**QUERY FUNCTIONS:**

**SELECT \*** = (Get all the columns from a table)

**SELECT “ENTITIY” FROM “PLACE”** = (Return specific column(s) from a table)

**SELECT SUM(ENTITY**) = Get total number of entity across all tables

**SELECT AVG(ENTITY)** = Get average number of entity across all tables

**SELECT MAX(ENTITY)** = Get entity with the highest value

**SELECT MIN(ENTITY)** – Get entity with the smallest value

**ORDER BY**

**ASC** – Ascending Order

**DESC** – Descending Order

**LIMIT #** - Limit your table result by a number

**WHERE** - Specifies criteria that field values mustto be included in the query results.

**OR:** The OR keyword is used to combine two or more conditions in a WHERE clause. The OR keyword returns a row if any of the conditions are met.

**AND:** The AND keyword is used to combine two or more conditions in a WHERE clause. The AND keyword returns a row only if all of the conditions are met.

**HAVING** - Filters data based on a condition within a group.

**COUNT(ENTITY) -** The # of rows in a table or the # of rows that match a specified criteria

**\*\*GROUP BY “ENTITY”** = GROUP BY clause in SQL is used to group rows together based on a common value. The GROUP BY clause is typically used with aggregate functions, such as COUNT(), SUM(), AVG(), MAX(), and MIN(), to calculate summary statistics for each group

**SYNTAX:**

SELECT column\_name(s)

FROM table\_name

GROUP BY column\_name(s);

**INNER JOIN**: The INNER JOIN is the most common type of join. It returns all rows from the first table that have matching rows in the second table

**LEFT JOIN**: The LEFT JOIN returns all rows from the first table, even if there are no matching rows in the second table. The rows from the second table that do not have a match in the first table are returned as NULL values.

**RIGHT JOIN**: The RIGHT JOIN returns all rows from the second table, even if there are no matching rows in the first table. The rows from the first table that do not have a match in the second table are returned as NULL values.

**FULL JOIN**: The FULL JOIN returns all rows from both tables, even if there are no matching rows in the other table. The rows from the two tables that do not have a match in each other are returned as NULL values.

**SYNTAX**:

SELECT column\_name(s)

FROM table\_name1

JOIN table\_name2 ON table\_name1.column\_name = table\_name2.column\_name;

**Advanced Query Functions()**

**AS:** The AS keyword is used to give a column a new name. This can be useful for clarity or to make the query more readable**.**

**WILDCARD:** A wildcard is a special character that can be used to match one or more characters in a string. The most common wildcards are the percent sign (%), which matches zero or more characters, and the underscore (\_), which matches one character**.**

**LIKE:** The LIKE operator is used to match a string against a pattern. The pattern can contain wildcards.

**DISTINCT**: The DISTINCT keyword is used to remove duplicate rows from the results of a query.

**BETWEEN:** The BETWEEN operator is used to specify a range of values. The BETWEEN operator returns a row only if the value of the column is within the specified range.

**NULL:** The NULL keyword is used to represent a missing value. The NULL keyword cannot be compared to any other value**.**

DB CRUD/CREATE FUNCTIONS:

**CREATE DATABASE:** Creates a new database.

CREATE DATABASE database\_name;

**CREATE TABLE**: Creates a new table.

CREATE TABLE table\_name (

column\_name1 data\_type,

column\_name2 data\_type,

...

);

**ALTER TABLE**: Modifies an existing table.

ALTER TABLE table\_name

ADD column\_name data\_type,

MODIFY column\_name data\_type,

DROP column\_name;

**DROP TABLE**: Deletes an existing table.

DROP TABLE table\_name;

**DROP DATABASE:** Deletes an existing database.

DROP DATABASE database\_name;

**INSERT INTO:** Inserts new data into a table.

INSERT INTO table\_name (column\_name1, column\_name2, ...)

VALUES (value1, value2, ...);

**UPDATE**: Updates existing data in a table.

UPDATE table\_name

SET column\_name1 = value1,

column\_name2 = value2,

...

WHERE condition;

**DELETE**: Deletes data from a table.

DELETE FROM table\_name

WHERE condition;