A Quick Introduction to SQL

We don’t expect that you have taken a database course, so most of you will not know the SQL language. SQL is an interesting beast (and not my area of expertise). On the surface it is very simple – there are only about ten command that are heavily used and at the same time the language is robust enough that entire economies can be run with it. For this lab, you only need to know the basic structure of a database and a few simple commands.

At a very simplistic level a database system can be thought of has being made up of four parts.

1) In the same way a filesystem may contain multiple files, each database system may have more than one databases. Before we can do anything with a database, we need to tell the system which one we want to interact with.
2) The fundamental unit of a database is the table. One database will have many tables in it.
3) Each table is made up a fixed number of columns, each column has a particular meaning and format. The structure of a table (i.e., the name and format for the columns) is called the schema.
4) The data in the database is represented in the rows of the tables.

For example, a simple database named “Golden Animal Registration” might look like this:

Table “Pets”		
ID	Name	Type
100	Rex	Dog
110	Fluffy	Cat
120	Spot	Dog
130	Midnight	Cat

Table “Food”	
Type	Food
Dog	Puppy Chow
Cat	Cat Chow

Table “Owner”		
ID	Owner	Tag#
100	Sam	342
110	Fred	467
120	Sue	34
130	Sam	3

- USE <database>
  The same way a filesystem contains many files, a database system may contain multiple databases. The USE command selects the database that all future commands operate on.

- SHOW tables;
  Will list all the tables in the selected database.

- DESCRIBE <table>
  Will list the name and type for each column in the table.

- SELECT <columns> FROM <table>
  Will return the data in the listed columns in the given table. For example:
  SELECT ID,Salary,SSN FROM credential;
  You can use wildcards
  SELECT * FROM credential;

- SELECT <columns> FROM <table> WHERE <condition>
  Will return the data in the requested columns ONLY for those records that match the condition.
  For example:
  SELECT * FROM credential WHERE Name='Alice';
  SELECT * FROM credential WHERE Salary>30000;

- UPDATE <table> SET <assignment> WHERE <condition>
  Will update the records that match the condition with the assignment. For example:
  UPDATE credential SET address='1500 Illinois' WHERE Name='Ryan';

- And pay special attention to the comment character (#). For example, the following would select everything from credential, not just those with salaries greater than 30000
  SELECT * FROM credential; # WHERE Salary>30000;