

# Welcome to the course!

JOINING DATA IN SQL



**Chester Ismay**

Data Science Evangelist, DataRobot

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

`left_table`

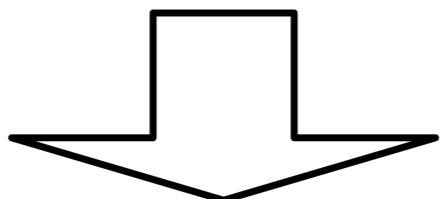
id val

1	L1
2	L2
3	L3
4	L4

`right_table`

id val

1	R1
4	R2
5	R3
6	R4



INNER JOIN

L_id	L_val	R_val
1	L1	R1
4	L4	R2

# 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

# **Let's practice!**

**JOINING DATA IN SQL**

# INNER JOIN via USING

JOINING DATA IN SQL

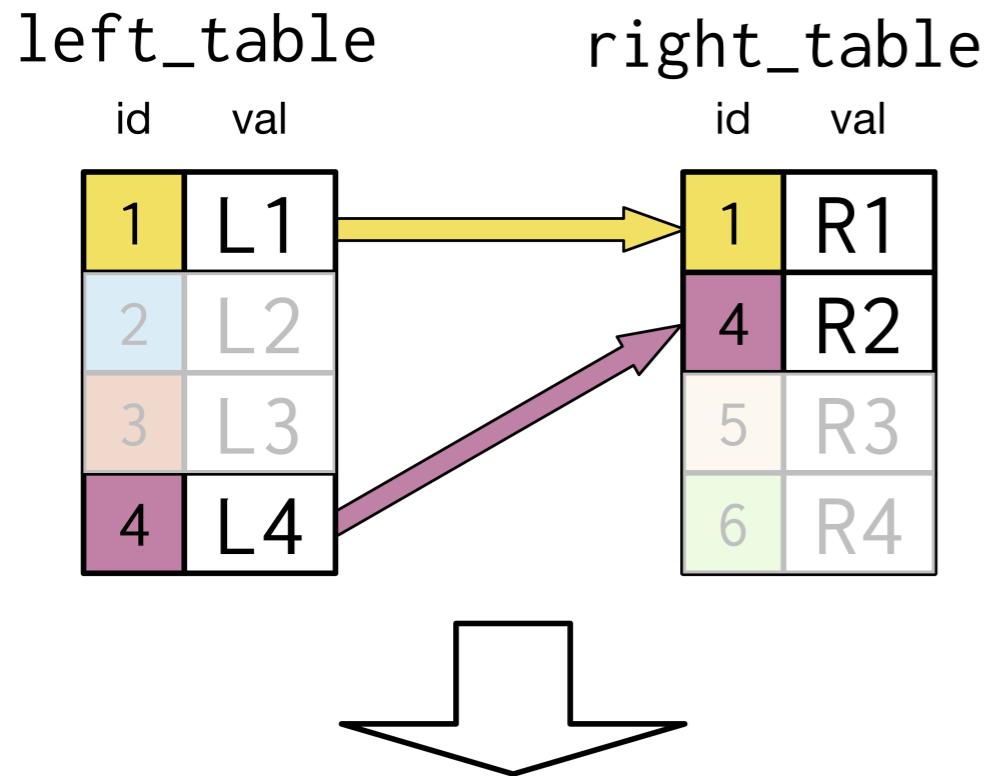


SQL

**Chester Ismay**

Data Science Evangelist, DataRobot

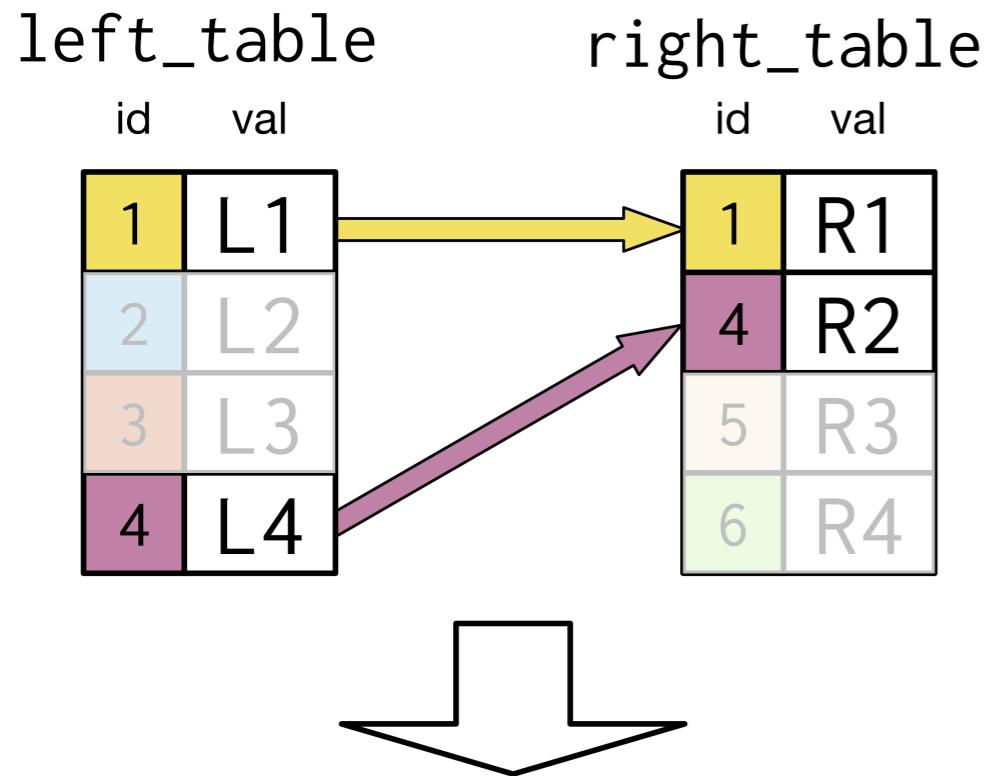
# The INNER JOIN diagram again



L_id	L_val	R_val
1	L1	R1
4	L4	R2

```
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



L_id	L_val	R_val
1	L1	R1
4	L4	R2

```
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

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

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

# **Let's practice!**

**JOINING DATA IN SQL**

# **Self-ish joins, just in CASE**

**JOINING DATA IN SQL**

A dark blue circular icon containing the white text "SQL".

**Chester Ismay**

Data Science Evangelist, DataRobot

# 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

# **Let's practice!**

**JOINING DATA IN SQL**

# LEFT and RIGHT JOINS

JOINING DATA IN SQL

SQL

**Chester Ismay**

Data Science Evangelist, DataRobot

`left_table`

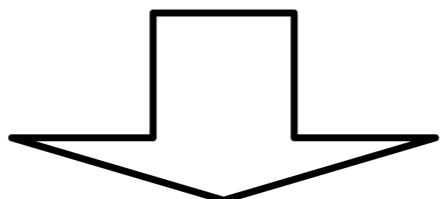
id val

1	L1
2	L2
3	L3
4	L4

`right_table`

id val

1	R1
4	R2
5	R3
6	R4



INNER JOIN

L_id	L_val	R_val
1	L1	R1
4	L4	R2

`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`

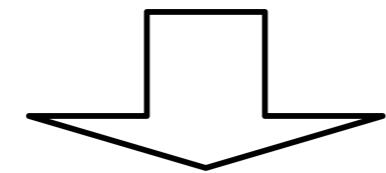
`id`    `val`

1	L1
2	L2
3	L3
4	L4

`right_table`

`id`    `val`

1	R1
4	R2
5	R3
6	R4



LEFT JOIN

<code>L_id</code>	<code>L_val</code>	<code>R_val</code>
1	L1	R1
2	L2	
3	L3	
4	L4	R2

`left_table`

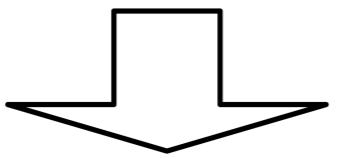
`id`   `val`

1	L1
2	L2
3	L3
4	L4

`right2`

`id`   `val`

1	R1
1	R2
4	R3
5	R4
6	R5



LEFT JOIN

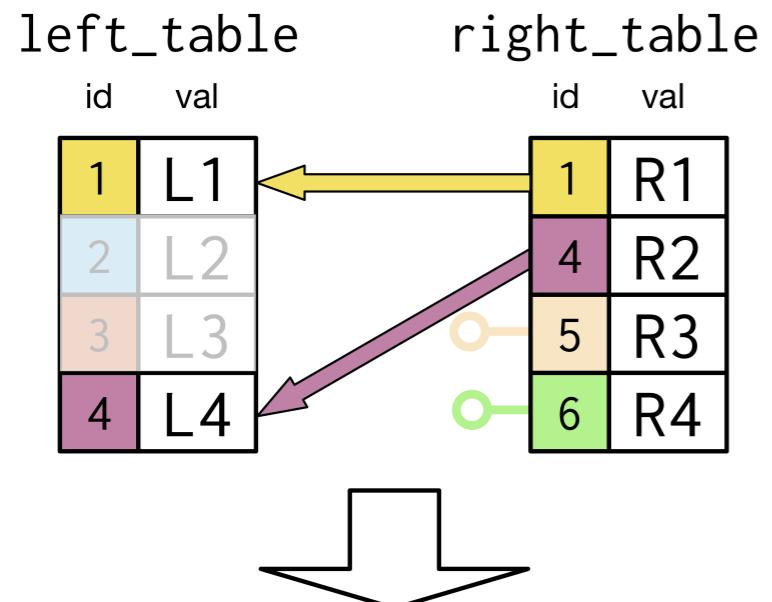
<code>L_id</code>	<code>L_val</code>	<code>R_val</code>
1	L1	R1
1	L1	R2
2	L2	
3	L3	
4	L4	R3

# The syntax of a LEFT JOIN

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

country	prime_minister	president
Egypt	Sherif Ismail	Abdel Fattah el-Sisi
Portugal	Antonio Costa	Marcelo Rebelo de Sousa
Vietnam	Nguyen Xuan Phuc	Tran Dai Quang
Haiti	Jack Guy Lafontant	Jovenel Moise
India	Narendra Modi	
Australia	Malcolm Turnbull	
Norway	Erna Solberg	
Brunei	Hassanal Bolkiah	
Oman	Qaboos bin Said al Said	
Spain	Mariano Rajoy	

# RIGHT JOIN



RIGHT JOIN

	R_id	L_val	R_val
1	1	L1	R1
4	4	L4	R2
5			R3
6	6		R4

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

# **Let's practice!**

**JOINING DATA IN SQL**

# FULL JOINS

JOINING DATA IN SQL

SQL

**Chester Ismay**

Data Science Evangelist, DataRobot

# INNER JOIN vs LEFT JOIN

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

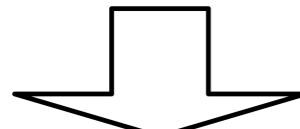
id val

1	L1
2	L2
3	L3
4	L4

right\_table

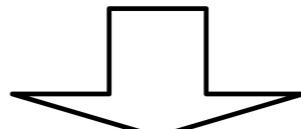
id val

1	R1
4	R2
5	R3
6	R4



INNER JOIN

L_id	L_val	R_val
1	L1	R1
4	L4	R2



LEFT JOIN

L_id	L_val	R_val
1	L1	R1
2	L2	
3	L3	
4	L4	R2

# LEFT JOIN vs RIGHT JOIN

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 JOIN

L\_id L\_val R\_val

1	L1	R1
2	L2	
3	L3	
4	L4	R2

RIGHT JOIN

R\_id L\_val R\_val

1	L1	R1
4	L4	R2
5		R3
6		R4

# left\_table

id val

1	L1
2	L2
3	L3
4	L4

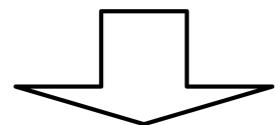
# right\_table

id val

1	R1
4	R2
5	R3
6	R4

# FULL JOIN diagram

left_table		right_table	
id	val	id	val
1	L1	1	R1
2	L2	4	R2
3	L3	5	R3
4	L4	6	R4



FULL JOIN

L_id	R_id	L_val	R_val
1	1	L1	R1
2		L2	
3		L3	
4	4	L4	R2
	5		R3
	6		R4

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

# FULL JOIN example using leaders database

```
SELECT p1.country AS pm_co, p2.country AS pres_co,  
prime_minister, president
```

# FULL JOIN example using leaders database

```
SELECT p1.country AS pm_co, p2.country AS pres_co,  
       prime_minister, president  
FROM prime_ministers AS p1
```

# FULL JOIN example using leaders database

```
SELECT p1.country AS pm_co, p2.country AS pres_co,  
       prime_minister, president  
  FROM prime_ministers AS p1  
FULL JOIN presidents AS p2
```

# FULL JOIN example using leaders database

```
SELECT p1.country AS pm_co, p2.country AS pres_co,  
       prime_minister, president  
  FROM prime_ministers AS p1  
 FULL JOIN presidents AS p2  
    ON p1.country = p2.country;
```

# FULL JOIN example results using leaders

pm_co	pres_co	prime_minister	president
Egypt	Egypt	Sherif Ismail	Abdel Fattah el-Sisi
Portugal	Portugal	Antonio Costa	Marcelo Rebelo de Sousa
Vietnam	Vietnam	Nguyen Xuan Phuc	Tran Dai Quang
Haiti	Haiti	Jack Guy Lafontant	Jovenel Moise
India		Narendra Modi	
Australia		Malcolm Turnbull	
Norway		Erna Solberg	
Brunei		Hassanal Bolkiah	
Oman		Qaboos bin Said al Said	
Spain		Mariano Rajoy	
	Uruguay		Jose Mujica
	Chile		Michelle Bachelet
	Liberia		Ellen Johnson Sirleaf

# **Let's practice!**

**JOINING DATA IN SQL**

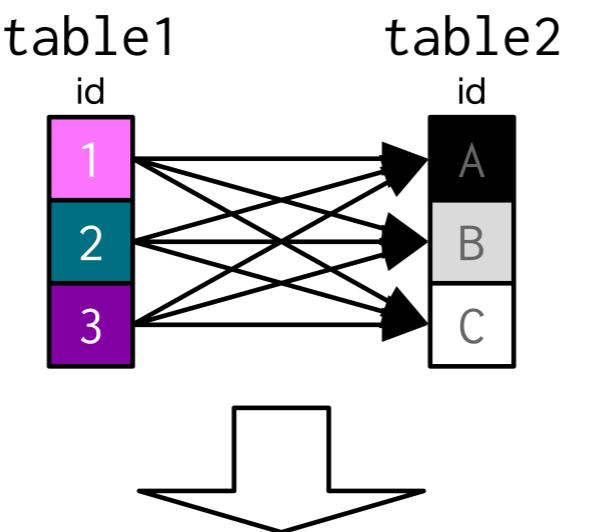
# CROSSing the Rubicon

JOINING DATA IN SQL

SQL

**Chester Ismay**

Data Science Evangelist, DataRobot



CROSS JOIN

id1	id2
1	A
1	B
1	C
2	A
2	B
2	C
3	A
3	B
3	C

# Pairing prime ministers with presidents

```
SELECT prime_minister, president
FROM prime_ministers AS p1
CROSS JOIN presidents AS p2
WHERE p1.continent IN ('North America', 'Oceania');
```

prime_minister	president
Jack Guy Lafontant	Abdel Fattah el-Sisi
Malcolm Turnbull	Abdel Fattah el-Sisi
Jack Guy Lafontant	Marcelo Rebelo de Sousa
Malcolm Turnbull	Marcelo Rebelo de Sousa
Jack Guy Lafontant	Jovenel Moise
Malcolm Turnbull	Jovenel Moise
Jack Guy Lafontant	Jose Mujica
Malcolm Turnbull	Jose Mujica
Jack Guy Lafontant	Ellen Johnson Sirleaf
Malcolm Turnbull	Ellen Johnson Sirleaf
Jack Guy Lafontant	Michelle Bachelet
Malcolm Turnbull	Michelle Bachelet
Jack Guy Lafontant	Tran Dai Quang
Malcolm Turnbull	Tran Dai Quang

# **Let's practice!**

**JOINING DATA IN SQL**

# State of the UNION

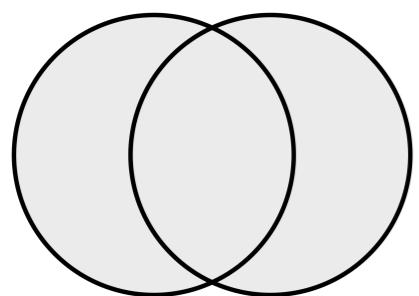
JOINING DATA IN SQL

A dark blue circular icon containing the letters "SQL" in white.

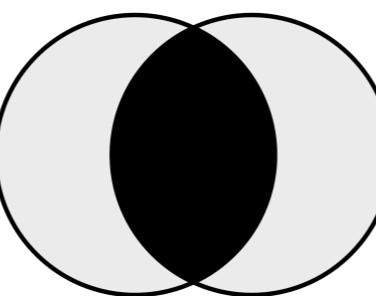
**Chester Ismay**

Data Science Evangelist, DataRobot

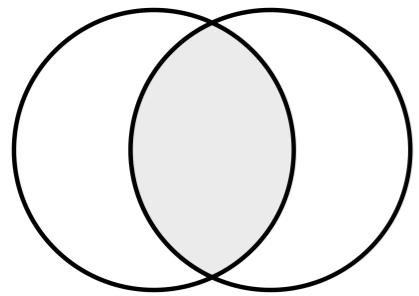
# Set Theory Venn Diagrams



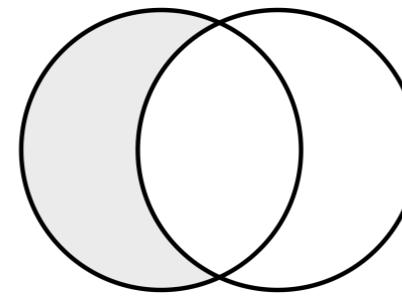
UNION



UNION ALL



INTERSECT

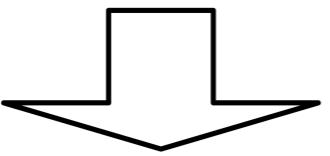


EXCEPT

`left_one`      `right_one`

<code>id</code>
1
2
3
4

<code>id</code>
1
4
5
6



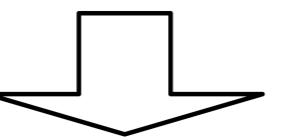
`UNION`

<code>id</code>
1
2
3
4
5
6

`left_one`      `right_one`

<code>id</code>
1
2
3
4

<code>id</code>
1
4
5
6



`UNION ALL`

<code>id</code>
1
1
2
3
4
4
5
6

# monarchs table

```
SELECT *  
FROM monarchs;
```

country	continent	monarch
Brunei	Asia	Hassanal Bolkiah
Oman	Asia	Qaboos bin Said al Said
Norway	Europe	Harald V
Spain	Europe	Felipe VI

# All prime ministers and monarchs

```
SELECT prime_minister AS Leader, country  
FROM prime_ministers  
UNION  
SELECT monarch, country  
FROM monarchs  
ORDER BY country;
```

# Resulting table from UNION

leader	country
Malcolm Turnbull	Australia
Hassanal Bolkiah	Brunei
Sherif Ismail	Egypt
Jack Guy Lafontant	Haiti
Narendra Modi	India
Harald V	Norway
Erna Solberg	Norway
Qaboos bin Said al Said	Oman
Antonio Costa	Portugal
Mariano Rajoy	Spain
Felipe VI	Spain
Nguyen Xuan Phuc	Vietnam

# UNION ALL with leaders

```
SELECT prime_minister AS leader, country
FROM prime_ministers
UNION ALL
SELECT monarch, country
FROM monarchs
ORDER BY country
LIMIT 10;
```

leader	country
Malcolm Turnbull	Australia
Hassanal Bolkiah	Brunei
Hassanal Bolkiah	Brunei
Sherif Ismail	Egypt
Jack Guy Lafontant	Haiti
Narendra Modi	India
Erna Solberg	Norway
Harald V	Norway
Qaboos bin Said al Said	Oman
Qaboos bin Said al Said	Oman

# **Let's practice!**

**JOINING DATA IN SQL**

# INTERSECTional data science

JOINING DATA IN SQL

A dark blue circular icon containing the white text "SQL".

**Chester Ismay**

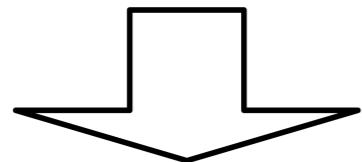
Data Science Evangelist, DataRobot

# INTERSECT diagram and SQL code

left\_one      right\_one

id
1
2
3
4

id
1
4
5
6



INTERSECT

id
1
4

```
SELECT id  
FROM left_one  
INTERSECT  
SELECT id  
FROM right_one;
```

# Prime minister and president countries

```
SELECT country  
FROM prime_ministers  
INTERSECT  
SELECT country  
FROM presidents;
```

country
Portugal
Vietnam
Haiti
Egypt

# INTERSECT on two fields

```
SELECT country, prime_minister AS Leader  
FROM prime_ministers  
INTERSECT  
SELECT country, president  
FROM presidents;
```

country	Leader

# **Let's practice!**

**JOINING DATA IN SQL**

# EXCEPTiOnal

## JOINING DATA IN SQL

SQL

**Chester Ismay**

Data Science Evangelist, DataRobot

# Monarchs that aren't prime ministers

```
SELECT monarch, country  
FROM monarchs  
EXCEPT  
SELECT prime_minister, country  
FROM prime_ministers;
```

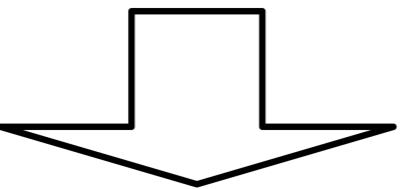
monarch	country
Harald V	Norway
Felipe VI	Spain

`left_one`

<code>id</code>
1
2
3
4

`right_one`

<code>id</code>
1
4
5
6



`EXCEPT`

<code>id</code>
2
3

# **Let's practice!**

**JOINING DATA IN SQL**

# Semi-joins and Anti-joins

JOINING DATA IN SQL

SQL

**Chester Ismay**

Data Science Evangelist, DataRobot

# Building up to a semi-join

```
SELECT name  
FROM states  
WHERE indep_year < 1800;
```

name
Portugal
Spain

# Another step towards the semi-join

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

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

# Finish the semi-join (an intro to subqueries)

```
SELECT president, country, continent  
FROM presidents  
WHERE country IN  
(SELECT name  
FROM states  
WHERE indep_year < 1800);
```

president	country	continent
Marcelo Rebelo de Sousa	Portugal	Europe

# An anti-join

```
SELECT president, country, continent
FROM presidents
WHERE ___ LIKE '___'
    AND country ___ IN
        (SELECT name
         FROM states
         WHERE indep_year < 1800);
```

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

# The result of the anti-join

president	country	continent
Jovenel Moise	Haiti	North America
Jose Mujica	Uruguay	South America
Michelle Bachelet	Chile	South America

# Semi-join and anti-join diagrams

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

Semi-join

L\_id L\_val

1	L1
4	L4

Anti-join

L\_id L\_val

2	L2
3	L3

# **Let's practice!**

**JOINING DATA IN SQL**

# Subqueries inside WHERE and SELECT clauses

JOINING DATA IN SQL

SQL

**Chester Ismay**

Data Science Evangelist, DataRobot

# Subquery inside WHERE clause set-up

name	indep_year	fert_rate	women_parli_perc
Australia	1901	1.88	32.74
Brunei	1984	1.96	6.06
Chile	1810	1.8	15.82
Egypt	1922	2.7	14.9
Haiti	1804	3.03	2.74
India	1947	2.43	11.58
Liberia	1847	4.64	11.65
Norway	1905	1.93	39.6
Oman	1951	2.75	8.82
Portugal	1143	1.31	34.8
Spain	1492	1.53	38.64
Uruguay	1828	2.03	22.31
Vietnam	1945	1.7	24

# Average fert\_rate

```
SELECT AVG(fert_rate)  
FROM states;
```

avg
2.28385

# Asian countries below average `fert\_rate`

```
SELECT name, fert_rate  
FROM states  
WHERE continent = 'Asia'
```

# Asian countries below average `fert\_rate`

```
SELECT name, fert_rate  
FROM states  
WHERE continent = 'Asia'  
AND fert_rate <
```

# Asian countries below average `fert\_rate`

```
SELECT name, fert_rate  
FROM states  
WHERE continent = 'Asia'  
    AND fert_rate <  
        (SELECT AVG(fert_rate)  
         FROM states);
```

# Asian countries below average `fert\_rate`

```
SELECT name, fert_rate  
FROM states  
WHERE continent = 'Asia'  
    AND fert_rate <  
        (SELECT AVG(fert_rate)  
         FROM states);
```

name	fert_rate
Brunei	1.96
Vietnam	1.7

# Subqueries inside SELECT clauses - setup

```
SELECT DISTINCT continent  
FROM prime_ministers;
```

continent
Africa
Asia
Europe
North America
Oceania

# Subquery inside SELECT clause - complete

```
SELECT DISTINCT continent,  
    (SELECT COUNT(*)  
     FROM states  
    WHERE prime_ministers.continent = states.continent) AS countries_num  
FROM prime_ministers;
```

continent	countries_num
Africa	2
Asia	4
Europe	3
North America	1
Oceania	1

# **Let's practice!**

**JOINING DATA IN SQL**

# Subquery inside the **FROM** clause

JOINING DATA IN SQL

A dark blue circular icon containing the white text "SQL".

**Chester Ismay**

Data Science Evangelist, DataRobot

# Build-up

```
SELECT continent, MAX(women_parli_perc) AS max_perc  
FROM states  
GROUP BY continent  
ORDER BY continent;
```

continent	max_perc
Africa	14.9
Asia	24
Europe	39.6
North America	2.74
Oceania	32.74
South America	22.31

# Focusing on records in monarchs

```
SELECT monarchs.continent  
FROM monarchs, states  
WHERE monarchs.continent = states.continent  
ORDER BY continent;
```

continent
Asia
Europe

# Finishing off the subquery

```
SELECT DISTINCT monarchs.continent, subquery.max_perc
FROM monarchs,
  (SELECT continent, MAX(women_parli_perc) AS max_perc
   FROM states
   GROUP BY continent) AS subquery
WHERE monarchs.continent = subquery.continent
ORDER BY continent;
```

continent	max_perc
Asia	24
Europe	39.6

# **Let's practice!**

**JOINING DATA IN SQL**

# Course Review

JOINING DATA IN SQL

SQL

**Chester Ismay**

Data Science Evangelist, DataRobot

# Types of joins

- INNER JOIN
  - Self-joins
- OUTER JOIN
  - LEFT JOIN
  - RIGHT JOIN
  - FULL JOIN
- CROSS JOIN
- Semi-join / Anti-join

# INNER JOIN vs LEFT JOIN

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

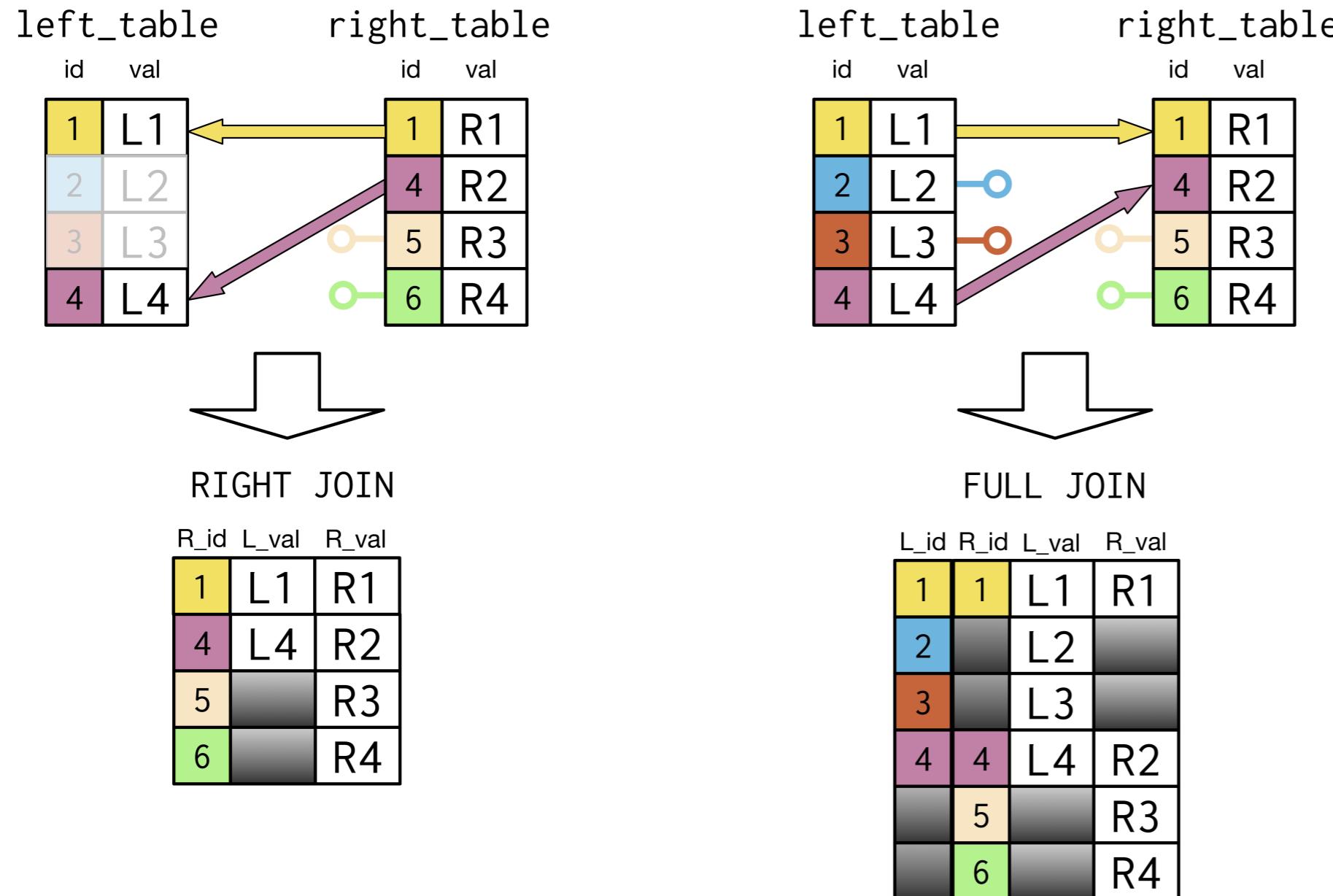
INNER JOIN

L_id	L_val	R_val
1	L1	R1
4	L4	R2

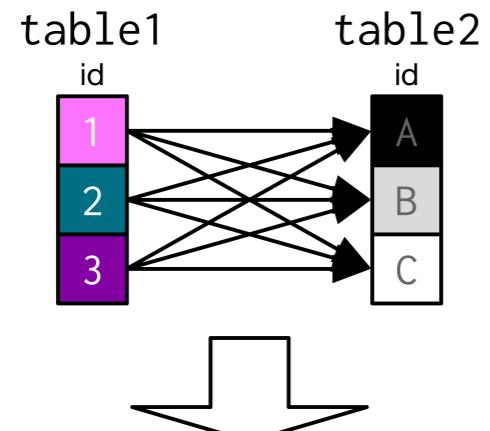
LEFT JOIN

L_id	L_val	R_val
1	L1	R1
2	L2	
3	L3	
4	L4	R2

# RIGHT JOIN vs FULL JOIN



# CROSS JOIN with code

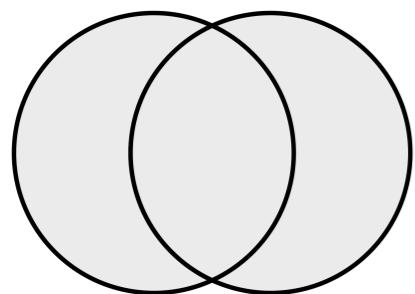


CROSS JOIN

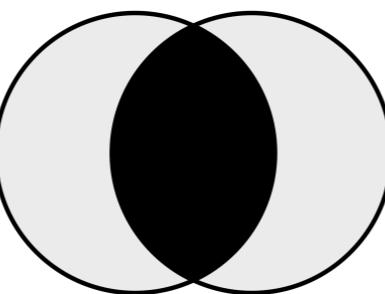
	id1	id2
1	1	A
1	1	B
1	1	C
2	2	A
2	2	B
2	2	C
3	3	A
3	3	B
3	3	C

```
SELECT table1.id AS id1,  
       table2.id AS id2  
  FROM table1  
CROSS JOIN table2;
```

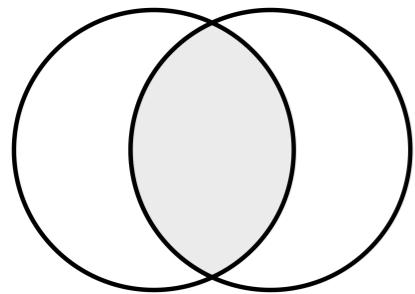
# Set Theory Clauses



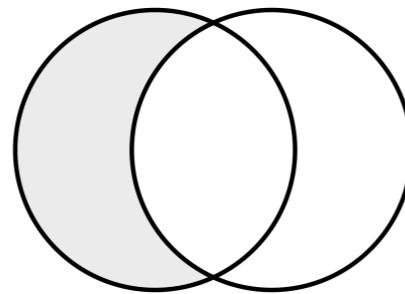
UNION



UNION ALL



INTERSECT



EXCEPT

# Semi-joins and Anti-joins

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

Semi-join

L\_id L\_val

1	L1
4	L4

Anti-join

L\_id L\_val

2	L2
3	L3

# Types of basic subqueries

- Subqueries inside WHERE clauses
- Subqueries inside SELECT clauses
- Subqueries inside FROM clauses

**Own the challenge  
problems! You got  
this!**

**JOINING DATA IN SQL**