11/23/2025

DAY 21
#SQLWithIDC

[21 DAYS SQL CHALLENGE]

- divyanshi doser
[INDIAN DATA CLUB]

PRACTICE QUESTIONS

1. Create a CTE to calculate service statistics, then query from it.

```
WITH service_stats AS (
SELECT service, SUM(available_beds) AS total_available_beds,
SUM(patients_admitted) AS total_admission,
SUM(patients_refused) AS total_refusals,
SUM(patients_request) AS total_request,
ROUND(AVG(patients_satisfaction),2) AS avg_satisfaction,
SUM(
CASE
WHEN event IS NULL OR event LIKE 'none' THEN 0
ELSE 1 END
) AS total_events
FROM services_weekly
GROUP BY service
)
SELECT * FROM service_stats
ORDER BY total_admission DESC;
```

2. Use multiple CTEs to break down a complex query into logical steps.

```
WITH weekly_totals AS (
SELECT week, service,
SUM(patients_admitted) AS weekly_admissions
FROM services_weekly
GROUP BY week, service
),
service_average AS (
SELECT service,
ROUND(AVG(weekly_admissions),2) AS avg_weekly_admissions
FROM weekly_totals
GROUP BY service
)
SELECT w.service, w.week, w.weekly_admissions, s.avg_weekly_admissions
FROM weekly_totals AS w
JOIN service_average AS s
ON w.service = s.service
ORDER BY w.service, w.week;
```

3. Build a CTE for staff utilization and join it with patient data.

```
WITH staff_utilization AS (
SELECT service,
COUNT(DISTINCT staff_id) AS total_staff,
SUM(present) AS total_weeks_presents,
ROUND(AVG(present),2) AS attendence_rate
FROM staff_schedule
GROUP BY service
)
SELECT p.service, p.patient_id, p.name AS patient_name, p.age,
s.total_staff, s.total_weeks_presents, s.attendence_rate
FROM patients AS p
JOIN staff_utilization AS s
ON p.service = s.service;
```

DAILY CHALLENGE – DAY 21

Create a comprehensive hospital performance dashboard using CTEs.

Calculate: 1) Service-level metrics(total admission, refusals, avg satisfaction)

2) Staff metrics per service (total staff, avg weeks present)

3) Patient demographics per service(avg age, count).

Then combine all three CTEs to create a final report showing: service name, all calculated metrics, and an overall performance score(weighted average of admission rate and satisfaction)

Order by performance score descending.

```
WITH service_metrics AS (
SELECT service, SUM(patients_admitted) AS total_admissions,
SUM(patients_refused) AS total_refusals,
ROUND(AVG(patients_satisfaction),2) AS avg_satisfaction
FROM services_weekly
GROUP BY service
),
staff_metrics AS (
SELECT service, COUNT(DISTINCT staff_id) AS total_staff,
ROUND(AVG(present),2) AS avg_weeks_present
FROM staff_schedule
GROUP BY service
),
patient_stats AS (
SELECT service, ROUND(AVG(age),2) AS avg_age,
COUNT(patient_id) AS total_patients
FROM patients
GROUP BY service
)
SELECT s.service, s.total_admissions, s.total_refusals, sm.avg_weeks_present, p.avg_age, p.total_patients,
ROUND((s.total_admissions * 0.6) + (s.avg_satisfaction * 0.4),2) AS performance_score
FROM service_metrics AS s
LEFT JOIN staff_metrics AS sm
ON s.service = sm.service
LEFT JOIN patient_stats AS p
ON s.service = p.service
ORDER BY performance_score DESC;
```