JSON: History of a Data Model*

Jan Van den Bussche

Hasselt University

*title stolen from “JSONiq: History of a Query Language” [Florescu & Fourny 2013]

Data model

- Repertoire of data structures and operations
- Storage & retrieval of persistent data
- Database system
- Programming language

COBOL (1959)

- Files
- Collection of records, atomic fields
- Indexed access: records with specified key value
- Sequential access

CODASYL Network Model 1973

- Allow performant updates, interactive, concurrent access
- Still: collections of records with indexed and sequential access
- New: links between records

CODASYL links

- One record linked to a set of records of another type
- Many–one relationship
- E.g., department linked to its employees
- Also, project linked to its workers
- Access: first child, next sibling, parent
- Object identity

MUMPS (1966)

- Associative arrays of atoms
- Direct & sequential access
- Multidimensional
- Nesting! Array of Array of int
- Concurrent updates
- First database programming language

MUMPS example

- Keep data for patients, when they were seen, what treatments were prescribed

```
data(111)=“John Doe”
data(111,2016-09-01,“decadron”,“qty”)=5
data(111,2016-09-01,“decadron”,“freq”)=3
data(111,2016-09-01,“dafalgan”,...)=...
data(111,2017-07-20,...)=...
data(112)=“Mary Lu”
...
```

Relational Model (1977)

- Again collections of records, atomic fields
- “Relation”
- Revolutionary idea: operate not on individual records but on entire relations
- Relational algebra: union, cartesian product (join), selection, projection, aggregation
- Queries expressed declaratively, logic
- SQL

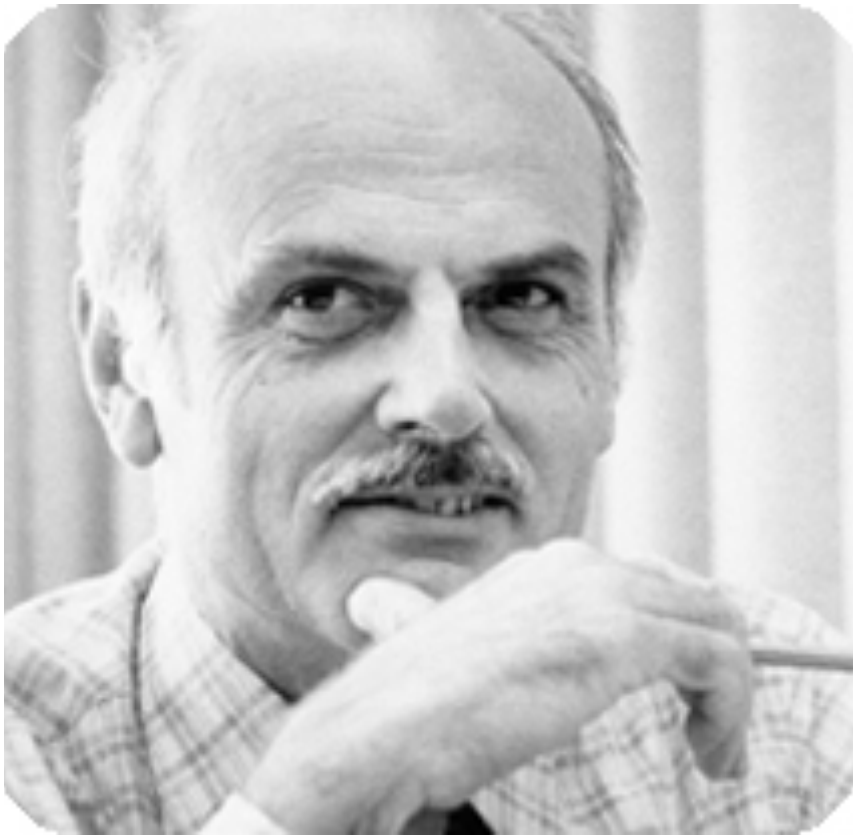
Query

- “Find departments that have an employee working on a project in China”
- Network model:
  1. Navigate departments–employees–projects
  2. Navigate projects–workers–departments
  3. …
- SQL: `select did from Department, Employee, Project where did=dept and pid=proj and location='China'`
- Relational algebra

Bachman vs Codd

“Data processing is all about navigation”*

“Data processing is all about logic”*



* Not actual quotes!

Codd wins

- A(a:int)
- B(b:int)
- `select a from A, B where a=b`

Unix dbm (1979)

- Collection of records, indexed acces
- Updates and concurrency, but no logic
- Early NoSQL databases essentially this, with JSON-valued fields

Nested relations (1980s, 1990s)

- Relation-valued fields
- E.g., give departments with their employees
- Dept(did, location, emps: Emp(name,salary))
- SQL:1999, SQL:2003, but never thoroughly supported by relational systems

Nested relational algebra

- e1 : t -> set(s)
- e2 : set(t)
- flatten(for x in e2 return e1(x))
- set constructor
- emptiness test
- Collection frameworks in prog.lang.
- NoSQL systems (map-reduce)
- SPARK (2014)

ODMG model (1999)

- Classes (record types with object identity)
- Collection types: set, bag, list, array, map
- OQL

```
select struct(
   name: x.name,
   hps: (select y from x.subordinates as y
         where y.salary > 100000))
```

The Web

- Hypertext
- Data exchange
  - Java
  - XML
  - XML databases
- Ordered tree
  - nodes labeled by atoms
- Depth not fixed: semi-structured

How to query XML?

- Consider each node an object
- class Node
  - id:atom
  - label:atom
  - children:list<Node>
  - descendants:list<Node>
  - ancestors, parent, left-siblings, right-siblings

XQuery (2007)

```
for $a in fn:distinct-values($bib/book/author)
order by $a
return
  <author>
    <name> {$a} </name>
    <books>  {  for $b in $bib/book[author = $a]
                order by $b/title
                return $b/title  } </books>
  </author>
```

- Result is a list of nodes (data model is not closed)

JSON (2013)

- Ajax
- Javascript
- JSON
- “Object” = associative array
- Also normal arrays
- Nested, semistructured
- SQL for JSON in development
- Solid query processing in its infancy

RDF and SPARQL (2008)

- Semantic Web
- Open Linked Data
- RDF: ternary relation
- SPARQL

```
(:id1, :name, ‘Jan’)
(:id1, :son, :id2)
(:id1, :son, :id3)
(:id2, :name, ‘Steven’)
(:id3, :job, :id4)
```

```
select x, y
where (x, :son, y)
optional (y, :name, z)
```

- Result is a heterogeneous relation

Graph Data Model (2013)

- Directed graph: nodes, edges
- Nodes and edges carry a JSON value
- Cypher: curious mix of OQL, SPARQL, and shortest paths

Zoo of data models

		LOGIC	OBJECT ID	NESTING	SEMISTRUCT	PROG.LANG
COBOL	1959					
MUMPS	1966					
NETWORK	1973					
RELATIONAL	1977					
DBM	1979					
NESTED	1999					
ODMG	1999					
XQUERY	2007					
SPARQL	2008					
JSON	2013					
GRAPH	2013					
SPARK	2014					