

## Set theory Clauses

### UNION

- Includes records from both tables but does not replicate duplicate records
- UNION operation can also be used to remove duplicates across a field in two tables.

### UNION ALL

- Includes records from both tables but does replicate duplicate records (Duplicate records present so centre is shaded black)

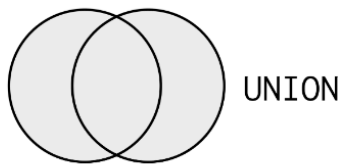
### INTERSECT

- Records found in both table

### EXCEPT

- Records found in one table and not the other

## Set Theory Venn Diagrams



UNION



UNION ALL



INTERSECT



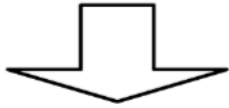
EXCEPT

left\_one

id
1
2
3
4

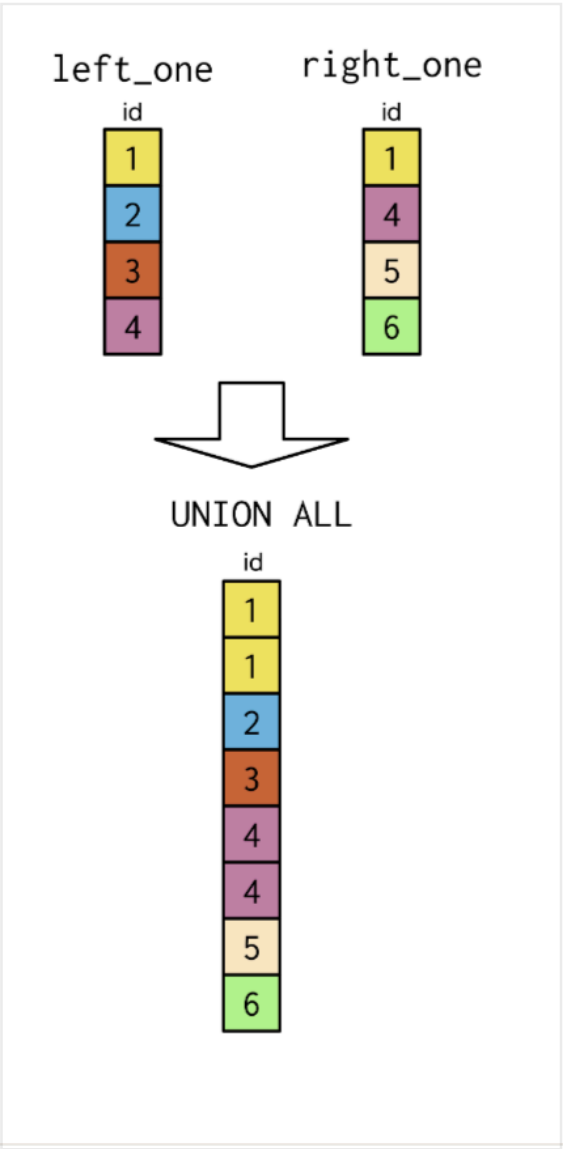
right\_one

id
1
4
5
6



UNION

id
1
2
3
4
5
6



**UNION**

```
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    |
+-----+-----+
```

Note: For Set operations the fields included in the operation must be of the same data type as they come back as a single data type.

Union and Union all do not do the look up step that the joins do, they just stack one table on top of another.

## UNION Example

```
SELECT *
FROM economies2010
UNION
SELECT *
FROM economies2015
ORDER BY code, year;
```

- This is similar to the vertical concatenation that we performed with `pd.concat` and we specified the axis to be vertical.
- Note: The above query can only be executed if `economies2010` and `economies2015` both have the same column types in the same order.

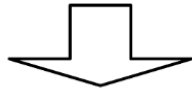
## INTERSECT

- Only includes common records from both tables

# INTERSECT diagram and SQL code

left\_one      right\_one

id	id
1	1
2	4
3	5
4	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 |
+-----+-----+
+-----+-----+
```

- an empty table was returned because, intersect looks at both the tables and then try to match records that have the same values, and in our example we did not have any records having the same president and prime minister.

## EXCEPT

- Allows us to select records that are in one table and not the other.

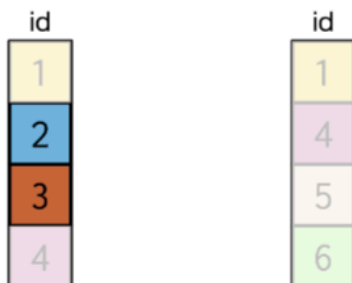


# Monarchs that aren't prime ministers

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

```
+-----+-----+
| monarch | country |
+-----+-----+
| Harald V | Norway  |
| Felipe VI | Spain   |
+-----+-----+
```

left\_one      right\_one



EXCEPT



LEFT - RIGHT