**SQL**

Function, Purpose, Modifier, in order of execution

| **Function** | **Modifier** | **Format** | **Other Details** |
| --- | --- | --- | --- |
| WITH |  | WITH (**tab1)** AS (**‘name’**) | Creates separate table to use before final query |
| SELECT |  | SELECT **Col1**, **Col2**, ... | Provide the columns you want |
|  | COUNT | COUNT**(Col)** | Counts # of rows with data in that column |
|  | SUM | SUM**(Col)** | Adds up data vertically in column |
|  | MIN | MIN**(Col)** | Gives smallest value in column |
|  | MAX | MAX**(Col)** | Gives largest value in column |
|  | AVG | AVG**(Col)** | Averages values in column |
|  | DISTINCT | DISTINCT **Col** | Provides unique rows for column |
|  | DATE\_TRUNC | DATE\_TRUNC(**‘day’,Col)** | Basically rounds data to the nearest whole value of what's in ‘\_’ |
|  | DATE\_PART | DATE\_PART**(‘day’,Col)** | Just pulls the part of the date you specify. |
|  | CASE WHEN / THEN/ ELSE | CASEWHEN **Col = ‘x’** then ‘**y**’ ELSE ‘**z’** END AS **column\_name** | SQL version of if/then. Can use any operators like in WHERE function. Multiple WHEN statements can be used per CASE |
|  | LEFT | LEFT **(Col, #)** AS | Returns characters from the left side up to the # |
|  | RIGHT | RIGHT **(Col, #)** AS | Returns characters from the right side up to the # |
|  | LENGTH | LENGTH **(Col)** | Counts # of characters in coloumn |
|  | POSITION | POSITION('\_' IN **Col)** | Returns # value of how far from left character appears |
|  | STRPOS | STRPOS(**Col**, ',') | Returns # value of how far from left character appears |
|  | SUBSTR | SUBSTR**(Col, #1, #2)** | Extracts a substring from a string (starting at any position) |
|  | LOWER | LOWER**(Col)** | Makes string lower case |
|  | UPPER | Upper**(Col)** | Makes string upper case |
|  | CONCAT | CONCAT(**Col1**, ' ', **Col2**) | Combines data from multiple columns ‘ ‘ optional |
|  | || | **Col1** || ' ' || **Col2** | Combines data from multiple columns ‘ ‘ optional |
|  | REPLACE | REPLACE(**Col**, ' ', '') | Replaces value in first ‘ ‘ with value in second |
|  | CAST | CAST(Col) AS date | Turns one form of data to another, like string to date |
|  | :: | (Col)::date |  |
|  | TO\_DATE | DATE\_PART('month', TO\_DATE(month, 'month') | Turns date part from string to number like May to 5 |
| FROM |  | FROM **Table** | Provide the table where the columns exist |
| JOIN |  | JOIN **Table 2** | Provides a second table to extract data |
| ON |  | ON **Table 1**  = **Table 2** | Provides row where 2 tables are joined |
|  |  |  |  |
| WHERE | >, = , < | WHERE **Col > 5** | A conditional statement to filter your results |
|  | LIKE | WHERE **Col LIKE '%me%'** | Only pulls rows where column has 'me' within the text. Could also use ‘\_me’ where each \_ is a character. |
|  | IN | WHERE **Col IN ('Y', 'N')** | A filter for only rows with column of 'Y' or 'N' |
|  | NOT | WHERE **Col NOT IN ('Y', 'N')** | **NOT** is frequently used with **LIKE** and **IN** |
|  | AND | WHERE **Col1 > 5 AND Col2 < 3** | Filter rows where two or more conditions must be true |
|  | OR | WHERE **Col1 > 5 OR Col2 < 3** | Filter rows where at least one condition must be true |
|  | BETWEEN | WHERE **Col BETWEEN 3 AND 5** | Often easier syntax than using an **AND** |
| GROUP BY |  | GROUP BY **Col** | Allows you to insert non-aggregate columns when aggregating data. Can use 1,2,3 to shorthand the order of numbers in select |
| HAVING |  | HAVING **SUM(Col)** > # | Specify data like WHERE function but for aggregated columns. Uses same modifiers as WHERE function. |
| ORDER BY |  | ORDER BY **Col** | Orders table based on the column. Used with **DESC**. |
| LIMIT |  | LIMIT **#** | Limits based number of rows returned |
| Database Data Types | | | |
| CHAR |  | CHAR(**size**) | Stores a string. Size specifies fixed length from 0 to 255 |
| VARCHAR |  | VARCHAR(**size**) | Stores a string. Size specifies max length. |
| BINARY |  | BINARY(**size**) | Stores binary strings. Size specifies fixed length. |
| VARBINARY |  | VARBINARY(**size**) | Stores binary strings. Size specifies Maxed length. |
| TEXT |  | TEXT**(size)** | Stores string with max length of 65,535 byes |
| BLOB |  | BLOB**(size)** | Stores BLOB with max length of 65,535 byes |
| TINYBLOB |  | TINYBLOB | Stores Binary Large Objects with maximum of 255 bytes |
| TINYTEXT |  | TINYTEXT | Stores string with maximum of 255 characters |
| MEDIUMBLOB |  | MEDIUMBLOB | Stores Binary Large Objects with maximum of 16k bytes |
| MEDIUMTEXT |  | MEDIUMTEXT | Stores string with maximum of 16k characters |
| LONGBLOB |  | LONGBLOB | Stores Binary Large Objects with maximum of 4b bytes |
| LONGTEXT |  | LONGTEXT | Stores string with maximum of 4b characters |
| ENUM |  | ENUM**(Val1, Val2, Val3,...)** | String object with 1 chosen value from list Val1,etc. Up to 65535 values list. Sorted in order entered |
| SET |  | SET**(Val1, Val2, Val3,...)** | String object with 0 or multiple values from list Val1,etc. Up to 64 values in a list. Sorted in order entered |
| BIT |  |  |  |
| BOOL | BOOLEAN |  |  |
| INT | INTEGER |  |  |
|  | TINYINT |  |  |
|  | SMALLINT |  |  |
|  | MEDIUMINT |  |  |
|  | BIGINT |  |  |
|  | INT |  |  |
| FLOAT |  | FLOAT | Number with decimals |
|  | DECIMAL | DECIMAL(size,d) | Number with exact amount of decimals. Size is total # of digits. D is # of digits after decimal. |
| DATE |  | DATE | YYYY-MM-DD |
| DATETIME |  | DATETIME( | Date & Time. YYYY-MM-DD hh:mm:ss |
| TIME |  | TIME | hh:mm:ss |
| YEAR |  | YEAR | YYYY |
| Database Manipulation | | | |
| INSERT |  | INSERT INTO **Table** **(col1, col 2,...)** VALUES **(val1, val2,...)** | Inserts values into columns specified |
| UPDATE |  | UPDATE **Table**  SET **Col1** =**’Val1’**, **Col2**=**’Val2’**  WHERE **Condition** | Changes the values in the columns specified (col1 & col2) to the new values (val1 & val2) when the where condition is satisfied. If no where conditions is satisfied, whole table is changed |
| DELETE |  | DELETE FROM **Table** WHERE **Condition** | Deletes rows where condition is met |
| CREATE |  | CREATE DATABASE **Name** | Creates new database with specified name |
| DROP |  | DROP TABLE **Table** | Removes the designated table from the database |
| TRUNCATE |  | TRUNCATE TABLE **Table** | Removes data in the designated table |
| ALTER |  | ALTER TABLE **Table**  ADD **Col1** **Datatype** | Adds col1 of the |
|  |  |  |  |
|  |  |  |  |

**Example of how to Join multiple Tables**

SELECT r.name region, s.name rep, a.name account

FROM sales\_reps s

JOIN region r

ON s.region\_id = r.id

JOIN accounts a

ON a.sales\_rep\_id = s.id

ORDER BY a.name;’

**Subqueries**

1. Can be used anywhere that a column would be pulled
2. Proper format is to indent entire sub-query
3. Most conditional logic works with subqueries that have 1 cell results, except for “in” which can include multiple

Resources

1. <https://mode.com/sql-tutorial/sql-string-functions-for-cleaning/>
2. <https://www.w3schools.com/sql/sql_isnull.asp>