Group By and Aggregation contd.

---

1. Problem Statement
2. Group By contd.
3. HAVING clause
4. WHERE vs. HAVING
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.

Formulating questions to be explored based on the data provided:

Patient Profiles:

- How many patients' records do we have in our database?
- What is the distribution of patients' ages across different medical conditions?
- 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?

Group By

Q. Identify the top 3 preferred insurance providers among patients.

Query:

```
SELECT
  Insurance_Provider,
  COUNT(*) AS Patient_Count
FROM med.hospital
GROUP BY Insurance_Provider
ORDER BY Patient_Count DESC
LIMIT 3;
```

Approach:

- Retrieve the Insurance_Provider column and use the COUNT(*) function to count the number of patients for each insurance provider.
- Group the results by the Insurance_Provider column to aggregate the counts for each provider.
- Sort the results in descending order of Patient_Count to list the most preferred insurance providers first.
- Use the LIMIT clause to restrict the output to the top 3 insurance providers based on patient count.

Q. What are the most common medical conditions among patients aged 60 and above?

Query:

```
SELECT
```

```
  Medical_Condition,
  COUNT(*) AS Patient_Count
FROM med.hospital
WHERE Age >= 60
GROUP BY Medical_Condition
ORDER BY Patient_Count DESC
LIMIT 5;
```

Approach:

- Use the WHERE clause to filter records for patients aged 60 or above.
- Utilize the COUNT(*) function to count the number of patients for each medical condition within the filtered dataset.
- Group the results by the Medical_Condition column to aggregate the counts for each condition.
- Sort the aggregated results in descending order of Patient_Count to identify the most prevalent medical conditions among the elderly patients.
- Use the LIMIT clause to restrict the output to the top 5 medical conditions based on patient count.

Q. Determine the count of universal blood donors (O- patients) and recipients (AB+ patients) within the patient population.

Query:

```
SELECT
  Blood_Type,
  COUNT(*) AS Number_of_Patients
FROM med.hospital
WHERE Blood_Type IN ('O-', 'AB+')
GROUP BY Blood_Type;
```

Approach:

- Retrieve the Blood_Type column and use the COUNT(*) function to count the number of patients for each blood type.
- Use the WHERE clause to filter records for blood types 'O-' (universal donors) and 'AB+' (universal receivers).

- Group the results by the Blood_Type column to aggregate the counts for each filtered blood type.

Q. How many patients fall into different BMI categories (underweight, normal weight, overweight, obese)?

Query:

```
SELECT
  CASE
    WHEN BMI < 20 THEN 'Underweight'
    WHEN BMI >= 20 AND BMI < 25 THEN 'Normal weight'
    WHEN BMI >= 25 AND BMI < 30 THEN 'Overweight'
    ELSE 'Obese'
  END AS Weight_Status,
  COUNT(*) AS Patient_Count
FROM med.hospital
GROUP BY Weight_Status;
```

Approach:

- Use the CASE statement to classify the BMI values into 'Underweight', 'Normal weight', 'Overweight', and 'Obese' categories.
- Create a derived column Weight_Status from the CASE statement and use the COUNT(*) function to count the number of patients in each category.
- Group the results by the derived Weight_Status column to aggregate the counts for each weight category.

Q. Which ethnic group exhibits the highest susceptibility to cancer?

Query:

```
SELECT
  Ethnicity,
  COUNT(*) AS Cancer_Patient_Count
FROM med.hospital
WHERE Medical_Condition = 'Cancer'
GROUP BY Ethnicity
```

```
ORDER BY Cancer_Patient_Count DESC
LIMIT 1;
```

Approach:

- Retrieve the Ethnicity column and use the COUNT(*) function to count the number of patients with cancer for each ethnicity.
- Use the WHERE clause to filter records for the medical condition 'Cancer'.
- Group the results by the Ethnicity column to aggregate the counts for each ethnicity.
- Sort the results in descending order of Cancer_Patient_Count to list the ethnicities with the highest number of cancer patients first.
- Use the LIMIT clause to restrict the output to the ethnicity with the highest cancer patient count.

Q. Identify the most commonly prescribed medications for each medical condition.

Query:

```
SELECT
   Medical_Condition,
   Medication,
   COUNT(*) AS Prescription_Count
FROM med.hospital
GROUP BY Medical_Condition, Medication
ORDER BY Prescription_Count DESC;
```

Approach:

- Retrieve the Medical_Condition and Medication columns, and use the COUNT(*) function to count the number of prescriptions for each combination of medical condition and medication.
- Group the results by both the Medical_Condition and Medication columns to aggregate the counts for each combination.
- Sort the results in descending order of Prescription_Count to list the most commonly prescribed medications for each medical condition first.

Filtering with Having

Filtering is another thing that can be done in the query after summarization occurs.

Using the **HAVING** clause allows you to filter the results of a query after the aggregate functions are applied, to grouped data.

This filters the groups based on the summary values.

Syntax:

```
SELECT aggregate_function(col)
FROM table
GROUP BY col
HAVING condition;
```

Recall the **Order of Execution** of a SQL query (as discussed earlier):

- **FROM** - The database gets the data from tables in 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.
- **HAVING** - The HAVING clause filters the groups of rows based on aggregate functions applied to the grouped data.
- **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.

The HAVING clause is executed after the WHERE and Group By clauses.

Q. Identify doctors who have treated more than 10 patients.

Query:

```
SELECT
  Doctor,
  COUNT(DISTINCT Name) AS Number_of_Patients
FROM med.hospital
GROUP BY Doctor
HAVING COUNT(DISTINCT Name) > 10
ORDER BY Number_of_Patients DESC;
```

Approach:

- Use COUNT(DISTINCT Name) to count the number of unique patients treated by each doctor.
- Use the GROUP BY clause to group the results by Doctor.
- Use the HAVING clause to filter the results to include only doctors who have treated more than 10 distinct patients.
- Use the ORDER BY clause to sort the results in descending order of Number_of_Patients.

Q. List medical conditions that have an average hospitalization period greater than 15 days and where the maximum billing amount exceeds $25,000.

Query:

```
SELECT
  Medical_Condition,
  AVG(Days_Hospitalised) AS Avg_Hospitalisation,
  MAX(Billing_Amount) AS Max_Billing
FROM med.hospital
GROUP BY Medical_Condition
```

```
HAVING
  AVG(Days_Hospitalised) > 15
  AND MAX(Billing_Amount) > 25000;
```

Approach:

- Use AVG(Days_Hospitalised) to calculate the average hospitalization period and MAX(Billing_Amount) to find the maximum billing amount.
- Use the GROUP BY clause to group the results by Medical_Condition, ensuring that the average hospitalization period and maximum billing amount are calculated for each medical condition separately.
- Use the HAVING clause to filter the grouped results.
    - Ensure that the average hospitalization period (AVG(Days_Hospitalised)) is greater than 15 days.
    - Ensure that the maximum billing amount (MAX(Billing_Amount)) exceeds $25,000.

Q. Find hospitals where the average billing amount exceeds the overall average billing amount by at least 50%.

Query:

```
SELECT
  Hospital,
  AVG(Billing_Amount) AS Avg_Billing
FROM med.hospital
GROUP BY Hospital
HAVING
  AVG(Billing_Amount) > (
 SELECT
   AVG(Billing_Amount) * 1.5
 FROM med.hospital);
```

Approach:

- Use AVG(Billing_Amount) to calculate the average billing amount for each hospital.
- Use the GROUP BY clause to group the results by Hospital.
- Use the HAVING clause to filter the results.
    - Calculate the overall average billing amount using a subquery

- Ensure that the average billing amount for each hospital exceeds the overall average billing amount by at least 50%.

Q. Calculate the total billing amount for emergency and elective admissions for each medical condition. Show only those medical conditions where emergency billing is less than elective billing.

Query:

```
SELECT
  Medical_Condition,
  SUM(CASE WHEN Admission_Type = 'Emergency' THEN
Billing_Amount ELSE 0 END) AS Emergency_Billing,
  SUM(CASE WHEN Admission_Type = 'Elective' THEN Billing_Amount
ELSE 0 END) AS Elective_Billing
FROM med.hospital
GROUP BY Medical_Condition
HAVING SUM(CASE WHEN Admission_Type = 'Emergency' THEN
Billing_Amount ELSE 0 END) <
    SUM(CASE WHEN Admission_Type = 'Elective' THEN Billing_Amount
ELSE 0 END)
ORDER BY Medical_Condition;
```

Approach:

- Use SUM(CASE WHEN Admission_Type = 'Emergency' THEN Billing_Amount ELSE 0 END) to calculate the total billing amount for emergency admissions.
- Use SUM(CASE WHEN Admission_Type = 'Elective' THEN Billing_Amount ELSE 0 END) to calculate the total billing amount for elective admissions.
- Use the GROUP BY clause to group the results by Medical_Condition.
- Use the HAVING clause to filter the results.
    - Ensure that the total billing amount for emergency admissions is less than the total billing amount for elective admissions.
- Use the ORDER BY clause to sort the results by Medical_Condition.

WHERE vs. HAVING

The main difference between the WHERE & HAVING clause is that

- the WHERE clause is used to specify a condition for filtering records before any groupings are made,
- while the HAVING clause is used to specify a condition for filtering values from a group.

Comparison Basis	WHERE Clause	HAVING Clause
Definition	It is used to perform filtration on individual rows.	It is used to perform filtration on groups.
Basic	It is implemented in row operations.	It is implemented in column operations.
Data fetching	The WHERE clause fetches the specific data from particular rows based on the specified condition	The HAVING clause first fetches the complete data. It then separates them according to the given condition.
Aggregate Functions	The WHERE clause does not allow to work with aggregate functions.	The HAVING clause can work with aggregate functions.
Act as	The WHERE clause acts as a pre-filter.	The HAVING clause acts as a post-filter.
Used with	We can use the WHERE clause with the SELECT, UPDATE, and DELETE statements.	The HAVING clause can only use with the SELECT statement.
GROUP BY	The GROUP BY clause comes after the WHERE clause.	The GROUP BY clause comes before the HAVING clause.

Q. Find the doctors who have treated 3 or more cancer patients, and among those doctors, identify the ones whose average billing amount is greater than $25,000.

Query:

```
SELECT
```

```
    Doctor,
    COUNT(*) AS Number_of_Patients,
    ROUND(AVG(Billing_Amount), 2) AS Average_Billing_Amount
FROM med.hospital
WHERE Medical_Condition = 'Cancer'
GROUP BY Doctor
HAVING COUNT(*) >= 3 AND AVG(Billing_Amount) > 25000
ORDER BY Doctor;
```

Approach:

- To identify doctors who have treated more than 3 cancer patients, start by filtering out patients suffering from cancer using a WHERE clause to exclude all other medical conditions.
- Once filtered, count the number of patients using COUNT(*) or COUNT(Admission_ID) to ensure the count exceeds 3.
- Next, calculate the average billing amount, which must be more than $25,000, using AVG(Billing_Amount).
- Group the results by doctor to aggregate data at the doctor level.
- Finally, apply the HAVING clause to filter doctors with more than 3 cancer patients and an average billing amount exceeding $25,000.
- This will provide the number of cancer patients and the average billing amount for each doctor meeting the criteria.

How can Apollo Hospitals benefit from this analysis?

1. **BMI Categories Distribution**
    - Knowing the BMI distribution helps in identifying population segments at risk of obesity-related conditions. Apollo Hospitals can implement weight management and nutrition programs to address these issues.

2. **Cancer Susceptibility by Ethnic Group**
    - Identifying ethnic groups with higher cancer susceptibility allows for targeted screening programs and preventive measures, potentially improving early detection and outcomes.

3. **Universal Blood Donors and Recipients Count**

- Knowing the count of universal blood donors (O-) and recipients (AB+) helps in emergency preparedness and ensuring adequate blood supply.

4. **Common Conditions in Patients aged 60+**
    - Tailoring geriatric care by focusing on prevalent conditions in older adults improves patient outcomes and enhances the hospital's reputation for elderly care.

5. **Frequently Prescribed Medications**
    - Analyzing prescription patterns can help in managing pharmacy inventories, negotiating better prices with suppliers, and ensuring the timely availability of essential medications.

6. **Top-performing Doctors**
    - Identifying top-performing healthcare providers can aid in recognizing and rewarding staff, as well as utilizing their expertise in training and mentoring.

7. **Preferred Insurance Providers**
    - Knowing the preferred insurance providers helps in negotiating better contracts and improving billing and claims processes with these insurers.