C: Basic database joins

Combining datasets by matching fields

- Terminology: **joining** or **merging** data

Example:

Dataset 1: State FIPS codes and names (names data)

FIPS	Name
09	Connecticut
23	Maine
25	Massachusetts
33	New Hampshire
44	Rhode Island
…	…

- Will often refer to as a **table** or **dataframe**

Dataset 2: FIPS codes and populations (pop data)

FIPS	Pop
25	6830193
09	3581504
33	1343622
23	1332813
44	1056611
50	624977

…	…

Want to add matching population to the names data:

FIPS	Name	Pop
09	Connecticut	3581504
23	Maine	1332813
25	Massachusetts	6830193
33	New Hampshire	1343622
44	Rhode Island	1056611
50	Vermont	624977
…	…	…

Name:	From name data
Pop:	From pop data
FIPS:	Shared by both

Basic approach:

1. Read names data into a **list** of dictionaries:

```
[
  { 'FIPS': '09', 'Name': 'Connecticut' },
  { 'FIPS': '23', 'Name': 'Maine' },
   ....
]
```

2. Use that to build a **dictionary** of dictionaries:

- one entry per state
- key: FIPS code
- value: state's entry

Referred to as **indexing the data**

Result:

```
{
   '09': { 'FIPS': '09', 'Name': 'Connecticut' },
   '23': { 'FIPS': '23', 'Name': 'Maine' },
   ....
}
```

3. Set up a similar dictionary for the pop data

Result:

```
{
   '25': { 'FIPS': '25', 'Pop': 6830193 },
   '09': { 'FIPS': '09', 'Pop': 3581504 },
   ....
}
```

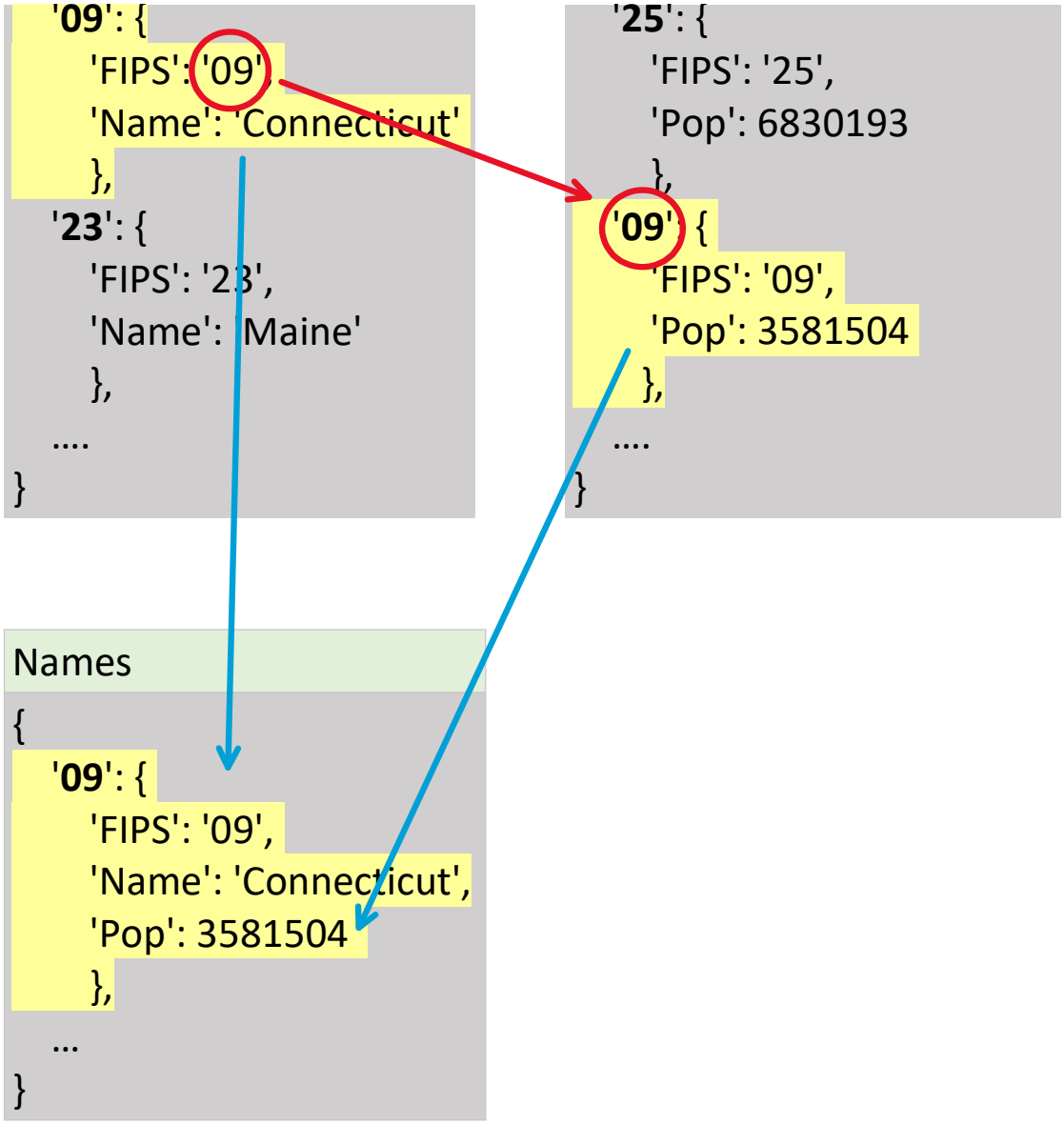
3. Loop through the names data
   - get the record's FIPS code
   - look up pop record for that code
   - add population to the names record

Names

```
{
   '09': {
      'FIPS': '09'
```

Pop

```
{
   '25': {
      'FIPS': '25',
```



Types of joins:

One-to-one (1:1) join:

- Each record in names matches exactly one record in pop

Many-to-one (m:1) and one-to-many (1:m):

- Many records in one dataset link to a single target record
- Example: many counties match to each state

Switch to G09 demo.py