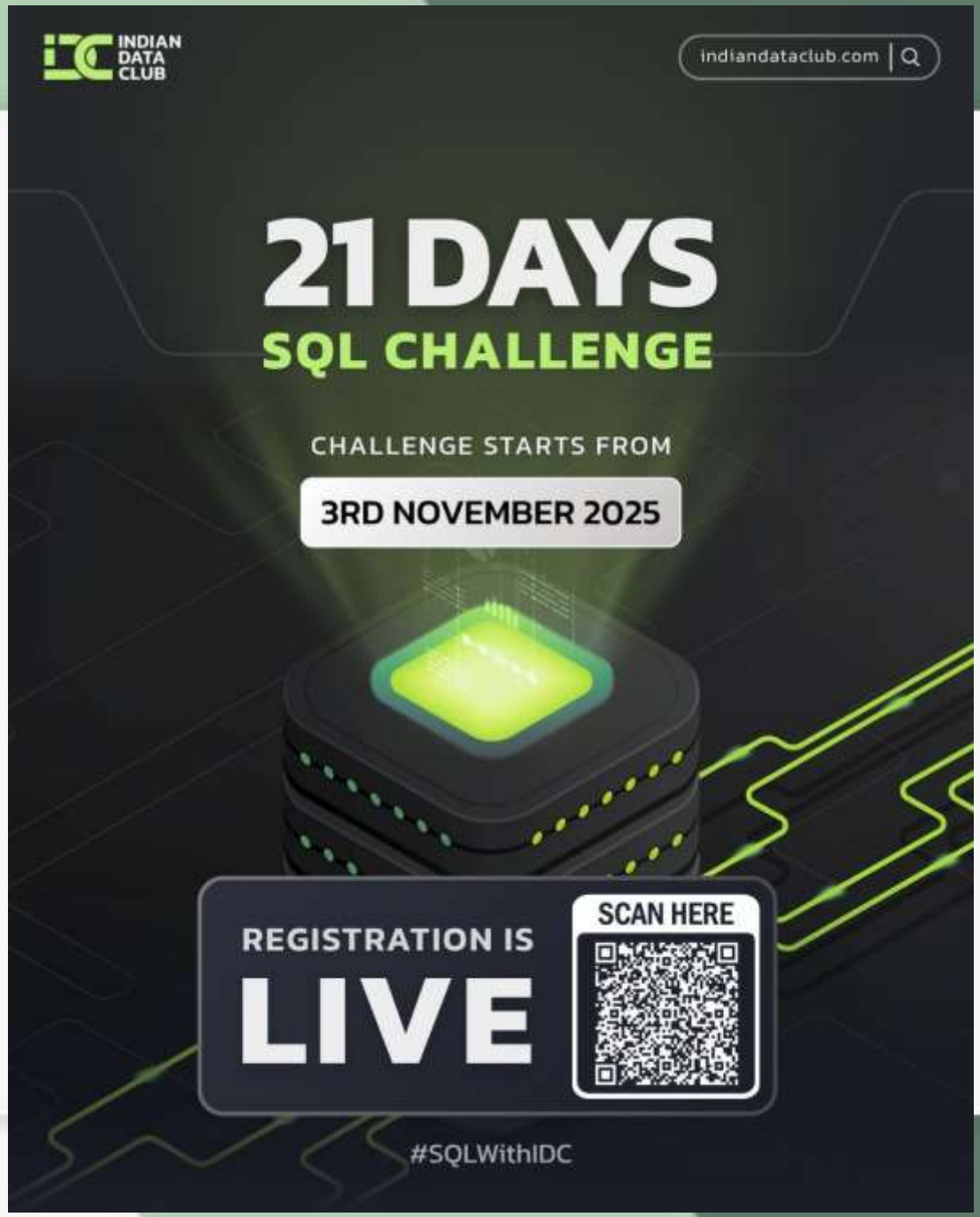
INDIAN DATA CLUB



Day 7 (10/11): HAVING Clause

🎯 Objective

To learn how to filter grouped results using the HAVING clause, especially when working with aggregate functions like SUM(), COUNT(), or AVG()

🔍 Topics Covered

- Difference between WHERE and HAVING
- Using GROUP BY with aggregate functions
- Applying HAVING to filter aggregate results
- Combining GROUP BY, HAVING, and ORDER BY in one query

INDIAN DATA CLUB

HAVING Clause

The HAVING clause is used to **filter groups** of data created by the GROUP BY clause.
It is **like WHERE**, but HAVING is used **after aggregation** (e.g., SUM, AVG, COUNT) to filter group results.

- **Why HAVING is Needed?**

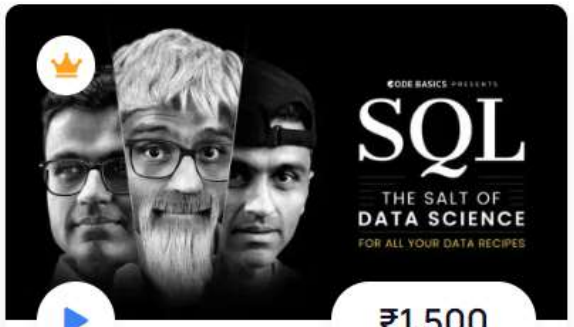
- WHERE filters **individual rows** (before grouping)
- HAVING filters **grouped results** (after GROUP BY)

Syntax :-

```
SELECT column1, AGG_FUNC(column2)
FROM table_name
GROUP BY column1
HAVING AGG_FUNC(column2) condition;
```

INDIAN DATA CLUB

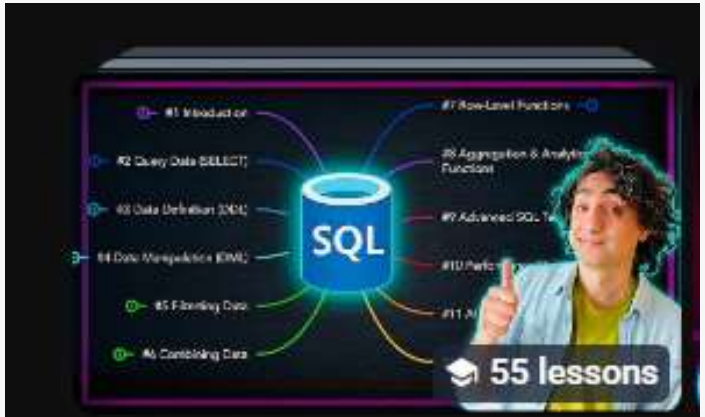
Resources:



SQL Beginner to Advanced For Data...

4.9 (1308) 9032 Enrolled

Beginners to Advanced SQL



SQL Bootcamp Playlist (2025) - Zero to Hero

View full course

Data with Baraa **SQL HAVING Clause - SQL Tutorial #30**

Neso Academy **GROUP BY and HAVING Clause in SQL**

Amigoscode **PostgreSQL: Group By Having | Course | 2019**

You can watch this topic from 46:24 in the video 👇

SQL Tutorial For Beginners | MySQL Tutorial

Beginners SQL tutorial with code and exercises. 🔑 Effortlessly set up MySQL, the open-source relational database, as we guide you through the installation process. 📊 Master SQL basics with essential clauses like SELECT, WHERE, DISTINCT, LIKE, ORDER BY, LIMIT, OFFSET, and BETWEEN. 💡 Elevate your skills with

https://www.youtube.com/watch?v=Rm0xH2Vpfi0&t=5s

INDIAN DATA CLUB

Practice Questions:

```sql
-- Find services that have admitted more than 500 patients in total.
SELECT
    service,
    SUM(patients_admitted) AS Total_Patient_Admitted
FROM services_weekly
GROUP BY service
HAVING Total_Patient_Admitted > 500 ;
```

service	Total_Patient_Admitted
emergency	1185
surgery	1686
general_medicine	2332
ICU	648

- **SELECT** → Picks the service column and calculates the total patients admitted using SUM().
- **FROM** → Data is taken from the services_weekly table.
- **GROUP BY** → Groups the data by each service so we can apply aggregate calculations.
- **HAVING** → Filters the grouped results and shows only those services where total admitted patients are more than 500.

INDIAN DATA CLUB

Practice Questions:

```
 9    -- Show services where average patient satisfaction is below 75.
10 •  SELECT
11        service,
12        AVG(patient_satisfaction) AS Avg_Patient_Satisfaction
13    FROM services_weekly
14    GROUP BY service
15    HAVING Avg_Patient_Satisfaction < 75;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

service	Avg_Patient_Satisfaction

- **SELECT** → Retrieves the service name and calculates the average satisfaction score using AVG().
- **FROM** → Uses data from the services_weekly table.
- **GROUP BY** → Groups the rows by each service so that the average can be computed per group.
- **HAVING** → Filters the grouped results and shows only services where the average satisfaction score is less than **75**.

INDIAN DATA CLUB

Practice Questions:

```sql
-- List weeks where total staff presence across all services was less than 50.
SELECT
    week,
    SUM(present) AS Total_Staff_Present
FROM staff_schedule
GROUP BY week
HAVING Total_Staff_Present < 50;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

week	Total_Staff_Present
3	0
6	0
9	0
12	0
15	0
18	0
21	0

- **SELECT** → Retrieves the week and calculates the total staff present using SUM().
- **FROM** → Reads data from the staff_schedule table.
- **GROUP BY** → Groups all rows by week so that staff from all services are added together.
- **HAVING** → Filters only those weeks where total staff presence is less than **50**.

INDIAN DATA CLUB

Daily Challenge:

```sql
-- Identify services that refused more than 100 patients in total
-- and had an average patient satisfaction below 80.
-- Show service name, total refused, and average satisfaction
SELECT
    service,
    SUM(patients_refused) AS Total_Patient_Refused,
    ROUND(AVG(patient_satisfaction),2) AS Average_Satisfaction
FROM services_weekly
GROUP BY service
HAVING Total_Patient_Refused > 100 AND
    Average_Satisfaction < 80;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

service	Total_Patient_Refused	Average_Satisfaction
emergency	5008	77.88
surgery	555	79.27

SELECT → Retrieves each service name, calculates the total patients refused using SUM(), and computes the average patient satisfaction using AVG() (rounded to 2 decimals).
FROM → Reads data from the services_weekly table.
GROUP BY → Groups the data by each service so totals and averages are calculated per service.
HAVING → Filters only those services where total patients refused are greater than 100 **and** the average satisfaction is below 80.