Grow Data Skills

AGENDA

- PATTERN MATCHING IN SQL
- HANDLING NULL VALUES
- FUNCTIONS IN SQL
- ORDER BY, GROUP BY, HAVING
- TOP, LIMIT,DISTINCT, ALIASES, UNION

PATTERN MATCHING IN SQL

Grow Data Skills

MySQL Wildcard Characters

A wildcard character is used to substitute one or more characters in a string.

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

Wildcard Characters in MySQL

Symbol	Description	Example
%	Represents zero or more characters	bl% finds bl, black, blue, and blob
_	Represents a single character	h_t finds hot, hat, and hit

The wildcards can also be used in combinations!

PATTERN MATCHING IN SQL

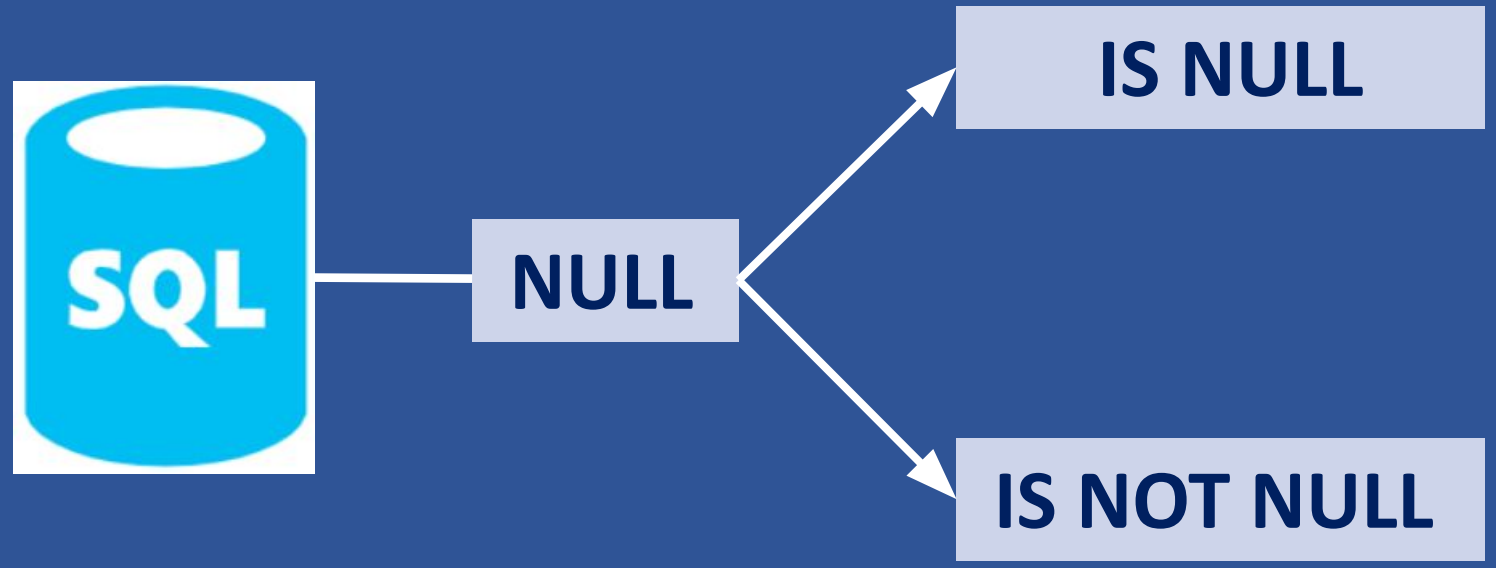
Grow Data Skills

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that starts with "a"
WHERE CustomerName LIKE '%a'	Finds any values that ends with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%_%'	Finds any values that starts with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that starts with "a" and ends with "o"

Regular Expressions(Regexp)

Pattern	What the Pattern matches
*	Zero or more instances of string preceding it
+	One or more instances of strings preceding it
-	Any single character
?	Match zero or one instances of the strings preceding it(contains)
^	caret(^) matches Beginning of string
$	End of string
[abc]	Any character listed between the square brackets
[^abc]	Any character not listed between the square brackets
[A-Z]	match any upper case letter.
[a-z]	match any lower case letter
[0-9]	match any digit from 0 through to 9.
p1|p2|p3	Alternation; matches any of the patterns p1, p2, or p3

HANDLING NULL VALUES



FUNCTIONS IN MYSQL

Aggregate Functions:

Aggregate functions perform calculations on a set of values and return a single value. They are often used with the `GROUP BY` clause to summarize data within groups.

- `SUM()`: Calculates the sum of values in a column.
- `AVG()`: Calculates the average of values in a column.
- `COUNT()`: Counts the number of rows or non-NULL values in a column.
- `MIN()`: Retrieves the minimum value in a column.
- `MAX()`: Retrieves the maximum value in a column.

FUNCTIONS IN MYSQL

Mathematical Functions:

Mathematical functions allow you to perform various mathematical operations on numerical data.

- `ROUND()`: Rounds a numeric value to a specified number of decimal places.
- `ABS()`: Returns the absolute value of a number.
- `SQRT()`: Calculates the square root of a number.
- `POWER()`: Raises a number to a specified power.
- `MOD()` or `%`: Calculates the remainder of a division operation.

FUNCTIONS IN MYSQL

Date and Time Functions:

- `NOW()`: Current date and time
- `CURDATE()`: Current date
- `CURTIME()`: Current time
- `DATE()`: Extract date part from a datetime value
- `TIME()`: Extract time part from a datetime value
- `YEAR()`: Extract year from a date or datetime value
- `MONTH()`: Extract month from a date or datetime value
- `DAY()`: Extract day of the month from a date or datetime value
- `HOUR()`: Extract hour from a datetime value
- `MINUTE()`: Extract minute from a datetime value
- `SECOND()`: Extract second from a datetime value
- `DATE_ADD()`: Add a specified time interval to a date or datetime value
- `DATE_SUB()`: Subtract a specified time interval from a date or datetime value
- `TIMESTAMPDIFF()`: Calculate the difference between two datetime values in a specified unit
- `DATEDIFF()`: Calculate the difference in days between two dates
- `DATE_FORMAT()`: Format a date or datetime value as a string
- `STR_TO_DATE()`: Convert a string to a date value using a specified format

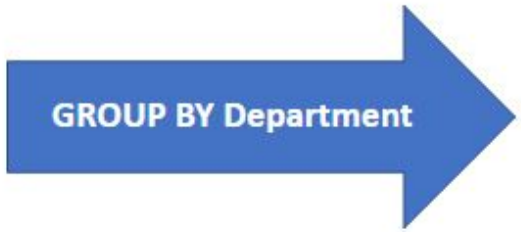
QUERY	FUNCTION
LIMIT	Specifies the maximum number of rows to be retrieved from the result set.
DISTINCT	Ensures that only unique values are returned, removing duplicates.
ALIASES	Aliases provide alternative names for columns or tables in a query.
UNION	Combines the result sets of two or more SELECT queries into a single result set, removing duplicates by default.

GROUP BY

The GROUP BY statement groups rows that have the same values into summary rows
It is often used with aggregate functions (COUNT(), MAX(), MIN(), SUM(), AVG()) to group the result-set by one or more columns.

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
ORDER BY column_name(s);
```

Employee ID	Name	Department
1001	Kundan	Sales
1002	Virat	Marketing
1003	Santosh	Education
1004	Veer	Marketing
1005	Shivani	Sales
1006	Yogesh	Education

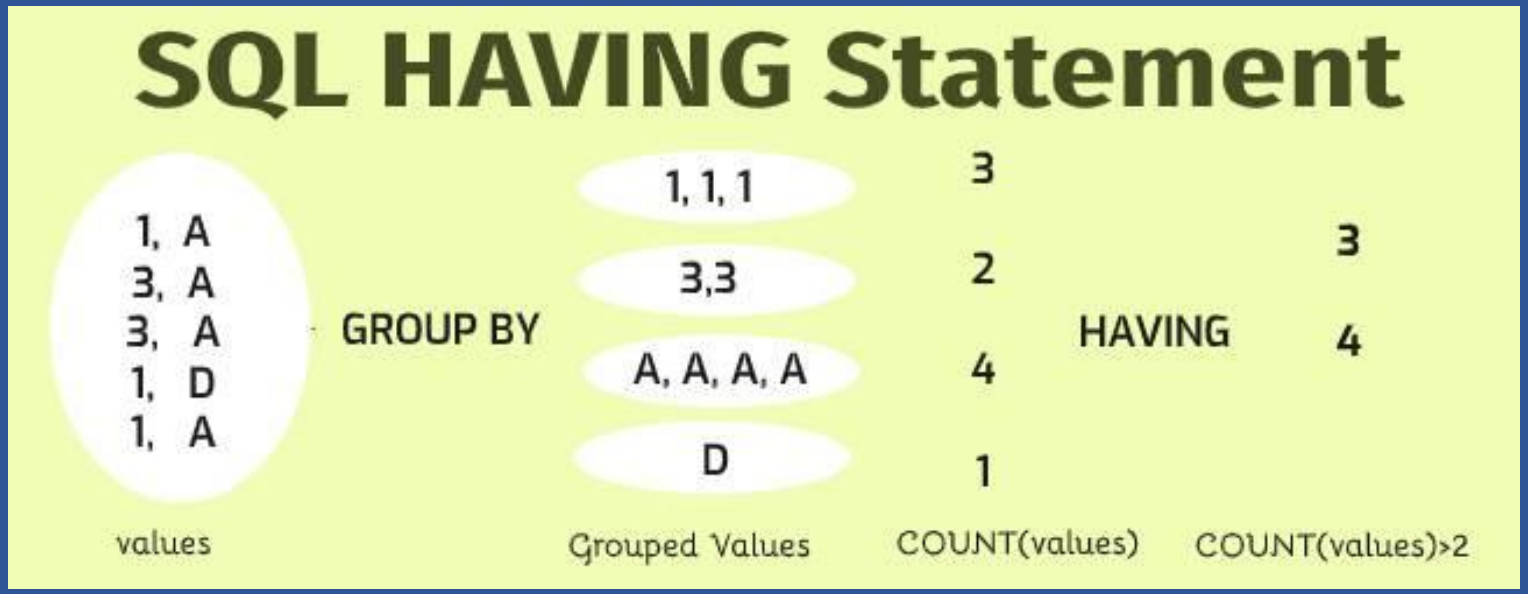


Department
Sales
Marketing
Education

HAVING

HAVING clause is added to SQL because the WHERE keyword cannot be used with aggregate functions

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition
```



ORDER BY

The `ORDER BY` keyword is used to sort the result-set in ascending or descending order.
It sorts the records in ascending order by default. To sort the records in descending order, use the `DESC` keyword.

EmployeeID	EmployeeLastName	EmployeeFirstName	EmailID
003	Jones	Amy	amy@gmail.com
006	Brown	Dan	dan@gmail.com
001	Donald	Jo	jo@gmail.com

```
SELECT *
FROM Employee
ORDER BY
EmployeeLastName;
```

Result

EmployeeID	EmployeeLastName	EmployeeFirstName	EmailID
006	Brown	Dan	dan@gmail.com
001	Donald	Jo	jo@gmail.com
003	Jones	Amy	amy@gmail.com