

In [1]:

```

from pathlib import Path
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)

def _load_json(json_path):
    with open(json_path) as f:
        return json.load(f)

class DocumentDB(object):
    def __init__(self, db_path):
        ## You can use the code from the previous example if you would like
        people_json = kv_data_dir.joinpath('people.json')
        visited_json = kv_data_dir.joinpath('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
        self._person_lookup = _load_json(people_json)
        self._visited_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):
        return self._site_lookup[str(site_id)]

    def _get_measurements(self, person_id):
        measurements = []
        for values in self._measurements_lookup.values():
            measurements.extend([value for value in values if str(value['person_id']) == person_id])
        return measurements

    def _get_visit(self, visit_id):
        visit = self._visit_lookup.get(str(visit_id))
        site_id = str(visit['site_id'])
        site = self._site_lookup[site_id]
        visit['site'] = site
        return visit

    def _load_db(self):
        self._db = TinyDB(self._db_path)
        persons = self._person_lookup.items()

        for person_id, record in persons:
            measurements = self._get_measurements(person_id)
            visit_ids = set([measurement['visit_id'] for measurement in measurements])
            visits = []
            for visit_id in visit_ids:
                visit = self._get_visit(visit_id)
                visit['measurements'] = [
                    measurement for measurement in measurements
                    if visit_id == measurement['visit_id']
                ]
                visits.append(visit)

```

```
record['visits'] = visits  
self._db.insert(record)
```