My name is Charles Schultz and I will be your instructor for CSC 191. We will be meeting in Room D208 from 5:00 pm to 7:00 pm every Tuesday. I am available by appointment for “office hours” prior to class. The best way to get a hold of me is email: sacrophyte at gmail.com

But I don’t want to talk about me right now; we need to talk about you. Why are you taking this class? What do you want to get out of it? There are no wrong answers. :)

[after all students have had a chance to share]

The purpose of this class is to introduce SQL, the language of databases. Some of you are part of the database or software development track, which requires this class. Others of you are re-training for a new job, or brushing up on your IT tools because your duties have changed. Some of you may just want to learn SQL for its own sake. \*grin\* My goal is to get you to a point where you have mastered the basics of SQL; we go over general relational database architecture (tables, rows, columns, constraints), writing queries that get information from the database, and alters database data. As time and interest permits, we will cover slightly more advanced topics like stored procedures.

**Overview of class**

**Phase 1 - what’s a database?** This first part of class is becoming familiar with the terminology. Some of you already know all this and will breeze through it. It is important to understand that we are talking about a relational database system, and that we have tables that relate to one another in some fashion.

**Phase 2 - mastering the single-table query.** The most simplest queries tend to be the best to start learning with. We will go over the “grammar” of a standard SQL query, learn about the different clauses (SELECT, FROM, WHERE, GROUP BY, ORDER BY) and simple functions that affect the output.

**Phase 3 - the meat of this course.** Here we apply our understanding of single-table queries and combine data from other tables. The secret magic sauce here is that once tables are joined, you can perceive them as one giant blob of data, otherwise known as a datasource or rowsource.

**Phase 4 - wrapping up with the challenging stuff.** Now that you have mastered the basics, it is only fitting that you get your feet wet with some more advanced topics. We will go over stored procedures and what makes a “complex SQL” complex.

**Expectations**

Our time in class will focus on practicing SQL. We will be busy typing, dialoguing, inquiring and researching. I don’t want to lecture that much, and I really want to avoid death by powerpoint. During time “online” for this hybrid class, you will be expected to pursue your own reading and work consistently on a semester-long class project that will culminate during the last week of class; you will slowly build a portfolio that can be shared after this class is done. Some of you will speed ahead, some of you will need to go a little slower; you are expected to communicate as much as possible in regards to what you as a student need.

**Self-Assessment**

Some of you will want a way to measure how well you grasp SQL. I would point you to the Konagora “[SQL Test](http://edu.konagora.com/SQLtest.php)”.

**Two Paths**

Some of you will want to rush to the “end” and finish this class as fast as possible. There are two ways to get credit for this class, no matter how long it takes (meaning, it might take you a full two hours, or it might take you 14 weeks):

1. Complete a semester project; you will decide what you want to work on by the second week of class
2. Complete a SQL Exam via Konagora (I have to give you a private link)

**Cliff Notes**

I am providing “cliff notes” for each week. This is primarily a tool for students that are participating online more than half of the time (and may not be physically present in class). In-class students are welcome and encouraged to work through the cliff notes as well. It is expected that online students will read and try out all the examples in the cliff notes. This will mirror what we do in class.

A caveat about using the cliff notes; I have tried to make sure that all the sql is “copyable”, but sometimes a stray typo will slip in. Even worse, sometimes a stray curly quote (‘) will slip in, like so:

select ‘hi’ from dual;

Unknown column '‘hi’' in 'field list'

While the syntax looks perfectly normal, most sql parsers will not like those silly curly quotes, and will require you use a normal quote:

select 'hi' from dual;

|  |
| --- |
| **hi** |
| hi |