Notes built from <https://www.w3schools.com/sql/default.asp>

* SQL stands for Structured Query Language
* To use SQL in your web site, you will need RDBMS – Relational Database Management System( ex: [MS SQL Server](https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017)) and a server-side scripting language such as PHP or ASP
* SQL keywords are not case sensitive
* Important commands
  + Select – extracts data from a database
  + Update – updates data in a database
  + Delete – deletes data from a database
  + Insert Into – inserts new data into a database
  + Create database – creates a new database
  + Alter database – modifies a database
  + Create table – creates a new table
  + Alter table – modifies a table
  + Drop table – deletes a table
  + Create index – creates an index
  + Drop index – deletes an index
* SELECT
  + Syntax: SELECT *column1, column2,* …   
     FROM *table\_name*
  + Use the \* to select all the fields available in the table
    - SELECT \* FROM *table\_name*
* SELECT DISTINCT is used to return only distinct values
  + Syntax: SELECT DISTINCT *column1, column2, …* FROM *table\_name*
* WHERE clauses is used to filter records
  + Syntax: SELECT *column1, column2, …* FROM *table\_name* WHERE *condition*
  + Operators include: =, <>, >, <, >=, <=, BETWEEN, LIKE, IN
  + AND operator
    - Syntax: WHERE *condition1* AND *condition2* AND *condition3*
  + OR operator
    - Syntax: WHERE *condition1* OR *condition2* OR *condition3*
  + NOT operator
    - Syntax: WHERE NOT *condition1*
  + You can use parenthesis to help prioritize
* ORDER BY sorts records in ascending or descending order
  + Syntax: SELECT *column1, column2, …* FROM *table\_name* ORDER BY *column1, column2, ….* ASC|DESC
  + Example: ORDER BY Country ASC, CustomerName DESC
  + When used, default is to sort records ascending
* INSERT INTO inserts new records into a table
  + If you are adding values for all columns of the table
    - Syntax: INSERT INTO *table\_name* VALUES (value1, value2, value3, ….);
  + If you are adding values for specific columns of table
    - Syntax: INSERT INTO *table\_name* (*column1, column2, column3, …*) VALUES (value1, value2, value3, ….);
* NULL value is a field with no value
  + Not the same thing as a zero value or a field that contains spaces
  + NULL operator
    - Syntax: SELECT *column\_names*  
       FROM *table\_name* WHERE *column\_name* IS NULL;
* UPDATE is used to modify the existing records in a table
  + Syntax: UPDATE *table\_name* SET *column1 = value1, column2 = value2, …* WHERE *condition*;
  + The WHERE operator is important to help specify which values to update, when excluded, will update all records
* DELETE deletes existing records in a table
  + Syntax: DELETE FROM *table\_name*  
     WHERE *condition*;
  + If you omit WHERE, everything will be deleted
  + If you want to delete everything
    - Syntax: DELETE \* FROM *table\_name*
* SELECT TOP selects the number of records to return
  + Syntax: SELECT TOP *number* \* | PERCENT column\_name(s)  
     FROM *table\_name* WHERE *condition;*
    - Include asterisk only when you are going to select all the columns
  + Syntax varies from different database systems
* MIN(), MAX() functions return the smallest and largest value of the selected column
  + Syntax: SELECT MIN | MAX(*column\_name*)  
     FROM *table\_name*  
     WHERE *condition;*
* COUNT() returns number of rows that matches a specified criteria
  + Syntax: SELECT COUNT(*column\_name)*  
     FROM *table\_name*  
     WHERE *condition*;
* AVG() returns average value of a numeric column
  + Syntax: SELECT AVG(*column\_name)*  
     FROM *table\_name*  
     WHERE *condition*;
* SUM() returns total sum of a numeric column
  + Syntax: SELECT SUM(*column\_name)*  
     FROM *table\_name*  
     WHERE *condition*;
* LIKE operator is used in a WHERE clause to search for a specified pattern in a column
  + Syntax: SELECT *column1, column2, ….* FROM *table\_name*  
     WHERE *column\_name* LIKE *pattern;*
  + Used with two wildcards: ‘%’ and ‘\_’ (MS Access uses ‘?’ instead of ‘\_’)
    - % - represents zero, one, or multiple characters
    - \_ - represents a single character
  + Examples:
    - WHERE *column\_name* LIKE ‘a%’ – Finds values that start with ‘a’
    - WHERE *column\_name* LIKE ‘%a’ – Finds values that end with ‘a’
    - WHERE *column\_name* LIKE ‘%or%’ – Finds values that have ‘or’ in any position
    - WHERE *column\_name* LIKE ‘\_r%’ – Finds values that have ‘r’ in second position
    - WHERE *column\_name* LIKE ‘a\_%\_%’ – Finds values that start with ‘a’ and are at least 3 characters in length
    - WHERE *column\_name* LIKE ‘a%o’ – Finds values that start with ‘a’ and ends with ‘o’
* Wildcards are characters that are used to substitute any other character in a string
  + Used with LIKE operator
  + Includes ‘%’ and ‘\_’
  + Also includes [*charlist*] or [*!charlist*]
    - [*charlist*] defines sets and ranges of characters to match
    - [!*charlist*] defines sets and ranges of characters NOT to match
    - Ex: SELECT \* FROM Customers   
       WHERE City LIKE ‘[!bsp]%’;
      * Selects all customers with a city NOT starting with ‘b’, ‘s’, or ‘p’
* IN operator allows you to specify multiple values in a WHERE clause – shorthand for multiple OR conditions
  + Syntax: SELECT *column\_name(s)*   
     FROM *table\_name*  
     WHERE *column\_name* IN *(value1, value2, ….)*;
* BETWEEN operator selects values within a given range – inclusive, and can be used on numbers, text, or dates
  + Syntax: SELECT *column\_names*   
     FROM *table\_name*  
     WHERE *column\_name* BETWEEN *value1* AND *value2*;
* Aliases are used to give a table, or columns in a table, a temporary name – used often to improve readability and is limited to the scope of the query
  + Syntax: SELECT *column1\_name* AS *alias\_name, colum2\_name* AS *alias\_name*   
     FROM *table\_name*;
* JOIN clause is used to combine rows from two or more tables, based on a related column between them
  + Syntax: SELECT *column1\_name*, *column2\_name*, *column3\_name*  
     FROM *table1* INNER JOIN *table 2* ON *table1.column1* = *table2.column1*
  + INNER JOIN selects records that matching values in both tables
  + LEFT JOIN returns all records from the left table, and the matched records from the right table - if there is no match, the result is NULL from the right side
  + RIGHT JOIN returns all records from the right table, and the matched records from the left table – if there is no match, the result is NULL from the left side.
  + FULL OUTER JOIN returns all records when there is a match in either left or right table records
  + self JOIN is a regular join, but the table is joined with itself
    - Syntax: SELECT *column\_name(s)* FROM *table1 T1, table1 T2*  
       WHERE *condition;*
* UNION operator is used to combine the result-set of two or more SELECT statements
  + Each SELECT statement within UNION must have the same number of columns
  + The columns must also have similar data types
  + The columns in each SELECT statement must also be in the same order
  + Selects only distinct values by default, to allow duplicate values, use UNION ALL
  + Syntax: SELECT *column\_name(s)* FROM *table1*  
     UNION  
     SELECT *column\_name(s)* FROM *table2;*
* GROUP BY statement is used with aggregate functions (COUNT,MAX,MIN,SUM,AVG) to group the result-set by one or more columns
* HAVING clause was added to SQL to use with aggregate functions