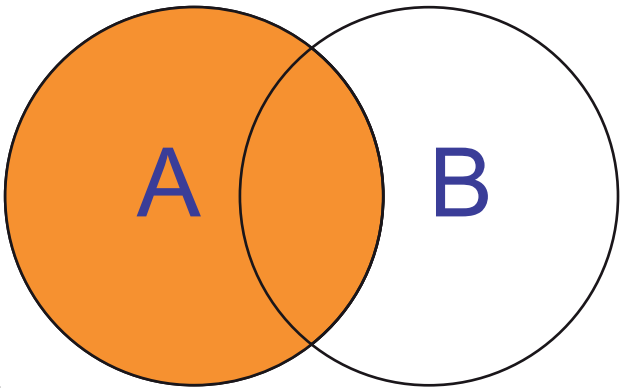
SQL-Joins bzw. Python pandas Objektvereinigungen

A bzw. B:

- in SQL eine Menge an Werten, die aus der Abfrage über eine Spalte einer Tabelle entstehen
- in pandas ein pandas.Series-Objekt (eindimensionales ndarray mit Achsen-Namen), das ggf. aus der Abfrage eines pandas.DataFrame (zweidimensionale, größenveränderliche, möglicherweise verschieden typisierte Tabellendaten) resultiert
- für Python pandas gilt: „left“ = 1st Dataframe; „right“ = 2nd Dataframe

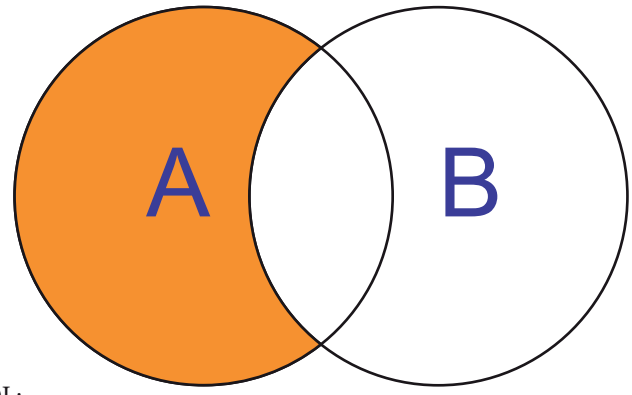


SQL:

```
SELECT * FROM A LEFT JOIN B ON A.Key = B.Key
```

Python pandas:

```
A.join(B, on=['A.Key'], rsuffix='_B')
```

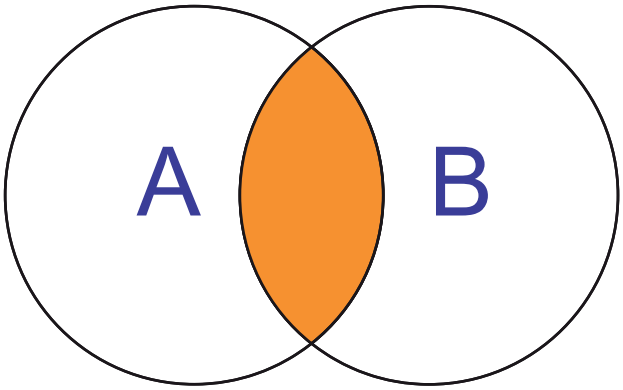


SQL:

```
SELECT * FROM A LEFT JOIN B ON A.Key = B.Key
WHERE B.Key IS NULL
```

Python pandas:

```
A[~A['A.Key'].isin(B['B.Key'])
```

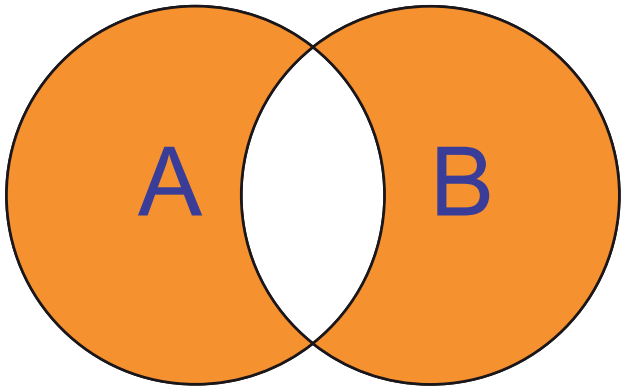


SQL:

```
SELECT * FROM A INNER JOIN B ON A.Key = B.Key
```

Python pandas:

```
A.join(B, on=['A.Key'], how='inner')
```

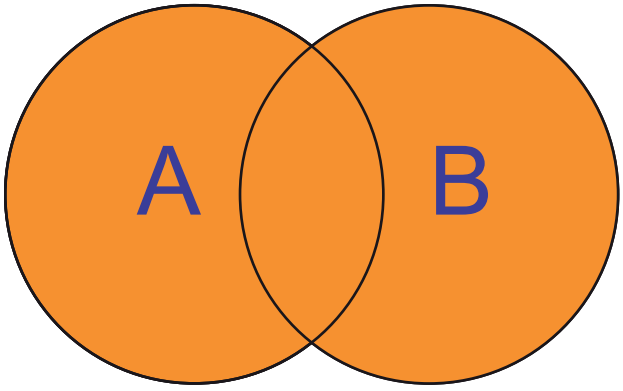


SQL:

```
SELECT * FROM A FULL OUTER JOIN B ON A.Key =
B.Key WHERE A.Key IS NULL OR B.Key IS NULL
```

Python pandas:

```
pd.concat([A, B].loc[lambda df: ~df.index.
duplicated()]
```



SQL:

```
SELECT * FROM A FULL OUTER JOIN B ON
A.Key = B.Key
```

Python pandas:

```
pd.merge(A, B, how='outer', on=['A.Key','B.
Key'], validate="one_to_one", indicator=True)
```

Weitere Varianten und komplexere Vereinigungen unter:
https://pandas.pydata.org/pandas-docs/stable/user_guide/merging.html