Structured Query Language(SQL)

Compro Prasad

October 31, 2016

Contents

1 Introduction **1**

2 Basics **2**
2.1 Logging in 2
2.2 What is a query? 2
2.3 Accessing Databases/Schemas 2
2.4 Accessing Table 2
2.5 Accessing columns 3
2.5.1 Most used types of columns: 3
2.6 Using databases 3
2.7 Accessing rows 4

3 Creating **4**
3.1 Databases 4
3.2 Tables 4
3.2.1 Basic table 4

1 Introduction

`SQL` is the interface for communicating to a binary file which is called a `database` in terms of `SQL`. So, keep in mind that if you change the binary file using an editor like hexedit then you could run into problems. So, don't mess with such kinds of files on your computer.
A **database** is further divided into **tables**. Tables are further divided into two categories: **columns** and **rows**.
Please make sure a `SQL` is installed on your machine before starting this tutorial.

2 Basics

2.1 Logging in

It is necessary to login to access the whole/part of your SQL. Now you would say that there is no use of login. But what if the frontend developer of Facebook gets into the database and steals information of everybody.
So, there is a concept of users in SQL. This removes the necessity of having different things in the same place. Instead, SQL just puts everything in the same place and permits the specific user to the specific information.

Here, is a simple command for logging into SQL using **tom** as the username:

```
mysql -u tom -p
```

After pressing RETURN mysql would ask for a password. Enter your password and you are good to go.

Note: During the time of installation the default user is **root**.

2.2 What is a query?

Query is a successful statement in SQL that gets a job done by the end user. It can be creating tables or inserting values or modifying values etc. The following are some examples of queries in MySQL:

```
create table information(column1 int);
use login_db;
select 1+2 from dual;
```

2.3 Accessing Databases/Schemas

Now, since you have logged in you have the control over what you have access to. You may not have been permitted to access even a single database. To see the schemas run this query:

```
show databases;
```

2.4 Accessing Table

Don't you know tables?! There can be many tables inside a database. Thats enough introduction to tables in SQL

```
show tables from information_schema;
```

Replace **information**$_{\textbf{schema}}$ with the database of your choice.
You can use:

```
show tables;
```

when you are using a specific database. See Using databases.

2.5 Accessing columns

```
show columns from information_schema.collations;
```

It is of the form `<database>.<table>`.
Want a shorter form? Use the following approach:

```
describe information_schema.collations;
```

2.5.1 Most used types of columns:

Type	Description
`char(n)`	A string of characters of length `n`. Same memory consumption for different strings.
`varchar(n)`	A string of characters of length `n`. Memory consumption according to length
`int`	An integer field
`bigint`	An integer field with a large range
`date`	A date of the form `YYYY-MM-DD`
`datetime`	A field addressing a date and time value.

2.6 Using databases

Databases are less in comparison to the tables in each database. Therefore, writing `<database>.<table>` can be a bit tedious. So, there is a feature in sql and that is using a database:

```
use information_schema;
```

You can now see columns using:

```
describe collations;
```

Since you are using the database, you cannot operate on a table thats inside another database using a normal syntax. But you can do two things outside the database. They are:

```
show databases;
use performance_schema;
```

Here **performance**$_{\textbf{schema}}$ can be another database you would like to use. From now on it will be assumed that you are using a database and then the respective query can be run after that.

2.7 Accessing rows

See all rows and columns from a table:

```
select * from collations;
```

Filter specific set of rows from a table:

```
select is_default, sortlen from collations;
```

Filter specific set of rows and columns from a table:

```
select is_default, sortlen from collations where id < 50 and is_default='Yes';
```

3 Creating

3.1 Databases

```
create database login_db;
```

3.2 Tables

3.2.1 Basic table