

EECS116/CS122A

Winter 2020

Mini-Project - MySQL

Acknowledgment: The project was originally designed by Professor Michael Carey with slight modifications tailored for MySQL.

In this project, we are going to create a database and tables using MySQL, and import the data into tables. Then you are required to form SQL queries for the problem statements shown in step 5. Due by 5PM on Thursday 03/05. Turn in the 2 deliverables to the folder “Mini Project” on Dropbox.

Deliverables

1. An SQL script that contains the queries listed in the same order as shown in step 5.
2. .csv files that contain the results obtained.

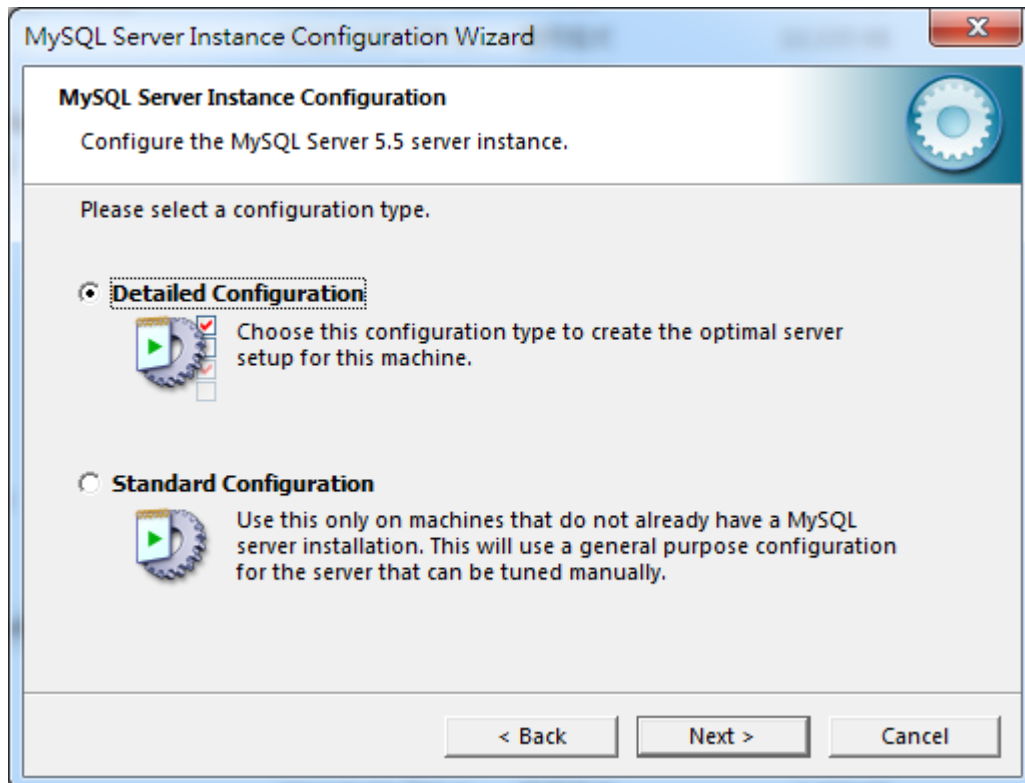
STEP 1 - Install MySQL

Go to MySQL download page (<http://dev.mysql.com/downloads/>) to download MySQL Community Server. Select the correct installer based on your system.

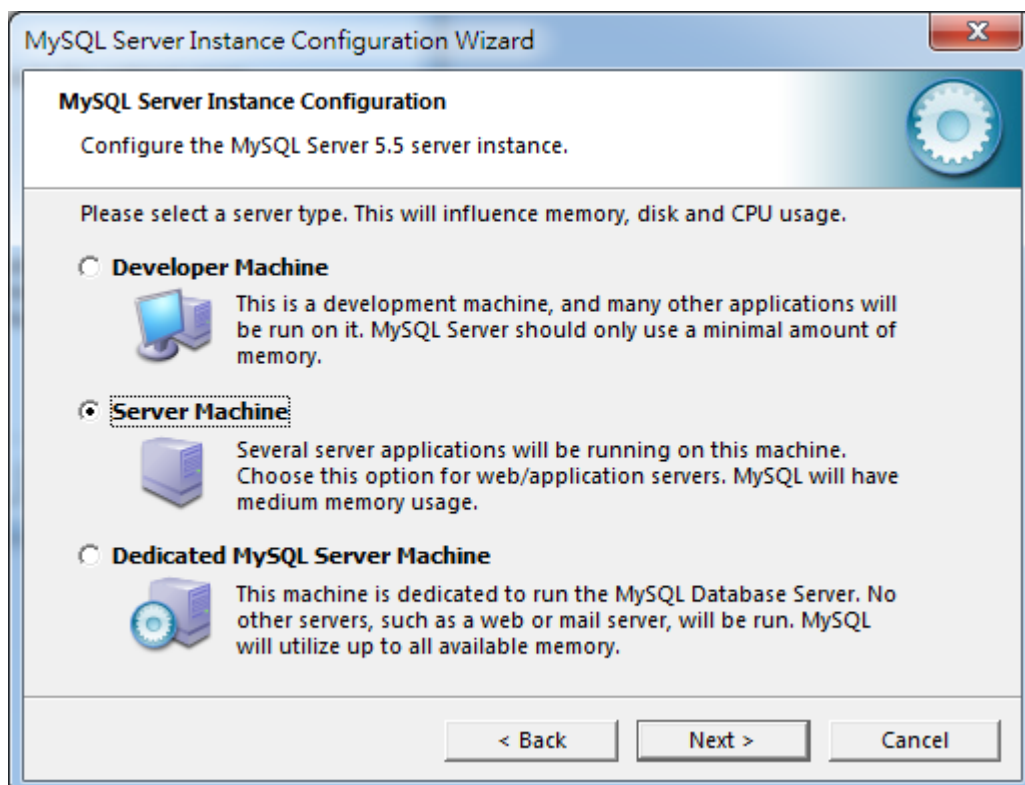
For Mac OS X, download the DMG files, extract it, and install by the order: mysql-5.5.xx-osx10.6-x86_64.pkg, MySQLStartupItem.pkg, MySQL.prePane

The following description is based on MySQL server 5.5 for Windows .

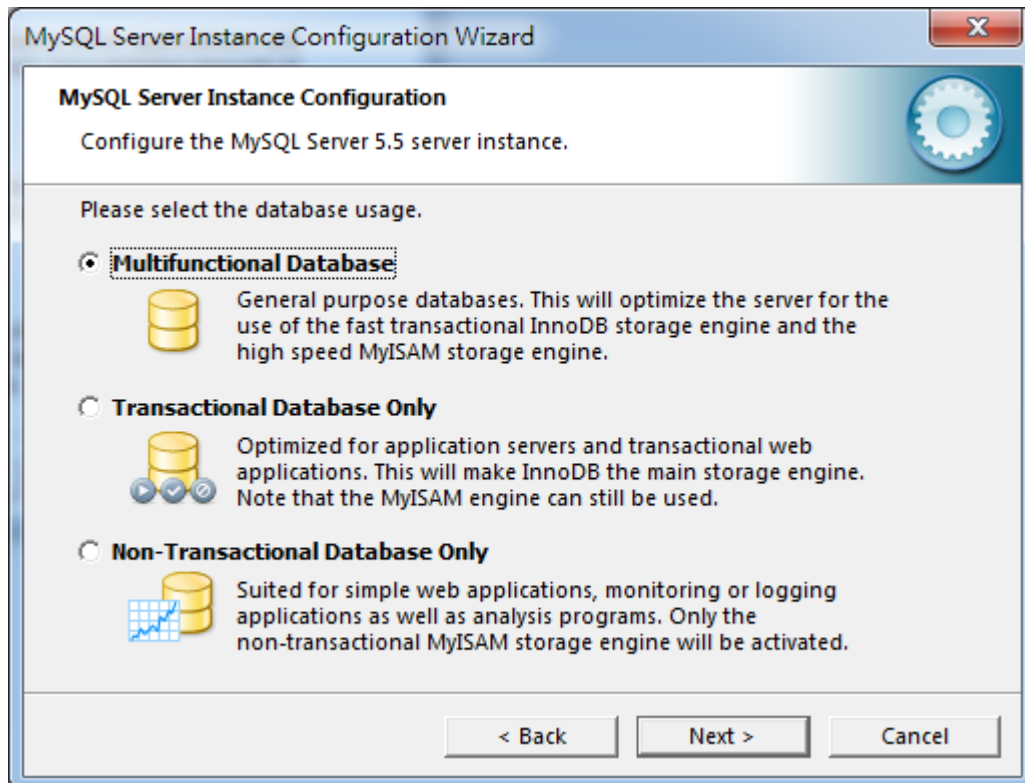
1. Execute the downloaded file and complete the installation, then it will ask you to set up the configuration file.



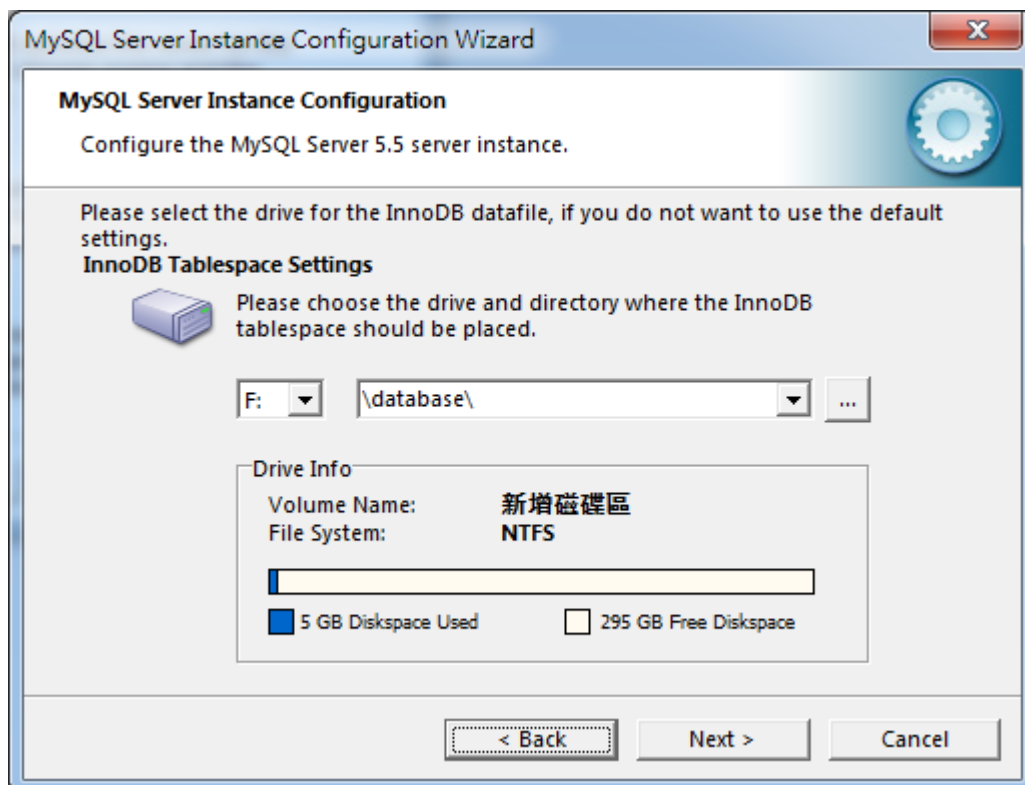
2. Choose a type of machine; you can select developer or server.



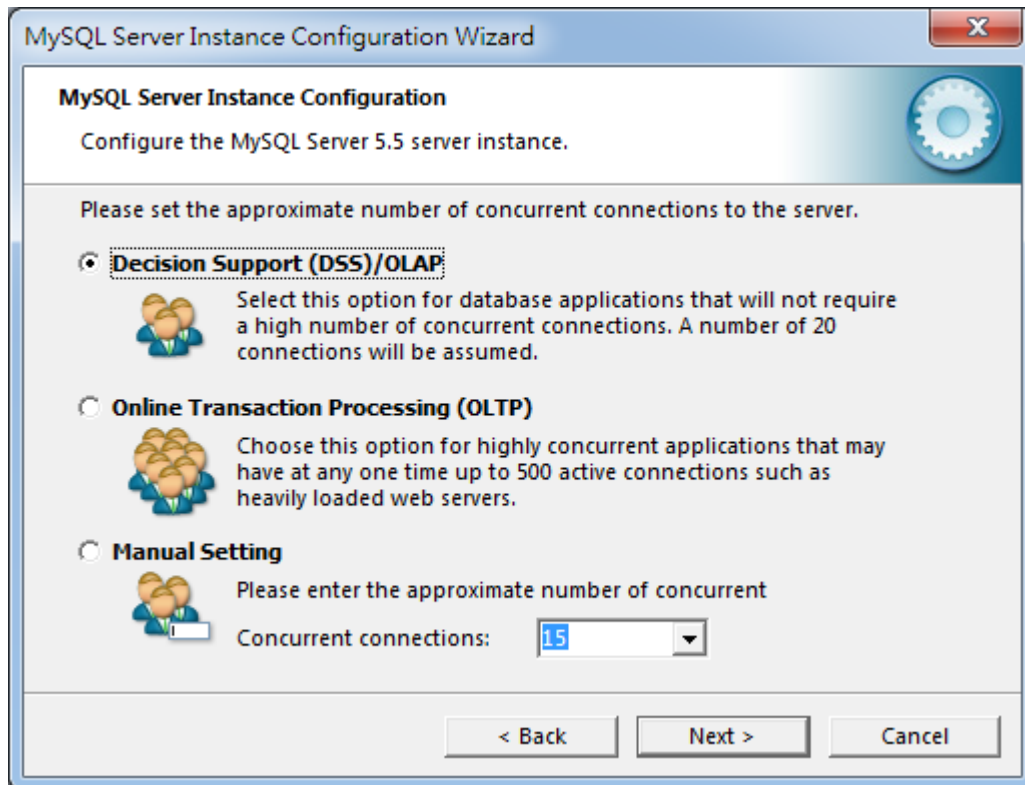
3. Choose "Multifunction database".



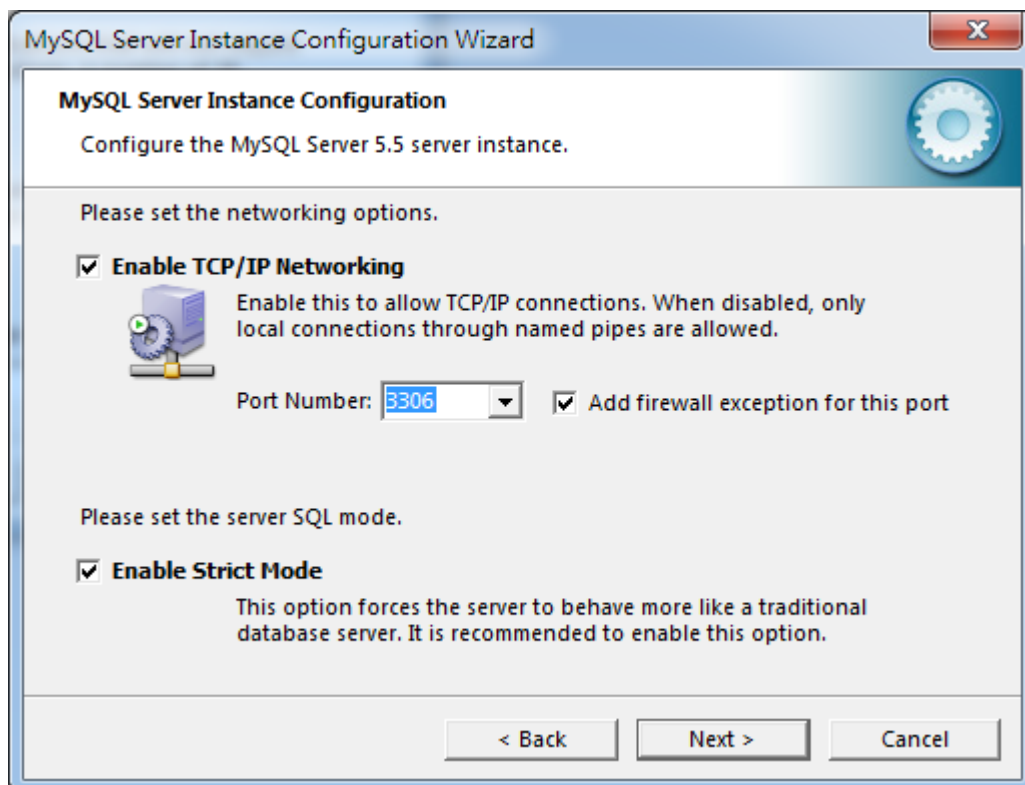
- Choose where you are going to save database data.



- Choose "Decision Support(DSS)/OLAP".

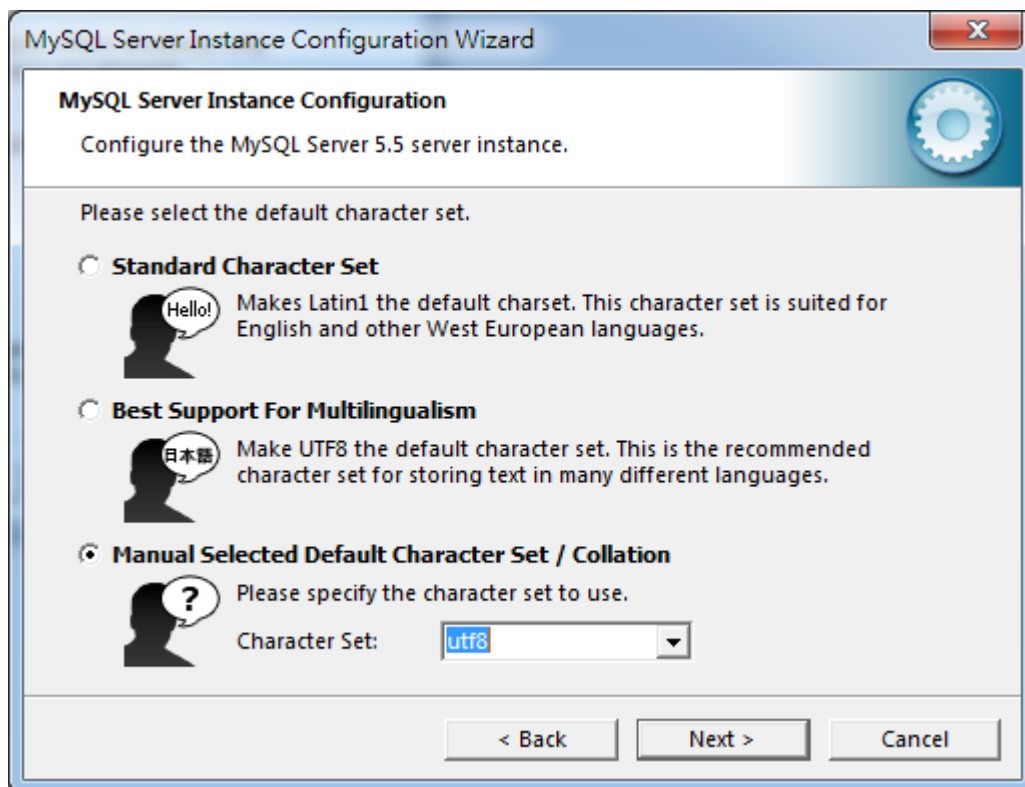


6. Continue (The default port is 3306 unless you want to change it)

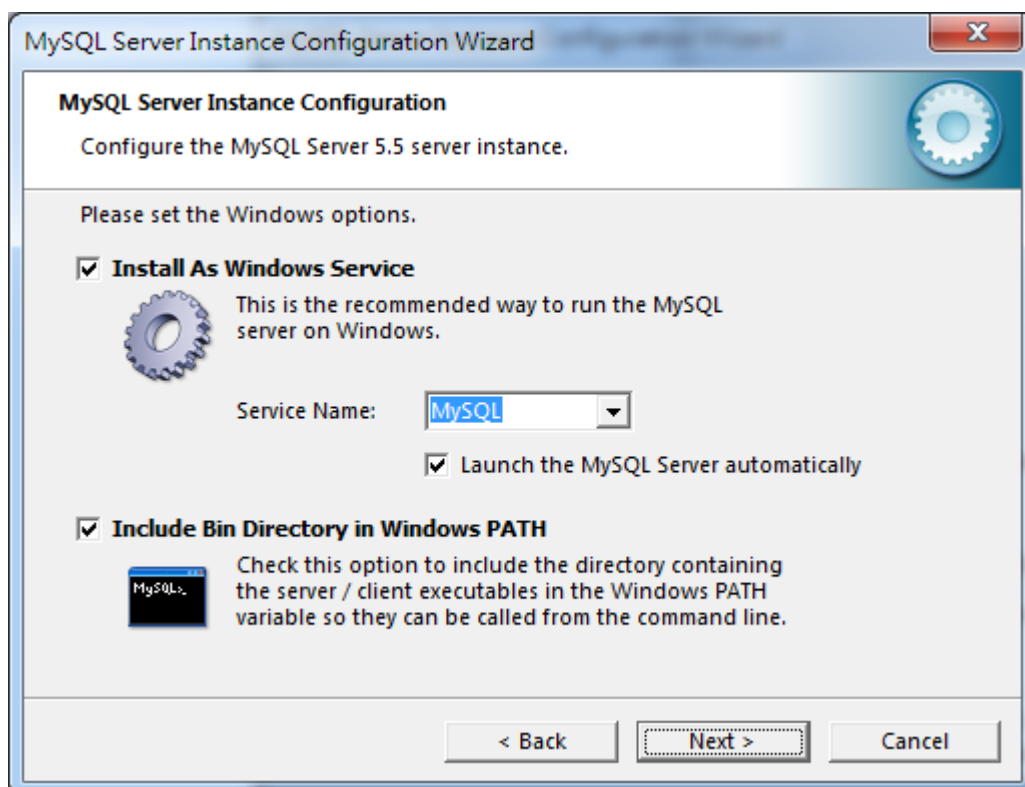


7. Choose "utf8" for char set (second choice or third choice), this will support

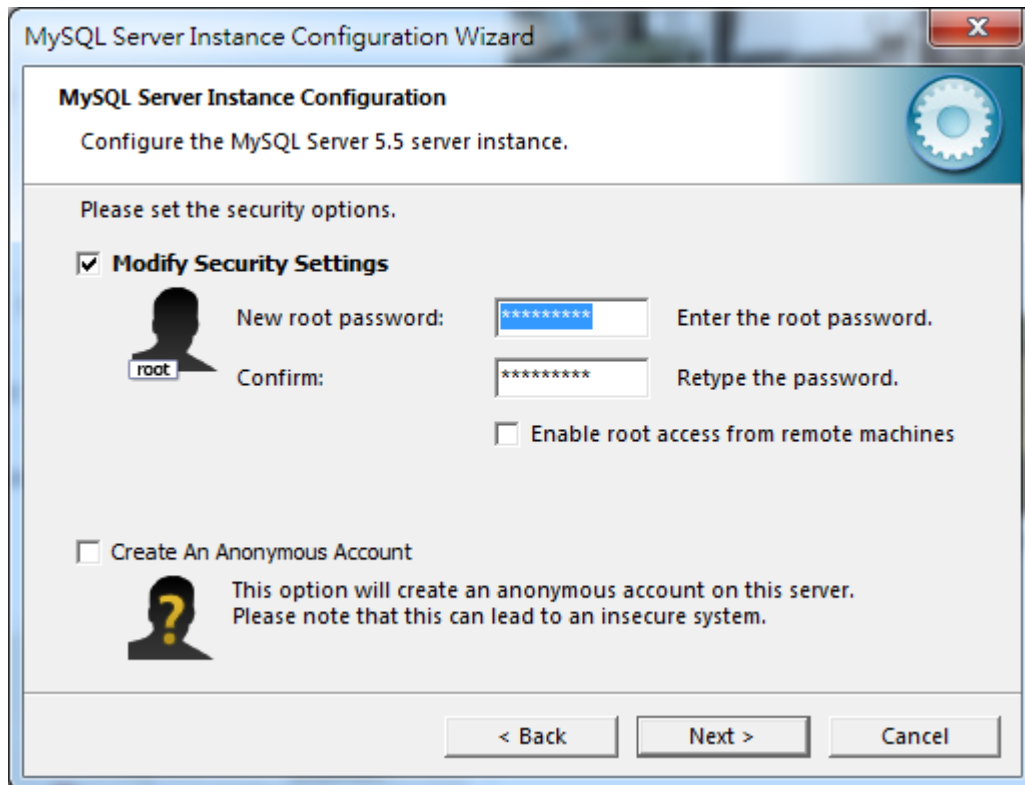
multi-language



8. Add service and set environmental variables.

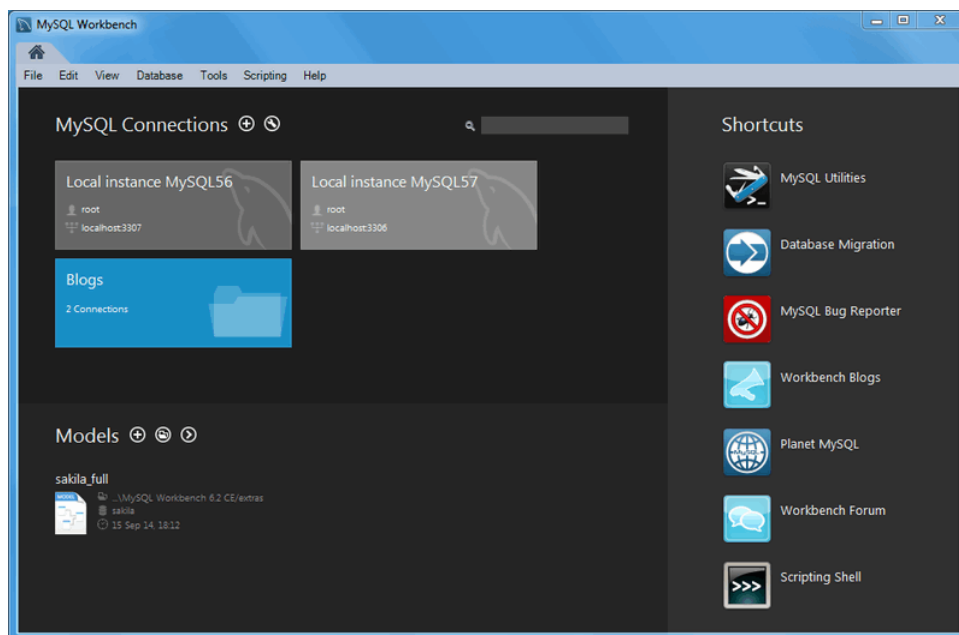


9. Set the root password, you can execute and command using the account "root".



STEP 2 - A GUI Tool for MySQL – MySQL Workbench

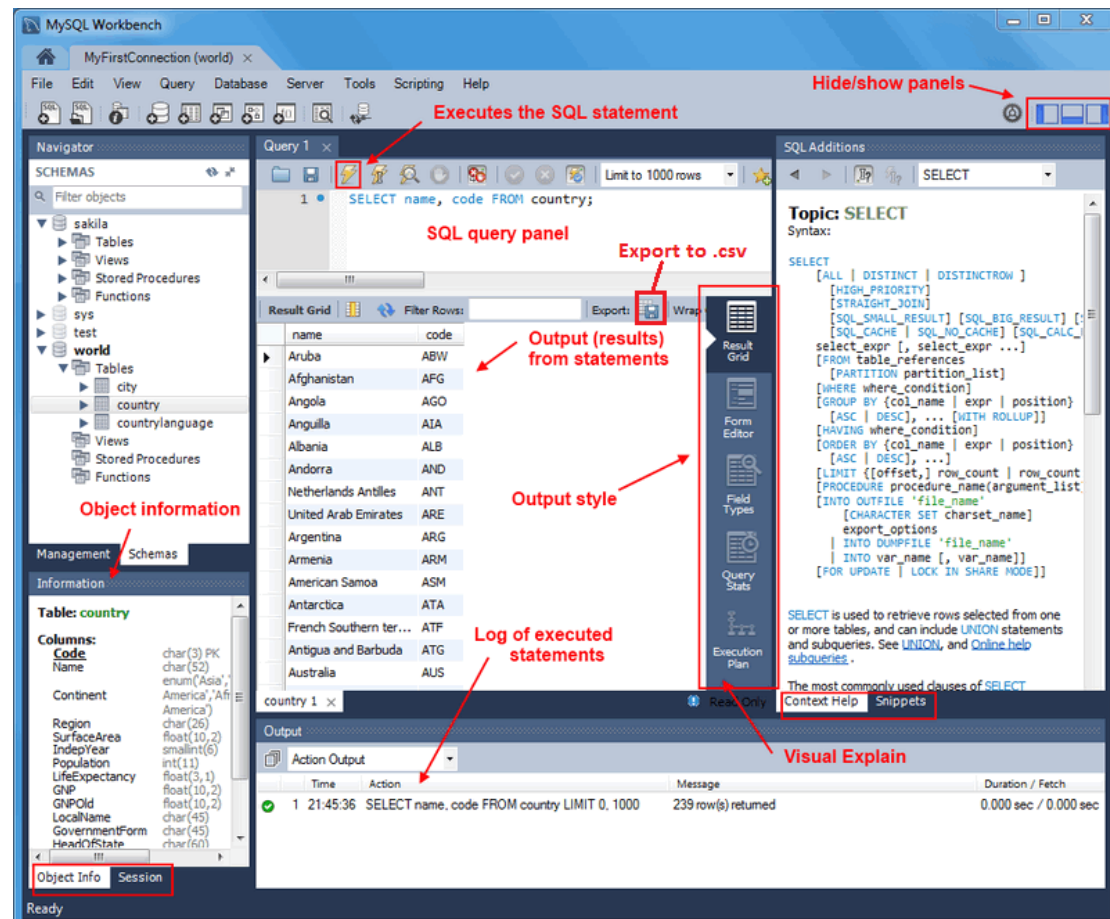
Download the MySQL Workbench from MySQL download page. After the installation, run it and you should see the following:



Double click the local instance to connect. Note that the port should be the same as when you installed MySQL (3306 in the case). If not, edit the connection by:

Windows and Linux: hover over the right side of a connection title and click the title.
OS X: hover over a connection title and click the little (i) in appears in the bottom right corner

Type in your password, and connect to the database. Now you can run queries or manipulate your database:



STEP 3 - Create Database and Tables

Reference tutorial for creating database using MySQL Workbench:

<https://www.youtube.com/watch?v=JyRWDQSQYNw>

Given below is the schema for the data. There are a total of 12 tables, thus 12 CSV files, each corresponding to a relational table.

user (email, password, name, date_of_birth, address, type)
 primary key(email)

celebrity (email, website, kind)
 primary key(email)

blurt (blurtid, email, text, location, time)
primary key(blurtid,email)
foreign key(email) references user(email)

hobby (email, hobby)
primary key(email,hobby)
foreign key(email) references user(email))

follow (follower, followee)
primary key(follower, followee)
foreign key(follower) references user(email)
foreign key(followee) references user(email))

vendor (id, name)
primary key(id)

vendor_ambassador (vendorid, email)
primary key(vendorid)
foreign key(email) references user(email)
foreign key(vendorid) references vendor(id))

topic (id, description)
primary key(id)

vendor_topics (vendorid, topicid)
primary key(vendorid, topicid)
foreign key(vendorid) references vendor(id)
foreign key(topicid) references topic(id))

blurt_analysis (email, blurtid, topicid, confidence, sentiment)
primary key(email, blurtid, topicid)
foreign key(email, blurtid) references blurt(email, blurtid)
foreign key(topicid) references topic(id)
constraint confidence >= 0 and confidence <=10
constraint sentiment >= -5 and sentiment <=5

advertisement (id, content, vendorid)

primary key(id)
foreign key(vendorid) references vendor(id))

user_ad (email,adid)
primary key(email,adid)
foreign key(email) references user(email)
foreign key(adid) references advertisement(id))

The design model:

Users can post their thoughts in form of short messages that we call “blurts”. When signing up, users need to provide their email and a password of their choice. In addition, they need to enter some basic information – name, date of birth, address, email ID and hobbies. Once signed up, they can (besides blurring) “follow” other users. To “follow” a user means subscribing to his/her “blurts”. Users are categorized into “regular” users and “celebrities”. A celebrity has an associated website url and an attribute called “kind” indicating whether he is a politician, actor, singer, etc. Each blurt by a user (regular or celebrity) is assigned an id. Blurt ids are serial and unique to the “blurts” by a given user; the first blurt by a given user would have blurt id 1 and ids are incremented for each successive blurt by the user. Note that blurt ids are unique only to a user, so blurts by two different users may have the same blurt id. Besides an id, each blurt also has its text, timestamp, and user location as additional attributes. The system should have a pre-defined notion of “topics” that are simply subjects that people may blurt about. Examples of topics might include music, pollution, disease, disaster, sports, weather, etc. A topic has a unique id and description (the name of the topic). Each blurt by a user is analyzed to associate with it zero or more topics. Related blurt-topic pairs are stored in blurt_analysis table. To account for the possible ambiguity arising from the choice of words or language used by a user, an association with a topic has a corresponding confidence level (an integer ranging from 1 – 10 indicating the strength of the association). For example consider the following blurt: “I absolutely hate the rainy weather, can’t go out, listening to the Beatles, just love them” is analyzed to be associated with two topics, weather and music (Beatles). For each topic, the associated sentiment is evaluated and quantified as an integral value ranging between -5 and 5, with higher values indicating a more positive sentiment. Considering the example blurt used above, the topic weather would have an associated sentiment of -5 (hate) while for music the corresponding value is 4 (love). **Note: You don’t need to implement the value constraint as MySQL doesn’t support it.** A vendor has interest in one or more topics and is

interested in tracking all users who are blurting about a topic of interest. A vendor may also have a celebrity as its brand ambassador. Vendors create advertisements that have an associated unique id and a textual content. These advertisements are stored in the system and are available to be shown to the regular set of users (that is, not to the celebrities, just to the other “regular” users). Careful matching is done based upon a historical analysis of all blurts by a user. Based upon the analysis, a user may be shown zero or more advertisements.

STEP 4 – Import CSV files

Script Template:

```
LOAD DATA LOCAL INFILE "[CSV file name]" INTO TABLE [table name]
COLUMNS TERMINATED BY ',' LINES TERMINATED BY '\n'
```

For each CSV file, replace [CSV file name] and [table name] with actual CSV file name and corresponding table name, e.g.:

```
LOAD DATA LOCAL INFILE "d:\\csvdata\\advertisement.csv" INTO TABLE
advertisement COLUMNS TERMINATED BY ',' LINES TERMINATED BY '\n'
```

Execute 12 scripts using the GUI client

STEP 5 - Form SQL Queries

For the following statements, you are required to form SQL queries and execute them using the GUI client. Then export the result using the name "Query x.csv", x being the label of each query. Put all the SQL you formed into a file named "Script.txt" in the same order. Then archive the file as "mp-xxxxxxx.zip", xxxxxxxx being your student id, and turn it in on eee dropbox under folder “mini project”. The filenames of your result has to follow the instructions exactly or you may get a deduction in your credit.

1. For each topic, find the total number of blurts that were analyzed as being related to the topic. Order the result by topic id. Your SQL query should print the topic id, topic description and the corresponding count.
2. For each celebrity user, find the total number of followers. Your SQL query should print the name of the celebrity and the corresponding number of followers.
3. For each celebrity, find the number of blurts. Order the result in decreasing order of the number of blurts. Your query should print the name of the celebrity and the associated count in decreasing order of the count.
4. Write an SQL query to print names of all celebrities who are not following

anyone!

5. Write an SQL query that gives the email of its brand ambassador and the number users who are following the brand ambassador for each vendor. Your SQL query should print the vendor name, email and the total number of users who are following it.
6. Let us define the term "advertisement-gap" as the number of users who have blurted about a topic that is of interest to a vendor but are not being shown in any advertisements from the vendor. Write an SQL query that gives the vendor name and the corresponding "advertisement-gap" in decreasing order of the advertisement_gap.
7. Write an SQL query to find all pairs of users (A,B) such that both A and B have blurted on a common topic but A is not following B. Your query should print the names of A and B in that order.
8. You need to help users connect with other users. There could be there different users A,B and C such that A follows B, B follows C but A does not follow C. Write an SQL query to find all such triplets of A,B, and C. Your query should print the emails of users A,B and C in that order.
9. For each topic, find the states (e.g., California) where “the average sentiment associated with the blurts related to the topic” is negative. Your query should print the topic id, topic name, state, total # of blurts and average sentiment for each topic.