Semistructured Data
## Structured Data

<table>
<thead>
<tr>
<th>ID</th>
<th>Last Name</th>
<th>First Name</th>
<th>Title</th>
<th>Birth Date</th>
<th>Hire Date</th>
<th>City</th>
<th>Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Davolio</td>
<td>Nancy</td>
<td>Ms.</td>
<td>08-dic-1968</td>
<td>01-mag-1992</td>
<td>Seattle</td>
<td>WA</td>
</tr>
<tr>
<td>2</td>
<td>Fuller</td>
<td>Andrew</td>
<td>Dr.</td>
<td>19-feb-1952</td>
<td>14-ago-1992</td>
<td>Tacoma</td>
<td>WA</td>
</tr>
<tr>
<td>3</td>
<td>Leverling</td>
<td>Janet</td>
<td>Ms.</td>
<td>30-ago-1963</td>
<td>01-apr-1992</td>
<td>Kirkland</td>
<td>WA</td>
</tr>
<tr>
<td>4</td>
<td>Peacock</td>
<td>Margaret</td>
<td>Mrs.</td>
<td>19-set-1958</td>
<td>03-mag-1993</td>
<td>Redmond</td>
<td>WA</td>
</tr>
<tr>
<td>5</td>
<td>Buchanan</td>
<td>Steven</td>
<td>Mr.</td>
<td>04-mar-1955</td>
<td>17-ott-1993</td>
<td>London</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Suyama</td>
<td>Michael</td>
<td>Mr.</td>
<td>02-lug-1963</td>
<td>17-ott-1993</td>
<td>London</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>King</td>
<td>Robert</td>
<td>Mr.</td>
<td>29-mag-1960</td>
<td>02-gen-1994</td>
<td>London</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Order ID</th>
<th>Customer</th>
<th>Emp ID</th>
<th>Order Date</th>
<th>Required Date</th>
<th>Shipped Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>10248</td>
<td>Wilman Kala</td>
<td>1</td>
<td>04-lug-1996</td>
<td>01-ago-1996</td>
<td>16-lug-1996</td>
</tr>
<tr>
<td>10249</td>
<td>Tradição Hiperm.</td>
<td>6</td>
<td>05-lug-1996</td>
<td>16-ago-1996</td>
<td>10-lug-1996</td>
</tr>
<tr>
<td>10250</td>
<td>Hanari Carnes</td>
<td>3</td>
<td>08-lug-1996</td>
<td>05-ago-1996</td>
<td>12-lug-1996</td>
</tr>
<tr>
<td>10253</td>
<td>Hanari Carnes</td>
<td>3</td>
<td>10-lug-1996</td>
<td>24-lug-1996</td>
<td></td>
</tr>
</tbody>
</table>
Unstructured data

• **Sample databases included with Access**
  – Microsoft Access provides sample databases that you can use while you're learning Access.
  – **Northwind Traders sample database**
  – The Northwind database and Access project (available from the *Sample Databases* command on the *Help* menu) contains the sales data for a fictitious company called Northwind Traders, which imports and exports specialty foods from around the world. By viewing the *database objects* included in the Northwind database. ...
Semistructured Data

complex object

atomic object
Sources of SSD

• Data integration
• Scientific data
• Documents
• WWW
SSD data models

• A data model:
  – The information behind the syntax (by a denotation or through equality)
  – The operators

• Some alternatives:
  – RDF: graphs with node equality – triples of URIs
  – OEM: graphs modulo bisimulation
  – XML: ordered trees with “node identity”, parent pointer, fusion of blanks in text nodes
  – JSON: unordered trees with ordered arrays
Graphs

• Graph with labelled edges:
  – \((N,E,f)\) con \(E \subseteq (N \times N)\) ed \(f: E \rightarrow \Phi\)

• Graph equality: node isomorphism
  – \((\{1,2\}; \{(1,2),(2,1)\}; \{(1,2)\rightarrow A; (2,1)\rightarrow B\})\)
  – \((\{a,b\}; \{(a,b),(b,a)\}; \{(a,b)\rightarrow A; (b,a)\rightarrow B\})\)
  – \((\{a,b\}; \{(a,b),(b,a)\}; \{(a,b)\rightarrow B; (b,a)\rightarrow A\})\)
Isomorphism

• Isomorphism between \((E, N, \Phi)\) and \((E', N', \Phi')\): bijection \(\sigma\) from \(N\) to \(N'\) such that:
  
  – \((n, m) \in E\) iff \((\sigma(n), \sigma(m)) \in E'\)
  
  – \((n, m) \in E\) \(\Rightarrow\) \(\Phi(n, m) = \Phi'(\sigma(n), \sigma(m))\)

• We say that nodes are ‘anonymous’, have ‘no identity’
RDF

• A graph is a set of triples:

  <http://www.w3.org/People/EM/contact#me>
  <http://www.w3.org/.../22-rdf-syntax-ns#type>
  <http://www.w3.org/swap/pim/contact#Person>

  <http://www.w3.org/People/EM/contact#me>
  <http://www.w3.org/swap/pim/contact#fullName>
  "Henry Miller"

• Nodes are *not* anonymous
• Graphs with labelled edges and anonymous nodes (leaves are labelled with atomic values)
• Bisimulation: set equality generalized to graphs
  – \{a: v, b: w\} = \{b: w, a: v\}
  – \{a: v, a: v, b: w\} = \{a: v, b: w, b: w\}
• Formally: exists \( R \subseteq G \times G' \) such that:
  • \( n R m \) and \( n,l,n' \) in \( G \) \( \Rightarrow \) exists \( m,l,m' \) in \( G' \) with \( n' R m' \) and vice versa
  • \( n R m \) and \( n \) leaf in \( G \) \( \iff \) \( m \) leaf in \( G' \)
• The idea: paths are observable, nodes are not, multiplicity of paths is not
Bisimulation
XDM (XML Data Model)

• Node-labelled ordered trees:
  – \{a: v, b: w\} \neq \{b: w, a: v\}
  – \{a: v, a: v, b: w\} \neq \{a: v, b: w, b: w\}
Ordered trees and binary trees

• Binary trees:
  – bt ::= _ | label[bt,bt]
  – Ad es.: a[b[_,_],_] ≠ a[_,b[_,_]]

• Ordered forests are isomorphic to binary trees
Ordered forests and binary trees
XDM: further details

• Every node has an identity, that can be compared to other nodes identity
• Who is the parent of a[b,c,d]?
• In XDM, you can navigate from a node to its parent
• An XDM tree is actually a pair
  – A whole tree (the ‘document’)
  – A pointer inside that tree (the ‘current node’)
XDM: further details

- 7 types of nodes: elements, attributes (unordered), text nodes, ...
<menu id="file">
  <popup>
    <menuitem value="New" onclick="CreateNew()" />
    <menuitem value="Close" onclick="CloseDoc()" />
  </popup>
</menu>

{"menu": {
  "id": "file",
  "popup": {
    "menuitem": [
      {"value": "New", "onclick": "CreateNew()"},
      {"value": "Close", "onclick": "CloseDoc()"}
    ]
  }
}}
JSON

• http://json.org/

object ::= \{\} | \{ (string : value,)* string : value\}
array ::= [] | [ (value ,)* value ]
value ::= object
       | array
       | string
       | number
       | true | false | null
string ::= " char * "
number ::= ....

• Objects are unordered
• The names in an object should be unique
The object model

• Similar to OEM

• However:
  – A priori schema, every object belong to a class, the class specifies the outward labels
  – Classes have methods
Other models

• TQL
  – Multiplicity is observable, order is not:
    • \{a: v, b: w\} = \{b: w, a: v\}
    • \{a: v, a: v, b: w\} \neq \{a: v, b: w, b: w\}
  – Si possono interpretare come alberi con archi etichettati oppure foreste con nodi etichettati

• Compositional graphs:
  – \((x,a,y) \mid (x,b,z)\)
  – \((\forall x.(x,a,y) \mid (x,b,z))\)
  – \((\forall x.(x,a,y) \mid (x,b,z)) \mid (\forall x.(x,a,y) \mid (x,b,z))\)
Sources

- RDF: [http://www.w3.org/TR/rdf-primer/](http://www.w3.org/TR/rdf-primer/)
- Json.org