Subquerying with semi joins and anti joins

JOINING DATA IN SQL
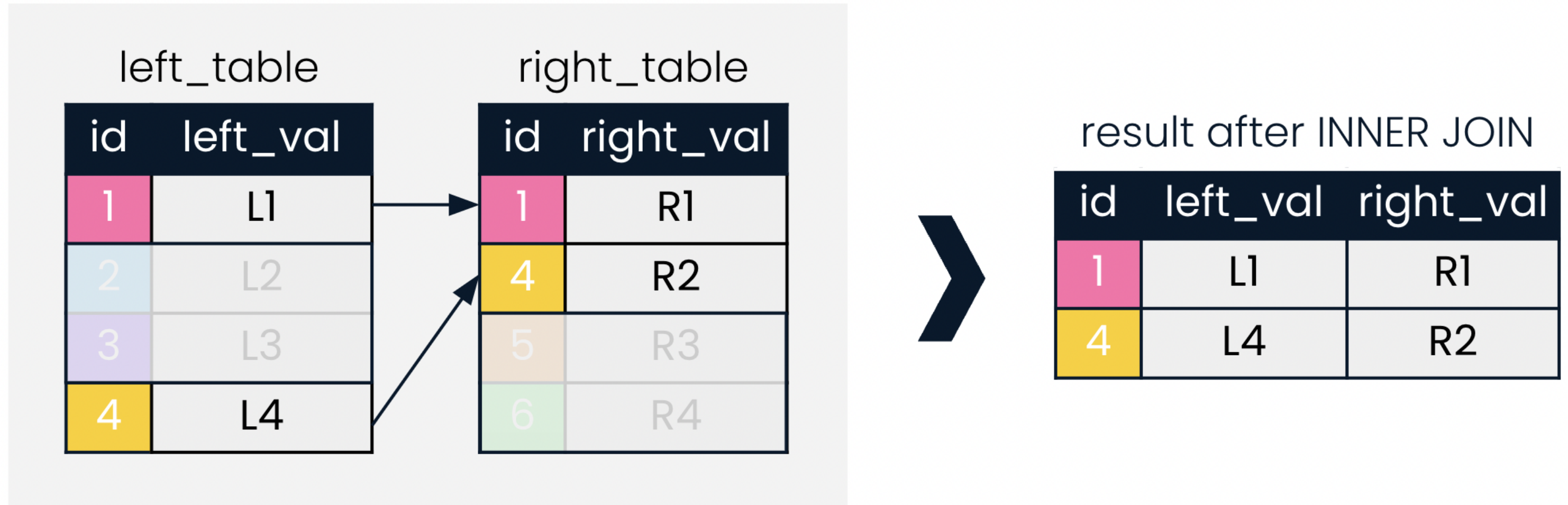
SQL

Maham Faisal Khan
Senior Content Developer, DataCamp
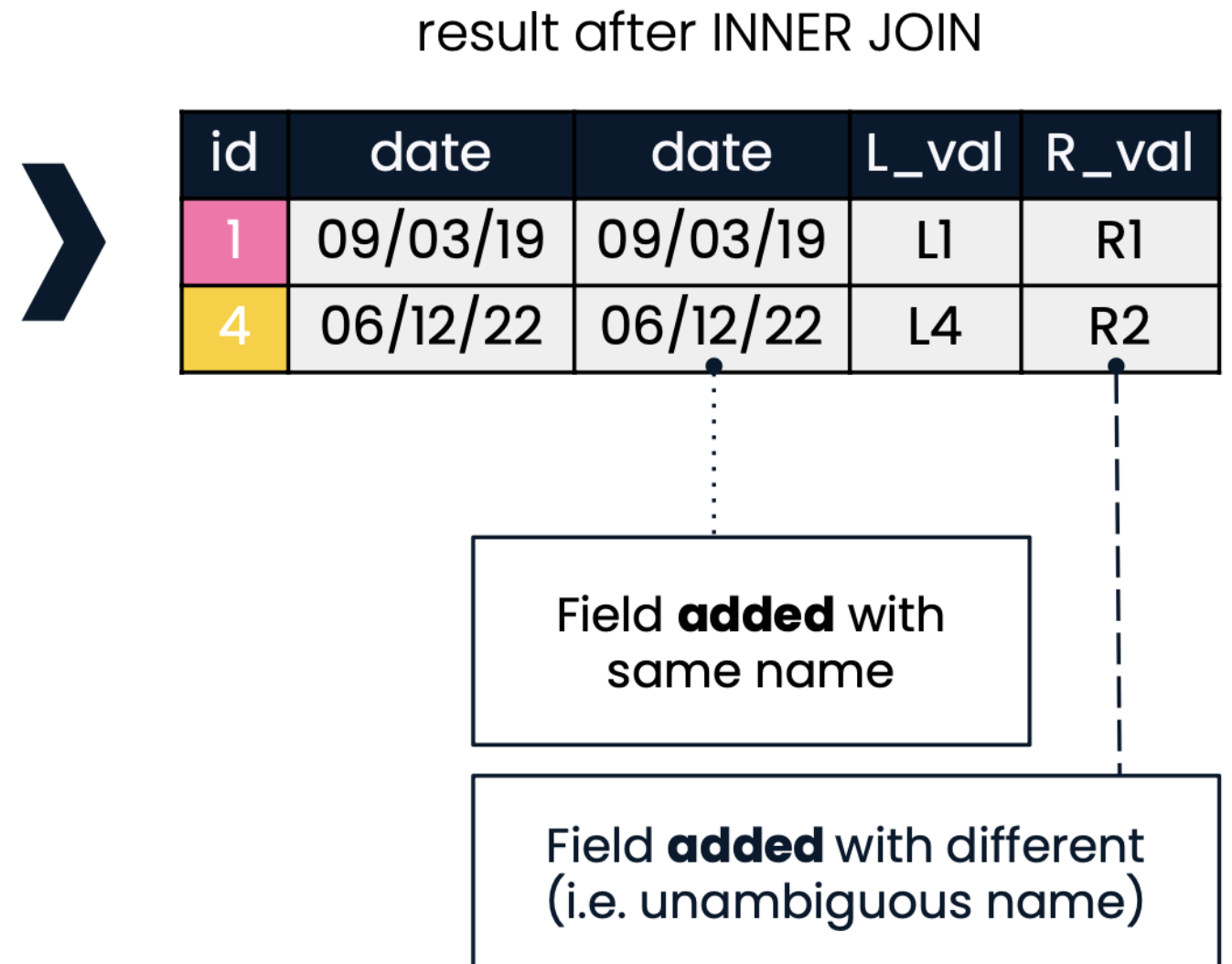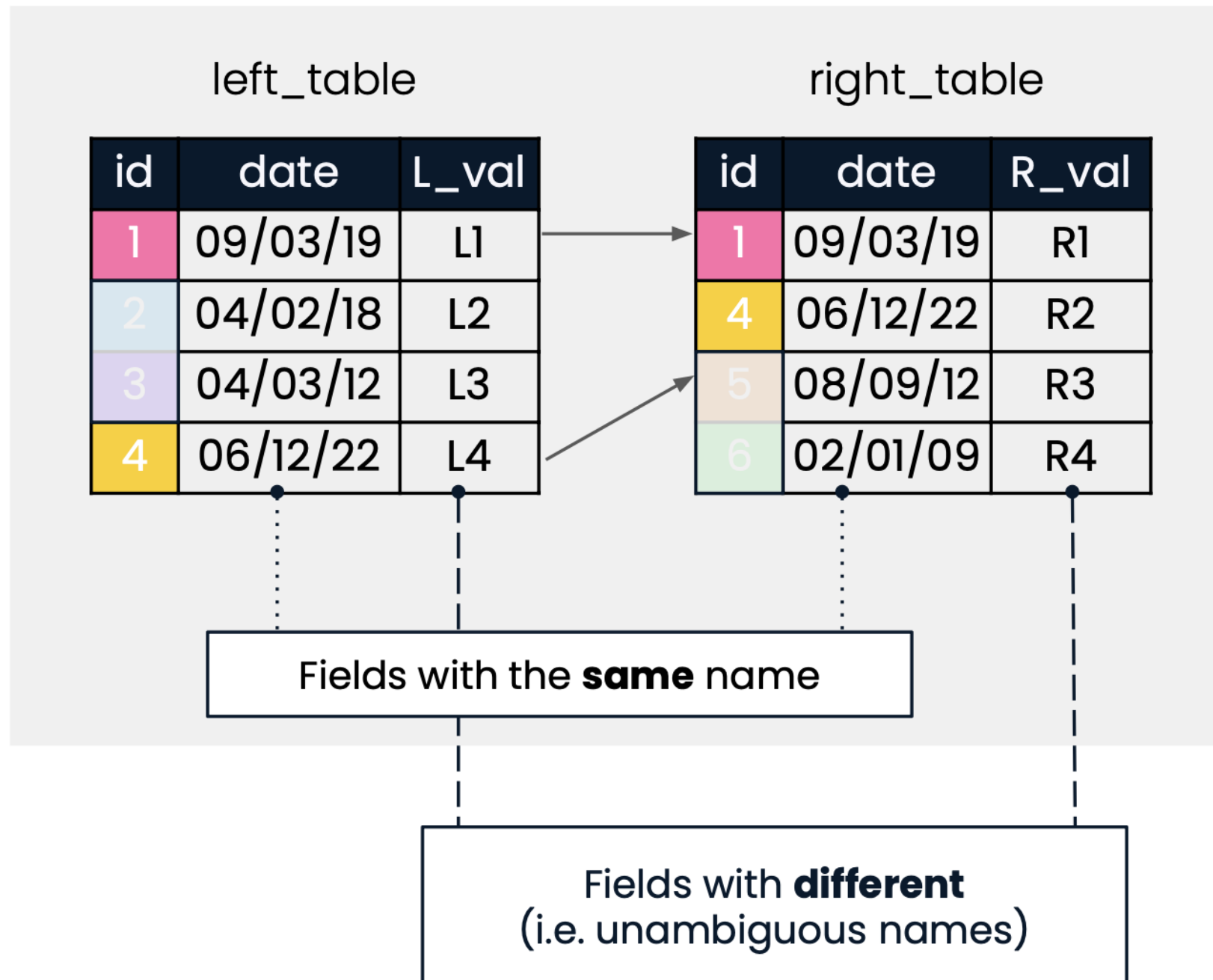
datacamp

Calling all joins

Diagram for an `INNER JOIN` **on the** `id` **field**

Additive joins

```sql
SELECT *
FROM left_table
INNER JOIN right_table
ON left_table.id = right_table.id;
```

Additive joins

Semi join

A **semi join** chooses records in the first table where a condition is met in the second table.

left_table

id	col1
1	A
2	B
3	C
4	D

right_table

col2
B
C

Semi join

left_table

id	col1
1	A
2	B
3	C
4	D

right_table

col2
B
C

Return all values from `left_table` where values of `col1` are in `col2`

Semi join

left_table

id	col1
1	A
2	B
3	C
4	D

right_table

col2
B
C

left_table
after semi-join

id	col1
2	B
3	C

Return all values from
`left_table` where values of
`col1` are in `col2`
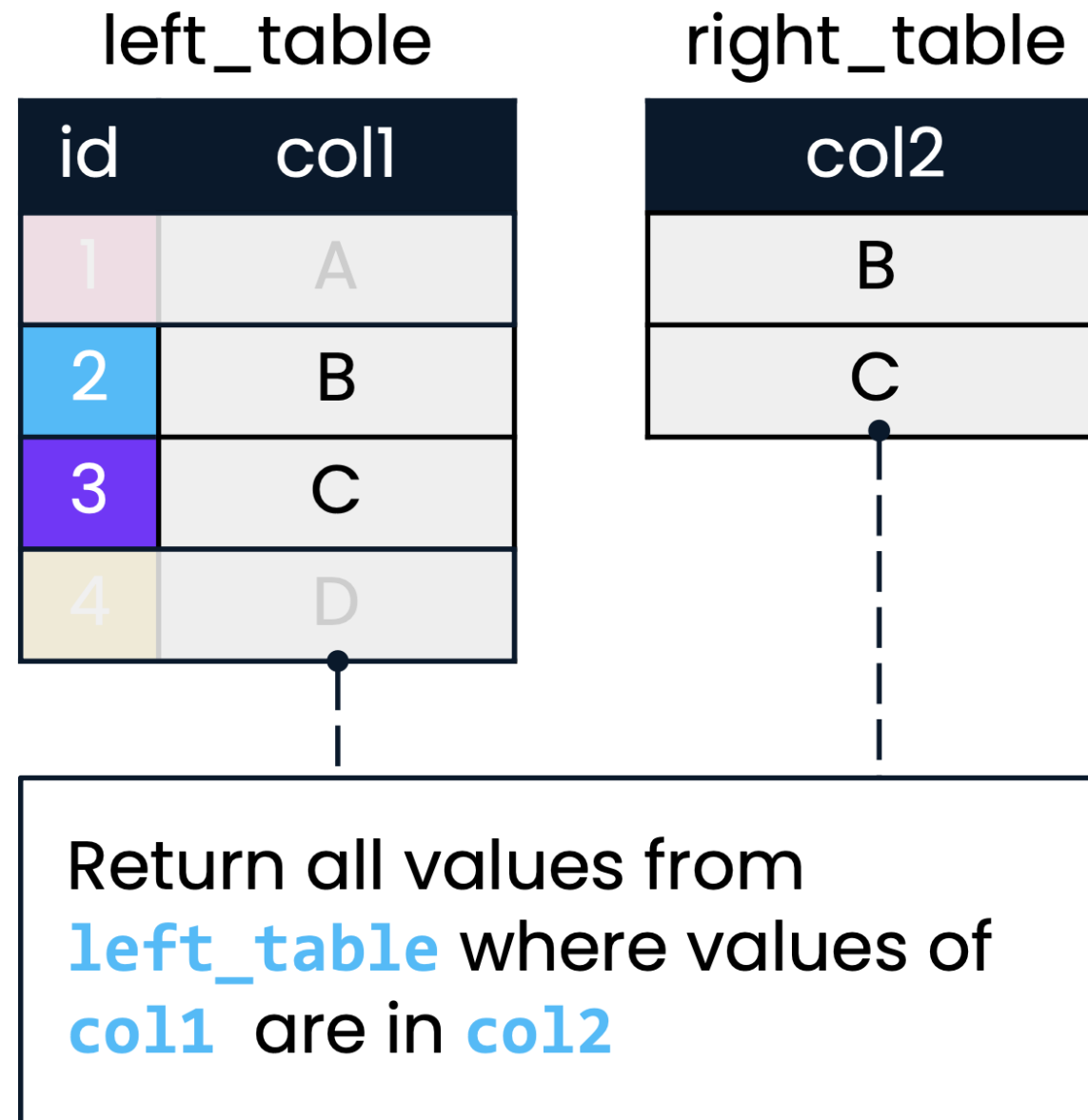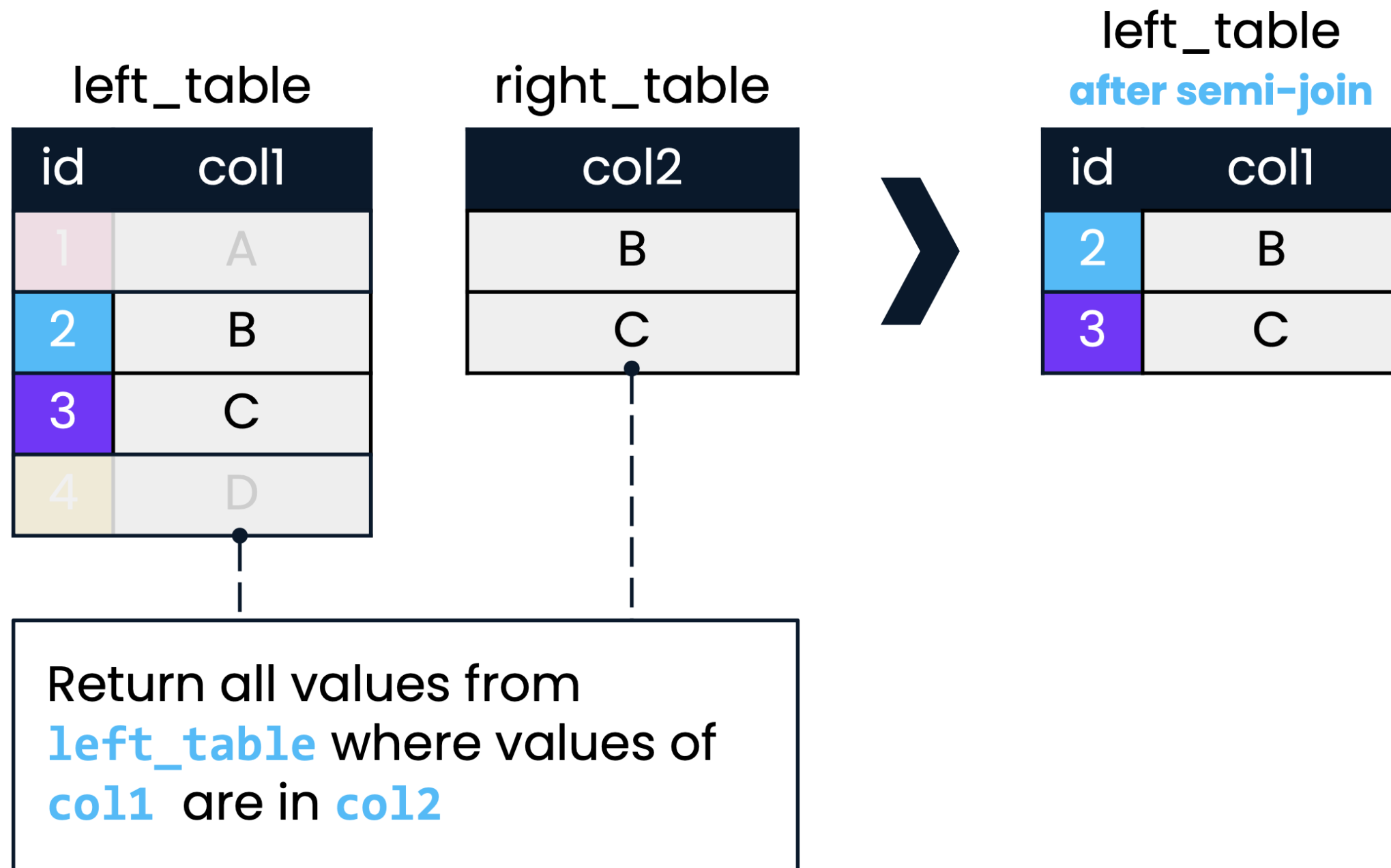
Kicking off our semi join

```sql
SELECT country, continent, president
FROM presidents;
```

country	continent	president
Egypt	Africa	Abdel Fattah el-Sisi
Portugal	Europe	Marcelo Rebelo de Sousa
USA	North America	Joe Biden
Uruguay	South America	Luis Lacalle Pou
Pakistan	Asia	Arif Alvi
Chile	South America	Gabriel Boric
India	Asia	Ram Nath Kovind

Building on our semi join

```sql
SELECT country
FROM states
WHERE indep_year < 1800;
```
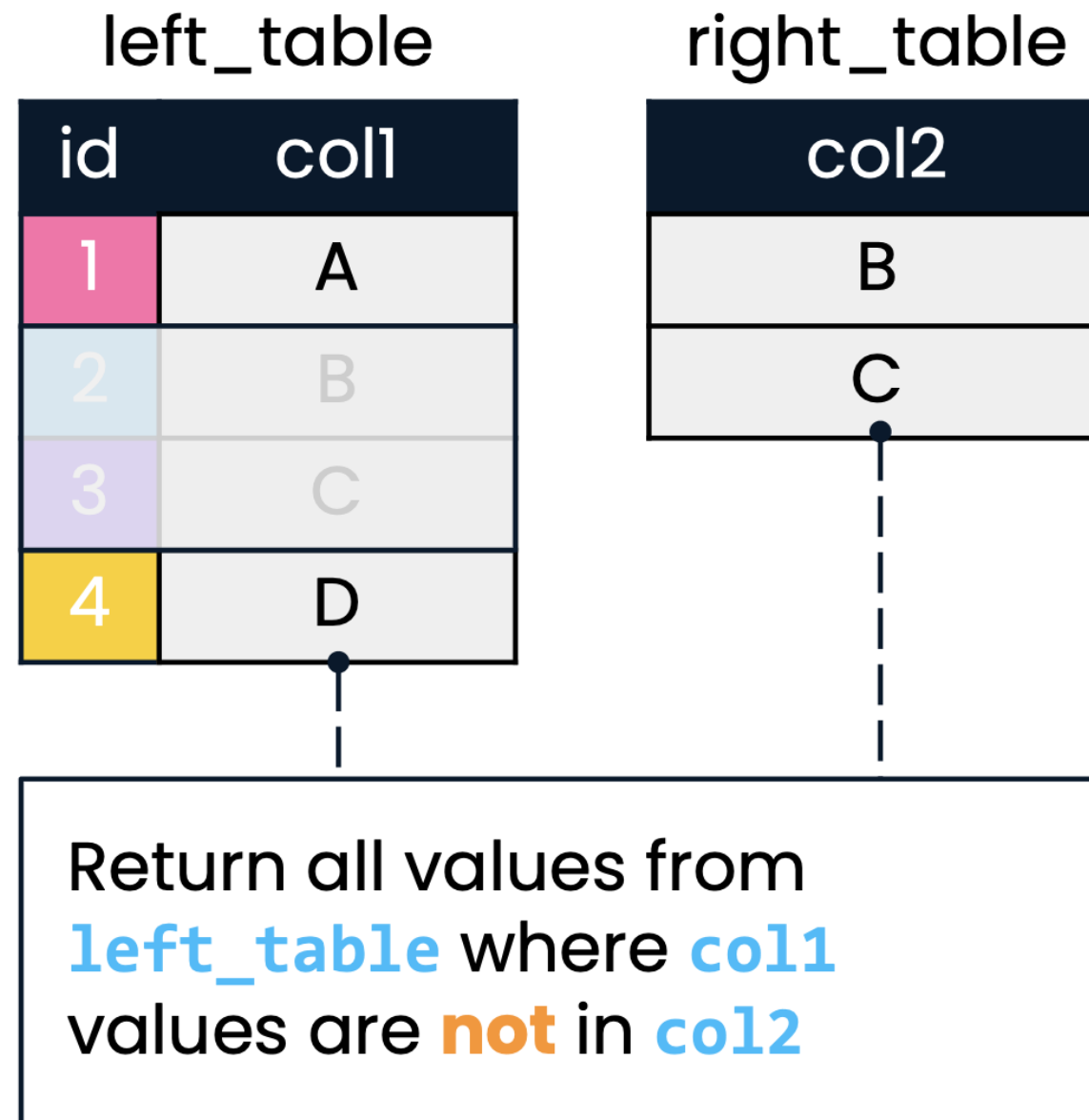
```
|----------|
| country  |
|----------|
| Portugal |
| Spain    |
|----------|
```

Finish the semi join (an intro to subqueries)
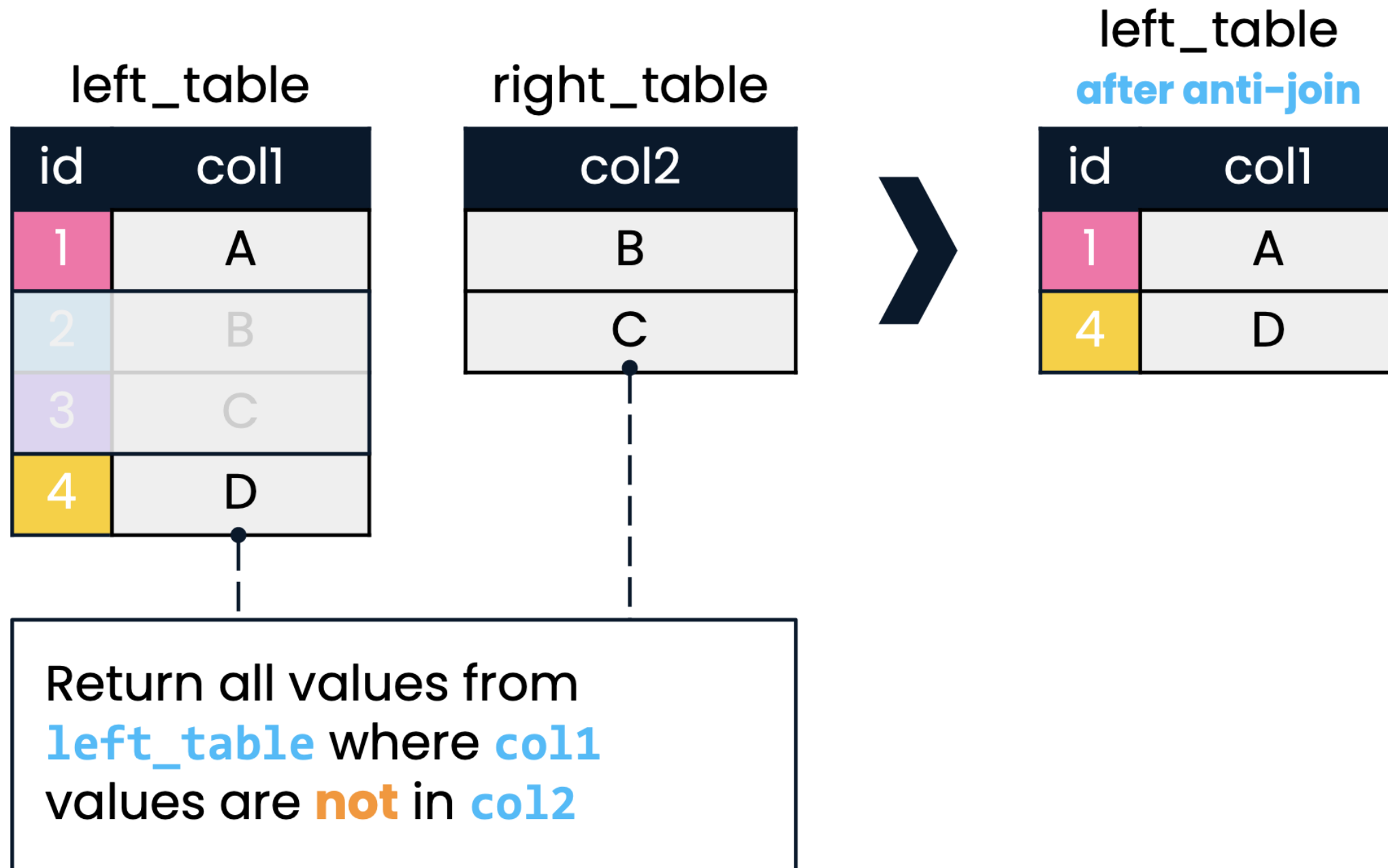
```sql
SELECT president, country, continent
FROM presidents
WHERE country IN
    (SELECT country
     FROM states
     WHERE indep_year < 1800);
```

president	country	continent
Marcelo Rebelo de Sousa	Portugal	Europe

Anti join

left_table

id	col1
1	A
2	B
3	C
4	D

right_table

col2
B
C

Return all values from left_table where col1 values are **not** in col2

Anti join

left_table

id	col1
1	A
2	B
3	C
4	D

right_table

col2
B
C

left_table
after anti-join

id	col1
1	A
4	D

Return all values from
`left_table` where `col1`
values are **not** in `col2`

An anti join with the presidents

```sql
SELECT country, president
FROM presidents
WHERE continent LIKE '%America'
    AND country NOT IN
        (SELECT country
         FROM states
         WHERE indep_year < 1800);
```

```
| country  |  president     |
| -------- | -------------- |
| Uruguay  | Luis Lacalle Pou |
| Chile    | Gabriel Boric    |
```

Let's practice!

JOINING DATA IN SQL

Syntax for subqueries inside WHERE

- All semi joins and anti joins we have seen included a subequery in `WHERE`

- `WHERE` is the most common place for subqueries

Syntax for query using `WHERE IN` statement

```sql
SELECT *
FROM some_table
WHERE some_numeric_field IN (4, 8, 12);
```

Syntax for subqueries inside WHERE

```sql
SELECT *
FROM some_table
WHERE some_field IN
        (include subquery here);
```

Syntax for subqueries inside WHERE

```sql
SELECT *
FROM some_table
WHERE some_field IN
    (SELECT some_numeric_field
     FROM another_table
     WHERE field2 = some_condition);
```

Subqueries inside SELECT

```sql
SELECT DISTINCT continent
FROM states;
```

```
|----------------|
| continent      |
|----------------|
| Africa         |
| Asia           |
| Europe         |
| North America  |
| Oceania        |
|----------------|
```

Subqueries inside SELECT

```sql
SELECT DISTINCT continent,
    (SELECT COUNT(*)
     FROM monarchs
     WHERE states.continent = monarch.continent) AS monarch_count
FROM states;
```

```
| continent      |   monarch_count |
|----------------|-----------------|
| Africa         |               0 |
| Asia           |               2 |
| Europe         |               2 |
| North America  |               0 |
| Oceania        |               0 |
```

Let's practice!

JOINING DATA IN SQL

datacamp

Subqueries inside FROM

JOINING DATA IN SQL

SQL

Maham Faisal Khan
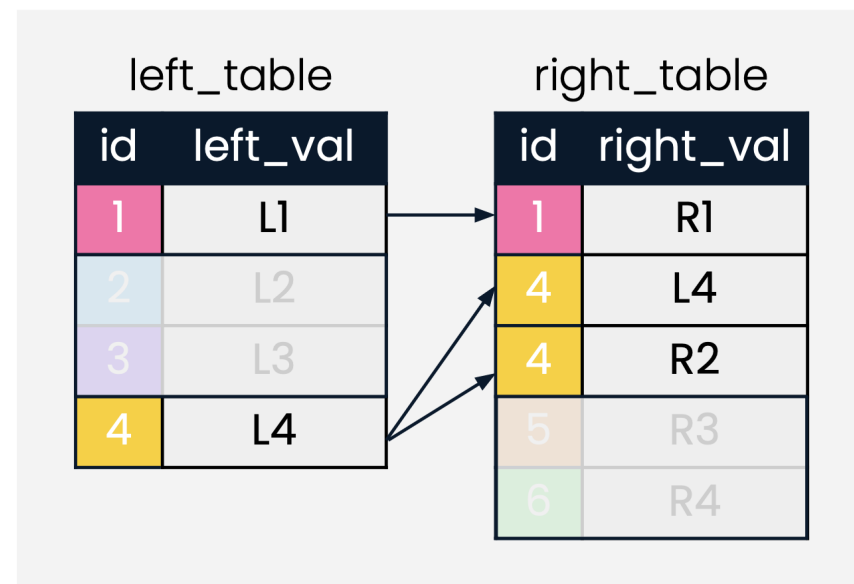Senior Content Developer, DataCamp

datacamp

Subqueries inside FROM

```sql
SELECT continent, MAX(indep_year) AS most_recent
FROM states
GROUP BY continent;
```
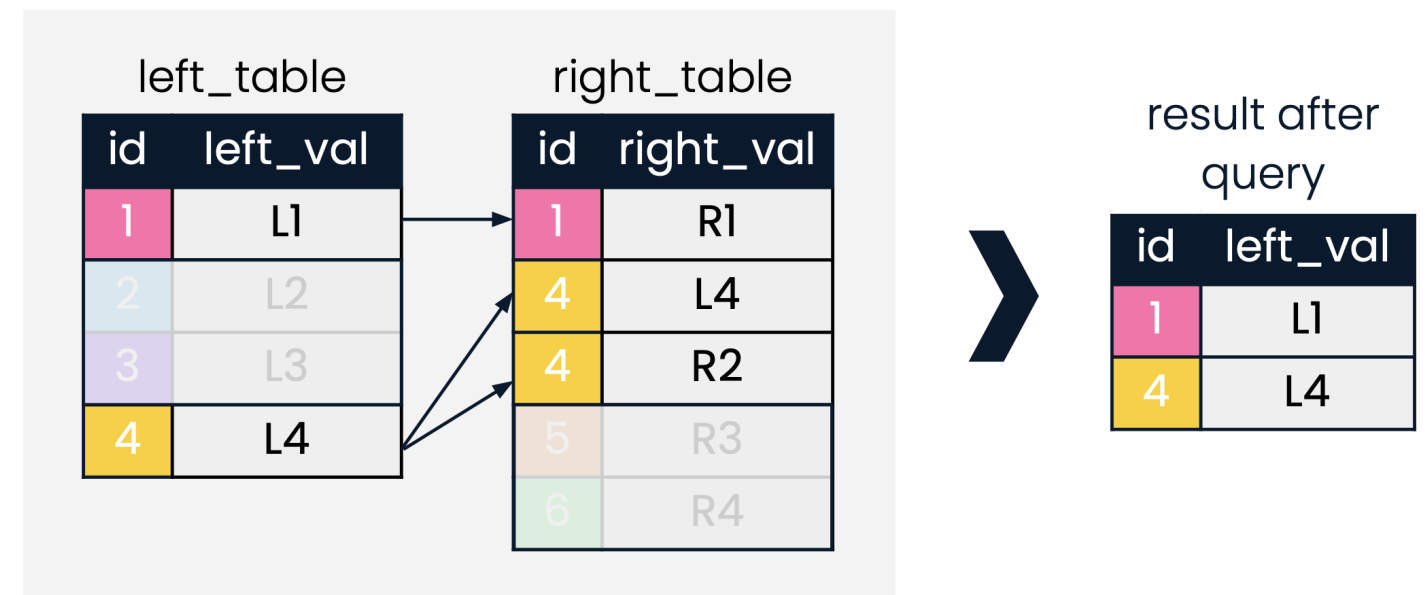
```
| continent     | most_recent  |
| --------------|------------- |
| Asia          |         1984 |
| Europe        |         1814 |
| Oceania       |         1901 |
| North America |         1776 |
| South America |         1818 |
```

Focusing on records inside monarchs

```sql
SELECT left_table.id, left_val
FROM left_table, right_table
WHERE left_table.id = right_table.id
```

```sql
SELECT DISTINCT left_table.id, left_val
FROM left_table, right_table
WHERE left_table.id = right_table.id
```


left_table

id	left_val
1	L1
2	L2
3	L3
4	L4

right_table

id	right_val
1	R1
4	R2
5	R3
6	R4

result after query

id	left_val
1	L1
4	L4
4	L4

left_table

id	left_val
1	L1
2	L2
3	L3
4	L4

right_table

id	right_val
1	R1
4	R2
5	R3
6	R4

result after query

id	left_val
1	L1
4	L4

Finishing off the subquery

```sql
-- Query to return continents with monarchs and the year the most recent country gained independence
SELECT DISTINCT monarchs.continent, sub.most_recent
FROM monarchs,
    (SELECT
        continent,
        MAX(indep_year) AS most_recent
     FROM states
     GROUP BY continent) AS sub
WHERE monarchs.continent = sub.continent
ORDER BY continent;
```

```
| continent   | most_recent   |
|-------------|---------------|
| Asia        |          1984  |
| Europe      |          1814  |
```

Let's practice!

JOINING DATA IN SQL

The finish line

JOINING DATA IN SQL

SQL

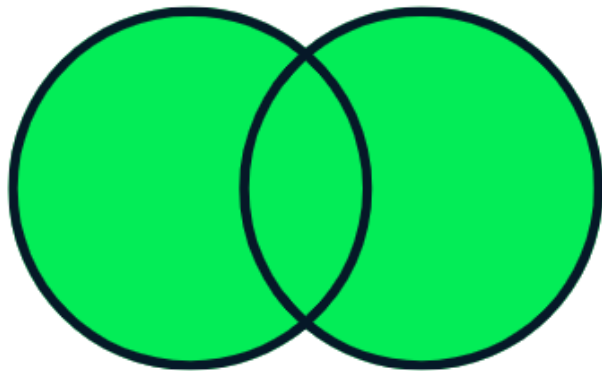**Maham Faisal Khan**
Senior Content Developer, DataCamp

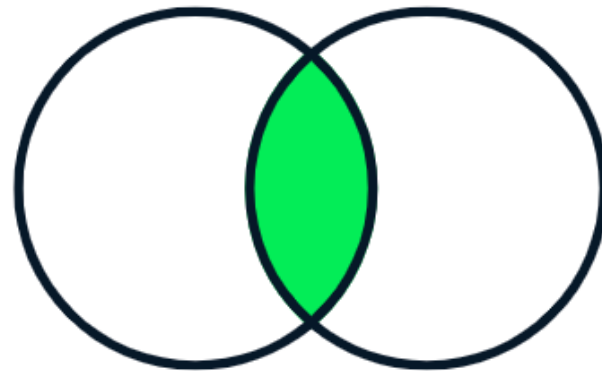datacamp

Types of joins

1. `INNER JOIN` , or just `JOIN`

2. Outer join
   - `LEFT JOIN`
   - `RIGHT JOIN`
   - `FULL JOIN`

3. `CROSS JOIN`

4. Semi join / anti join

5. Self join
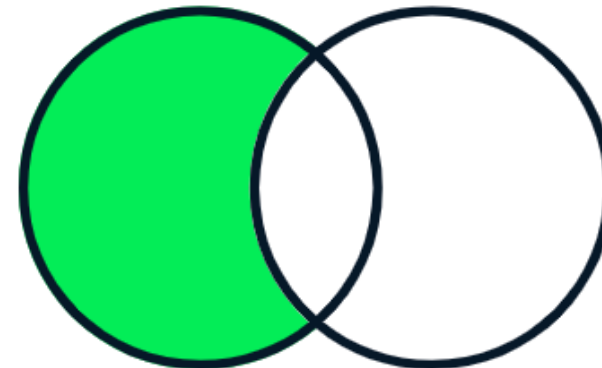
Set operations

UNION
UNION ALL

INTERSECT

EXCEPT

Types of basic subqueries

- Subqueries inside `SELECT` clauses

- Subqueries inside `WHERE` clauses

- Subqueries inside `FROM` clauses

WHERE to from here?

- DataCamp's **Intermediate SQL** course

- Projects, practice exercises and competitions

- Workspace

The finish line!

JOINING DATA IN SQL