In [1]:
```
# load libraries
from pathlib import Path
import json
import os
from tinydb import TinyDB
```

In [2]:
```
# set file directories
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)
```

In [3]:
```
def _load_json(json_path):
    '''loads data from .json file'''
    with open(json_path) as f:
        return json.load(f)
```

In [4]:
```
class DocumentDB(object):
    def __init__(self, db_path):
        ## You can use the code from the previous exmaple if you would like
        people_json = kv_data_dir.joinpath('people.json')
        visited_json = kv_data_dir.joinpath('single_key_visited.json')
        sites_json = kv_data_dir.joinpath('sites.json')
        measurements_json = kv_data_dir.joinpath('measurements.json')
        self._db_path = Path(db_path)
        self._db = None

        ## TODO
        # load json data:
        self._person_lookup = _load_json(people_json)
        self._visit_lookup = _load_json(visited_json)
        self._site_lookup = _load_json(sites_json)
        self._measurements_lookup = _load_json(measurements_json)

        self._load_db()

    def _get_site(self, site_id):
        '''returns info about individual site'''
        return self._site_lookup[str(site_id)]

    def _get_measurements(self, person_id):
        '''returns list of dictionaries of measurements associated with person_ic
        measurements = []
        measurements.extend([
            values for values in self._measurements_lookup.values()
            if str(values['person_id']) == str(person_id)
        ])
        return measurements

    def _get_visit(self, visit_id):
        '''returns visit information about a specified site visit_id'''
        # retrieve corresponding row of info from visit:
        visit = self._visit_lookup.get(str(visit_id)) # {'visit_id':, 'site_id':,

        ## TODO:
        site_id = visit['site_id'] # extract site_id from visit info

        ## TODO:
        site = self._get_site(site_id) # retrieve info about site

        visit['site'] = site # append site info to visit info
        return visit

    def _load_db(self):
        self._db = TinyDB(self._db_path)
        # return dict_items of people_json:
        persons = self._person_lookup.items()
        for person_id, record in persons:
            # return individual's list of records:
            measurements = self._get_measurements(person_id)
            # extract set of unique visit_id's from id's in list of measurements
            visit_ids = set([measurement['visit_id'] for measurement in measureme
            visits = []
```

```
            for visit_id in visit_ids: # iterate through set of individual's visi
                visit = self._get_visit(visit_id) # returns info from visit
                # add measurement info from visit:
                visit['measurements'] = [
                    measurement for measurement in measurements
                    if visit_id == measurement['visit_id']
                ]
                visits.append(visit)
            record['visits'] = visits # add visit info to record
            self._db.insert(record)
```

In [5]:
```
db_path = results_dir.joinpath('patient-info.json')
if db_path.exists():
    os.remove(db_path)

db = DocumentDB(db_path)
```

In [ ]:
```

```