**SQL MIN() and MAX() Functions**

The MIN() function returns the smallest value of the selected column.

The MAX() function returns the largest value of the selected column.

SELECT MIN(*column\_name*)

FROM *table\_name*

WHERE *condition*;

Example :

SELECT MIN(price)

FROM products;

SELECT MAX(*column\_name*)

FROM *table\_name*

WHERE *condition*;

Example :

SELECT MAX(price)

FROM products;

**The SQL COUNT(), AVG() and SUM() Functions**

The COUNT() function returns the number of rows that matches a specified criteria.

The AVG() function returns the average value of a numeric column.

The SUM() function returns the total sum of a numeric column.

Count() Syntax

SELECT COUNT(*column\_name*)

FROM *table\_name*

WHERE *condition*;

Ex :

SELECT COUNT(productName)

FROM products

WHERE price > 40;

AVG() Syntax

SELECT AVG(*column\_name*)

FROM *table\_name*

WHERE *condition*;

Ex :

SELECT AVG(price)

FROM products;

SUM() Syntax

SELECT SUM(*column\_name*)

FROM *table\_name*

WHERE *condition*;

Ex :

SELECT SUM(price)

FROM products;

**LIKE operator**

The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards used in conjunction with the LIKE operator:

* % - The percent sign represents zero, one, or multiple characters
* \_ - The underscore represents a single character

LIKE Syntax

SELECT *column1, column2, ...*

FROM *table\_name*

WHERE *columnN* LIKE *pattern*;

Here are some examples showing different LIKE operators with '%' and '\_' wildcards:

**LIKE Operator Description**

WHERE CustomerName LIKE 'a%' Finds any values that start with "a"

WHERE CustomerName LIKE '%a' Finds any values that end with "a"

WHERE CustomerName LIKE '%or%' Finds any values that have "or" in any position

WHERE CustomerName LIKE '\_r%' Finds any values have "r" in the second position

WHERE CustomerName LIKE 'a\_%\_%' Finds any values start with "a" and are at least 3 characters in length

WHERE ContactName LIKE 'a%o' Finds any values start with "a" and ends with "o"

**SQL Wildcard Characters**

A wildcard character is used to substitute any other character(s) in a string.

Wildcard characters are used with the SQL LIKE operator. The LIKE operator is used in a WHERE clause to search for a specified pattern in a column.

There are two wildcards used in conjunction with the LIKE operator:

* % - The percent sign represents zero, one, or multiple characters
* \_ - The underscore represents a single character

**IN operator**

The IN operator allows you to specify multiple values in a WHERE clause.

The IN operator is a shorthand for multiple OR conditions.

IN Syntax

SELECT *column\_name(s)*

FROM *table\_name*

WHERE *column\_name* IN (*value1*, *value2*, ...);

or:

SELECT *column\_name(s)*

FROM *table\_name*

WHERE *column\_name* IN (*SELECT STATEMENT*);

EX :

SELECT \*

FROM products

WHERE productName in ("sugar", "butter");

Output : Lists products sugar and butter

Ex 2 :

SELECT \*

FROM products

WHERE productName not in ("sugar", "butter");

Output : Lists products other than sugar and butter

Ex 3 : subquery

SELECT \*

FROM products

WHERE productName in (select productName from products where productName like "s%");

Output : products listed whose productName starts from letter S

**Aliases**

SQL aliases are used to give a table, or a column in a table, a temporary name.

Aliases are often used to make column names more readable.

An alias only exists for the duration of the query.

Alias Column Syntax

SELECT *column\_name* AS *alias\_name*

FROM *table\_name;*

Ex :

SELECT productName AS Label

FROM products;

Alias Table Syntax

SELECT column\_name(s)

FROM table\_name AS alias\_name;

Ex :

SELECT productName

FROM products AS Items;

Ex : multiple column alias

SELECT productId as ID, productName AS product

FROM products;