

CTE+Window Q1 – Recent 2 Healthcare Visits

-- SQL Query:

```
-----  
  
WITH visits_stats AS (  
    SELECT  
        citizen_id,  
        visit_date,  
        department,  
        visit_cost,  
        ROW_NUMBER() OVER (  
            PARTITION BY citizen_id  
            ORDER BY visit_date DESC  
        ) AS recent_visits,  
        COUNT(*) OVER (  
            PARTITION BY citizen_id  
        ) AS total_visits  
    FROM healthcare_visits  
)  
  
SELECT  
    c.citizen_name,  
    vs.visit_date,  
    vs.department,
```

```

        vs.visit_cost,

        vs.total_visits

FROM citizens c

JOIN visits_stats vs ON c.citizen_id = vs.citizen_id

WHERE vs.recent_visits <= 2

ORDER BY vs.visit_date DESC

```

-- Sample Output:

	citizen_name character varying (100) 🔒	visit_date date 🔒	department character varying (50) 🔒	visit_cost numeric (8,2) 🔒	total_visits bigint 🔒
1	Anjali Rao	2025-06-24	Orthopedics	1100.00	1
2	Sunil Singh	2025-06-23	Cardiology	1600.00	1
3	Kiran Naik	2025-06-22	General	550.00	1
4	Ravi Mehta	2025-06-21	ENT	800.00	1
5	Neha Kulkarni	2025-06-20	General	400.00	1
6	Ankit Joshi	2025-06-19	Pediatrics	650.00	1
7	Sneha Patil	2025-06-18	Orthopedics	1200.00	1
8	Rahul Verma	2025-06-17	Cardiology	1500.00	1
9	Priya Desai	2025-06-16	ENT	750.00	1
10	Amit Sharma	2025-06-15	General	500.00	1

-- Insight:

Shows frequency of visits by the patients

Q4 – Total Transport Fare Paid by Each Citizen

-- SQL Query:

```
-----  
  
SELECT  
  
    c.citizen_name,  
  
    SUM(t.fare) AS total_fare  
  
FROM citizens c  
  
JOIN transport_usage t ON c.citizen_id = t.citizen_id  
  
GROUP BY c.citizen_name  
  
ORDER BY total_fare DESC;  
  
-----
```

-- Sample Output:

	citizen_name character varying (100) 🔒	total_fare numeric 🔒
1	Rahul Verma	60.00
2	Sneha Patil	40.00
3	Amit Sharma	35.00
4	Ravi Mehta	26.00
5	Priya Desai	22.50
6	Sunil Singh	22.00
7	Neha Kulkarni	18.75
8	Kiran Naik	15.00
9	Ankit Joshi	12.00
10	Anjali Rao	10.00

-- Insight:

This query shows which citizen spends the most on transport

Q5 – Average Visit Cost in Healthcare Departments

-- SQL Query:

```
-----  
  
SELECT  
  
    department,  
  
    ROUND(AVG(visit_cost), 2) AS avg_visit_cost  
  
FROM healthcare_visits  
  
GROUP BY department  
  
ORDER BY avg_visit_cost DESC;  
  
-----
```

-- Sample Output:

	department character varying (50) 🔒	avg_visit_cost numeric 🔒
1	Cardiology	1550.00
2	Orthopedics	1150.00
3	ENT	775.00
4	Pediatrics	650.00
5	General	483.33

-- Insight:

Helps analyze medical cost per department

Q6 – Citizens Giving Feedback < 3

-- SQL Query:

```
-----  
  
SELECT  
  
    c.citizen_name,  
  
    f.rating,  
  
    f.city_department,  
  
    f.comments  
  
FROM citizens c  
  
JOIN public_feedback f ON c.citizen_id = f.citizen_id  
  
WHERE f.rating < 3;  
  
-----
```

-- Sample Output:

	citizen_name character varying (100) 🔒	rating integer 🔒	city_department character varying (50) 🔒	comments text 🔒
1	Sneha Patil	2	Transport	Metro delays during peak hours.
2	Neha Kulkarni	1	Healthcare	Staff was rude and unhelpful.

-- Insight:





Identifies unhappy citizens for public service improvement

CTE Q4 – Areas Where Electricity Bill > City Avg

-- SQL Query:

```
-----  
  
WITH avg_elec_bill_stats AS (  
    SELECT  
        area_id,  
        ROUND(AVG(bill_amount), 2) AS avg_elec_bill  
    FROM electricity_usage  
    GROUP BY area_id  
,  
city_avg_stats AS (  
    SELECT ROUND(AVG(bill_amount), 2) AS ov_city_avg FROM  
    electricity_usage  
)  
  
SELECT  
    a.area_id,  
    a.area_name,  
    aebs.avg_elec_bill,  
    cas.ov_city_avg  
FROM avg_elec_bill_stats aebs  
JOIN city_area a ON aebs.area_id = a.area_id  
CROSS JOIN city_avg_stats cas  
WHERE aebs.avg_elec_bill > cas.ov_city_avg
```

-- Sample Output:

	area_id [PK] integer 	area_name character varying (100) 	avg_elec_bill numeric 	ov_city_avg numeric 
1	2	Kothrud	1680.30	1575.26
2	5	Hadapsar	1807.60	1575.26

-- Insight:

Flags areas with excessive utility spending