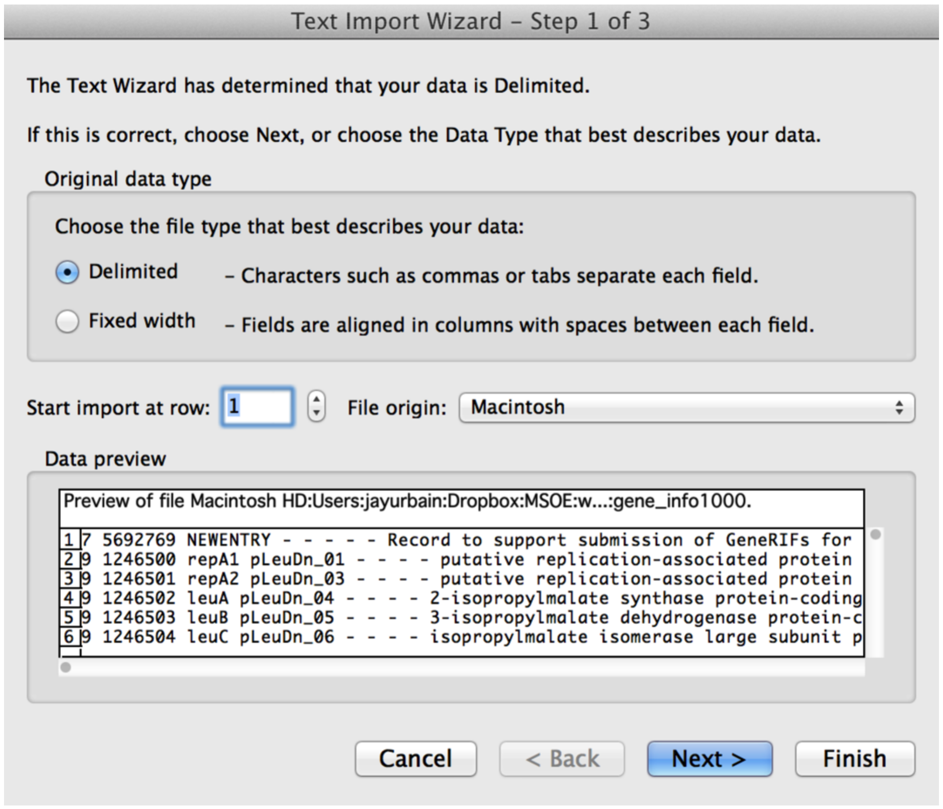
Note that creating the table can also be done using the command line by following instructions on subsequent pages.

To further your understanding, first take a look at the data in excel [record observations about this major step]:

Our goal is to import the *Entrez gene* information database file (a tab-delimited text file) from your *data* directory.

First, we will look at the data in Excel. Open Excel. From within **Excel**, open up the file *gene_info1000*. In the file dialog, you may need to enable '*All Files'* to see the file. Follow the import steps shown below.

Inspect the different fields. Note that it can be difficult to initially size the fields. *gene_info1000* contains the first 1000 rows of the Entrez Gene – gene information file.


Text Import Wizard – Step 1 of 3

The Text Wizard has determined that your data is Delimited.

If this is correct, choose Next, or choose the Data Type that best describes your data.

Original data type

Choose the file type that best describes your data:

● Delimited – Characters such as commas or tabs separate each field.

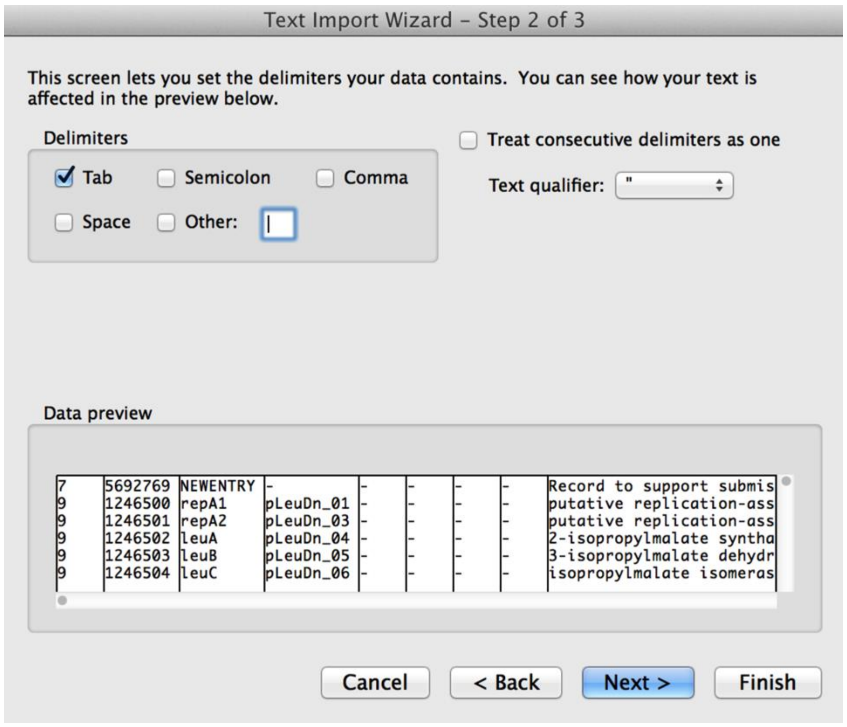
○ Fixed width – Fields are aligned in columns with spaces between each field.

Start import at row: 1 File origin: Macintosh

Data preview

Preview of file Macintosh HD:Users:jayurbain:Dropbox:MSOE:w...:gene_info1000.

```
1 7 5692769 NEWENTRY - - - - - Record to support submission of GeneRIFs for
2 9 1246500 repA1 pLeuDn_01 - - - - putative replication-associated protein
3 9 1246501 repA2 pLeuDn_03 - - - - putative replication-associated protein
4 9 1246502 leuA pLeuDn_04 - - - - 2-isopropylmalate synthase protein-coding
5 9 1246503 leuB pLeuDn_05 - - - - 3-isopropylmalate dehydrogenase protein-c
6 9 1246504 leuC pLeuDn_06 - - - - isopropylmalate isomerase large subunit p
```



Text Import Wizard – Step 2 of 3

This screen lets you set the delimiters your data contains. You can see how your text is affected in the preview below.

Delimiters

[✓ Tab] Semicolon Comma Space Other:

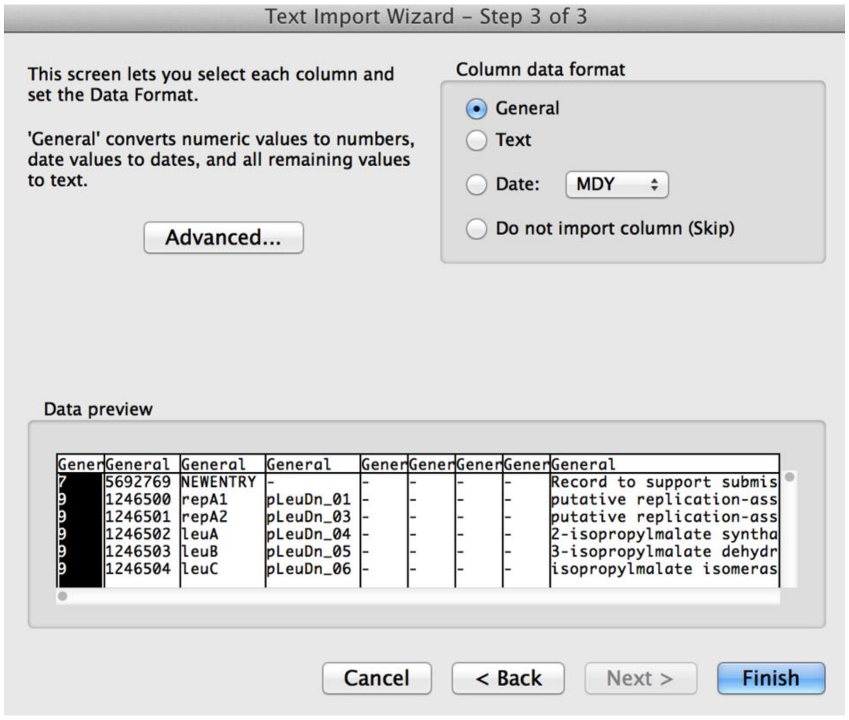
Treat consecutive delimiters as one

Text qualifier: "

Data preview

7	5692769	NEWENTRY	-	-	-	-	-	Record to support submis
9	1246500	repA1	pLeuDn_01	-	-	-	-	putative replication-ass
9	1246501	repA2	pLeuDn_03	-	-	-	-	putative replication-ass
9	1246502	leuA	pLeuDn_04	-	-	-	-	2-isopropylmalate syntha
9	1246503	leuB	pLeuDn_05	-	-	-	-	3-isopropylmalate dehydr
9	1246504	leuC	pLeuDn_06	-	-	-	-	isopropylmalate isomeras

Cancel < Back Next > Finish



Text Import Wizard – Step 3 of 3

This screen lets you select each column and set the Data Format.

'General' converts numeric values to numbers, date values to dates, and all remaining values to text.

Advanced...

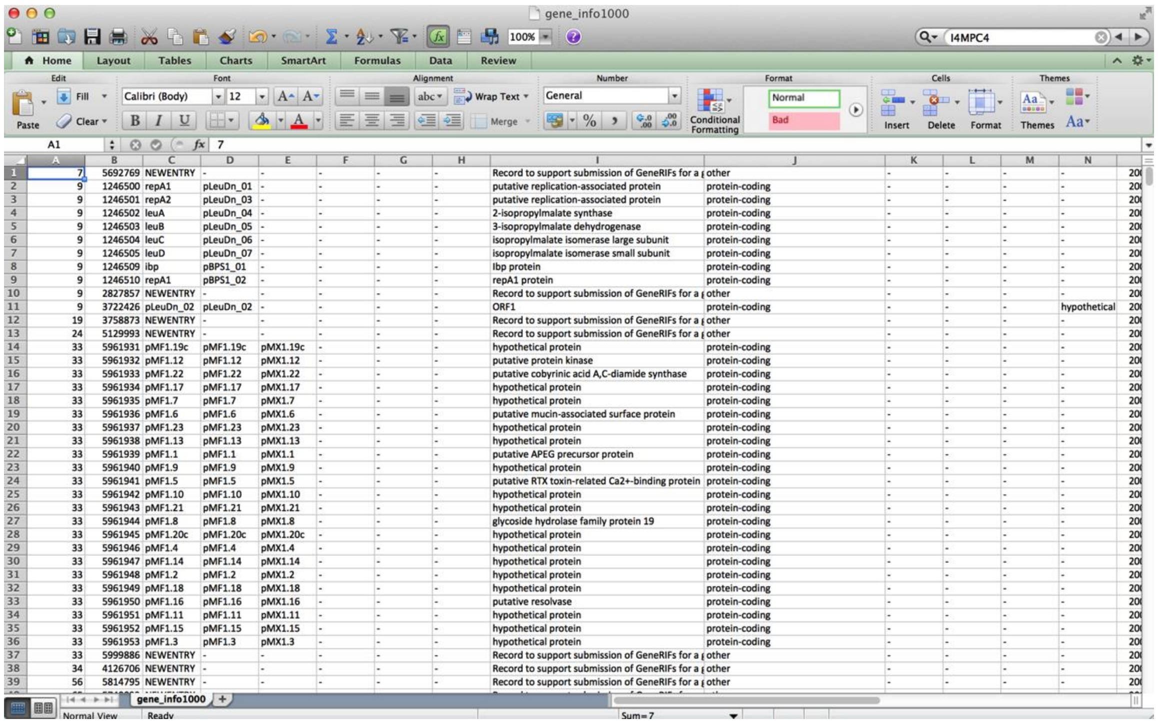
Column data format

[✓ General] Text Date: MDY Do not import column (Skip)

Data preview

Gener	General	General	General	Gener	Gener	Gener	Gener	General
7	5692769	NEWENTRY	-	-	-	-	-	Record to support submis
9	1246500	repA1	pLeuDn_01	-	-	-	-	putative replication-ass
9	1246501	repA2	pLeuDn_03	-	-	-	-	putative replication-ass
9	1246502	leuA	pLeuDn_04	-	-	-	-	2-isopropylmalate syntha
9	1246503	leuB	pLeuDn_05	-	-	-	-	3-isopropylmalate dehydr
9	1246504	leuC	pLeuDn_06	-	-	-	-	isopropylmalate isomeras

Cancel < Back Next > Finish



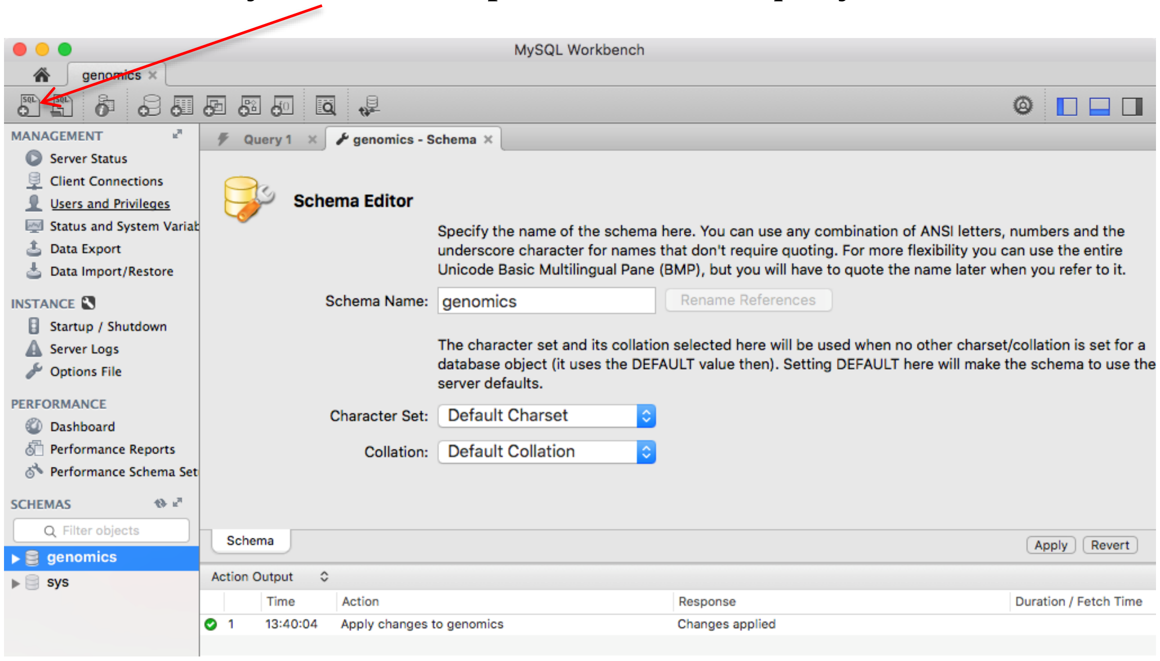
gene_info1000

A1 fx 7

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
1	7	5692769	NEWENTRY	-	-	-	-	-	Record to support submission of GeneRIFs for a g	other	-	-	-	-	20
2	9	1246500	repA1	pLeuDn_01	-	-	-	-	putative replication-associated protein	protein-coding	-	-	-	-	20
3	9	1246501	repA2	pLeuDn_03	-	-	-	-	putative replication-associated protein	protein-coding	-	-	-	-	20
4	9	1246502	leuA	pLeuDn_04	-	-	-	-	2-isopropylmalate synthase	protein-coding	-	-	-	-	20
5	9	1246503	leuB	pLeuDn_05	-	-	-	-	3-isopropylmalate dehydrogenase	protein-coding	-	-	-	-	20
6	9	1246504	leuC	pLeuDn_06	-	-	-	-	isopropylmalate isomerase large subunit	protein-coding	-	-	-	-	20
7	9	1246505	leuD	pLeuDn_07	-	-	-	-	isopropylmalate isomerase small subunit	protein-coding	-	-	-	-	20
8	9	1246509	ibp	pBPS1_01	-	-	-	-	Ibp protein	protein-coding	-	-	-	-	20
9	9	1246510	repA1	pBPS1_02	-	-	-	-	repA1 protein	protein-coding	-	-	-	-	20
10	9	2827857	NEWENTRY	-	-	-	-	-	Record to support submission of GeneRIFs for a g	other	-	-	-	-	20
11	9	3722426	pLeuDn_02	pLeuDn_02	-	-	-	-	ORF1	protein-coding	-	-	-	hypothetical	20
12	19	3758873	NEWENTRY	-	-	-	-	-	Record to support submission of GeneRIFs for a g	other	-	-	-	-	20
13	24	5129993	NEWENTRY	-	-	-	-	-	Record to support submission of GeneRIFs for a g	other	-	-	-	-	20
14	33	5961931	pMF1.19c	pMF1.19c	pMX1.19c	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
15	33	5961932	pMF1.12	pMF1.12	pMX1.12	-	-	-	putative protein kinase	protein-coding	-	-	-	-	20
16	33	5961933	pMF1.22	pMF1.22	pMX1.22	-	-	-	putative cobyrinic acid A,C-diamide synthase	protein-coding	-	-	-	-	20
17	33	5961934	pMF1.17	pMF1.17	pMX1.17	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
18	33	5961935	pMF1.7	pMF1.7	pMX1.7	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
19	33	5961936	pMF1.6	pMF1.6	pMX1.6	-	-	-	putative mucin-associated surface protein	protein-coding	-	-	-	-	20
20	33	5961937	pMF1.23	pMF1.23	pMX1.23	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
21	33	5961938	pMF1.13	pMF1.13	pMX1.13	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
22	33	5961939	pMF1.1	pMF1.1	pMX1.1	-	-	-	putative APEG precursor protein	protein-coding	-	-	-	-	20
23	33	5961940	pMF1.9	pMF1.9	pMX1.9	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
24	33	5961941	pMF1.5	pMF1.5	pMX1.5	-	-	-	putative RTX toxin-related Ca2+-binding protein	protein-coding	-	-	-	-	20
25	33	5961942	pMF1.10	pMF1.10	pMX1.10	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
26	33	5961943	pMF1.21	pMF1.21	pMX1.21	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
27	33	5961944	pMF1.8	pMF1.8	pMX1.8	-	-	-	glycoside hydrolase family protein 19	protein-coding	-	-	-	-	20
28	33	5961945	pMF1.20c	pMF1.20c	pMX1.20c	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
29	33	5961946	pMF1.4	pMF1.4	pMX1.4	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
30	33	5961947	pMF1.14	pMF1.14	pMX1.14	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
31	33	5961948	pMF1.2	pMF1.2	pMX1.2	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
32	33	5961949	pMF1.18	pMF1.18	pMX1.18	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
33	33	5961950	pMF1.16	pMF1.16	pMX1.16	-	-	-	putative resolvase	protein-coding	-	-	-	-	20
34	33	5961951	pMF1.11	pMF1.11	pMX1.11	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
35	33	5961952	pMF1.15	pMF1.15	pMX1.15	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
36	33	5961953	pMF1.3	pMF1.3	pMX1.3	-	-	-	hypothetical protein	protein-coding	-	-	-	-	20
37	33	5999886	NEWENTRY	-	-	-	-	-	Record to support submission of GeneRIFs for a g	other	-	-	-	-	20
38	34	4126706	NEWENTRY	-	-	-	-	-	Record to support submission of GeneRIFs for a g	other	-	-	-	-	20
39	56	5814795	NEWENTRY	-	-	-	-	-	Record to support submission of GeneRIFs for a g	other	-	-	-	-	20

Normal View Ready Sum=7

Back in the MySQL GUI, Open a new SQL query tab.



Schema Editor

Specify the name of the schema here. You can use any combination of ANSI letters, numbers and the underscore character for names that don't require quoting. For more flexibility you can use the entire Unicode Basic Multilingual Pane (BMP), but you will have to quote the name later when you refer to it.

Schema Name: genomics

The character set and its collation selected here will be used when no other charset/collation is set for a database object (it uses the DEFAULT value then). Setting DEFAULT here will make the schema to use the server defaults.

Character Set: Default Charset

Collation: Default Collation

	Time	Action	Response	Duration / Fetch Time
1	13:40:04	Apply changes to genomics	Changes applied	

Create a database table to accommodate the data from *gene_info*. Creating a database table with the correct data types and sizes for fields is often a trial and error process.

[major step] You can paste the SQL below directly in the SQL Query window and execute it, or you can use the *Add Table* wizard under your *genomics* MySQL schema tab, and enter each field individually. *Tip: Watch out for special characters from Word. Especially quotes.*

```
create table gene_info (
tax_id int,
GeneID int,
Symbol varchar(48),
LocusTag varchar(48),
Synonyms varchar(1000),
dbXrefs varchar(512),
chromosome varchar(48),
map_location varchar(48),
description varchar(4000),
type_of_gene varchar(12),
Symbol_from_nomenclature_authority varchar(48),
Full_name_from_nomenclature_authority varchar(256),
Nomenclature_status varchar(12),
Other_designations varchar(12),
Modification_date varchar(12));
```

To execute your SQL, make sure the genomics database is selected in the Object Browser (left-side panel in Workbench should be **bold**).

Next, select the entire text of the query, and execute the query by selecting the lightning bolt to the left (execute ALL or selected), *or* just click (select) anywhere within your query text and select the lightning bolt to the right with the big “I” superimposed on it (execute current query).

You can also execute either type of query form the menu under SQL.

You should get the following message:
Create table gene_info(….) 0 rows affected.

You can see the table now in the object browser under your genomics->Tables->gene_info but you may have to right click and refresh the Tables to have it show up.

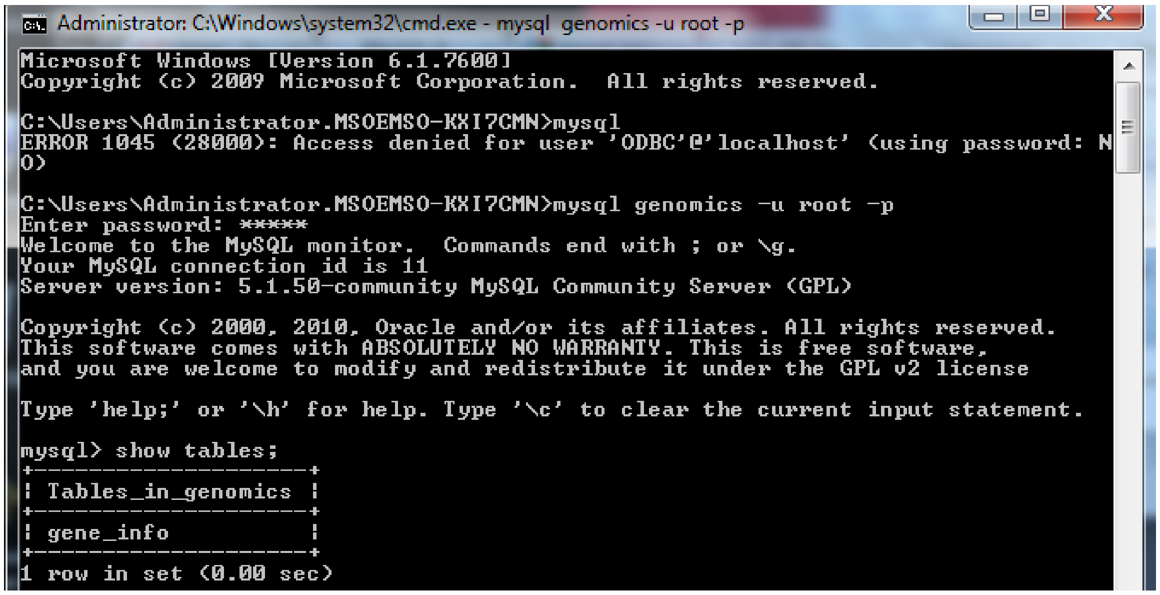
MySQL reference for create table:
http://dev.mysql.com/doc/refman/8.0/en/show-create-table.html

Next perform the *load data* command using the command line SQL query window as follows. I recommend using the command line since workbench can be too picky with file imports.

Command line: Open up a *command line* connection to mySQL as described previously.

And enter your password at the prompt.

Type *show tables* to verify you are connected to the correct database, etc.


Administrator: C:\Windows\system32\cmd.exe - mysql genomics -u root -p

```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\Administrator.MSOEMSO-KXI7CMN>mysql
ERROR 1045 (28000): Access denied for user 'ODBC'@'localhost' (using password: N
O)

C:\Users\Administrator.MSOEMSO-KXI7CMN>mysql genomics -u root -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 11
Server version: 5.1.50-community MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show tables;
+---------------------+
| Tables_in_genomics  |
+---------------------+
| gene_info           |
+---------------------+
1 row in set (0.00 sec)
```

Execute:

```
SELECT @@GLOBAL.`local_infile`;
```

Make sure it returns a 1.

If not, enter the following:
SET GLOBAL local_infile=1

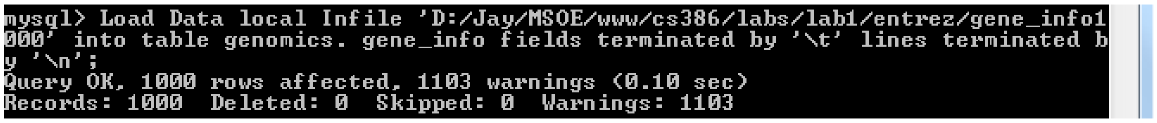
[major step] Enter the following *load data local infile* command – you have to change it to the location of the file on your system (no carriage returns, unix-style slashes!):

```
Load Data local Infile
'D:/Dropbox/MSOE/www/cs386/labs/lab1/entrez/gene_info1000'
into table genomics.gene_info fields terminated by '\t' lines
terminated by '\n';
```

If you get "ERROR 2068 (HY000): LOAD DATA LOCAL INFILE file request rejected due to restrictions on access." Then type exit on the command prompt.
Reopen mysql command prompt with following (so not via the link on the start menu):
`mysql -u root -p --local-infile=1` and try the load data command again after running "use genomics;"

Note: Always check the data file for column delimiters and end-of-line character(s).

- *In Unix and all Unix-like systems (OS-X), \n is the code for end-of-line, \r means nothing special.*
- *In Windows (and some old OS's), the code for end of line is 2 characters, \r\n, in this order (CR, LF)*



```
mysql> Load Data local Infile 'D:/Jay/MSOE/www/cs386/labs/lab1/entrez/gene_info1
000' into table genomics. gene_info fields terminated by '\t' lines terminated b
y '\n';
Query OK, 1000 rows affected, 1103 warnings (0.10 sec)
Records: 1000  Deleted: 0  Skipped: 0  Warnings: 1103
```

Note: the large number of warnings! In MySQL workbench, you hover over the message column with your cursor in the output window towards the bottom.

Note: gene_info1000 is the first 1000 rows of the Entrez gene database table. We will import the whole table after we fix a few problems first.

Note: I deleted the header block from the top of the text file.

MySQL reference for load data: http://dev.mysql.com/doc/refman/8.0/en/load-data.html

[major step] In your MySQL workbench query window (or *cmd* window) execute:

```
show warnings;
```

You should see several error messages as follows:

```
Data truncated for column 'type_of_gene' at row 2
Data truncated for column 'type_of_gene' at row 3
…
Data truncated for column 'Other_designations' at row 11
…
```

This indicates that we did not size the *type_of_gene* and *Other_designations* database fields correctly. This may not be important if you do not care about the data in these fields, but let's assume you want to import the entire file correctly.

From the command line, or your MySQL Workbench SQL window in the GUI:

First, drop (delete) the current gene_info database table:

```
drop table gene_info;
```

[major step] Create a new table with larger fields for *type_of_gene* and *Other_designations*. We see from inspecting the file, that neither of these entries is very commonly supplied. A typical entry for *type_of_gene* is *'protein coding'*, and *Other_designations* may have an entry like '*putative F plasmid gene 32-like protein*'.

```
create table gene_info (
tax_id int,
GeneID int,
Symbol varchar(48),
LocusTag varchar(48),
Synonyms varchar(1000),
dbXrefs varchar(512),
chromosome varchar(48),
map_location varchar(48),
description varchar(4000),
type_of_gene varchar(24),
Symbol_from_nomenclature_authority varchar(48),
Full_name_from_nomenclature_authority varchar(256),
Nomenclature_status varchar(12),
Other_designations varchar(4000),
Modification_date varchar(12));
```

[major step] Reload the entire full length Entrez gene data set (not just the 1000 record version, and be a little patient this can take a couple minutes) using load data local

```
Load Data local infile
'D:/Dropbox/MSOE/www/cs386/labs/lab1/entrez/gene_info' into table
gene_info fields terminated by '\t' lines terminated by '\n' IGNORE 1
LINES;
```

```
Note: I added the "IGNORE 1 LINES" option to skip the first line on
when loading the whole file. For the abbreviated file gene_info1000, I
had removed the column header line. In the original file, the header is
still there and we need to deal with it.
```

[major step] Check for warnings and verify the length of the database table. You may still see some warnings about rows not having all columns, which you can ignore.

```
select count(*) from gene_info;
4354758
```

Creating our working table [major steps – entire section, but feel free to abbreviate results shown in report]

Issue the following select statements, limiting the result set to the first 100 records, and inspect the table.

```
select * from gene_info
limit 100;
```

MySQL reference for select: http://dev.mysql.com/doc/refman/8.0/en/select.html

We can see that several fields including *dbXrefs*, *chromosome*, *map_location*, and others are typically blank, and may not be needed for our experiments. So let's create a new table with just the information we need.

```
create table gene_info2 (
tax_id int,
GeneID int,
Symbol varchar(48),
LocusTag varchar(48),
Synonyms varchar(1000),
description varchar(4000),
type_of_gene varchar(24),
Modification_date varchar(12));
```

We can use the SQL *insert into* statement to populate our new table from our existing table (takes a couple minutes to run) :

```
insert into gene_info2
select tax_id, GeneID, Symbol, LocusTag, Synonyms, description,
type_of_gene, Modification_date
from gene_info
where GeneID is not null;
```

Note: We are not including rows where there is no GeneID with the statement:
`where GeneID is not null;`
Without being able to uniquely identify a record, its useless.

MySQL reference to insert:
http://dev.mysql.com/doc/refman/8.0/en/insert.html

We no longer need the original *gene_info* table we used for importing gene data, so we can drop it.

```
drop table gene_info;
```

Note: To delete all rows from a table without removing a table:
`delete from gene_info;`
Drop and create table is generally much faster than delete for large tables.

Primary keys and indexes [major steps – entire section, but feel free to abbreviate results shown in report]

Issue the following select statement to retrieve the record for a specific gene, and record the time:

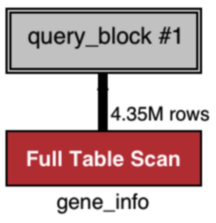
```
select * from gene_info2 where GeneID=4226999;
```

```
303, 4226999, 'pWW53_43', 'pWW53_43', '-', 'putative reductase
protein', 'protein-coding', '20080514'
```

```
1 row fetch (12.4951s)
```

This next step should be done in the GUI, even if you have been using the command line! **Re-execute the command with the explain option by selecting the *Lighting Bolt* with the *magnifying glass* from the Toolbar.** Explain will display how the table is accessed. You may need to click on the various GUI buttons arraigned vertically on the right side of the results to see this figure and text (view source). ***Record the result.***

```
explain select * from gene_info2 where GeneID=4226999;
```



```
  "query_block": {
    "select_id": 1,
    "table": {
      "table_name": "gene_info",
      "access_type": "ALL",
      "rows": 4354758,
      "filtered": 100,
      "attached_condition":
"(`genomics`.`gene_info`.`GeneID` =
4226999)"
    }
  }
}
```

```
Full table scan. I.e., we are scanning every record in the database.
```

This is very *slow* for retrieving a record. Also, we want to ensure that our table is in first normal form. At a minimum, this means we need to be able to address each record/row of a database table uniquely using a single field or some combination of fields. We do this by defining the *primary key* for the table, which is GeneID. When you define a primary key, the DBMS automatically creates an index on this key.

An index reduces search time by creating a tree-like or hash data structure to access rows in a database table. Importing data into a table goes much faster if you define the *primary key* and other indexes *after* the data has been imported. Otherwise the indexes have to be verified and updated with each row insert.

```
alter table gene_info2 add constraint pk_gene_info primary key(GeneID);
```

```
select * from gene_info2 where GeneID=4226999;
```

```
303, 4226999, 'pWW53_43', 'pWW53_43', '-', 'putative reductase
protein', 'protein-coding', '20080514'

1 row fetch (0.0004s)
```

That's more like it. *Note: You can create primary keys and indexes as a composite of multiple fields as well.*

We may also want to create indexes on other fields in the database. For example we may be interested in identifying all genes within a specific taxonomy. So, let's create an index on *tax_id*. Note: the *tax_id* is not distinct.

```
create index tax_id_idx on gene_info2(tax_id);
```

Next let's import the gene2pubmed data. For large data imports, you may get a time-out error in the GUI, which is why we use the command line to import data. The gene2pubmed table allows us to cross-reference Pubmed articles. Format: tax_id, GeneID, PubMed_ID

```
create table gene2pubmed (
tax_id int,
GeneID int,
PubMed_ID int);

Load Data local Infile '/data/gene2pubmed' into table
genomics.Gene2pubmed fields terminated by '\t' lines terminated by
'\n';
```

ERD
Create an ERD diagram (We learn about this in the 2nd lecture) of gene_info2, gene2pubmed and the following Pubmed table (we are not going to populate it right now):

```
Create table Pubmed (
Pubmed_ID int,
Text varchar(4000));
```

You can use *desc* to display each table's data types as follows:

```
desc gene2pubmed;
```

https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model

Deliverables
Complete a lab report containing the following:

- What you've learned from the lab
- How it could be improved
- Observations and query results for each major step of the lab.
- ERD
- Submit to Canvas in PDF format!