

Lecture 12 Notes - Wednesday 11/16/16

Reading Quiz

Question 1: Ans = E

Question 2: Ans = A

Question 3: Ans = C

Question 4: Ans = A

Question 5: Ans = D

Notes

JSON is a specification language for describing the structure of data. A JSON document is basically an unordered set of name-value pairs. The names are strings, but the values can be of several different types. These values can be nested; so a value can be of type Array, Object, ... Note that JSON is self-describing -- you don't define a schema before hand and make a bunch of records match that format. Instead, each "record" labels its values for you via the name-value pairs. So JSON lets you define the structure of your data on the fly.

Concept Question 1: C. For *A*, we'd be making basically a list of name-value pairs with a name corresponding to an object like the one shown here. This works, but the use of an array implies an order -- it's an ordered list. This doesn't make a whole lot of sense here, but it's not terrible. *B* almost works but we need a name to correspond to each of these values, the nested objects. This is why *C* is the right answer, because it adds these labels to match each person with their corresponding object.

Concept Question 2: B. We can see that *favorite_count* is not inside the value (which is *null*) corresponding to the *contributors* label, so *A* does not work. The same reasoning leads us to see that *C* and *D* do not work either. Since *favorite_count* is at the "first level" inside of this object (as in, it is not contained inside a nested object), *B* gives us the value we want. It goes to the name "favorite_count" and gives us the corresponding value.

Concept Question 3: C. In this code, we are using the API to search for tweets containing "#childlabor" and storing these in *cursor*. We are then iterating over the pages of these results and storing each item, where each *item* is a tweet. In the function, we are storing each tweet in the same table (*tweets*), so *C* is the correct answer.