**TERMS**

SQL – A programming language designed to manipulate and manage data in relational databases.

Relational Database – A database that organizes information into one or more tables.

Table – A collection of data organized into rows and columns.

Statement – A string of characters that the database recognizes as a valid command.

Constraints / Column Constraints – The rules applied to the values of individual columns about how a column can be used:

Different Constraints:

* PRIMARY KEY – Primary key constraints can be used to uniquely identify the row.
* UNIQUE – Unique columns have a different value for every row.
* NOT NULL – Not null columns must have a value.
* DEFAULT – Default assigns a default value for the column when no value is specified.

There can be only one PRIMARY KEY column per table and multiple UNIQUE columns.

Foreign Key – A column that contains the primary key to another table.

Aggregate Functions – Combines multiple rows together by performing calculations on them to form a single value of more meaningful information.

Column References – The GROUP BY and ORDER BY clauses can reference the selected columns by number in which that appear in the SELECT statement.

Outer Join – Combines rows from different tables even if the join condition is not met.

Inner Join – The JOIN clause allows for the return of results from more than one table by joining them together with other results based on common column values specified using an ON clause. INNER JOIN is the default JOIN and it will only return results matching the condition specified by ON.

**Statements**

CREATE TABLE – Statement that creates a new table in a database. It allows one to specify the name of the table and the name of each column in the table.

INSERT INTO – Statement that adds a new record (row) to a table.

ALTER TABLE – Statement used to modify the columns of an existing table. When combined with the ADD COLUMN clause, it is used to add a new column.

DELETE FROM – Statement used to delete records (rows) in a table. The WHERE clause specifies which record or records that should be deleted. If the WHERE clause is omitted, all records will be deleted.

UPDATE – Statement used to edit records (rows) in a table. It includes a SET clause that indicates the column to edit and a WHERE clause for specifying the record(s).

AND – Operator that allows multiple conditions to be combined. Records must match both conditions that are joined by AND to be included in the result set.

AS – Columns or tables can be aliased using the AS clause. This allows columns or tables to be specifically renamed in the returned result set.

OR – Operator that allows multiple conditions to be combined. Records matching either condition joined by the OR are included in the result set.

% - The % wildcard can be used in a LIKE operator pattern to match zero or more unspecified character(s).

SELECT – Statement that returns all columns from the provided table in the result set.

\_ - The \_ wildcard can be used in a LIKE operator pattern to match any single unspecified character.

ORDER BY – Clause used to sort the result set by a particular column wither alphabetically or numerically.

Different Ways to Order:

* DESC – A keyword used to sort the results in descending order.
* ASC – A keyword used to sort the results in ascending order (default).

LIKE – Operator that can be used inside of a WHERE clause to match a specified pattern.

DISTINCT – Unique values of a column can be selected using a DISTINCT query.

BETWEEN – Operator that can be used to filter by a range of values. The range of values can be text, numbers, or date data.

LIMIT – Clause that is used to narrow, or limit, a result set to the specified number of rows.

NULL – Column values can be NULL, or have no value. These records can be matched (or not matched) using the IS NULL and IS NOT NULL operators in combination with the WHERE clause.

WHERE – Clause used to filter records (rows) that match a certain condition.

SUM() – An aggregate function that takes the name of a column as an argument and return the sum of all the value in that column.

MAX() – An aggregate function that takes the name of a column as an argument and returns the largest value in a column.

COUNT() – An aggregate function that returns the total number of rows that match the specified criteria.

Note: A column name of the table can also be used instead of \*. Unlike COUNT(\*). this variation COUNT(column) will not count NULL values.

GROUP BY – Clause that will group records in a result set by identical values in one or more columns. It is often used in combination with aggregate functions to query information of similar records. The GROUP BY clause can come after FROM or WHERE but must come before and ORDER BY or LIMIT clause.

MIN() – An aggregate function that returns the smallest value in a column.

HAVING – Clause used to further filter the result set groups provided by the GROUP BY clause. HAVING is often used with aggregate functions to filter the result set groups based on an aggregate property.

Note: The HAVING clause must always come after a GROUP BY clause but must come before any ORDER BY or LIMIT clause.

ROUND() – Function that will round a number value to a specified number of places. It takes two arguments: a number, and a number of decimal places. It can be combined with other aggregate functions.

AVG() – The average of the values in a column.

JOIN – Will combine rows from different tables if the join condition is true.

LEFT JOIN – Will return every row in the left table, and if the join condition is not met, NULL values are used to fill in the columns from the right table.

WITH – Clause that stores the result of a query in a temporary table using an alias. Multiple temporary tables can be defined with one instance of the WITH keyword.

UNION – Clause used to combine results that appear from multiple SELECT statements and filter duplicates.

CROSS JOIN – Clause used to combine each row from one table with each tow from another in the result set. This JOIN is helpful for creating all possible combinations for the records (rows) in two tables.

CASE – Creates different outputs.