

SELF JOIN

Joins in SQL, a self join is a regular join that is used to join a table with itself. It basically allows us to combine the rows from the same table based on some specific conditions. It is very useful and easy to work with, and it allows us to retrieve data or information which involves comparing records within the same table.

Syntax

```
SELECT columns
FROM table AS alias1
JOIN table AS alias2
ON alias1.column = alias2.column;
```

Explanation

- **SELECT columns:** With the help of this we specify the columns you want to retrieve from the self-joined table.
- **FROM table AS alias1:** With the help of this we specify the name of the table you want to join with itself.
- **JOIN table AS alias2:** In this we use the JOIN keyword to show that we are performing a self join on the same table.

Example

Let's use an illustration to further understand how the self-join functions. Assume that we have a table called "GFGemployees" with the columns employee_id, employee_name, and manager_id. Each employee in the company is assigned a manager, and using the manager-ids, we can identify each employee. We need to extract the list of employees along with the names of their managers because the manager_id column contains the manager ID for each employee.

Step 1 First, we need to create the "GFGemployees" table with following query.

```
CREATE TABLE GFGemployees(employee_id INT PRIMARY KEY,
employee_name VARCHAR(50), manager_id INT);
```

Step 2 Now we will add data into the 'GFGemployees' table using INSERT INTO statement:

```
INSERT INTO GFGemployees (employee_id, employee_name,
manager_id)
VALUES
(1, 'Zaid', 3),
(2, 'Rahul', 3),
(3, 'Raman', 4),
(4, 'Kamran', NULL),
(5, 'Farhan', 4);
```

Output

employee_id	employee_name	manager_id
1	Zaid	3
2	Rahul	3
3	Raman	4
4	Kamran	
5	Farhan	4

Step 3 Explanation and implementation of Self Join

Now, we need to perform self join on the table we created i.e.”GFGemployees” in order to retrieve the list of employees and their corresponding managers name and for that we need to write a query, where we will create two different aliases for the “GFGemployees” table as “e” which will represent the GFG employee’s information and “m” will represent the manager’s information. This way by joining the table with itself using the manager_id and employee_id columns, we can generate relationship between employees and their managers.

Step 4 Query for Self-join

```
SELECT e.employee_name AS employee,  
m.employee_name AS manager  
FROM  
GFGemployees AS e  
JOIN GFGemployees  
AS m  
ON e.manager_id = m.employee_id;
```

Output

The resultant table after performing self join will be as follows:

employee	manager
Zaid	Raman
Rahul	Raman
Raman	Kamran
Farhan	Kamran