CSE 154: Web Programming

Lecture 21 Warmup: Motivating SQL

Using the following JSON template, write a JavaScript function `filterJSON` that takes as a parameter a JSON object with a key 'pokemon' that maps to an array of pokemon objects as detailed below, and returns a new JSON object that is a subset of the passed data with the following rules:

- Each item in the processed JSON must have a **name** containing the letter 'r'
- Each item must have an **id** less than 145
- Each item must have a **weakness** of 'rock'
- Each item in the returned JSON must be sorted by **type** alphabetically, breaking ties by **name** reverse-alphabetically.
- Each item in the returned JSON should only include the **name**, **id**, **type**, and **weakness**.

The following is a general schema and an example of an (unfiltered) input JSON object holding only one Pokemon object (to save space, but your solution should return potentially multiple objects).

JSON format (schema)

```
{ pokemon :
    [
      { name : pokemonname,
        description : description
        id: pokedexid,
        moves: [
          {
            name: move1name,
            type: move1type
          },
          {
            name: move2name,
            type: move2type
          },
          {
            name: move3name,
            type: move3type
          },
          {
            name: move4name,
            type: move4type
          }
        ],
        type: pokemontype,
        weakness: pokemonweakness
      },
      … more pokemon objects in the
array
    ]
}
```

Example

```
{ pokemon :
    [
      { name : "Articuno",
        description: "A legendary bird Pokemon that is said to
        appear to doomed people who are lost in icy
mountains.",
        id: 144,
        moves: [
          {
            name: "Ancient Power",
            type: "rock"
          },
          {
            name: "Ice Beam",
            type: "ice"
          },
          {
            name: "Hurricane",
            type: "flying"
          },
          {
            name: "Double Team",
            type: "normal"
          }
        ],
        type: "ice",
        weakness: "rock"
      }
    ],
    ... more pokemon objects in the array
}
```

Discuss and write your solution with your neighbors on the back of this page.

```
/**
 * Returns a filtered JSON object from the given JSON 'pokemon' containing
 * only the elements in the 'pokemon' array which:
 *   - have a name containing 'r'
 *   - have an id < 145
 *   - have a weakness of 'rock'
 *
 * Elements in the returned JSON have only the attributes 'name', 'id',
 * 'type', and 'weakness', and are returned in sorted order, first by
'type'
 * alphabetically and breaking ties with 'name' reverse-alphabetically.
 * Example of one object that may be in the filtered array:
 *
 * { "name" : "Articuno", "id" : 144, "type" : "ice", "weakness" : "rock"}
 */
function filterJSON(pokemonJSON) {
  let pokemon = pokemonJSON["pokemon"];
  let filtered = [];









  // return result back in JSON format, assigning the pokemon key to our
  // filtered array of JSON items
  let result = { "pokemon" : filtered };
  return result;
}
```