INDIAN DATA CLUB

INDIAN DATA CLUB

Day 15 (19/11): Multiple Joins

🎯 ***Objective***

To learn how to join **three or more tables** and understand how complex relationships work in SQL.

🔍 ***Topics Covered***

- Joining **three or more tables** using different join types
- Understanding **complex relationships** (one-to-many, many-to-many)
- Using **bridge/intermediate tables** for multi-step joins
- Avoiding duplicates and writing **clean, efficient** multi-table queries

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 FULL OUTER JOIN - SQL Tutorial #25**

The Data Millennials **How to Join two or more than two Tables using multiple colum...**

INDIAN DATA CLUB

Practice Questions:

```sql
-- Join patients, staff, and staff_schedule to
-- show patient service and staff availability.

SELECT
    p.patient_id,
    p.name AS patient_name,
    p.age,
    p.arrival_date,
    p.departure_date,
    p.service,
    COUNT(CASE WHEN ss.present = 1 THEN 1 END) AS weeks_staff_present
FROM patients p
LEFT JOIN staff s
USING(service)
LEFT JOIN staff_schedule ss
USING(staff_id)
GROUP BY p.patient_id, p.name, p.age, p.arrival_date, p.departure_date, p.service
ORDER BY weeks_staff_present DESC;
```

INDIAN DATA CLUB

QUERY EXPLANATION

patient_id	patient_name	age	arrival_date	departure_date	service	weeks_staff_present
PAT-003ce690	Larry Dixon	29	2025-01-19	2025-01-21	ICU	1497
PAT-00883d3c	Victor Taylor	0	2025-11-06	2025-11-13	ICU	1497
PAT-02ae68da	Corey Whitaker	36	2025-08-27	2025-09-10	ICU	1497
PAT-030f2b7d	Chris Velazquez	44	2025-06-05	2025-06-10	ICU	1497
PAT-04a8031e	Jake Shaw	83	2025-06-10	2025-06-22	ICU	1497
PAT-08591375	Garrett Lin	25	2025-02-18	2025-02-25	ICU	1497
PAT-0b69b6d1	Joann Ferguson	69	2025-07-30	2025-08-04	ICU	1497
PAT-0c687528	Monica Ellis	39	2025-04-24	2025-05-06	ICU	1497
PAT-0ced5480	Samuel Turner	23	2025-01-11	2025-01-12	ICU	1497
PAT-0f73df9c	Jennifer Banks	32	2025-05-09	2025-05-16	ICU	1497
PAT-104fadfd	Lisa Evans	46	2025-09-13	2025-09-20	ICU	1497
PAT-110d5794	Travis Hull	48	2025-11-12	2025-11-18	ICU	1497
PAT-12523d2c	Amanda Sulliva...	65	2025-01-26	2025-02-06	ICU	1497
PAT-128d605f	Mrs. Kristen Re...	74	2025-07-27	2025-07-31	ICU	1497
PAT-14da658e	Dennis Marshall	49	2025-05-20	2025-05-27	ICU	1497
PAT-14f1c2aa	Elizabeth Sanders	66	2025-01-05	2025-01-08	ICU	1497
PAT-15da980a	Kendra Wang ...	43	2025-11-12	2025-11-26	ICU	1497
PAT-17ceaafc	Mary Miller	54	2025-04-01	2025-04-07	ICU	1497
PAT-1a762a65	Vanessa Hatfield	85	2025-12-30	2026-01-02	ICU	1497
PAT-1bc9130a	Michael Miles	82	2025-02-13	2025-02-26	ICU	1497

Result 3

- **SELECT** → shows patient details + staff availability count.
- **FROM patients** → start with all patients.
- **LEFT JOIN staff** → attach staff based on service.
- **LEFT JOIN staff_schedule** → attach their weekly schedule.
- **COUNT(CASE WHEN present = 1 THEN 1 END)** → counts weeks staff were present.
- **GROUP BY** → one result per patient.
- **ORDER BY** → highest availability first.

INDIAN DATA CLUB

Practice Questions:

```sql
-- Combine services_weekly with staff and staff_schedule
-- for comprehensive service analysis.
SELECT
    sw.week,
    sw.service,
    sw.available_beds,
    sw.patients_request,
    sw.patients_admitted,
    sw.patients_refused,
    sw.patient_satisfaction,
    sw.staff_morale,
    COUNT(DISTINCT s.staff_id) AS total_staff_assigned,
    COUNT(CASE WHEN ss.present = 1 THEN 1 END) AS staff_present_count
FROM services_weekly sw
LEFT JOIN staff s
    ON sw.service = s.service
LEFT JOIN staff_schedule ss
    ON s.staff_id = ss.staff_id
   AND sw.week = ss.week
GROUP BY
    sw.week, sw.service, sw.available_beds, sw.patients_request, sw.patients_admitted,
    sw.patients_refused, sw.patient_satisfaction,sw.staff_morale
ORDER BY sw.week, sw.service;
```

INDIAN DATA CLUB

QUERY EXPLANATION

week	service	available_beds	patients_request	patients_admitted	patients_refused	patient_satisfaction	staff_morale	total_
1	emergency	32	76	32	44	67	70	29
1	general_medicine	37	201	37	164	97	43	27
1	ICU	22	31	22	9	84	91	48
1	surgery	45	130	45	85	83	78	22
2	emergency	28	169	28	141	75	64	29
2	general_medicine	43	183	43	140	73	93	27
2	ICU	16	7	7	0	79	85	48
2	surgery	40	26	26	0	96	56	22
3	emergency	32	177	32	145	73	58	29
3	general_medicine	37	58	37	21	95	63	27
3	ICU	20	21	20	1	82	89	48
3	surgery	27	66	27	39	63	72	22
4	emergency	32	157	32	125	83	75	29
4	general_medicine	43	152	43	109	67	66	27
4	ICU	20	21	20	1	64	85	48
4	surgery	56	57	56	1	74	94	22
5	emergency	25	388	25	363	93	72	29
5	general_medicine	40	103	40	63	73	52	27
5	ICU	22	13	13	0	73	88	48

- **SELECT** → retrieves weekly service data plus staff details and attendance.
- **FROM services_weekly sw** → starts with weekly service performance.
- **LEFT JOIN staff** → adds all staff assigned to each service.
- **LEFT JOIN staff_schedule** → adds staff attendance for that same week.

INDIAN DATA CLUB

Daily Challenge:

```sql
-- Create a comprehensive service analysis report for week 20 showing:
-- service name, total patients admitted that week, total patients refused,
-- average patient satisfaction, count of staff assigned to service,
-- and count of staff present that week.
-- Order by patients admitted descending.
SELECT
    sw.service,
    MAX(sw.patients_admitted) AS total_patients_admitted,
    MAX(sw.patients_refused) AS total_patients_refused,
    ROUND(AVG(sw.patient_satisfaction), 1) AS avg_patient_satisfaction,
    COUNT(DISTINCT s.staff_id) AS total_staff_assigned,
    COUNT(DISTINCT CASE WHEN ss.present = 1 THEN s.staff_id END) AS staff_present_this_week
FROM services_weekly sw
LEFT JOIN staff s
    ON sw.service = s.service
LEFT JOIN staff_schedule ss
    ON s.staff_id = ss.staff_id
   AND ss.week = sw.week
WHERE sw.week = 20
GROUP BY sw.service
ORDER BY total_patients_admitted DESC;
```

This query summarizes **service performance for week 20** by combining data from services_weekly, staff, and staff_schedule.

- **MAX(patients_admitted/refused)** → Weekly totals (avoids double-counting after joins)
- **AVG(patient_satisfaction)** → Average satisfaction for the week
- **COUNT(DISTINCT staff_id)** → Total staff assigned to that service
- **COUNT(DISTINCT staff_id with present=1)** → Staff actually present that week
- Finally, it groups results **per service** and sorts them by highest admitted patients.

service	total_patients_admitted	total_patients_refused	avg_patient_satisfaction	total_staff_assigned	staff_present_this_week
general_medicine	1080	810	64.0	27	24
surgery	682	176	99.0	22	22
emergency	609	696	93.0	29	27
ICU	480	192	85.0	48	42