

# Semistructured data model

Tuesday, April 23, 2019 1:23 PM

- e.g., XML, JSON, protobuf.
- JSON human-readable; light-weighted text-based

- **JavaScript Object Notation (JSON)**

- Compared to relational data model, JSON is
    - flexible, nested structure (trees)
    - No need to predefine schema (self-describing)
    - Text representation (good for exchange, bad for performance)

RA : binary representation

- **JSON Types**

- Primitive: **number, string, Boolean, null.**
    - **Object** collection of name-value pairs

e.g. `{"name1": val1, "name2": val2, ...}`

name is a key; avoid using duplicate keys. (names)

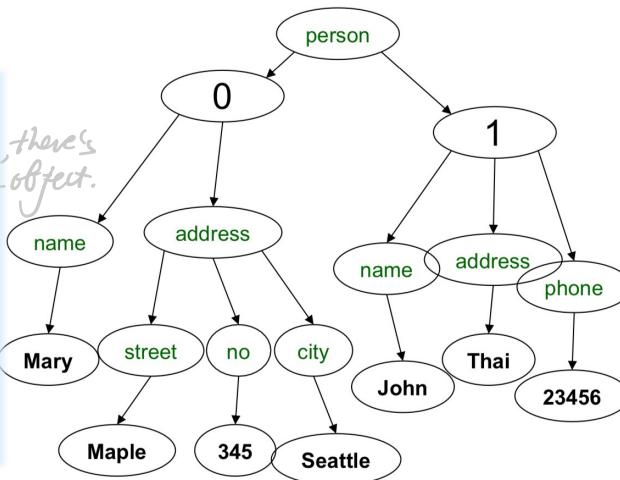
e.g., a book with 3 authors, use an array instead of three attributes.

- **Array** (ordered)

[obj<sub>1</sub>, obj<sub>2</sub>, ...]

## JSON Semantics: a Tree !

```
{"person":  
  [ {"name": "Mary",  
      "address": {  
        "street": "Maple",  
        "no": 345,  
        "city": "Seattle"}},  
   {"name": "John",  
    "address": {"  
      "street": "Thailand",  
      "phone": 2345678}}]  
}
```



- Map relational data to JSON (Accept nested relation)  
↳ Not necessarily flat

{ "person": [

  { "name": "xx", "age": 18 },

  { "name": "yy", "age": 19 }

]

- Notice that arrays are ordered in JSON

- Semi-Structured Data

0 JSON is self-describing

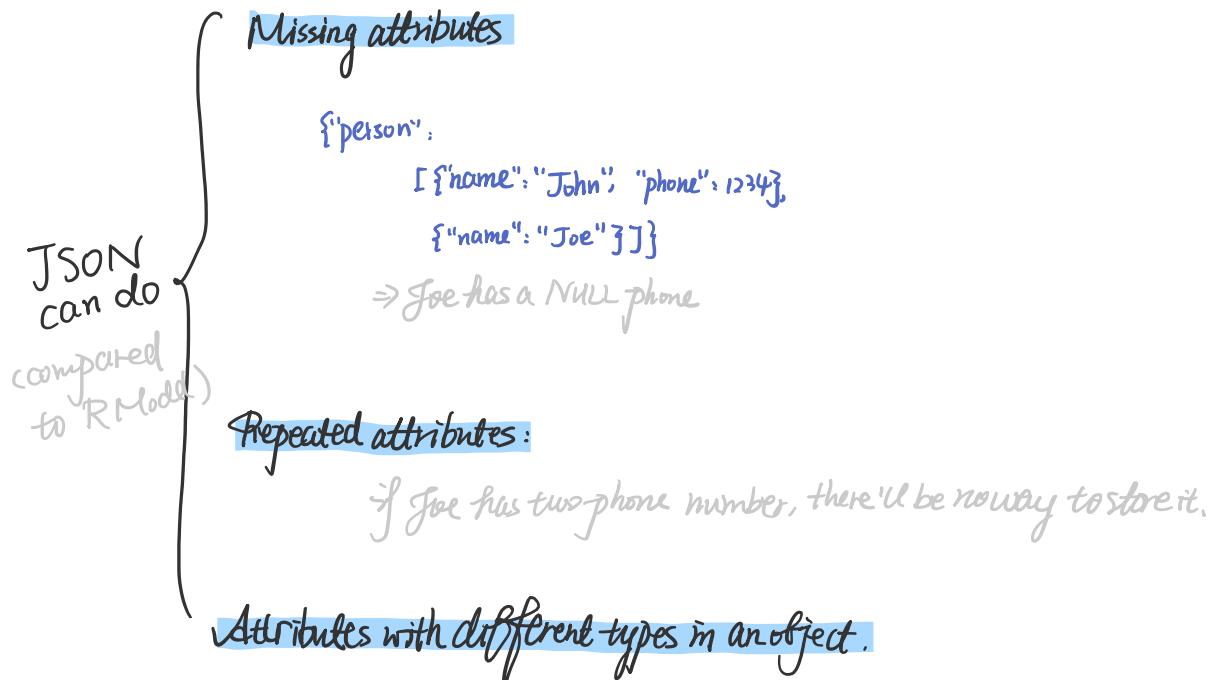
- Schema along with document or data

- some elements become a part of schema.

- Mapping relational data to JSON:

- Use 3 flat relations

- Use nested relation.



- Why semi-structured data

- is a good data exchange format

- Increasing systems start to use them as data model.

- e.g. Snowflake

- Query Language for semi-structured data
  - XML: XPath, XQuery
  - Protobuf: SQL-ish language (Dremel) used internally by google.
  - JSON