Joins

Agenda:

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

- Problem Statement
- INNER JOIN
- LEFT JOIN
- RIGHT JOIN
- FULL OUTER JOIN
- UNION DISTINCT
- UNION ALL

Summary of Previous Lecture:

COUNT(*) vs **COUNT(column_name)** vs **COUNT(DISTINCT column_name)**

- COUNT(*) returns the total number of records in the table. COUNT(1), COUNT(999), COUNT("xyz") are all same as COUNT(*).
- COUNT(column_name) returns the total number of records of the table including the duplicate entries in the column *column_name* but ignores the NULL values.
- COUNT(DISTINCT column_name) neither includes the duplicate entries nor the NULL values.
- **Example:** Applying each function on the *product* table of Farmer's market database schema to count the number of product sizes.
- **Query:**

```
SELECT
     COUNT(*),
     COUNT(product_size),
     COUNT(DISTINCT product_size)
 FROM product;
```

GROUP BY

- GROUP BY groups together rows that have the same values in specified columns. It computes summaries (aggregated measures) for each group using the aggregate functions.
- Syntax:

```
SELECT [column_1], [column_2],
     COUNT() AS [column_name]
```

```
      SUM() AS [column_name]
FROM [table_name]
WHERE [condition]
GROUP BY [column_1], [column_2];
```

HAVING clause

- WHERE keyword fails when we use it with aggregate expressions like COUNT(), MAX(), AVG(), etc. along with the grouping.
- The HAVING clause is used to filter the results obtained by the GROUP BY clause based on some specific conditions.
- In a query, the HAVING clause is placed after the GROUP BY clause and before the ORDER BY clause.

- Syntax:

```
SELECT column1, column2,....,columnN
FROM tableName
WHERE [conditions]
GROUP BY column1
HAVING [conditons]
ORDER BY column_name
```

HAVING vs. WHERE

- WHERE is used to filter rows before grouping.
- HAVING is used to filter groups after aggregation.
- WHERE filters individual rows, while HAVING filters groups based on aggregate results.