INDIAN DATA CLUB

Day 14 (18/11): LEFT JOIN and RIGHT JOIN

🎯 *Objective*

To understand how **LEFT JOIN** and **RIGHT JOIN** work, how they handle matching and non-matching records, and when to use each join to extract complete and meaningful insights from relational datasets.

🔍 *Topics Covered*

- Concept and purpose of **LEFT JOIN**
- Concept and purpose of **RIGHT JOIN**
- Difference between LEFT and RIGHT JOIN
- Identifying matched vs. unmatched records
- Using ON conditions effectively
- Real-world scenarios where LEFT/RIGHT JOIN are commonly used

INDIAN DATA CLUB

Left Join OR Left outer join

LEFT JOIN returns all records from the **left table**, and the **matched records** from the **right table**.
Both **LEFT JOIN** and **LEFT OUTER JOIN** mean the same thing.
OUTER is optional and commonly omitted.

Syntax :-

```
SELECT table1.column1, table2.column2
FROM table1
LEFT JOIN table2
ON table1.common_column = table2.common_column;
```

Syntax :-

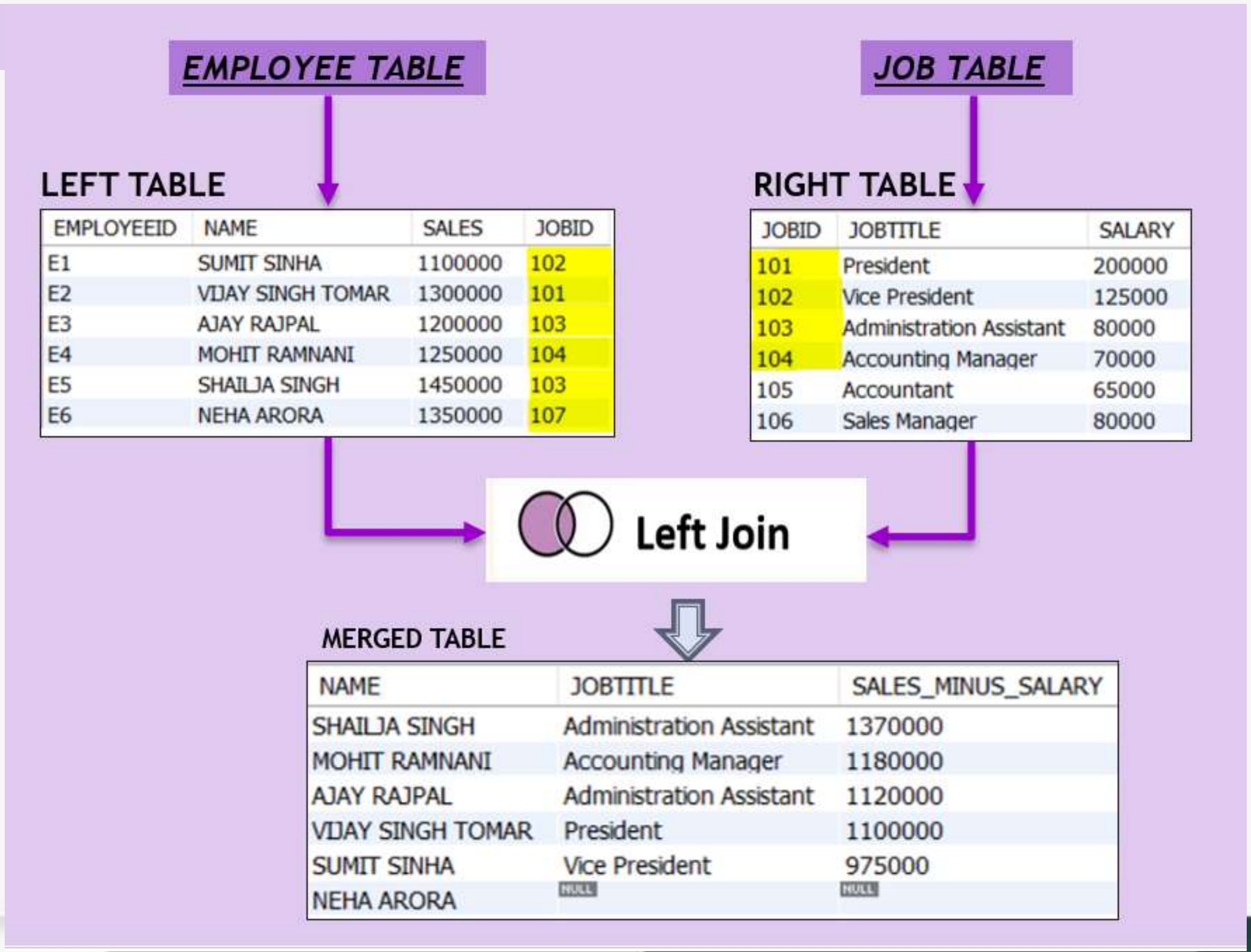
```
SELECT table1.column1, table2.column2
FROM table1
LEFT JOIN table2
USING(common_column);
```

Use ON when:
- The **column names are different** in the two tables.
- You need more flexibility in the join condition.

Use USING when:
- **The column name is the same in both tables.**
- **You want a cleaner, simpler join syntax.**

❖ These are our EMPLOYEE and JOB tables, which we are going to perform the LEFT JOIN / LEFT OUTER JOIN

RESULT :-



LEFT TABLE

EMPLOYEEID	NAME	SALES	JOBID
E1	SUMIT SINHA	1100000	102
E2	VIJAY SINGH TOMAR	1300000	101
E3	AJAY RAJPAL	1200000	103
E4	MOHIT RAMNANI	1250000	104
E5	SHAILJA SINGH	1450000	103
E6	NEHA ARORA	1350000	107

RIGHT TABLE

JOBID	JOBTITLE	SALARY
101	President	200000
102	Vice President	125000
103	Administration Assistant	80000
104	Accounting Manager	70000
105	Accountant	65000
106	Sales Manager	80000

MERGED TABLE

NAME	JOBTITLE	SALES_MINUS_SALARY
SHAILJA SINGH	Administration Assistant	1370000
MOHIT RAMNANI	Accounting Manager	1180000
AJAY RAJPAL	Administration Assistant	1120000
VIJAY SINGH TOMAR	President	1100000
SUMIT SINHA	Vice President	975000
NEHA ARORA	NULL	NULL

INDIAN DATA CLUB

Right Join OR Right outer join

A RIGHT JOIN returns all rows from the right table, and the matching rows from the left table. If there is no match on the left side, you'll see NULL values for the left table's columns. Note: RIGHT OUTER JOIN and RIGHT JOIN mean the same thing — "OUTER" is optional.

Syntax :-

```
SELECT table1.column1, table2.column2
FROM table1
RIGHT JOIN table2
ON table1.common_column = table2.common_column;
```

Syntax :-

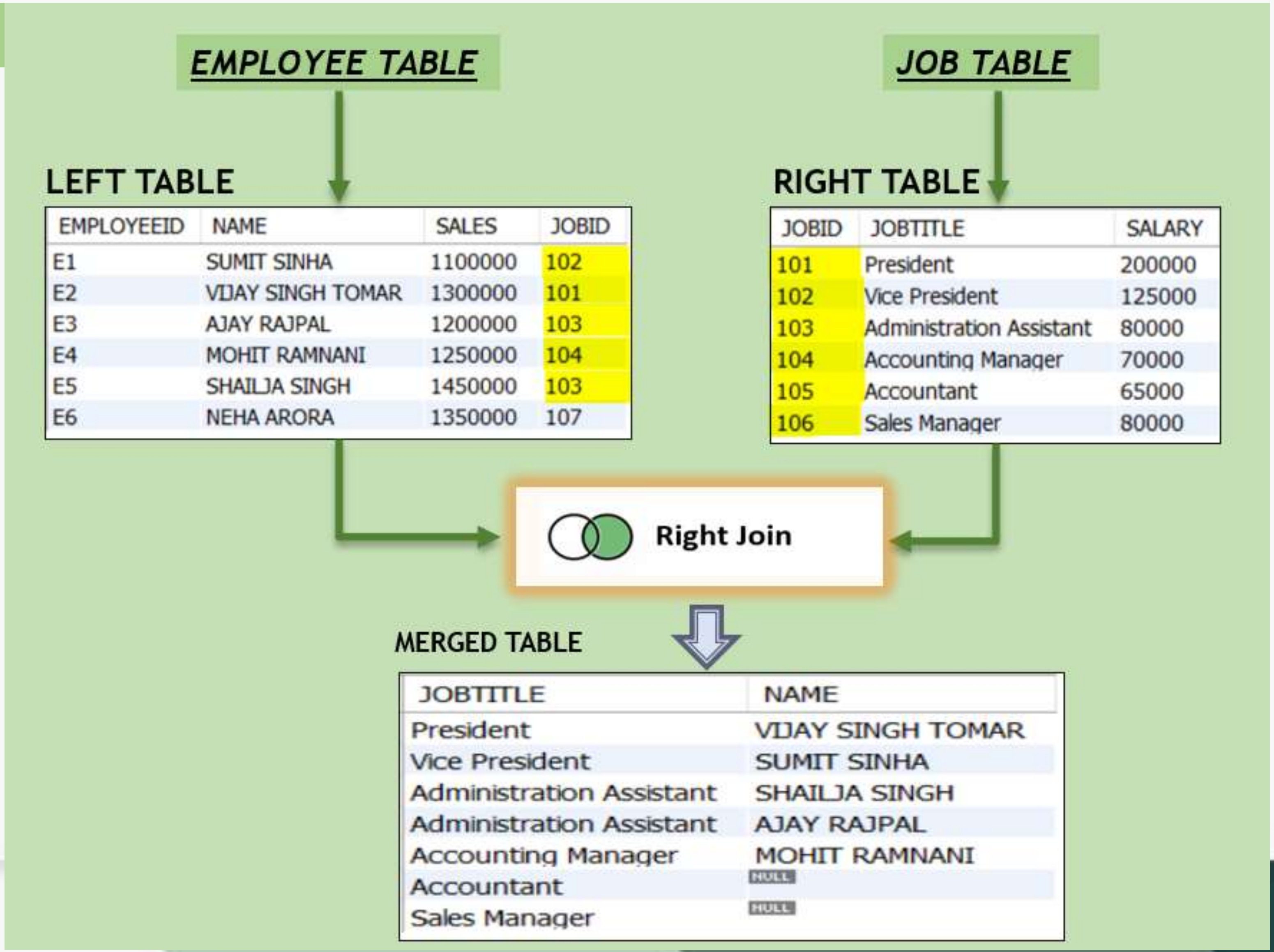
```
SELECT table1.column1, table2.column2
FROM table1
RIGHT JOIN table2
USING(common_column);
```

Use ON when:
- The **column names are different** in the two tables.
- You need more flexibility in the join condition.

Use USING when:
- **The column name is the same in both tables.**
- **You want a cleaner, simpler join syntax.**

❖ These are our EMPLOYEE and JOB tables, which we are going to perform the RIGHT JOIN / RIGHT OUTER JOIN

RESULT :-



LEFT TABLE

EMPLOYEEID	NAME	SALES	JOBID
E1	SUMIT SINHA	1100000	102
E2	VIJAY SINGH TOMAR	1300000	101
E3	AJAY RAJPAL	1200000	103
E4	MOHIT RAMNANI	1250000	104
E5	SHAILJA SINGH	1450000	103
E6	NEHA ARORA	1350000	107

RIGHT TABLE

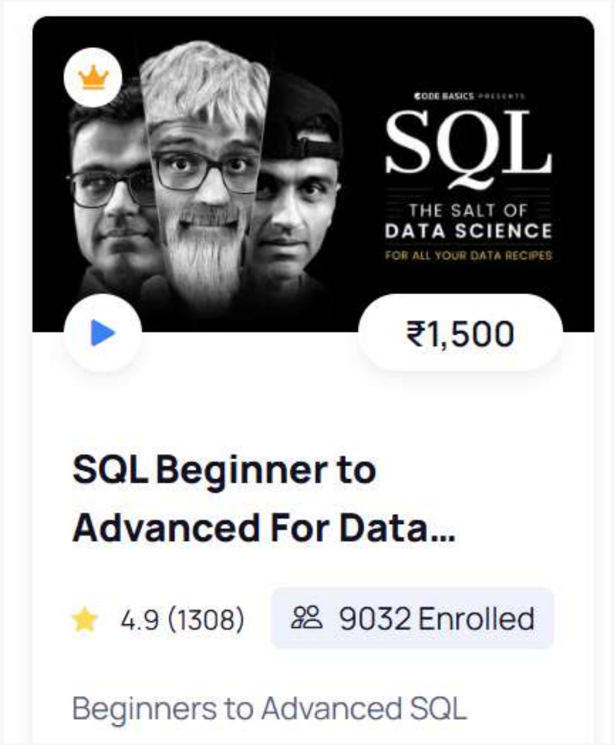
JOBID	JOBTITLE	SALARY
101	President	200000
102	Vice President	125000
103	Administration Assistant	80000
104	Accounting Manager	70000
105	Accountant	65000
106	Sales Manager	80000

MERGED TABLE

JOBTITLE	NAME
President	VIJAY SINGH TOMAR
Vice President	SUMIT SINHA
Administration Assistant	SHAILJA SINGH
Administration Assistant	AJAY RAJPAL
Accounting Manager	MOHIT RAMNANI
Accountant	NULL
Sales Manager	NULL

INDIAN DATA CLUB

📚 Resources:



₹1,500

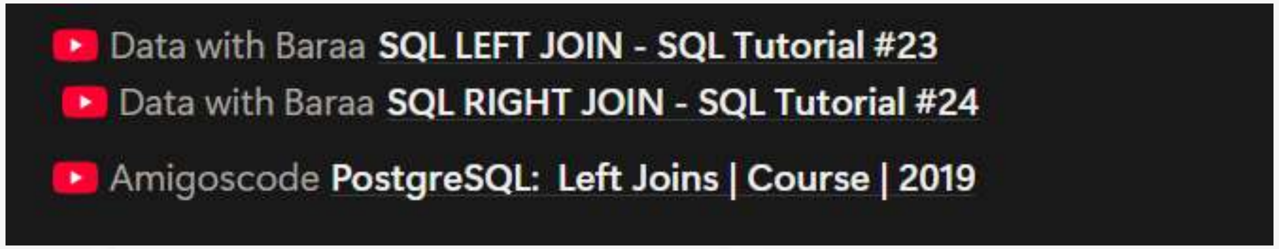
SQL Beginner to Advanced For Data...

4.9 (1308) 9032 Enrolled

Beginners to Advanced SQL

SQL Bootcamp Playlist (2025) - Zero to Hero

View full course

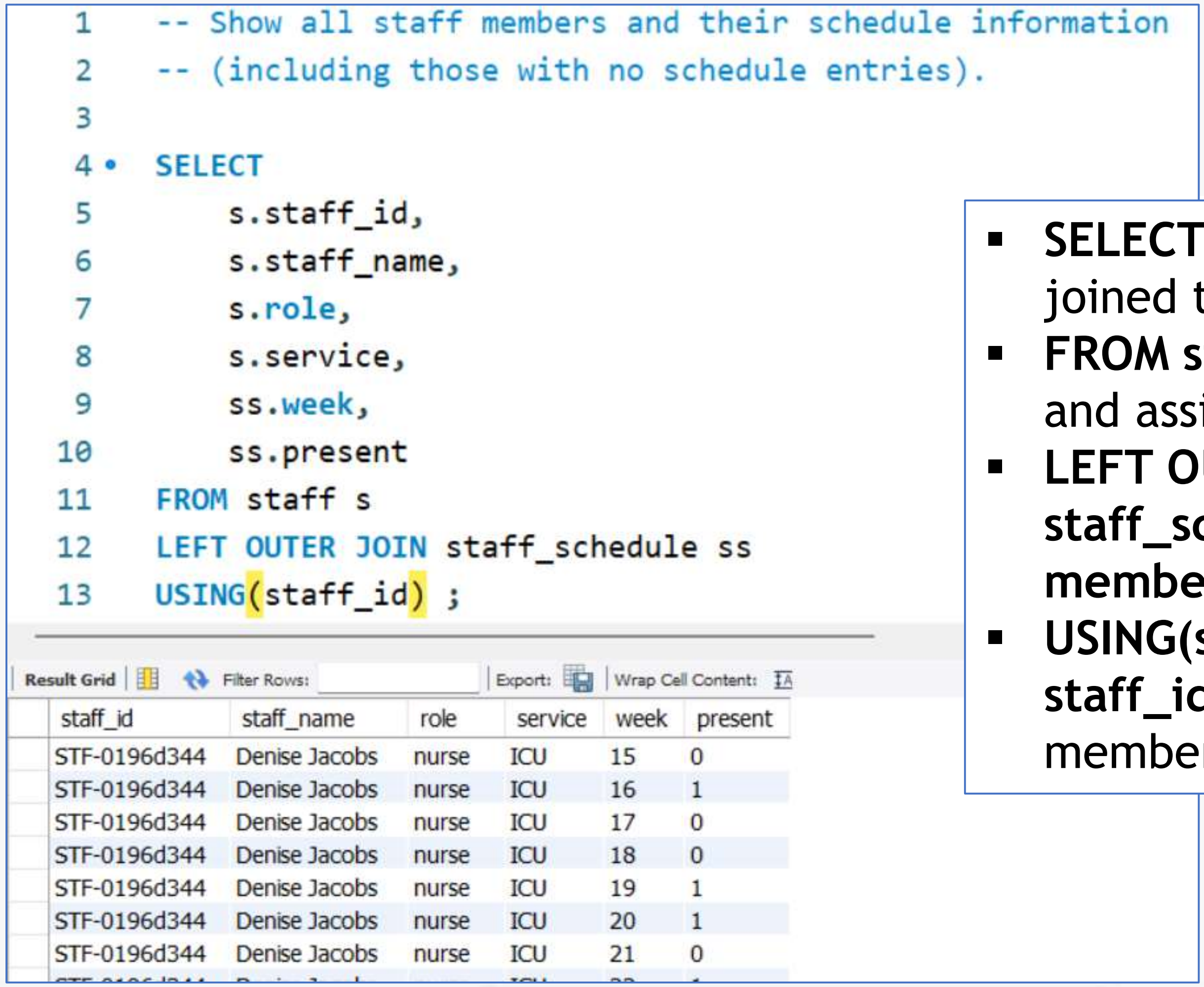


Data with Baraa SQL LEFT JOIN - SQL Tutorial #23

Data with Baraa SQL RIGHT JOIN - SQL Tutorial #24

Amigoscode PostgreSQL: Left Joins | Course | 2019

Practice Questions:



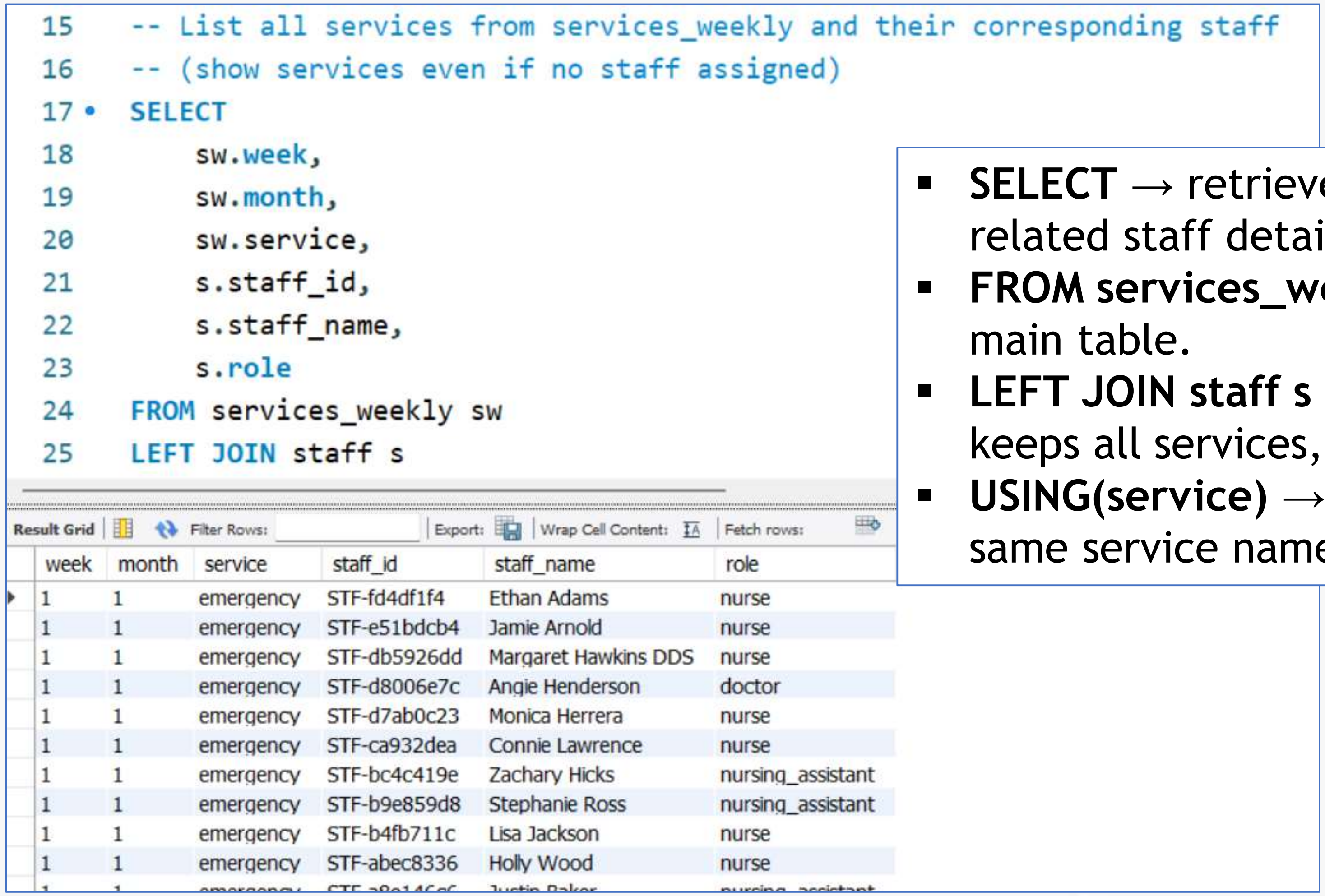
```
-- Show all staff members and their schedule information
-- (including those with no schedule entries).

SELECT
    s.staff_id,
    s.staff_name,
    s.role,
    s.service,
    ss.week,
    ss.present
FROM staff s
LEFT OUTER JOIN staff_schedule ss
USING(staff_id) ;
```

staff_id	staff_name	role	service	week	present
STF-0196d344	Denise Jacobs	nurse	ICU	15	0
STF-0196d344	Denise Jacobs	nurse	ICU	16	1
STF-0196d344	Denise Jacobs	nurse	ICU	17	0
STF-0196d344	Denise Jacobs	nurse	ICU	18	0
STF-0196d344	Denise Jacobs	nurse	ICU	19	1
STF-0196d344	Denise Jacobs	nurse	ICU	20	1
STF-0196d344	Denise Jacobs	nurse	ICU	21	0

- **SELECT** → retrieves selected columns from the joined tables.
- **FROM staff s** → selects data from the **staff** table and assigns it the alias **s**.
- **LEFT OUTER JOIN staff_schedule ss** → joins the **staff_schedule** table while keeping **all staff members**, even if they have no schedule.
- **USING(staff_id)** → matches rows based on the same **staff_id** value in both tables, linking each staff member to their schedule records.

Practice Questions:

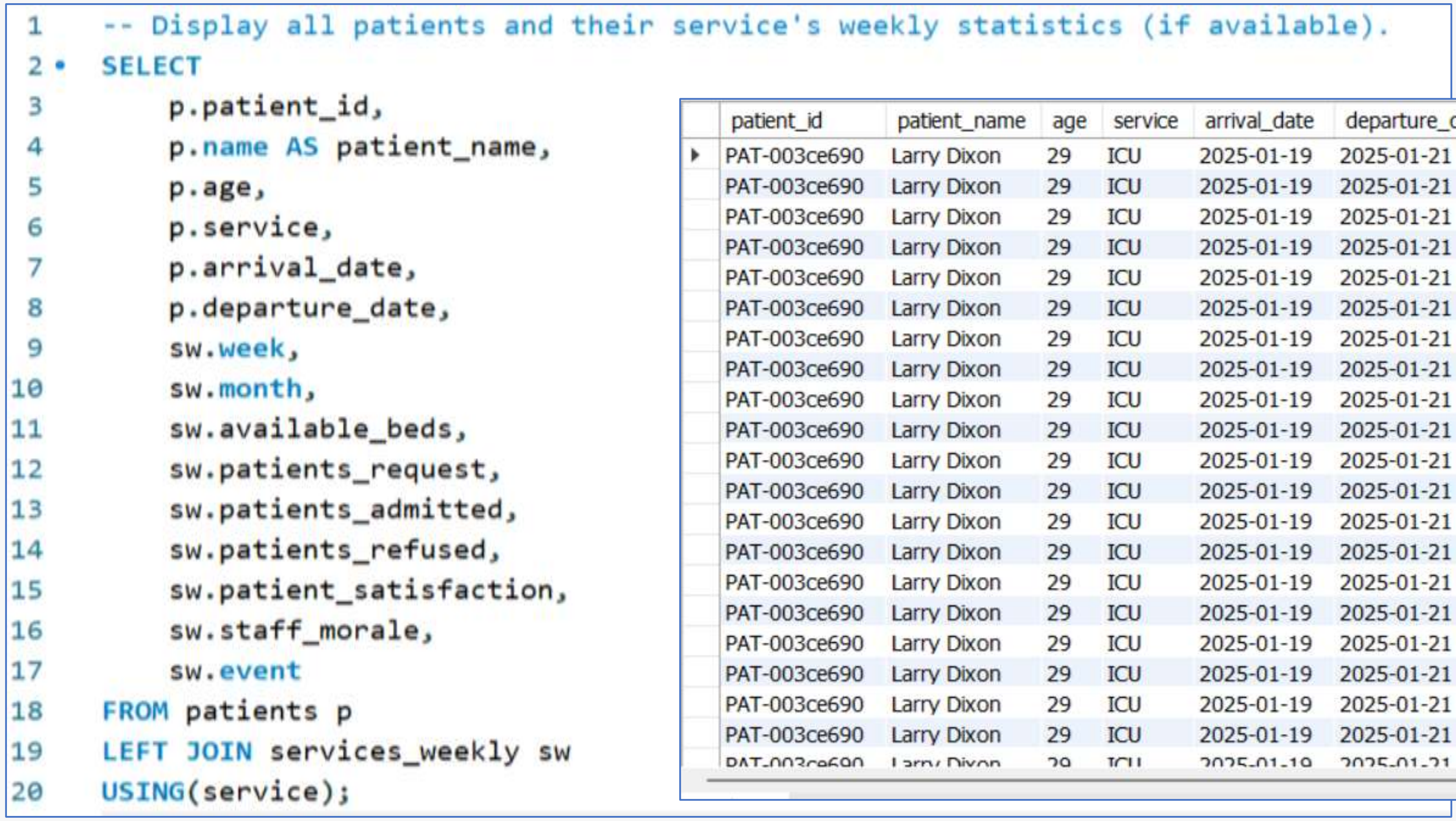


```sql
-- List all services from services_weekly and their corresponding staff
-- (show services even if no staff assigned)
SELECT
    sw.week,
    sw.month,
    sw.service,
    s.staff_id,
    s.staff_name,
    s.role
FROM services_weekly sw
LEFT JOIN staff s
```

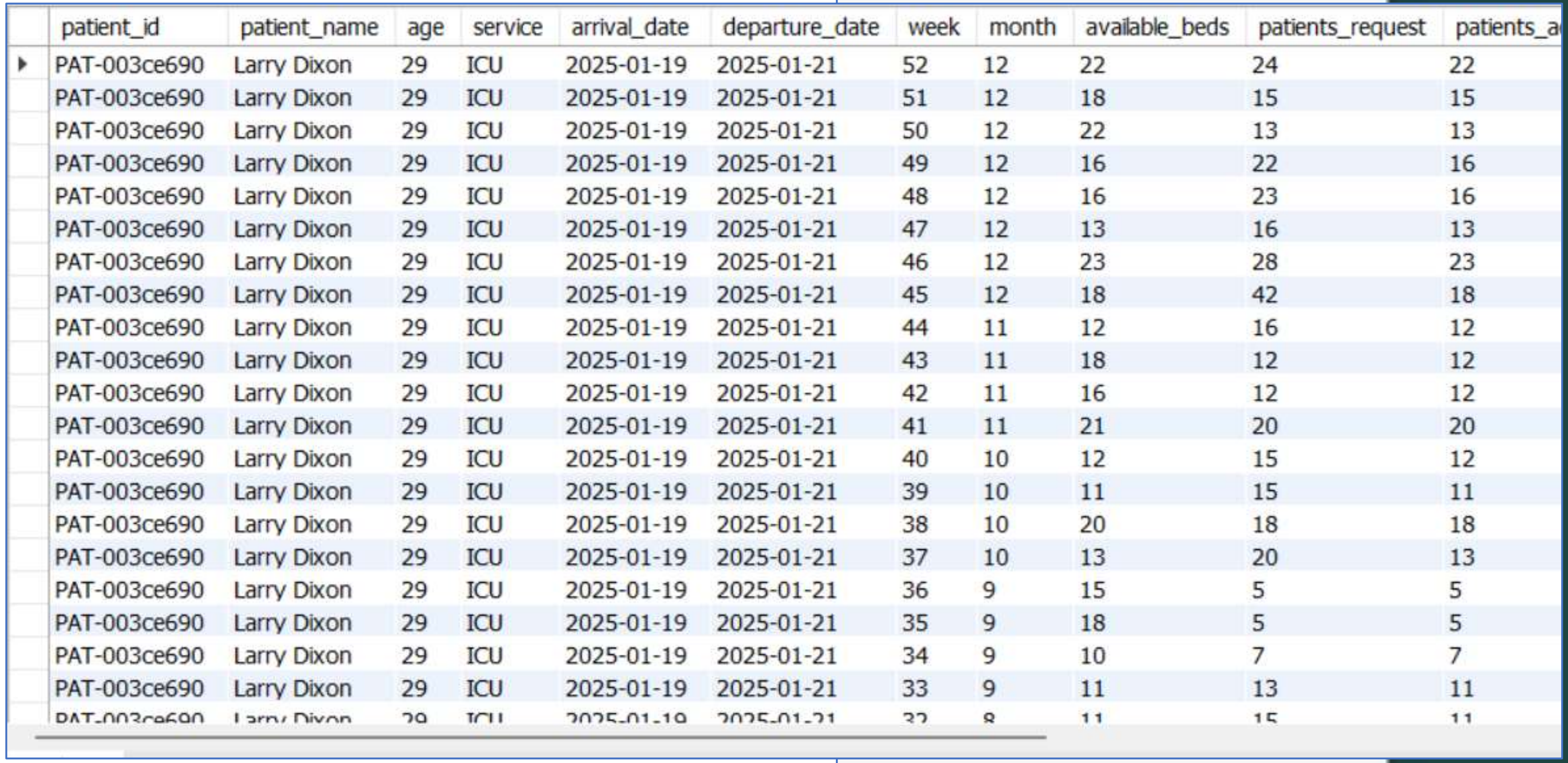
week	month	service	staff_id	staff_name	role
1	1	emergency	STF-fd4df1f4	Ethan Adams	nurse
1	1	emergency	STF-e51bdcb4	Jamie Arnold	nurse
1	1	emergency	STF-db5926dd	Margaret Hawkins DDS	nurse
1	1	emergency	STF-d8006e7c	Angie Henderson	doctor
1	1	emergency	STF-d7ab0c23	Monica Herrera	nurse
1	1	emergency	STF-ca932dea	Connie Lawrence	nurse
1	1	emergency	STF-bc4c419e	Zachary Hicks	nursing_assistant
1	1	emergency	STF-b9e859d8	Stephanie Ross	nursing_assistant
1	1	emergency	STF-b4fb711c	Lisa Jackson	nurse
1	1	emergency	STF-abec8336	Holly Wood	nurse

- **SELECT** → retrieves the week, month, service, and related staff details.
- **FROM services_weekly sw** → takes all services as the main table.
- **LEFT JOIN staff s** → brings in staff information but keeps all services, even if no staff is assigned.
- **USING(service)** → matches both tables based on the same service name.

Practice Questions:



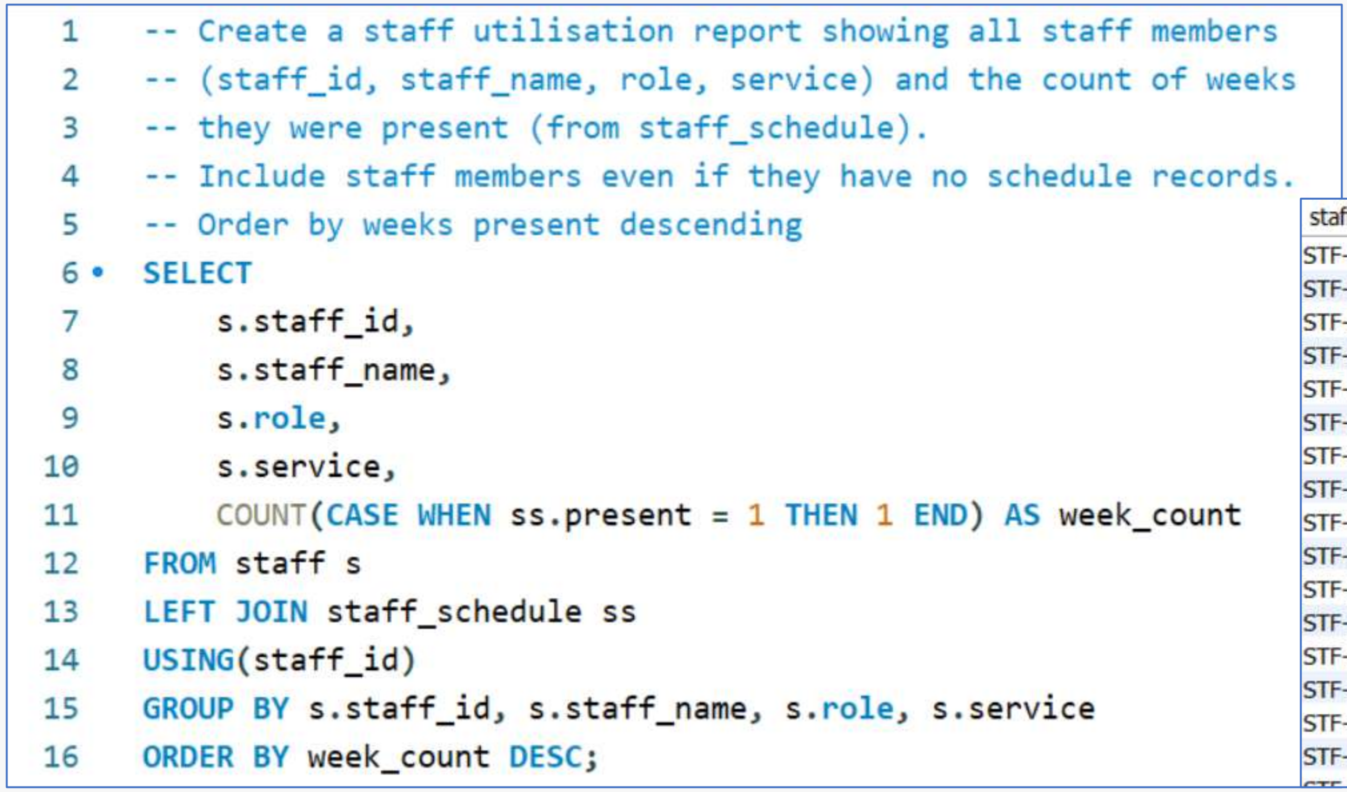
```sql
-- Display all patients and their service's weekly statistics (if available).
SELECT
    p.patient_id,
    p.name AS patient_name,
    p.age,
    p.service,
    p.arrival_date,
    p.departure_date,
    sw.week,
    sw.month,
    sw.available_beds,
    sw.patients_request,
    sw.patients_admitted,
    sw.patients_refused,
    sw.patient_satisfaction,
    sw.staff_morale,
    sw.event
FROM patients p
LEFT JOIN services_weekly sw
USING(service);
```



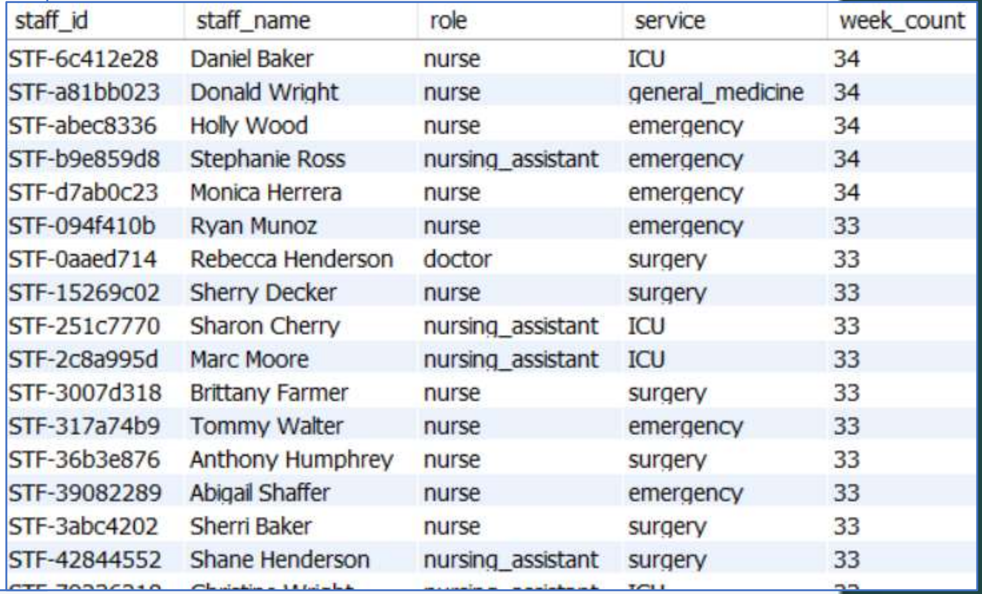
patient_id	patient_name	age	service	arrival_date	departure_date	week	month	available_beds	patients_request	patients_a
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	52	12	22	24	22
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	51	12	18	15	15
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	50	12	22	13	13
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	49	12	16	22	16
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	48	12	16	23	16
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	47	12	13	16	13
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	46	12	23	28	23
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	45	12	18	42	18
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	44	11	12	16	12
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	43	11	18	12	12
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	42	11	16	12	12
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	41	11	21	20	20
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	40	10	12	15	12
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	39	10	11	15	11
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	38	10	20	18	18
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	37	10	13	20	13
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	36	9	15	5	5
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	35	9	18	5	5
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	34	9	10	7	7
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	33	9	11	13	11
PAT-003ce690	Larry Dixon	29	ICU	2025-01-19	2025-01-21	32	8	11	15	11

- **SELECT** → retrieves patient info along with weekly service statistics.
- **FROM patients p** → uses patients as the main table so all patients appear.
- **LEFT JOIN services_weekly sw** → brings service statistics when they exist.
- **USING(service)** → matches both tables by the common service name.

Daily Challenge:



```sql
-- Create a staff utilisation report showing all staff members
-- (staff_id, staff_name, role, service) and the count of weeks
-- they were present (from staff_schedule).
-- Include staff members even if they have no schedule records.
-- Order by weeks present descending
SELECT
    s.staff_id,
    s.staff_name,
    s.role,
    s.service,
    COUNT(CASE WHEN ss.present = 1 THEN 1 END) AS week_count
FROM staff s
LEFT JOIN staff_schedule ss
USING(staff_id)
GROUP BY s.staff_id, s.staff_name, s.role, s.service
ORDER BY week_count DESC;
```



staff_id	staff_name	role	service	week_count
STF-6c412e28	Daniel Baker	nurse	ICU	34
STF-a81bb023	Donald Wright	nurse	general_medicine	34
STF-abec8336	Holly Wood	nurse	emergency	34
STF-b9e859d8	Stephanie Ross	nursing_assistant	emergency	34
STF-d7ab0c23	Monica Herrera	nurse	emergency	34
STF-094f410b	Ryan Munoz	nurse	emergency	33
STF-0aaed714	Rebecca Henderson	doctor	surgery	33
STF-15269c02	Sherry Decker	nurse	surgery	33
STF-251c7770	Sharon Cherry	nursing_assistant	ICU	33
STF-2c8a995d	Marc Moore	nursing_assistant	ICU	33
STF-3007d318	Brittany Farmer	nurse	surgery	33
STF-317a74b9	Tommy Walter	nurse	emergency	33
STF-36b3e876	Anthony Humphrey	nurse	surgery	33
STF-39082289	Abigail Shaffer	nurse	emergency	33
STF-3abc4202	Sherri Baker	nurse	surgery	33
STF-42844552	Shane Henderson	nursing_assistant	surgery	33

- **SELECT** → shows each staff member and how many weeks they were present.
- **FROM staff s** → staff is the main table so all staff appear.
- **LEFT JOIN staff_schedule ss** → adds schedule data but keeps staff with no records.
- **COUNT(CASE WHEN ss.present = 1 THEN 1 END)** → counts only the weeks the staff was present.
- **GROUP BY** → groups results by each staff member.
- **ORDER BY week_count DESC** → sorts staff by highest to lowest presence.