Semi-structured Data

1 - Introduction
Outline

• Structured Data
• Semi-structured Data
• Why Semi-structured Data?
• The Data Model
• Store Semi-structured Data
Structured Data

- Data is structured in semantic chunks - entities
  
  VIE, Vienna International, Vienna
  LHR, London Heathrow, London
  VIE, LHR, BA
  VIE, LHR, OS
  BA, British Airlines
  OS, Austria Airlines

- Similar entities are grouped together - classes

  Flights
  
  VIE, Vienna International, Vienna
  LHR, London Heathrow, London
  VIE, LHR, BA
  VIE, LHR, OS
  BA, British Airlines
  OS, Austria Airlines

  Airports
  
  Airlines
Structured Data

- Entities in the same class have the same descriptions - attributes

<table>
<thead>
<tr>
<th>Airports</th>
<th>Flights</th>
</tr>
</thead>
<tbody>
<tr>
<td>(VIE, Vienna International, Vienna)</td>
<td>(VIE, LHR, BA)</td>
</tr>
<tr>
<td>(LHR, London Heathrow, London)</td>
<td>(VIE, LHR, OS)</td>
</tr>
<tr>
<td>(Airport_Code, Name, City)</td>
<td>(Origin, Destination, Airline)</td>
</tr>
</tbody>
</table>

| Airlines |  |
|----------|  |
| (BA, British Airlines) |  |
| (OS, Austria Airlines) |  |
| (Airline_Code, Name) |  |
Structured Data

- Entities in the same class have the same descriptions - attributes

<table>
<thead>
<tr>
<th>Airports</th>
<th>Flights</th>
<th>Airlines</th>
</tr>
</thead>
<tbody>
<tr>
<td>(VIE, Vienna International, Vienna)</td>
<td>(VIE, LHR, BA)</td>
<td>(BA, British Airways)</td>
</tr>
<tr>
<td>(LHR, London Heathrow, London)</td>
<td>(VIE, LHR, OS)</td>
<td>(OS, Austria Airlines)</td>
</tr>
<tr>
<td>(Airport_Code, Name, City)</td>
<td>(Origin, Destination, Airline)</td>
<td>(Airline_Code, Name)</td>
</tr>
</tbody>
</table>

- Attributes in similar entities
  - same format (string, integer, date, etc.)
  - predefined length
  - all present
  - same order

... strict structure forced by a schema!!!
Structured Data - Relational Model

- Database model for structured data:
  - entities $\rightarrow$ records (or tuples)
  - classes $\rightarrow$ tables (or relations)

- Records grouped in tables
### Structured Data: “On the Fly” Example

#### Airports

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIE</td>
<td>Vienna International</td>
<td>Vienna</td>
</tr>
<tr>
<td>LHR</td>
<td>London Heathrow</td>
<td>London</td>
</tr>
<tr>
<td>LGW</td>
<td>London Gatwick</td>
<td>London</td>
</tr>
<tr>
<td>LCA</td>
<td>Larnaca International</td>
<td>Larnaca</td>
</tr>
<tr>
<td>GLA</td>
<td>Glasgow</td>
<td>Glasgow</td>
</tr>
<tr>
<td>EDI</td>
<td>Edinburgh</td>
<td>Edinburgh</td>
</tr>
</tbody>
</table>

#### Airlines

<table>
<thead>
<tr>
<th>Code</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIE</td>
<td>British Airways</td>
</tr>
<tr>
<td>VIE</td>
<td>Austrian Airlines</td>
</tr>
<tr>
<td>LHR</td>
<td>British Airways</td>
</tr>
<tr>
<td>LGW</td>
<td>EasyJet</td>
</tr>
</tbody>
</table>

#### Flights

<table>
<thead>
<tr>
<th>Origin</th>
<th>Destination</th>
<th>Airline</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIE</td>
<td>LHR</td>
<td>British Airways</td>
</tr>
<tr>
<td>VIE</td>
<td>LHR</td>
<td>Austrian Airlines</td>
</tr>
<tr>
<td>LHR</td>
<td>EDI</td>
<td>British Airways</td>
</tr>
<tr>
<td>LGW</td>
<td>GLA</td>
<td>EasyJet</td>
</tr>
</tbody>
</table>
“Persons” Example

Gerti Kappel, 18870, 18896, gerti@big.tuwien.ac.at

Andreas, Pieris, pieris@dbai.tuwien.ac.at, 740072, 18493

Wolfgang Fischl, wfischl@dbai.tuwien.ac.at, 740050

Martin, Fleck, 58801, fleck@big.tuwien.ac.at
Semi-structured Data (SSD)

- Data is structured in semantic entities
- Similar entities are grouped in classes

- Entities in the same class may not have the same attributes
  - may have different format
  - may have different length
  - not all required
  - may have different order

there is structure

but not too much structure
Semi-structured Data: “Persons” Example

<table>
<thead>
<tr>
<th>Name</th>
<th>ID</th>
<th>Email</th>
<th>Telephone</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gerti Kappel</td>
<td>18870, 18896</td>
<td><a href="mailto:gerti@big.tuwien.ac.at">gerti@big.tuwien.ac.at</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Andreas Pieris</td>
<td>740072, 18493</td>
<td><a href="mailto:pieris@dbai.tuwien.ac.at">pieris@dbai.tuwien.ac.at</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wolfgang Fischl</td>
<td>740050</td>
<td><a href="mailto:wfischl@dbai.tuwien.ac.at">wfischl@dbai.tuwien.ac.at</a></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Martin Fleck</td>
<td>58801</td>
<td><a href="mailto:fleck@big.tuwien.ac.at">fleck@big.tuwien.ac.at</a></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **There is structure**
  - Each row is a semantic entity - person
  - All entities are grouped in a class - persons

- **But not too much structure**
  - Entities have no regular structure
  - Structure of future entities is unpredictable
Why Semi-structured Data?

- There are data sources that we would like to treat as **databases**, but which cannot be constraint by a schema

- Flexible format for **exchanging data** between different places

... the WEB

**GOAL:** Reconcile document view (web) with strict structures (databases)
Data Model

• We need an effective way to represent semi-structured data

• Like the relational model for structured data

... any ideas?
Trees as Data Model

Gerti Kappel, 18870, 18896, gerti@big.tuwien.ac.at
Andreas, Pieris, pieris@dbai.tuwien.ac.at, 740072, 18493
Trees as Data Model

- SSD can be represented as a (labelled) tree:
  - leaf nodes standing for single data items
  - inner nodes have no label
  - edges labelled with elements

- Such a model is called **self-describing** - information that is usually associated with a schema is contained within the data

- Data carries its own description
SSD: Representing Relational Data

Structured data is a **special case** of semi-structured data

relational data can be represented as a tree (with an overhead)

<table>
<thead>
<tr>
<th>R</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>a₁</td>
<td>b₁</td>
<td>c₁</td>
<td></td>
</tr>
<tr>
<td>a₂</td>
<td>b₂</td>
<td>c₂</td>
<td></td>
</tr>
</tbody>
</table>
Store Semi-structured Data

- There are various formalisms to store semi-structured data
  - Object Exchange Model (OEM) - close to previous examples
  - JavaScript Object Notation (JSON)
  - eXtensible Markup Language (XML)
Store Semi-structured Data

{persons:
  {person:
    {name: "Gerti Kappel"
     tel: 18870
     fax: 18896
     email: "gerti@big.tuwien.ac.at"}
  }
  {person:
    {name: {first: "Andreas", last: "Pieris"}
     email: "pieris@dbai.tuwien.ac.at"
     tel: 740072
     fax: 18493}
  }
}
Store Semi-structured Data

XML Representation

```xml
<persons>
  <person>
    <name>Gerti Kappel</name>
    <tel>18870</tel>
    <fax>18896</fax>
    <email>gerti@big.tuwien.ac.at</email>
  </person>
  <person>
    <name>
      <first>Andreas</first>
      <last>Pieris</last>
    </name>
    <email>pieris@dbai.tuwien.ac.at</email>
    <tel>740072</tel>
    <fax>18493</fax>
  </person>
</persons>
```
Store Semi-structured Data

• There are various formalisms to store semi-structured data
  o Object Exchange Model (OEM) - close to previous examples
  o JavaScript Object Notation (JSON)
  o eXtensible Markup Language (XML)

• Different syntax

• Different mechanisms for self-describing

• Different description mechanisms
  o Which attributes are allowed/required
  o Which values are allowed/required

• Different query languages and manipulation mechanisms