

PostgreSQL Self-Join

Dashboard Properties SQL Statistics Dependencies Dependents **dvdrental/postgres@PostgreSQL 13 ***

Query Editor Query History

```
1 --This following statements create the employee table and insert some sample data into the table.
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     manager_id INT,
8     FOREIGN KEY (manager_id)
9     REFERENCES employee (employee_id)
10    ON DELETE CASCADE
11 );
12
```

Data Output Explain **Messages** Notifications

CREATE TABLE

Query returned successfully in 695 msec.

Dashboard Properties SQL Statistics Dependencies Dependents **dvdrental/postgres@PostgreSQL 13 ***

Query Editor Query History

```
1 INSERT INTO employee (
2     employee_id,
3     first_name,
4     last_name,
5     manager_id
6 )
7 VALUES
8     (1, 'Windy', 'Hays', NULL),
9     (2, 'Ava', 'Christensen', 1),
10    (3, 'Hassan', 'Conner', 1),
11    (4, 'Anna', 'Reeves', 2),
12    (5, 'Sau', 'Norman', 2),
13    (6, 'Kelsie', 'Hays', 3),
14    (7, 'Tory', 'Goff', 3),
15    (8, 'Salley', 'Lester', 3);
```

Data Output Explain **Messages** Notifications

INSERT 0 8

Query returned successfully in 206 msec.

Dashboard Properties SQL Statistics Dependencies Dependents dvdrental/postgres@PostgreSQL 13 *

Query Editor Query History

```

1  --This query uses the self-join to find who reports to whom
2  SELECT
3      e.first_name || ' ' || e.last_name employee,
4      m.first_name || ' ' || m.last_name manager
5  FROM
6      employee e
7  INNER JOIN employee m ON m.employee_id = e.manager_id
8  ORDER BY manager;
9

```

Data Output Explain Messages Notifications

	employee text	manager text
1	Sau Norman	Ava Christensen
2	Anna Reeves	Ava Christensen
3	Salley Lester	Hassan Conner
4	Kelsie Hays	Hassan Conner
5	Tory Goff	Hassan Conner
6	Ava Christensen	Windy Hays
7	Hassan Conner	Windy Hays

Dashboard Properties SQL Statistics Dependencies Dependents dvdrental/postgres@PostgreSQL 13 *

Query Editor Query History

```

1  --Notice that the top manager does not appear on the output.
2
3  --To include the top manager in the result set, you use
4  --the LEFT JOIN instead of INNER JOIN clause as shown in the following query
5  SELECT
6      e.first_name || ' ' || e.last_name employee,
7      m.first_name || ' ' || m.last_name manager
8  FROM
9      employee e
10 LEFT JOIN employee m ON m.employee_id = e.manager_id
11 ORDER BY manager;

```

Data Output Explain Messages Notifications

	employee text	manager text
1	Anna Reeves	Ava Christensen
2	Sau Norman	Ava Christensen
3	Salley Lester	Hassan Conner
4	Kelsie Hays	Hassan Conner
5	Tory Goff	Hassan Conner
6	Hassan Conner	Windy Hays
7	Ava Christensen	Windy Hays
8	Windy Hays	[null]

```
1  --The following query finds all pair of films that have the same length,
2  SELECT
3      f1.title,
4      f2.title,
5      f1.length
6  FROM
7      film f1
8  INNER JOIN film f2
9      ON f1.film_id <> f2.film_id AND
10     f1.length = f2.length;
```

	title character varying (255)	title character varying (255)	length smallint
1	Chamber Italian	Resurrection Silverado	117
2	Chamber Italian	Magic Mallrats	117
3	Chamber Italian	Graffiti Love	117
4	Chamber Italian	Affair Prejudice	117
5	Grosse Wonderful	Hurricane Affair	49
6	Grosse Wonderful	Hook Chariots	49
7	Grosse Wonderful	Heavenly Gun	49
8	Grosse Wonderful	Doors President	49
9	Airport Pollock	Sense Greek	

✓ Successfully run. Total query runtime: 2 secs 999 msec. 6972 rows affected