



# Data Analysis with Python

## Session-9





# Combining DataFrames

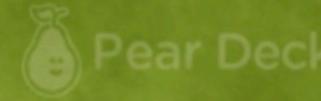


# ► Table of Contents

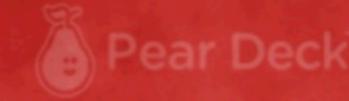
- ▶ Types of Combining
- ▶ Appending of DataFrames
- ▶ Concatenating of DataFrames
- ▶ Joining of DataFrames
- ▶ Merging of DataFrames

I've completed the pre-class content?

True



False



True

False

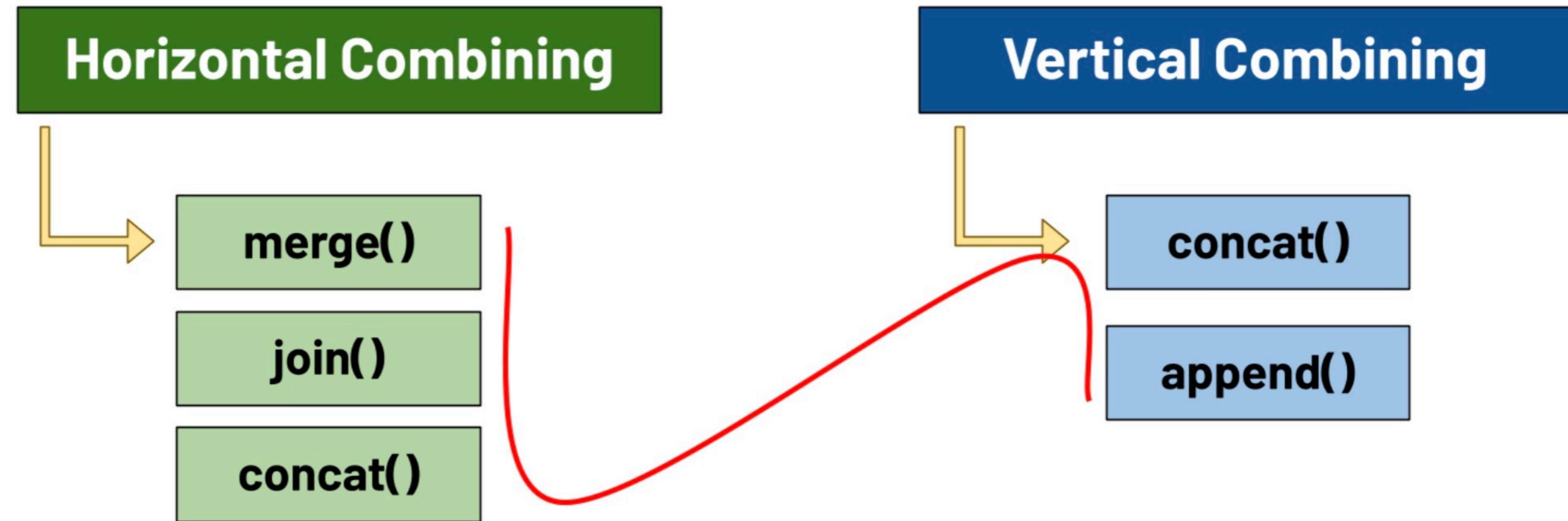


Students choose an option

Pear Deck Interactive Slide  
Do not remove this bar



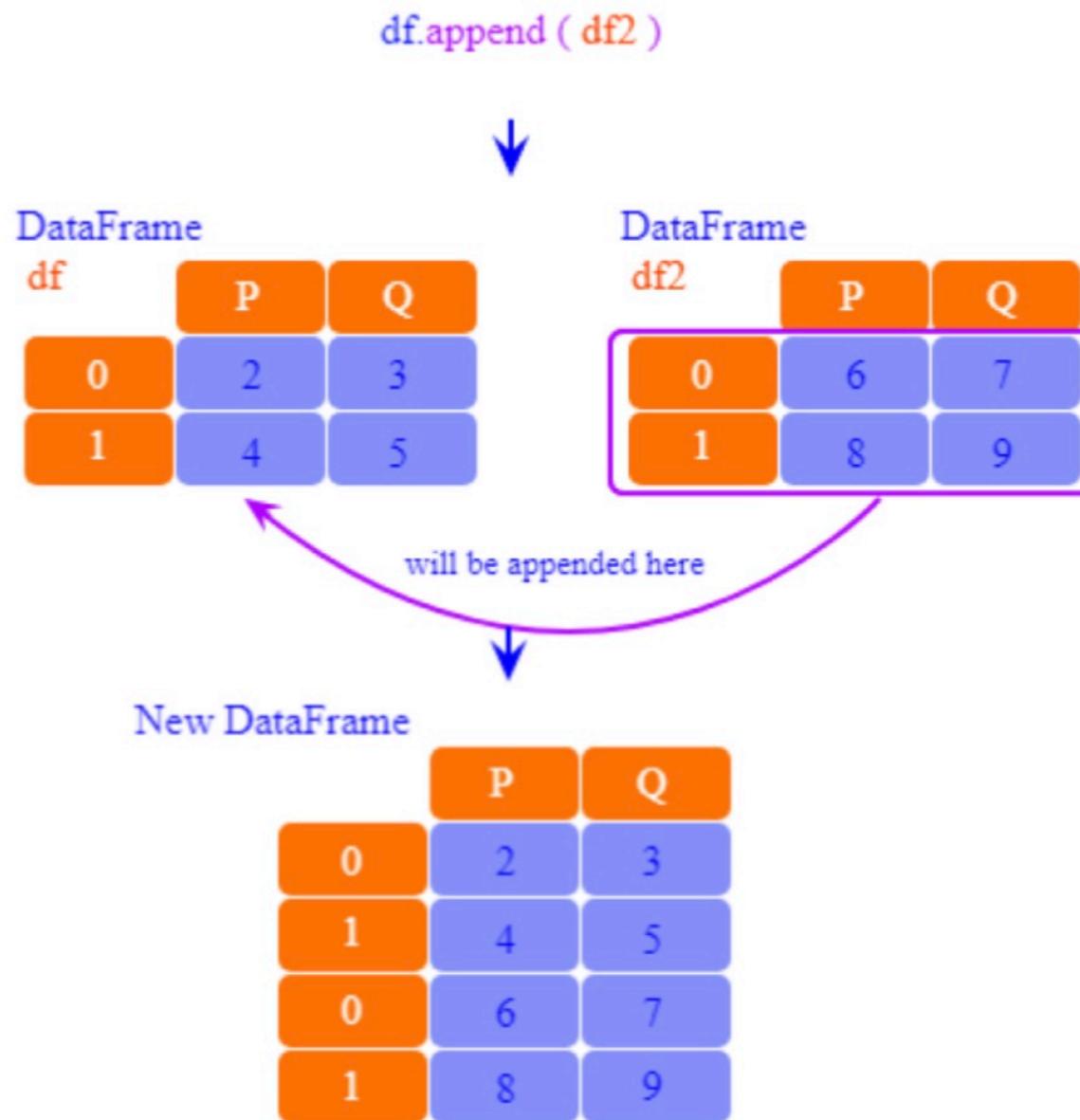
# ► Types of Combining





# Appending of DataFrame

```
DataFrame.append(other, ignore_index=False, verify_integrity=False, sort=False)
```



- If there are non-matching columns in the DataFrames, the result will contain all the columns and missing values will be set to NaN.

# ▶ Concatenating of DataFrame



```
pandas.concat(objs, axis=0, join='outer', ignore_index=False, keys=None, levels=None, names=None, verify_integrity=False, sort=False, copy=True)
```

 **Concat Pandas DataFrame**

student_df1			
index	Roll No	Name	Class
0	101	Ankit	AI
1	102	John	ML
2	103	Ali	DS
3	104	Ram	Python

student_df2			
index	Roll No	Name	Class
0	104	Juhi	AI
1	105	Malika	ML
2	106	Josef	DS
3	107	Akansha	Python

→ { student\_df1 } + { student\_df2 } → { student\_df1 + student\_df2 }

student_df1 + student_df2			
index	Roll No	Name	Class
0	101	Ankit	AI
1	102	John	ML
2	103	Ali	DS
3	104	Ram	Python
0	104	Juhi	AI
1	105	Malika	ML
2	106	Josef	DS
3	107	Akansha	Python

[www.PandasTutorial.net](http://www.PandasTutorial.net)

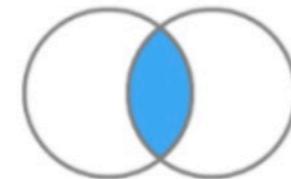
# ▶ Concatenating of DataFrame



	A	B
0		
1		

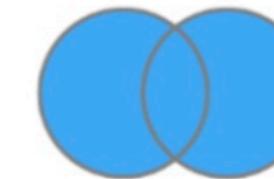
	A	C
0		
2		

inner join



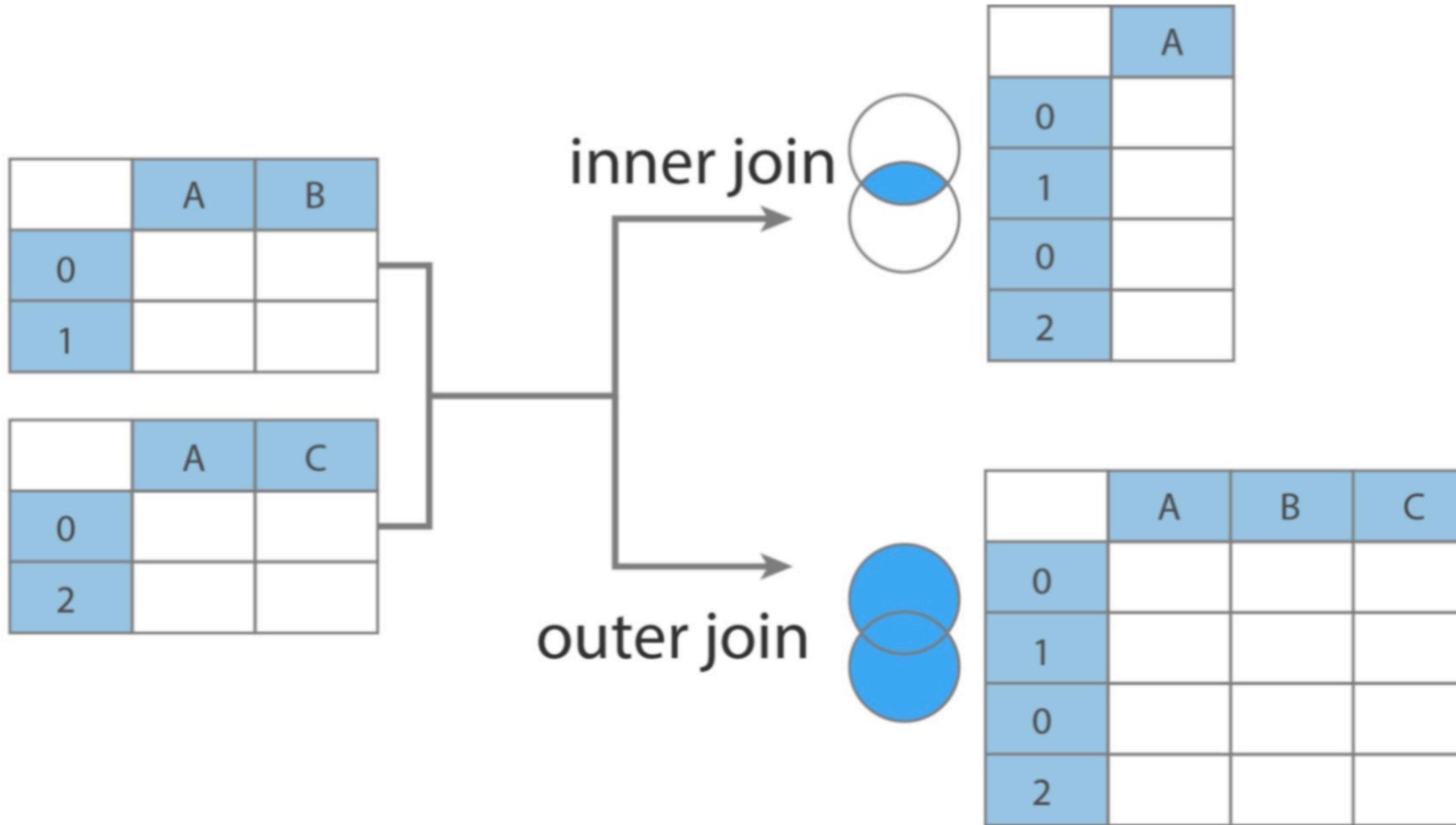
	A	B	A	C
0				

outer join



	A	B	A	C
0				
1				
2				

# ▶ Concatenating of DataFrame





# ► Concatenating of DataFrame

	orange	apple	grapes
0	3	0	7
1	2	3	14
2	0	7	6
3	1	2	15

	grapes	mango	banana	pear	pineapple
0	13	10	20	21	30
1	12	13	23	24	33
3	2	2	4	51	30
4	55	9	0	22	36
5	98	76	9	25	31

Concat with axis = 0  
is same as Append

Concat with axis = 1

	orange	apple	grapes	mango	banana	pear	pineapple
0	3.0	0.0	7	NaN	NaN	NaN	NaN
1	2.0	3.0	14	NaN	NaN	NaN	NaN
2	0.0	7.0	6	NaN	NaN	NaN	NaN
3	1.0	2.0	15	NaN	NaN	NaN	NaN
0	NaN	NaN	13	10.0	20.0	21.0	30.0
1	NaN	NaN	12	13.0	23.0	24.0	33.0
3	NaN	NaN	2	2.0	4.0	51.0	30.0
4	NaN	NaN	55	9.0	0.0	22.0	36.0
5	NaN	NaN	98	76.0	9.0	25.0	31.0

Concat  
axis = 0      Concat  
axis = 1  
Append

	orange	apple	grapes	grapes	mango	banana	pear	pineapple
0	3.0	0.0	7.0	13.0	10.0	20.0	21.0	30.0
1	2.0	3.0	14.0	12.0	13.0	23.0	24.0	33.0
2	0.0	7.0	6.0	NaN	NaN	NaN	NaN	NaN
3	1.0	2.0	15.0	2.0	2.0	4.0	51.0	30.0
4	NaN	NaN	NaN	55.0	9.0	0.0	22.0	36.0
5	NaN	NaN	NaN	98.0	76.0	9.0	25.0	31.0



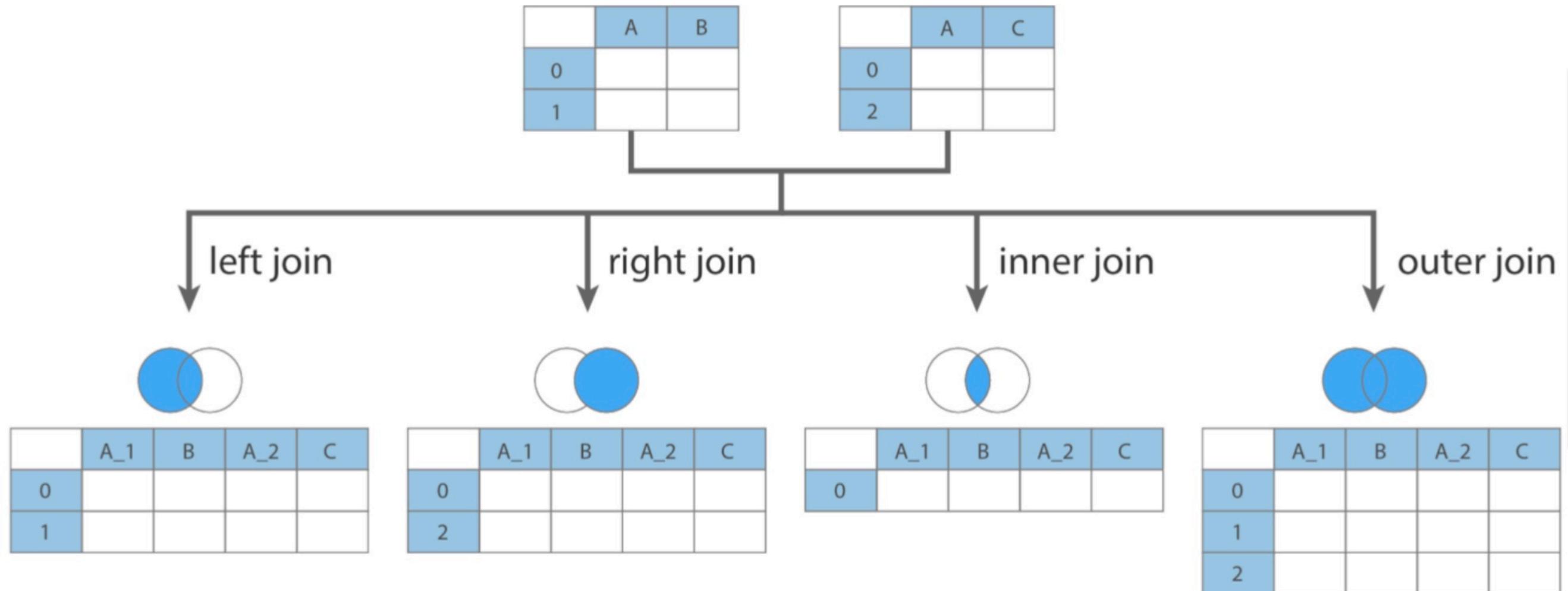
# ▶ Joining of DataFrame

```
DataFrame.join(other, on=None, how='Left', lsuffix='', rsuffix='', sort=False)
```

- ▶ It will combine all the columns from the two tables, with the common columns renamed with the defined lsuffix and rsuffix.
- ▶ The way that rows from the two tables are combined is defined by how.



# ▶ Joining of DataFrame





# ► Merging of DataFrame

```
DataFrame.merge(right, how='inner', on=None, left_on=None, right_on=None, left_index=False,  
right_index=False, sort=False, suffixes=('_x', '_y'), copy=True, indicator=False, validate=None)
```

- ▶ Similar to join, merge also combines all the columns from the two tables, with the common columns renamed with the defined suffixes.
- ▶ However, merge provides three ways of flexible control over row-wise alignment.



# ► Merging of DataFrame

```
DataFrame.merge(right, how='inner', on=None, left_on=None, right_on=None, left_index=False,  
right_index=False, sort=False, suffixes=('_x', '_y'), copy=True, indicator=False, validate=None)
```

- ▶ **The first way** is to use “**on = COLUMN NAME**”, here the given column must be the common column in both tables.
- ▶ **The second way** is to use “**left\_on = COLUMN NAME and right\_on = COLUMN NAME**”, and it allows to align the two tables using two different columns.



# ► Merging of DataFrame

```
DataFrame.merge(right, how='inner', on=None, left_on=None, right_on=None, left_index=False,  
right_index=False, sort=False, suffixes=('_x', '_y'), copy=True, indicator=False, validate=None)
```

- The third way is to use “**left\_index = True** and **right\_index = True**”, and the two tables are aligned based on their indexes.



# Merging of DataFrame

## Pandas Merge

*Join Two DataFrames Together*

```
df.merge(right=other_DF, how='how_to_join', on='common_column')
```

```
df_merged = df1.merge(df2, how='left', on='Name')
```

df1			+	df2		=	df_merged			
Index	Name	Type		Index	Name	AvgBill	Index	Name	Type	AvgBill
0	Loho Loho	Restaurant		0	Loho Loho	\$45.32	0	Loho Loho	Restaurant	\$45.32
1	Chambers	Restaurant		1	Chambers	\$65.33	1	Chambers	Restaurant	\$65.33
2	The Square	Bar		2	The Square	\$12.45	2	The Square	Bar	\$12.45
3	Tosca Cafe	Restaurant		3	Tosca Cafe	\$180.34	3	Tosca Cafe	Restaurant	\$180.34

# Data Analysis with Python



let's start the  
hands-on phase

# Did you find this lesson interesting and challenging?



Too hard



Just right



Too easy



ISWAY<sup>©</sup>  
ENT YOURSELF

Students, drag the icon!



Pear Deck Interactive Slide  
Do not remove this bar