Big Data and Data Mining

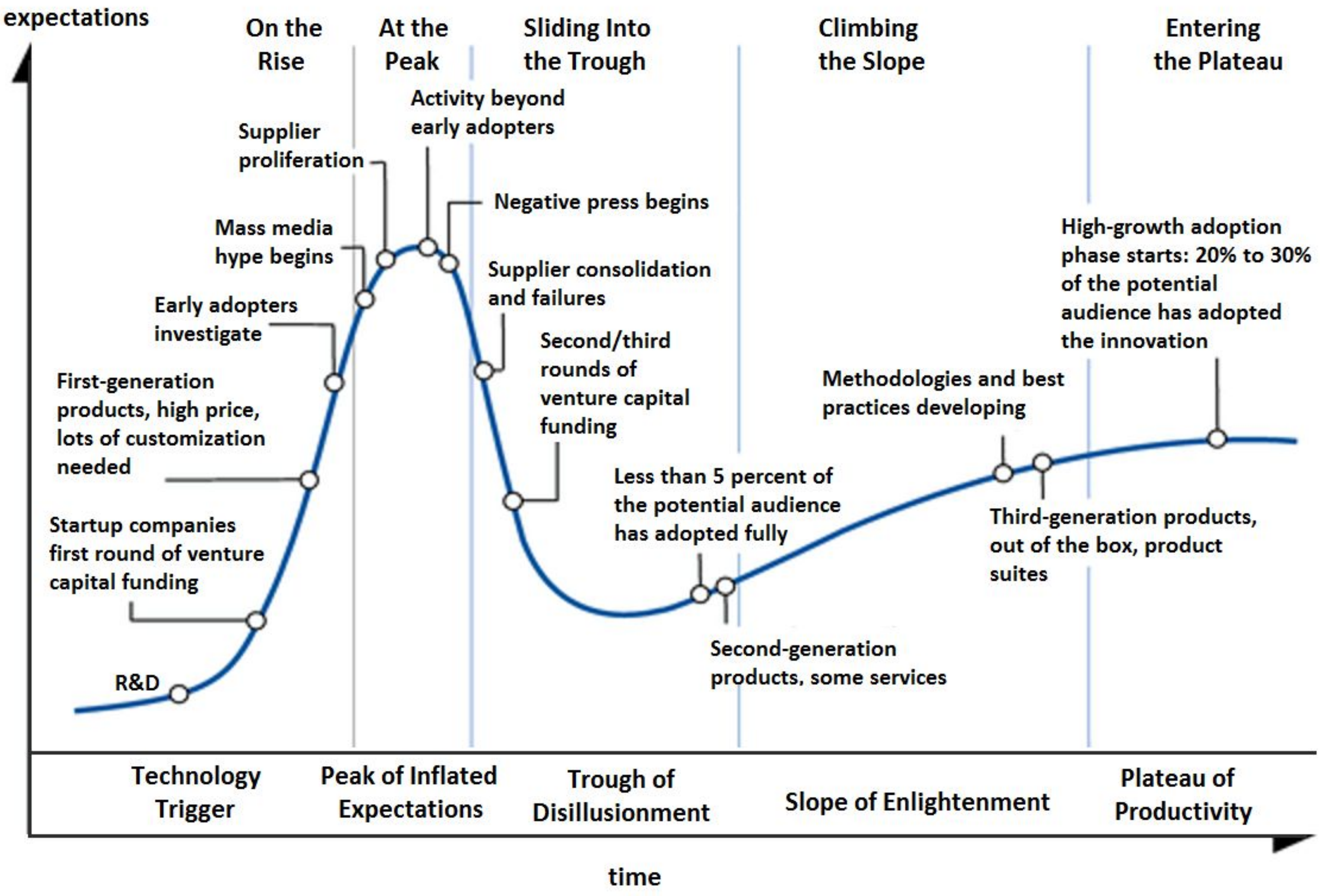
Semi-structured and unstructured data

Flavio Bertini

flavio.bertini@unipr.it

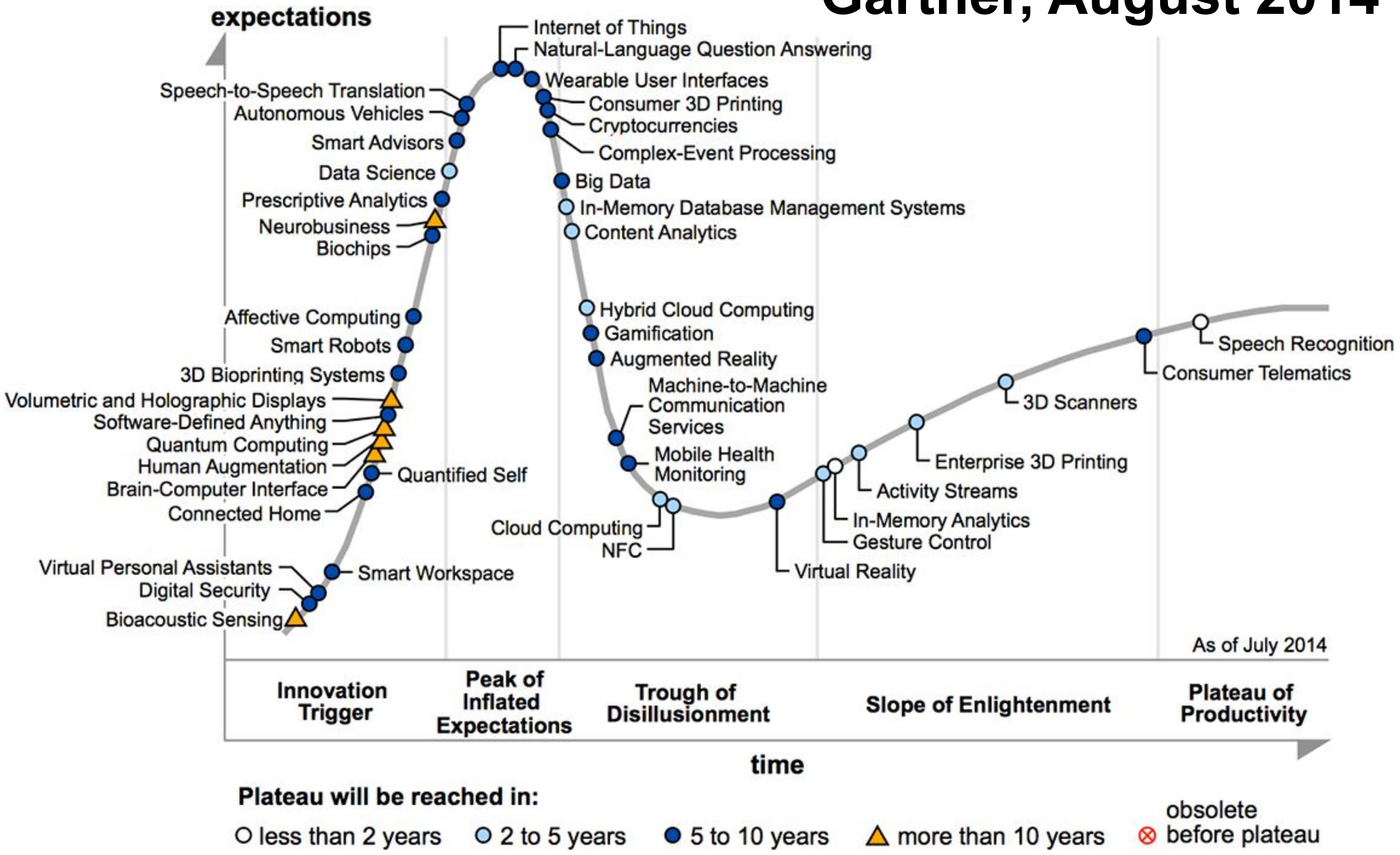
Motivations (1)

Motivations (2)


expectations
On the Rise
At the Peak
Sliding Into the Trough
Climbing the Slope
Entering the Plateau
Supplier proliferation
Activity beyond early adopters
Negative press begins
Mass media hype begins
Supplier consolidation and failures
High-growth adoption phase starts: 20% to 30% of the potential audience has adopted the innovation
Early adopters investigate
Second/third rounds of venture capital funding
Methodologies and best practices developing
First-generation products, high price, lots of customization needed
Less than 5 percent of the potential audience has adopted fully
Third-generation products, out of the box, product suites
Startup companies first round of venture capital funding
Second-generation products, some services
R&D
Technology Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity
time

Motivations (3)

Gartner, August 2014



Introduction

- Management systems of relational data are used in many significant applications, as in information systems of banks and big companies
    - However, most digital data available today is not in relational form
- The production of massive amounts of non-relational data has intensified over time, due to the diffusion of Internet and media sharing platforms
    - This kind of data has generally different properties with respect to the data managed by relational systems

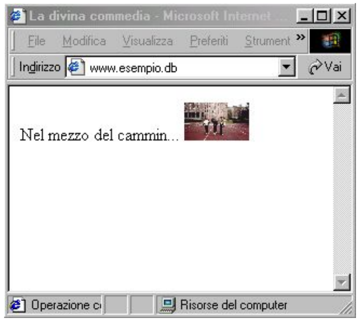
Structural classification of data

STRUCTURED

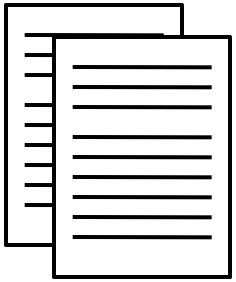
UNSTRUCTURED

id-pers	*name*	*surname*
0000001	Jon	Doe
0000002	Bob	Walker

id-pers	*phone*
0000001	051 1234
0000001	333 3333



Nel mezzo del cammin...



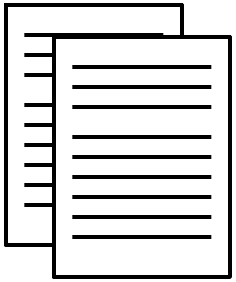
Structured data

id-pers	*name*	*surname*
0000001	Jon	Doe
0000002	Bob	Walker

id-pers	*phone*
0000001	051 1234
0000001	333 3333

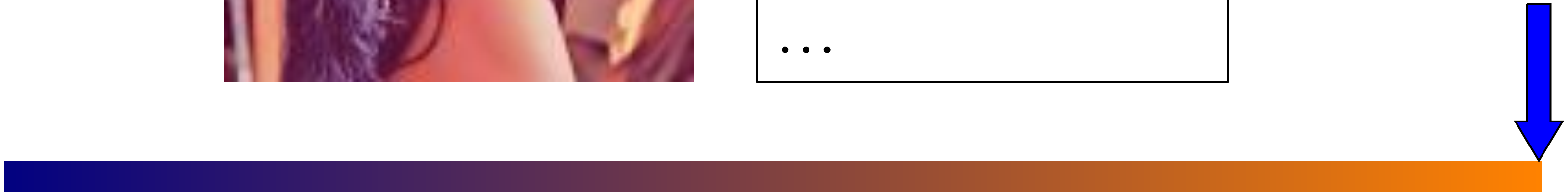
STRUCTURED DATA (SCHEMA)

Nel mezzo del cammin...



Unstructured data (1)

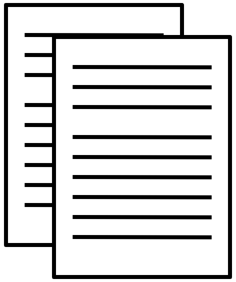
vu45s89gysJPGi
8gbyygsvs954gy
4598y9syg5vts9
4lygs98yg9s45y
g584gyt459gyg4
…



id-pers	*name*	*surname*
0000001	Jon	Doe
0000002	Bob	Walker

id-pers	*phone*
0000001	051 1234
0000001	333 3333

Nel mezzo del cammin...



RAW DATA

Unstructured data (2)

The old believe everything, the middle-aged suspect everything, the young know everything ...
O.Wilde

How can you prove whether at this moment we are sleeping, and all our thoughts are a dream; or whether we are awake, and talking to one another in the waking state?
Plato

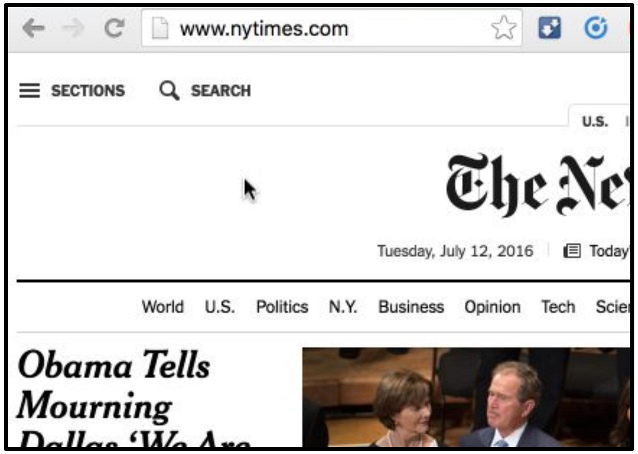
id-pers	*name*	*surname*
0000001	Jon	Doe
0000002	Bob	Walker

id-pers	*phone*
0000001	051 1234
0000001	333 3333

Nel mezzo del cammin...

DATA WITHOUT SCHEMA

Semi-structured data



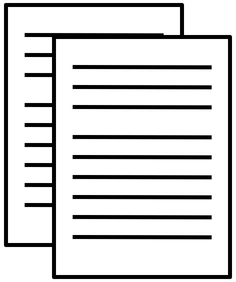
Obama Tells Mourning Dallas 'We Are

```
<html>

<title>The New York Times</title>

Tuesday, July 12, 2016 ...

<img/></html>
```

id-pers	*name*	*surname*
0000001	Jon	Doe
0000002	Bob	Walker

id-pers	*phone*
0000001	051 1234
0000001	333 3333

DATA WITH PARTIAL STRUCTURE

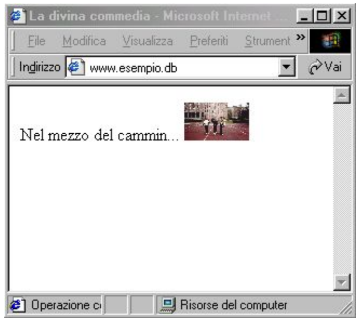


Non-relational data

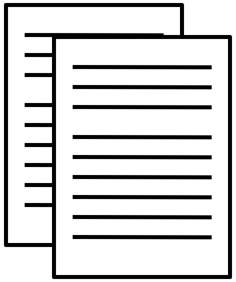
- Rightmost data of this classification is called **unstructured** data, and it needs specific processing, as we will see
  - The research area which study how to manage and access these kind of data is called **Information Retrieval**
- Data in the middle is called **semi-structured** data, and it shows properties of both structured and unstructured data
  - One of the most used language for semi-structured data representation is **XML**
- We will see now some examples to show why the relational model is not suitable to manage this kind of data

id-pers	*name*	*surname*
0000001	Jon	Doe
0000002	Bob	Walker

id-pers	*phone*
0000001	051 1234
0000001	333 3333



Nel mezzo del cammin...



Semi-structured data

Relational and semi-structured data: first comparison

Relational	Semi-structured
Clear distinction between schema and data	Partial schema with same properties of data
Based on the concept of set	Based on the concept of list
Unordered	Ordered
Not nested	Nested

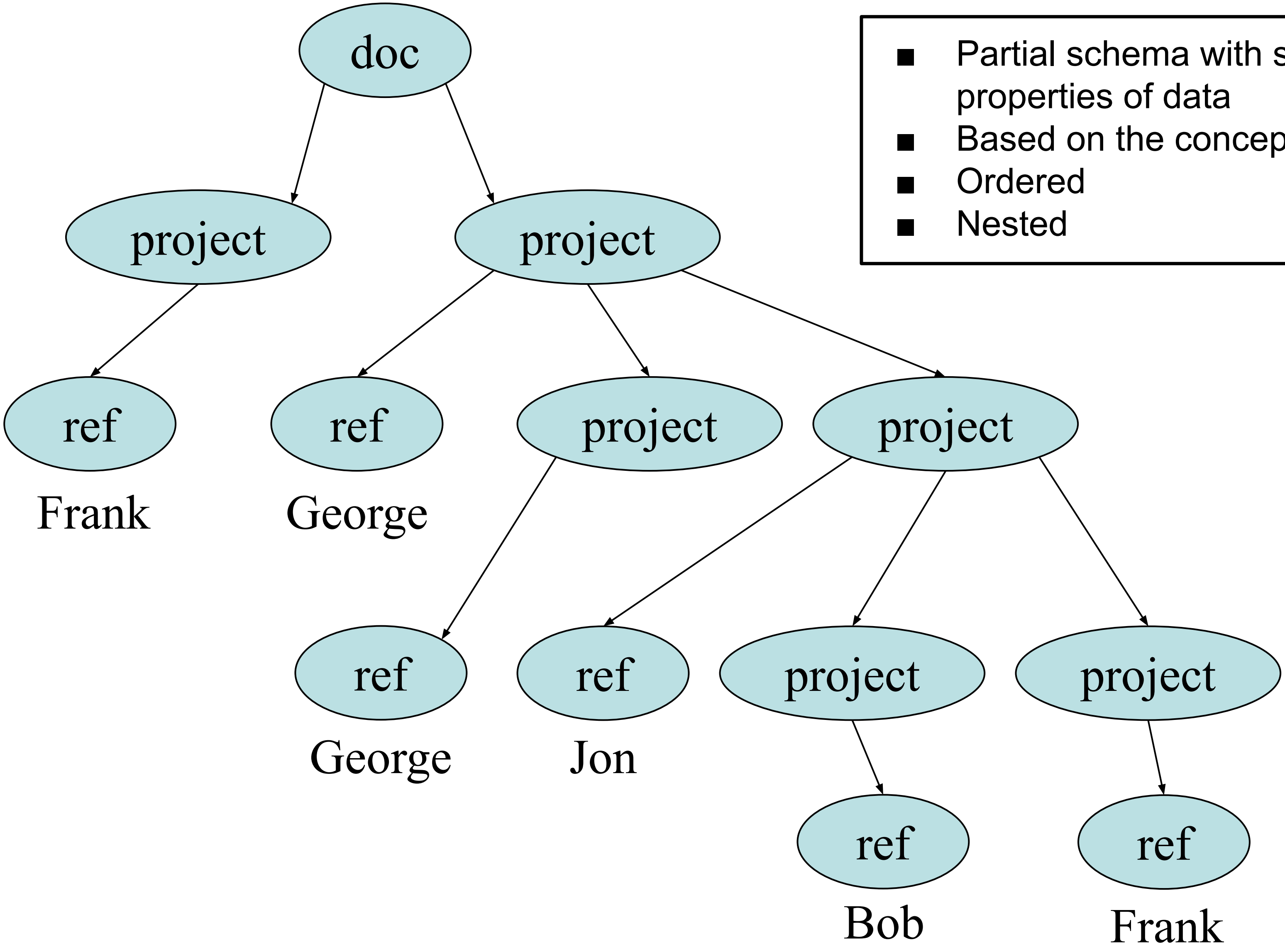
Extensible Markup Language a.k.a. XML

- The main format for semi-structured data representation is XML (E**x**tensible **M**arkup **L**anguage), that is a markup language similar to HTML, but without predefined tags
- It can be used
  1. to represent structured data, for example in order to exchange them between different applications, but also
  2. to represent semi-structured data, exploiting the flexibility and the capability of indicating both the data and the schema
- In the first case, data can be first located in a relational system, and then be converted to XML
- In the second case, the relational model is not particularly suited to manage this data. Let's see an example

A relational model for XML data (1)

```
<doc>
    <project><ref>Frank</ref></project>
    <project>
        <ref>George</ref>
        <project><ref>George</ref></project>
        <project><ref>Jon</ref>
            <project><ref>Bob</ref></project>
            <project><ref>Frank</ref></project>
        </project>
    </project>
</doc>
```

A relational model for XML data (2)

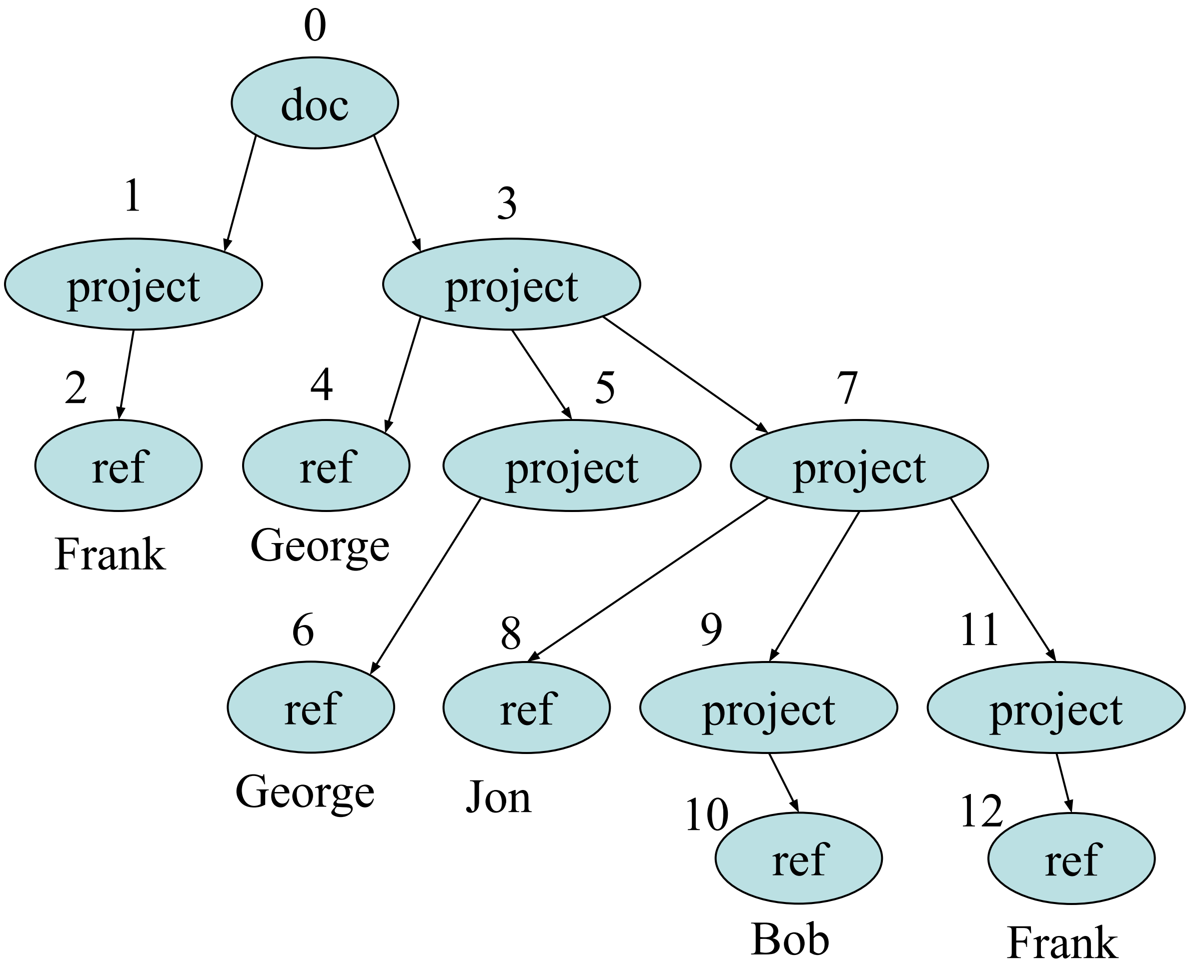


- Partial schema with same properties of data
- Based on the concept of list
- Ordered
- Nested

A relational model for XML data (3)

id	name
0	doc
1	project
2	ref
3	project
4	ref
5	project
6	ref
7	project
8	ref
9	project
10	ref
11	project
12	ref

id	child
0	1
0	3
1	2
3	4
3	5
3	7
5	6
7	8
7	9
7	11
9	10
11	12



Main limits of this solution

- XML was created to exchange data between applications and to represent data understandable by human beings
  - The example tables lose these properties. The data model is therefore more complex than the original format
- Some "reasonable" questions can not be written in SQL without using recursion, or they can be inefficient, requiring more time access to the same table, for example:

  “Find all representatives participating in the second project”

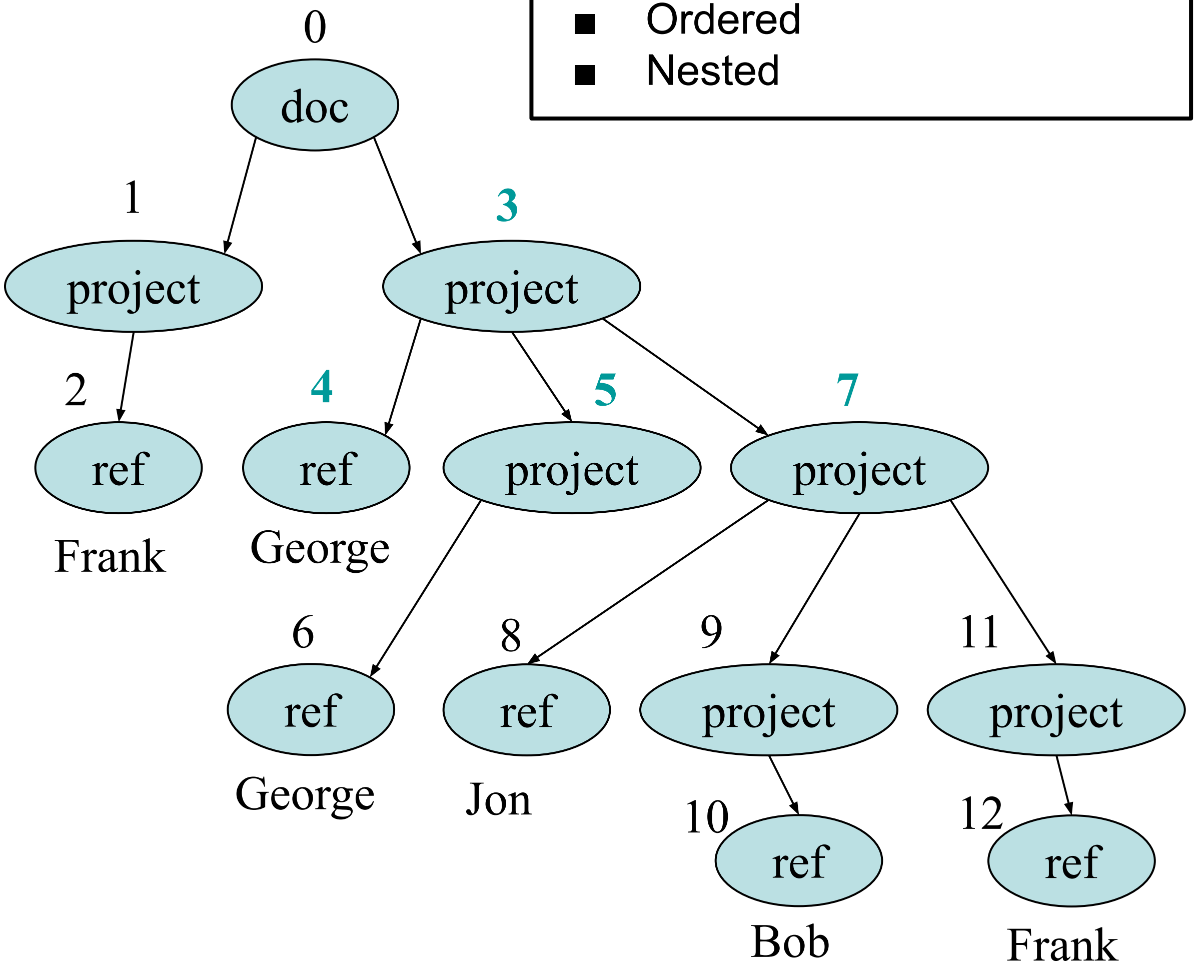
  - It should be mentioned that this road, suitably improved, has been covered by the scientific community with good results
  - So this is a way forward. However, the current trend is to develop specific systems for XML

Find all representatives participating in the second project

id	name
0	doc
1	project
2	ref
3	**project**
4	ref
5	project
6	ref
7	project
8	ref
9	project
10	ref
11	project
12	ref

id	child
0	1
0	3
1	2
3	**4**
3	**5**
3	**7**
5	6
7	8
7	9
7	11
9	10
11	12

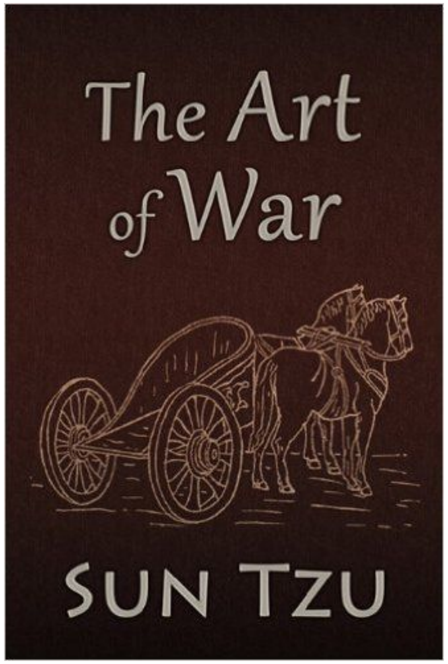
- Partial schema with same properties of data
- Based on the concept of list
- Ordered
- Nested



Limits of the relational model (1)

- Let’s consider a book and its relational representation

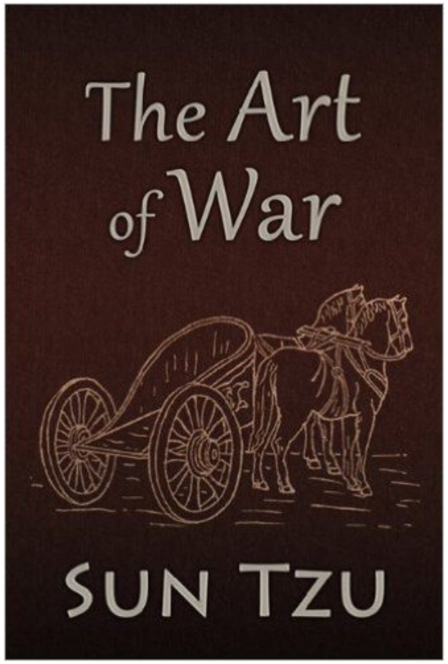
Author	Title	Birth	Text
Sun Tzu	The Art of War	544 b.c.	The supreme art of war is ...



Limits of the relational model (2)

- Let’s consider a book and its relational representation

Author	Title	Birth	Text
Sun Tzu	The Art of War	544 b.c.	The supreme art of war is ...

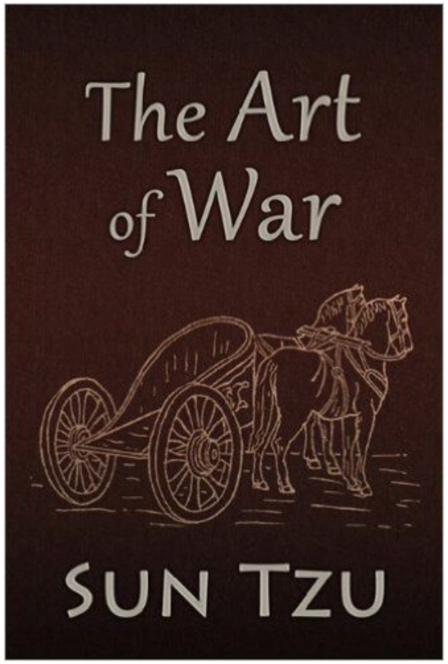


TEXT field contains thousands character without any structure (*CLOB, character large object, data type stores variable-length character data*)

Limits of the relational model (3)

- Let’s consider a book and its relational representation

Author	Title	Birth	Text
Sun Tzu	The Art of War	544 b.c.	The supreme art of war is ...



Death?

Description?

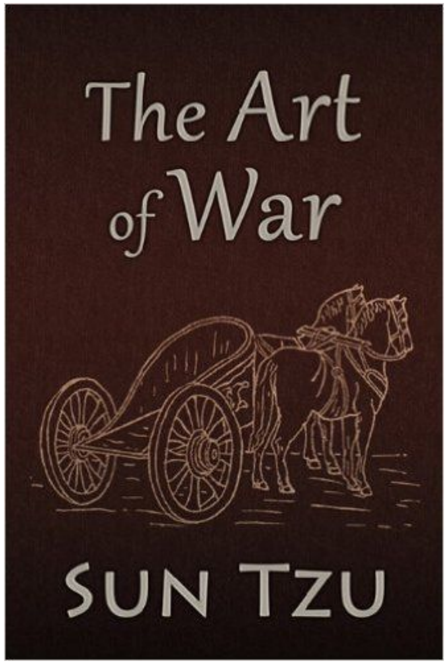
In order to add other info, we need to modify the structure of the table

Limits of the relational model (4)

- Let’s consider a book and its relational representation

Sun Tzu	The Art of War	544 b.c.	The supreme art of war is ...

Death? Birth? Publication date?



If we want to exchange data with other applications, we need to send over also the structure, together with the data, otherwise data could be unintelligible

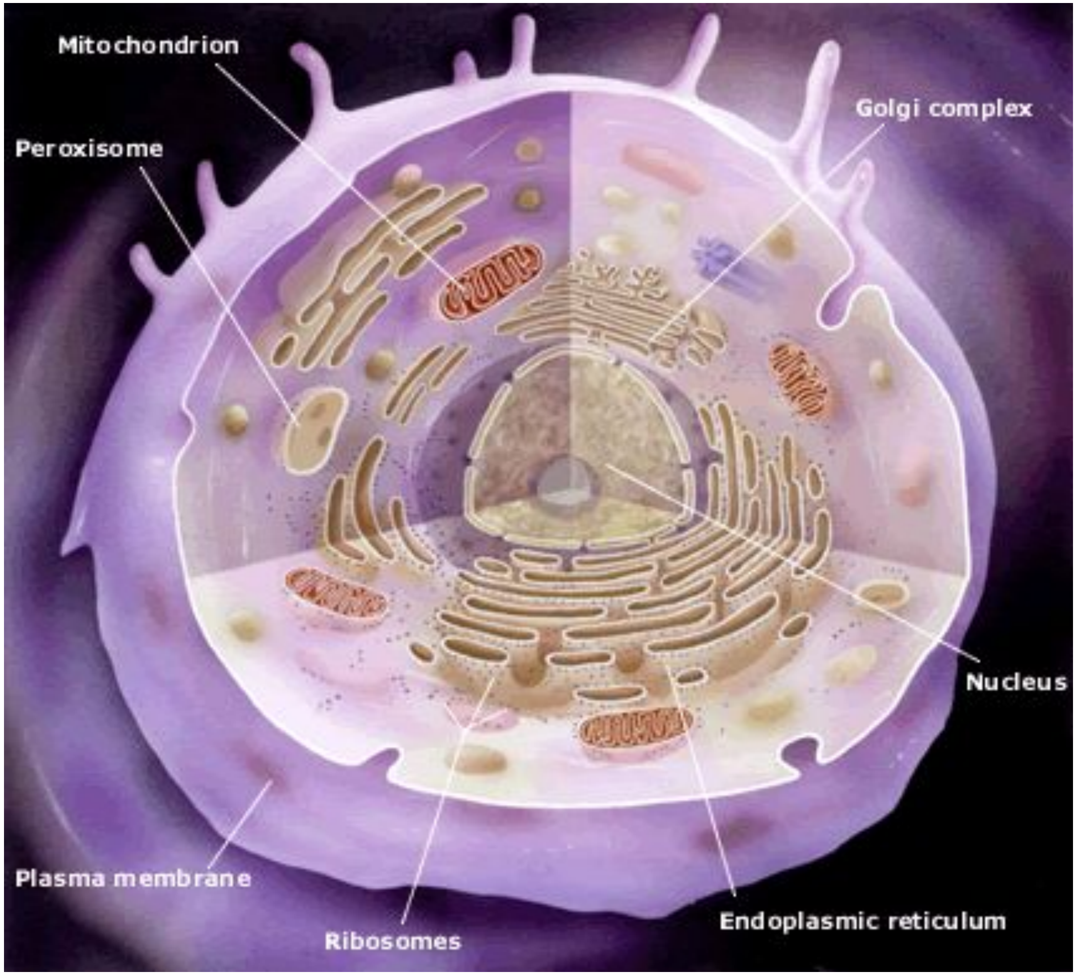
Properties of semi-structured data (1)

- The structure is irregular or partial

```
<html>
<head><title>The Art of War</title></head>
<body>
<h1>The Art of War</h1>
<h2>Chapter 1</h2>
The supreme art of war is ...<br/>
…
<img src=“img/war_picture_1.jpg”/>
…
</body>
</html>
```

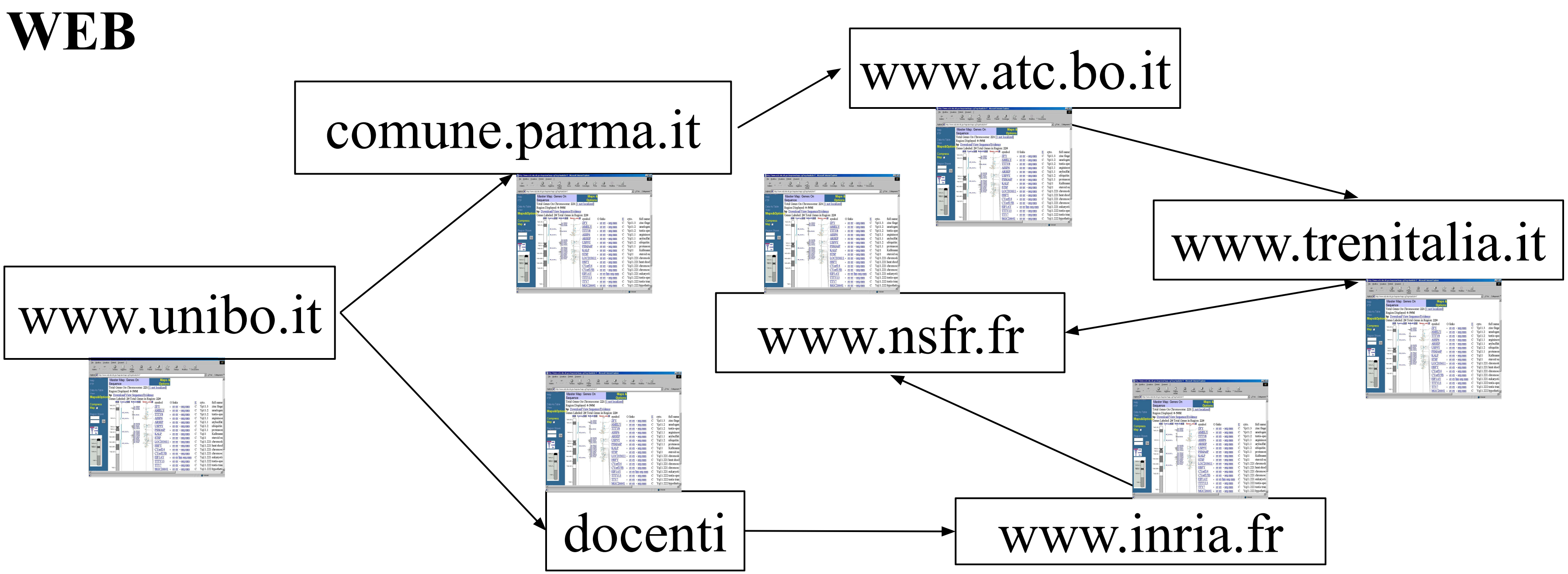
Properties of semi-structured data (2)

- Schema is built *a posteriori* (data guide)



Properties of semi-structured data (3)

- Schema is very broad



WEB

www.unibo.it

comune.parma.it

docenti

www.atc.bo.it

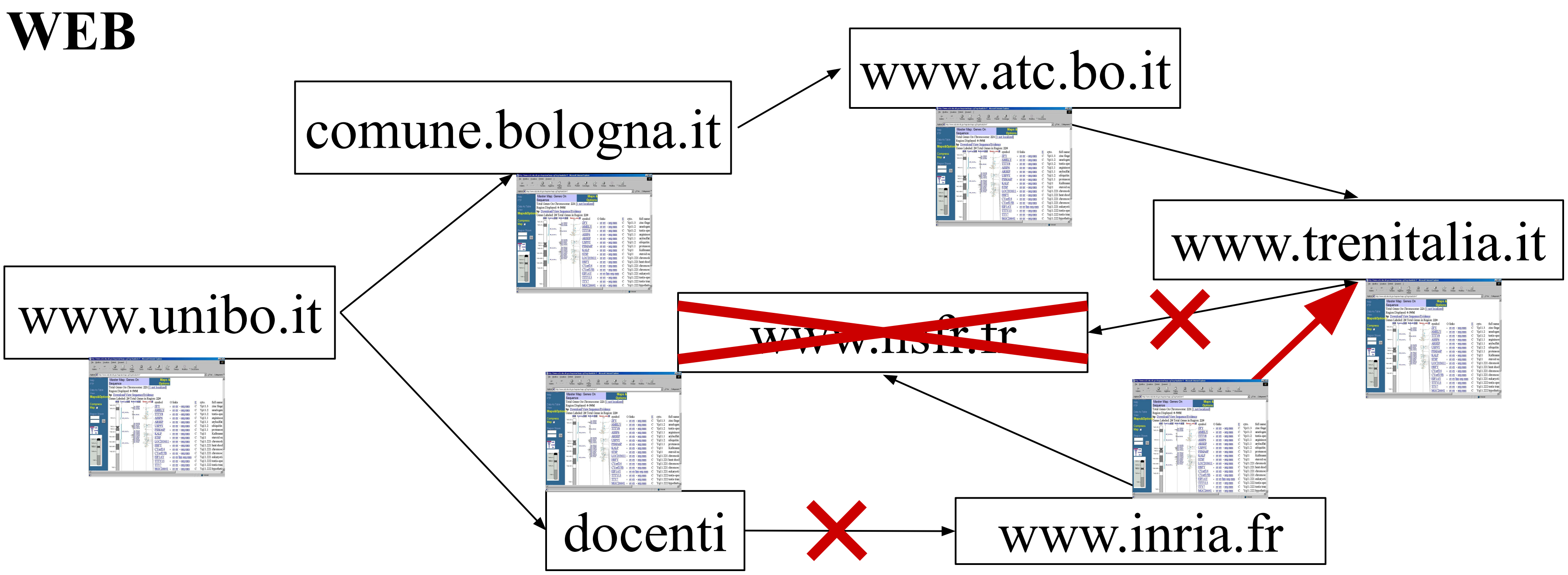
www.nsfr.fr

www.trenitalia.it

www.inria.fr

Properties of semi-structured data (4)

- The schema evolves rapidly (data as well)



In summary

- The structure is irregular or partial
- The scheme is constructed *a posteriori* (data guides)
- The scheme is very broad
- The scheme is rapidly evolving
- The differences between schema and data are not significant
  - The scheme is changed
  - The scheme is communicated with the data
  - The scheme does not impose constraints unappealable
  - The questions (query) also concern the scheme
- In the case of XML, they are relevant order and the mutual nesting of data
- Being ordered, the data are represented by lists

Unstructured data

Data without schema

- In XML and semi-structured data, the scheme has the characteristics that we saw, but it is still present
  - For example, a query in XPath (XML Path Language) on an XML document typically accesses tags of the items, that is, the schema
- If the **schema is not present**, as in the case of multimedia objects and only narrative text files, the data management change significantly
  - The main discipline that studies this data is called **Information Retrieval**
- No schema data, or which typically do not use the scheme, are of great importance: just think of the Internet and search engines, which are mostly of Web Information Retrieval systems

Simple queries

- Despite more complex query languages they have been proposed, in the majority of cases the queries on unstructured data (mostly on textual data) are very simple, usually composed of lists of keywords, like

  “Return documents that contain the word «battle».”

- … unlike to

```
SELECT name, COUNT(DISTINCT project), SUM(months)
FROM Person NATURAL JOIN Allocation
WHERE name LIKE ‘M%’ AND Age > 40
GROUP BY name
```

Boolean results and ranking (1)

- In a relational database, queries express precise requirements, and each tuple of the solution satisfies those requirements
- The construction of the response using a relation follows therefore a **Boolean** model: a tuple is *present* or is *not present* in the result
- Given the nature of the questions, given the large amount of possible answers, and given that several documents can respond more or less well to the requirements expressed in the question, **in Information Retrieval a Boolean model is often not usable**

Boolean results and ranking (2)

- Given the gap between data and information, due to the ambiguity of data, often you can not accurately determine whether a result is completely or not at all relevant
- The results are then ranked by degree of **relevance** (see for example Google), and the user admits possible “errors”
- Since it is usually not possible to return **correct** and **complete** results, as it happens in the case of structured data, there are metrics to describe the quality of a result

“Return the documents that have as an argument the war”

Know thy self, know thy enemy. A thousand battles, a thousand victories.

Sun Tzu

?

Relational data and unstructured data: comparison

Relational	Unstructured
Clear distinction between schema and data	No schema
Query language	Search language
Boolean model (correctness, completeness)	Ranking based model
Partial updates and queries	Total updates

In summary

- From the examples shown above emerges that the properties of data, queries and results are significantly different from those found in relational systems
- In addition to sorting by relevance, the result of a query on unstructured data typically does not provide for the manipulation of data, but only the selection of some of them
- Even in this case, as and more than for the semi-structured data, there is therefore the need to use different models and systems
- Note that well known DBMSs have already integrated some capabilities derived from Information Retrieval, such as indexing of columns that contain only text (CLOB, character large object) or columns for multimedia data (BLOB, binary large object)