

```
In [6]: 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)
```

```
In [7]: def _load_json(json_path):
    with open(json_path) as f:
        return json.load(f)
```

```
In [10]: class DocumentDB(object):
    def __init__(self, db_path):
        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.people_dict = _load_json(people_json)
        self.visited_dict = _load_json(visited_json)
        self.sites_dict = _load_json(sites_json)
        self.measurements_dict = _load_json(measurements_json)
        self._load_db()

    def _load_db(self):
        self._db = TinyDB(self._db_path)
        # Create records by people dictionary
        for people_k, people_v in self.people_dict.items():

            # Create a list of visits
            people_v['visits'] = []
            for visited_k, visited_v in self.visited_dict.items():
                for sites_k, sites_v in self.sites_dict.items():
                    # Matching up site_id by sites and visits. Then storing in visited
                    if sites_v['site_id'] == visited_v['site_id']:
                        visited_v['site'] = sites_v

            # Create measurements List
            visited_v['measurements'] = []
            for measurements_k, measurements_v in self.measurements_dict.items():
                # Matching visit_id and person_id with the measurements equ
                # and then add them to measurements_v
                if measurements_v['visit_id'] == visited_v['visit_id']
                    and measurements_v['person_id'] == people_v['person_id']:
                    visited_v['measurements'].append(measurements_v)
            # If there are measurements, add it to visit using people_v vis
            if len(visited_v['measurements']) != 0:
                people_v['visits'].append(visited_v)
```

```
# Once person record is complete save it to db  
self._db.insert(people_v)
```

```
In [11]: db_path = results_dir.joinpath('patient-info.json')  
         if db_path.exists():  
             os.remove(db_path)  
  
         db = DocumentDB(db_path)
```

```
In [ ]:
```

```
In [ ]:
```