**SQL Notes**

1. Basics
   1. SQL = Structured Query Language
   2. Designed to manage data stored in relational databases
      1. Relational Database - a database that organizes information into one or more tables
   3. SQL Commands: https://www.codecademy.com/articles/sql-commands
2. Theory
   1. Table - a collection of data organized into rows and columns
      1. Sometimes referred to as ‘relations’
   2. Column - a set of data values of a particular type (ex. integer, text, etc.)
   3. Row - a single record in a table
   4. Data Types:
      1. INTEGER - a positive or negative whole number
      2. TEXT - a string
      3. DATE - YYYY-MM-DD format
      4. REAL - a decimal value
   5. Constraints - add information about how a column can be used
      1. CREATE TABLE table\_name (

param1\_name INTEGER PRIMARY KEY,

param2\_name TEXT UNIQUE,

param3\_name TEXT NOT NULL,

param4\_name TEXT DEFAULT ‘Not Applicable’

);

* + 1. ‘PRIMARY KEY’ is used to uniquely identify a row, no two rows can have the same primary key value
    2. ‘UNIQUE’ is used to validate a different value for each row, similar to primary key; but a table can have multiple columns with the unique constraint
    3. ‘NOT NULL’ indicates the column must have a value
    4. ‘DEFAULT’ takes an argument to be inserted in a new row if a value is not specified for that column
  1. Statement - text a database recognizes as a valid command, always ends with a semi-colon (;)
     1. CREATE TABLE table\_name (

column\_1 data\_type,

column\_2 data\_type

);

* + 1. ‘CREATE TABLE’ is a clause or command. Should be written in capital letters.
    2. ‘table\_name’ refers to the name of the table the command is applied to
    3. Everything inside the ( parenthesis ) are parameters - values passed to a clause as an argument
    4. The number of lines used in a statement does not matter, can all be on one line or split
  1. Schema - the blueprint of the database, tells you how many tables and which columns are what data types, and how many rows (or entries) exist per table
  2. Query - ability to retrieve information stored in a database
  3. Operators:
     1. = equal to
     2. != not equal to
     3. > greater than
     4. < less than
     5. >= greater than or equal to
     6. <= less than or equal to
     7. NULL is an operator?
        1. IS NULL
        2. IS NOT NULL

1. Commands (Clauses)
   1. Essentials
      1. SELECT argument FROM table;
         1. Fetches data from a database, looks for tables/rows called ‘argument’ from the table called ‘table’
         2. \* is a wildcard that can be used in place of ‘argument’ to fetch all data in ‘table’
         3. SELECT always returns a new table called the ‘result set’
         4. Can select multiple arguments by separating arguments with commas, ex: ‘argument1, argument2’
      2. CREATE TABLE table\_name ( column\_name INTEGER, column2\_name TEXT);
         1. Creates a new table with 2 columns, named ‘column\_name’ and ‘column2\_name’, with the data types of Integer and Text (respectively)
      3. INSERT INTO table\_name (param1, param2, param3) VALUES (value1, value2, value3);
         1. Adds a specified row (or rows) into a table
      4. UPDATE table\_name SET param1\_name = param1\_value WHERE param2\_name = param2\_value;
         1. Edits a row in ‘table\_name’ to update with a new value
      5. DELETE FROM table\_name WHERE param1\_name IS param1\_value
         1. This will work to delete multiple rows if parameter names matches the values specified
      6. WHERE is used to restrict query results to only the information we want
         1. SELECT \* FROM table\_name WHERE parameter\_name > argument\_value;
         2. ‘parameter\_name > argument\_value’ is a condition
         3. ‘>’ is an operator
      7. LIKE is used to compare similar values
         1. SELECT \* FROM movies\_table WHERE name LIKE ‘Se\_en’;
            1. This would find both ‘Seven’ and ‘Se7en’ if they existed in ‘movies\_table’
            2. ‘\_’ is a single character wildcard
         2. SELECT \* FROM movies\_table WHERE name LIKE ‘A%’;
            1. This would find all movies whose name begins with ‘A’
            2. To search movies that end with a, use ‘%a’
            3. To search for movies that contain a man, use ‘%man%’
         3. Like is NOT case-sensitive
   2. Non-Essentials
      1. ALTER TABLE table\_name ADD COLUMN param\_name PARAMDATATYPE
         1. Adds a the column ‘param\_name’ with a datatype of ‘PARAMDATATYPE’ to the table ‘table\_name’
         2. NULL is the value in SQL that will default to row values for newly added columns
      2. AS allows you to rename column outputs in a query
         1. SELECT column\_name AS ‘new\_name’ FROM table\_name;
         2. This does not modify the original table, only the output
      3. DISTINCT filters a query to only show unique values
         1. SELECT DISTINCT paramater\_name FROM table\_name;