Introduction to SQL

Part-5

Hitesh Kumar Sharma

Instructor, Pluralsight

Proprietary and confidential

Complex Queries in PostgreSQL

PLURALSIGHT

Querying Multiple Tables (JOINS)

In PostgreSQL, the JOIN operation is used to combine rows from two or more tables based on a related column between them. There are different types of joins:

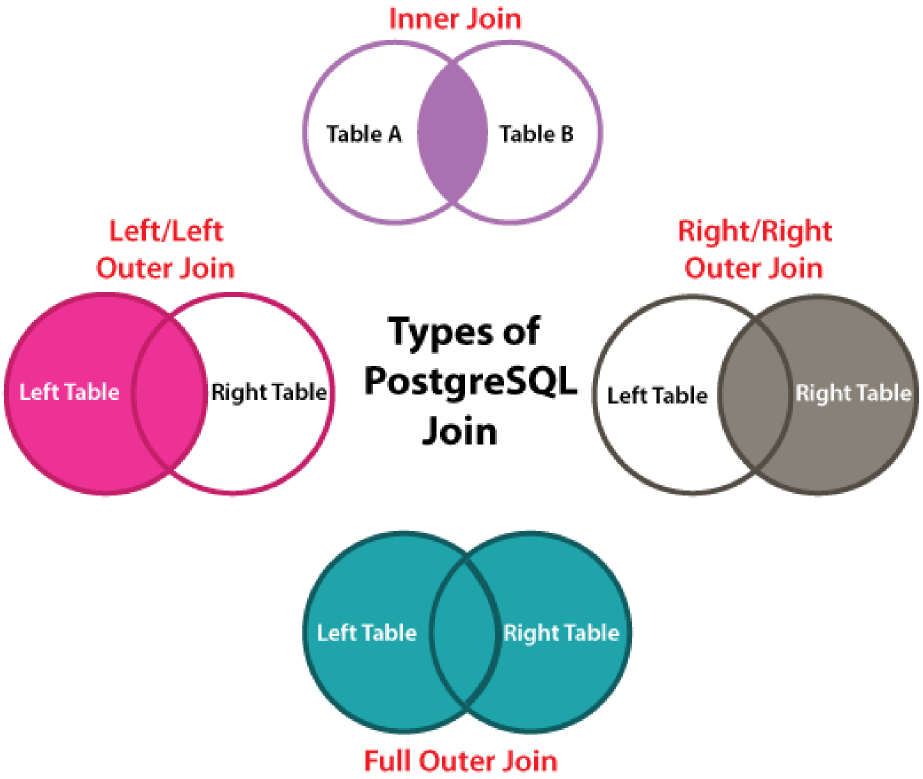
INNER JOIN: Returns records that have matching values in both tables.

```
SELECT * FROM table1 INNER JOIN table2 ON table1.column_name = table2.column_name;
```

LEFT JOIN (or LEFT OUTER JOIN): Returns all records from the left table and the matched records from the right table. If there are no matches, NULL values are used for the missing fields in the result set.

```
SELECT * FROM table1 LEFT JOIN table2 ON table1.column_name = table2.column_name;
```

Querying Multiple Tables (JOINS)



Querying Multiple Tables (JOINS)

RIGHT JOIN (or RIGHT OUTER JOIN): Returns all records from the right table and the matched records from the left table. If there are no matches, NULL values are used for the missing fields in the result set.

```
SELECT * FROM table1 RIGHT JOIN table2 ON table1.column_name = table2.column_name;
```

FULL JOIN (or FULL OUTER JOIN): Returns all records when there is a match in either the left or right table. If there are no matches, NULL values are used for the missing fields in the result set.

```
SELECT * FROM table1 FULL JOIN table2 ON table1.column_name = table2.column_name;
```

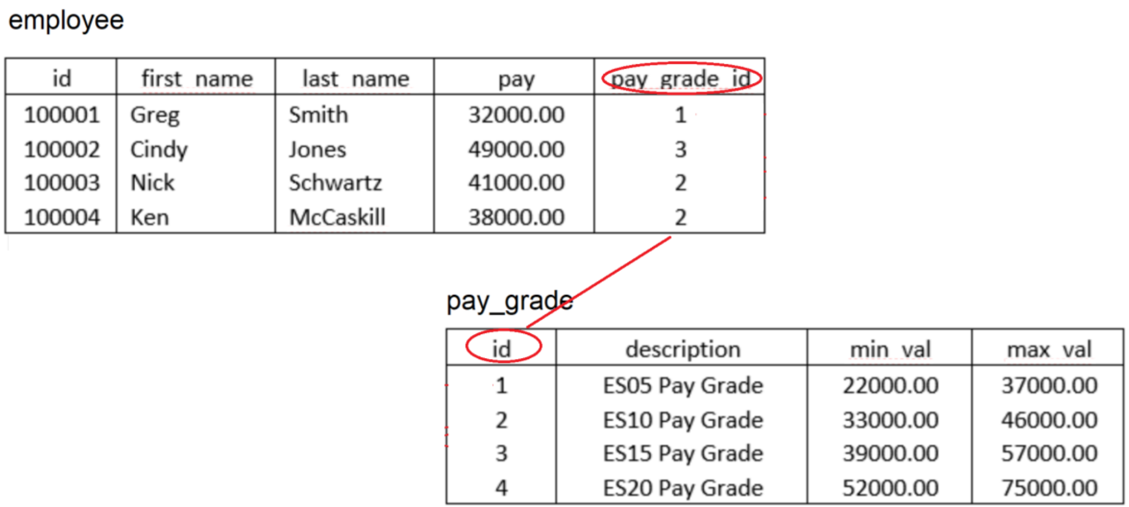
CROSS JOIN: Returns the Cartesian product of the sets of rows from the joined tables.

```
SELECT * FROM table1 CROSS JOIN table2;
```

Understanding and using these different types of joins in PostgreSQL can help you retrieve data that is spread across multiple tables and combine it in a meaningful way.

Querying Multiple Tables (JOINS)

- **A join is where you take a row from one table and *join* it to a row in another table based on some condition**
- The condition is usually matching a foreign key to a primary key



employee

id	first_name	last_name	pay	pay_grade_id
100001	Greg	Smith	32000.00	1
100002	Cindy	Jones	49000.00	3
100003	Nick	Schwartz	41000.00	2
100004	Ken	McCaskill	38000.00	2

pay_grade

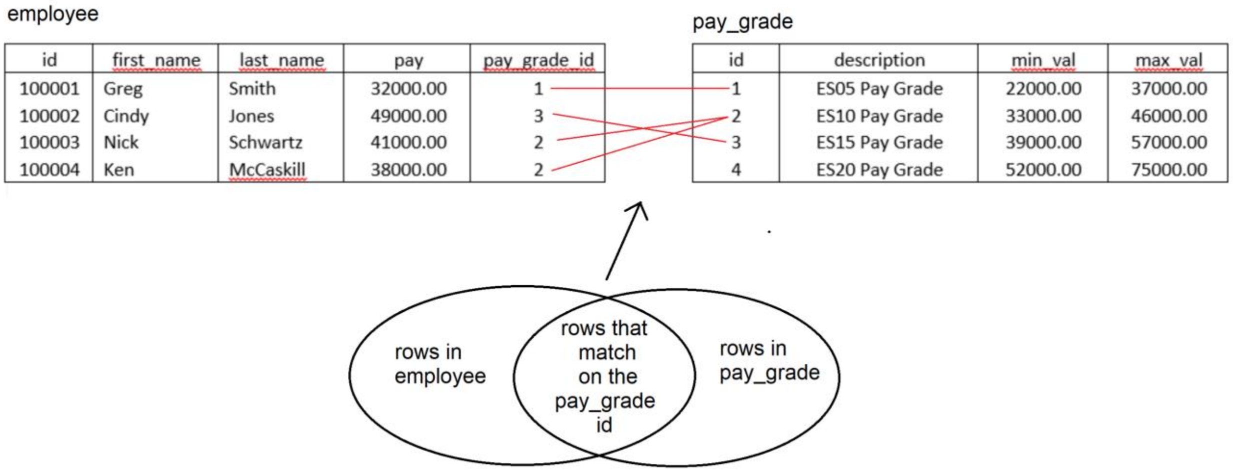
id	description	min_val	max_val
1	ES05 Pay Grade	22000.00	37000.00
2	ES10 Pay Grade	33000.00	46000.00
3	ES15 Pay Grade	39000.00	57000.00
4	ES20 Pay Grade	52000.00	75000.00

- **The result set is "new" rows containing columns from all of the tables in the join**

INNER JOIN

- With an INNER JOIN, a row in one table is joined with a row in another table based on a column match

- Only matched rows will be included in the result
  - If a row in the first table doesn't match any rows in the second table, it will be excluded
  - Similarly, if a row in the second table doesn't match a row in the first table, it will be excluded



employee

id	first_name	last_name	pay	pay_grade_id
100001	Greg	Smith	32000.00	1
100002	Cindy	Jones	49000.00	3
100003	Nick	Schwartz	41000.00	2
100004	Ken	McCaskill	38000.00	2

pay_grade

id	description	min_val	max_val
1	ES05 Pay Grade	22000.00	37000.00
2	ES10 Pay Grade	33000.00	46000.00
3	ES15 Pay Grade	39000.00	57000.00
4	ES20 Pay Grade	52000.00	75000.00

INNER JOIN

- The syntax for doing an INNER JOIN is a bit more complex than a simple SELECT because you must specify both tables and the join condition

  Syntax

```
SELECT column1, columns2, …
FROM table1
INNER JOIN table2
    ON table1.column-name = table2.column-name;
```

- **You don't have to include the word INNER**

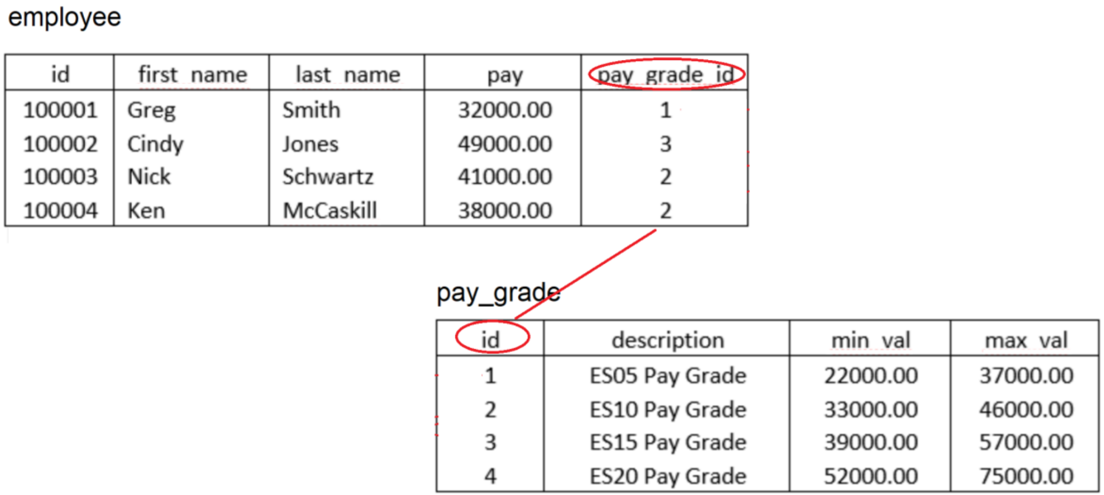
– **A JOIN is considered to be an INNER JOIN unless specifically written as one of the other types you will learn about**

  Syntax

```
SELECT column1, columns2, …
FROM table1
JOIN table2
  ON table1.column-name = table2.column-name;
```

INNER JOIN- Example

QUERY: We want to list each employee, along with their pay_grade description and the min/max salary of that pay grade.



employee

id	first_name	last_name	pay	pay_grade_id
100001	Greg	Smith	32000.00	1
100002	Cindy	Jones	49000.00	3
100003	Nick	Schwartz	41000.00	2
100004	Ken	McCaskill	38000.00	2

pay_grade

id	description	min_val	max_val
1	ES05 Pay Grade	22000.00	37000.00
2	ES10 Pay Grade	33000.00	46000.00
3	ES15 Pay Grade	39000.00	57000.00
4	ES20 Pay Grade	52000.00	75000.00

INNER JOIN- Example

APPROACH: Join the employee table to the pay_grade table and match employee.pay_grade_id to pay_grade.id

```
SELECT employee.id, first_name, last_name, description, min_val, max_val
FROM employee
JOIN pay_grade
  ON employee.pay_grade_id = pay_grade.id;
```

RESULTS:

```
+--------+------------+-----------+-----------------+-----------+-----------+
| id     | first_name | last_name | description     | min_val   | max_val   |
+--------+------------+-----------+-----------------+-----------+-----------+
| 100001 | Greg       | Smith     | ES05 Pay Grade | 22000.00  | 37000.00  |
| 100002 | Cindy      | Jones     | ES15 Pay Grade | 39000.00  | 46000.00  |
| 100003 | Nick       | Schwartz  | ES10 Pay Grade | 33000.00  | 46000.00  |
| 100004 | Ken        | McCaskill | ES10 Pay Grade | 33000.00  | 46000.00  |
+--------+------------+-----------+-----------------+-----------+-----------+
```

OUTER JOIN- Example

QUERY: What orders were sold when?

```
SELECT order.id, sold_date, name, email
FROM order
LEFT JOIN customer
       ON order.customer_id = customer.id;
```

RESULTS:

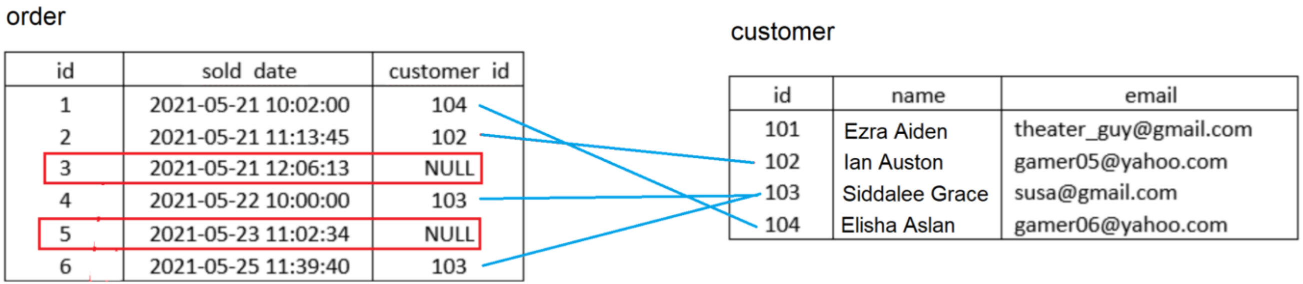
```
+-----+---------------------+------------------+--------------------+
| id  | sold_date           | name             | email              |
+-----+---------------------+------------------+--------------------+
| 1   | 2021-05-21 10:02:00 | Elisha Aslan     | gamer06@yahoo.com  |
| 2   | 2021-05-21 11:13:45 | Ian Auston       | gamer05@yahoo.com  |
| 3   | 2021-05-21 12:06:13 | NULL             | NULL               |
| 4   | 2021-05-22 10:00:00 | Siddalee Grace   | susa@gmail.com     |
| 5   | 2021-05-23 11:02:34 | NULL             | NULL               |
| 6   | 2021-05-25 11:39:40 | Siddalee Grace   | susa@gmail.com     |
+-----+---------------------+------------------+--------------------+
```

OUTER JOIN

- An OUTER JOIN is used when might want rows in one table that don't match rows in the other table
  – You must specify which of the two tables isn't required to have matching data in its JOIN column
- You do this by creating a LEFT OUTER JOIN or a RIGHT OUTER JOIN
  – The first table listed in the JOIN is considered to be on the left and the second is considered to be on the right
- For a LEFT OUTER JOIN, the first table doesn't require matching data to be included
  – Selected columns without a matching value will be NULL

OUTER JOIN- Example

QUERY: What orders were sold when?



order

id	sold_date	customer_id
1	2021-05-21 10:02:00	104
2	2021-05-21 11:13:45	102
3	2021-05-21 12:06:13	NULL
4	2021-05-22 10:00:00	103
5	2021-05-23 11:02:34	NULL
6	2021-05-25 11:39:40	103

customer

id	name	email
101	Ezra Aiden	theater_guy@gmail.com
102	Ian Auston	gamer05@yahoo.com
103	Siddalee Grace	susa@gmail.com
104	Elisha Aslan	gamer06@yahoo.com

orders 3 and 5 are included
in a LEFT OUTER JOIN

PLURALSIGHT

Hands-On Lab Exercise-17

(Topic: Character Functions)

PLURALSIGHT