

SQL SELF JOINS

By Randy Leon



Steps One and Two:

1. Create the table. Each row should minimally include the person's name, their supervisor, and their job title. Using ID columns is encouraged.
2. Populate the table with some rows.

```
1  --Gene sequencing firm.
2
3  CREATE TABLE employee (
4      employee_id INT PRIMARY KEY,
5      first_name VARCHAR (255) NOT NULL,
6      last_name VARCHAR (255) NOT NULL,
7      job_title VARCHAR (255) NOT NULL,
8      manager_id INT,
9      FOREIGN KEY (manager_id)
10     REFERENCES employee (employee_id)
11     ON DELETE CASCADE
12 );
13 INSERT INTO employee (
14     employee_id,
15     first_name,
16     last_name,
17     job_title,
18     manager_id
19 )
20 VALUES
21     (1, 'Windy', 'Hays', 'CEO', NULL),
22     (2, 'Ava', 'Christensen', 'VP', 1),
23     (3, 'Hassan', 'Conner', 'VP', 1),
24     (4, 'Anna', 'Reeves', 'Scientist', 2),
25     (5, 'Sau', 'Norman', 'Scientist', 2),
26     (6, 'Kelsie', 'Hays', 'Scientist', 3),
27     (7, 'Tory', 'Goff', 'Scientist', 3),
28     (8, 'Salley', 'Lester', 'Scientist', 3);
```

Explain Messages Notifications

INSERT @ 8

Query returned successfully in 70 msec.

Step Three:

3. Provide a single `SELECT` statement that displays the information in the table, showing who reports to whom.



Employee_assignment/postgres@PostgreSQL 11

Query Editor

Query History

Data Output

89
90
91
92
93
94
95
96
97
98
99
100
101

```
SELECT * FROM employee;
```

	employee_id [PK] integer	first_name character varying (255)	last_name character varying (255)	job_title character varying (255)	manager_id integer
1	1	Windy	Hays	CEO	[null]
2	2	Ava	Christensen	VP	1
3	3	Hassan	Conner	VP	1
4	4	Anna	Reeves	Scientist	2
5	5	Sau	Norman	Scientist	2
6	6	Kelsie	Hays	Scientist	3
7	7	Tory	Goff	Scientist	3
8	8	Salley	Lester	Scientist	3

I would like to make known that I found out how to do this assignment via this Postgres Tutorial: <https://www.postgresqltutorial.com/postgresql-self-join/>

