# Self JOIN in MySQL

A **Self JOIN** is a regular join, but the table is joined **with itself**.

This is useful when rows in the same table are related to each other. For example, when users refer other users, and we store the ID of the person who referred them in the same `users` table.

---

## Step 1: Add a `referred_by_id` Column

We'll extend the existing `users` table to include a column called `referred_by_id`, which holds the `id` of the user who referred them.

```
ALTER TABLE users
ADD COLUMN referred_by_id INT;
```

This column:

- Will be **NULL** for users who were not referred.
- Will contain the **`id` of another user** who referred them.

---

## Step 2: Insert Referral Data (Optional)

Assuming `id = 1` is the first user, and referred others:

```
UPDATE users SET referred_by_id = 1 WHERE id IN (2, 3); -- User 1 referred Users 2 and 3
UPDATE users SET referred_by_id = 2 WHERE id = 4;        -- User 2 referred User 4
```

---

## Step 3: Use a Self JOIN to Get Referrer Names

We want to get each user’s name **along with the name of the person who referred them**.

```
SELECT
  a.id,
  a.name AS user_name,
  b.name AS referred_by
FROM users a
INNER JOIN users b ON a.referred_by_id = b.id;
```

### Explanation:

- `a` refers to the user being queried.
- `b` refers to the user who referred them.
- `LEFT JOIN` is used so that users with `NULL` in `referred_by_id` are also included.

---

### Sample Output:

| id | user_name | referred_by |
|---|---|---|
| 1 | Aarav | NULL |
| 2 | Sneha | Aarav |
| 3 | Raj | Aarav |
| 4 | Fatima | Sneha |

---

## Summary

- Use **Self JOIN** when you need to **join a table with itself**.
- In referral-based relationships, store the referrer’s `id` in the same table.

- Use aliases like `a` and `b` to differentiate the two instances of the same table.