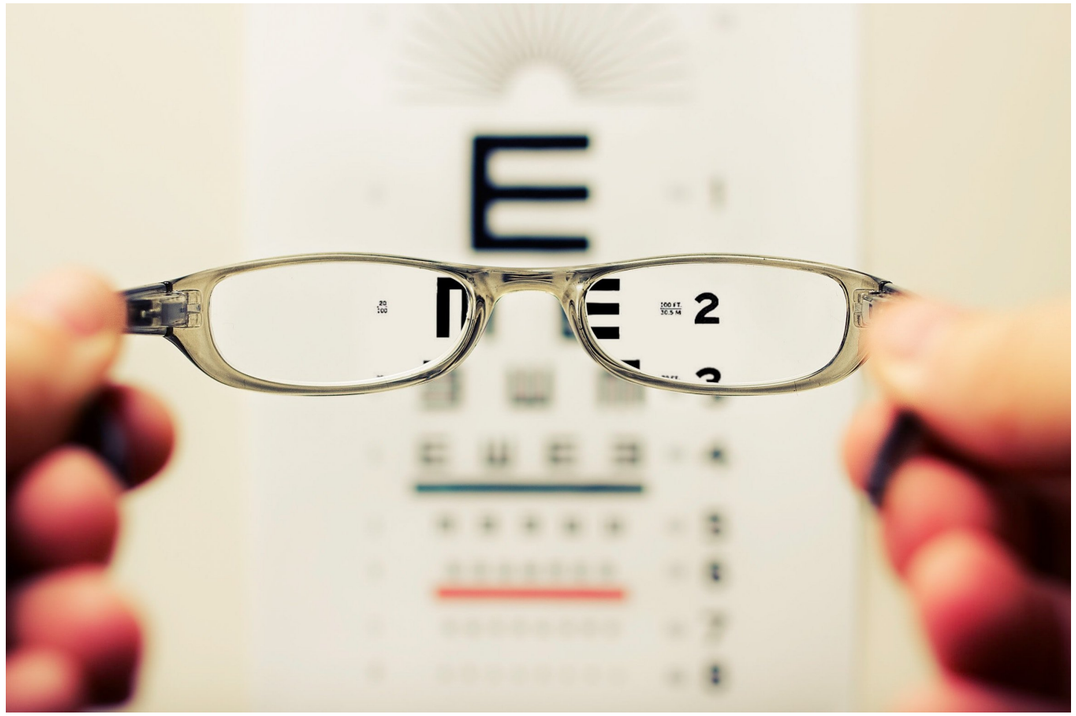
Inner join

JOINING DATA WITH PANDAS

Aaren Stubberfield
Instructor

datacamp

For clarity



Tables = DataFrames

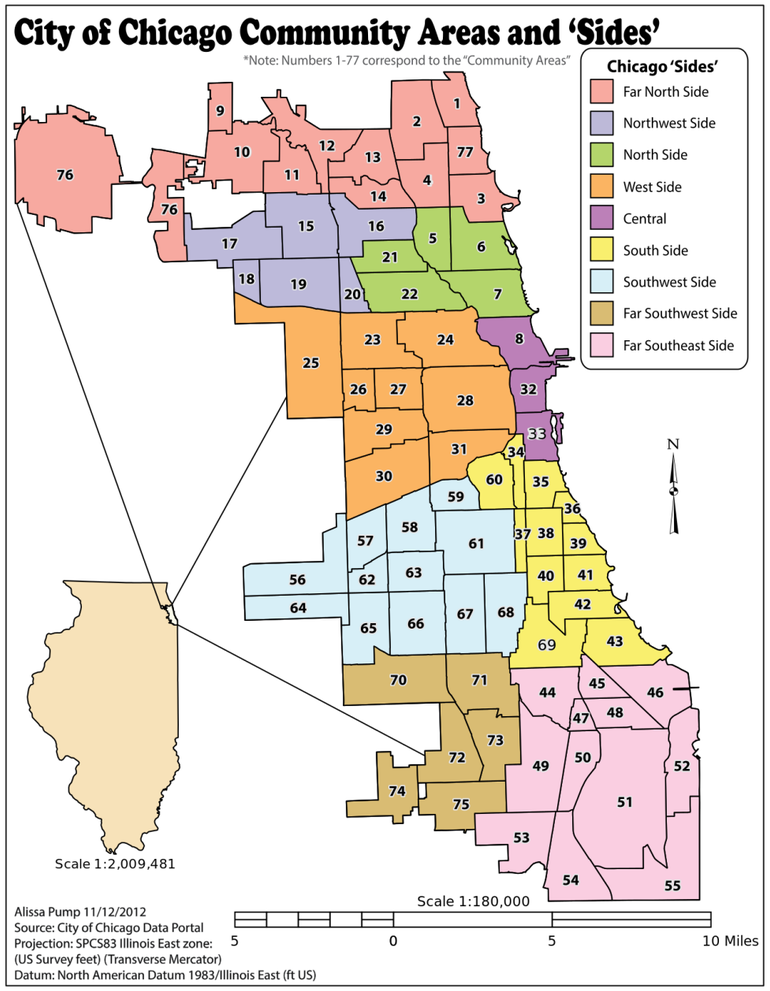
Merging = Joining

[1] Photo by David Travis on Unsplash

Chicago data portal dataset

[1] Photo by Pedro Lastra on Unsplash

Datasets for example



United States® Census Bureau

[1] Ward image By Alissapump, Own work, CC BY-SA 3.0

The ward data

```
wards = pd.read_csv('Ward_Offices.csv')
print(wards.head())
print(wards.shape)
```

```
  ward  alderman          address           zip
0 1     Proco "Joe" ...   2058 NORTH W...   60647
1 2     Brian Hopkins     1400 NORTH  ...   60622
2 3     Pat Dowell        5046 SOUTH S...   60609
3 4     William D. B...   435 EAST 35T...   60616
4 5     Leslie A. Ha...   2325 EAST 71...   60649
(50, 4)
```

Census data

```
census = pd.read_csv('Ward_Census.csv')
print(census.head())
print(census.shape)
```

```
  ward  pop_2000  pop_2010  change  address          zip
0 1     52951     56149     6%      2765 WEST SA...  60647
1 2     54361     55805     3%      WM WASTE MAN...  60622
2 3     40385     53039     31%     17 EAST 38TH...  60653
3 4     51953     54589     5%      31ST ST HARB...  60653
4 5     55302     51455     -7%     JACKSON PARK...  60637
(50, 6)
```

Merging tables

```
   ward  alderman          address          zip
0  1     Proco "Joe" ...   2058 NORTH W...  60647
1  2     Brian Hopkins     1400 NORTH  ...  60622
2  3     Pat Dowell        5046 SOUTH S...  60609
3  4     William D. B...   435 EAST 35T...  60616
4  5     Leslie A. Ha...   2325 EAST 71...  60649
```

```
   ward  pop_2000  pop_2010  change  address          zip
0  1     52951     56149     6%      2765 WEST SA...  60647
1  2     54361     55805     3%      WM WASTE MAN...  60622
2  3     40385     53039     31%     17 EAST 38TH...  60653
3  4     51953     54589     5%      31ST ST HARB...  60653
4  5     55302     51455     -7%     JACKSON PARK...  60637
```

Inner join

wards cols comes first , followed by census cols

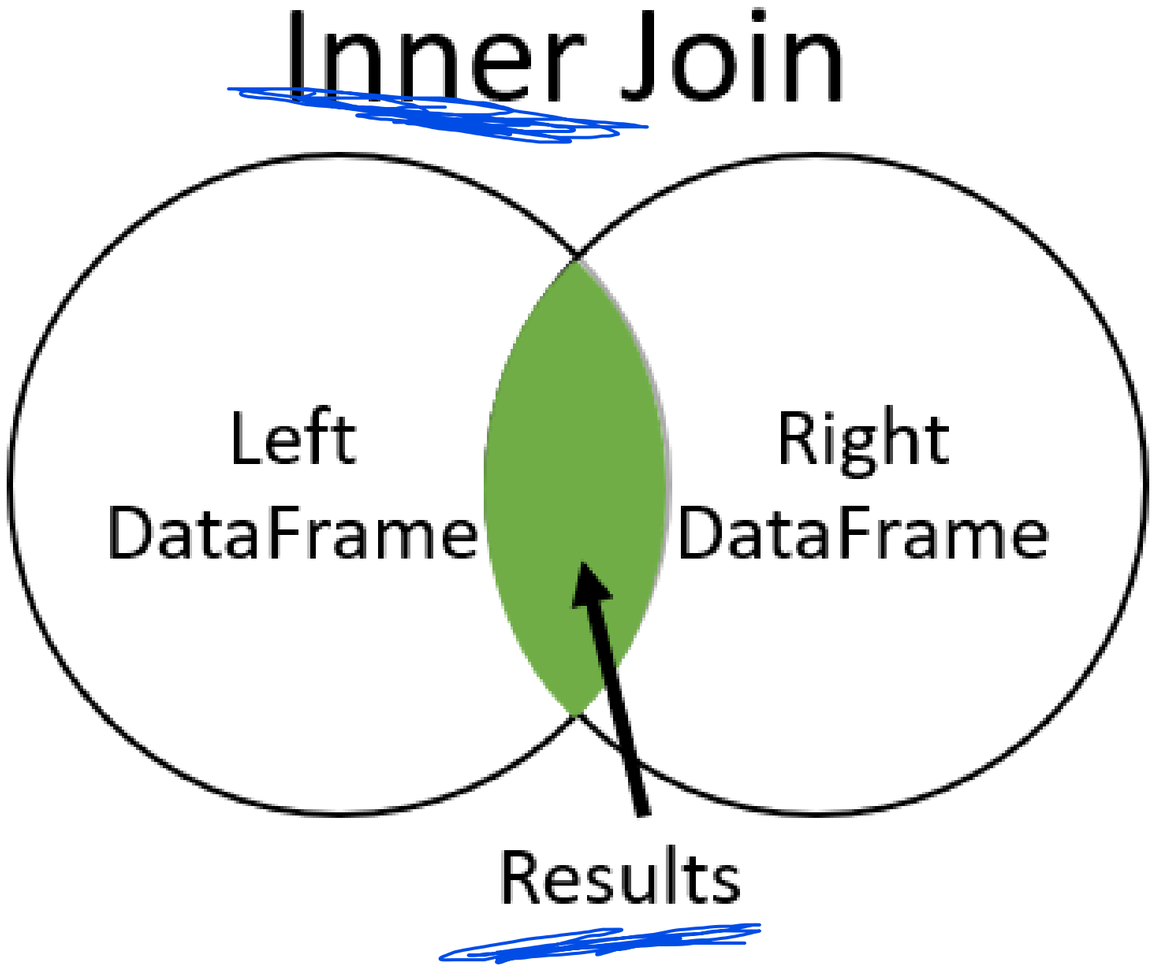
```
wards_census = wards.merge(census, on='ward')
print(wards_census.head(4))
```

```
  ward  alderman          address_x         zip_x  pop_2000  pop_2010  change  address_y         zip_y
0 1     Proco "Joe" ...   2058 NORTH W...   60647  52951     56149     6%      2765 WEST SA...   60647
1 2     Brian Hopkins     1400 NORTH  ...   60622  54361     55805     3%      WM WASTE MAN...   60622
2 3     Pat Dowell        5046 SOUTH S...   60609  40385     53039     31%     17 EAST 38TH...   60653
3 4     William D. B...   435 EAST 35T...   60616  51953     54589     5%      31ST ST HARB...   60653
```

```
print(wards_census.shape)
```

```
(50, 9)
```

Inner join



Suffixes

```
print(wards_census.columns)
```

```
Index(['ward', 'alderman', 'address_x', 'zip_x', 'pop_2000', 'pop_2010', 'change',
       'address_y', 'zip_y'],
      dtype='object')
```

Suffixes

```
wards_census = wards.merge(census, on='ward', suffixes=('_ward','_cen'))
print(wards_census.head())
print(wards_census.shape)
```

suffixes=('_ward', '_cen'): If there are columns with the same name (other than ward), they'll be renamed with these suffixes to avoid conflicts:
From wards, columns will get '_ward' added.
From census, columns will get '_cen' added

```
  ward  alderman         address_ward     zip_ward  pop_2000  pop_2010  change  address_cen      zi
0 1     Proco "Joe" ...  2058 NORTH W...  60647     52951     56149     6%      2765 WEST SA...  60
1 2     Brian Hopkins    1400 NORTH  ...  60622     54361     55805     3%      WM WASTE MAN...  60
2 3     Pat Dowell       5046 SOUTH S...  60609     40385     53039     31%     17 EAST 38TH...  60
3 4     William D. B...  435 EAST 35T...  60616     51953     54589     5%      31ST ST HARB...  60
4 5     Leslie A. Ha...  2325 EAST 71...  60649     55302     51455     -7%     JACKSON PARK...  60
(50, 9)
```

Let's practice!

JOINING DATA WITH PANDAS

Part-02

One to many relationships

JOINING DATA WITH PANDAS

Aaren Stubberfield
Instructor

datacamp

One-to-one

A	B	C
A1	B1	C1
A2	B2	C2
A3	B3	C3

C	D
C1	D1
C2	D2
C3	D3

One-To-One = Every row in the left table is related to only one row in the right table

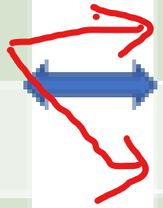
One-to-one example

```
   ward  alderman          address          zip
0  1     Proco "Joe" ...   2058 NORTH W...  60647
1  2     Brian Hopkins     1400 NORTH  ...  60622
2  3     Pat Dowell        5046 SOUTH S...  60609
3  4     William D. B...   435 EAST 35T...  60616
4  5     Leslie A. Ha...   2325 EAST 71...  60649
```

```
   ward  pop_2000  pop_2010  change  address          zip
0  1     52951     56149     6%      2765 WEST SA...  60647
1  2     54361     55805     3%      WM WASTE MAN...  60622
2  3     40385     53039     31%     17 EAST 38TH...  60653
3  4     51953     54589     5%      31ST ST HARB...  60653
4  5     55302     51455     -7%     JACKSON PARK...  60637
```

One-to-many

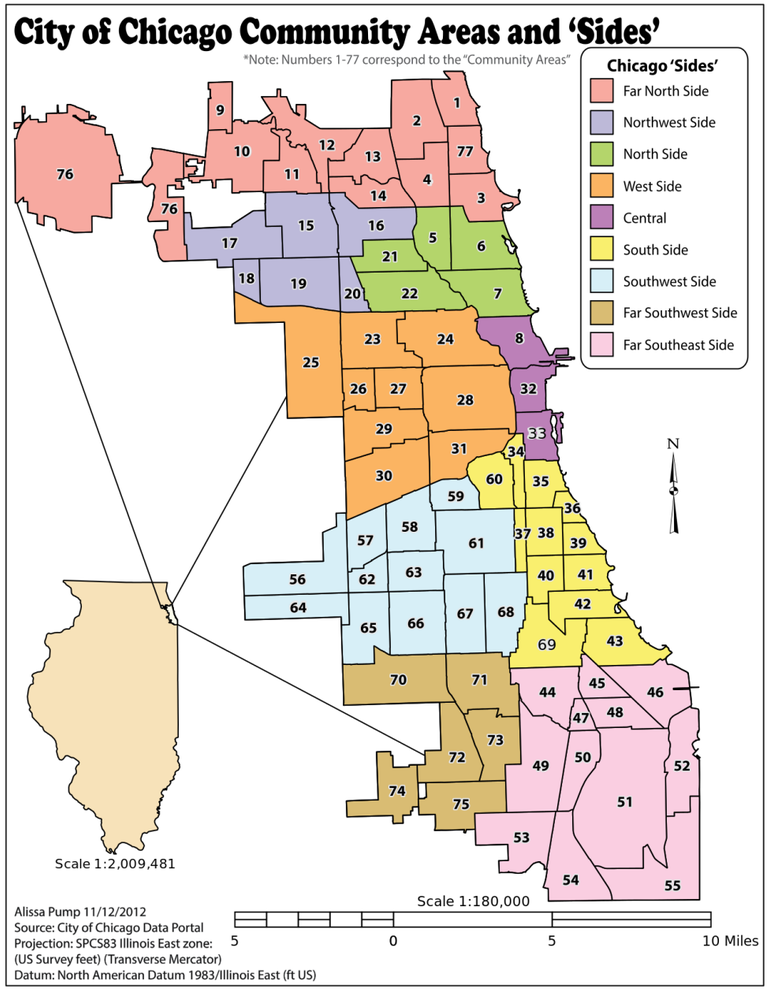
A	B	C
A1	B1	C1
A2	B2	C2
A3	B3	C3



C	D
C1	D1
C1	D2
C1	D3
C2	D4

One-To-Many = Every row in left table is related to one or more rows in the right table

One-to-many example



One-to-many example

```
licenses = pd.read_csv('Business_Licenses.csv')
print(licenses.head())
print(licenses.shape)
```

```
  account  ward  aid  business          address           zip
0 307071   3     743  REGGIE'S BAR...   2105 S STATE ST   60616
1 10       10    829  HONEYBEERS        13200 S HOUS...   60633
2 10002    14    775  CELINA DELI       5089 S ARCHE...   60632
3 10005    12    nan  KRAFT FOODS ...   2005 W 43RD ST    60609
4 10044    44    638  NEYBOUR'S TA...   3651 N SOUTH...   60613
(10000, 6)
```

One-to-many example

	ward	alderman	address	zip
0	1	Proco "Joe" ...	2058 NORTH W...	60647
1	2	Brian Hopkins	1400 NORTH ...	60622
2	3	Pat Dowell	5046 SOUTH S...	60609
3	4	William D. B...	435 EAST 35T...	60616
4	5	Leslie A. Ha...	2325 EAST 71...	60649

	account	ward	aid	business	address	zip
0	307071	3	743	REGGIE'S BAR...	2105 S STATE ST	60616
1	10	10	829	HONEYBEERS	13200 S HOUS...	60633
2	10002	14	775	CELINA DELI	5089 S ARCHE...	60632
3	10005	12	nan	KRAFT FOODS ...	2005 W 43RD ST	60609
4	10044	44	638	NEYBOUR'S TA...	3651 N SOUTH...	60613

One-to-many example

suffixes=('_ward', '_cen'): If there are columns with the same name (other than ward), they'll be renamed with these suffixes to avoid conflicts:
From wards, columns will get '_ward' added.
From census, columns will get '_cen' added

```
ward_licenses = wards.merge(licenses, on='ward', suffixes=('_ward','_lic'))
print(ward_licenses.head())
```

```
  ward  alderman          address_ward     zip_ward  account  aid  business          address_lic
0 1     Proco "Joe" ...   2058 NORTH W...  60647     12024    nan  DIGILOG ELEC...   1038 N ASHLA...
1 1     Proco "Joe" ...   2058 NORTH W...  60647     14446    743  EMPTY BOTTLE...   1035 N WESTE...
2 1     Proco "Joe" ...   2058 NORTH W...  60647     14624    775  LITTLE MEL'S...   2205 N CALIF...
3 1     Proco "Joe" ...   2058 NORTH W...  60647     14987    nan  MR. BROWN'S ...   2301 W CHICA...
4 1     Proco "Joe" ...   2058 NORTH W...  60647     15642    814  Beat Kitchen      2000-2100 W ...
```

One-to-many example

```
print(wards.shape)
```

```
(50, 4)
```

```
print(ward_licenses.shape)
```

```
(10000, 9)
```

Let's practice!

JOINING DATA WITH PANDAS

Part-03

Merging multiple DataFrames

JOINING DATA WITH PANDAS

Aaren Stubberfield
Instructor

datacamp

Merging multiple tables

A	B	C
A1	B1	C1
A2	B2	C2
A3	B3	C3

C	D
C1	D1
C2	D2
C3	D3

A	B	C
A1	B1	C1
A2	B2	C2
A3	B3	C3

C	E
C1	E1
C2	E2
C3	E3

E	F	G
E1	F1	G1
E2	F2	G2
E3	F3	G3

Remembering the licenses table

```
print(licenses.head())
```

```
   account  ward  aid  business          address           zip
0 307071    3     743  REGGIE'S BAR...   2105 S STATE ST   60616
1 10        10    829  HONEYBEERS        13200 S HOUS...   60633
2 10002     14    775  CELINA DELI       5089 S ARCHE...   60632
3 10005     12    nan  KRAFT FOODS ...   2005 W 43RD ST    60609
4 10044     44    638  NEYBOUR'S TA...   3651 N SOUTH...   60613
```

Remembering the wards table

```
print(wards.head())
```

```
   ward  alderman          address           zip
0  1     Proco "Joe" ...   2058 NORTH W...   60647
1  2     Brian Hopkins     1400 NORTH  ...   60622
2  3     Pat Dowell        5046 SOUTH S...   60609
3  4     William D. B...   435 EAST 35T...   60616
4  5     Leslie A. Ha...   2325 EAST 71...   60649
```

Review new data

```
grants = pd.read_csv('Small_Business_Grant_Agreements.csv')
print(grants.head())
```

```
   address           zip    grant       company
0 1000 S KOSTN...  60624  148914.50  NATIONWIDE F...
1 1000 W 35TH ST   60609  100000.00  SMALL BATCH,...
2 1000 W FULTO...  60612  34412.50   FULTON MARKE...
3 10008 S WEST...  60643  12285.32   LAW OFFICES ...
4 1002 W ARGYL...  60640  28998.75   MASALA'S IND...
```

Tables to merge

	address	zip	grant	company
0	1031 N CICER...	60651	150000.00	1031 HANS LLC
1	1375 W LAKE ST	60612	150000.00	1375 W LAKE ...
2	1800 W LAKE ST	60612	47700.00	1800 W LAKE LLC
3	4311 S HALST...	60609	87350.63	4311 S. HALS...
4	1747 W CARRO...	60612	50000.00	ACE STYLINE ...

	account	ward	aid	business	address	zip
0	307071	3	743	REGGIE'S BAR...	2105 S STATE ST	60616
1	10	10	829	HONEYBEERS	13200 S HOUS...	60633
2	10002	14	775	CELINA DELI	5089 S ARCHE...	60632
3	10005	12	nan	KRAFT FOODS ...	2005 W 43RD ST	60609
4	10044	44	638	NEYBOUR'S TA...	3651 N SOUTH...	60613

Theoretical merge

```
grants_licenses = grants.merge(licenses, on='zip')
print(grants_licenses.loc[grants_licenses['business']=="REGGIE'S BAR & GRILL",
                          ['grant','company','account','ward','business']])
```

```
   grant      company          account  ward  business
0  136443.07  CEDARS MEDIT...  307071   3     REGGIE'S BAR...
1  39943.15   DARRYL & FYL...  307071   3     REGGIE'S BAR...
2  31250.0    JGF MANAGEMENT   307071   3     REGGIE'S BAR...
3  143427.79  HYDE PARK AN...  307071   3     REGGIE'S BAR...
4  69500.0    ZBERRY INC       307071   3     REGGIE'S BAR...
```

Single merge

```
grants.merge(licenses, on=['address','zip'])
```

	address	zip	grant	company	account	ward	aid	business
0	1020 N KOLMA...	60651	68309.8	TRITON INDUS...	7689	37	929	TRITON INDUS...
1	10241 S COMM...	60617	33275.5	SOUTH CHICAG...	246598	10	nan	SOUTH CHICAG...
2	11612 S WEST...	60643	30487.5	BEVERLY RECO...	3705	19	nan	BEVERLY RECO...
3	1600 S KOSTN...	60623	128513.7	CHARTER STEE...	293825	24	nan	LEELO STEEL,...
4	1647 W FULTO...	60612	5634.0	SN PECK BUIL...	85595	27	673	S.N. PECK BU...

Merging multiple tables

line break

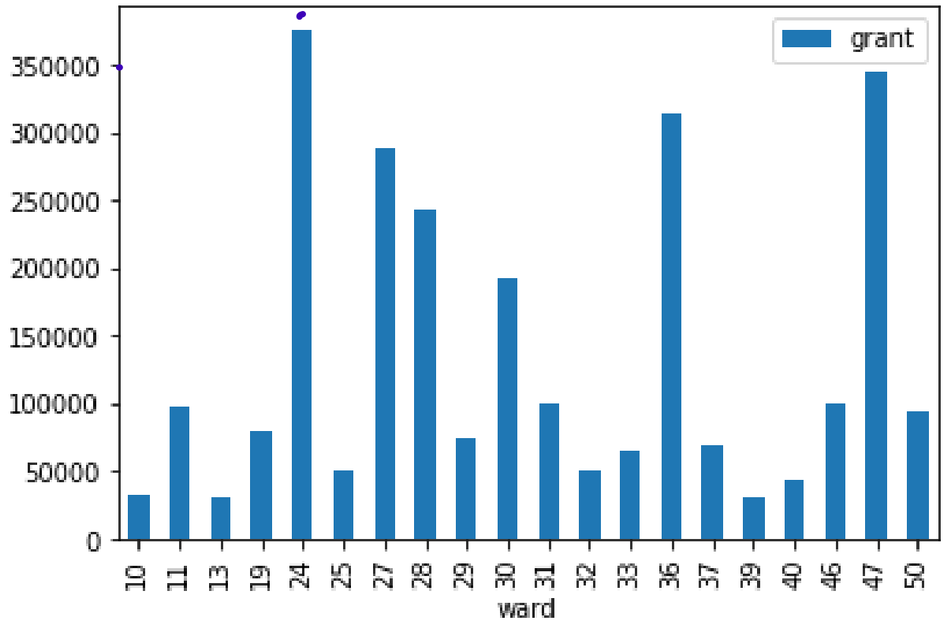
```
grants_licenses_ward = grants.merge(licenses, on=['address','zip']) \
                        .merge(wards, on='ward', suffixes=('_bus','_ward'))
grants_licenses_ward.head()
```

	address_bus	zip_bus	grant	company	account	ward	aid	business	alderma
0	1020 N KOLMA...	60651	68309.8	TRITON INDUS...	7689	37	929	TRITON INDUS...	Emma M.
1	10241 S COMM...	60617	33275.5	SOUTH CHICAG...	246598	10	nan	SOUTH CHICAG...	Susan S
2	11612 S WEST...	60643	30487.5	BEVERLY RECO...	3705	19	nan	BEVERLY RECO...	Matthew
3	3502 W 111TH ST	60655	50000.0	FACE TO FACE...	263274	19	704	FACE TO FACE	Matthew
4	1600 S KOSTN...	60623	128513.7	CHARTER STEE...	293825	24	nan	LEELO STEEL,...	Michael

Results

it sums every col possible

```
import matplotlib.pyplot as plt
grant_licenses_ward.groupby('ward').agg('sum').plot(kind='bar', y='grant')
plt.show()
```



Merging even more...

Three tables:

```
df1.merge(df2, on='col') \
    .merge(df3, on='col')
```

Four tables:

```
df1.merge(df2, on='col') \
    .merge(df3, on='col') \
    .merge(df4, on='col')
```

Let's practice!

JOINING DATA WITH PANDAS