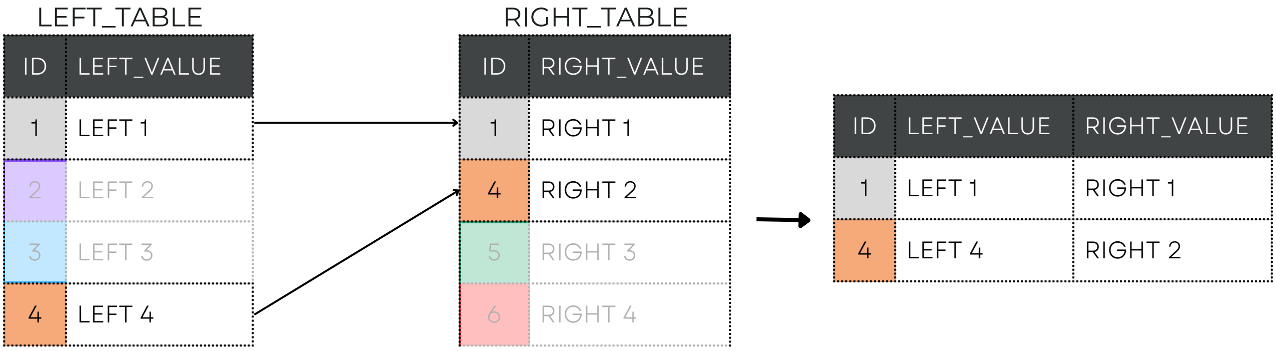
SQL Joins

v/s

Python Pandas

@vimanyuchaturvedi

INNER JOIN



LEFT_TABLE

ID	LEFT_VALUE
1	LEFT 1
2	LEFT 2
3	LEFT 3
4	LEFT 4

RIGHT_TABLE

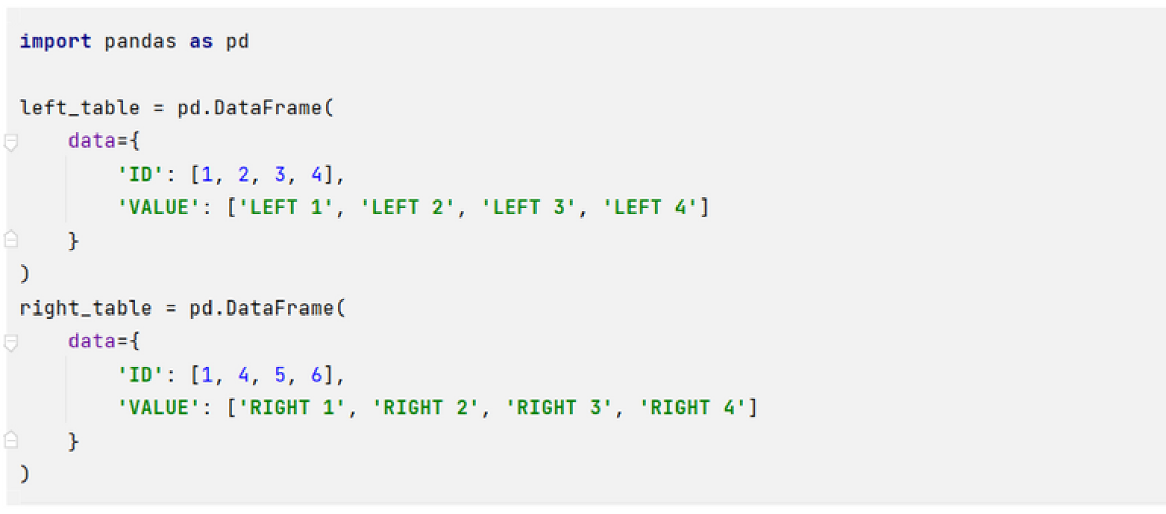
ID	RIGHT_VALUE
1	RIGHT 1
4	RIGHT 2
5	RIGHT 3
6	RIGHT 4

ID	LEFT_VALUE	RIGHT_VALUE
1	LEFT 1	RIGHT 1
4	LEFT 4	RIGHT 2

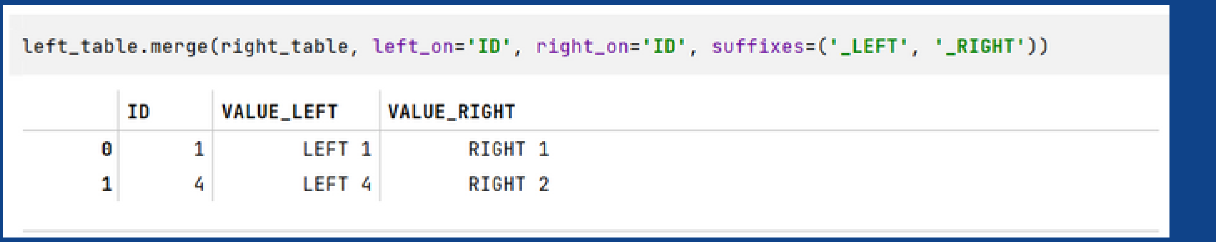
SQL

SELECT * FROM LEFT_TABLE AS LT INNER JOIN RIGHT_TABLE AS RT
ON LT.ID = RT.ID

PANDAS



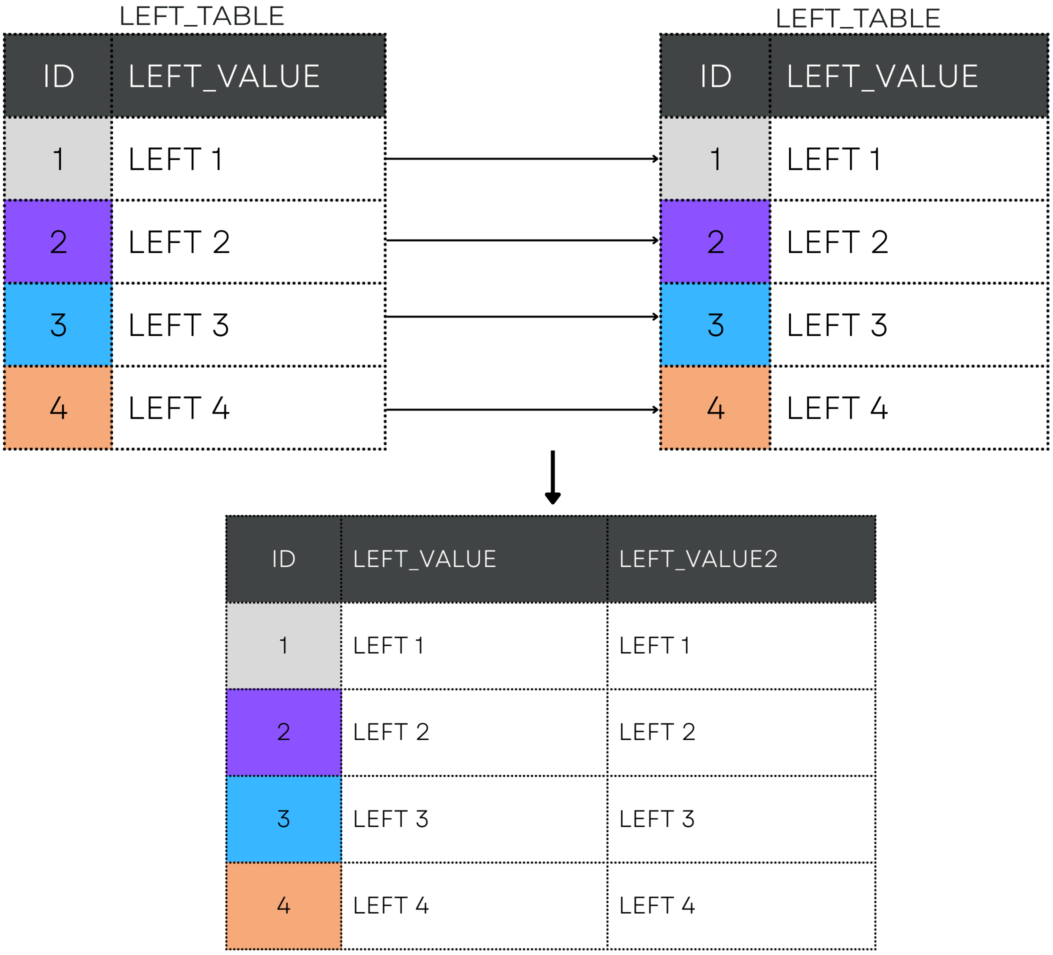
```python
import pandas as pd

left_table = pd.DataFrame(
    data={
        'ID': [1, 2, 3, 4],
        'VALUE': ['LEFT 1', 'LEFT 2', 'LEFT 3', 'LEFT 4']
    }
)
right_table = pd.DataFrame(
    data={
        'ID': [1, 4, 5, 6],
        'VALUE': ['RIGHT 1', 'RIGHT 2', 'RIGHT 3', 'RIGHT 4']
    }
)
```



```python
left_table.merge(right_table, left_on='ID', right_on='ID', suffixes=('_LEFT', '_RIGHT'))
```

```
     ID   VALUE_LEFT   VALUE_RIGHT
0     1       LEFT 1       RIGHT 1
1     4       LEFT 4       RIGHT 2
```

SELF JOIN



LEFT_TABLE

ID	LEFT_VALUE
1	LEFT 1
2	LEFT 2
3	LEFT 3
4	LEFT 4

LEFT_TABLE

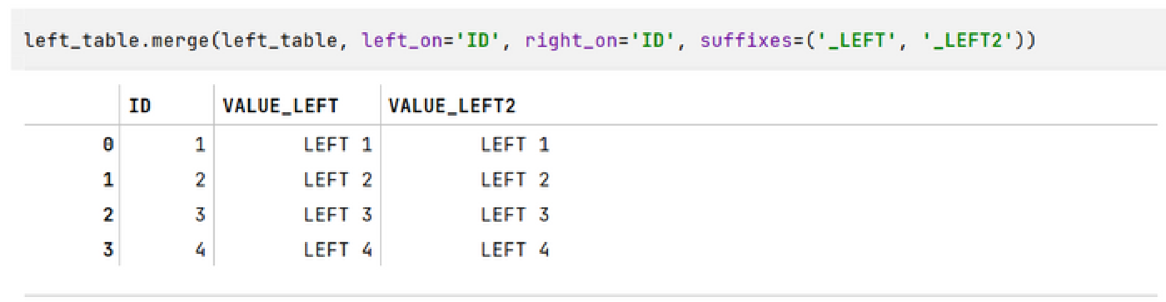
ID	LEFT_VALUE
1	LEFT 1
2	LEFT 2
3	LEFT 3
4	LEFT 4

ID	LEFT_VALUE	LEFT_VALUE2
1	LEFT 1	LEFT 1
2	LEFT 2	LEFT 2
3	LEFT 3	LEFT 3
4	LEFT 4	LEFT 4

SQL

SELECT * FROM LEFT_TABLE AS LT INNER JOIN LEFT_TABLE AS LT2
ON LT.ID = LT2.ID

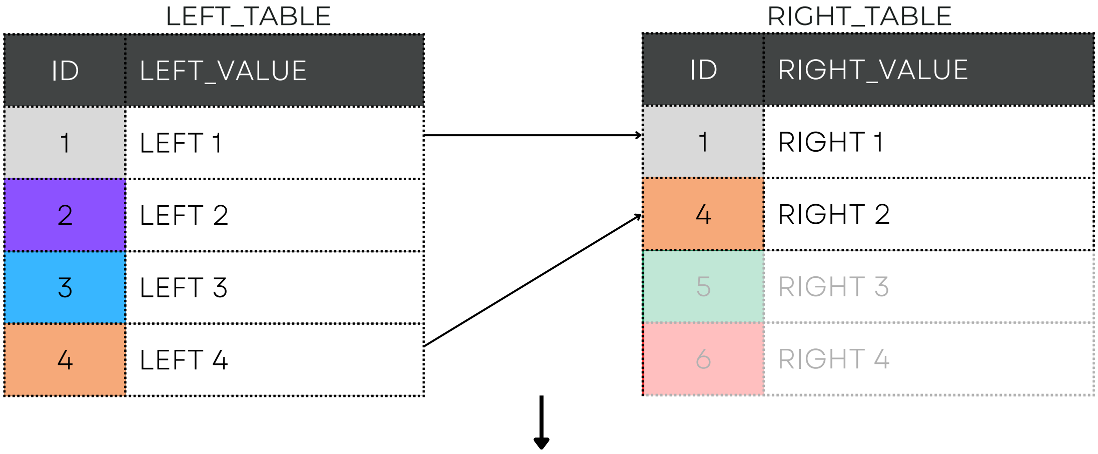
PANDAS



```
left_table.merge(left_table, left_on='ID', right_on='ID', suffixes=('_LEFT', '_LEFT2'))
```

```
   ID  VALUE_LEFT  VALUE_LEFT2
0   1      LEFT 1       LEFT 1
1   2      LEFT 2       LEFT 2
2   3      LEFT 3       LEFT 3
3   4      LEFT 4       LEFT 4
```

LEFT JOIN



LEFT_TABLE

ID	LEFT_VALUE
1	LEFT 1
2	LEFT 2
3	LEFT 3
4	LEFT 4

RIGHT_TABLE

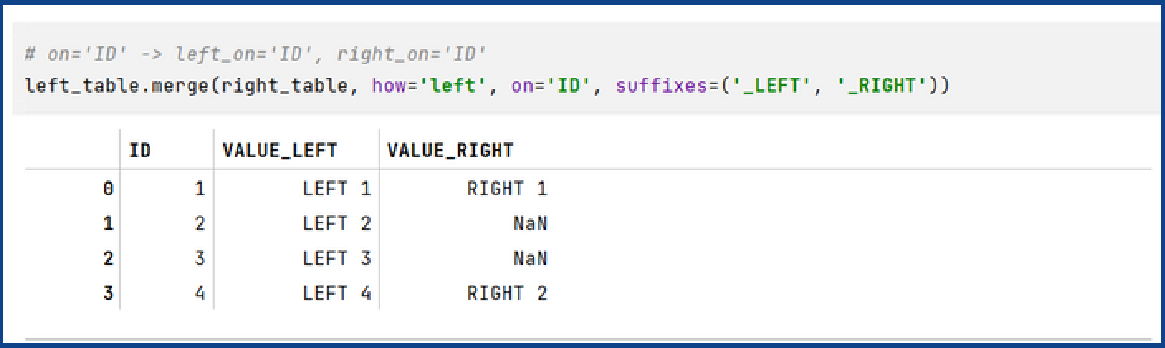
ID	RIGHT_VALUE
1	RIGHT 1
4	RIGHT 2
5	RIGHT 3
6	RIGHT 4

ID	LEFT_VALUE	RIGHT_VALUE
1	LEFT 1	RIGHT 1
2	LEFT 2	NULL
3	LEFT 3	NULL
4	LEFT 4	RIGHT 2

SQL

SELECT * FROM LEFT_TABLE AS LT LEFT JOIN RIGHT_TABLE AS RT ON LT.ID = RT.ID

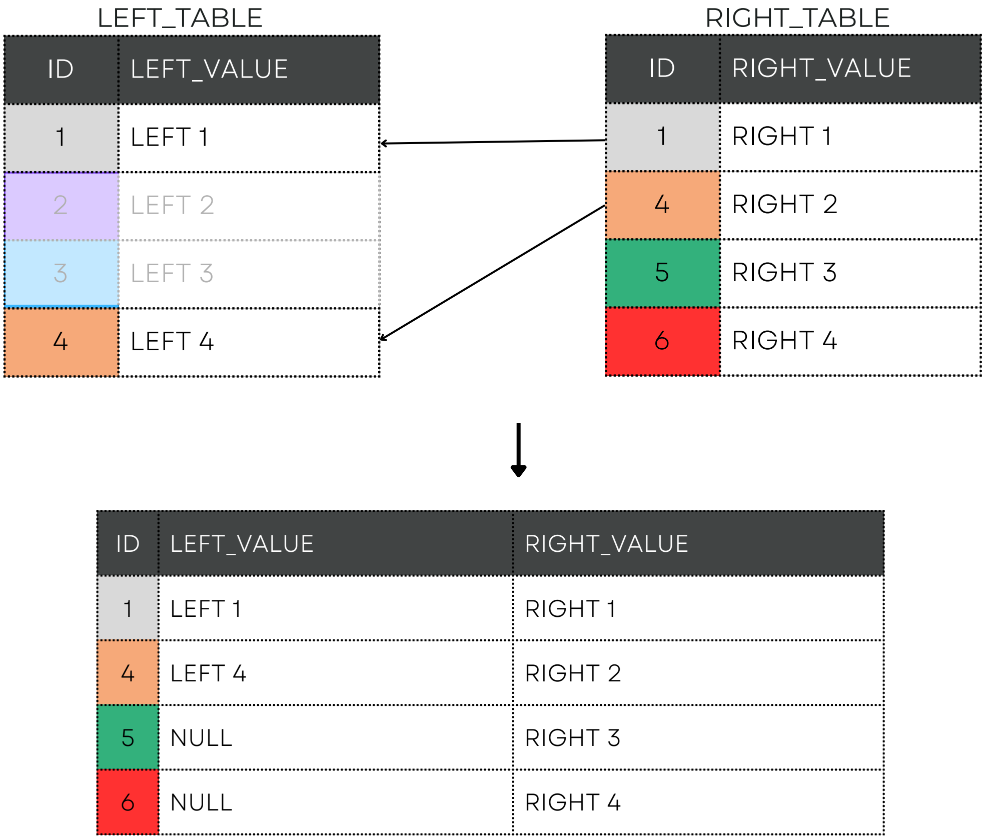
PANDAS



```
# on='ID' -> left_on='ID', right_on='ID'
left_table.merge(right_table, how='left', on='ID', suffixes=('_LEFT', '_RIGHT'))
```

```
   ID  VALUE_LEFT  VALUE_RIGHT
0   1      LEFT 1      RIGHT 1
1   2      LEFT 2          NaN
2   3      LEFT 3          NaN
3   4      LEFT 4      RIGHT 2
```

RIGHT JOIN



LEFT_TABLE

ID	LEFT_VALUE
1	LEFT 1
2	LEFT 2
3	LEFT 3
4	LEFT 4

RIGHT_TABLE

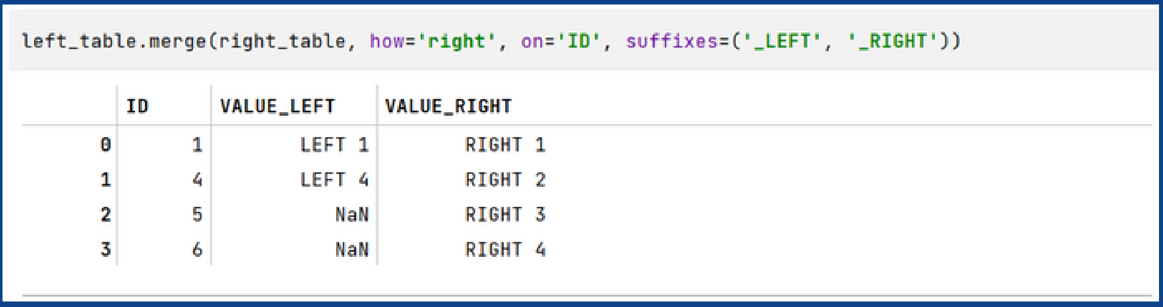
ID	RIGHT_VALUE
1	RIGHT 1
4	RIGHT 2
5	RIGHT 3
6	RIGHT 4

ID	LEFT_VALUE	RIGHT_VALUE
1	LEFT 1	RIGHT 1
4	LEFT 4	RIGHT 2
5	NULL	RIGHT 3
6	NULL	RIGHT 4

SQL

SELECT * FROM LEFT_TABLE AS LT RIGHT JOIN RIGHT_TABLE AS RT ON LT.ID = RT.ID

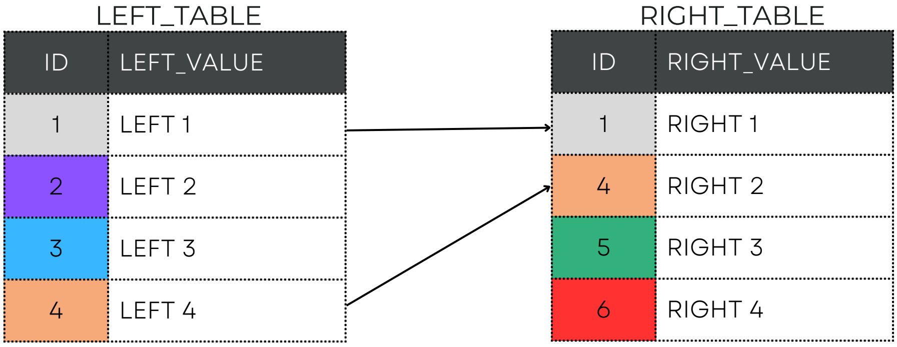
PANDAS



```
left_table.merge(right_table, how='right', on='ID', suffixes=('_LEFT', '_RIGHT'))
```

```
    ID  VALUE_LEFT  VALUE_RIGHT
0    1      LEFT 1      RIGHT 1
1    4      LEFT 4      RIGHT 2
2    5         NaN      RIGHT 3
3    6         NaN      RIGHT 4
```

FULL JOIN



LEFT_TABLE

ID	LEFT_VALUE
1	LEFT 1
2	LEFT 2
3	LEFT 3
4	LEFT 4

RIGHT_TABLE

ID	RIGHT_VALUE
1	RIGHT 1
4	RIGHT 2
5	RIGHT 3
6	RIGHT 4

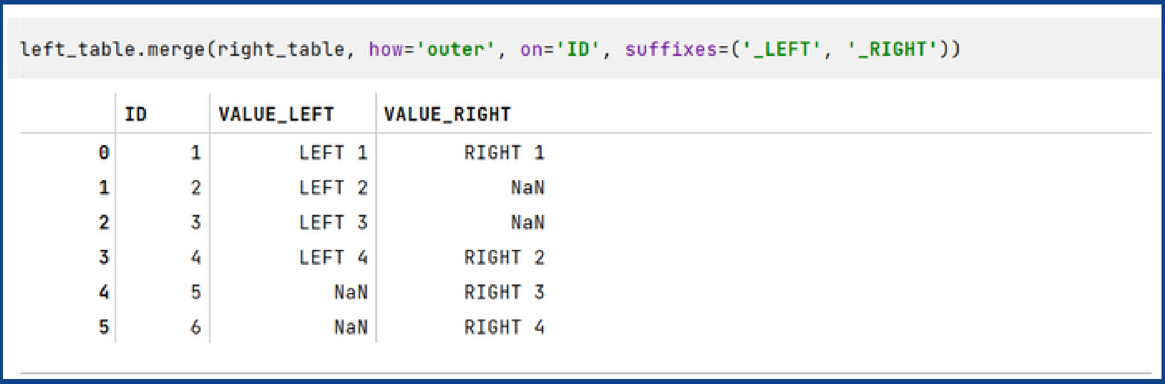
↓

ID	LEFT_VALUE	RIGHT_VALUE
1	LEFT 1	RIGHT 1
2	LEFT 2	NULL
3	LEFT 3	NULL
4	LEFT 4	RIGHT 2
5	NULL	RIGHT 3
6	NULL	RIGHT 4

SQL

```
SELECT * FROM LEFT_TABLE AS LT FULL OUTER JOIN RIGHT_TABLE
AS RT ON LT.ID = RT.ID
```

PANDAS



```
left_table.merge(right_table, how='outer', on='ID', suffixes=('_LEFT', '_RIGHT'))
```

```
   ID  VALUE_LEFT  VALUE_RIGHT
0   1      LEFT 1      RIGHT 1
1   2      LEFT 2          NaN
2   3      LEFT 3          NaN
3   4      LEFT 4      RIGHT 2
4   5         NaN      RIGHT 3
5   6         NaN      RIGHT 4
```

UNION ALL

LEFT_TABLE

ID	VALUE
1	VALUE 1
2	VALUE 2
3	VALUE 3
4	VALUE 4

RIGHT_TABLE

ID	VALUE
1	VALUE 1
4	VALUE 2
5	VALUE 3
6	VALUE 4

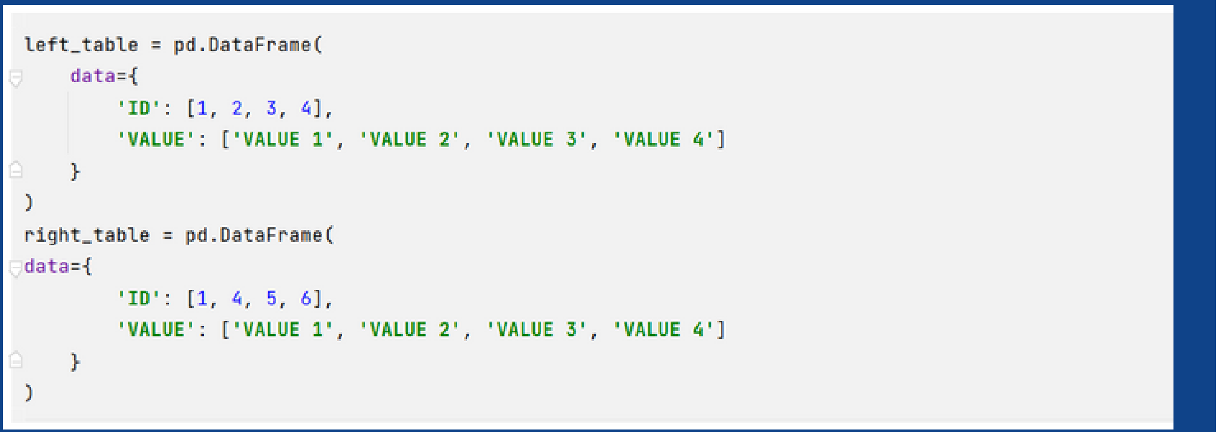
→

ID	VALUE
1	VALUE 1
2	VALUE 2
3	VALUE 3
4	VALUE 4
1	VALUE 1
4	VALUE 2
5	VALUE 3
6	VALUE 4

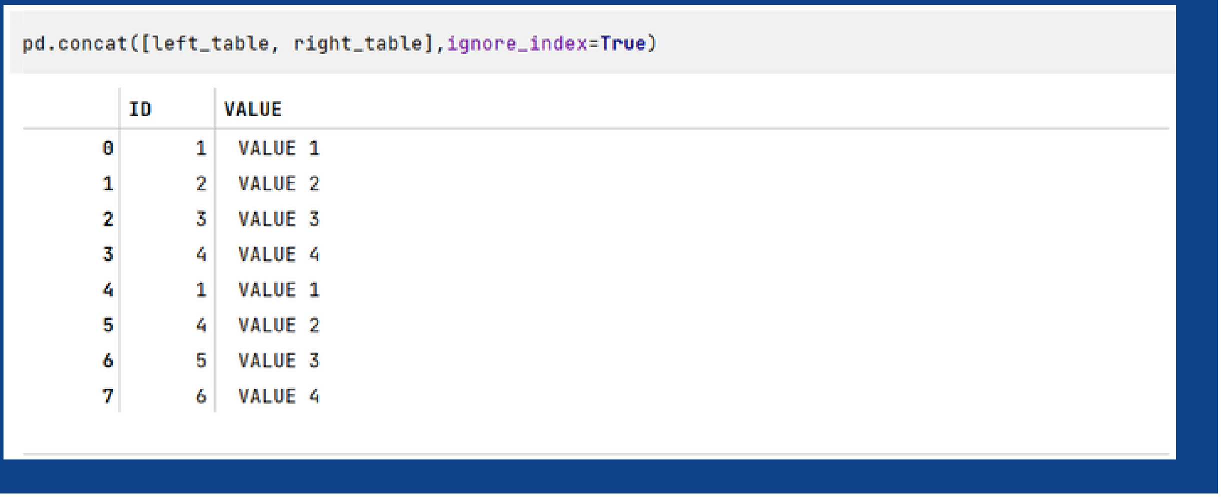
SQL

SELECT * FROM LEFT_TABLE UNION ALL SELECT * FROM RIGHT_TABLE

PANDAS



```
left_table = pd.DataFrame(
    data={
        'ID': [1, 2, 3, 4],
        'VALUE': ['VALUE 1', 'VALUE 2', 'VALUE 3', 'VALUE 4']
    }
)
right_table = pd.DataFrame(
data={
        'ID': [1, 4, 5, 6],
        'VALUE': ['VALUE 1', 'VALUE 2', 'VALUE 3', 'VALUE 4']
    }
)
```



```
pd.concat([left_table, right_table],ignore_index=True)
```

```
     ID   VALUE
0     1   VALUE 1
1     2   VALUE 2
2     3   VALUE 3
3     4   VALUE 4
4     1   VALUE 1
5     4   VALUE 2
6     5   VALUE 3
7     6   VALUE 4
```

UNION

LEFT_TABLE

ID	VALUE
1	VALUE 1
2	VALUE 2
3	VALUE 3
4	VALUE 4

RIGHT_TABLE

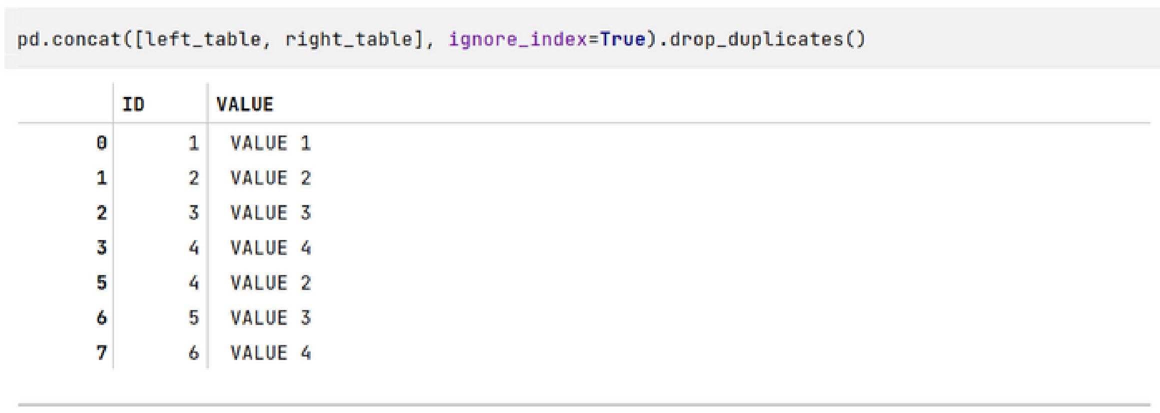
ID	VALUE
1	VALUE 1
4	VALUE 2
5	VALUE 3
6	VALUE 4

ID	VALUE
1	VALUE 1
2	VALUE 2
3	VALUE 3
4	VALUE 4
4	VALUE 2
5	VALUE 3
6	VALUE 4

SQL

SELECT * FROM LEFT_TABLE UNION SELECT * FROM RIGHT_TABLE

PANDAS



```
pd.concat([left_table, right_table], ignore_index=True).drop_duplicates()
```

```
      ID   VALUE
0      1   VALUE 1
1      2   VALUE 2
2      3   VALUE 3
3      4   VALUE 4
5      4   VALUE 2
6      5   VALUE 3
7      6   VALUE 4
```

INTERSECT

LEFT_TABLE

ID	VALUE
1	VALUE 1
2	VALUE 2
3	VALUE 3
4	VALUE 4

RIGHT_TABLE

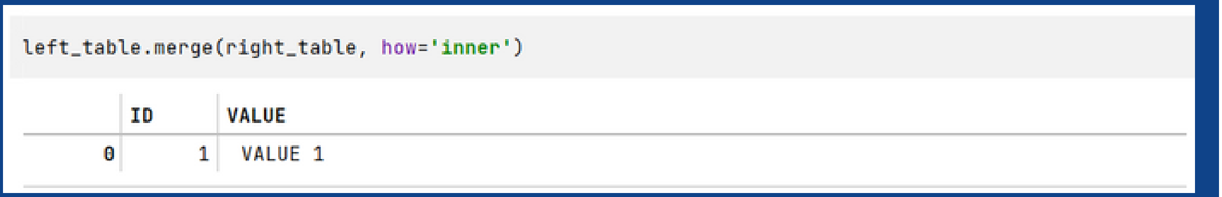
ID	VALUE
1	VALUE 1
4	VALUE 2
5	VALUE 3
6	VALUE 4

ID	VALUE
1	VALUE 1

SQL

```
SELECT *  FROM LEFT_TABLE  INTERSECT SELECT * FROM  RIGHT_TABLE
```

PANDAS



```
left_table.merge(right_table, how='inner')
```

```
     ID   VALUE
0     1   VALUE 1
```

EXCEPT

LEFT_TABLE

ID	VALUE
1	VALUE 1
2	VALUE 2
3	VALUE 3
4	VALUE 4

RIGHT_TABLE

ID	VALUE
1	VALUE 1
4	VALUE 2
5	VALUE 3
6	VALUE 4

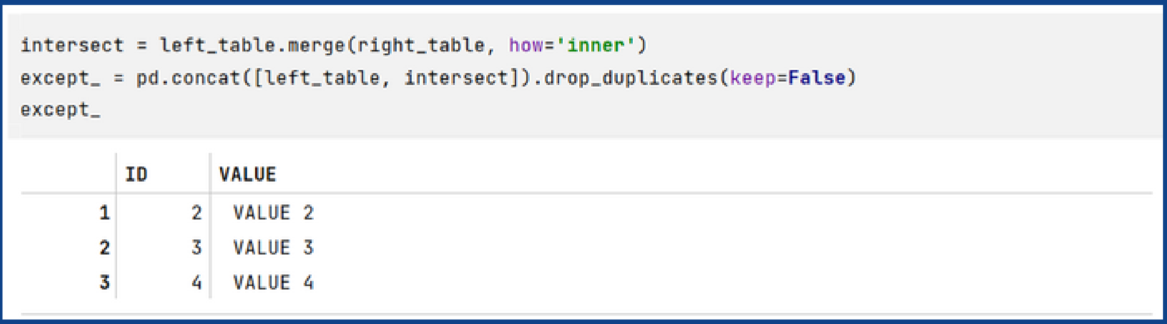
↓

ID	VALUE
2	VALUE 2
3	VALUE 3
4	VALUE 4

SQL

SELECT * FROM LEFT_TABLE EXCEPT SELECT * FROM RIGHT_TABLE

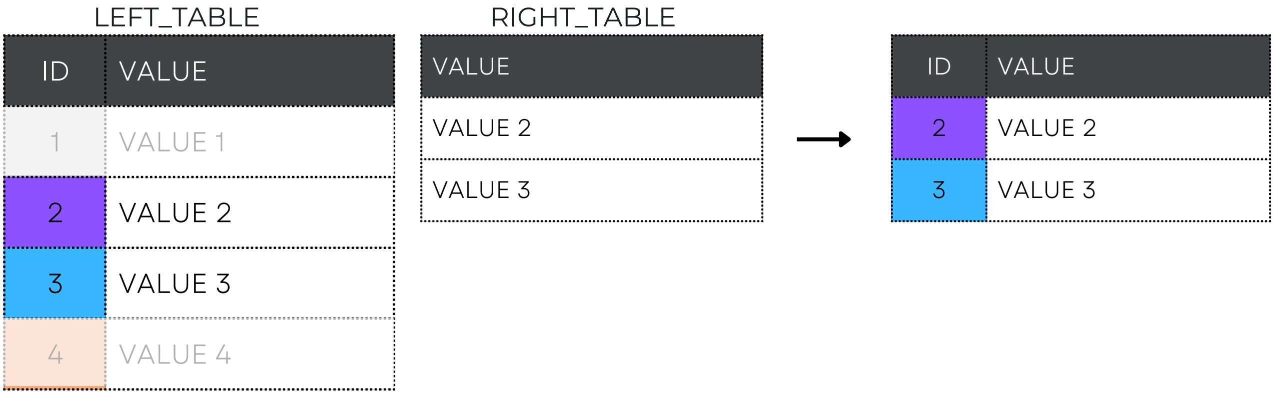
PANDAS



```
intersect = left_table.merge(right_table, how='inner')
except_ = pd.concat([left_table, intersect]).drop_duplicates(keep=False)
except_
```

```
     ID    VALUE
1     2    VALUE 2
2     3    VALUE 3
3     4    VALUE 4
```

SEMI JOIN



LEFT_TABLE

ID	VALUE
1	VALUE 1
2	VALUE 2
3	VALUE 3
4	VALUE 4

RIGHT_TABLE

VALUE
VALUE 2
VALUE 3

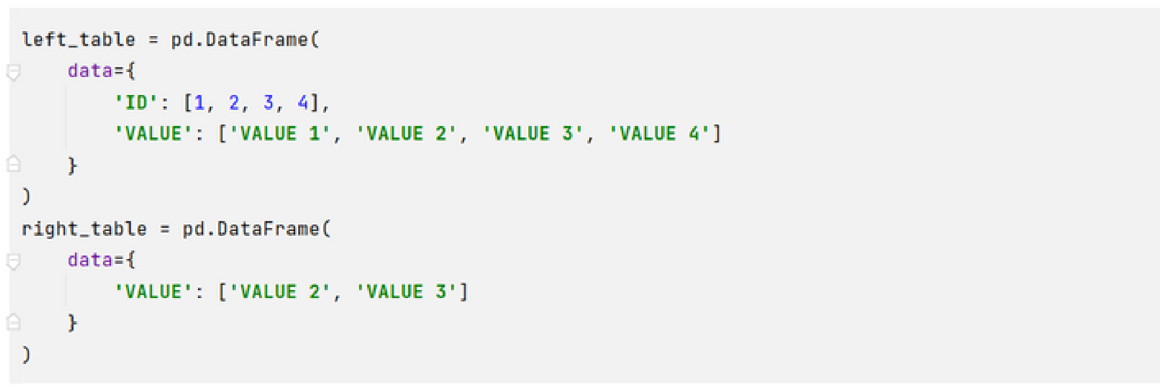
→

ID	VALUE
2	VALUE 2
3	VALUE 3

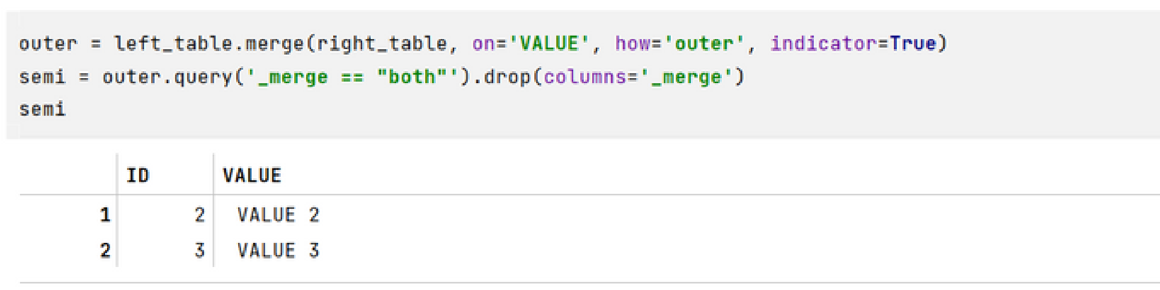
SQL

SELECT * FROM LEFT_TABLE WHERE VALUE IN (SELECT VALUE FROM RIGHT_TABLE)

PANDAS



```
left_table = pd.DataFrame(
    data={
        'ID': [1, 2, 3, 4],
        'VALUE': ['VALUE 1', 'VALUE 2', 'VALUE 3', 'VALUE 4']
    }
)
right_table = pd.DataFrame(
    data={
        'VALUE': ['VALUE 2', 'VALUE 3']
    }
)
```



```
outer = left_table.merge(right_table, on='VALUE', how='outer', indicator=True)
semi = outer.query('_merge == "both"').drop(columns='_merge')
semi
```

```
     ID   VALUE
1     2   VALUE 2
2     3   VALUE 3
```

ANTI JOIN

LEFT_TABLE

ID	VALUE
1	VALUE 1
2	VALUE 2
3	VALUE 3
4	VALUE 4

RIGHT_TABLE

VALUE
VALUE 2
VALUE 3

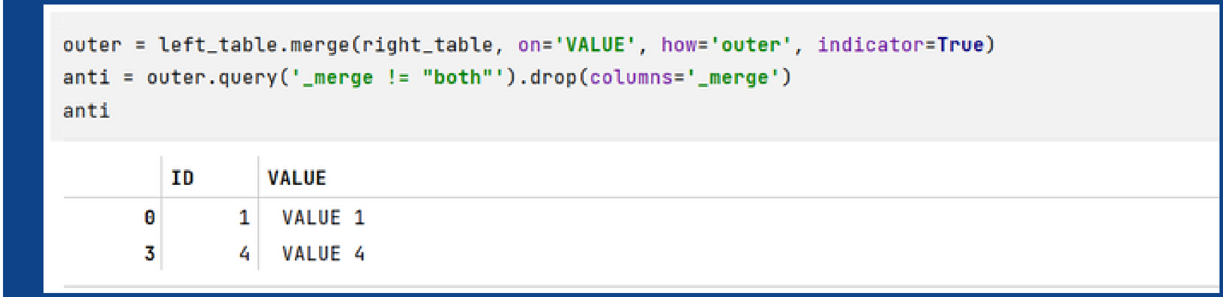
→

ID	VALUE
1	VALUE 1
4	VALUE 4

SQL

SELECT * FROM LEFT_TABLE WHERE VALUE NOT IN (SELECT VALUE FROM RIGHT_TABLE)

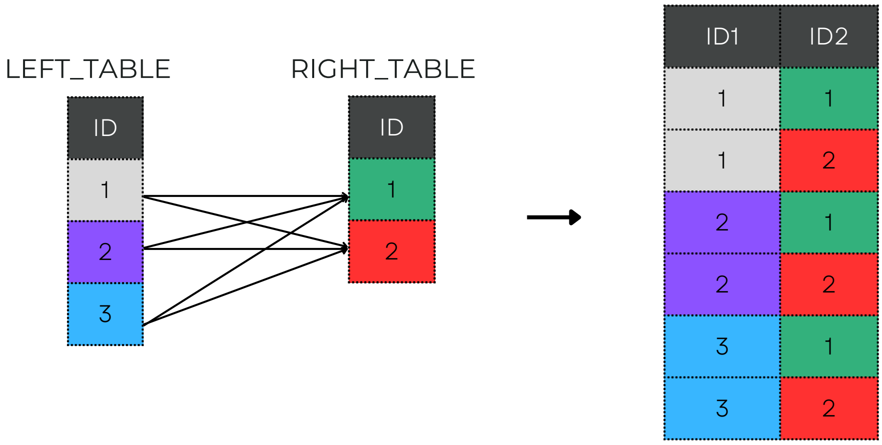
PANDAS



```
outer = left_table.merge(right_table, on='VALUE', how='outer', indicator=True)
anti = outer.query('_merge != "both"').drop(columns='_merge')
anti
```

```
   ID   VALUE
0   1   VALUE 1
3   4   VALUE 4
```

CROSS JOIN

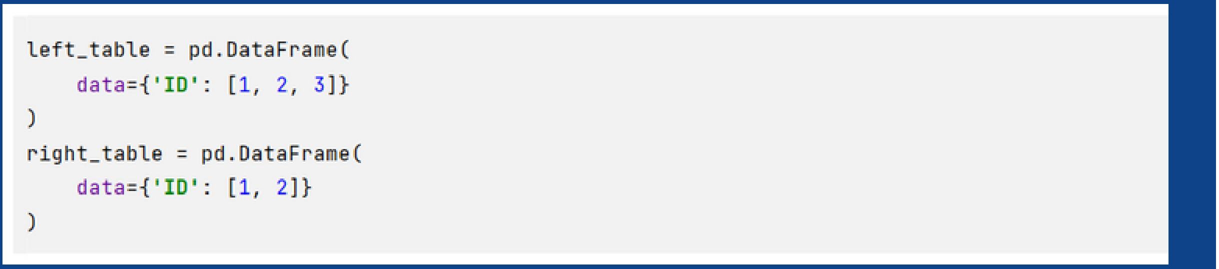


ID1	ID2
1	1
1	2
2	1
2	2
3	1
3	2

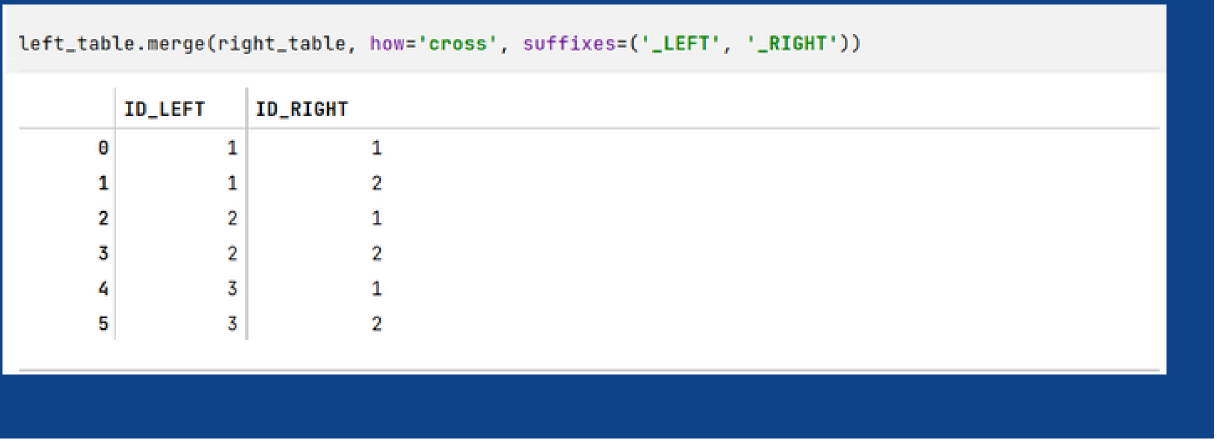
SQL

SELECT * FROM LEFT_TABLE CROSS JOIN RIGHT_TABLE

PANDAS



```
left_table = pd.DataFrame(
    data={'ID': [1, 2, 3]}
)
right_table = pd.DataFrame(
    data={'ID': [1, 2]}
)
```



```
left_table.merge(right_table, how='cross', suffixes=('_LEFT', '_RIGHT'))
```

```
     ID_LEFT  ID_RIGHT
0          1         1
1          1         2
2          2         1
3          2         2
4          3         1
5          3         2
```