# Group By and Aggregation

## Agenda:

In today's session, we'll cover essential topics, including:-

🔸 Problem Statement

🔸 Aggregate Functions

🔸MIN, MAX

🔸 SUM

🔸AVG

🔸 COUNT(\*) vs. COUNT(1) vs. COUNT DISTINCT

🔸 Group By

🔸Having

🔸Having vs Where

## Summary of Previous Lecture:

Problem Statement:

* Amazon Fresh needs to leverage it data analytics team for a deeper understanding of:
  + Customer Behavior
  + Vendor Performance
  + Market Operations
  + Inventory management
* After gaining a basic understanding of SQL commands and data extraction techniques, you have been delving deeper into the Farmer’s Market database to understand all of these mentioned parameters.

Concatenating Strings:

* Concatenation is combining strings in SQL.
* CONCAT() function is used to merge string values together.
* Example: Merging first and last names into a full name column.
* Syntax: CONCAT(first\_name, " ", last\_name) AS full\_name

String Functions:

* Uppercase and Lowercase (UPPER, LOWER):
  + Modify the case of string values.
  + Use UPPER() and LOWER() functions.
  + Example: UPPER(first\_name), LOWER(last\_name)
* Substrings (SUBSTR):
  + Extract part of a string.
  + Syntax: SUBSTR(value, position, length).
  + Example: SUBSTR(last\_name, 1, 1) for the first character.
* InitCap (INITCAP):
  + Capitalize the first character of each word.
  + Example: INITCAP(full\_name)

Filtering Data - The WHERE Clause:

* WHERE clause filters records in the SELECT statement.
* Example: Selecting products from a specific category.
* Example: WHERE product\_category\_id = 1

Filtering with Multiple Conditions:

* Using operators (AND, OR, NOT) for multiple conditions
* Combine conditions in filtering.
* AND: All conditions must be true.
* OR: At least one condition must be true.
* NOT: Invert the condition.
* Syntax:

SELECT \*

FROM product

WHERE (product\_category\_id > 3 AND product\_category\_id <= 8) OR (product\_category\_id = 10)

Range Filtering (BETWEEN):

* Filter rows within a range.
* Syntax: BETWEEN value1 AND value2.
* Example: WHERE market\_date BETWEEN "2019-04-03" AND "2019-05-18"

Filtering with Lists (IN):

* Filter rows with specific values.
* Syntax: value IN (list).
* Example: WHERE customer\_last\_name IN ('Diaz', 'Edwards', 'Wilson')

Excluding Values (NOT IN):

* Exclude rows with specific values.
* Example: WHERE customer\_last\_name NOT IN ('Diaz', 'Edwards', 'Wilson')

Wildcard Filtering (LIKE):

* Search for partially matched strings.
* Use % for zero or more characters, \_ for one character.
* Example: WHERE customer\_first\_name LIKE "jer%"
* Wildcard Examples:
  + %ah%e matches names like "Michael".
  + a\_\_h%e matches names like "Matthew".
  + a%h%e matches names like "Alexander".

Distinct Keyword:

* DISTINCT keyword is used to retrieve unique values from one or more columns.
* Eliminate duplicate rows using SELECT DISTINCT col1, col2, ….
* Example: Selecting unique customer IDs from the customer\_purchases table.
* **Syntax:** SELECT DISTINCT customer\_id FROM farmers\_market.customer\_purchases

NULL Values:

* NULL represents the absence of a value in a cell.
* Use IS NULL and IS NOT NULL to filter rows with or without NULL values.
* The IS NULL condition is used in SQL to test for a NULL value. It returns TRUE if a NULL value is found, otherwise returns FALSE.
* Syntax: expression IS NULL
* Finding products without specified sizes and handling blank strings with TRIM().
* Example: WHERE product\_size IS NULL
* Use TRIM() to handle blank strings.
* Example: WHERE product\_size IS NULL OR TRIM(product\_size) = ""
* Similarly, The IS NOT NULL condition is used in SQL to test for a non-NULL value.

IFNULL:

* The IFNULL function specifies a value other than a null that is returned to your application when a null is encountered.
* The IFNULL() function is specified as follows: IFNULL(v1,v2)
* If the value of the first argument is not null, IFNULL returns the value of the first argument. If the first argument evaluates to a null, IFNULL returns the second argument.

Subqueries:

* Subqueries are queries within queries, used to retrieve data based on the results of another query.
* Syntax:

SELECT column\_name

FROM table\_name

WHERE column\_name expression operator

( SELECT COLUMN\_NAME from TABLE\_NAME WHERE ... );

* Example: Analyzing purchases made on rainy days by first getting the list of rainy dates and then using it to retrieve purchases.

SELECT \*

FROM `farmers\_market.customer\_purchases`

WHERE market\_date IN (SELECT market\_date

FROM `farmers\_market.market\_date\_info`

WHERE market\_rain\_flag = 1 )

CASE Statement:

* The CASE statement in SQL handles if/then logic.
* The CASE statement is followed by at least one pair of WHEN and THEN statements and finally an ELSE clause.
* Syntax:

SELECT cols,

CASE

WHEN [first conditional statement]

THEN [value or calculation]

WHEN [second conditional statement]

THEN [value or calculation]

ELSE [value or calculation]

END AS alias

FROM table

IF statement:

* The IF() function returns a value if the condition is TRUE and another value if the condition is FALSE.
* The IF() function can return values that can be either numeric or strings depending upon the context in which the function is used.
* The IF() function accepts one parameter which is the condition to be evaluated.
* **Syntax:**

IF(condition, true\_value, false\_value)

* **Parameters used:**
  + **condition –** It is used to specify the condition to be evaluated.
  + **true\_value –** It is an optional parameter that is used to specify the value to be returned if the condition evaluates to be true.
  + **false\_value –** It is an optional parameter that is used to specify the value to be returned if the condition evaluates to be false.