SQL workshop

Sepideh Mosaferi

smosafer@umd.edu, University of Maryland, College Park

December 14, 2016

What is SQL?

SQL stands for Structured Query Language. You may hear it pronounced in the same way as the word "sequel" or as S-Q-L. SQL is the language used to interact with a relational database. A "SQL database" is any database one can interact with via SQL.

Common SQL database management systems include mySQL and postgreSQL, but SQL also underlies the most common of database software - Microsoft Access.

SQL statements can be used to create tables, delete tables, add rows and columns to tables, and query tables.

Why we should learn SQL?

There afre many reasons that we recommend you to learn SQL. Among them, the followings might be more conspicuous:

- SQL is one of the most sought-after skills by hiring employers.
- You can get an answer for any question that you might have about the dataset.
- You no longer have to deal with Excel crashing.
- You won't ever have to ask yourself, "How did I make that report again?".
- You can incorporate other programming codes with it such as SAS, C++, JAVA, etc.

How to Install SQL?

In order to install SQL, you could easily type https://www.mysql.com in the browser. Then click on the DOWNLOADS section. MySQL Community Server under http://dev.mysql.com/downloads/mysql/ is the right one since we need SQL package as a whole. Currently the available version is **5.7**. You could select your specific platform and then grab your interested version. We recommend **All MySQL Products. For All Windows Platforms. In One Package.** for the sake of convinience as it has all the necessary items in just one package.

After download, you can install the progarm. The installation takes time around 5 minutes, but it should be handy! For last products and options similar to these, click execute, and you can select default for other options. We also recommend to install all products such as "MySQL Server", "MySQL Workbench", "Conector/C++", etc.

After installation and configuration, the most important part is setting a password for the MySQL server as you can see in Figure 1. Then you are done. So, you can see double click the program and see the workbench environment which is look like Figure 2.

Figure 1: Setting A Password

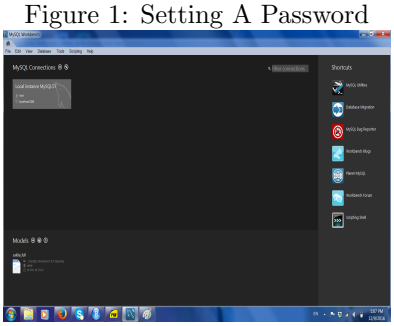


Figure 2: Workbench Environment

How to Create Database?

From the toolbar go to the Database and click on the Connect to Database. Then you can store the password. Then SQL work environment be oppend, see Figure 3.

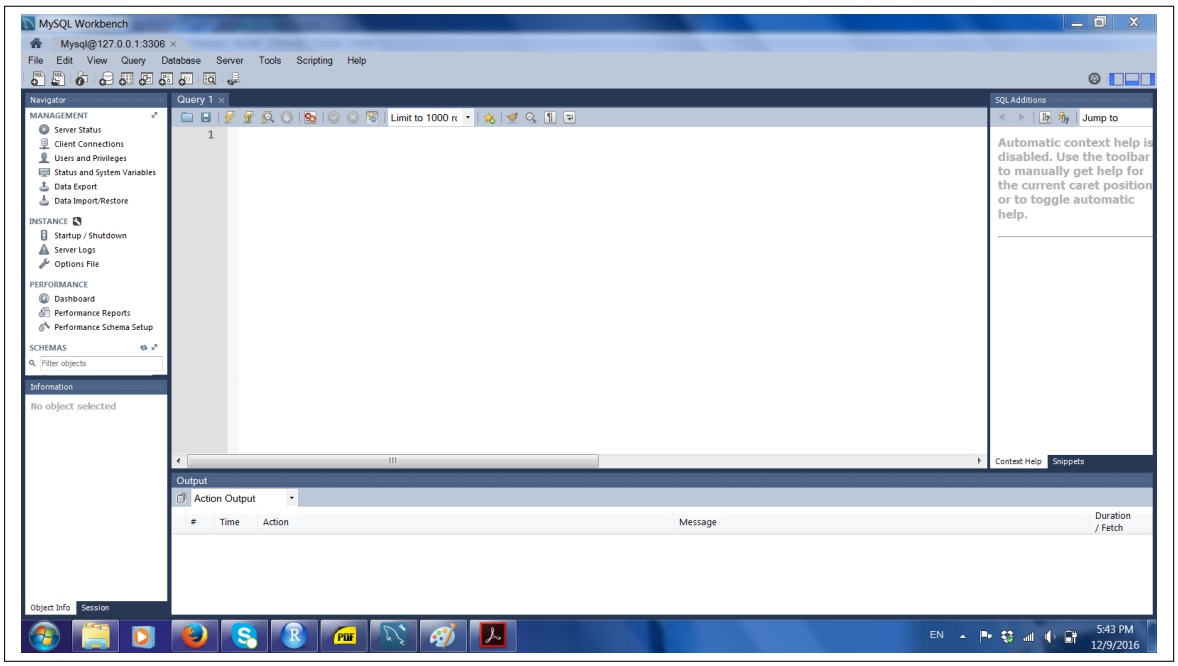
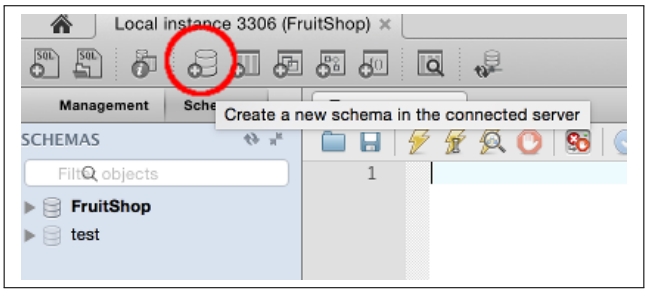


Figure 3: Main Environment

Create Our Own Database

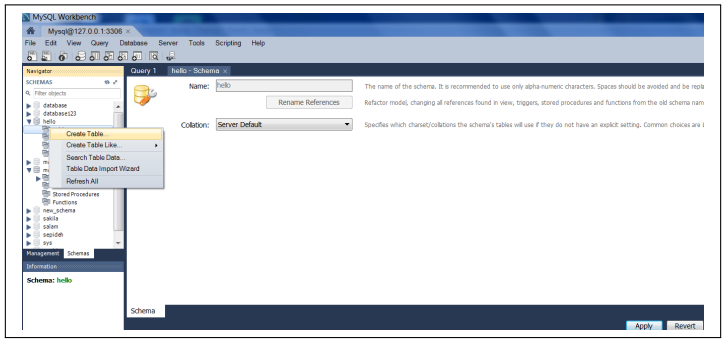
In order to automatically make a new schema, which is a new database, we can select the following icon from the toolbar. Then you can give a name to the new schema and click apply.



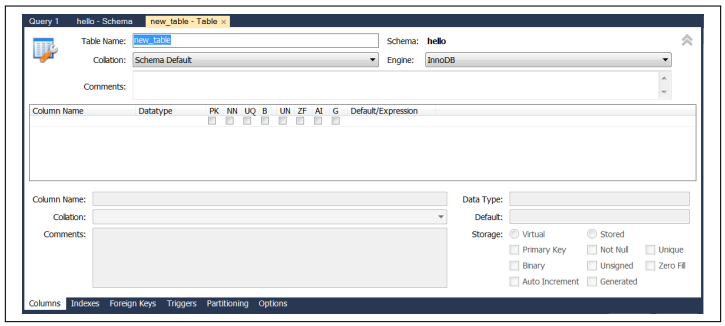
After that, it can also produce the query [1] related code. As an eample if we would like to produce a schema called "hello", the following query is produced.

```
CREATE SCHEMA 'hello' ;
```

After producing the schema, we are able to produce Tables by rigt clicking on the Table icon under the produced schema and click on the *Create Table* as you can see in the following Figure.



Then the following page comes that we are able to enter the information of table including the name of tables and corresponding columns.



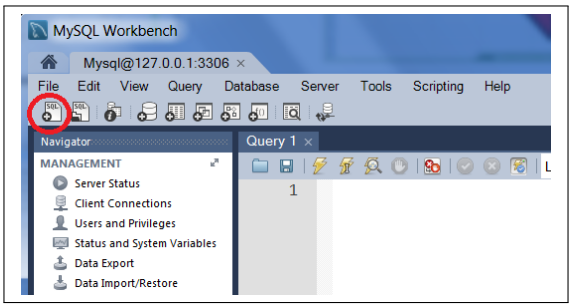
Now, assume that we have simply created the following table called *employee-info* under the database *database123*. We can do any changes that we are interested in manually. In the next part, we show how to use query for entering codes and doing changes.

[1] Query coding will be explained later.

idEmployeeInfo	name	surname	age
1	Sepideh	Mosaferi	20
2	Sahar	Asghari	54
3	Josh	Langled	22
4	Mark	Ventura	35
6	Mark	Bill	34
7	Mark	Gill	29
8	Ying	Wang	32
9	Will	Howard	35
10	Sharen	Eddy	24
NULL	NULL	NULL	NULL

Query and Tables

Query is an environment in the SQL that you are able to enter codes for tasks that you are interested in. Here we have given the icon shape in the SQL.



After writing the commands, we should execute the code by clicking on the thunder shape from the toolbar. In the following, we provide some examples:

Example 1 (Creating A Database):

```
CREATE SCHEMA ‘hello’ ;
```

Example 2 (How to Create A Table):

```
CREATE TABLE ‘Data’.‘stundentsscores’ (
  ‘ID’ INT NOT NULL,
  ‘Name’ VARCHAR(45) NULL,
  ‘Surname’ VARCHAR(45) NULL,
  ‘Score’ INT NULL,
  PRIMARY KEY (‘ID’));
```

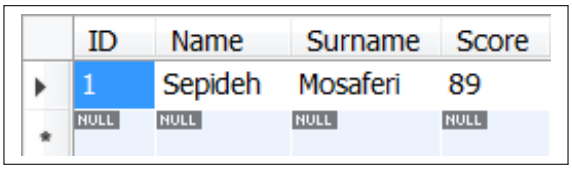
Example 3 (How to Add A New Col):

In order to add a new column to the created table, we can replicate the following code in the query:

```
INSERT INTO ‘Data’.‘stundentsscores’ (‘ID’, ‘Name’, ‘Surname’,
```

```
'Score') VALUES ('1', 'Sepideh', 'Mosaferi', '89');
```

The results of the table is the following:



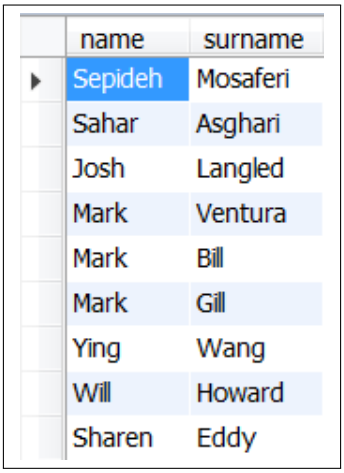
ID	Name	Surname	Score
1	Sepideh	Mosaferi	89
NULL	NULL	NULL	NULL

Hint Pay attention that for the characteristics variables such as name, surname, we need to put values in a quotation. which is not necessary for the integer values.

Example 4 (How to Select Specific Cols):

In this example, we can write the following query code to select name and surname from the existed table called *employeeinfo* under the database *database123*.

```
SELECT name,surname from database123.employeeinfo;
```



name	surname
Sepideh	Mosaferi
Sahar	Asghari
Josh	Langled
Mark	Ventura
Mark	Bill
Mark	Gill
Ying	Wang
Will	Howard
Sharen	Eddy

Example 5 (Select Whole Database):

If we want to select the whole database, we need to type * such as the following code.

```
SELECT * FROM database123.employeeinfo;
```

Example 6 (Select Specific Row):

In order to select specific row from the table based on ID, we can type the following query:

```
select * from database123.employeeinfo where idEmployeeInfo='10';
```

If we want to select more than 1 row (such as rows with ids smaller than 5), we can type the following query:

```
select * from database123.employeeinfo where idEmployeeInfo<=5;
```

Example 7 (Find Specific Employee):
In order to find a specific employee with certain characteristics, we can type the following query:

```
update database123.employeeinfo set name='Sarah', age='20' where
idEmployeeInfo='1';
```

Example 8 (Sorting Table):
In order to sort a table based on the family names in the descending or ascending order (as an e.g.), we can have the following query:

```
select * from database123.employeeinfo order by surname desc;
select * from database123.employeeinfo order by surname asc;
```

Example 9 (Update Info) *"set"*:
In order to update the information (e.g. first name) of for example an employee with specific ID, we can write the following query:

```
update database123.employeeinfo set name='Sarah' where
idEmployeeInfo='1';
```

Example 10 (Delete Info)
In order to delete the information of a person with specific id, we can enter the following query:

```
delete from database123.employeeinfo where idEmployeeInfo='5';
```

Example 11 Change
If for example we want to change the name of column from *idEmplyeeInfo* simply into *ID*, we can enter the following code into the query:

```
alter table database123.employeeinfo change idEmployeeInfo ID
int unsigned not null;
```

Example 12 Combine Cols

In order to combine columns such as name and surname under a single column *Name*, we can type the following codes in query:

```
select concat(name," ", surname) as 'Name' from database123.employeeinfo;
```

Example 13 Count Members

In order to count the total number of employees (as an e.g.), we can use the following commands:

```
select count(*) from database123.employeeinfo;
```

Example 14 Other fucntions

There are variety of other functions such as avg, min, max, etc. that we can use for editing a table that some of them provided here:

```
select min(age) from database123.employeeinfo;
select count(name) from database123.employeeinfo;
```

SQL Commmand Line Client & Tables

SQL command line client is look like the Figure 4.

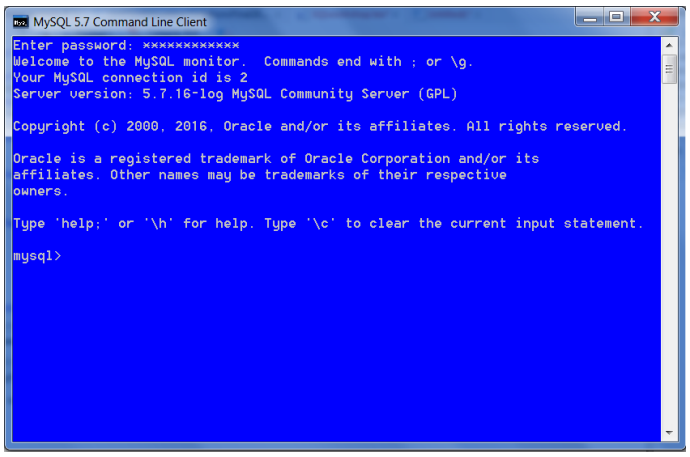


Figure 4: SQL Command Terminal

You can also code on the main terminal of computer. In order to work on the main computer terminal, which does not vary from one operation system to another, you first needs to get into the SQL by typing the following command:

```
mysql5 -u root -p;
```

where *u is for the userid*, *root is for the server (any server that you'd like to use*, and *p is for the password*. Please note that there is no difference in the commands if you enter them in the computer terminal or SQL command terminal itself. **The codes are very look like to the Queries**, and here we just give more information surrounding them.

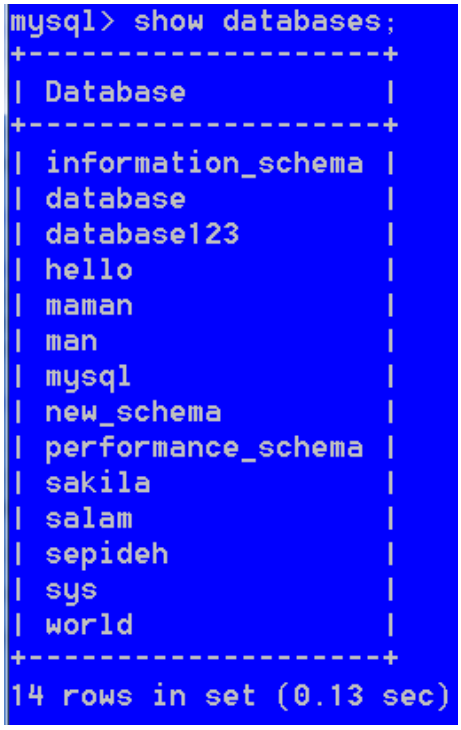
In the following of this section, we provide some useful examples to become more familiar with this command line client.

Example 1: Show Databases

In order to show all of the databases that we have, we can type

```
show databases;
```

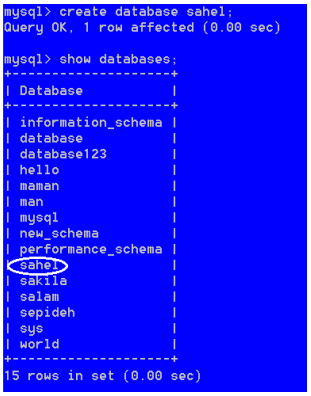
The result is look like the following



Example 2: Create Database

```
create database sahel;
show databases;
```

The result is look like the following

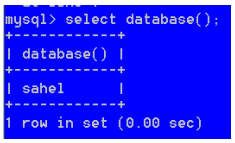


Example 3: Use Database

To use the database, we can type

```
use sahel;
```

where sahel is the name of database that we are going to use.



In order to destroy the database, we can type

```
drop database if exists sahel;
```

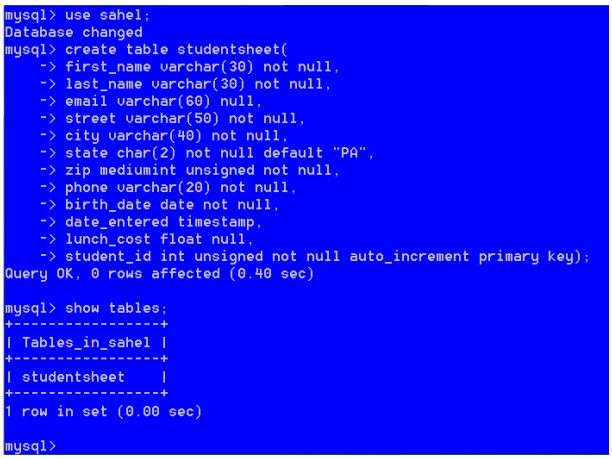
Example 4: Create a Student Database

In order to create a student database, we can type

```
create table studentsheet(
  first_name varchar(30) not null,
  last_name varchar(30) not null,
  email varchar(60) null,
  street varchar(50) not null,
  city varchar(40) not null,
  state char(2) not null default "PA",
  zip mediumint unsigned not null,
  phone varchar(20) not null,
  birth_date date not null,
  date_entered timestamp,
  lunch_cost float null,
```

```
  student_id int unsigned not null auto_increment primary key);
```

Hint: Please note that the PRIMARY KEY allows us to have a unique id for each student or individual to identify them and it cannot be NULL or changed.



To see how the created table is look like, we can type

```
describe studentsheet;
```

Example 5: Insert Information into Table

In order to insert infromation (e.g. students information) inot the table, we should type

```
insert into studentsheet value
  ('Sarah','Stone','sarah@umd.edu',
  '123 Main Street','College Park','MD','20740',
  '301-456-2342','1978-2-12',
  'NOW()','7.00', NULL);
```

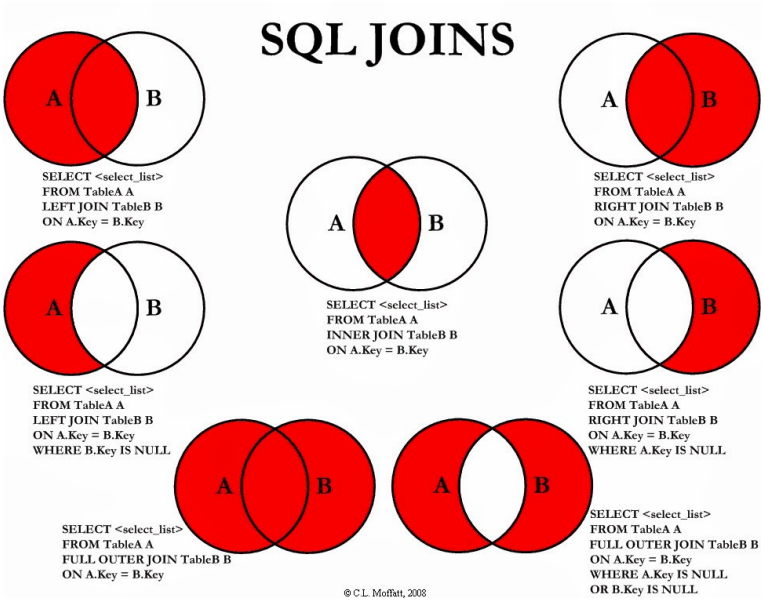
We can continue to enter all other students information just like this. In this way we are able to create any kinds of tables that we are interested in.

To see the entered dataset, we can type

```
select * from studentsheet;
```

Join Tables

One of the strengths of a relational database is that data items are split across tables in a manner that reduces the redundancy in information stored. So, for example, if you are maintaining a database for a store that tracks sales transactions, you would track customer information in one table and the transactions in another table, that includes the key to that customer in the customer table which would allow linkage. In this setup the customer information only needs to be in the database once - in the customer table - and not in each transaction record - which would be unnecessarily duplicative if each customer had many transactions. In order to link these tables back together to perform analyses one must join the tables in an SQL query. There are multiple types of joins that can be used that manage how you want your output table to be constructed.



SAS/SQL

You may never need to implement your own mySQL database, however these SQL commands you have learned will not go to waste. SQL statements can be used in SAS by invoking PROC SQL. Often using SQL to manipulate your datasets in SAS will allow you to do things quicker and with less code than it would take in a data step. You can even use SAS data step options like drop, keep, and rename within your SQL statement in PROC SQL.

PROC SQL executes without a run statement and you can include multiple

SQL statements without reinvoking PROC SQL. You enter "quit;" when you are done entering SQL statements.

An example:

```
/* average cluster size per cluster */
data ncs3;
set ncs;
person = 1;
run;

proc sql;
CREATE TABLE ncs4 as
SELECT SESTRAT, SECLUSTR, sum(person) as numppl
FROM ncs3
GROUP BY SESTRAT, SECLUSTR;
quit;

proc means data=ncs4;
var numppl;
run;
```

Some PROC SQL references:

- DATA Step vs. PROC SQL: What's a neophyte to do?
- Top Ten Reasons to Use PROC SQL

ER Diagram

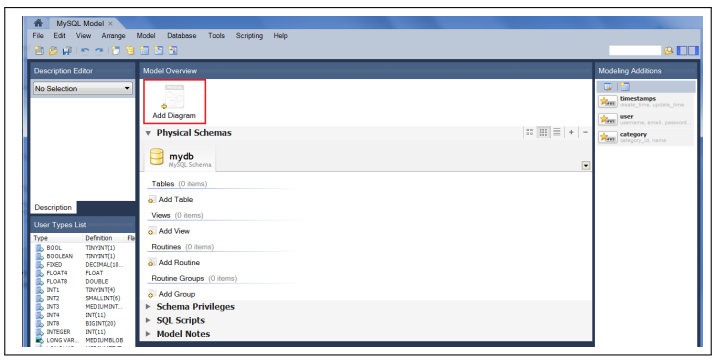
Entity Relation Diagram (ERD) is a graphical representation of an information system that shows the relationship between people, objects, places, concepts or events within a company or organization. ERD is a data modeling technique for defining a relationship.

For example, an ER diagram representing the information system for a company's sales department might start with graphical representations of entities such as the sales representative, the customer, the customer's address, the customer's order, the product and the warehouse. Lines and symbols can be used to represent the relationship between entities, and text can be used to label the relationships [2]

While creating diagrams we should pay attention which kinds of relationships exist among entities. Three main cardinal relationships are:

- **One-to-One (1:1)** For example, if each customer in a database is associated with one mailing address.
- **One-to-Many (1:M)** For example, a single customer places an order for multiple products.
- **Many-to-Many (M:N)** For example, at a company where all call center agents work with multiple customers.

In these set of diagrams the boxes (entities) are connected by lines (relationships) which express the associations and dependencies between entities. In order to create ER diagram in the SQL, we need to go *Models+* after openning the MySQL Benchmark. From there click on the *Add Diagram* as you can see from the following figure to create a new ER diagram.



[2]The definition extracted from the Online Sources.

In the following of this section we provide an example. In this example we provide a business school relationship among department, employee, division, assignment, and project. We throughly explain diagrams in the Workshop.