In [11]:

```
from pathlib import Path
import pandas as pd
import json
import os

from tinydb import TinyDB

current_dir = Path(os.getcwd()).absolute()
results_dir = current_dir.joinpath('results')
kv_data_dir = results_dir.joinpath('kvdb')
kv_data_dir.mkdir(parents=True, exist_ok=True)


class DocumentDB(object):
    def __init__(self, db_path):

        people_json = open('results/kvdb/people.json')
        visited_json = open('results/kvdb/visited.json')
        sites_json = open('results/kvdb/sites.json')
        measurements_json = open('results/kvdb/measurements.json')

        self._db_path = Path(db_path)
        self._db = None
        self._person_lookup = json.load(people_json)
        self._site_lookup = json.load(sites_json)
        self._visit_lookup = json.load(visited_json)
        self._measurement_lookup = json.load(measurements_json)
        self._load_db()

    def _get_site(self, site_id):
        return self._site_lookup[str(site_id)]

    def _get_measurements(self, person_id):
        measurements = []
        df = pd.read_csv('measurements.csv')
        x = {}
        for visit_id, group_df in df.groupby(['visit_id', 'person_id', 'quar
            x[str(visit_id)] = group_df.to_dict(orient='records')[0]
        for k, v in x.items():
            val = v
            valname = val['person_id']
            if valname == str(person_id):
                measurements.append(v)
            return measurements

    def _get_visit(self, visit_id):
        visit = pd.read_csv('visited.csv')
        visit = visit.loc[visit['visit_id'] == visit_id]
        site_id = visit['site_id']
        site_id  = site_id.to_string()
        site = pd.read_csv('site.csv')
        xs = site.loc[site['site_id'] == site_id]
        return visit
```

```
    def _load_db(self):
        self._db = TinyDB(self._db_path)
        persons = self._person_lookup.items()
        recs = {}
        for person_id, record in persons:
            measurements = self._get_measurements(person_id)
            visit_ids = set([measurement['visit_id'] for measurement in meas
            visits = []
            for visit_id in visit_ids:
                visit = self._get_visit(visit_id)
                visit['measurements'] = [measurement for measurement in meas
                    if visit_id == measurement['visit_id']]
                visits.append(visit)
                record['visits'] = str(visits)
                recs['person_id'] = record
                self._db.insert(record)

        site = pd.read_csv('site.csv')
        site = site.set_index('site_id')

        dict1 = site.to_dict("index")

        visit = pd.read_csv('visited.csv')
        visit = visit.set_index('visit_id')
        visit = visit.fillna(method='ffill')

        vals = []

        for x in visit['site_id']:
            pair = {}
            y = dict1[x]
            pair[x] = y
            vals.append(pair)

        visit['site_id'] = vals

        meas = pd.read_csv('measurements.csv')
        meas2 = meas.groupby('visit_id')[['quantity',
                'reading']].apply(lambda x: x.to_dict(orient='index')).to_di

        measures = []

        for k, v in meas2.items():
            measures.append(v)

        visit['readings'] = measures

        person = pd.read_csv('person.csv')

        people = []
        ids = []

        for pers in person['person_id']:
            for index, row in meas.iterrows():
                if row['visit_id'] not in ids:
                    ids.append(row['visit_id'])
```

```
                    people.append(row['person_id'])

        v = visit.reset_index()
        v['person_id'] = people
        v = v.groupby('person_id')[['visit_id','visit_date', 'site_id', 'rea

        with open('results/patient-info.json', 'w') as f:
            json.dump(v, f)
```

In [12]:

```
db_path = results_dir.joinpath('patient-info.json')
if db_path.exists():
    os.remove(db_path)

db = DocumentDB(db_path)
```