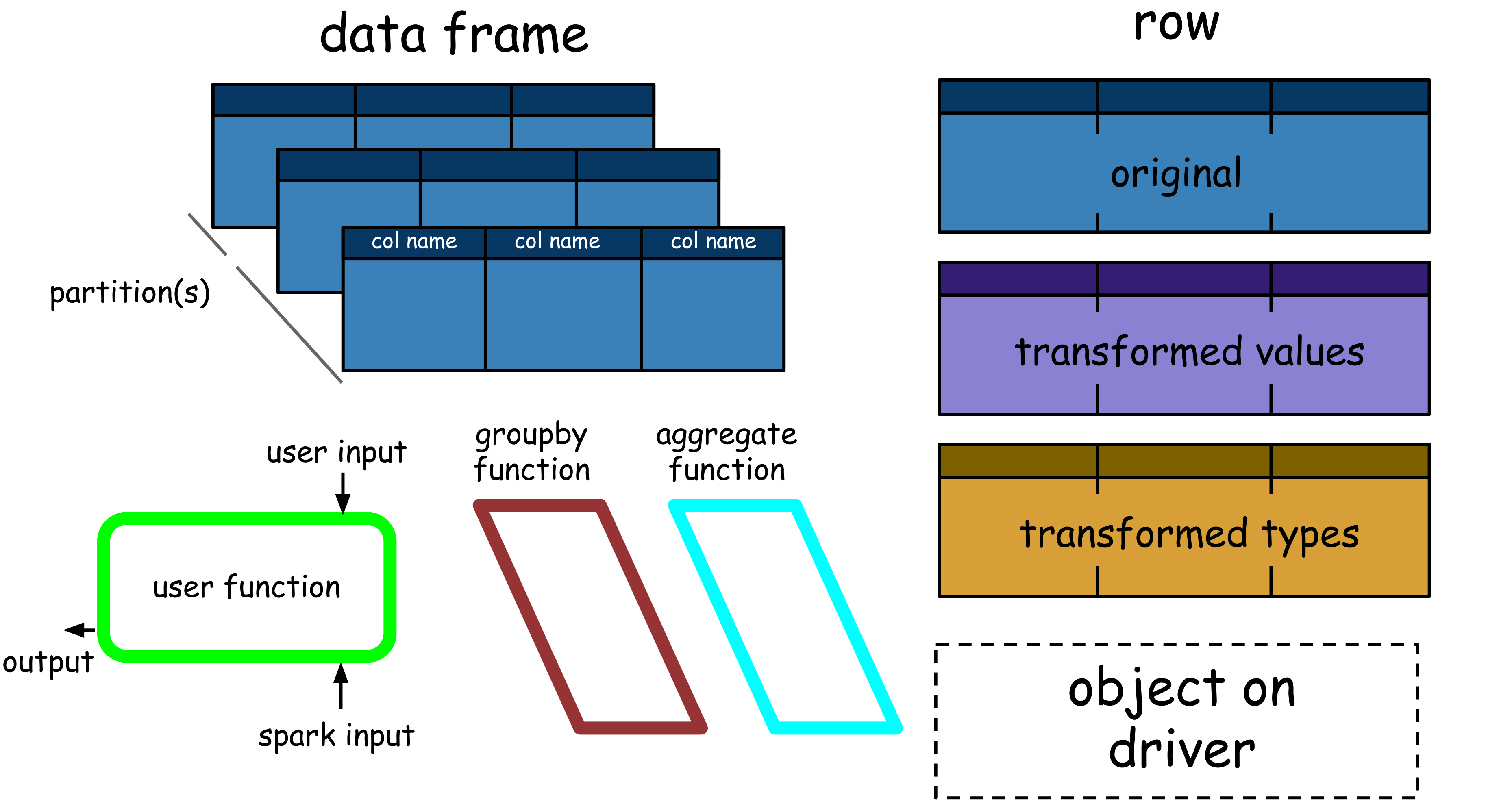
pyspark-pictures data frames

Learn the pyspark API through pictures and simple examples



data frame

col name

col name

col name

partition(s)

row

original

transformed values

transformed types

object on driver

user input

user function

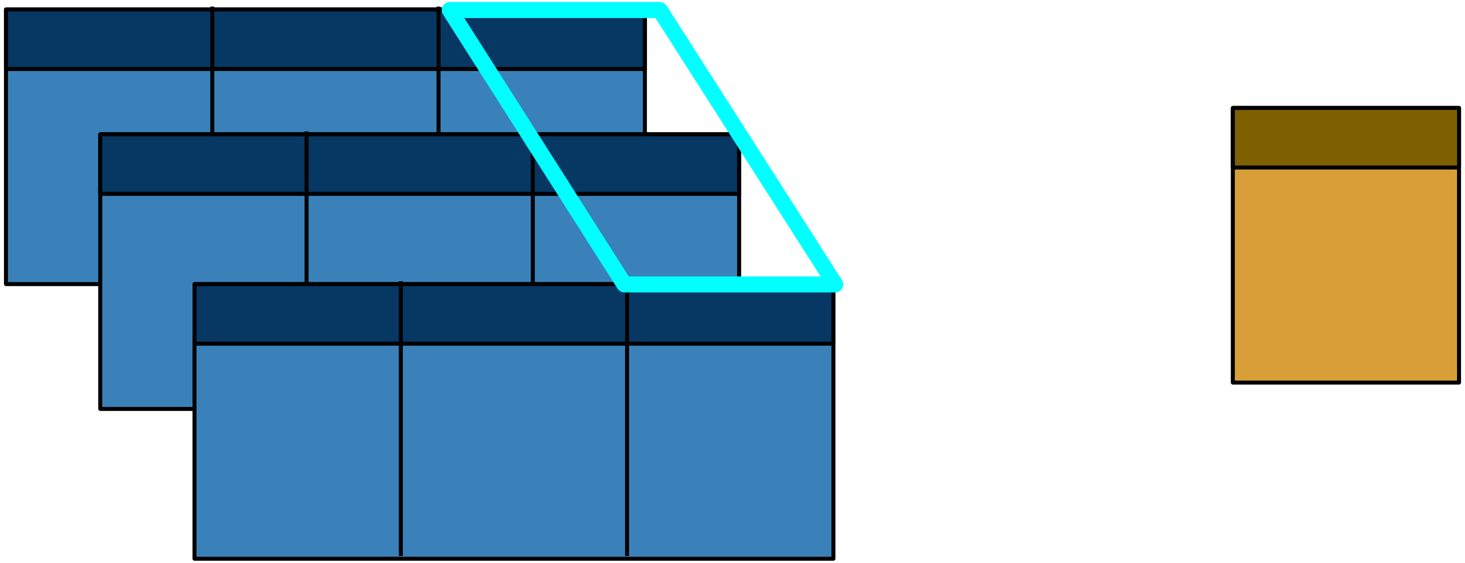
output

spark input

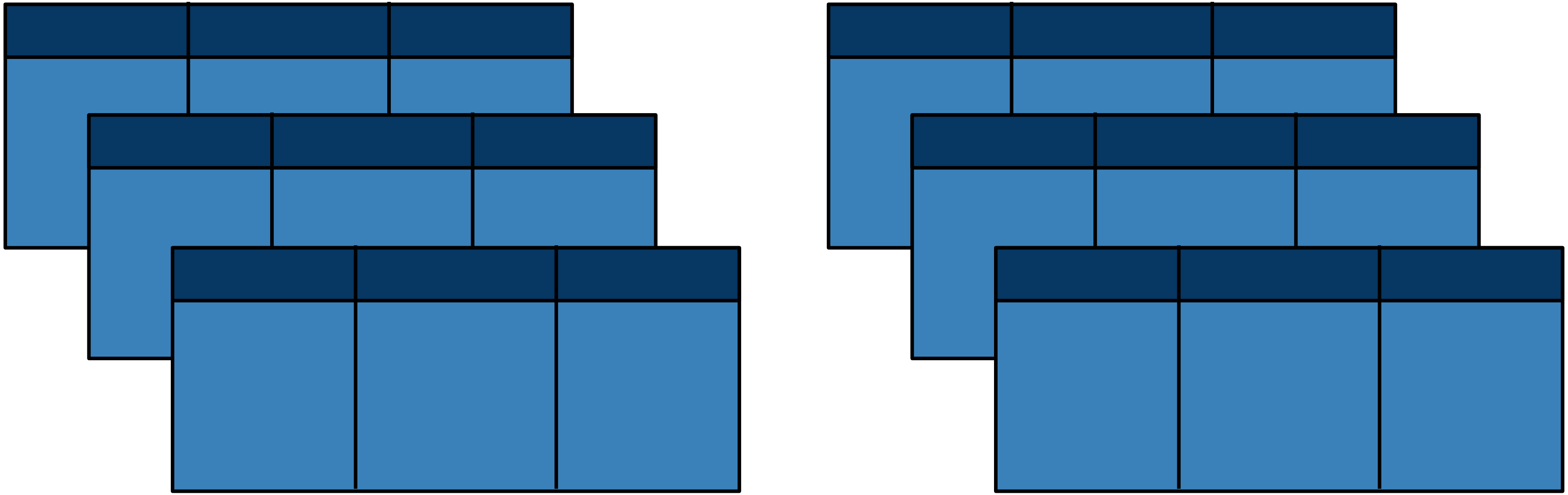
groupby function

aggregate function

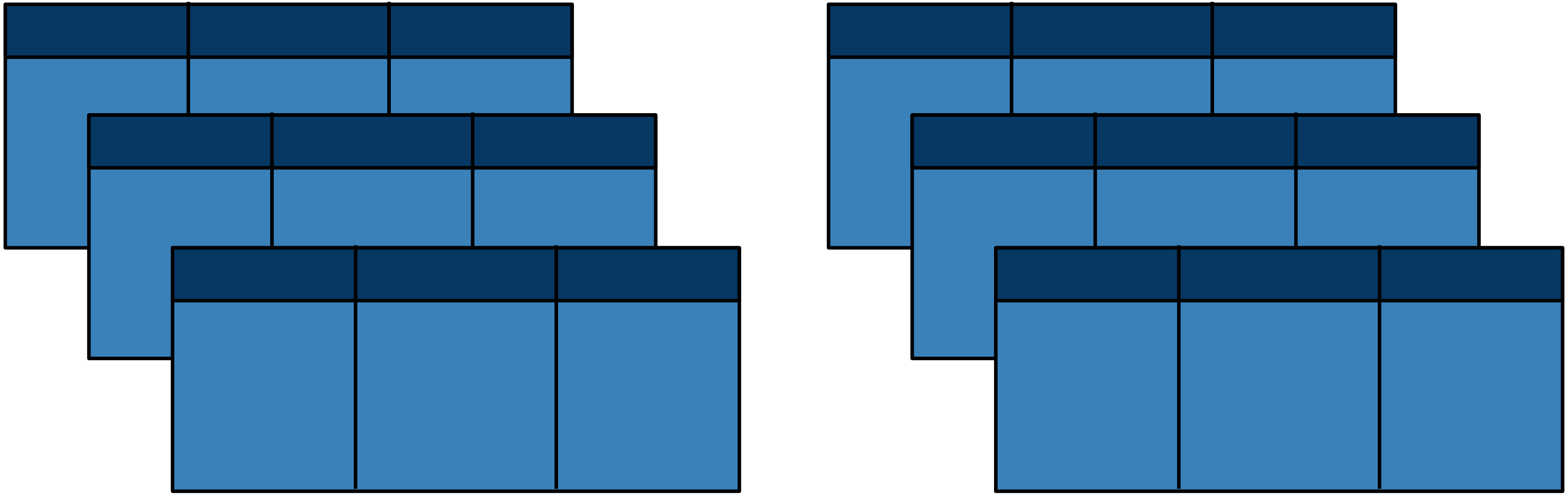
agg



alias

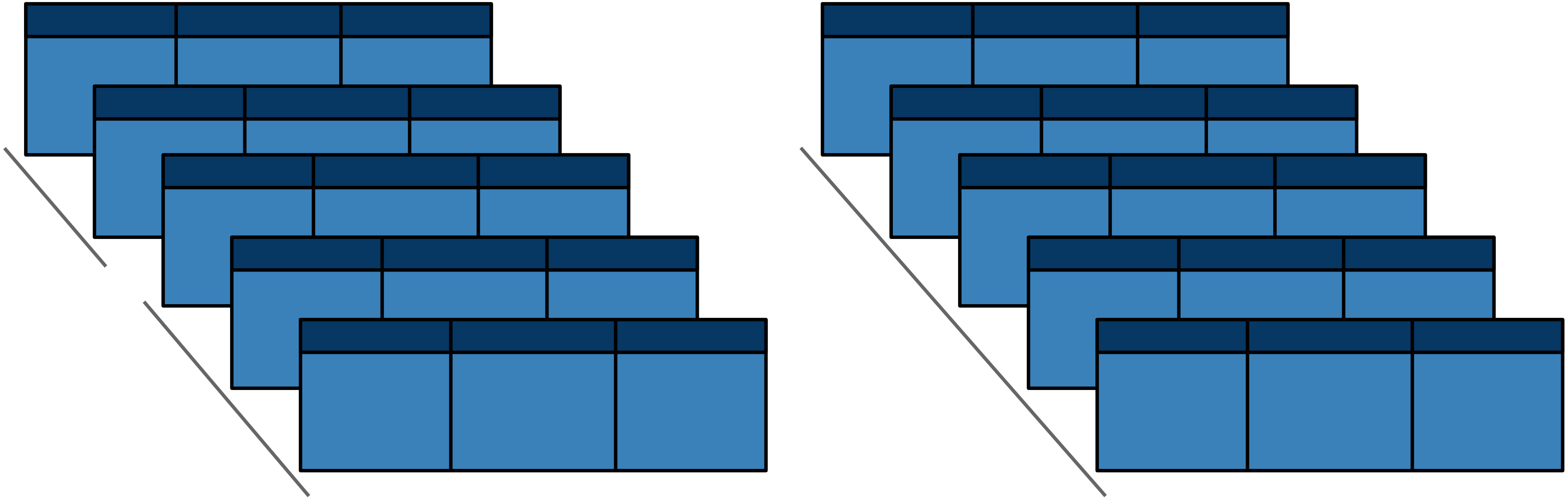


cache

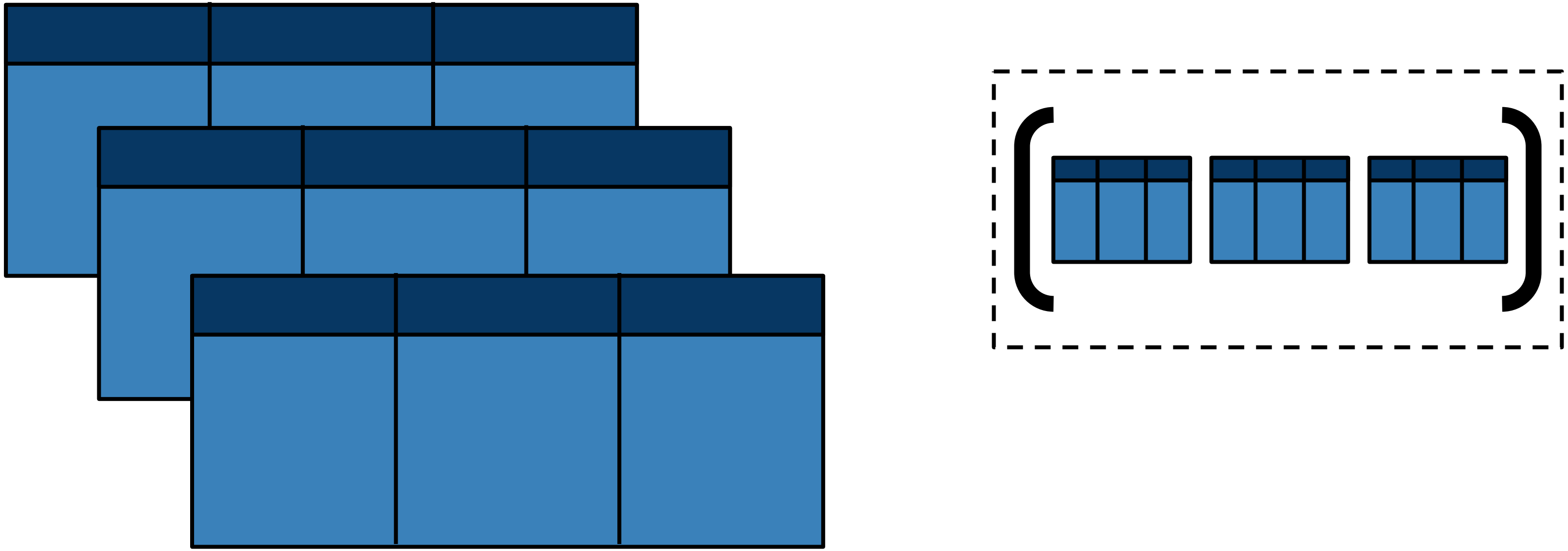


coalesce

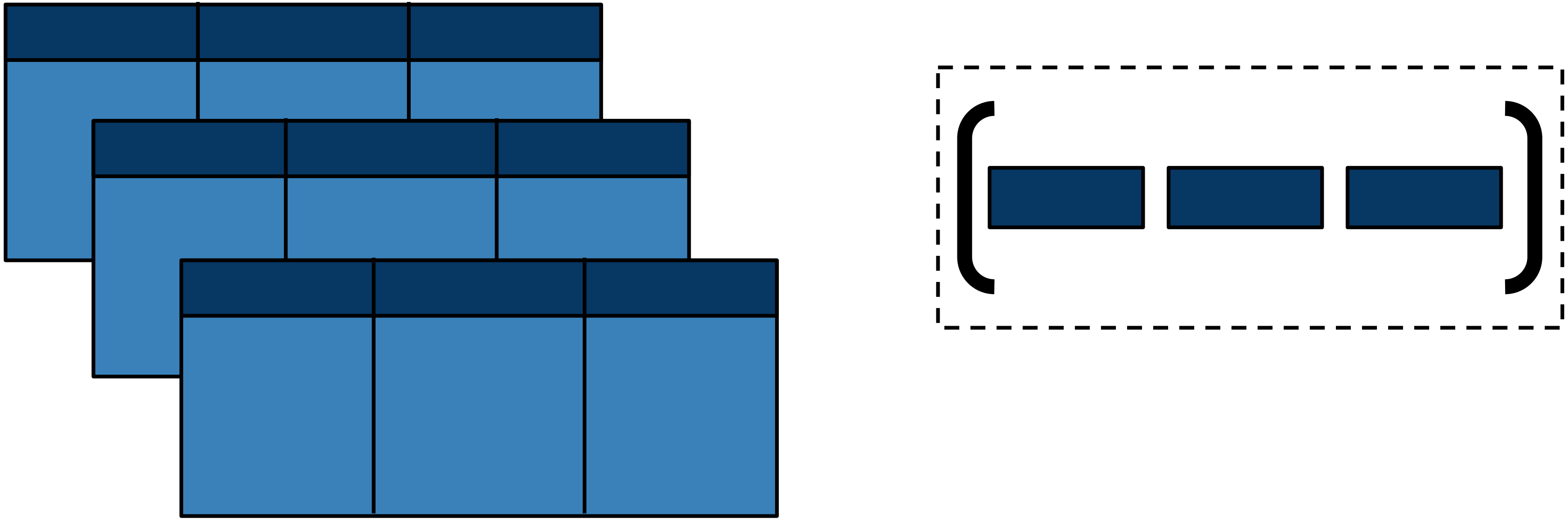
numPartitions = 1



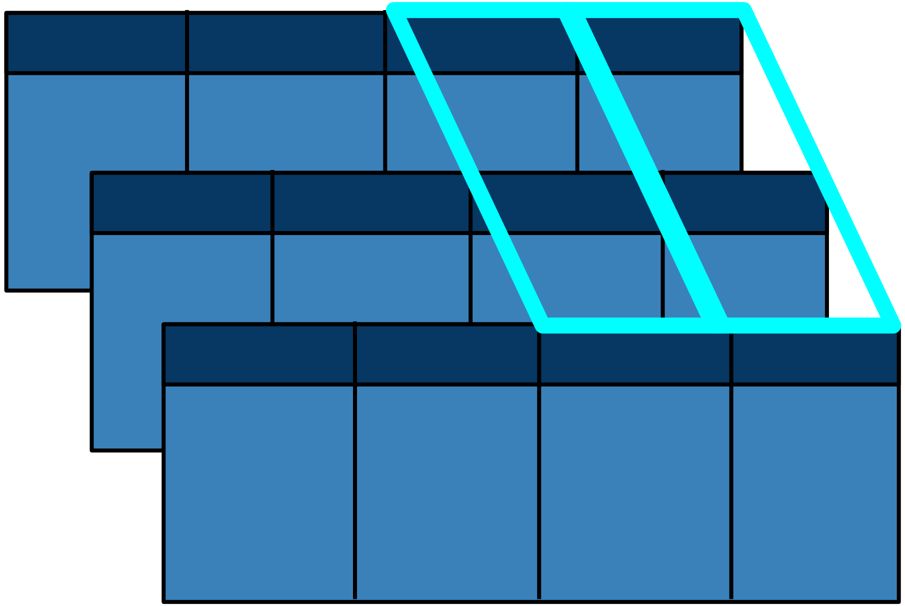
collect



columns



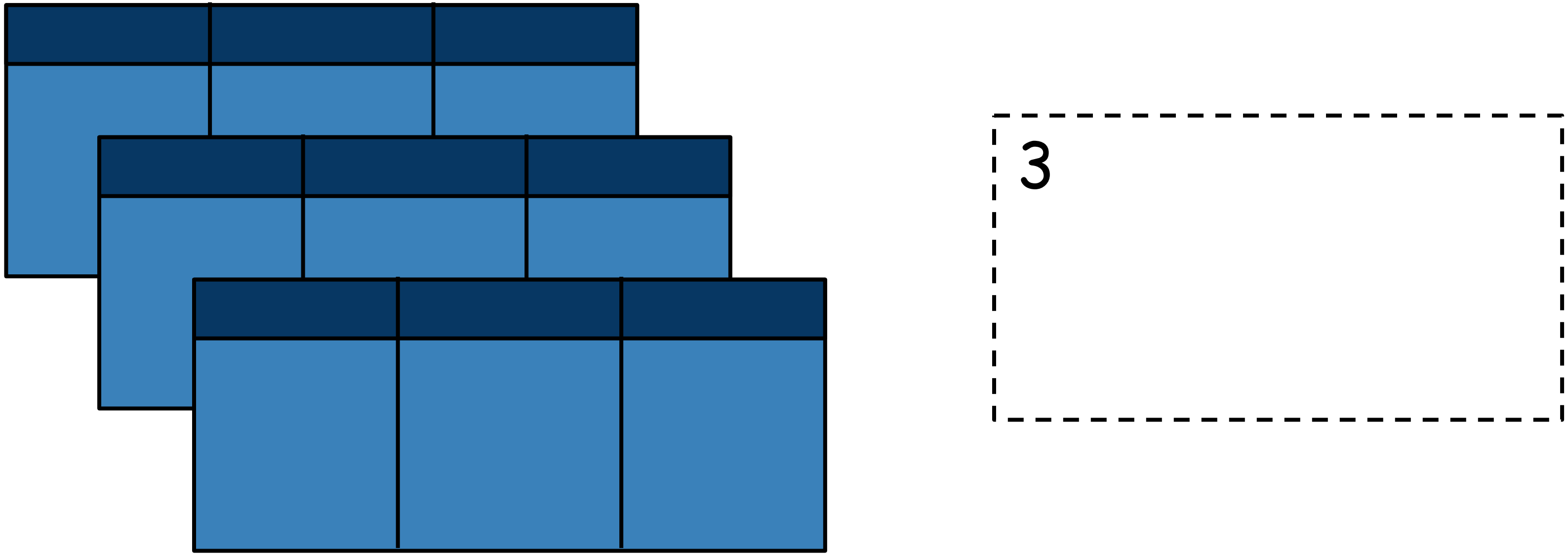
corr



Pearson's r

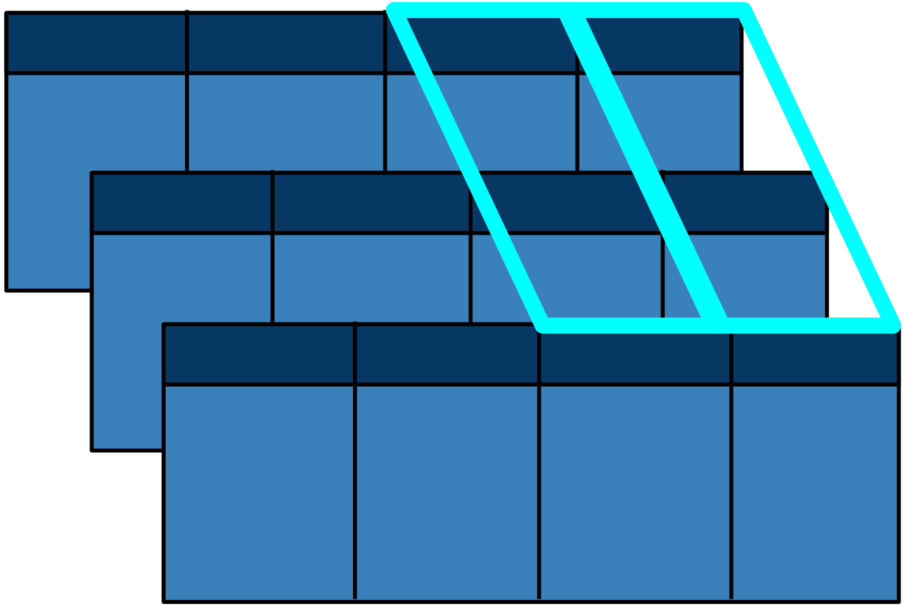
$$r = \frac{\sum_i (A_i - \bar{A})(C_i - \bar{C})}{\sqrt{\sum_i (A_i - \bar{A})^2}\sqrt{\sum_i (C_i - \bar{C})^2}}$$

count



3

cov

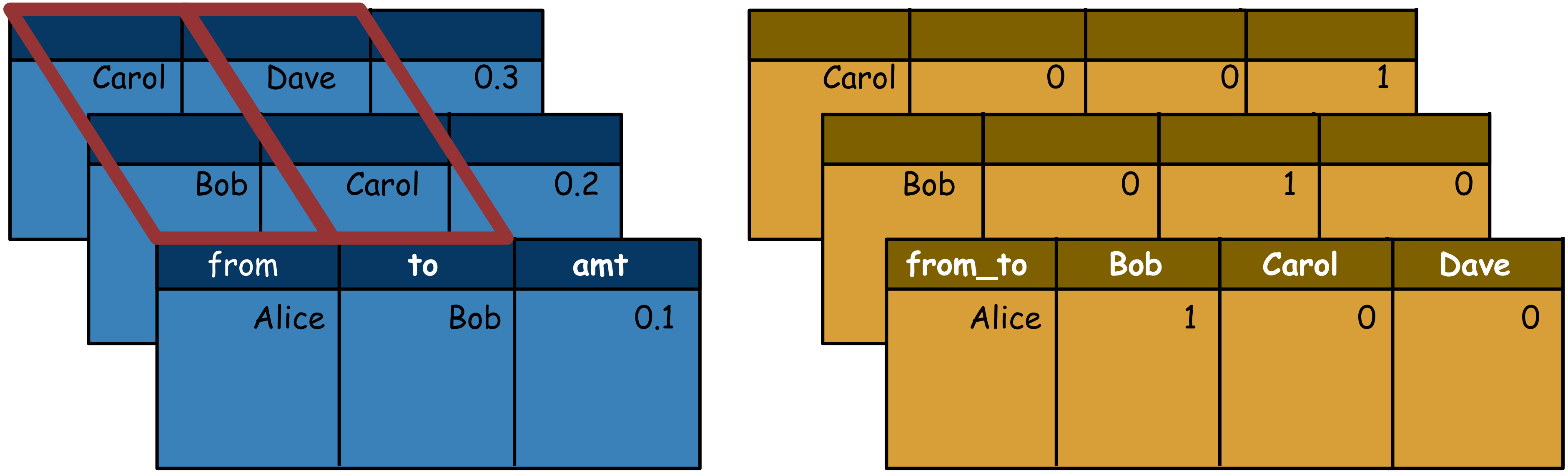


Sample Covariance

$$\frac{1}{N-1}\sum_i (A_i - \bar{A})(C_i - \bar{C})$$

crosstab

col1 = ‘from’   col2 = ‘to’

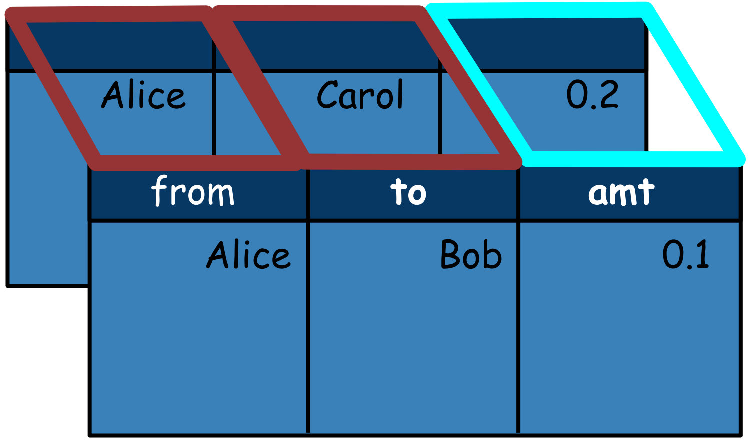


from	to	amt
Alice	Bob	0.1
Bob	Carol	0.2
Carol	Dave	0.3

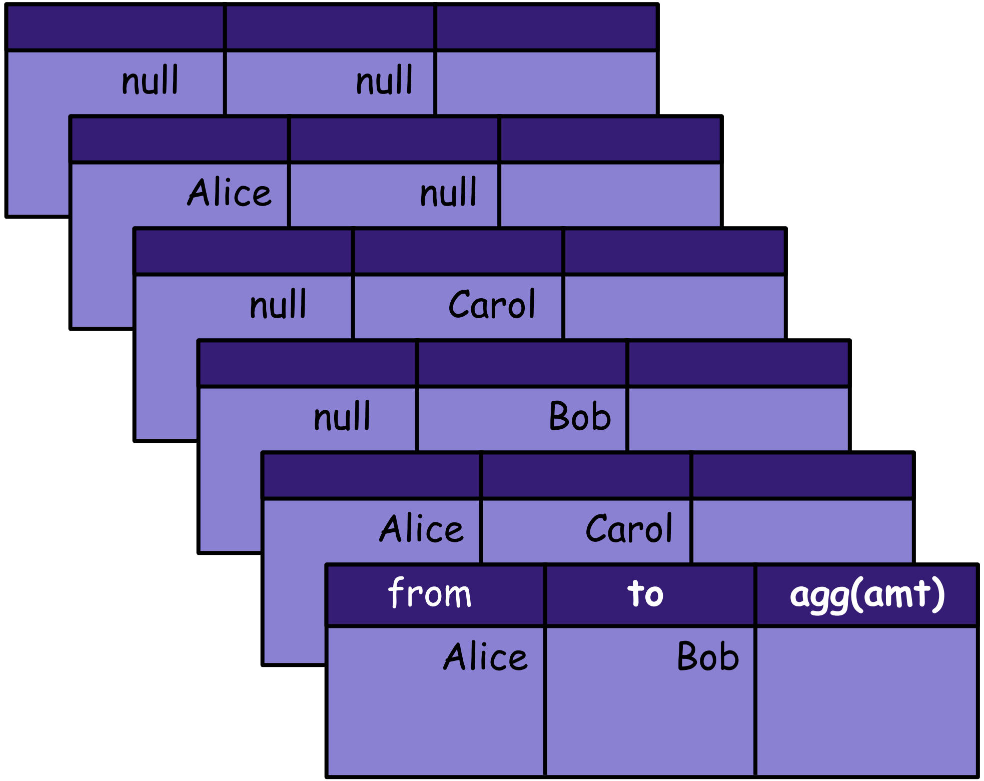
from_to	Bob	Carol	Dave
Alice	1	0	0
Bob	0	1	0
Carol	0	0	1

cube

*cols = 'from', 'to'

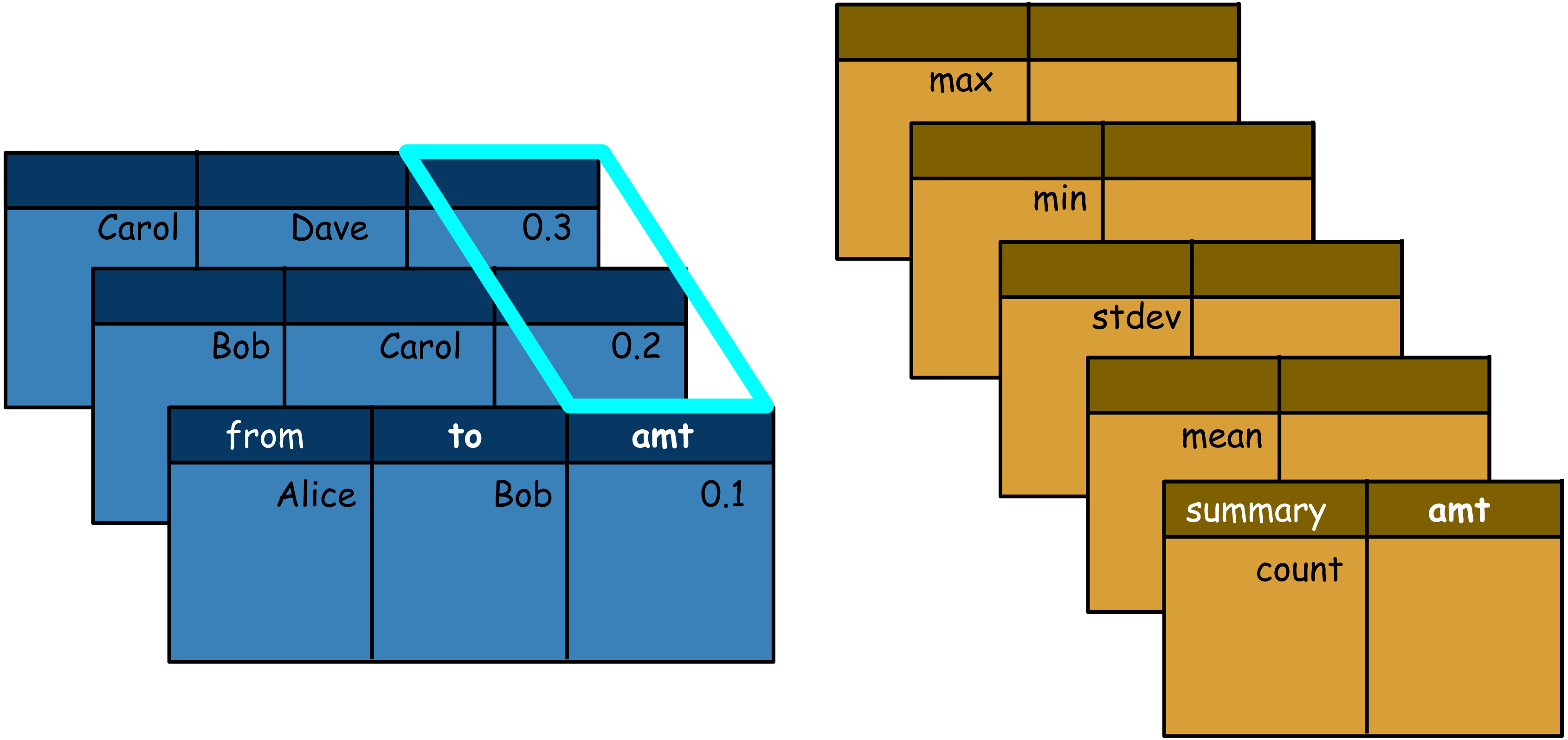


from	to	amt
Alice	Carol	0.2
Alice	Bob	0.1

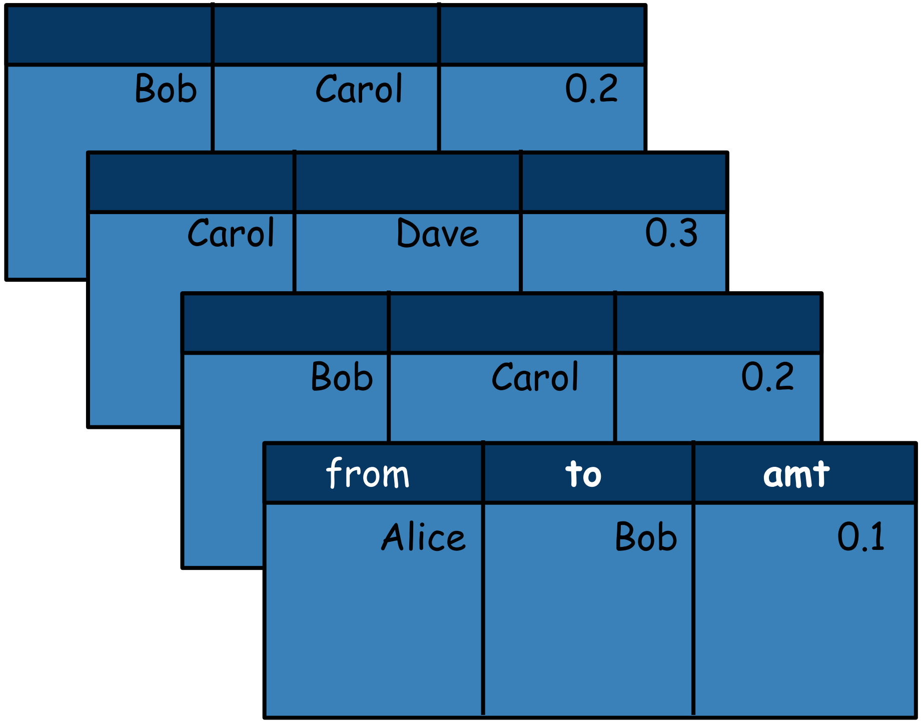


from	to	agg(amt)
null	null	
Alice	null	
null	Carol	
null	Bob	
Alice	Carol	
Alice	Bob	

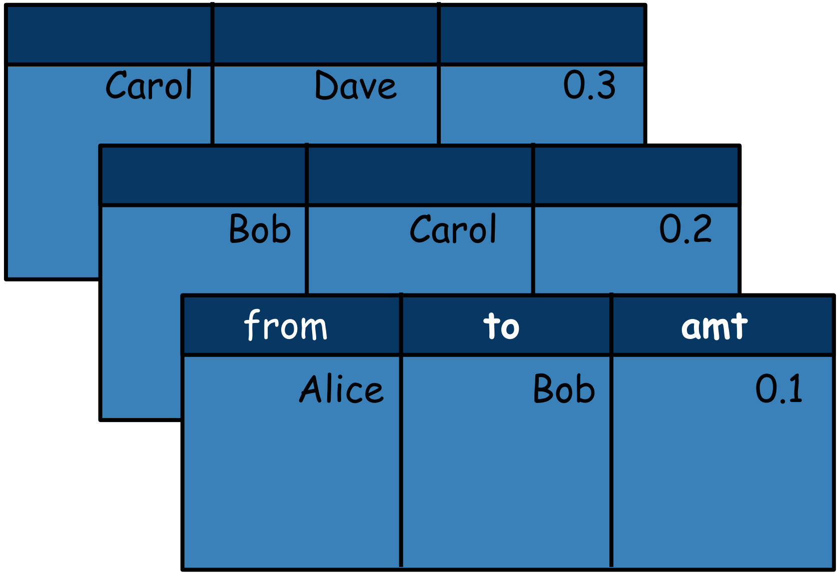
describe


Carol
Dave
0.3
Bob
Carol
0.2
from
to
amt
Alice
Bob
0.1
max
min
stdev
mean
summary
amt
count

distinct

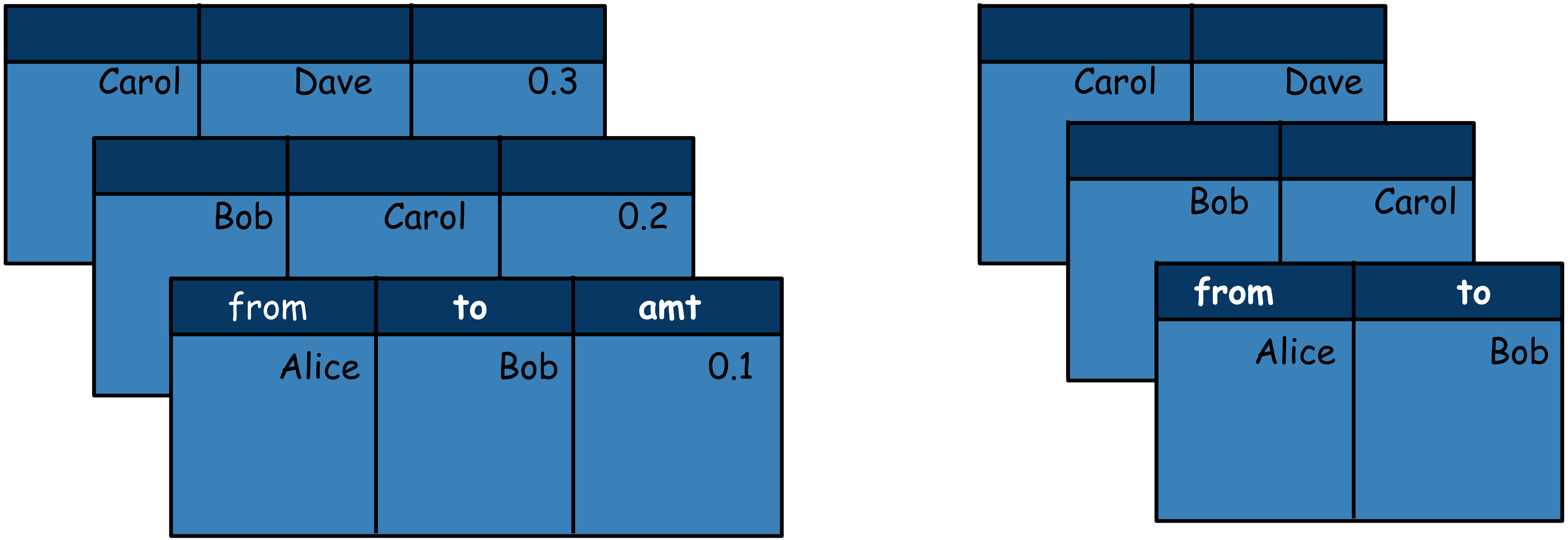


from	to	amt
Bob	Carol	0.2
Carol	Dave	0.3
Bob	Carol	0.2
Alice	Bob	0.1



from	to	amt
Carol	Dave	0.3
Bob	Carol	0.2
Alice	Bob	0.1

drop



col = 'amt'

Carol	Dave	0.3

Bob	Carol	0.2

from	to	amt
Alice	Bob	0.1

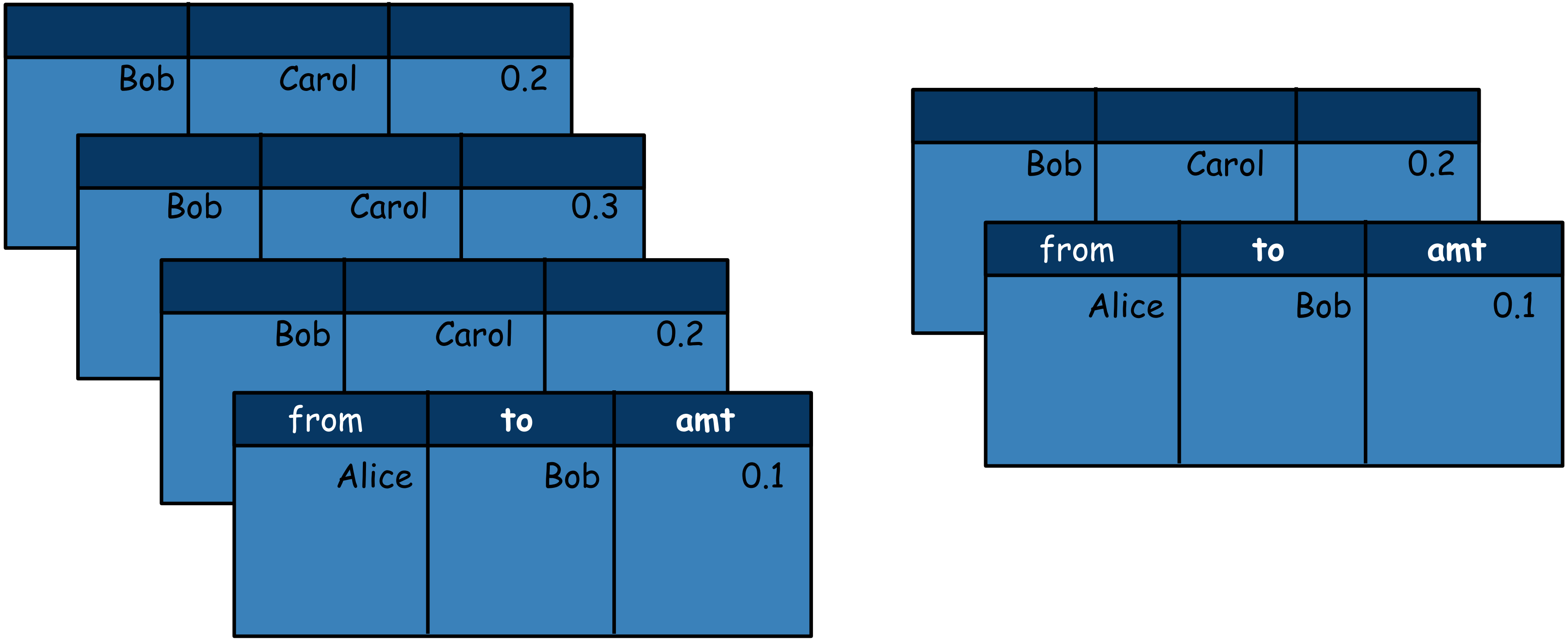
Carol	Dave

Bob	Carol

from	to
Alice	Bob

dropDuplicates

subset = ['from','to']

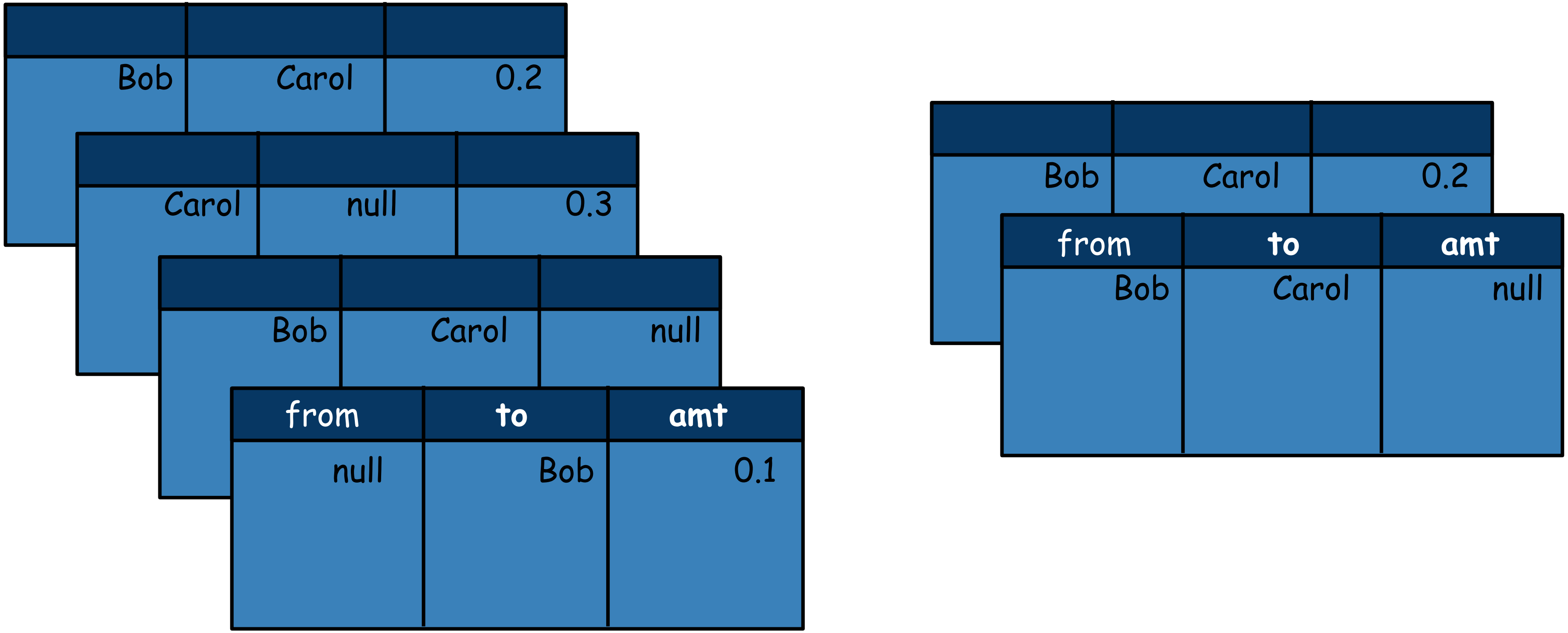


from	to	amt
Bob	Carol	0.2
Bob	Carol	0.3
Bob	Carol	0.2
Alice	Bob	0.1

from	to	amt
Bob	Carol	0.2
Alice	Bob	0.1

dropna

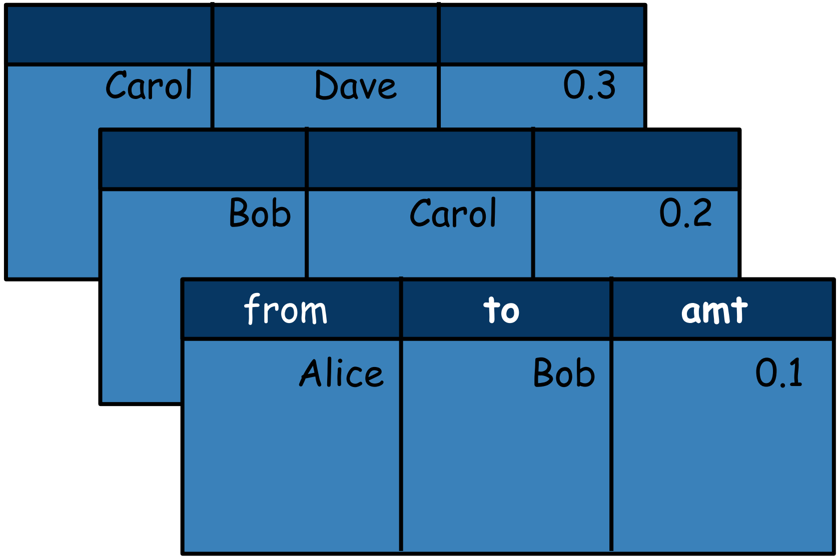
how = 'any' subset = ['from', 'to']



from	to	amt
Bob	Carol	0.2
Carol	null	0.3
Bob	Carol	null
null	Bob	0.1

from	to	amt
Bob	Carol	0.2
Bob	Carol	null

dtypes



Carol	Dave	0.3

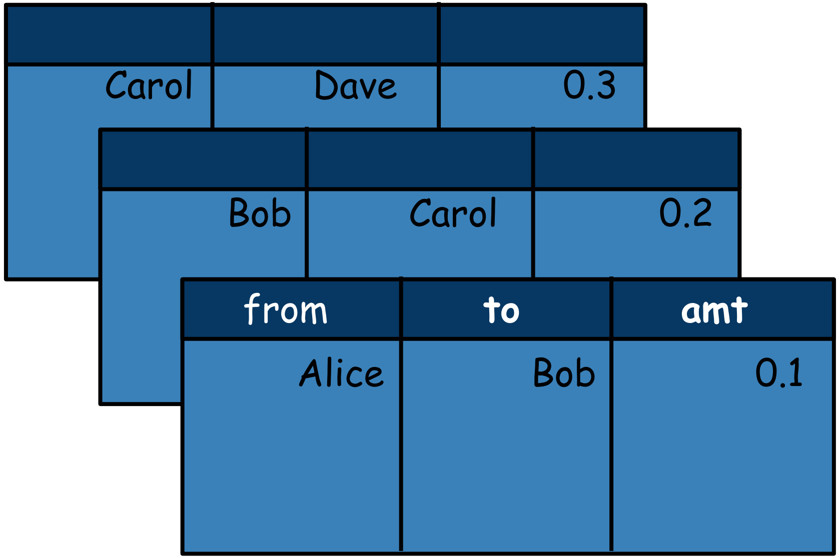
Bob	Carol	0.2

from	to	amt
Alice	Bob	0.1

[(‘from’,‘string’), (‘to’, ‘string’), (‘amt’, ‘double’)]

explain

extended = True

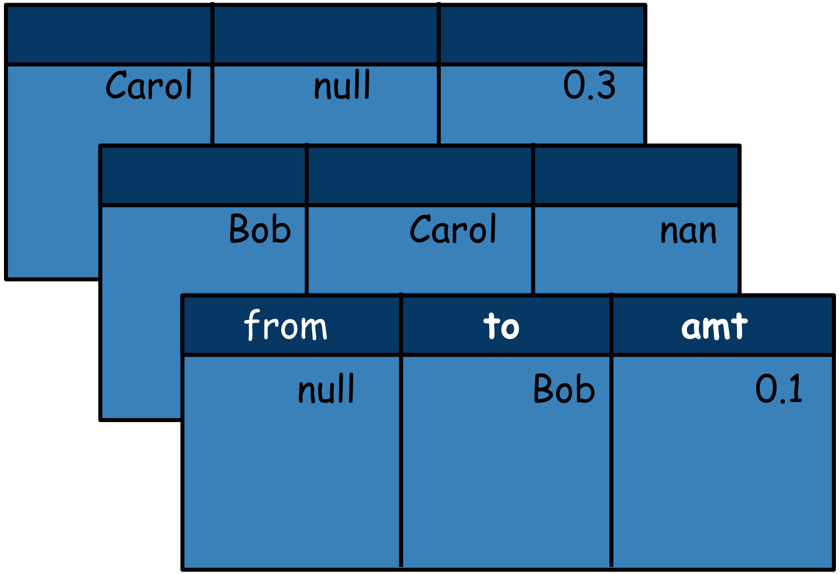


from	to	amt
Alice	Bob	0.1
Bob	Carol	0.2
Carol	Dave	0.3

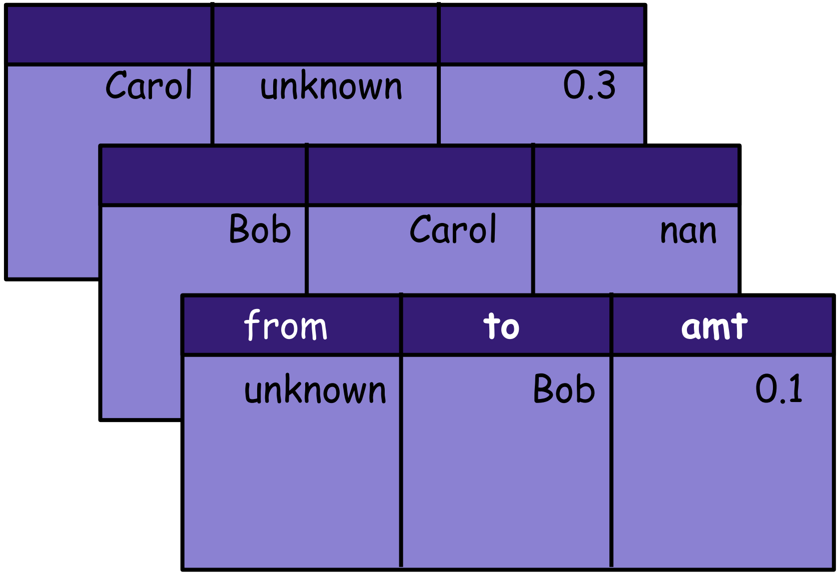
```
== Parsed Logical Plan ==
...
== Analyzed Logical Plan ==
...
== Optimized Logical Plan ==
...
== Physical Plan ==
...
== RDD ==
```

fillna

value = ‘unknown” subset = [‘from’, ‘to’]



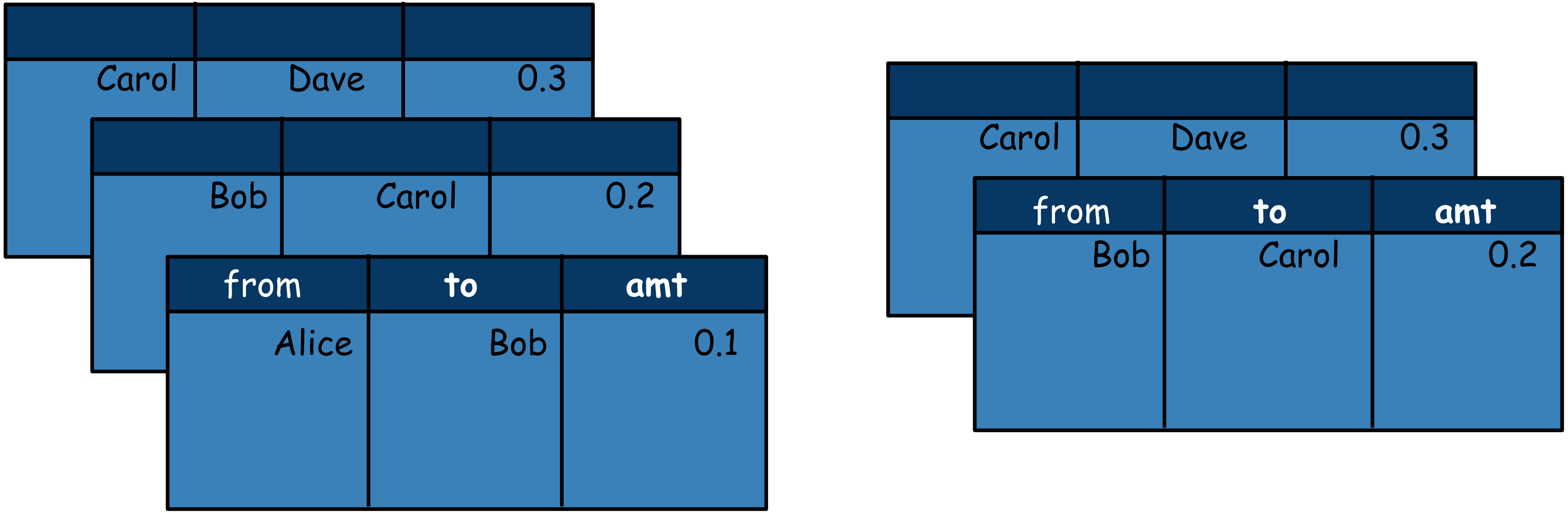
from	to	amt
null	Bob	0.1
Bob	Carol	nan
Carol	null	0.3



from	to	amt
unknown	Bob	0.1
Bob	Carol	nan
Carol	unknown	0.3

filter

condition = “amt > 0.1”



Carol	Dave	0.3

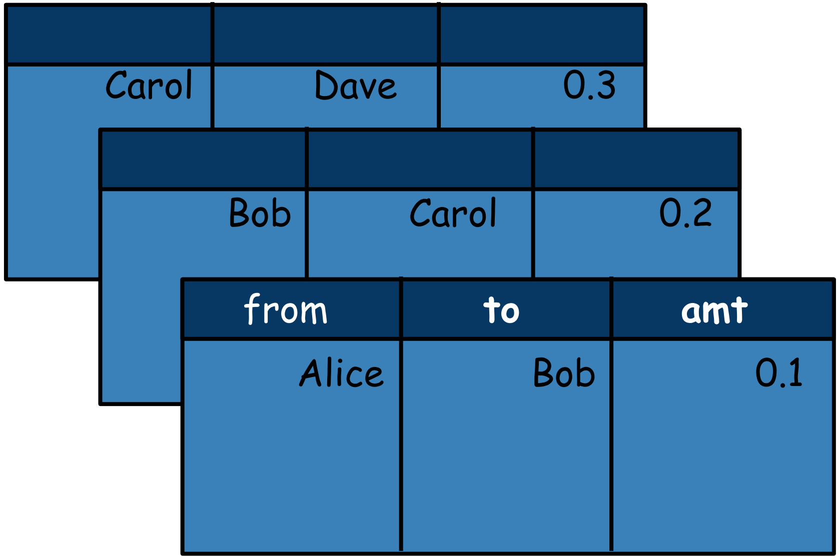
Bob	Carol	0.2

from	to	amt
Alice	Bob	0.1

Carol	Dave	0.3

from	to	amt
Bob	Carol	0.2

first



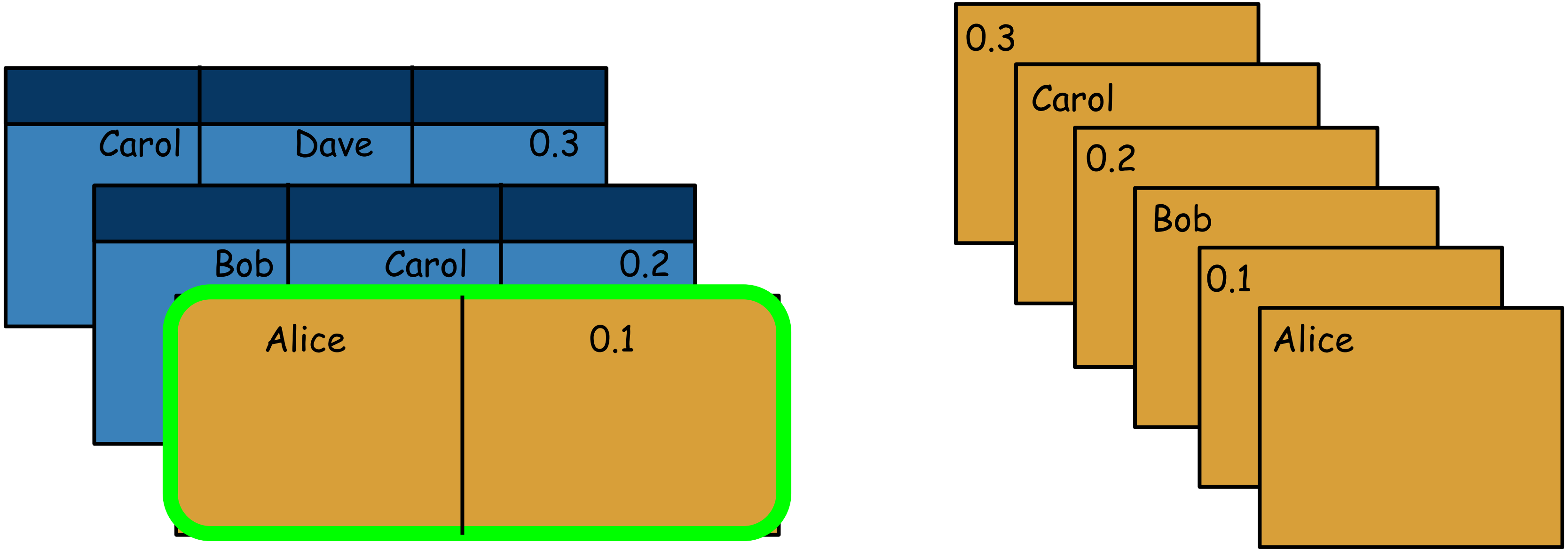
Carol	Dave	0.3

Bob	Carol	0.2

from	to	amt
Alice	Bob	0.1

```
Row(from='Alice', to='Bob', amt=0.1)
```

flatMap



Carol Dave 0.3

Bob Carol 0.2

Alice 0.1

0.3

Carol

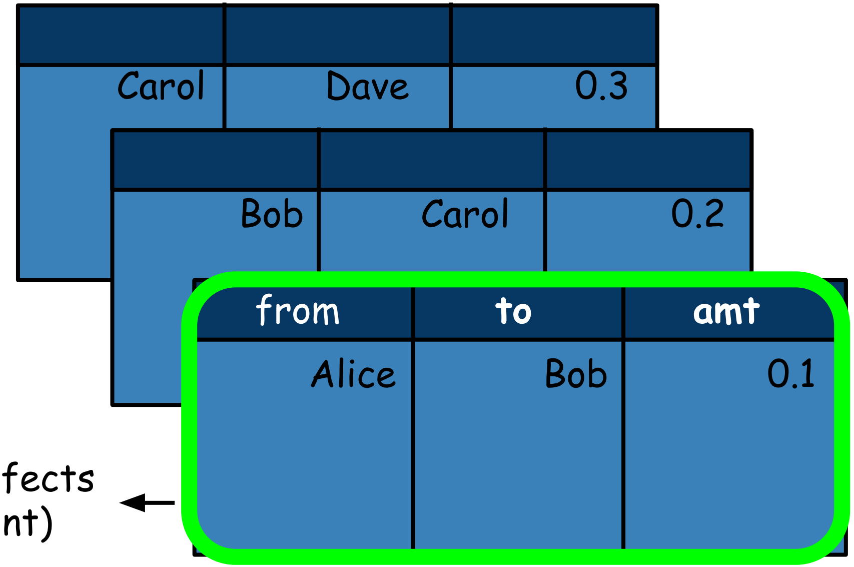
0.2

Bob

0.1

Alice

foreach

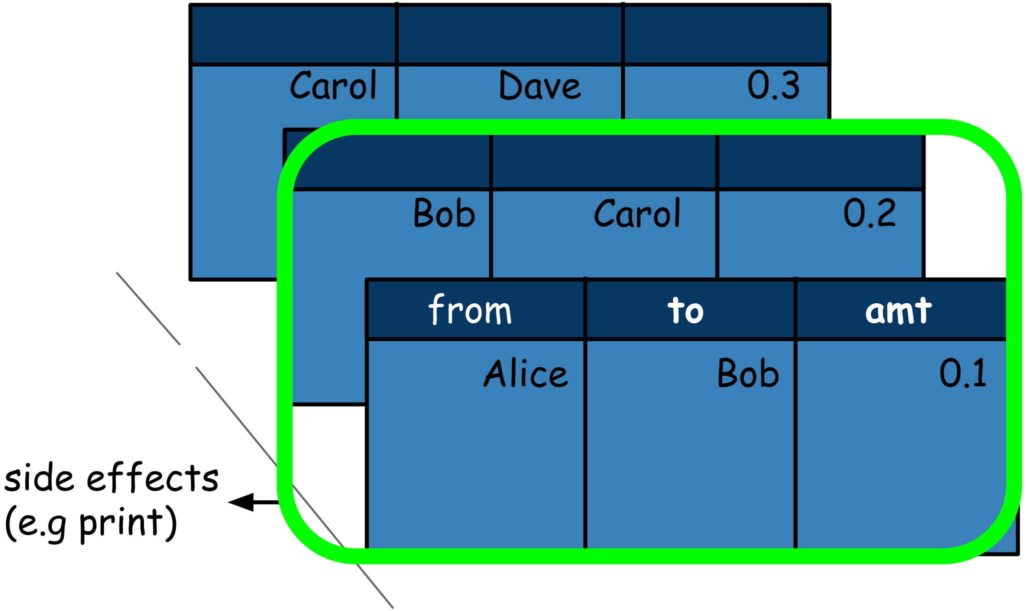


from	to	amt
Carol	Dave	0.3
Bob	Carol	0.2
Alice	Bob	0.1

side effects
(e.g print)

*no return value, original DataFrame unchanged

foreachPartition



Carol	Dave	0.3

Bob	Carol	0.2

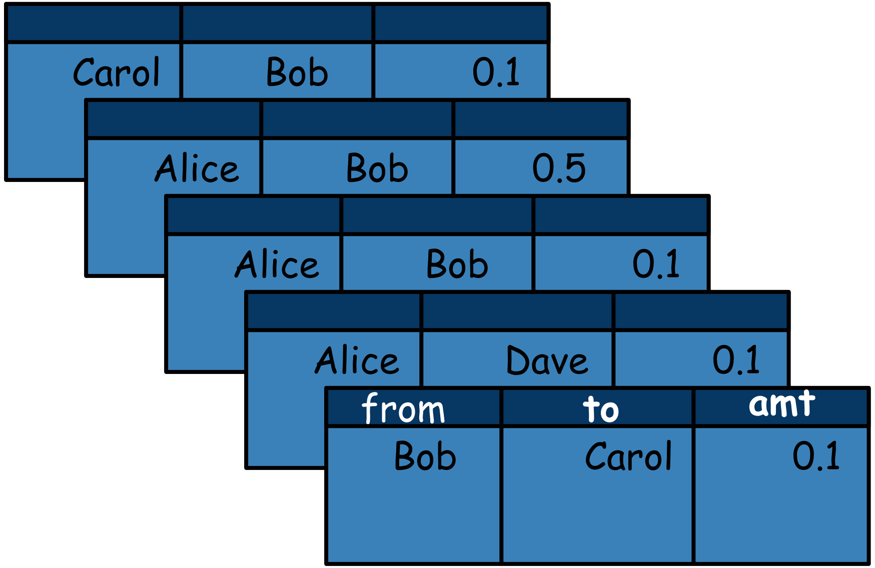
from	to	amt
Alice	Bob	0.1

side effects
(e.g print)

*no return value, original DataFrame unchanged

freqItems

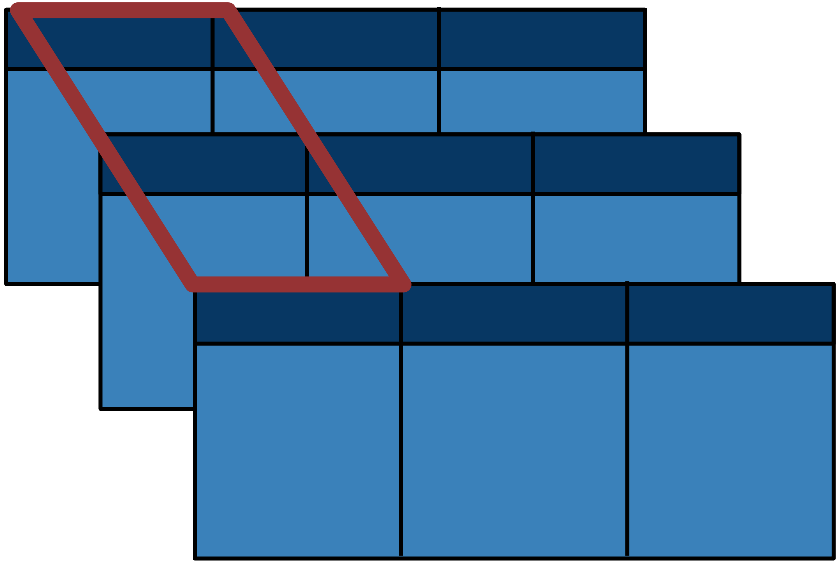
cols = ['from', 'amt'] support = 0.8



from	to	amt
Carol	Bob	0.1
Alice	Bob	0.5
Alice	Bob	0.1
Alice	Dave	0.1
Bob	Carol	0.1

from_freqItems	amt_freqItems
[Alice]	[0.1]

groupBy (groupby)

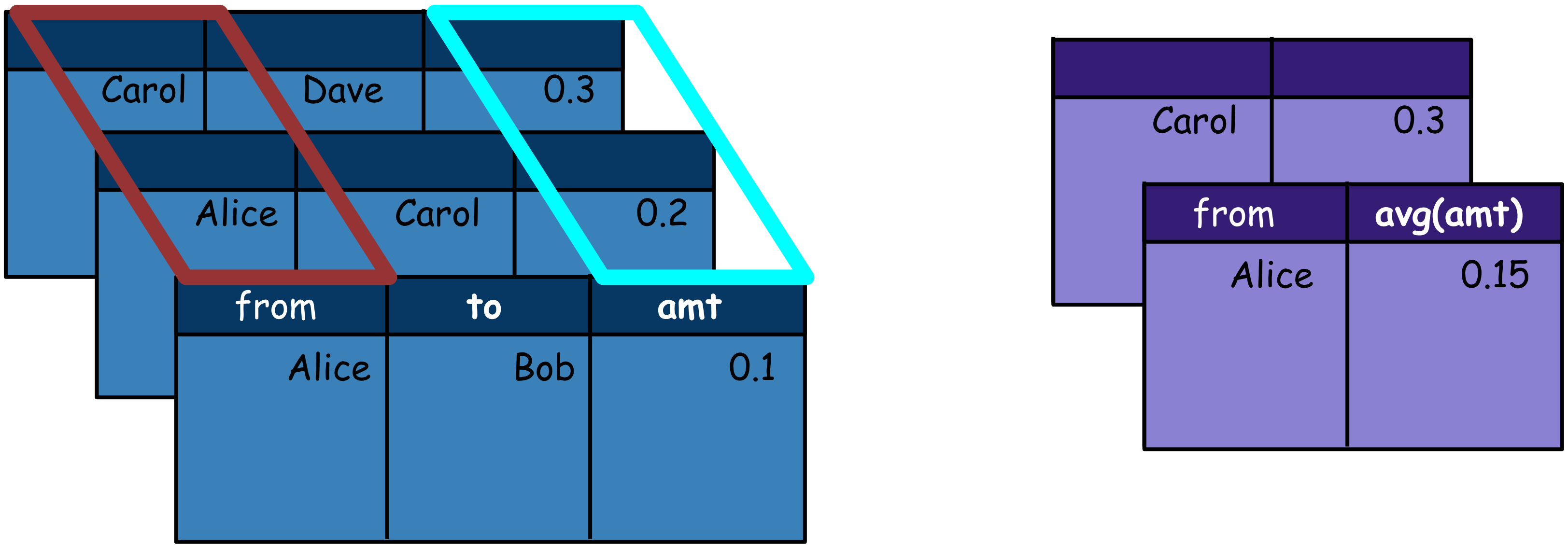


GroupedData Object

with methods: agg, avg, count, max, mean, min, pivot, sum

groupBy(col1).avg(col2)

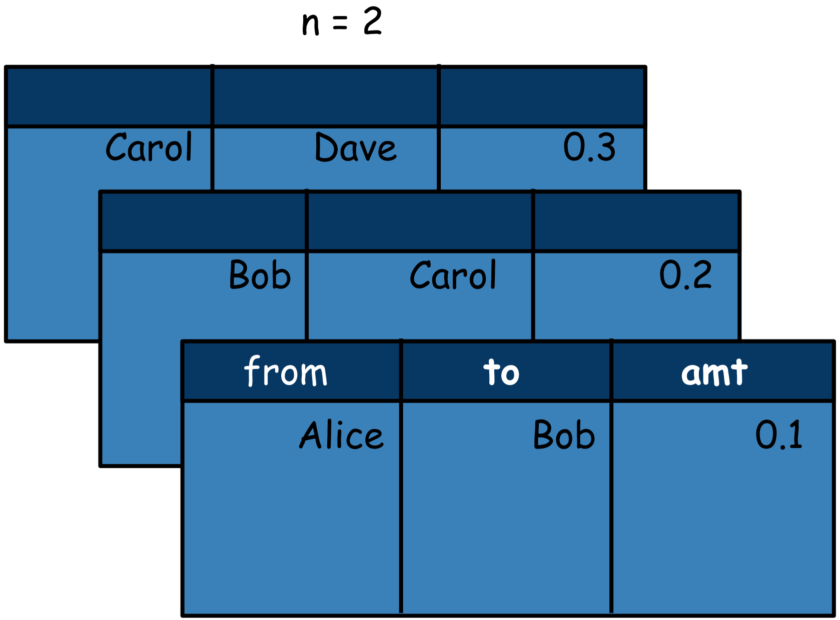
col1 = 'from' col2 = 'amt'



from	to	amt
Alice	Bob	0.1
Alice	Carol	0.2
Carol	Dave	0.3

from	avg(amt)
Alice	0.15
Carol	0.3

head



from	to	amt
Alice	Bob	0.1
Bob	Carol	0.2
Carol	Dave	0.3

```
[Row(from=u'Alice', to=u'Bob', amt=0.1),
Row(from=u'Bob', to=u'Carol', amt=0.2)]
```

intersect

from	to	amt
Alice	Bob	0.1
Bob	Carol	0.2
Carol	Dave	0.3

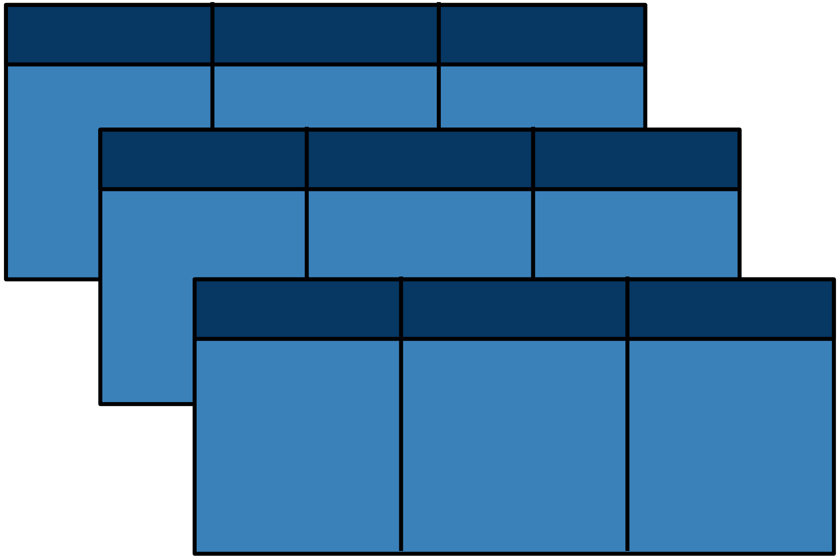
+

from	to	amt
Alice	Bob	0.1
Bob	Alice	0.2
Carol	Dave	0.1

=

from	to	amt
Alice	Bob	0.1

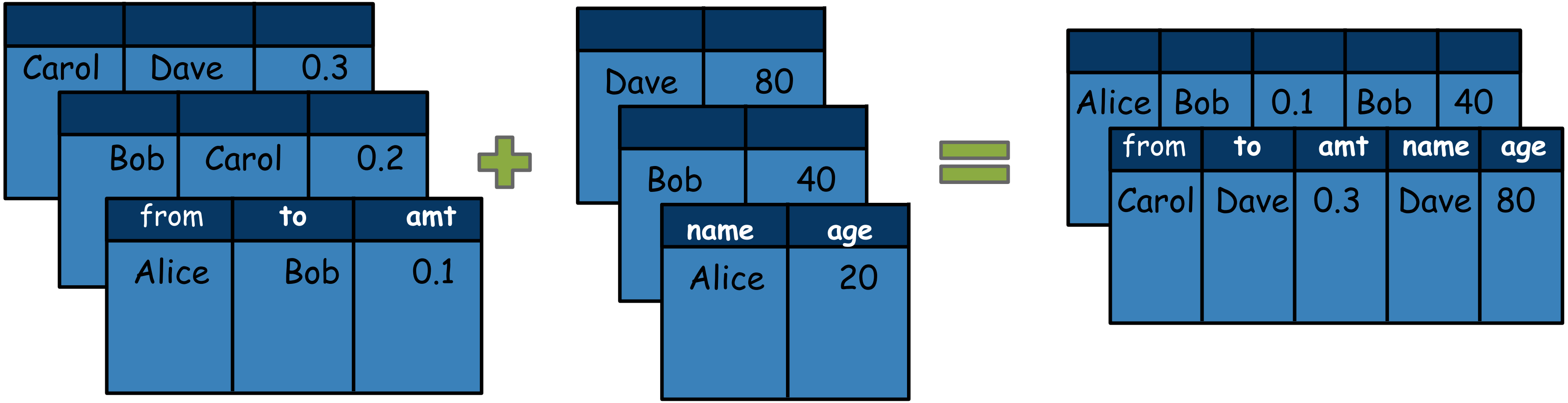
isLocal



False

join

joinExprs = x.to==y.name joinType = 'inner'



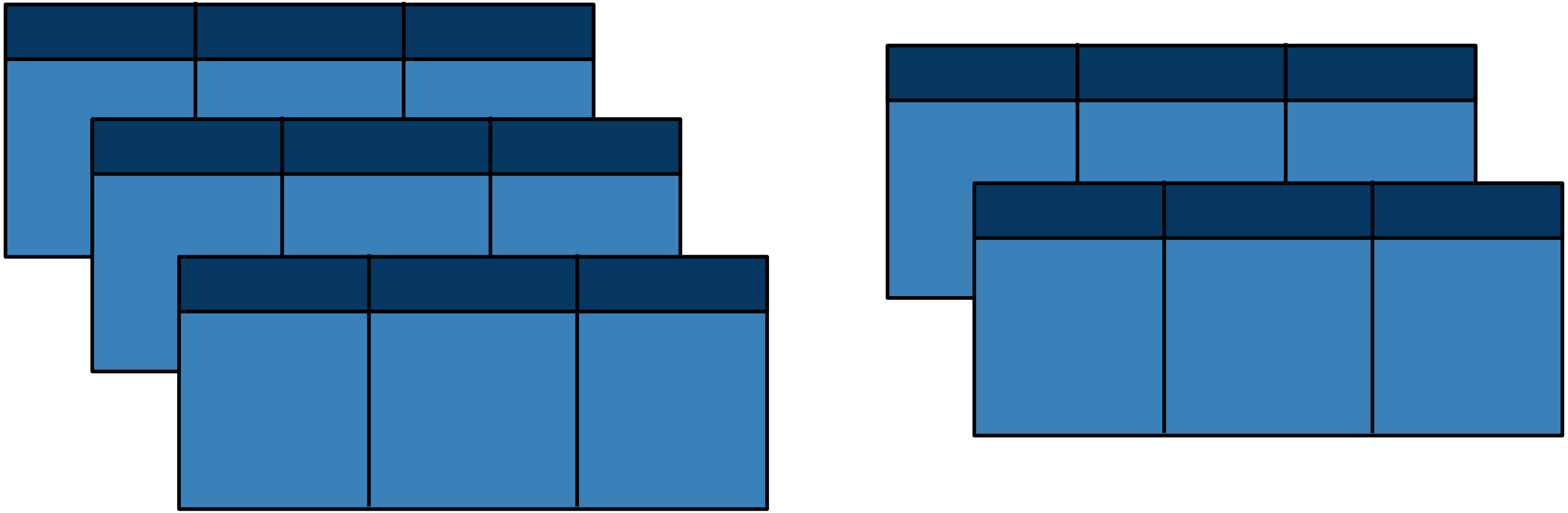
from	to	amt
Alice	Bob	0.1
Bob	Carol	0.2
Carol	Dave	0.3

name	age
Alice	20
Bob	40
Dave	80

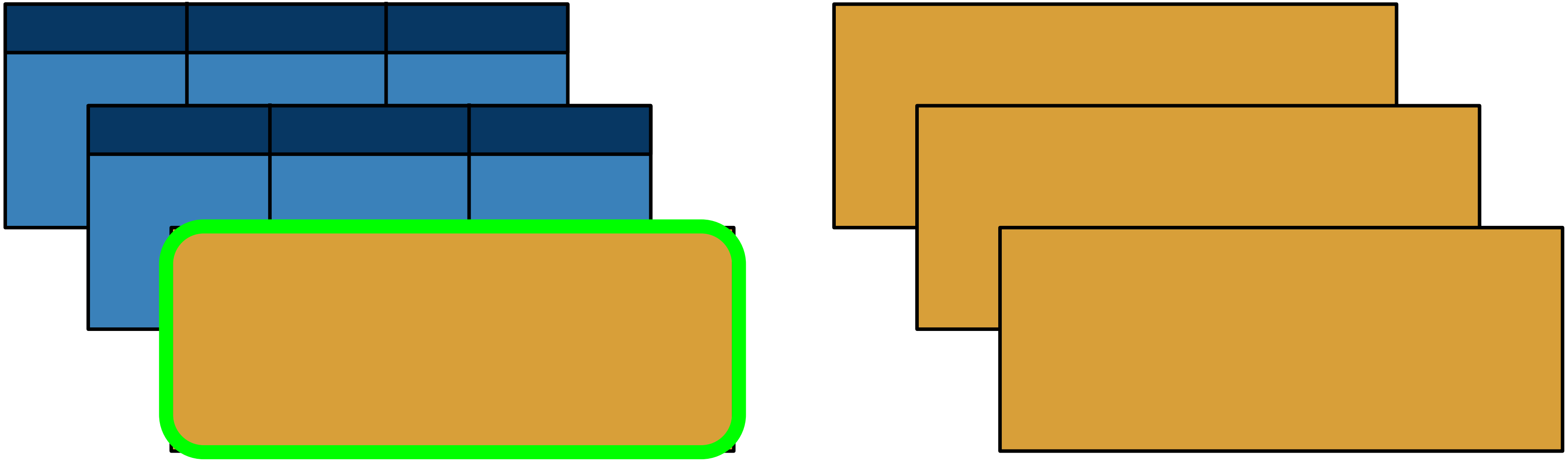
from	to	amt	name	age
Carol	Dave	0.3	Dave	80
Alice	Bob	0.1	Bob	40

limit

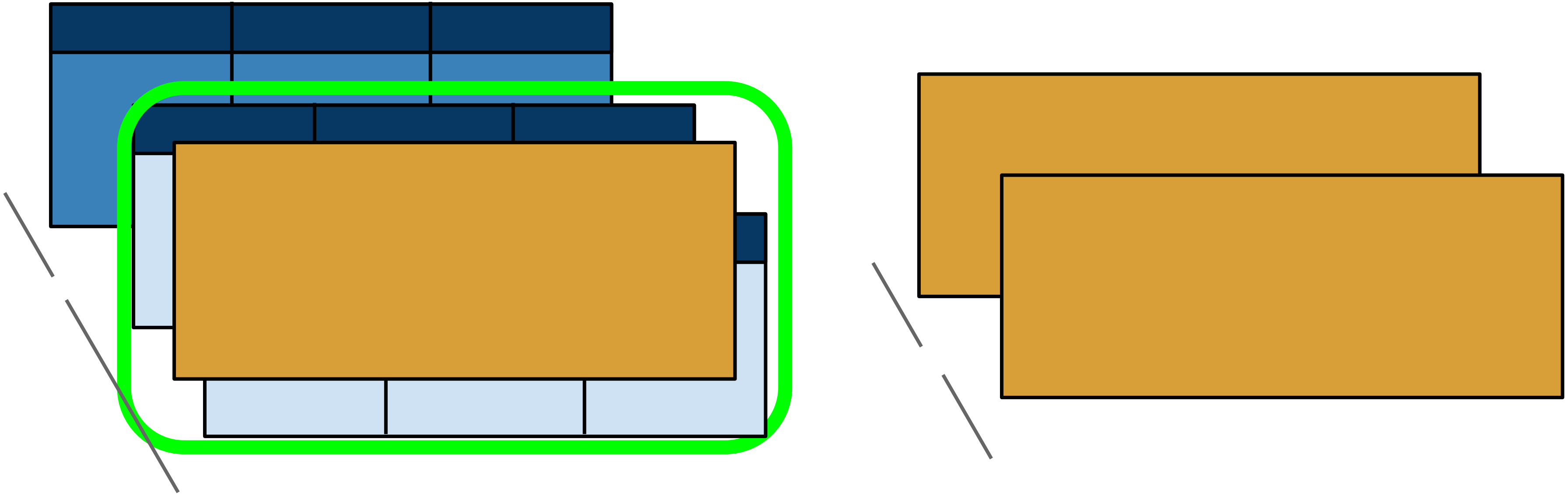
num = 2



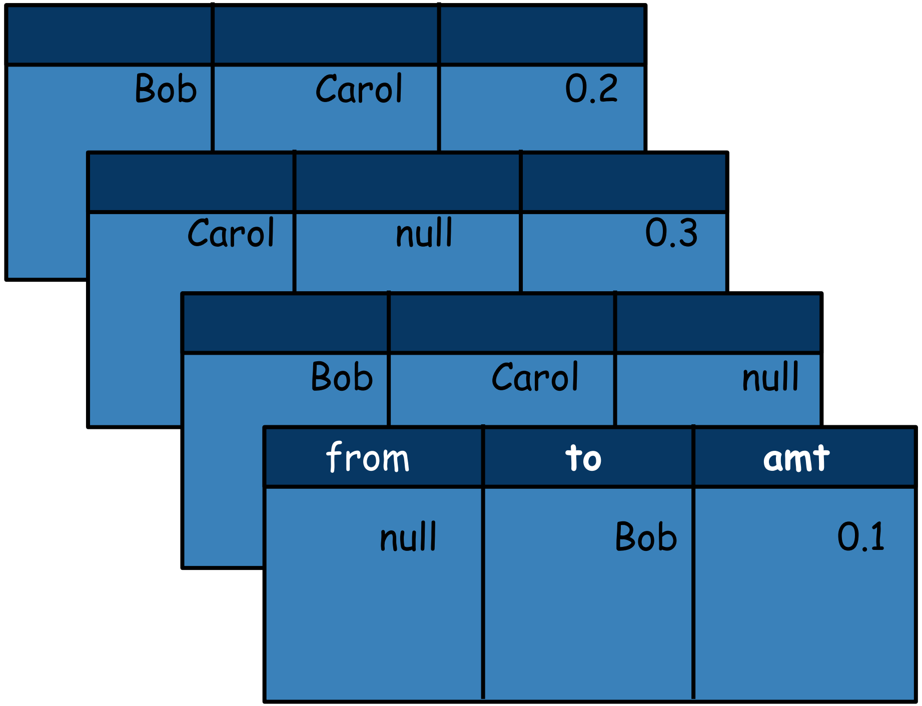
map



mapPartitions



na



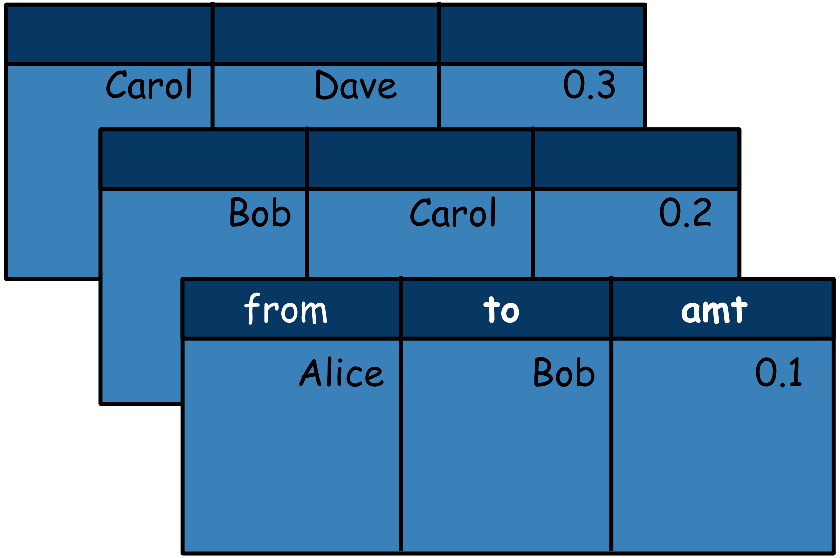
from	to	amt
Bob	Carol	0.2
Carol	null	0.3
Bob	Carol	null
null	Bob	0.1

DataFrameNaFunctions Object

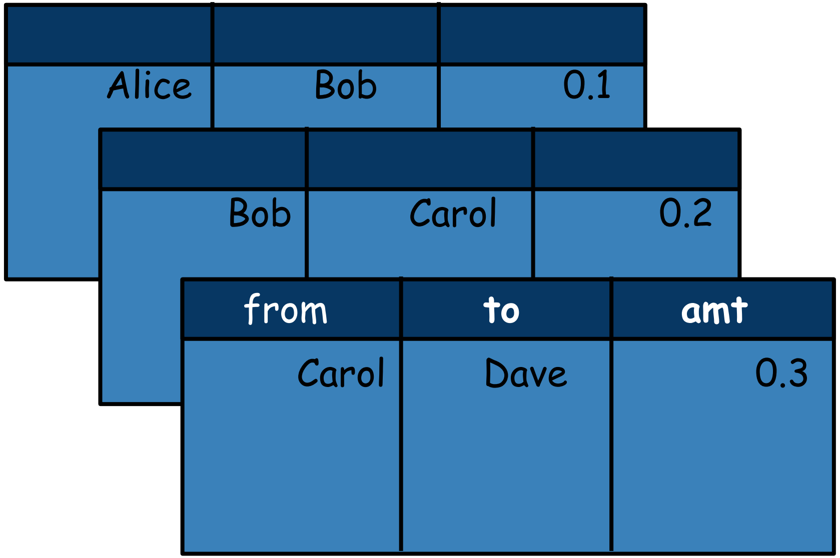
with methods: drop, fill, replace

orderBy

cols = ['from'], ascending = [False]



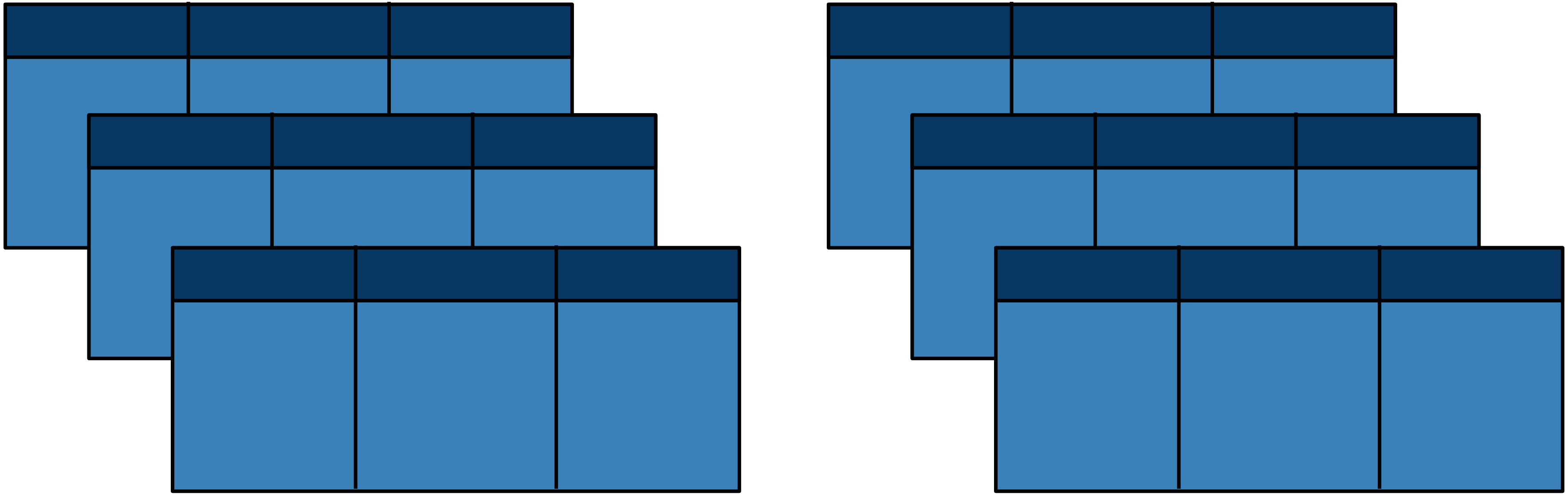
from	to	amt
Alice	Bob	0.1
Bob	Carol	0.2
Carol	Dave	0.3



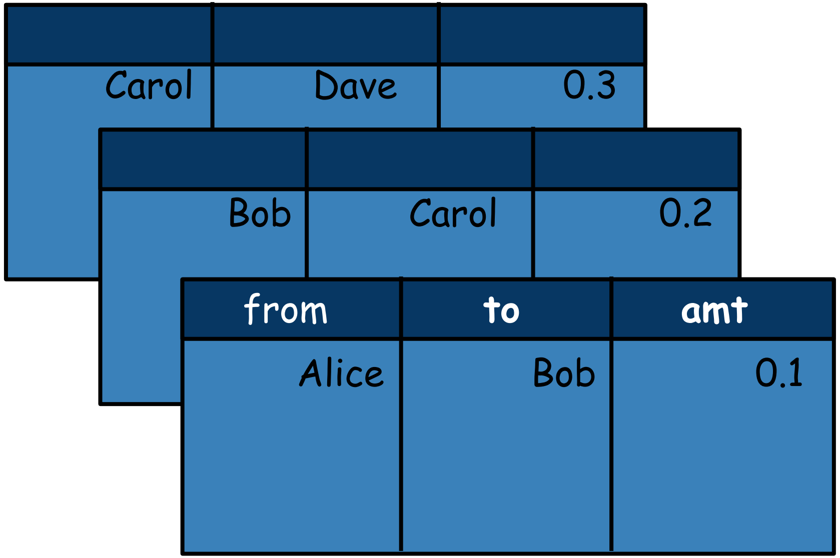
from	to	amt
Carol	Dave	0.3
Bob	Carol	0.2
Alice	Bob	0.1

persist

strorageLevel =
StorageLevel(MEMORY_ONLY_SER)



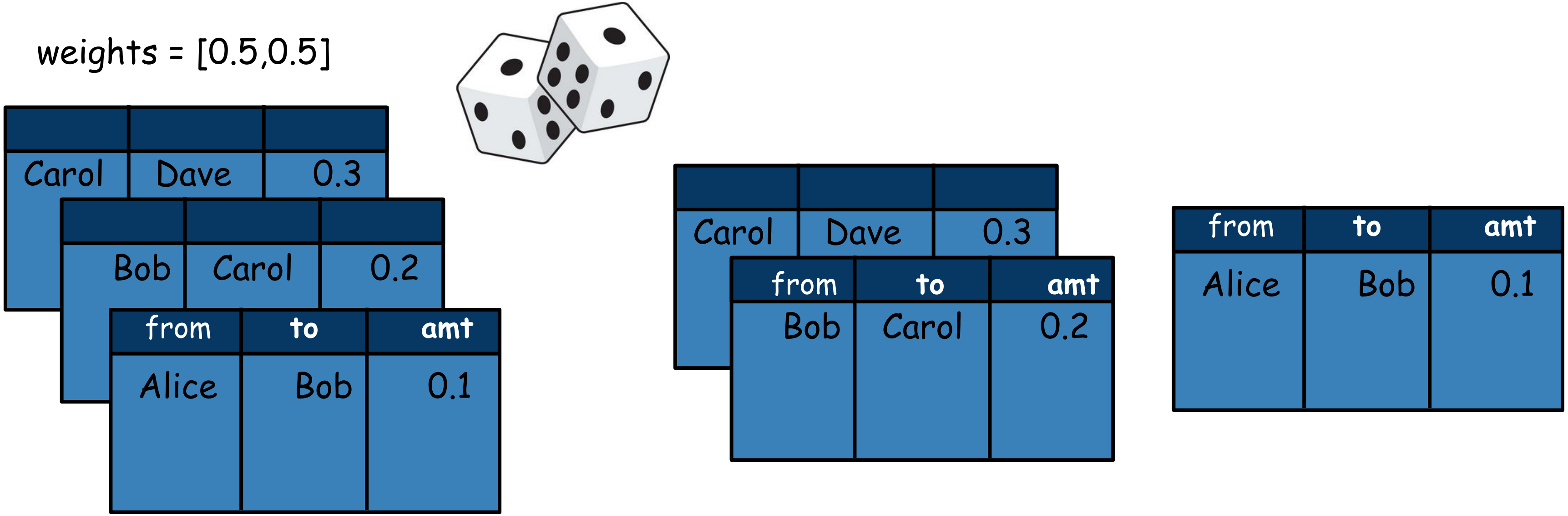
printSchema



stdout

```
root
 |-- from: string (nullable = true)
 |-- to: string (nullable = true)
 |-- amt: double (nullable = true)
```

randomSplit



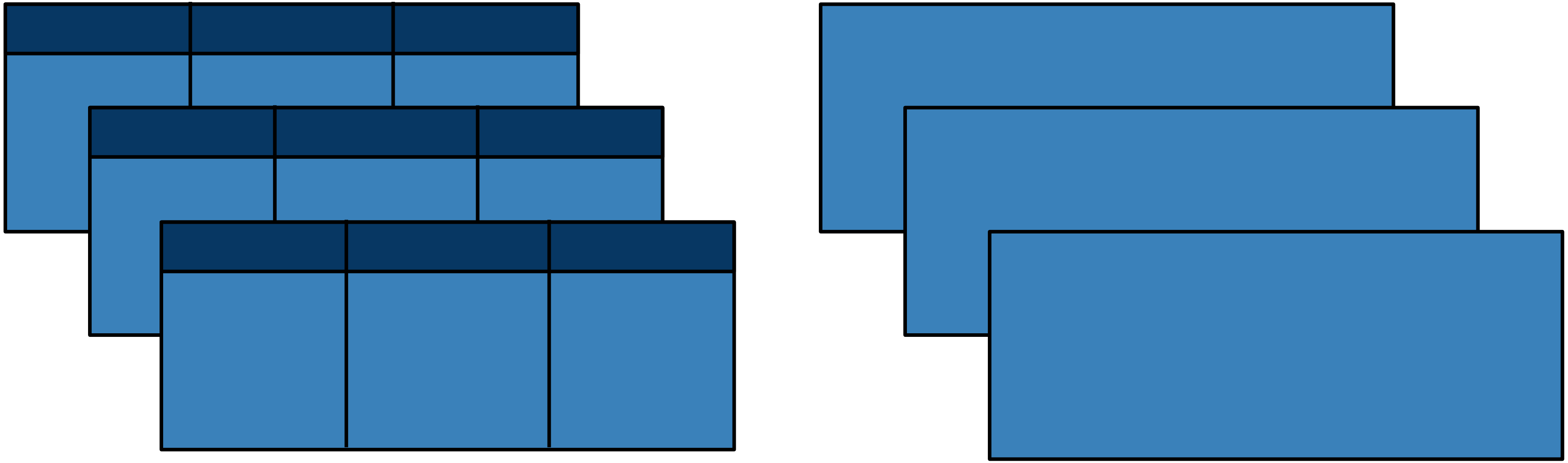
weights = [0.5,0.5]

from	to	amt
Carol	Dave	0.3
Bob	Carol	0.2
Alice	Bob	0.1

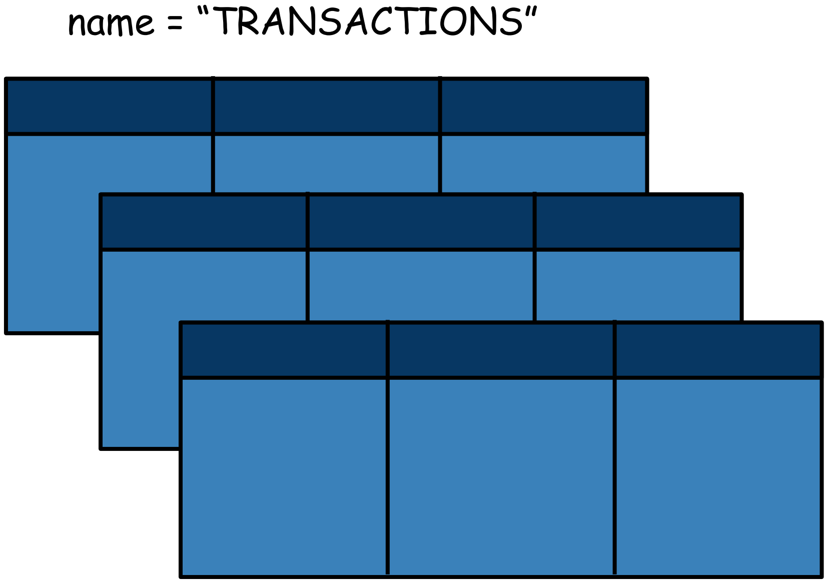
from	to	amt
Carol	Dave	0.3
Bob	Carol	0.2

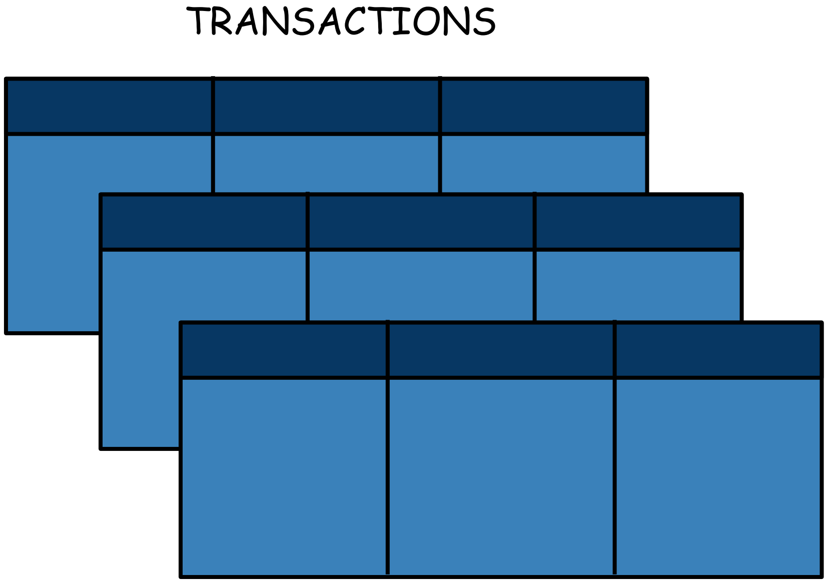
from	to	amt
Alice	Bob	0.1

rdd

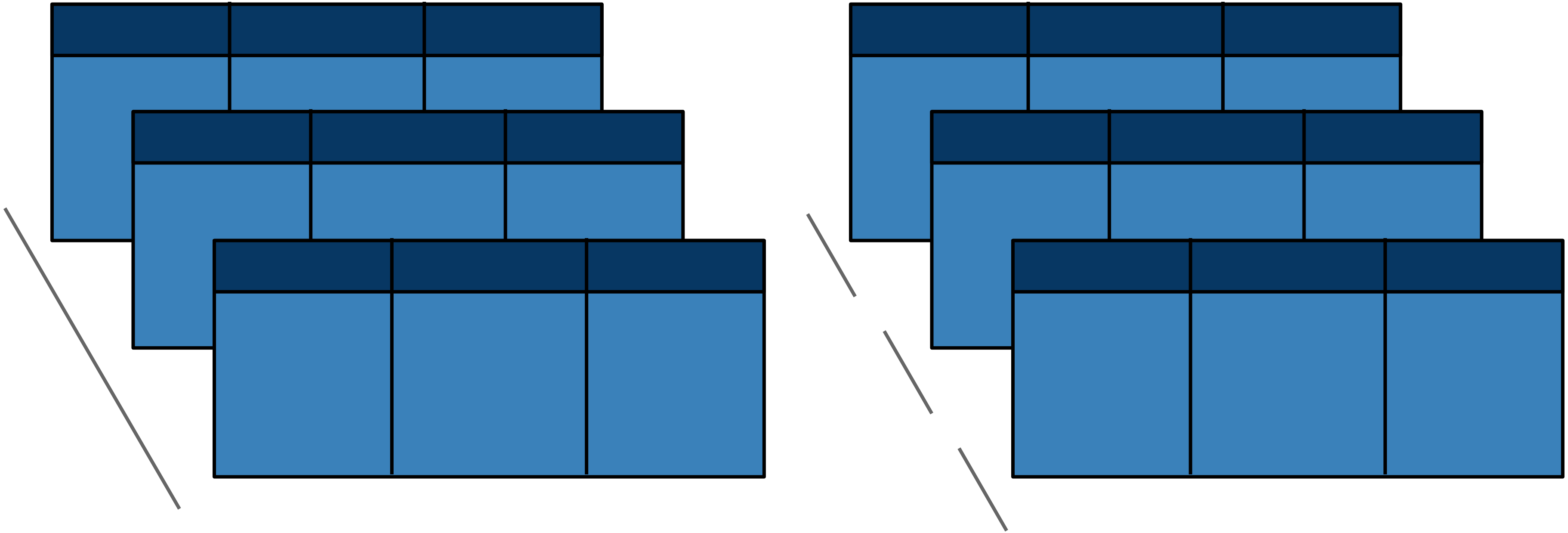


registerTempTable



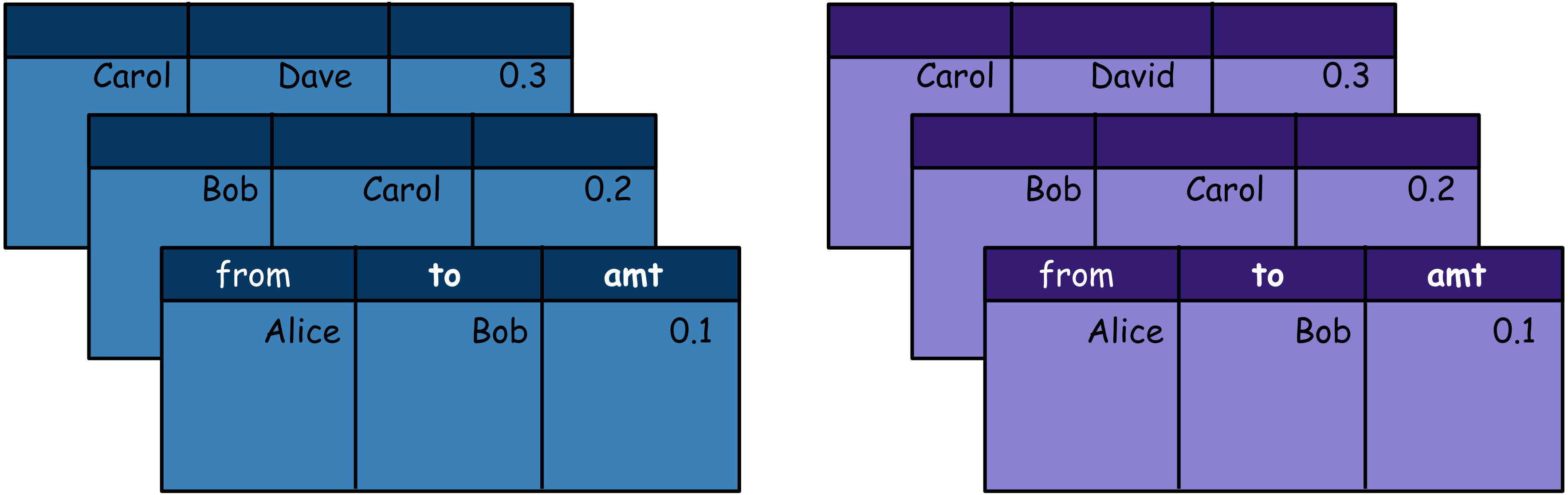


repartition



replace

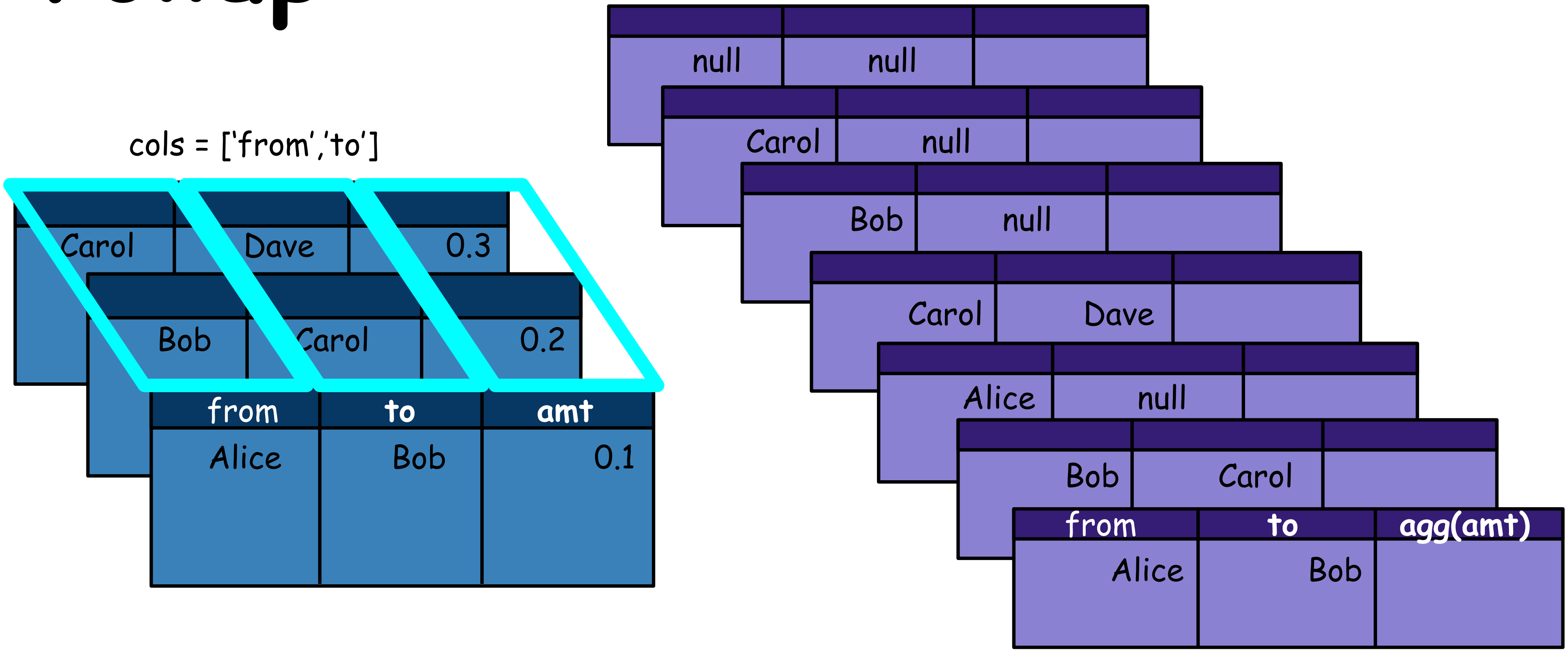
to_replace = 'Dave' value = 'David'



from	to	amt
Alice	Bob	0.1
Bob	Carol	0.2
Carol	Dave	0.3

from	to	amt
Alice	Bob	0.1
Bob	Carol	0.2
Carol	David	0.3

rollup

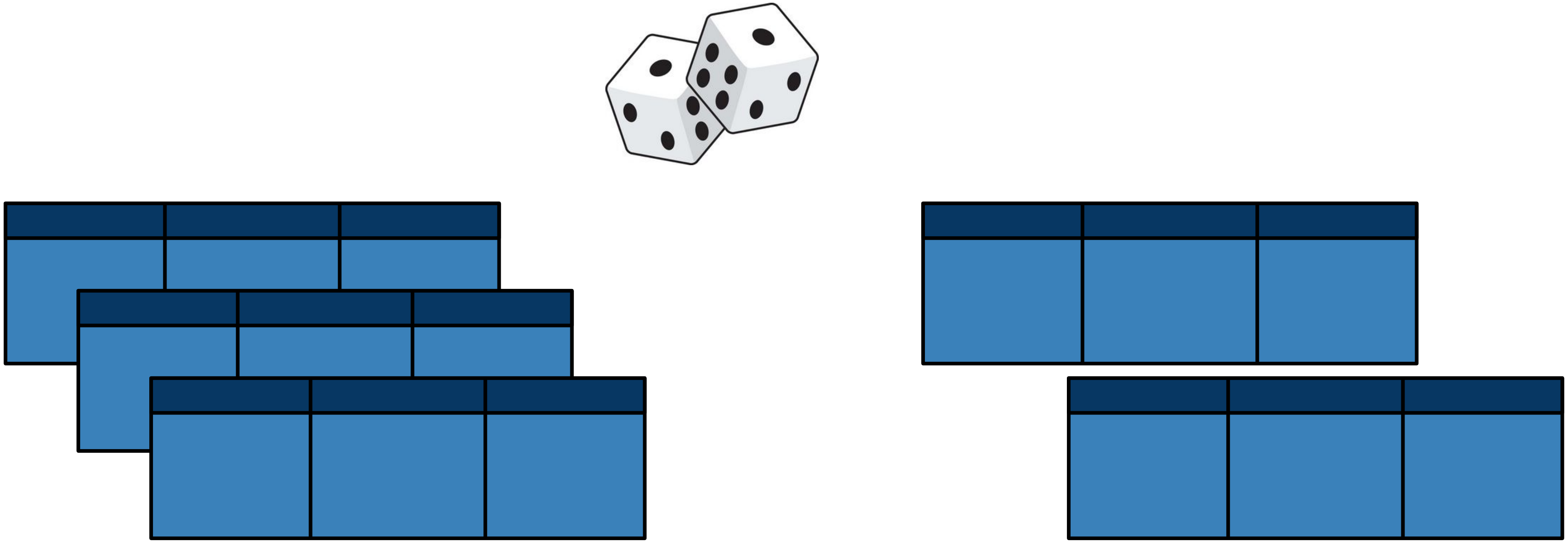


cols = ['from','to']

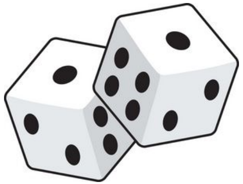
from	to	amt
Alice	Bob	0.1
Bob	Carol	0.2
Carol	Dave	0.3

from	to	agg(amt)
Alice	Bob	
Bob	Carol	
Alice	null	
Carol	Dave	
Bob	null	
Carol	null	
null	null	

sample



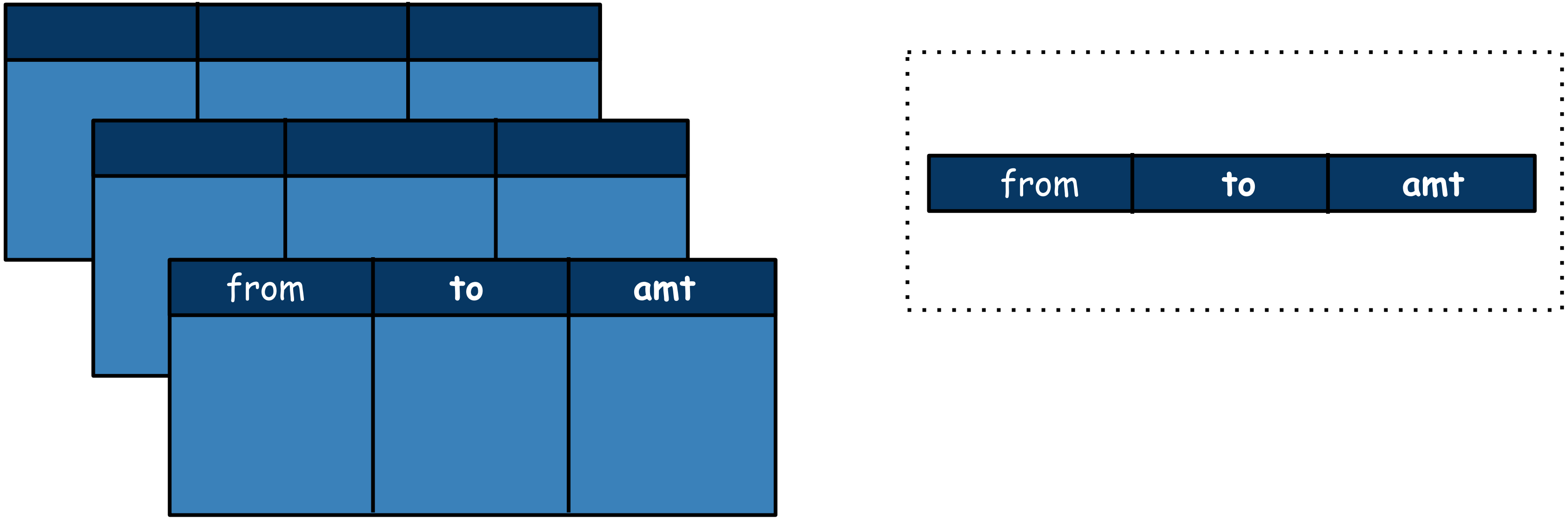
sampleBy



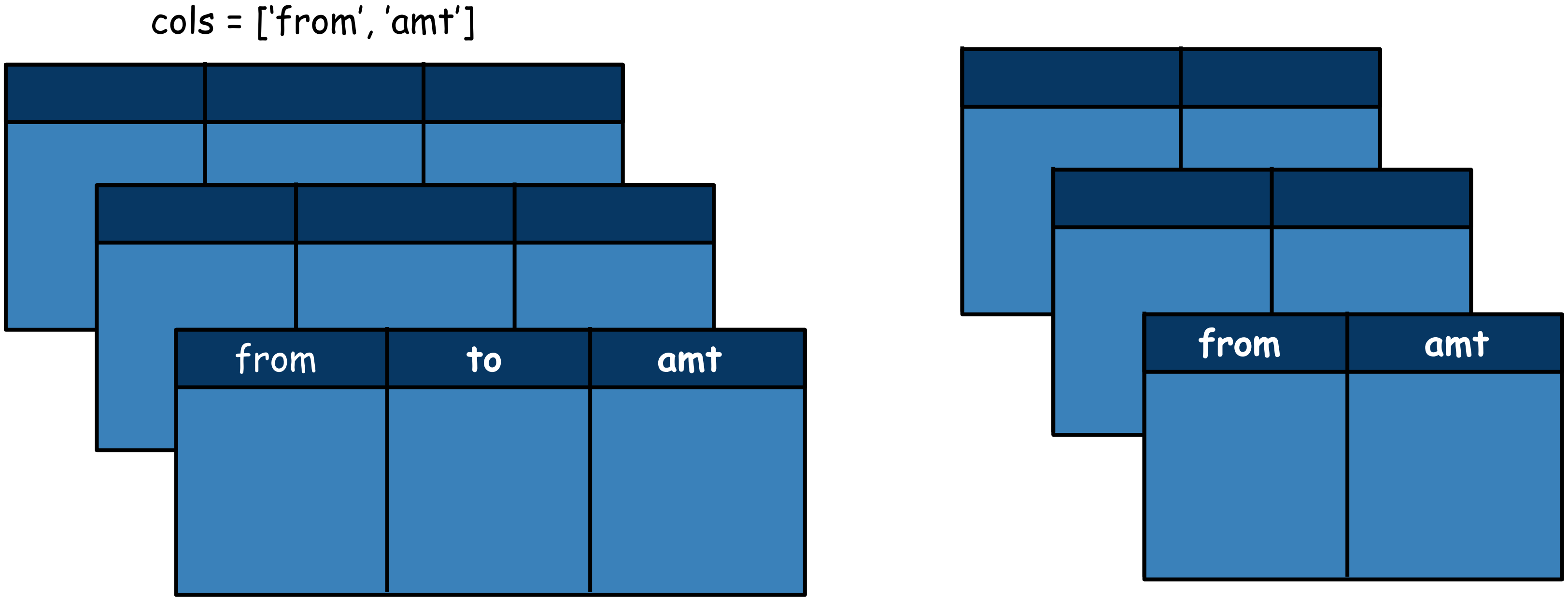
from	to	amt
Bob	Carol	0.6
Bob	Bob	0.5
Alice	Dave	0.4
Alice	Alice	0.3
Alice	Carol	0.2
Alice	Bob	0.1

from	to	amt
Bob	Carol	0.6
Bob	Bob	0.5
Alice	Bob	0.3

schema



select



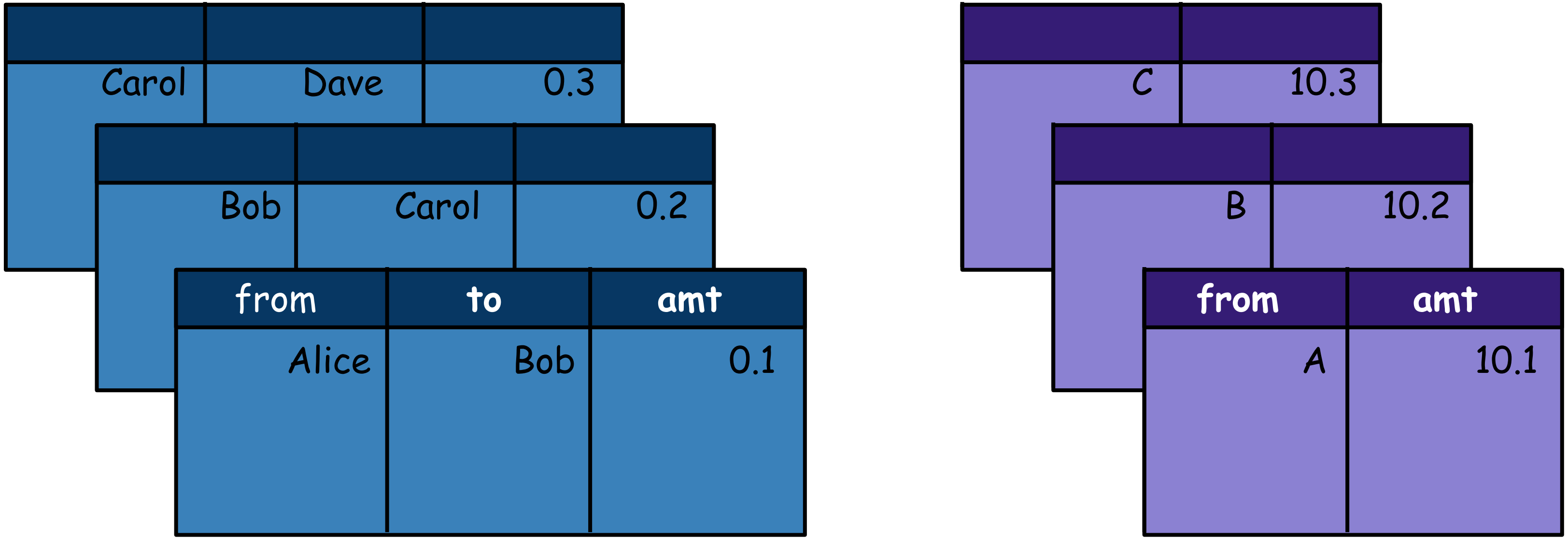
cols = [‘from’, ‘amt’]

from to amt

from amt

selectExpr

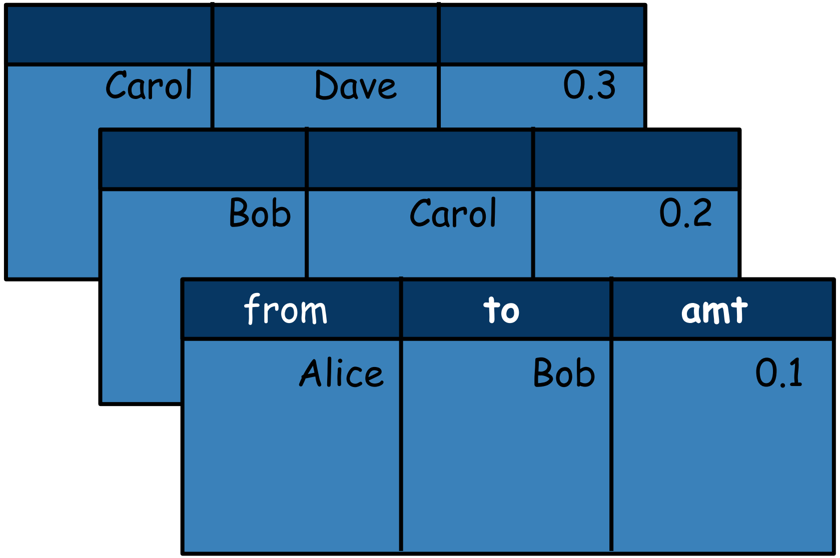
expr = [“substr(from,1,1)”, “amt + 10”]



from	to	amt
Alice	Bob	0.1
Bob	Carol	0.2
Carol	Dave	0.3

from	amt
A	10.1
B	10.2
C	10.3

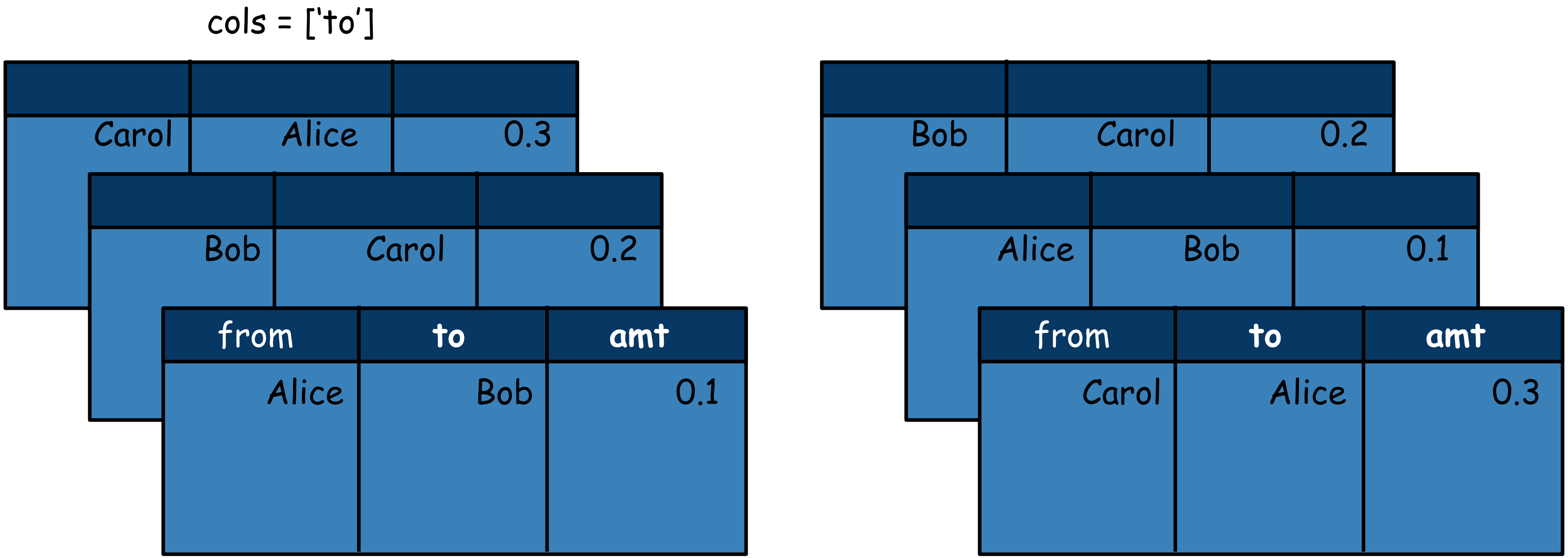
show



stdout

```
+-----+-----+---+
| from|   to|amt|
+-----+-----+---+
|Alice|  Bob|0.1|
|  Bob|Carol|0.2|
|Carol| Dave|0.3|
+-----+-----+---+
```

sort

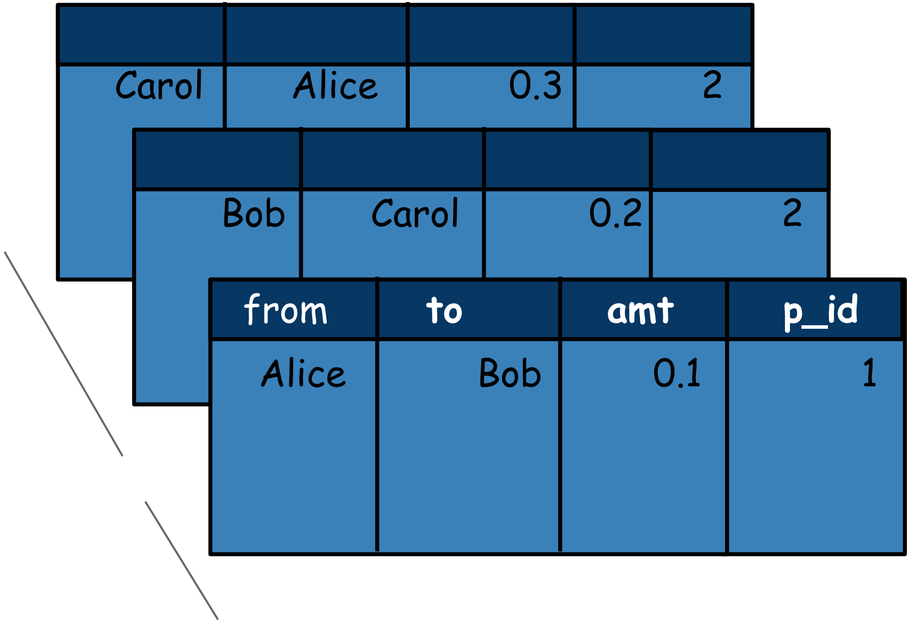


cols = [‘to’]

from	to	amt
Alice	Bob	0.1
Bob	Carol	0.2
Carol	Alice	0.3

from	to	amt
Carol	Alice	0.3
Alice	Bob	0.1
Bob	Carol	0.2

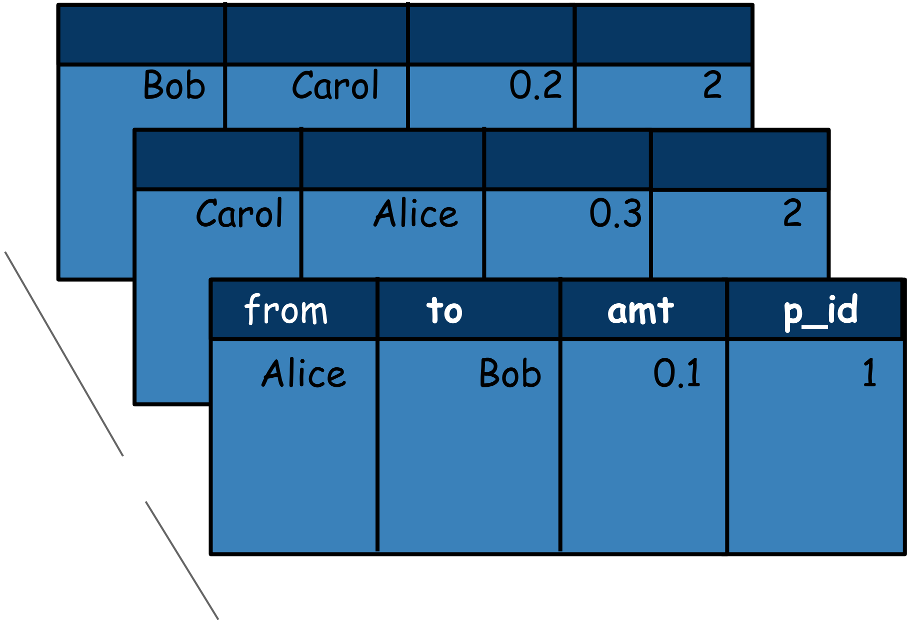
sortWithinPartitions



Carol	Alice	0.3	2

Bob	Carol	0.2	2

from	to	amt	p_id
Alice	Bob	0.1	1

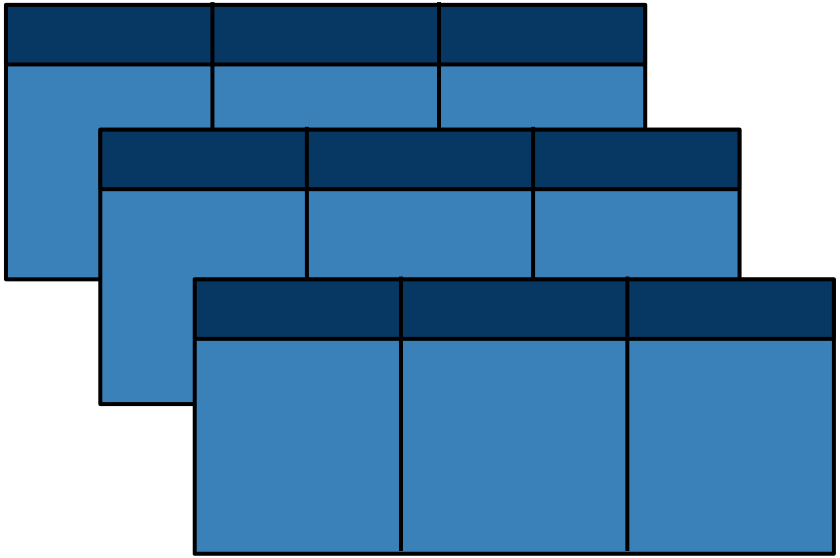


Bob	Carol	0.2	2

Carol	Alice	0.3	2

from	to	amt	p_id
Alice	Bob	0.1	1

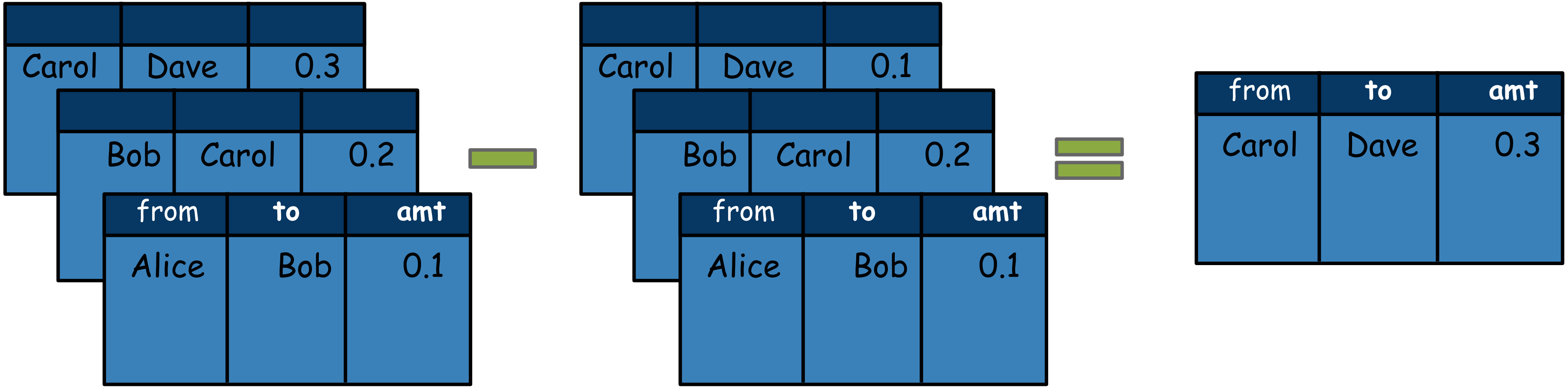
stat



DataFrameStatFunctions Object

with methods: corr, cov, corsstab, freqItems, sampleBy

subtract



from	to	amt
Alice	Bob	0.1
Bob	Carol	0.2
Carol	Dave	0.3

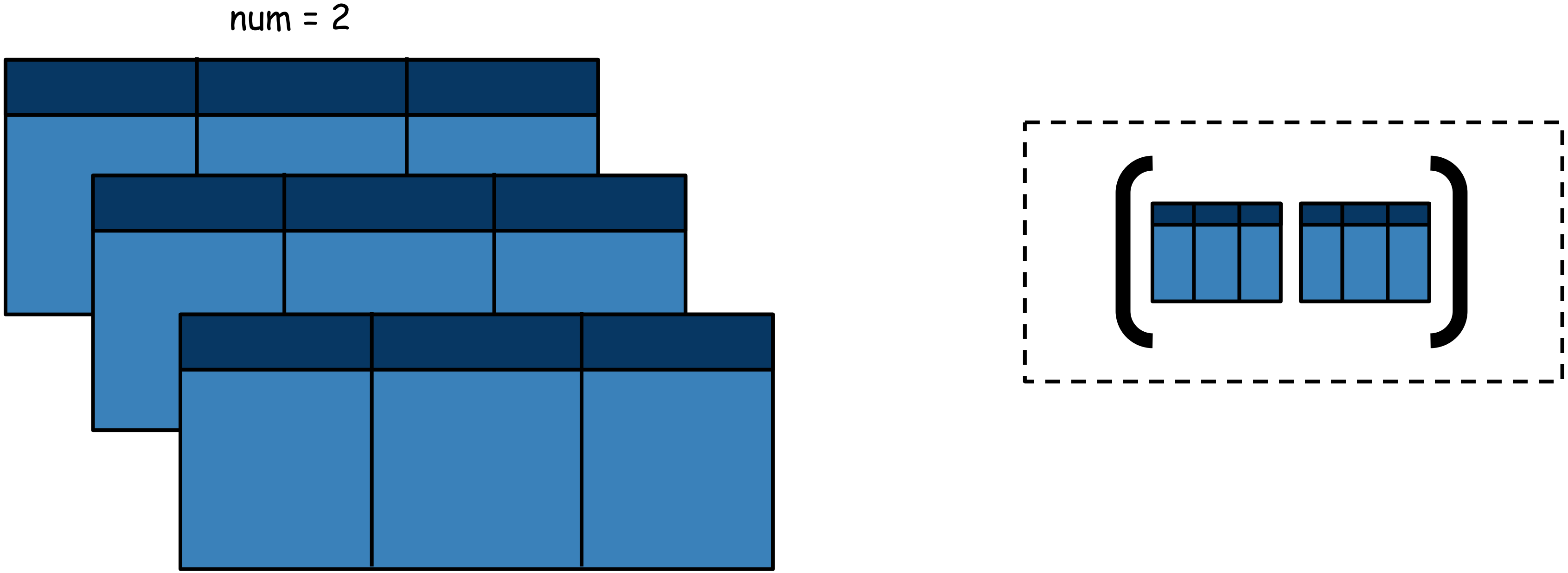
−

from	to	amt
Alice	Bob	0.1
Bob	Carol	0.2
Carol	Dave	0.1

=

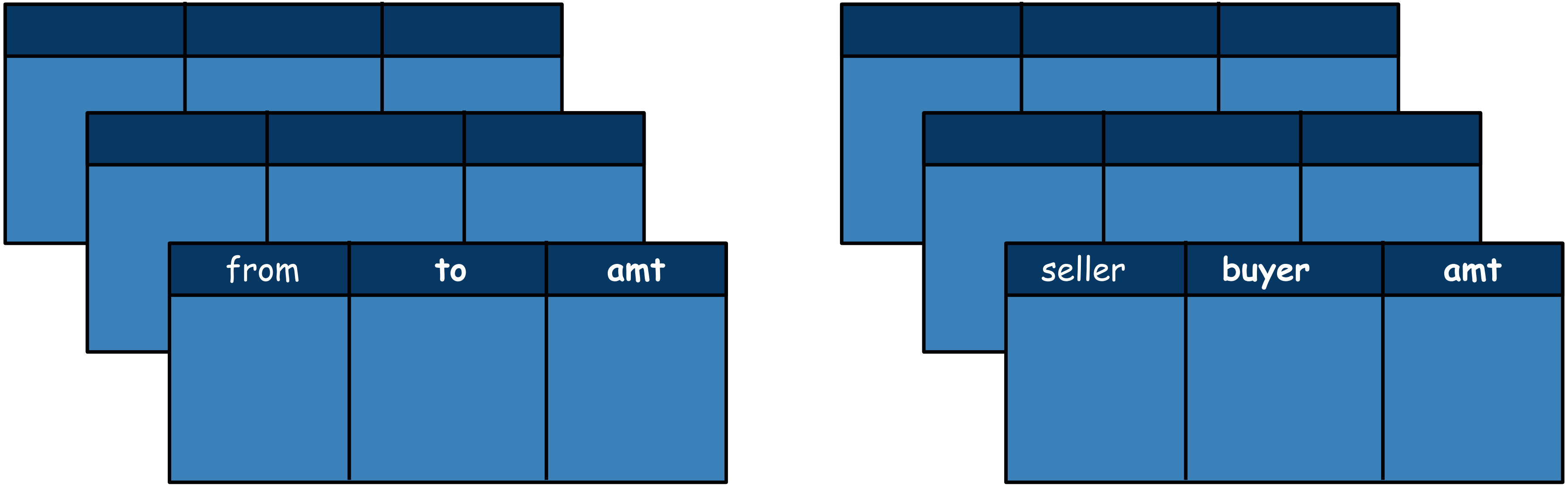
from	to	amt
Carol	Dave	0.3

take

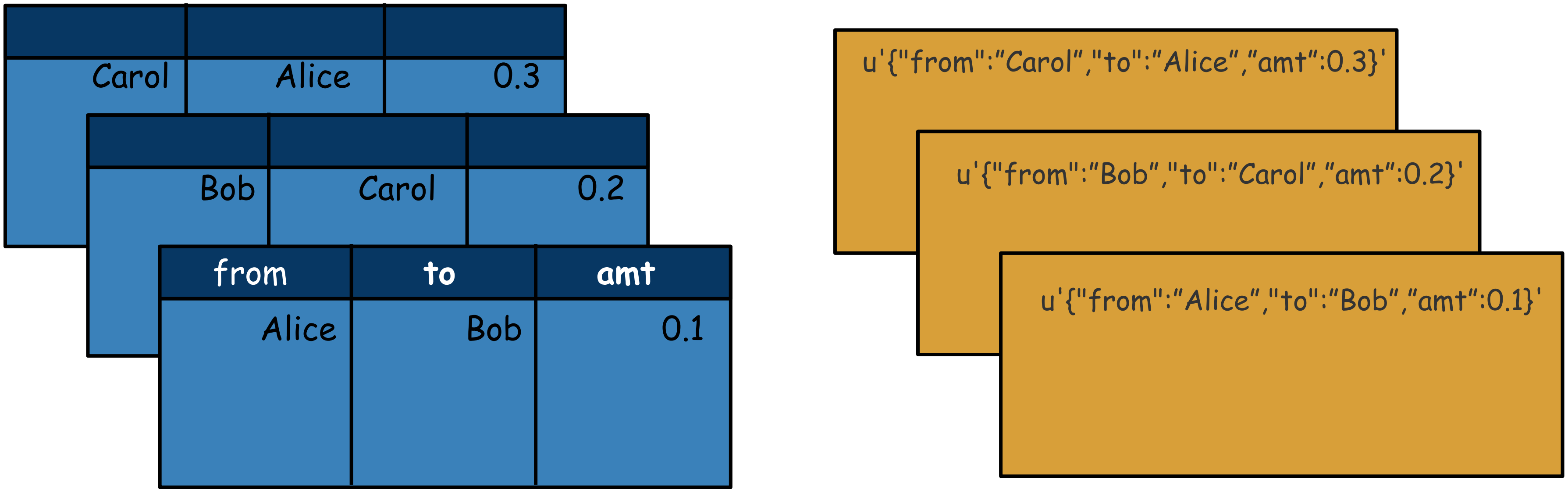


toDF

cols = [“seller”, “buyer”]



toJSON



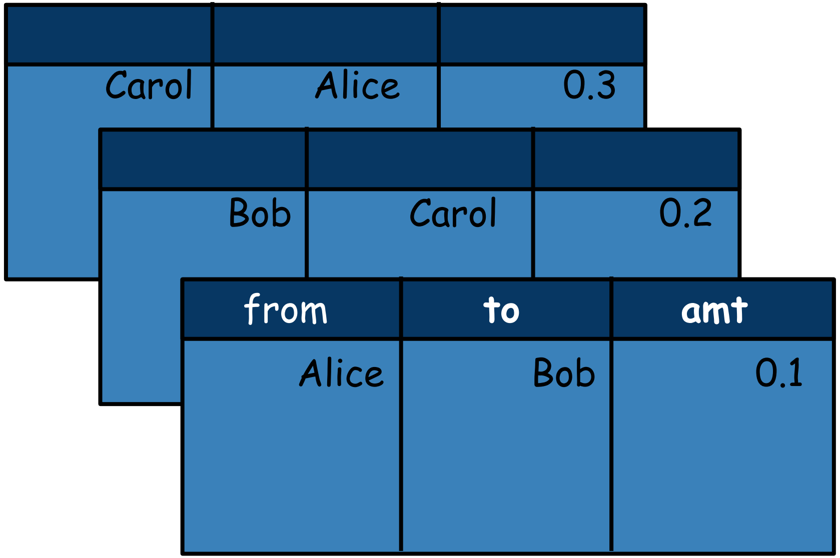
Carol	Alice	0.3

Bob	Carol	0.2

from	to	amt
Alice	Bob	0.1

```
u'{"from":"Carol","to":"Alice","amt":0.3}'
u'{"from":"Bob","to":"Carol","amt":0.2}'
u'{"from":"Alice","to":"Bob","amt":0.1}'
```

toPandas



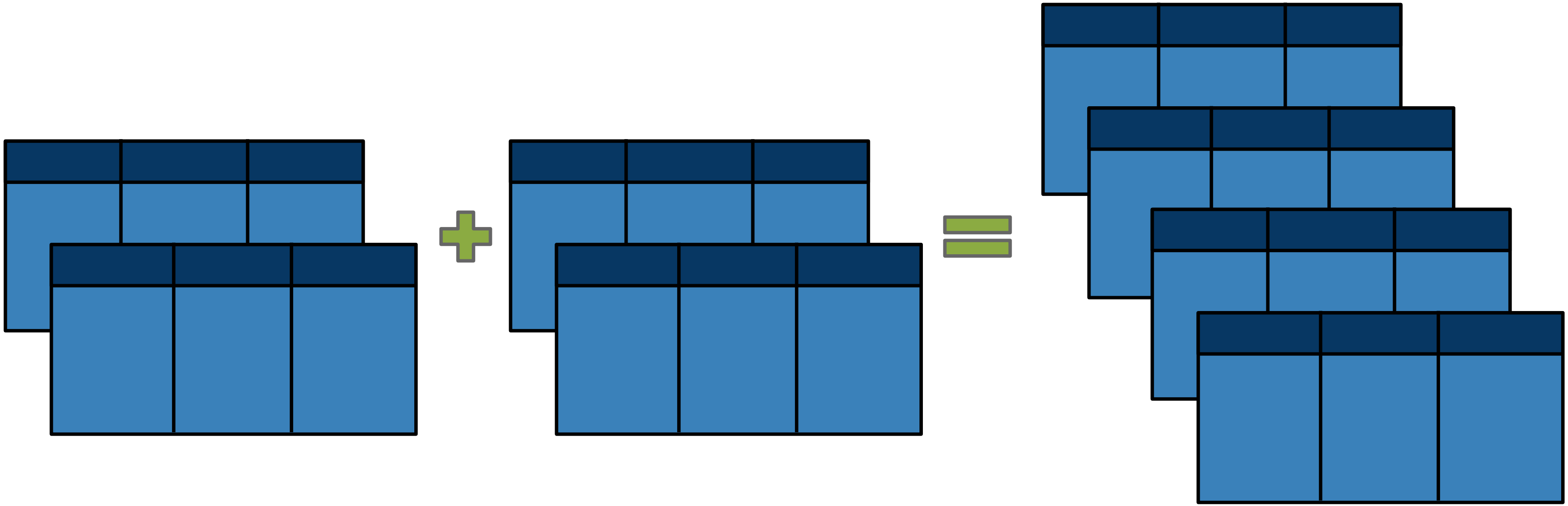
Carol Alice 0.3

Bob Carol 0.2

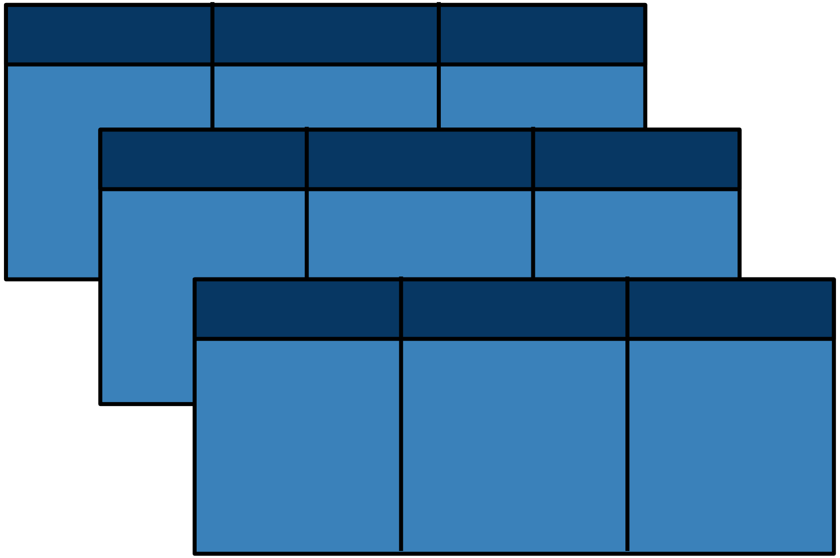
from	to	amt
Alice	Bob	0.1

```
   from  to   amt
0  Alice  Bob  0.1
1  Bob  Carol  0.2
2  Carol  Alice  0.3
```

unionAll

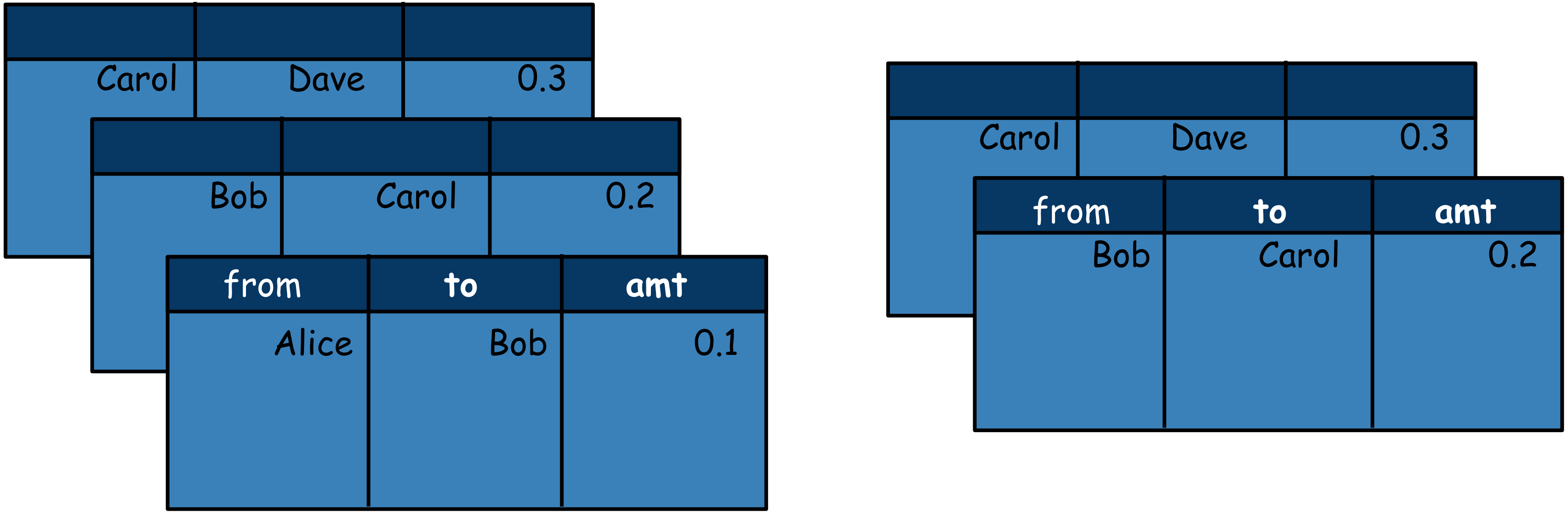


unpersist



where (filter)

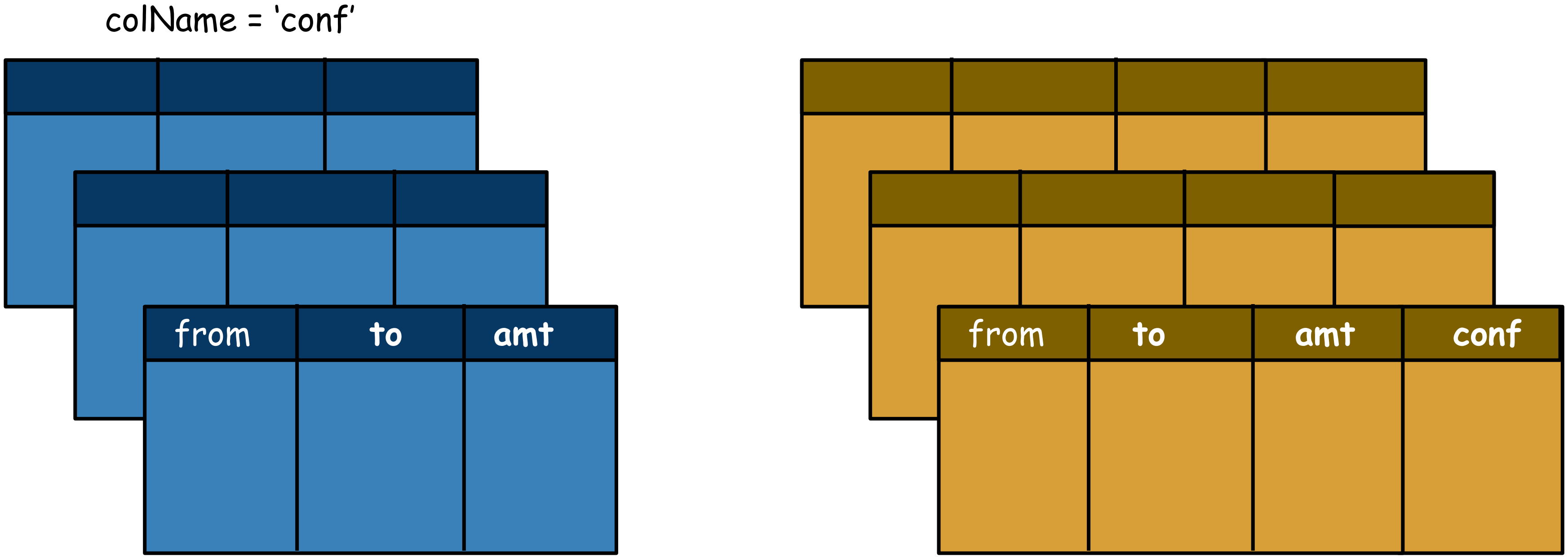
condition = “amt > 0.1”



from	to	amt
Alice	Bob	0.1
Bob	Carol	0.2
Carol	Dave	0.3

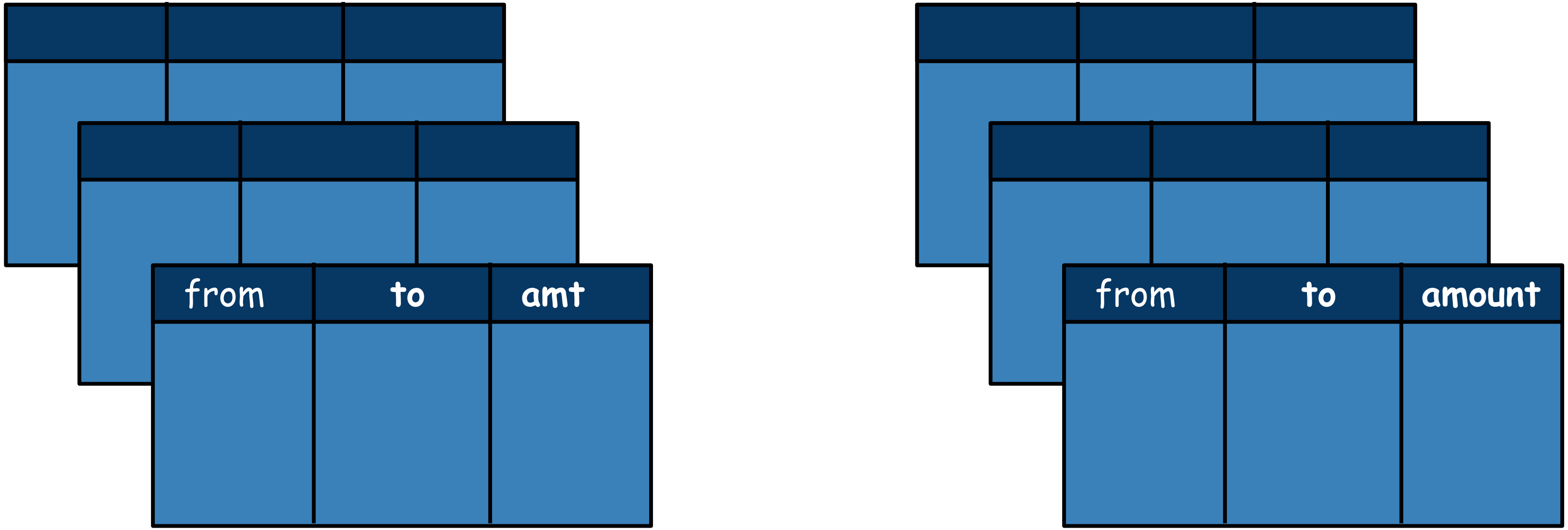
from	to	amt
Bob	Carol	0.2
Carol	Dave	0.3

withColumn



withColumnRenamed

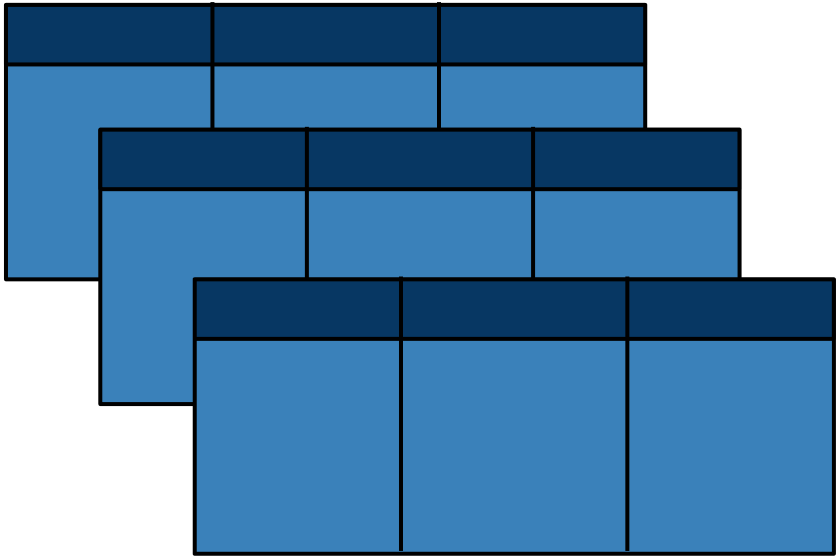
existing = ‘amt’ col = ‘amount’



from to amt

from to amount

write



DataFrameWriter Object

with methods: format, insertInto, jdbc, json, mode, option, options, orc, parquet, partitionBy, save, saveAsTable, text