Group By and Aggregation

1. Problem Statement
2. Aggregate functions
    a. MIN, MAX
    b. SUM
    c. AVG
    d. COUNT
3. COUNT(*), COUNT (1), COUNT DISTINCT
4. Group By
5. Impact of the analysis

Please note that any topics that are not covered in today's lecture will be covered in the next lecture.

Problem Statement: **Apollo Hospitals**

Apollo Hospitals, founded by Dr. Prathap C. Reddy in 1983 in Chennai, India, is one of Asia's largest and most renowned healthcare groups. With over 70 hospitals and more than 10,000 beds, Apollo provides comprehensive medical services across various specialties, including cardiology, oncology, neurology, and orthopedics.

The goal of this project is to conduct a comprehensive analysis of patient demographics, health metrics, hospital performance, and financial statistics using SQL.

This analysis aims to extract meaningful insights from the data to address the following key areas:

- **Patient Profiles:**
    a. Understand the distribution of patients based on age, gender, and ethnicity.

   b. Assess patients' BMI and primary medical conditions to identify prevalent health issues.

- **Health Conditions:**
   a. Identify the most common medical conditions leading to hospital admissions.
- **Hospital Performance:**
   a. Evaluate hospital traffic and patient turnover by analyzing admission types (emergency vs. elective, discharge dates, and length of stay.
- **Financial Statistics:**
   a. Investigate billing amounts across different hospitals and insurance providers to identify trends and outliers.
   b. Compare costs associated with different medical conditions to uncover potential areas for cost optimization.

The ultimate objective is to leverage these insights to enhance patient care quality, streamline hospital operations, and ensure cost-effective billing practices.

Dataset: link

Dataset Description:

- **Admission_ID**: Unique identifier for each patient record.
- **Name**: Patient's name.
- **Age**: Patient's age.
- **Gender**: Patient's gender (Male / Female).
- **BMI**: Body Mass Index, a measure of body fat based on height and weight.
- **Ethnicity**: Patient's ethnic background.
- **Height**: Patient's height (in cm).
- **Weight**: Patient's weight (in kg).
- **Blood_Type**: Patient's blood type (e.g., A+, O-, etc.).
- **Medical_Condition**: Primary medical condition for admission.
- **Admission_Date**: Date of admission to the hospital.
- **Doctor**: Name or ID of the attending doctor.
- **Hospital**: Name or ID of the hospital.
- **Insurance_Provider**: Name of the insurance provider.

- **Billing_Amount**: Total billing amount for the hospital stay.
- **Room_Number**: Room number where the patient is admitted.
- **Admission_Type**: Type of admission (e.g., Emergency, Elective, etc.).
- **Discharge_Date**: Date of discharge from the hospital.
- **Medication**: Medications prescribed during the hospital stay.
- **Test_Results**: Results of medical tests conducted.
- **Days_Hospitalised**: Total number of days the patient was hospitalized.

<u>Formulating questions to be explored based on the data provided</u>:

Patient Profiles:

- How many patient records do we have in our database?
- What is the distribution of patients' ages across the data?
- What is the gender ratio (male to female) for each medical condition?
- How is the patient population distributed across different ethnic backgrounds?
- What percentage of patients fall into different BMI categories (underweight, normal weight, overweight, obese)?
- What is the average BMI of patients diagnosed with different medical conditions?
- How are the different blood types distributed among patients with diabetes?
- Which ethnic group exhibits the highest susceptibility to cancer?
- Determine the count of universal blood donors and recipients within the patient population.

Health Conditions:

- What are the various medical conditions listed in the database?
- What percentage of patients are diagnosed with each medical condition?
- What are the most common medical conditions among patients aged 60 and above?
- Find the most frequently prescribed medication for specific medical conditions.

Hospital Performance:

- What is the average discharge time for patients based on the type of admission (emergency vs. elective)?
- Analyze and compare the average duration of hospitalization for various medical conditions.
- Identify healthcare providers who have treated a significant number of patients.

Financial Statistics:

- Identify the top 3 preferred insurance providers among patients.
- What is the average billing amount for treating different medical conditions?
- How do billing amounts vary based on the patient's insurance provider?

Aggregate Functions

SQL starts becoming more powerful when you use it to aggregate data.

Here we have a bunch of aggregate functions that take all the values present in a column as input and return a single value as a result.

They are commonly used with the SELECT statement to perform calculations on groups of rows or to **summarize data**.

Later on, we'll talk about groups also but for now, let's just stick to performing aggregations on a column.

1. **MIN()** - Returns the minimum value of a column.
2. **MAX()** - Returns the maximum value of a column.
3. **COUNT()** - Returns the number of rows in a column
4. **SUM()** - Returns the sum of values in a column
5. **AVG()** - Returns the average of values in a column

Q. Determine the age range of patients admitted to the hospital.

Approach:

- Use the MIN(Age) function to retrieve the age of the youngest patient.
- Use the MAX(Age) function to retrieve the age of the oldest patient.

Query:

```
SELECT
   MIN(Age) AS min_age,
   MAX(Age) AS max_age
FROM med.hospital;
```

Q. What is the average BMI (Body Mass Index) of the patients diagnosed with Obesity?

Approach:

- Filter the records to include only those patients who are diagnosed with Obesity.
- Use the AVG aggregate function to compute the average value of the BMI column.

Query:

```
SELECT
   AVG(BMI) AS Avg_BMI
FROM med.hospital
WHERE Medical_Condition = 'Obesity';
```

Q. How many patients' records do we have in our database?

Approach: Use the COUNT(*) aggregate function, which counts all rows in the table.

Query:

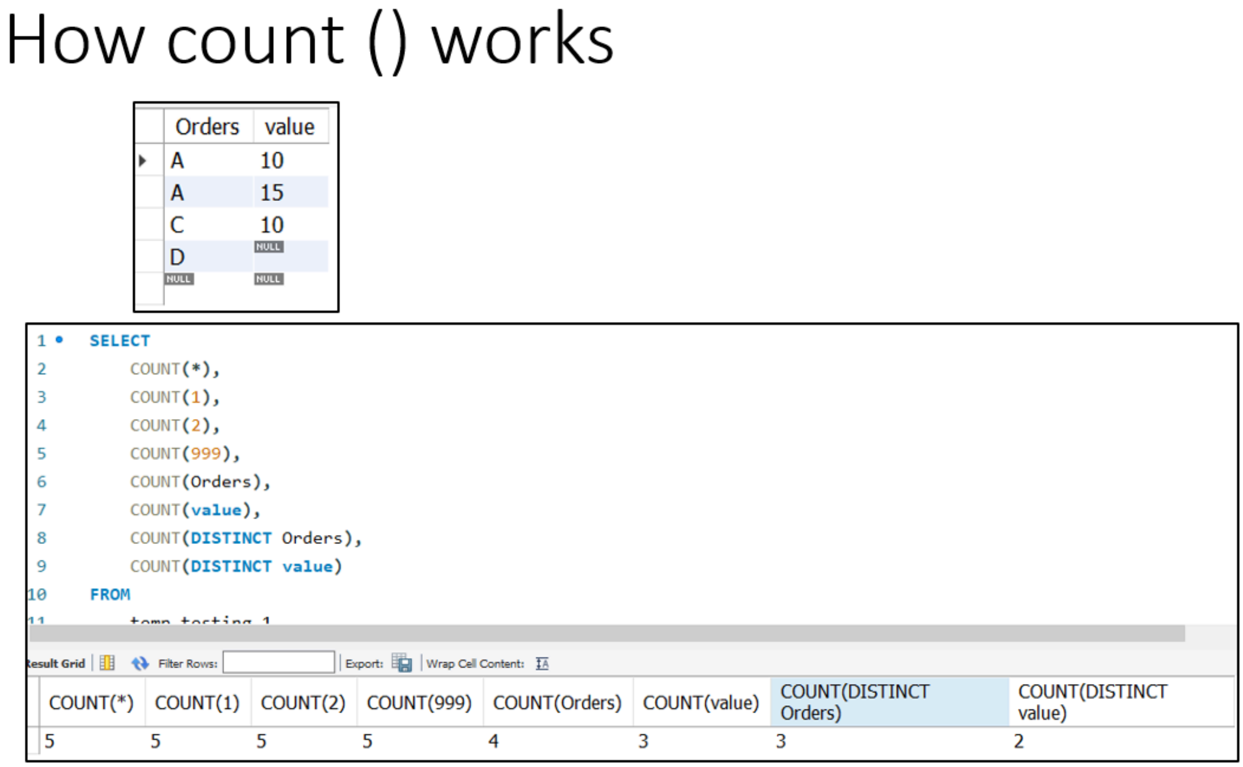
```
SELECT
   COUNT(*) AS Total_Patients
FROM med.hospital;
```

Q. What are the various medical conditions listed in our database?

Approach: Use the COUNT aggregate function combined with DISTINCT to find the number of unique entries in the Medical_Condition column.

Query:

```
SELECT
   COUNT(DISTINCT Medical_Condition) AS Unique_Conditions
FROM med.hospital;
```



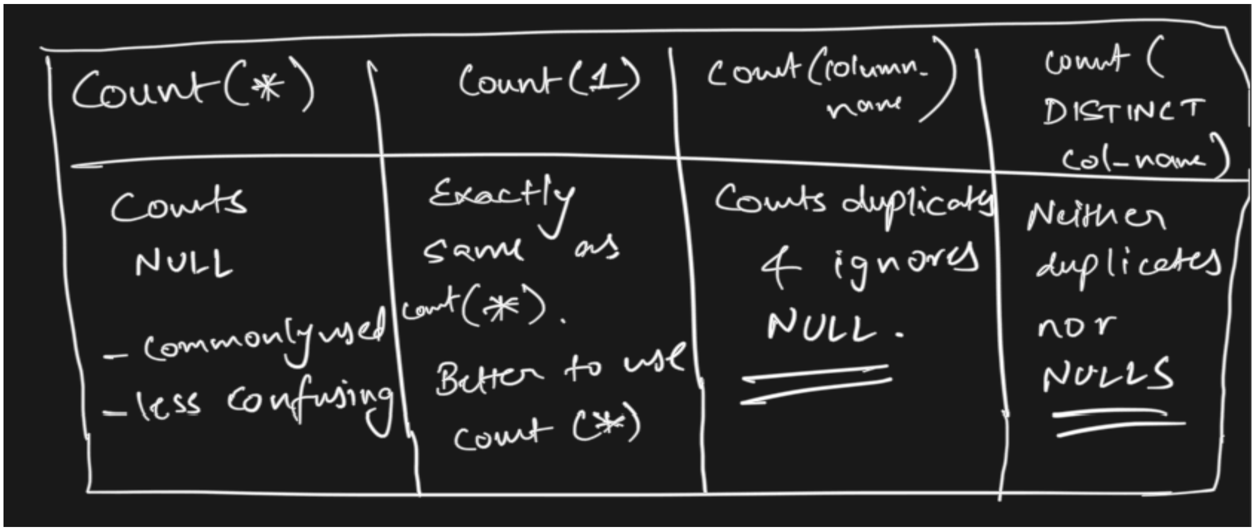
How count () works

Orders	value
A	10
A	15
C	10
D	NULL
NULL	NULL

```
SELECT
    COUNT(*),
    COUNT(1),
    COUNT(2),
    COUNT(999),
    COUNT(Orders),
    COUNT(value),
    COUNT(DISTINCT Orders),
    COUNT(DISTINCT value)
FROM
    temp_testing_1
```

COUNT(*)	COUNT(1)	COUNT(2)	COUNT(999)	COUNT(Orders)	COUNT(value)	COUNT(DISTINCT Orders)	COUNT(DISTINCT value)
5	5	5	5	4	3	3	2

Difference between COUNT(*) vs COUNT(1) vs COUNT(col_name) vs COUNT(DISTINCT col_name)



Count(*)	Count(1)	Count(column_name)	count(DISTINCT col_name)
Counts NULL - commonly used - less confusing	Exactly same as count(*). Better to use count(*)	Counts duplicates & ignores NULL.	Neither duplicates nor NULLS

Now that you all have seen how these aggregate functions work, let's move on to how we can use them on a group of records.

Group By

Using the GROUP BY statement, you can specify the level of summarization and then use aggregate functions to summarize values for the records in each group.

Syntax:

```
SELECT [columns to return]
FROM [table]
WHERE [conditional filter statements]
GROUP BY [columns to group on]
HAVING [conditional filter statements that are run after grouping]
ORDER BY [columns to sort on]
LIMIT [first x number of rows to be selected];
```

The GROUP BY keyword is followed by a comma-separated list of column names that indicate how you want to summarize the query results.

Order of Execution :

In a SQL query, the Group By clause comes right after the WHERE clause, followed by the Aggregate functions.

- **FROM** - The database gets the data from tables in the FROM clause and if necessary, performs the JOINs.
- **WHERE** - The data is filtered based on the conditions specified in the WHERE clause. Rows that do not meet the criteria are excluded.
- **GROUP BY** - After filtering the rows using the WHERE clause, the rows that remain are grouped together based on the columns specified in the GROUP BY clause.
- **Aggregate functions** - The aggregate functions are applied to the groups created in the GROUP BY clause.
- **SELECT** - After grouping and filtering, the SELECT clause specifies which columns and aggregate functions should be included in the result set.
- **ORDER BY** - It allows you to sort the result set based on one or more columns, either in ascending or descending order.
- **OFFSET** - The specified number of rows are skipped from the beginning of the result set.
- **LIMIT** - After skipping the rows, the LIMIT clause is applied to restrict the number of rows returned.

Q. What is the distribution of patients' ages across the data?

Approach:

- Use COUNT(*) to count the number of records (patients) for each age and assign this count to the alias Patient_Count.
- Use the GROUP BY clause to group the results by the Age column.
    - Grouping by age ensures that the count of patients is calculated separately for each distinct age.
- Use ORDER BY to sort the results by the Age column in ascending order.
    - Sorting by age makes it easy to see the distribution of patients' ages sequentially.

Query:

```
    SELECT
```

```
  Age,
  COUNT(*) AS Patient_Count
FROM med.hospital
GROUP BY Age
ORDER BY Age;
```

Q. How do billing amounts vary based on the patient's insurance provider?

Approach:

- Use the SUM(Billing_Amount) function to calculate the total billing amount claimed through each insurance provider.
- Use the GROUP BY clause to group the results by the Insurance_Provider column. This ensures that the total claimed amount is calculated separately for each insurance provider.
- Use the ORDER BY clause to sort the results in descending order of the total billing amount.

Query:

```
SELECT
   Insurance_Provider,
   SUM(Billing_Amount) AS Total_Claimed
FROM med.hospital
GROUP BY Insurance_Provider
ORDER BY Total_Claimed DESC;
```

Q. Analyze and compare the average duration of hospitalization for various medical conditions.

Approach:

- Use the AVG(Days_Hospitalised) function to calculate the average hospitalization period for each medical condition.
- Use the GROUP BY clause to group the results by the Medical_Condition column. This ensures that the average hospitalization period is calculated separately for each medical condition.

- Use the ROUND() function to round the calculated average hospitalization period to two decimal places for better readability.

Query:

```
SELECT
   Medical_Condition,
   ROUND(AVG(Days_Hospitalised)) AS Avg_Days_Hospitalised
FROM med.hospital
GROUP BY Medical_Condition;
```

Q. Calculate the average billing amount for cancer patients in each hospital.

Approach:

- Use the AVG function to calculate the average billing amount for cancer patients.
- Use the ROUND function to round the average billing amount to 2 decimal places for better readability.
- Use the WHERE clause to filter the records to include only those where the Medical_Condition is 'Cancer'.
- Use the GROUP BY clause to group the results by the Hospital column.

Query:

```
SELECT
   Hospital,
   ROUND(AVG(Billing_Amount), 2) AS Avg_Billing
FROM hospital
WHERE Medical_Condition = 'Cancer'
GROUP BY Hospital;
```

Q. How are the different blood types distributed among patients with diabetes?

Approach:

- Apply a WHERE clause to filter the records where the Medical_Condition is 'Diabetes'. This will restrict the analysis to only patients diagnosed with diabetes.
- Group the filtered data by the Blood_Type column. This will allow us to aggregate the patient counts for each blood type among patients with diabetes.
- Use the COUNT(*) function to count the number of patients for each blood type within the filtered dataset.
- Display the blood type and the corresponding patient count for each blood type among patients diagnosed with diabetes.

Query:

```
SELECT
  Blood_Type,
  COUNT(*) AS Patients_Count
FROM med.hospital
WHERE Medical_Condition = 'Diabetes'
GROUP BY Blood_Type;
```

Q. What percentage of patients are diagnosed with each medical condition?

Approach:

- Use a subquery to count the total number of patients in the patient table.
- Group the patients by Medical_Condition and count the number of patients in each group using COUNT(*).
- For each medical condition, calculate the percentage of patients by dividing the count of patients in that condition by the total number of patients and multiplying by 100.
- Use the ROUND function to round the percentage to two decimal places.
- Use the CONCAT function to append a "%" sign to the rounded percentage value.
- Use the ORDER BY clause to sort the results in descending order of the calculated percentage.

Query:

```
SELECT
  Medical_Condition,
  CONCAT(ROUND(COUNT(*) * 100.0 / (SELECT COUNT(*) FROM med.patient), 2), "%") AS Percentage_Patients
FROM med.hospital
GROUP BY Medical_Condition
ORDER BY Percentage_Patients DESC;
```

Q. What is the gender ratio (male to female) for each medical condition?

Approach:

- Begin by selecting the Medical_Condition and Gender columns from the hospital table.
- Group the data by the Medical_Condition column. This will enable us to analyze the gender distribution within each medical condition category.
- Use conditional aggregation with CASE statements to count the number of male and female patients within each group.
  - This is achieved by summing up 1 for each occurrence of 'Male' and 'Female', respectively.
- Calculate the male-to-female ratio for each medical condition.
  - This is done by dividing the count of male patients by the count of female patients.
  - However, to avoid division by zero errors, a condition is added to check if the count of female patients is greater than 0.
  - If it is, then the male-to-female ratio is calculated and rounded to two decimal places using the ROUND function; otherwise, NULL is returned.
- Display the calculated male and female counts along with the male-to-female ratio for each medical condition.

Query:

```
SELECT
  Medical_Condition,
  SUM(CASE WHEN Gender = 'Male' THEN 1 ELSE 0 END) AS Male_Count,
  SUM(CASE WHEN Gender = 'Female' THEN 1 ELSE 0 END) AS
Female_Count,
  CASE
    WHEN SUM(CASE WHEN Gender = 'Female' THEN 1 ELSE 0 END) > 0
      THEN ROUND(SUM(CASE WHEN Gender = 'Male' THEN 1 ELSE 0
END) / SUM(CASE WHEN Gender = 'Female' THEN 1 ELSE 0 END), 2)
    ELSE NULL
  END AS Male_to_Female_Ratio
FROM med.hospital
GROUP BY Medical_Condition;
```

How can Apollo Hospitals benefit from this analysis?

1. **List of Medical Conditions**
   - A comprehensive list of conditions helps in cataloging the hospital's treatment capabilities and identifying areas requiring further specialization or resource allocation.

2. **Age Distribution across Medical Conditions**
   - Understanding age distribution can help in tailoring medical services and preventive measures to specific age groups. For example, if a particular condition is more prevalent in older adults, Apollo can focus on specialized services for these patients.

3. **Gender Ratio for each Medical Condition**
   - Identifying gender disparities in medical conditions can aid in developing targeted health programs and awareness campaigns. For instance, if a condition predominantly affects females, Apollo can focus on women's health initiatives and early detection programs.

4. **Ethnic Distribution of Patient Population**
    - By understanding the ethnic distribution, Apollo Hospitals can ensure cultural competence in healthcare delivery, develop community-specific health programs, and address health disparities effectively.

5. **Blood Types Distribution among Diabetics**
    - Understanding the distribution of blood types in diabetic patients can assist in planning blood transfusions and managing blood bank inventories efficiently.

6. **Average Hospitalization Period by Condition**
    - Understanding the average length of stay for various conditions helps in resource planning, capacity management, and identifying areas for efficiency improvement.