Information Systems, Fall 2025

Homework 4

Due Sunday, November 16, 11:59 PM

This homework refers to a class example from October 27 and is about semi-structured data**.**

Below is the JSON database used for the example in question:

{

"1": {"web": "Caesar@spqr.com", "name":"Caesar", "dob":"long ago", "occupation": {"name":"Roman General", "income": "Heaps of Cestertii"}, "known_for": ["conquered Gaul", "Won a Civil War", "Wrote a book about himself"]},

"2":{"web": "https:///en.wikipedia.org//wiki//Caesar_salad", "name":"Caesar", "dob":"1924", "occupation": "Salad"},

"3": {"web": "Augustus@spqr.com", "name":"Octavian", "dob":"Between Caesar and Nero", "occupation":{"name":"Roman Princeps", "income": "Heaps of Cestertii"}, "known_for": ["Founded the Empire", "Won a Civil War", "Rebuilt Rome"]},

"4": {"web": "Nero@spqr.com", "name":"Nero", "dob":"also long ago", "occupation": {"name":"Roman Princeps", "income": "Heaps of Cestertii"}, "known_for": ["Burned Rome", "Won Olympics"]},

"5": {"web": "@elonmusk", "name":"Elon", "dob":"recently", "occupation": {"name":"Microblogger", "income": "Jigabytes of Dogecoins"}, "known_for": ["Founded Tesla", "Founded SpaceX", "Talks too much"]},

"6": {"web": "drake@goldenhind.com", "name":"Francis", "dob":"not so long ago", "occupation":{"name":"Privateer", "income": "Ships of Silver"}, "known_for": ["Looted Spanish colonies", "Defeated the Spanish Armada", "Clearly didn't like Spain", "Second to circumnavigate the Earth"]},

"7": {"web": "https:///www.wbkidsgo.com//looney-tunes", "name":"Bugs", "occupation": {"name":"Bunny", "income": "Bunches of Carrots"}, "known_for": ["Star at Hollywood", "Won an Oscar"]}

}

Question 1 (10 pts):

In the class example we couldn't list names of all Romans. This is because we can't properly loop over names of the occupations of each object in the database.

Code that didn't work:

```
#3) List names of all Romans

for (smth in Rich_Famous)

{

 if (grepl("Roman", smth$occupation$name))   #TRUE if "Roman" is a substring of occupation name

 {print (smth$name)}

}

# HAHA, doesn't work :) We'll try to fix it in the Homework 4 :)
```

a. Explain what in the data prevents the loop from working? (4 pts)

Part A:

The reason why the loop isn't working is because of the inconsistencies in the data. More specifically, the second instance of the occupation key is not an object, while all of the other occupation keys for the other objects are objects. Also, this second object doesn't include a 'known for' array like the other objects. This prevents the loop from working because we assume all occupation keys are objects.

b. Suggest how to fix the **data** *(not the code!)* so that the loop will work as intended (copy/paste relevant object(s) and highlight your change). (6 pts)

Part B:

We can fix this data by turning the second instance of the occupation key into an object. We can add name and income properties to this occupation key. Furthermore, we can add the 'known for' array to the second object directly after the occupation object. We can make it like the image below.

```
"2": {

  "web": "https:///en.wikipedia.org//wiki//Caesar_salad",
  "name": "Caesar",
  "dob": "1924",
  "occupation": {
    "name": "Salad Chef",
    "income": "Fruits and Vegetables"
  },

  "known_for": [
    "Invented Caesar Salad",
    "Popularized croutons",
    "Famous dressing recipe"
  ]
},

```

Question 2 (5 pts):

Our database isn't very consistent. By common logic, it contains 3 entity types: real people, fictional characters, and food. Suggest how to change it to introduce this distinction (by adding attributes or otherwise rearranging it). Again, copy-paste it below, and highlight your changes. You can show it just for one entity in each category.

We could add an entity_type field to each object. Depending on the entity type of the object, the field can have either a person key for real people, a character key for fictional characters, and a food key for food. The image below is a good example of how we can do it.

```
{

  "1": {

    "entity_type": "person",
    "web": "Caesar@spqr.com",
    "name": "Caesar",
    "dob": "long ago",
    "occupation": {
      "name": "Roman General",
      "income": "Heaps of Cestertii"
    },

    "known_for": [
      "conquered Gaul",
      "Won a Civil War",
      "Wrote a book about himself"
    ]
  },

  "2": {

    "entity_type": "food",
    "web": "https:///en.wikipedia.org//wiki//Caesar_salad",
    "name": "Caesar",
    "dob": "1924",
    "occupation": {
      "name": "Salad Chef",
      "income": "Fruits and Vegetables"
    },

    "known_for": [
      "Invented Caesar Salad",
      "Popularized croutons",
      "Famous dressing recipe"
    ]
  },

  "7": {

    "entity_type": "character",
    "web": "https:///www.wbkidsgo.com//looney-tunes",
    "name": "Bugs",
    "occupation": {
      "name": "Bunny",
      "income": "Bunches of Carrots"
    },

    "known_for": [
      "Star at Hollywood",
      "Won an Oscar"
    ]
  }

}
```

Question 3 (10 pts):

Now that we have different categories of entities, we can take some further steps. Not all existing attributes are relevant for all categories in the updated database. Assume that we want to be consistent and have the same attributes for objects within a category.

a. Which of the existing attributes do you think objects in each category (people, characters, food) should retain as relevant? (5 pts)

Part A:

Relevant Attributes for People: entity_type, name, web, dob, occupation, known for

Relevant Attributes for Characters: entity_type, name, web, known for, role (formerly occupation)

Relevant Attributes for Food: entity_type, name, web, ingredients/description (formerly known for)

b. What attribute changes or additions can you suggest in order to make categories more distinct/relevant without losing any of the information currently in the database (e.g., that the 'food Caesar' is a salad, and Bugs character is a Bunny, there is some more)? (assuming the database may have other food or character objects!) (5 pts)

Part B:

For real people in the database, we don't need to change any of the categories.

For fictional characters, we can change dob to when they were created or first appearance. We can change occupation to their most common role in stories.

For food, we can change dob to the year the food was created or invented. We can replace occupation with the type/category of food it is. We can also replace known for with the ingredients used in the food or a description of the food.