



## JOINING DATA IN POSTGRESQL

**Welcome to the  
course!**

Chester Ismay

Curriculum Lead, DataCamp



left\_table

id	val
1	L1
2	L2
3	L3
4	L4

right\_table

id	val
1	R1
4	R2
5	R3
6	R4



left\_table

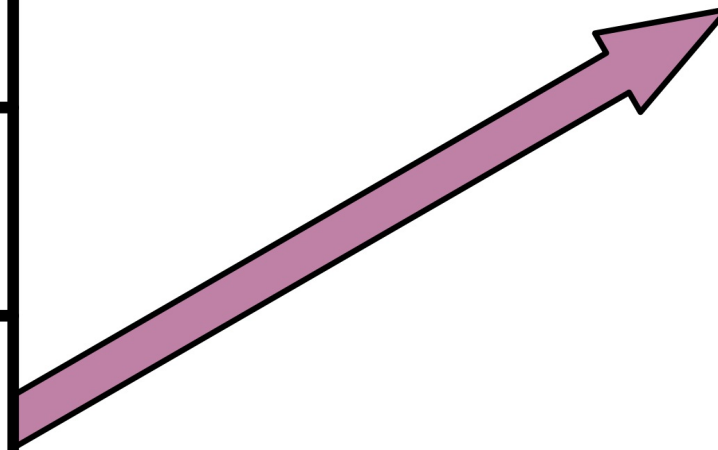
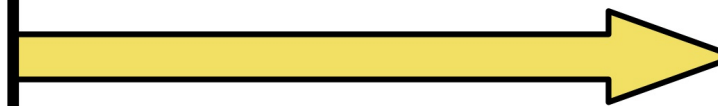
id val

1	L1
2	L2
3	L3
4	L4

right\_table

id val

1	R1
4	R2
5	R3
6	R4





left\_table

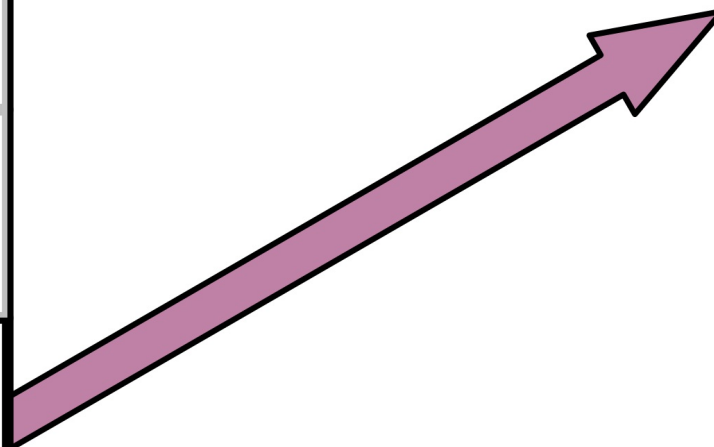
id val

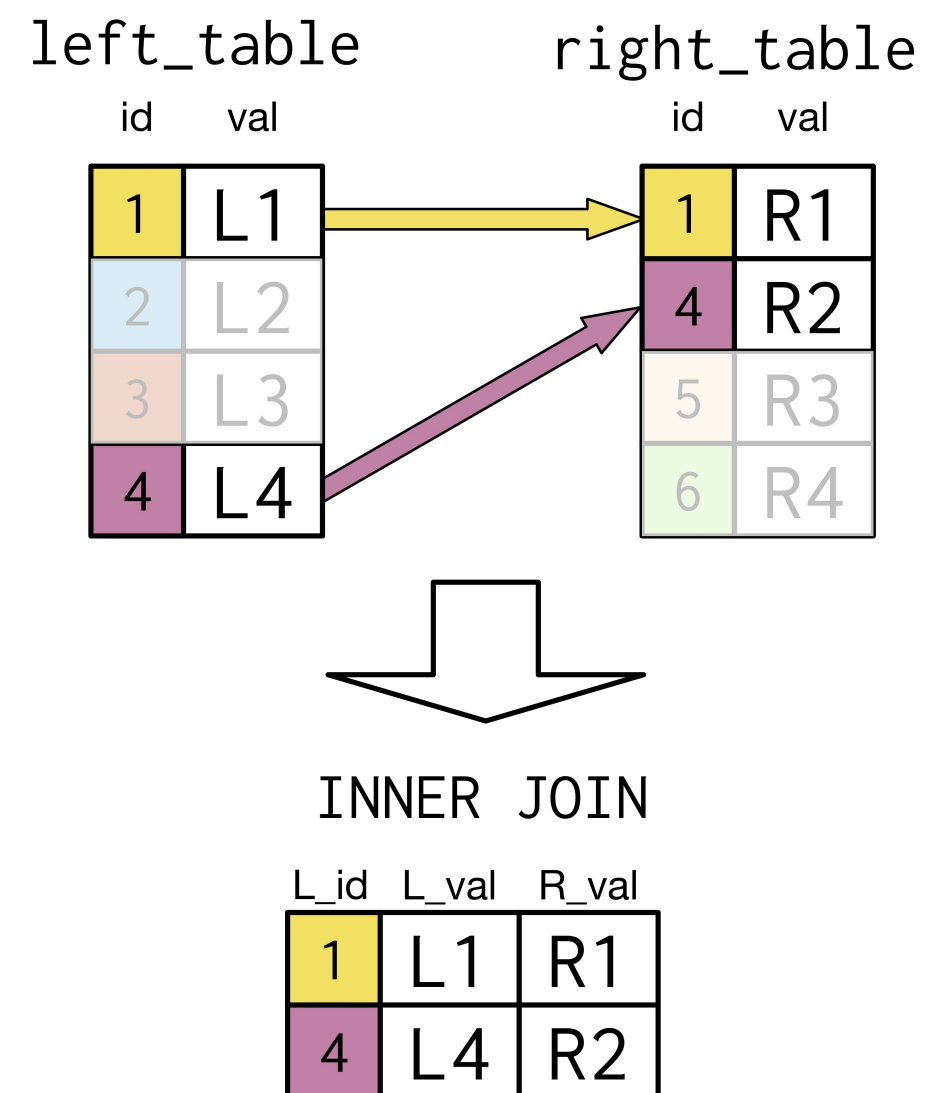
1	L1
2	L2
3	L3
4	L4

right\_table

id val

1	R1
4	R2
5	R3
6	R4







# prime\_ministers table

country	continent	prime_minister
Egypt	Africa	Sherif Ismail
Portugal	Europe	Antonio Costa
Vietnam	Asia	Nguyen Xuan Phuc
Haiti	North America	Jack Guy Lafontant
India	Asia	Narendra Modi
Australia	Oceania	Malcolm Turnbull
Norway	Europe	Erna Solberg
Brunei	Asia	Hassanal Bolkiah
Oman	Asia	Qaboos bin Said al Said
Spain	Europe	Mariano Rajoy

# presidents table

```
SELECT *  
FROM presidents;
```

country	continent	president
Egypt	Africa	Abdel Fattah el-Sisi
Portugal	Europe	Marcelo Rebelo de Sousa
Haiti	North America	Jovenel Moise
Uruguay	South America	Jose Mujica
Liberia	Africa	Ellen Johnson Sirleaf
Chile	South America	Michelle Bachelet
Vietnam	Asia	Tran Dai Quang

# INNER JOIN in SQL

```
SELECT p1.country, p1.continent,  
       prime_minister, president  
FROM prime_ministers AS p1  
INNER JOIN presidents AS p2  
ON p1.country = p2.country;
```

country	continent	prime_minister	president
Egypt	Africa	Sherif Ismail	Abdel Fattah el-Sisi
Portugal	Europe	Antonio Costa	Marcelo Rebelo de Sousa
Vietnam	Asia	Nguyen Xuan Phuc	Tran Dai Quang
Haiti	North America	Jack Guy Lafontant	Jovenel Moise





## JOINING DATA IN POSTGRESQL

**Let's practice!**



## JOINING DATA IN POSTGRESQL

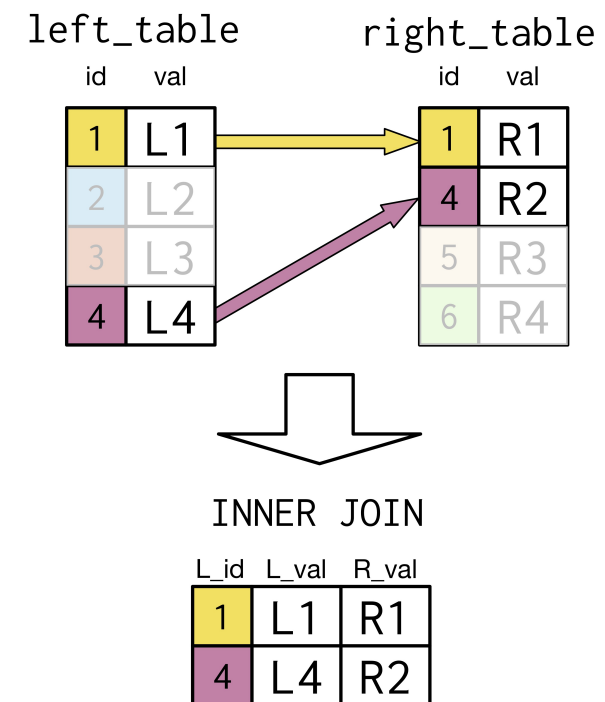
# INNER JOIN via USING

Chester Ismay

Curriculum Lead, DataCamp



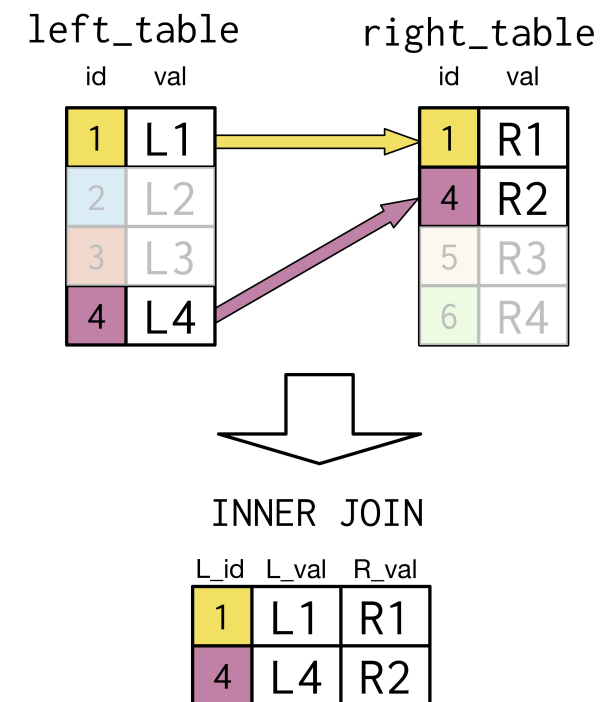
# The INNER JOIN diagram again



```
SELECT left_table.id AS L_id,  
       left_table.val AS L_val,  
       right_table.val AS R_val  
FROM left_table  
INNER JOIN right_table  
ON left_table.id = right_table.id;
```



# The INNER JOIN diagram with USING



```
SELECT left_table.id AS L_id,  
       left_table.val AS L_val,  
       right_table.val AS R_val  
FROM left_table  
INNER JOIN right_table  
USING (id);
```

# Countries with prime ministers and presidents - again

```
SELECT p1.country, p1.continent, prime_minister, president
FROM _____ AS p1
INNER JOIN _____ AS p2
  ON _____;
```

One answer:

```
SELECT p1.country, p1.continent, prime_minister, president
FROM presidents AS p1
INNER JOIN prime_ministers AS p2
  USING (country);
```

country	continent	prime_minister	president
Egypt	Africa	Sherif Ismail	Abdel Fattah el-Sisi
Portugal	Europe	Antonio Costa	Marcelo Rebelo de Sousa
Vietnam	Asia	Nguyen Xuan Phuc	Tran Dai Quang
Haiti	North America	Jack Guy Lafontant	Jovenel Moise



## JOINING DATA IN POSTGRES SQL

**Let's practice!**



## JOINING DATA IN POSTGRESQL

# Self-ish joins, just in CASE

Chester Ismay

Curriculum Lead, DataCamp



# Join a table to itself?

country	continent	prime_minister
Egypt	Africa	Sherif Ismail
Portugal	Europe	Antonio Costa
Vietnam	Asia	Nguyen Xuan Phuc
Haiti	North America	Jack Guy Lafontant
India	Asia	Narendra Modi
Australia	Oceania	Malcolm Turnbull
Norway	Europe	Erna Solberg
Brunei	Asia	Hassanal Bolkiah
Oman	Asia	Qaboos bin Said al Said
Spain	Europe	Mariano Rajoy



# Join prime\_ministers to itself?

```
SELECT p1.country AS country1, p2.country AS country2, p1.continent
FROM prime_ministers AS p1
INNER JOIN prime_ministers AS p2
ON p1.continent = p2.continent
LIMIT 14;
```

country1	country2	continent
Egypt	Egypt	Africa
Portugal	Spain	Europe
Portugal	Norway	Europe
Portugal	Portugal	Europe
Vietnam	Oman	Asia
Vietnam	Brunei	Asia
Vietnam	India	Asia
Vietnam	Vietnam	Asia
Haiti	Haiti	North America
India	Oman	Asia
India	Brunei	Asia
India	India	Asia
India	Vietnam	Asia
Australia	Australia	Oceania

# Finishing off the self-join on prime\_ministers

```
SELECT p1.country AS country1, p2.country AS country2, p1.continent
FROM prime_ministers AS p1
INNER JOIN prime_ministers AS p2
ON p1.continent = p2.continent AND p1.country <> p2.country
LIMIT 13;
```

country1	country2	continent
Portugal	Spain	Europe
Portugal	Norway	Europe
Vietnam	Oman	Asia
Vietnam	Brunei	Asia
Vietnam	India	Asia
India	Oman	Asia
India	Brunei	Asia
India	Vietnam	Asia
Norway	Spain	Europe
Norway	Portugal	Europe
Brunei	Oman	Asia
Brunei	India	Asia
Brunei	Vietnam	Asia



# CASE WHEN and THEN

name	continent	indep_year
Australia	Oceania	1901
Brunei	Asia	1984
Chile	South America	1810
Egypt	Africa	1922
Haiti	North America	1804
India	Asia	1947
Liberia	Africa	1847
Norway	Europe	1905
Oman	Asia	1951
Portugal	Europe	1143
Spain	Europe	1492
Uruguay	South America	1828
Vietnam	Asia	1945



# Preparing indep\_year\_group in states

```
SELECT name, continent, indep_year,  
       CASE WHEN ____ < ____ THEN 'before 1900'  
            WHEN indep_year <= 1930 THEN '____'  
            ELSE '____' END  
       AS indep_year_group  
FROM states  
ORDER BY indep_year_group;
```

# Creating indep\_year\_group in states

```
SELECT name, continent, indep_year,  
       CASE WHEN indep_year < 1900 THEN 'before 1900'  
            WHEN indep_year <= 1930 THEN 'between 1900 and 1930'  
            ELSE 'after 1930' END  
       AS indep_year_group  
FROM states  
ORDER BY indep_year_group;
```

name	continent	indep_year	indep_year_group
Brunei	Asia	1984	after 1930
India	Asia	1947	after 1930
Oman	Asia	1951	after 1930
Vietnam	Asia	1945	after 1930
Liberia	Africa	1847	before 1900
Chile	South America	1810	before 1900
Haiti	North America	1804	before 1900
Portugal	Europe	1143	before 1900
Spain	Europe	1492	before 1900
Uruguay	South America	1828	before 1900
Norway	Europe	1905	between 1900 and 1930
Australia	Oceania	1901	between 1900 and 1930
Egypt	Africa	1922	between 1900 and 1930



## JOINING DATA IN POSTGRESQL

**Let's practice!**