```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": 5,
   "metadata": {
    "pycharm": {
     "name": "#%%\n"
    }
   },
   "outputs": [],
   "source": [
    "import json\n",
    "from pathlib import Path\n",
    "import os\n",
    "\n",
    "import pandas as pd\n",
    "import s3fs\n",
    "\n",
    "\n",
    "def read_cluster_csv(file_path, endpoint_url='https://storage.budsc.midwest-datascience.com'):\n",
    "    s3 = s3fs.S3FileSystem(\n",
    "        anon=True,\n",
    "        client_kwargs={\n",
    "            'endpoint_url': endpoint_url\n",
    "        }\n",
    "    )\n",
    "    return pd.read_csv(s3.open(file_path, mode='rb'))\n",
    "\n",
    "current_dir = Path(os.getcwd()).absolute()\n",
    "results_dir = current_dir.joinpath('results')\n",
    "kv_data_dir = results_dir.joinpath('kvdb')\n",
    "kv_data_dir.mkdir(parents=True, exist_ok=True)\n",
    "\n",
    "people_json = kv_data_dir.joinpath('people.json')\n",
    "visited_json = kv_data_dir.joinpath('visited.
```

```
36 json')\n",
37     "sites_json = kv_data_dir.joinpath('sites.json')\
   n",
38     "measurements_json = kv_data_dir.joinpath('
   measurements.json')"
39    ]
40   },
41   {
42    "cell_type": "code",
43    "execution_count": 2,
44    "metadata": {
45     "pycharm": {
46      "name": "#%%\n"
47     }
48    },
49    "outputs": [],
50    "source": [
51     "class KVDB(object):\n",
52     "    def __init__(self, db_path):\n",
53     "        self._db_path = Path(db_path)\n",
54     "        self._db = {}\n",
55     "        self._load_db()\n",
56     "\n",
57     "    def _load_db(self):\n",
58     "        if self._db_path.exists():\n",
59     "            with open(self._db_path) as f:\n",
60     "                self._db = json.load(f)\n",
61     "\n",
62     "    def get_value(self, key):\n",
63     "        return self._db.get(key)\n",
64     "\n",
65     "    def set_value(self, key, value):\n",
66     "        self._db[key] = value\n",
67     "\n",
68     "    def save(self):\n",
69     "        with open(self._db_path, 'w') as f:\n",
70     "            json.dump(self._db, f, indent=2)"
71    ]
72   },
73   {
74    "cell_type": "code",
```

```
 75    "execution_count": 3,
 76    "metadata": {
 77     "pycharm": {
 78      "name": "#%%\n"
 79     }
 80    },
 81    "outputs": [],
 82    "source": [
 83     "def create_sites_kvdb():\n",
 84     "    db = KVDB(sites_json)\n",
 85     "    df = pd.read_csv('C:/Users/taylo/OneDrive/
    Documents/dsc650/data/external/tidynomicon/site.csv
    ')\n",
 86     "    for site_id, group_df in df.groupby('
    site_id'):\n",
 87     "        db.set_value(site_id, group_df.to_dict(
    orient='records')[0])\n",
 88     "    db.save()\n",
 89     "\n",
 90     "\n",
 91     "def create_people_kvdb():\n",
 92     "    db = KVDB(people_json)\n",
 93     "    ## TODO: Implement code\n",
 94     "    df = pd.read_csv('C:/Users/taylo/OneDrive/
    Documents/dsc650/data/external/tidynomicon/person.
    csv')\n",
 95     "    for person_id, group_df in df.groupby('
    person_id'):\n",
 96     "        db.set_value(person_id, group_df.
    to_dict(orient='records')[0])\n",
 97     "    db.save()\n",
 98     "\n",
 99     "\n",
100     "def create_visits_kvdb():\n",
101     "    db = KVDB(visited_json)\n",
102     "    ## TODO: Implement code\n",
103     "    df = pd.read_csv('C:/Users/taylo/OneDrive/
    Documents/dsc650/data/external/tidynomicon/visited.
    csv')\n",
104     "    for index, row in df.iterrows():\n",
105     "        key = str(str(row['visit_id
```

```
105 ']) + ', ' + str(row['site_id']))\n",
106     "        value = dict(\n",
107     "            visit_id=row['visit_id'],\n",
108     "            site_id=row['site_id'],\n",
109     "            visit_date=row['visit_date']\n",
110     "        )\n",
111     "        db.set_value(key, value)\n",
112     "    db.save()\n",
113     "\n",
114     "\n",
115     "def create_measurements_kvdb():\n",
116     "    db = KVDB(measurements_json)\n",
117     "    ## TODO: Implement code\n",
118     "    df = pd.read_csv('C:/Users/taylo/OneDrive/
    Documents/dsc650/data/external/tidynomicon/
    measurements.csv')\n",
119     "    for index, row in df.iterrows():\n",
120     "        key = str(str(row['visit_id
    ']) + ', ' + str(row['person_id']) + ', ' + str(row
    ['quantity']))\n",
121     "        value = dict(\n",
122     "            visit_id=row['visit_id'],\n",
123     "            person_id=row['person_id'],\n",
124     "            quantity=row['quantity'],\n",
125     "            reading=row['reading']\n",
126     "        )\n",
127     "        db.set_value(key, value)\n",
128     "    db.save()"
129    ]
130   },
131   {
132    "cell_type": "code",
133    "execution_count": 4,
134    "outputs": [],
135    "source": [
136     "create_sites_kvdb()\n",
137     "create_people_kvdb()\n",
138     "create_visits_kvdb()\n",
139     "create_measurements_kvdb()"
140    ],
141    "metadata": {
```

```
    "collapsed": false,
    "pycharm": {
     "name": "#%%\n"
    }
   }
  }
 ],
 "metadata": {
  "kernelspec": {
   "display_name": "Python 3",
   "language": "python",
   "name": "python3"
  },
  "language_info": {
   "codemirror_mode": {
    "name": "ipython",
    "version": 3
   },
   "file_extension": ".py",
   "mimetype": "text/x-python",
   "name": "python",
   "nbconvert_exporter": "python",
   "pygments_lexer": "ipython3",
   "version": "3.8.3"
  }
 },
 "nbformat": 4,
 "nbformat_minor": 4
}
```