

Assignment 2

Relational Database

PROBLEMS

0. Describe your strategy to store the database. Which files are you going to generate to keep your database? Why? Try to relate to the concepts presented in class.

ANSWERS:

Main ideas from given problems

Main ideas that can be inferred from the problems and sample database:

- Job ID is the only unique ID among the records. One job post must have one single and unique ID. Therefore, Job ID has to be the main identifier attribute to which other attributes are referring to.
- All attributes (except Job ID) has relatively equal of importance, meaning:
 - a. All executable commands/functions (e.g. Select, Find, View) have to be applicable to any attributes.
 - b. When reading from database, most functions will only need to get one attribute value or the whole records for specified reference ID (Job ID)

Therefore, designing *schema* that consists of a group of attributes is **not** needed (creating a schema and view per several attributes or breaking main data into more than 1 table will overcomplicate the design, since none of the specified in the problems require the database application to provide set of attributes).

- Schema is needed in some functions such as *find*, but handling it with python dictionary (*logical schema*) during run time is considered is enough.
- For storage and indexing, some attributes values in each record is expected to have repetitive values, e.g., Agency, Salary Frequency, Location, Minimum

qualification requirement. Reference database (ID-indexing database) for these attributes so that in main database we do not need to store the whole string, hence save the cost of storage when storing large data.

- The database design is assuming that there will be no addition of attributes name. Attribute name is designed to be fixed in the same order as the problems:

Job ID | Agency | # Of Positions | Business Title | Civil Service Title | Salary Range From| Salary Range To| Salary Frequency |Work Location | Division /Work Unit | Job Description |Minimum Qual Requirements | Preferred Skills | Additional Information | Posting Date

Database Design

The database design is divided into two main levels:

1. Main database (Jobs.db)

Main database consists of all the attributes, sorted in the same order as stated in the problems. However, for some repetitive values (Agency, Salary Frequency, Location, Minimum qualification requirement) the storage indexing id is stored instead of the real value. The example of database output:

```
1|1|1|Assistant|AGENCY ATTORNEY|66970|92000|0|1|General Counsel's Office|Responsibilities|1| |
|08/27/2014 00:00:00

3|2|1|College Aide|COLLEGE AIDE|14|17|1|0|ITS|COLLEGE|3|Preference| |07/14/2014 00:00:00

2|0|1|SCIENTIST|SCIENTIST|71220|100000|0|0|Off|The Office|0|Candidates must have| |07/03/2014 00:00:00

5|3|1|Assessor|ASSESSOR|40623|100000|0|3|Tax|supervision|2|eligible| |11/26/2013 00:00:00

4|0|1|PROCUREMENT ANALYST|MANAGER|49492|90000|0|2|Procurement|Financial|2| |P468|08/04/2014 00:00:00
```

2. Reference database (ID-mapping database)

a. Salary frequency ID database (frequency.db)

This database stores every single type with its respective ID. Example of the database contents:

```
Hourly|1
Annual|0
```

b. Agency ID database (agency.db)

This database stores every single unique name of agency with its respective ID. Example of the database contents:

```
HRA|1  
DESIGN|3  
LAW DEPARTMENT|0  
DEPARTMENT OF BUILDINGS|2
```

c. Location ID database (location.db)

This database stores every single unique name of location with its respective ID. Example of the database contents:

```
NY|1  
Brooklyn|3  
N.Y.|2  
Queens|0
```

d. Minimum Qualification ID database (requirement.db)

This database stores every single unique name of agency with its respective ID. Example of the database contents:

```
Assignment|1  
Admission|2  
baccalaureate|0  
Level I|3
```