PANDAS PART 3

- **merge(), join()**
- **concat(), append()**

Joining the Tables (or Data Frames)

When we want to combine the records from tables, there are 4 types of joining. Joining the tables can be done only when there is a common field. In the following tables, ‘Player Name’ is the common field.

Players-age.xlsx

Player Name	Age
Rahul Dravid	30
Virat kohle	24
Vinod kamble	22

Players-salary.xlsx

Player Name	Salary (in lakhs)
Rahul Dravid	45
Virat kohle	60
Harbajan singh	44

1. Inner join: gives intersection of records. That means the records which appear commonly in both the tables. This is the default joining for the tables.

Player Name	Age	Salary (in lakhs)
Rahul Dravid	30	45
Virat kohle	24	60

2. Left join: the left side table will have all records. To them, the records from right table will be joined.

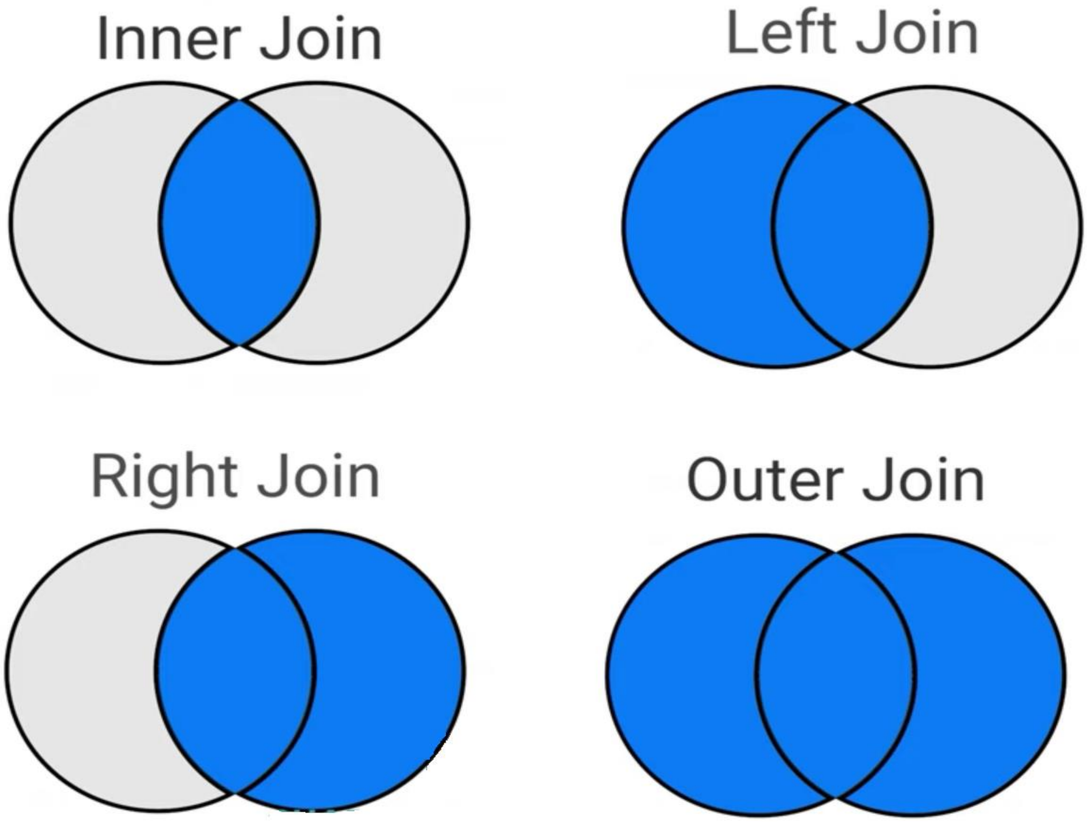
Player Name	Age	Salary (in lakhs)
Rahul Dravid	30	45
Virat kohle	24	60
Vinod kamble	22	NaN

3. Right join: the right side table will have all records. To them, the records from left side table are joined.

Player Name	**Age**	**Salary (in lakhs)**
Rahul Dravid	30	45
Virat kohle	24	60
Harbajan singh	NaN	44

4. Full outer join: All the records of both the tables are joined.

Player Name	**Age**	**Salary (in lakhs)**
Rahul Dravid	30	45
Virat kohle	24	60
Vinod kamble	22	NaN
Harbajan singh	NaN	44



Inner Join

Left Join

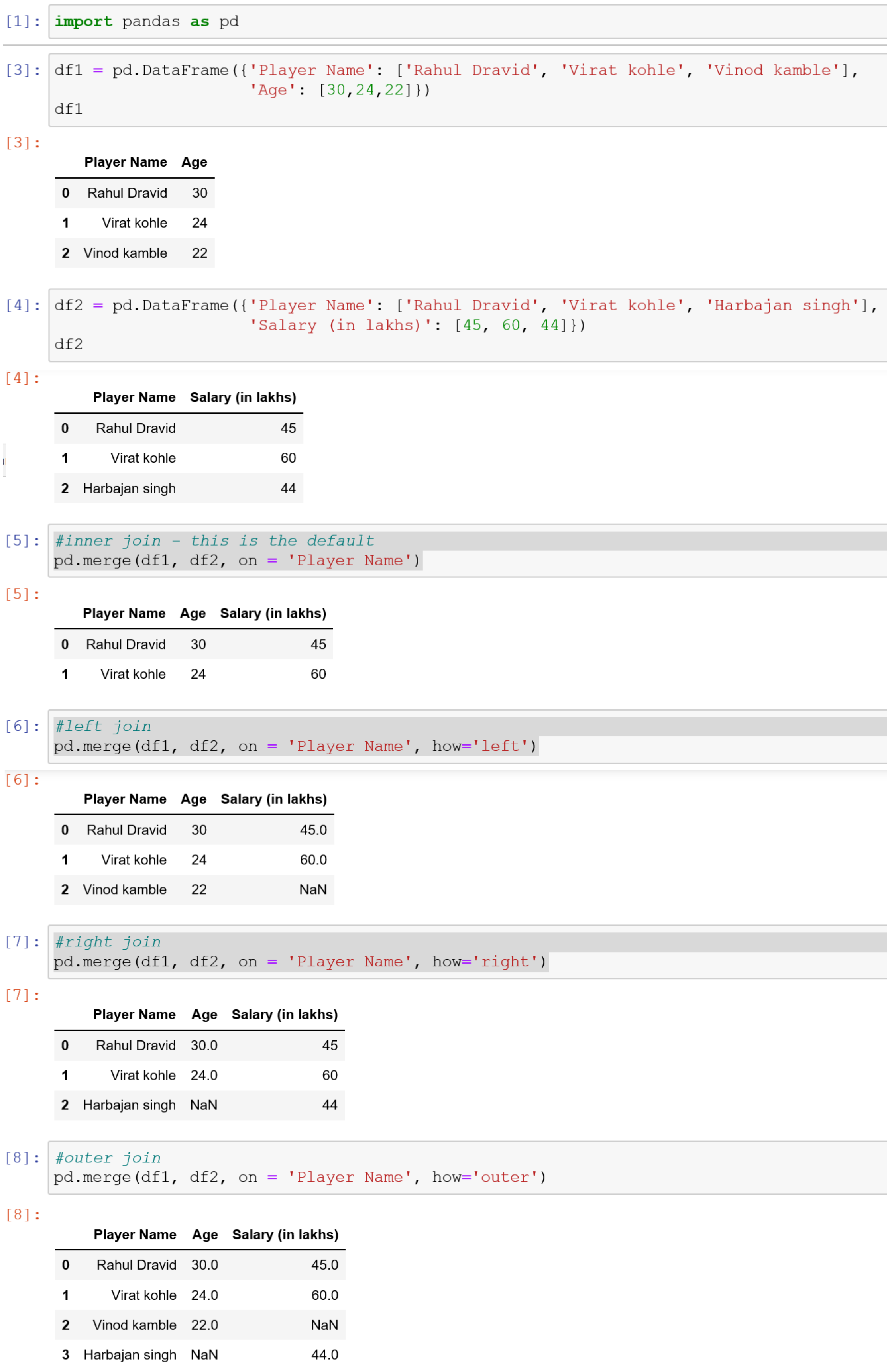
Right Join

Outer Join

How to do in pandas

Joining the tables can be done in pandas using merge() and join() methods. Both work same.

```
merge(df1, df2, on='col', how='inner') # default merge is inner
df1.join(df2, on='col', how='inner')  # default join is left
```



[1]:

```
import pandas as pd
```

[3]:

```
df1 = pd.DataFrame({'Player Name': ['Rahul Dravid', 'Virat kohle', 'Vinod kamble'],
                    'Age': [30,24,22]})
df1
```

[3]:

	Player Name	Age
0	Rahul Dravid	30
1	Virat kohle	24
2	Vinod kamble	22

[4]:

```
df2 = pd.DataFrame({'Player Name': ['Rahul Dravid', 'Virat kohle', 'Harbajan singh'],
                    'Salary (in lakhs)': [45, 60, 44]})
df2
```

[4]:

	Player Name	Salary (in lakhs)
0	Rahul Dravid	45
1	Virat kohle	60
2	Harbajan singh	44

[5]:

```
#inner join - this is the default
pd.merge(df1, df2, on = 'Player Name')
```

[5]:

	Player Name	Age	Salary (in lakhs)
0	Rahul Dravid	30	45
1	Virat kohle	24	60

[6]:

```
#left join
pd.merge(df1, df2, on = 'Player Name', how='left')
```

[6]:

	Player Name	Age	Salary (in lakhs)
0	Rahul Dravid	30	45.0
1	Virat kohle	24	60.0
2	Vinod kamble	22	NaN

[7]:

```
#right join
pd.merge(df1, df2, on = 'Player Name', how='right')
```

[7]:

	Player Name	Age	Salary (in lakhs)
0	Rahul Dravid	30.0	45
1	Virat kohle	24.0	60
2	Harbajan singh	NaN	44

[8]:

```
#outer join
pd.merge(df1, df2, on = 'Player Name', how='outer')
```

[8]:

	Player Name	Age	Salary (in lakhs)
0	Rahul Dravid	30.0	45.0
1	Virat kohle	24.0	60.0
2	Vinod kamble	22.0	NaN
3	Harbajan singh	NaN	44.0

How to join when there is NO common column

Even though there is no common column, we can join the tables if there are columns with common data is found. For example, in Players-age table we have ‘Player Name’ that contains players names. Similarly, in Players-salary table, we have ‘Sports Person’ that also contains players names. Hence, we can join these two tables on ‘Player Name’ in the left table and on ‘Sports Person’ in the right table, as:

```
pd.merge(df1, df2, left_on='Player Name', right_on= 'Sports Person')  # inner join
pd.merge(df1, df2, left_on='Player Name', right_on= 'Sports Person', how='left')  # left join
```

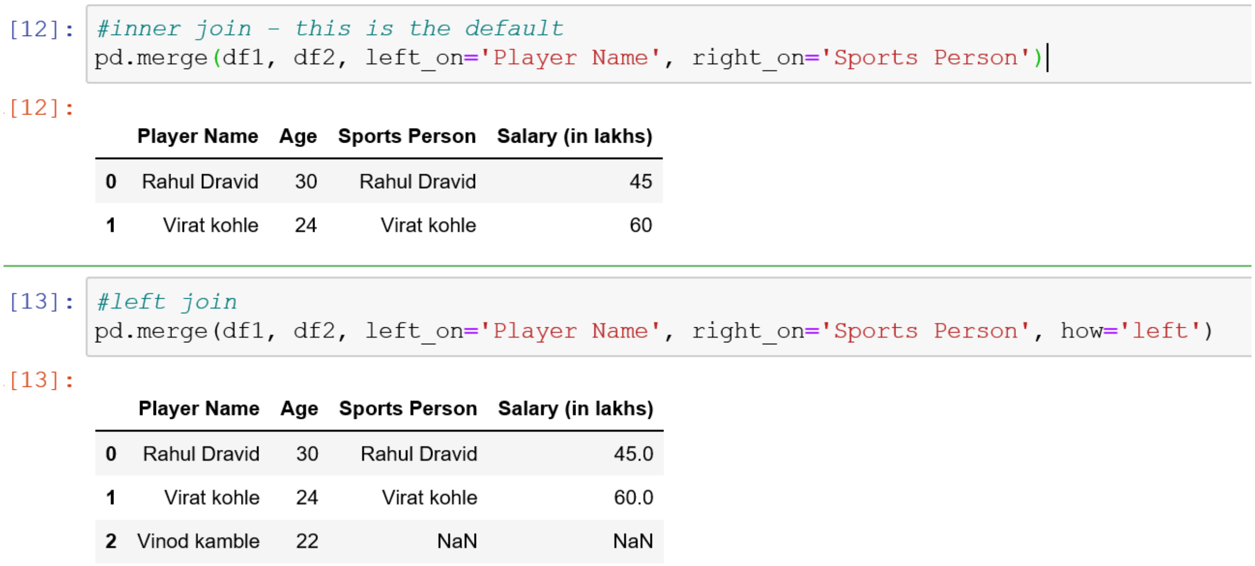
Players-age table

Player Name	Age
Rahul Dravid	30
Virat kohle	24
Vinod kamble	22

Players-salary table

Sports Person	Salary (in lakhs)
Rahul Dravid	45
Virat kohle	60
Harbajan singh	44

Examples



```
[12]: #inner join - this is the default
      pd.merge(df1, df2, left_on='Player Name', right_on='Sports Person')

[12]:
     Player Name  Age  Sports Person  Salary (in lakhs)
  0  Rahul Dravid  30   Rahul Dravid                 45
  1   Virat kohle  24    Virat kohle                 60
```

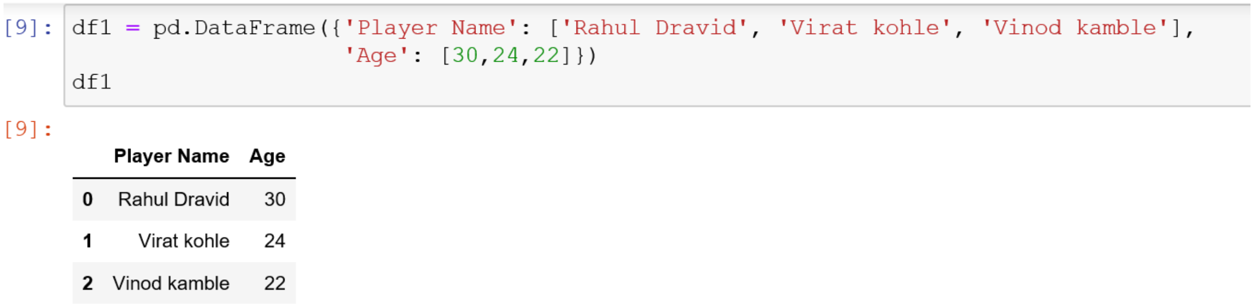
```
[13]: #left join
      pd.merge(df1, df2, left_on='Player Name', right_on='Sports Person', how='left')

[13]:
     Player Name  Age  Sports Person  Salary (in lakhs)
  0  Rahul Dravid  30   Rahul Dravid               45.0
  1   Virat kohle  24    Virat kohle               60.0
  2  Vinod kamble  22            NaN                NaN
```

Concatenation of tables

concat() can be used for attaching the tables side by side. This is called ‘union’ of data frames.

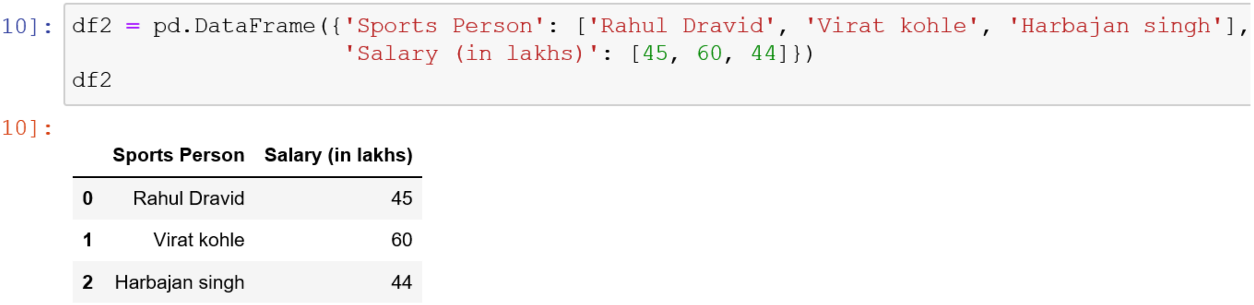
Let us take the following two Data Frames:



```
[9]: df1 = pd.DataFrame({'Player Name': ['Rahul Dravid', 'Virat kohle', 'Vinod kamble'],
                         'Age': [30,24,22]})
     df1
```

[9]:

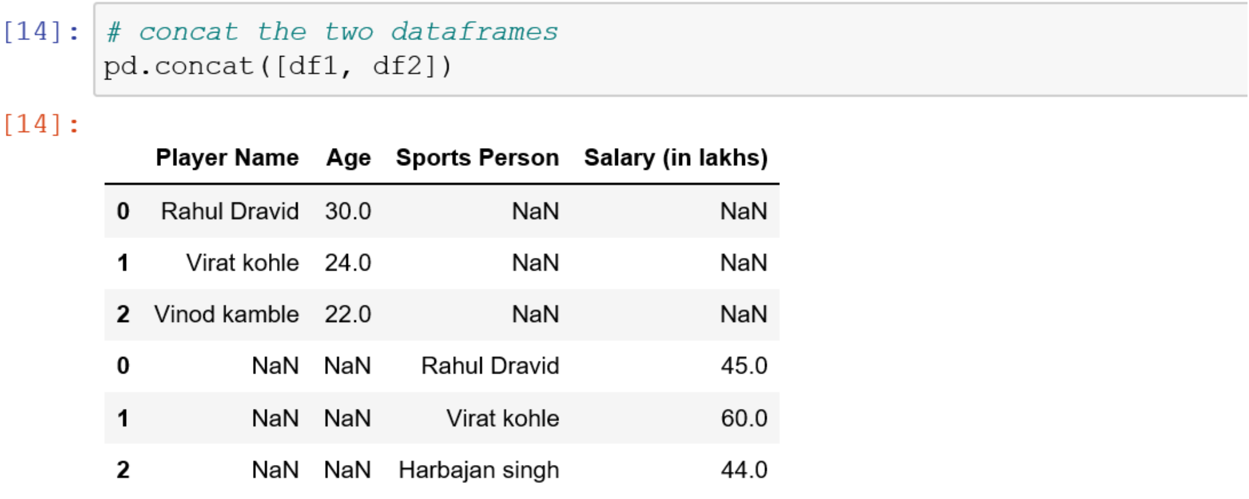
	Player Name	Age
0	Rahul Dravid	30
1	Virat kohle	24
2	Vinod kamble	22



```
10]: df2 = pd.DataFrame({'Sports Person': ['Rahul Dravid', 'Virat kohle', 'Harbajan singh'],
                         'Salary (in lakhs)': [45, 60, 44]})
     df2
```

10]:

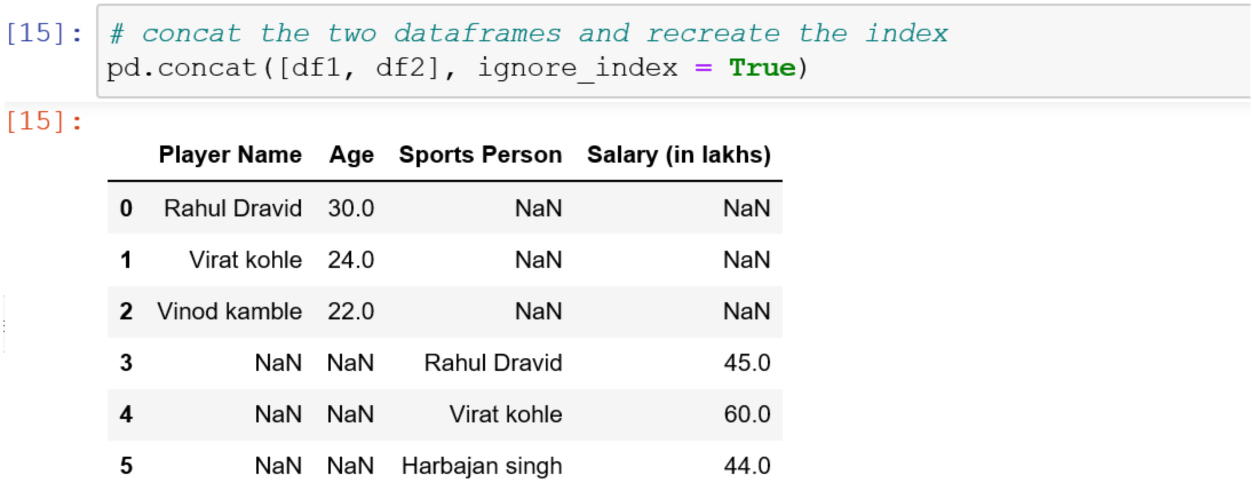
	Sports Person	Salary (in lakhs)
0	Rahul Dravid	45
1	Virat kohle	60
2	Harbajan singh	44



```
[14]: # concat the two dataframes
      pd.concat([df1, df2])
```

[14]:

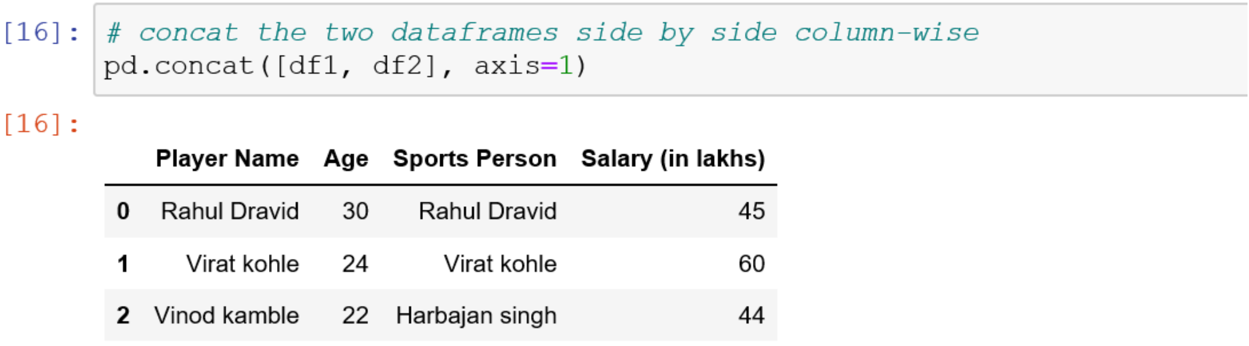
	Player Name	Age	Sports Person	Salary (in lakhs)
0	Rahul Dravid	30.0	NaN	NaN
1	Virat kohle	24.0	NaN	NaN
2	Vinod kamble	22.0	NaN	NaN
0	NaN	NaN	Rahul Dravid	45.0
1	NaN	NaN	Virat kohle	60.0
2	NaN	NaN	Harbajan singh	44.0



```
[15]: # concat the two dataframes and recreate the index
      pd.concat([df1, df2], ignore_index = True)
```

[15]:

	Player Name	Age	Sports Person	Salary (in lakhs)
0	Rahul Dravid	30.0	NaN	NaN
1	Virat kohle	24.0	NaN	NaN
2	Vinod kamble	22.0	NaN	NaN
3	NaN	NaN	Rahul Dravid	45.0
4	NaN	NaN	Virat kohle	60.0
5	NaN	NaN	Harbajan singh	44.0



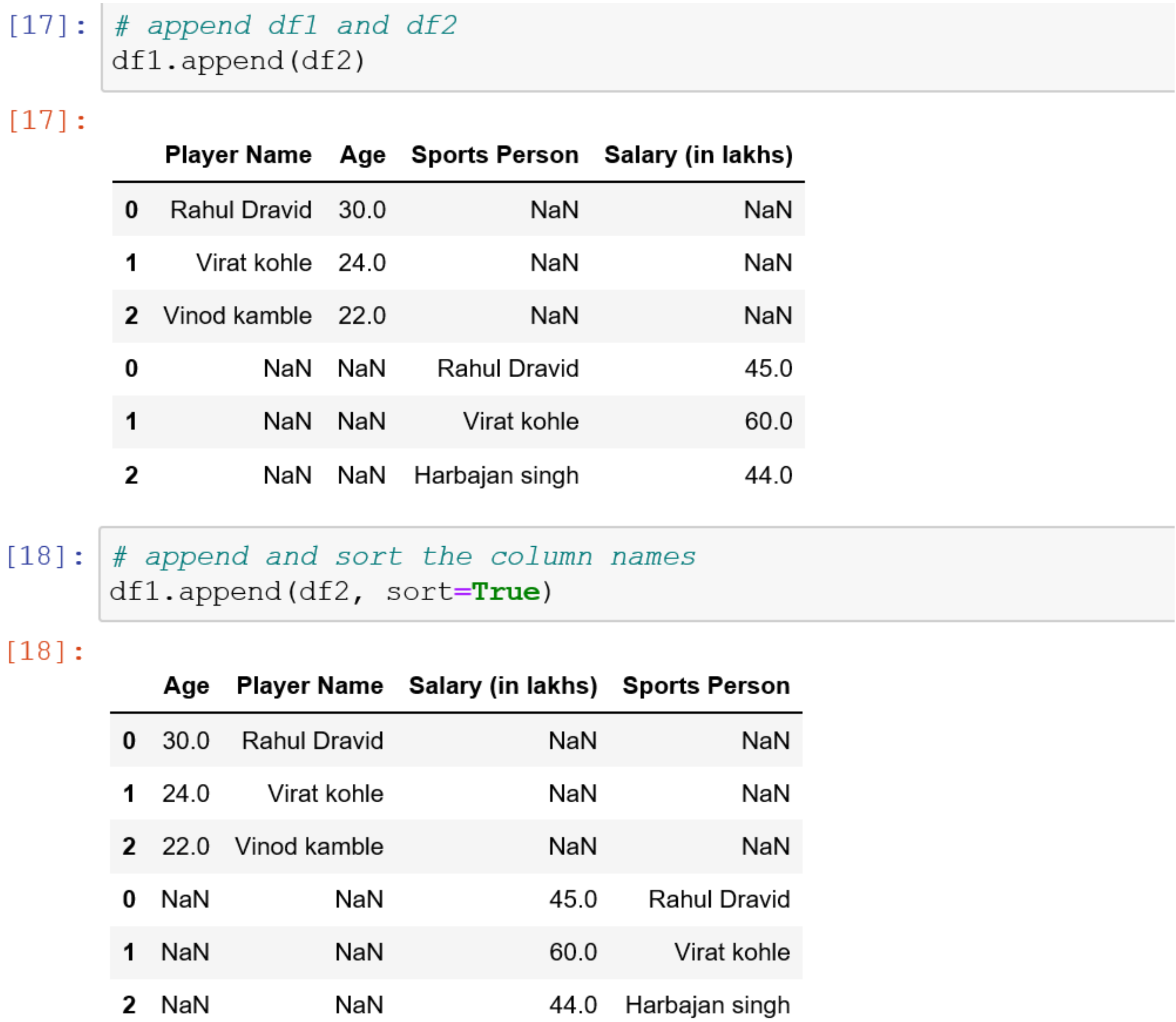
```
[16]: # concat the two dataframes side by side column-wise
      pd.concat([df1, df2], axis=1)
```

[16]:

	Player Name	Age	Sports Person	Salary (in lakhs)
0	Rahul Dravid	30	Rahul Dravid	45
1	Virat kohle	24	Virat kohle	60
2	Vinod kamble	22	Harbajan singh	44

NOTE: append() will also do the same.

Examples:



[17]:

```
# append df1 and df2
df1.append(df2)
```

[17]:

	Player Name	Age	Sports Person	Salary (in lakhs)
0	Rahul Dravid	30.0	NaN	NaN
1	Virat kohle	24.0	NaN	NaN
2	Vinod kamble	22.0	NaN	NaN
0	NaN	NaN	Rahul Dravid	45.0
1	NaN	NaN	Virat kohle	60.0
2	NaN	NaN	Harbajan singh	44.0

[18]:

```
# append and sort the column names
df1.append(df2, sort=True)
```

[18]:

	Age	Player Name	Salary (in lakhs)	Sports Person
0	30.0	Rahul Dravid	NaN	NaN
1	24.0	Virat kohle	NaN	NaN
2	22.0	Vinod kamble	NaN	NaN
0	NaN	NaN	45.0	Rahul Dravid
1	NaN	NaN	60.0	Virat kohle
2	NaN	NaN	44.0	Harbajan singh