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
- Impact of the analysis

Summary of Previous Lecture:

Problem Statement:

- You are a Data Analyst at Amazon Fresh tasked with studying the Farmer's Market.
- Dataset: Farmer's Market database.

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.