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 Airways
  OS, Austrian Airlines

• Similar entities are grouped together - classes

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

  Airports
  Airlines
Structured Data

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

**Airports**
- (VIE, Vienna International, Vienna)
- (LHR, London Heathrow, London)

**Airlines**
- (BA, British Airways)
- (OS, Austrian Airlines)

**Flights**
- (VIE, LHR, BA)
- (VIE, LHR, OS)
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, Austrian 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 → records (or tuples)
  - classes → tables (or relations)

- Records grouped in tables

![Diagram of a table with attributes and records]
Structured Data: “On the Fly” Example

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

<table>
<thead>
<tr>
<th>Airlines</th>
<th>Code</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>BA</td>
<td>British Airways</td>
</tr>
<tr>
<td></td>
<td>OS</td>
<td>Austrian Airlines</td>
</tr>
<tr>
<td></td>
<td>U2</td>
<td>EasyJet</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Flights</th>
<th>Origin</th>
<th>Destination</th>
<th>Airline</th>
</tr>
</thead>
<tbody>
<tr>
<td>VIE</td>
<td>LHR</td>
<td></td>
<td>British Airways</td>
</tr>
<tr>
<td>VIE</td>
<td>LHR</td>
<td></td>
<td>Austrian Airlines</td>
</tr>
<tr>
<td>LHR</td>
<td>EDI</td>
<td></td>
<td>British Airways</td>
</tr>
<tr>
<td>LGW</td>
<td>GLA</td>
<td></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

Bill, Robert, 188316, bill@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
- Attributes of similar entities may have different format, may have different length, not all required, may have different order
Semi-structured Data: “Persons” Example

<table>
<thead>
<tr>
<th>Name</th>
<th>ID Numbers</th>
<th>Email</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>
</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>, 740072, 18493</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>, 740050</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>
</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
Structured data is a **special case** of semi-structured data

relational data can be represented as a tree (with an overhead)
Store Semi-structured Data

- There are various formalisms to store semi-structured data
  - Object Exchange Model (OEM)
  - 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}  
  }  
}
```

OEM Representation
Store Semi-structured Data

<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
  - Object Exchange Model (OEM)
  - JavaScript Object Notation (JSON)
  - eXtensible Markup Language (XML)

- Different syntax

- Different mechanisms for self-describing

- Different description mechanisms
  - Which attributes are allowed/required
  - Which values are allowed/required

- Different query languages and manipulation mechanisms

but the goal is the same: store SSD
Sum Up

• **Structured Data**
  o Similar entities grouped in classes
  o Similar entities have a regular structure
  o Relational Model

• **Semi-structured Data**
  o Similar entities grouped in classes
  o Similar entities have irregular structure
  o Trees as a Model

• **Store Semi-structured Data**
  o Various formalisms
  o eXtensible Markup Language (XML)