Assignment 2.

Problem 0

For this assignment I implemented the database as one table using a list of lists. Some performance improvement could be realized by splitting the database in separate tables, however the improvements would be modest given the small size of the database: there are fewer than 2000 jobs listed on NYC Open Data website.

I examined the 1907 rows of data from NYC Open Data to see where creating multiple tables can save space. Because of python’s dynamic typing, using default data types does not yield maximum space savings. For example a typical value of Agency is `'DEPT OF ENVIRONMENT PROTECTION'`, which on my system python stores in a string of 67 bytes. In the first 1907 rows there were 41 departments, so if we stored Agency as a separate table, its foreign key could be as small as an unsigned 1 byte integer, but, on this system, the size of a python integer is 24 bytes, so we are not realizing huge savings by using a separate table. We can further improve memory use by creating separate tables for `Business Title` and `Civil Service Title`, but these savings will not be as great as with Agency, since there are many more unique values in these fields. `Salary Frequency` has only 3 unique values in the first 1907 rows, so it warrants a separate table or some kind of enumeration implementation. A separate table for addresses could also be useful.

Implementation I coded:

Table 1.

```
Job ID: string, primary key
Agency: string
# Of Positions: string
Business Title: string
Civil Service Title: string
Salary Range From: string
Salary Range To: string
Salary Frequency: string
Work Location: string
Division/Work Unit: string
Job Description: string
Minimum Qual Requirements: string
Preferred Skills: string
Additional Information: string
Posting date: string
```

Implementation that would improve efficiency:

Table 1.

```
Job ID: int, primary key
Agency: short, foreign key
# Of Positions: int
Business Title: string
Civil Service Title: string
```

```
Salary Range From: float
Salary Range To: float
Salary Frequency: short, foreign key
Work Location: string
Division/Work Unit: string
Job Description: string
Minimum Qual Requirements: string
Preferred Skills: string
Additional Information: string
Posting date: datetime
```

Table 2.

```
Agency ID: short, primary key
Agency: string
```

Table 3.

```
Freq ID: short, primary key
Salary Frequency: string
```