

Introduzione a XML e basi di dati

Obiettivi

- Rapidissimo richiamo di XML.
- Come e perché gestire dati XML in un DBMS relazionale (quando si potrebbero utilizzare anche tabelle).
- Classificazione strutturale dei dati e importanza teorica di XML (quando le tabelle non sono più adatte).

Introduzione a XML e basi di dati

Richiamo di XML

Un esempio di documento XML

```
<?xml version="1.0"?>
<?tex doctype[report] ?>
<doc isbn="2-266-04744-2">
  <!-- manca l'editore! -->
  <autore>T. Harris</autore>
  <titolo xml:lang="en">The silence of the lambs</titolo>
  <titolo xml:lang="fr">Le silence des agneaux</titolo>
  <commento>
    Un libro pieno di <i>suspance</i>.
  </commento>
  <prezzo valuta="euro">7</prezzo>
</doc>
```

Markup e *character data*

```
<?xml version="1.0"?>
<?tex doctype[report] ?>
<doc isbn="2-266-04744-2">
  <!-- manca l'editore! -->
  <autore>T. Harris</autore>
  <titolo xml:lang="en">The silence of the lambs</titolo>
  <titolo xml:lang="fr">Le silence des agneaux</titolo>
  <commento>
    Un libro pieno di <i>suspance</i>.
  </commento>
  <prezzo valuta="euro">7</prezzo>
</doc>
```

Prologo

```
<?xml version="1.0"?>
```

```
<?tex doctype[report] ?>
```

PROLOGO

```
<doc isbn="2-266-04744-2">
```

```
  <!-- manca l'editore! -->
```

```
  <autore>T. Harris</autore>
```

```
  <titolo xml:lang="en">The silence of the lambs</titolo>
```

```
  <titolo xml:lang="fr">Le silence des agneaux</titolo>
```

```
  <commento>
```

```
    Un libro pieno di <i>suspance</i>.
```

```
  </commento>
```

```
  <prezzo valuta="euro">7</prezzo>
```

```
</doc>
```

Corpo del documento

```
<?xml version="1.0"?>
```

```
<?tex doctype[report] ?>
```

```
<doc isbn="2-266-04744-2">
```

```
<!-- manca l'editore! -->
```

```
<autore>T. Harris</autore>
```

```
<titolo xml:lang="en">The silence of the lambs</titolo>
```

```
<titolo xml:lang="fr">Le silence des agneaux</titolo>
```

```
<commento>
```

```
    Un libro pieno di <i>suspance</i>.
```

```
</commento>
```

```
<prezzo valuta="euro">7</prezzo>
```

```
</doc>
```

CORPO DEL
DOCUMENTO

XML e Schema

- I dati relazionali sono *vincolati* da uno Schema.
- Uno Schema definisce quante colonne ha ogni tabella, quali valori possono appartenere ad una colonna e altri tipi di vincoli.
- Un documento XML ha invece in generale l'unico vincolo di essere sintatticamente corretto (tag aperti e poi chiusi, non innestati, nomi *case-sensitive*, attributi tra apici...).
- E' possibile specificare uno Schema anche per un documento XML, tramite un DTD o un documento XMLSchema.
- Una **dichiarazione del tipo di documento** (DTD) o un XMLSchema hanno il doppio scopo di *vincolare* e *completare* documenti.

XML e Schema

```
<?xml version="1.0" encoding="UTF-8"?>  
<!DOCTYPE saluti [  
  <!ELEMENT saluti (#PCDATA)>  
>  
<saluti>Ciao, mondo!</saluti>
```

```
<?xml version="1.0"?>  
<!DOCTYPE saluti SYSTEM "ciao.dtd">  
<saluti>Ciao, mondo!</saluti>
```

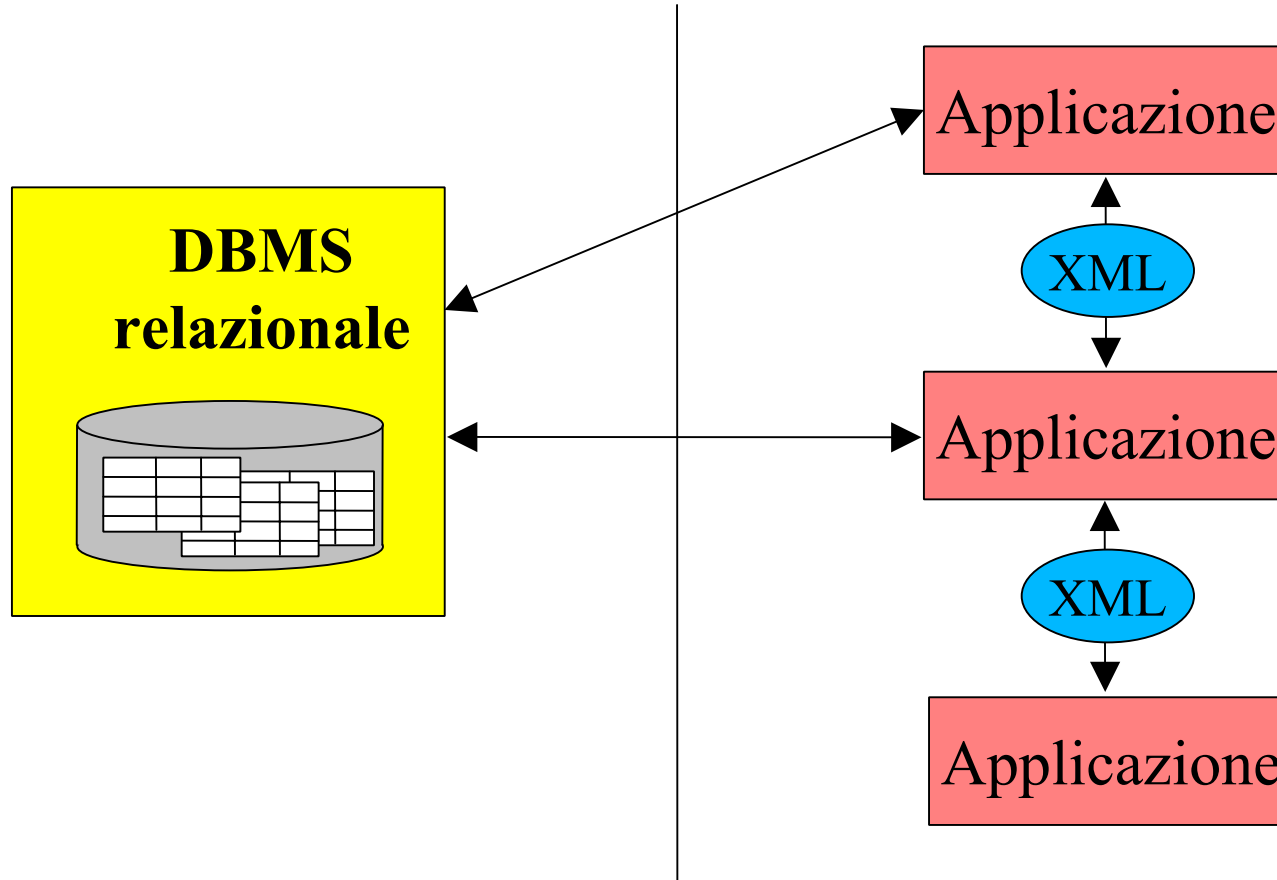
Introduzione a XML e basi di dati

Utilizzo di XML con sistemi per la gestione di basi di dati relazionali

Formato dei dati (storage/application)

Memorizzazione dei dati

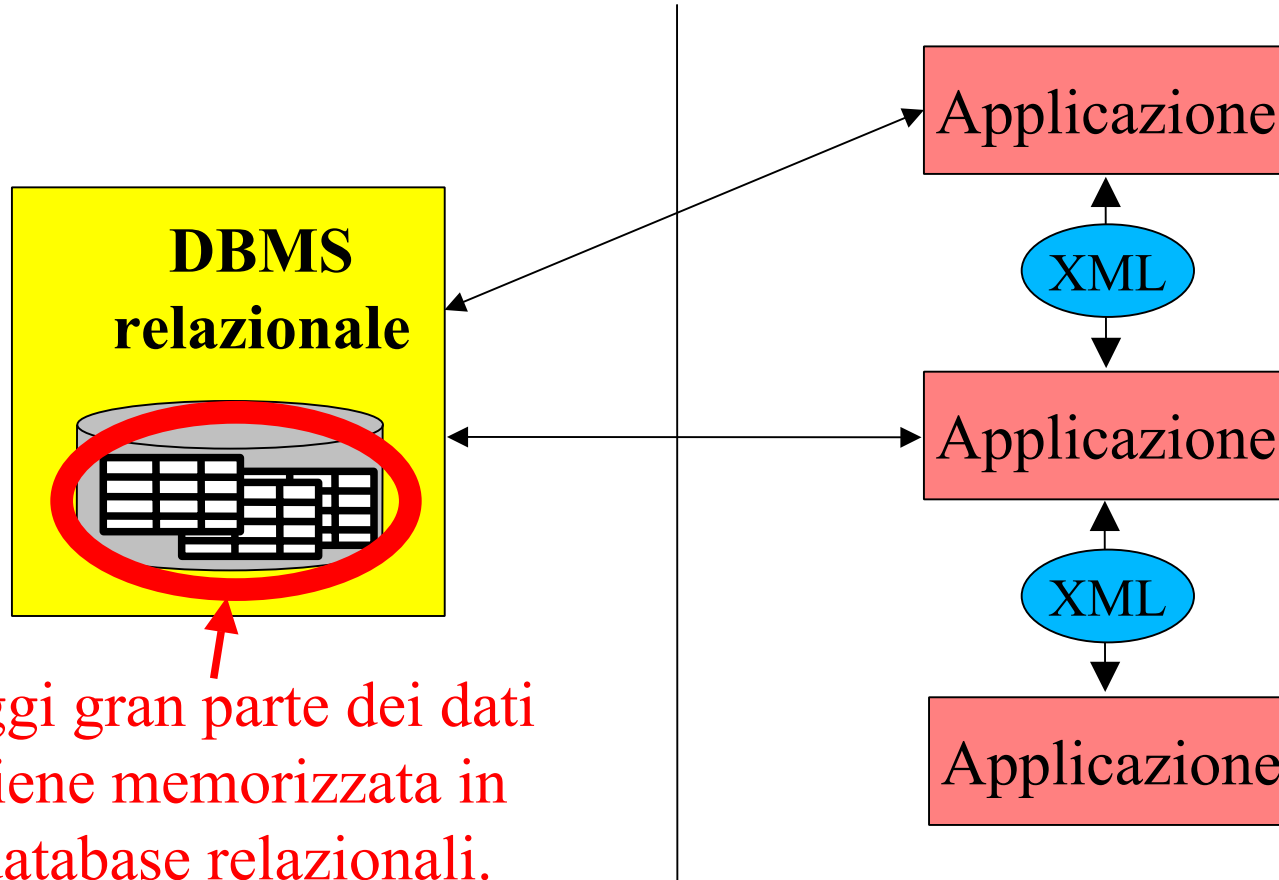
Utilizzo dei dati



Formato dei dati (storage/application)

Memorizzazione dei dati

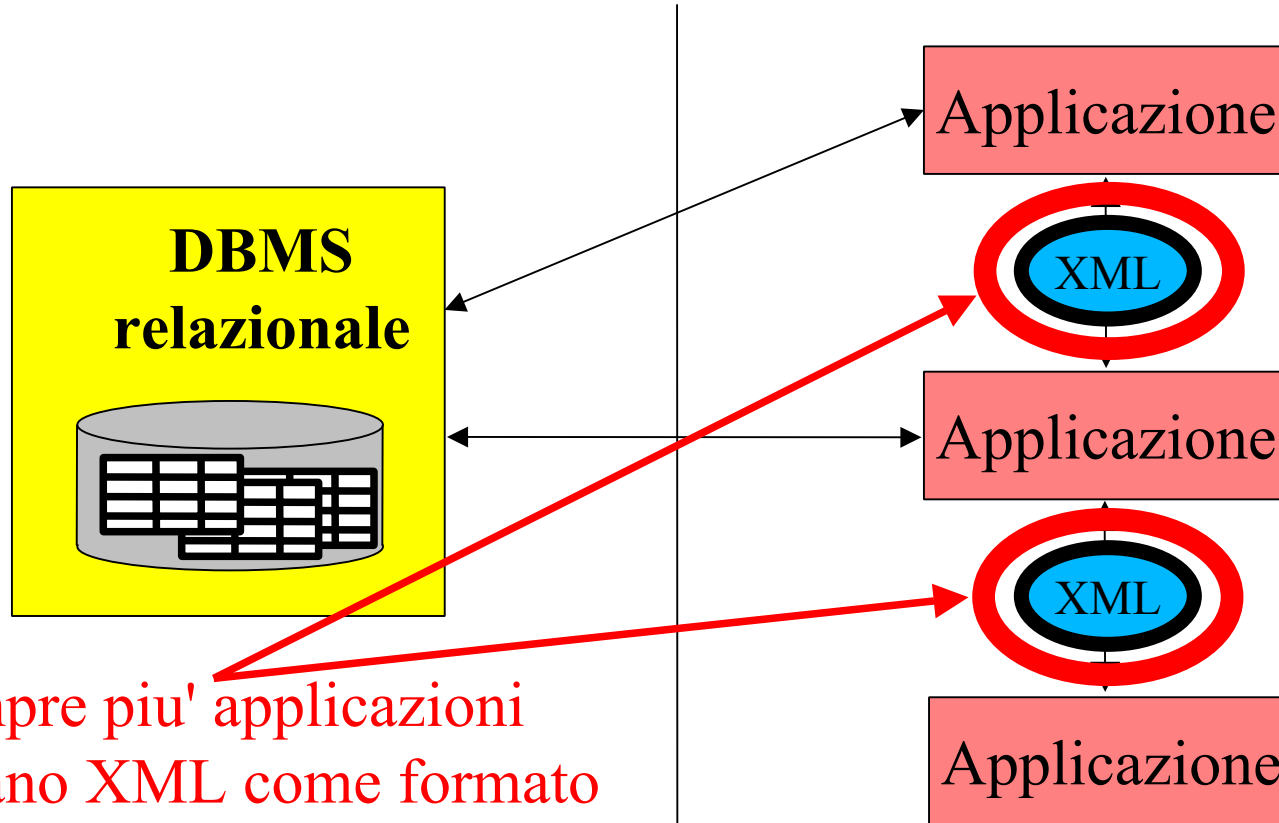
Utilizzo dei dati



Formato dei dati (storage/application)

Memorizzazione dei dati

Utilizzo dei dati

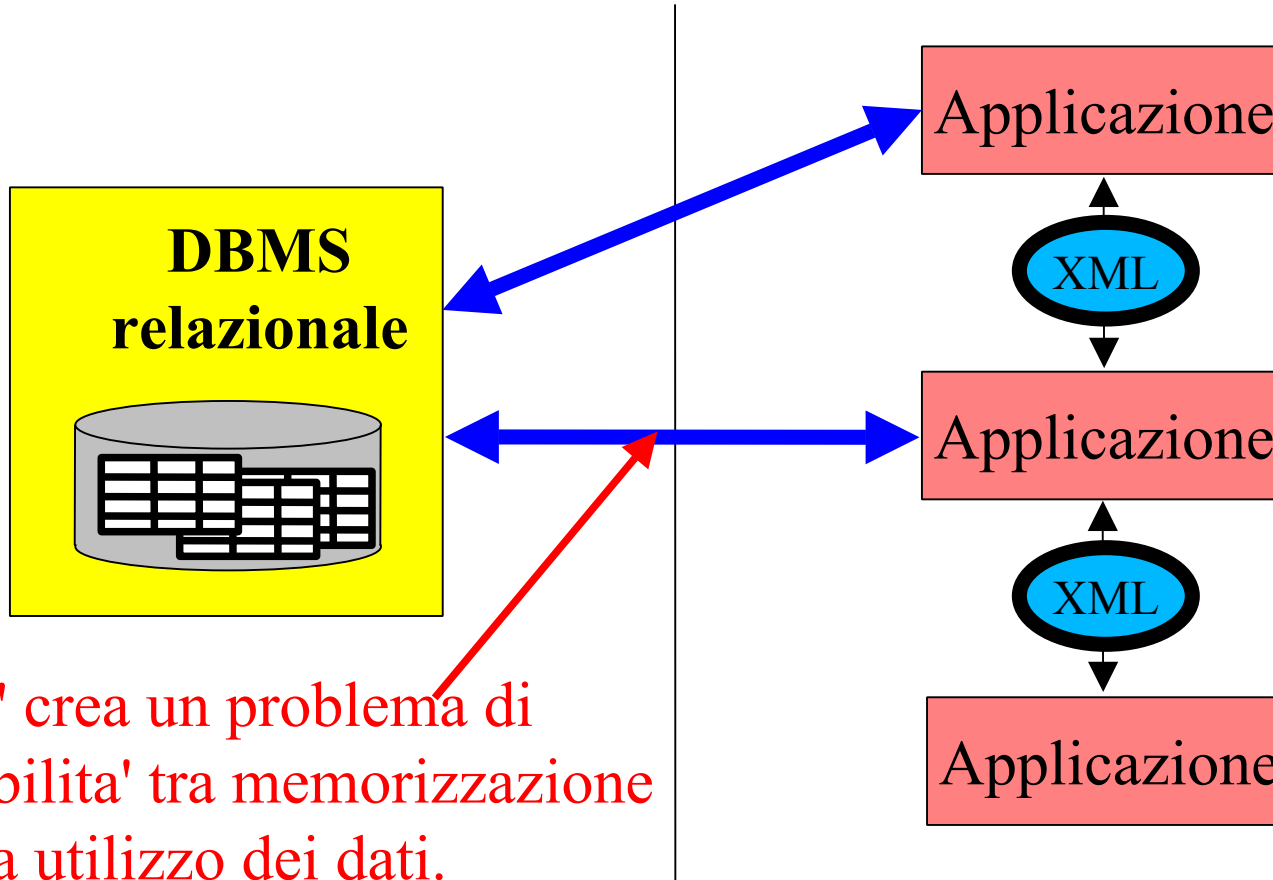


Sempre piu' applicazioni utilizzano XML come formato di scambio e manipolazione dei dati.

Formato dei dati (storage/application)

Memorizzazione dei dati

Utilizzo dei dati

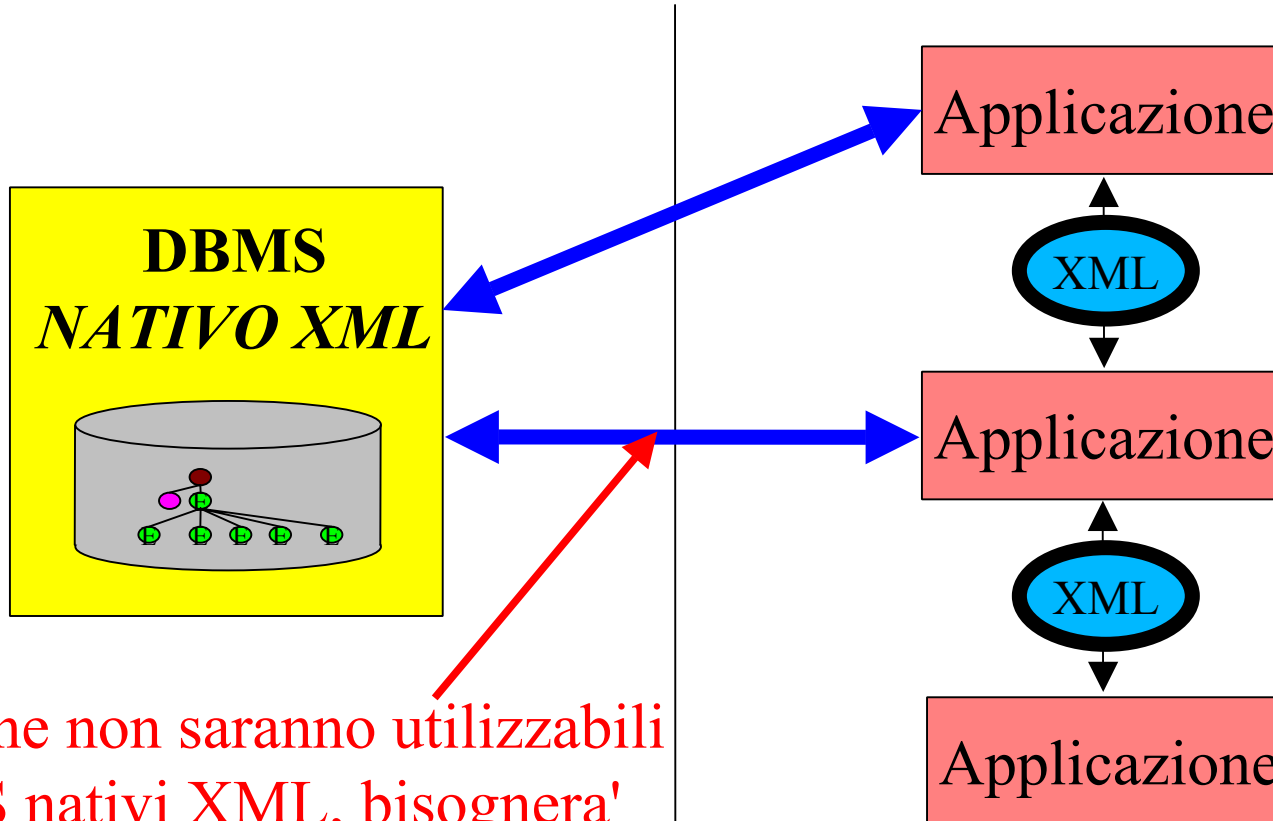


Cio' crea un problema di compatibilita' tra memorizzazione a utilizzo dei dati.

Formato dei dati (storage/application)

Memorizzazione dei dati

Utilizzo dei dati



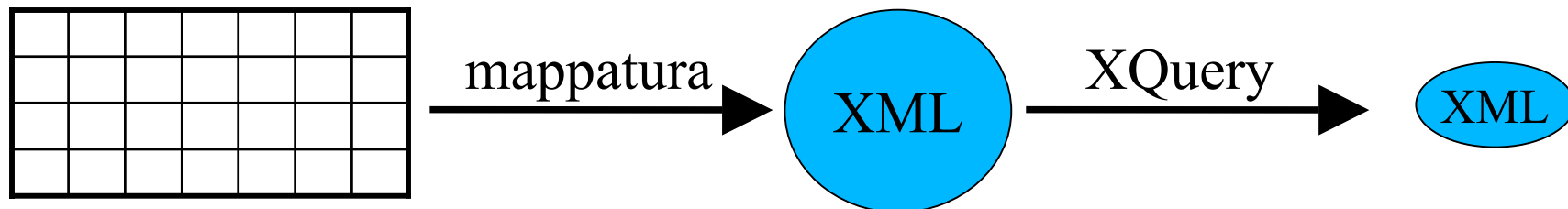
Fino a che non saranno utilizzabili
DBMS nativi XML, bisognerà
interfacciare tabelle e XML.

Operazioni

- In questo contesto, sono due le operazioni che vorremmo poter compiere:
 - Estrarre XML da una o piu' tabelle relazionali.
 - Memorizzare XML in una (o piu') tabelle relazionali.
- Estrarre XML a partire da una tabella e' semplice, perche' qualsiasi dato rappresentabile in una tabella e' anche rappresentabile in XML.

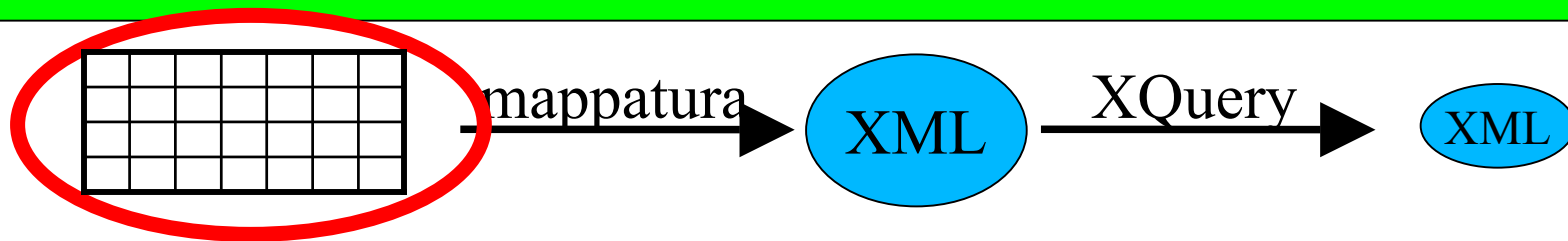
Estrarre XML – 1

- Una possibile procedura e' la seguente:
 - Rappresentare la tabella in XML (mappatura).
 - Estrarre dati utilizzando tecnologie XML (XQuery).



Introduzione a XML e Basi di Dati

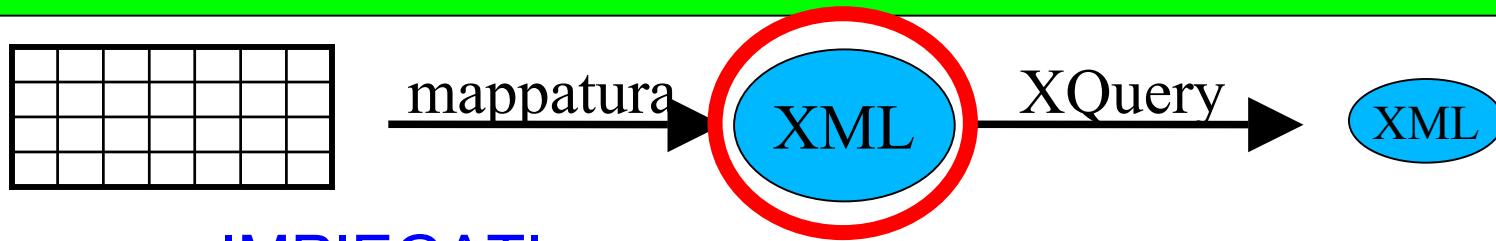
© Matteo Magnani, Danilo Montesi – Università di Bologna



IMPIEGATI			
<i>ID</i>	<i>NOME</i>	<i>COGNOME</i>	<i>SALARIO</i>
imp0001	Mario	Rossi	20000
imp0002	Giorgio	Bianchi	18000

Introduzione a XML e Basi di Dati

© Matteo Magnani, Danilo Montesi – Università di Bologna



<IMPIEGATI>

<row>

<ID>imp0001</ID>

<NOME>Mario</NOME>

<COGNOME>Rossi</COGNOME>

<SALARIO>20000</SALARIO>

</row>

<row>

<ID>imp0002</ID>

<NOME>Giorgio</NOME>

<COGNOME>Bianchi</COGNOME>

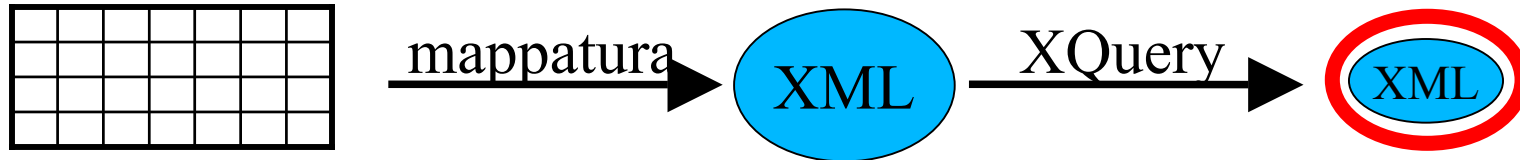
<SALARIO>18000</SALARIO>

</row>

</IMPIEGATI>

Introduzione a XML e Basi di Dati

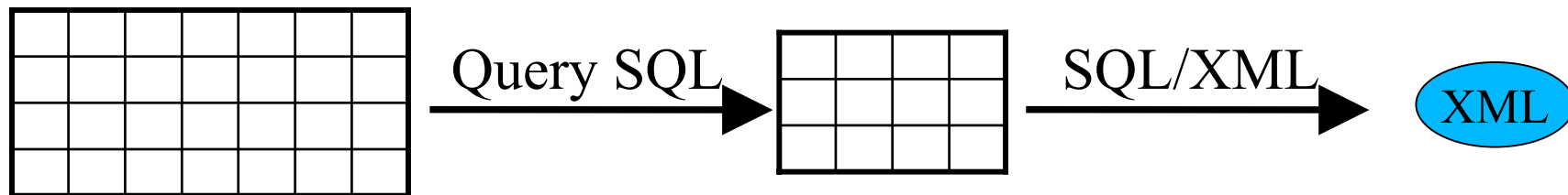
© Matteo Magnani, Danilo Montesi – Università di Bologna



```
<IMPIEGATIMOLTOPAGATI>  
  <IMPIEGATO>  
    <NOME>Mario</NOME>  
    <COGNOME>Rossi</COGNOME>  
  </IMPIEGATO>  
</IMPIEGATIMOLTOPAGATI>
```

Estrarre XML – 2

- Una seconda possibilita' consiste nel:
 - Estrarre dati dalla tabella (SQL).
 - Trasformare questi dati in XML (SQL/XML).



Esempio di interrogazione SQL/XML

```
SELECT id, XMLELEMENT(NAME "imp", i.nome)  
FROM IMPIEGATI AS i
```

<i>id</i>	<i>risultato</i>
imp0001	<imp>Ma rio</imp>
imp0002	<imp>Giorgio</imp>
imp0003	<imp>Luca</imp>
imp0004	<imp>Gia nni</imp>

















Inserire XML – 1

- Un documento XML puo' essere considerato come un tipo di dato di SQL.
- In questo caso, un intero documento viene memorizzato in un attributo (colonna) di una tabella.

						XML		
						XML		
						XML		
						XML		


Inserire XML – 2

- Quando la struttura di un documento e' abbastanza regolare, questo puo' essere spezzato in frammenti e memorizzato a pezzi.

Esempio di inserimento

```
<impiegato>  
  <ID>imp0001</ID>  
  <COGNOME>Rossi</COGNOME>  
  <SALARIO>20000</SALARIO>  
</impiegato>  
<impiegato>  
  <ID>imp0002</ID>  
  <COGNOME>Bianchi</COGNOME>  
  <SALARIO>18000</SALARIO>  
</impiegato>
```



<i>ID</i>	<i>COGNOME</i>	<i>SALARIO</i>
imp0001	Rossi	20000
imp0002	Bianchi	18000

Struttura dei dati

- Abbiamo visto come talvolta XML possa essere tranquillamente utilizzato in sistemi relazionali.
- Cio' accade quando XML presenta una struttura regolare.
- Questo ci suggerisce come diversi dati possano avere strutture di tipo differente, piu' o meno adatte ad essere rappresentate tramite un particolare modello (ad esempio relazionale).

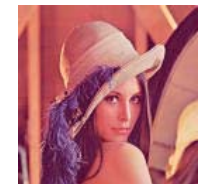
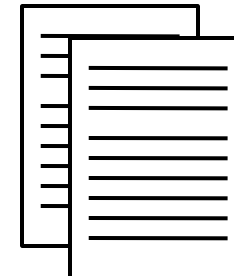
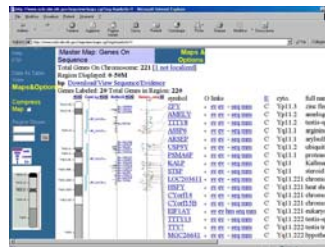
Una classificazione strutturale dei dati

STRUTTURATI

NON STRUTTURATI

ID	Nome	Cognome
00000001	Maria	Rossi

ID	telefono
00000001	051 123456
00000001	333 333333



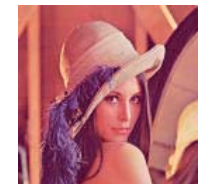
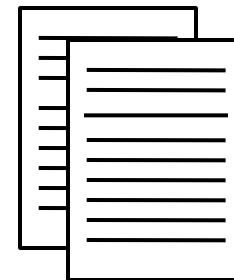
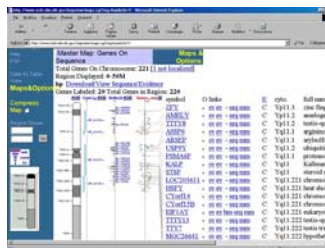
Una classificazione strutturale dei dati

ID	Nome	Cognome
00000001	Mario	Rossi
00000002	Giorgio	Verdi

ID	telefono
00000001	051 123456
00000001	333 333333



**DATI
STRUTTURATI
(SCHEMA)**



Una classificazione strutturale dei dati

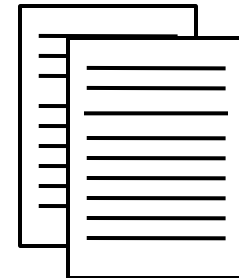
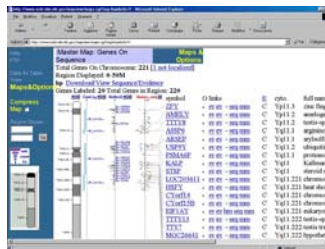


vu45s89gysJPGi
8gbyygsvs954gy
4598y9syg5vts9
4lygs98yg9s45y
g584gyt459gyg4
...



ID	Nome	Cognome
00000001	Maria	Rossi

ID	telefono
00000001	051 123456
00000001	333 333333



**RAW
DATA**

Una classificazione strutturale dei dati

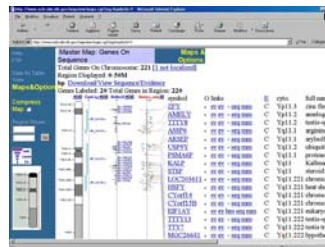
E' molto meglio essere bello che buono; ma e' meglio essere buono piuttosto che brutto.
O.Wilde

L'ETERNO
Tra un fiore colto e l'altro donato
l'inesprimibile nulla
G.Ungaretti

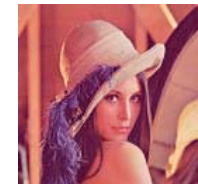


ID	Nome	Cognome
00000001	Maria	Rossi

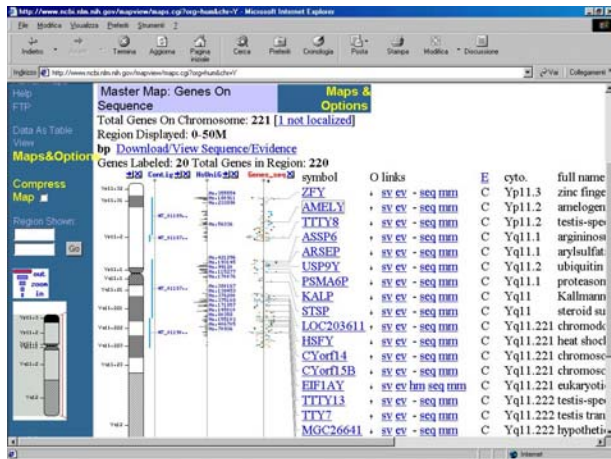
ID	telefono
00000001	051 123456
00000001	333 333333



**DATI
SENZA
SCHEMA**



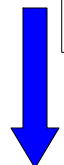
Una classificazione strutturale dei dati



```

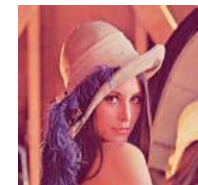
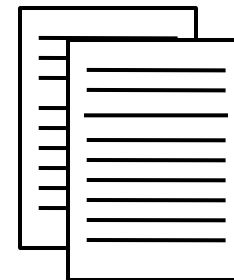
<html>
<title>La divina commedia</title>
Nel mezzo del cammin...
<img/></html>

```



ID	Nome	Cognome
00000001	Mario	Rossi
00000002		
ID	telefono	
00000001	051 123456	
00000001	333 333333	

**DATI CON
STRUTTURA
PARZIALE**

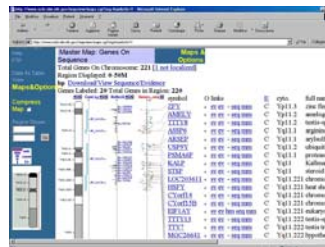


Moltissimi dati stanno nel mezzo

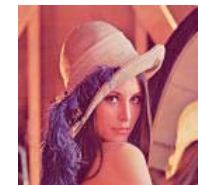
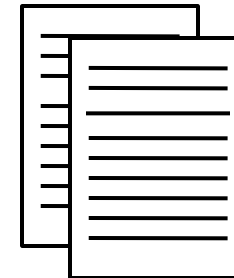
- Molti dati non sono classificabili né come dati strutturati, né come *raw data*.
- Il modello relazionale non è adatto a memorizzare questo tipo di dati.

ID	Nome	Cognome
00000001	Maria	Rossi

ID	telefono
00000001	051 123456
00000001	333 333333



ID	Nome	Telefono
00000001	Maria	051 123456
00000001	Maria	333 333333



Limiti del modello relazionale

- Il principale limite del modello relazionale consiste nella **necessità di conoscere lo schema a priori**.
- Questo non è sempre possibile, in quanto lo **schema può cambiare nel tempo** o essere in parte **assente**.
- A volte è necessario **comunicare lo schema**.
- Molti dati non sono schematizzabili a priori (dati **semi-strutturati (SS)**).

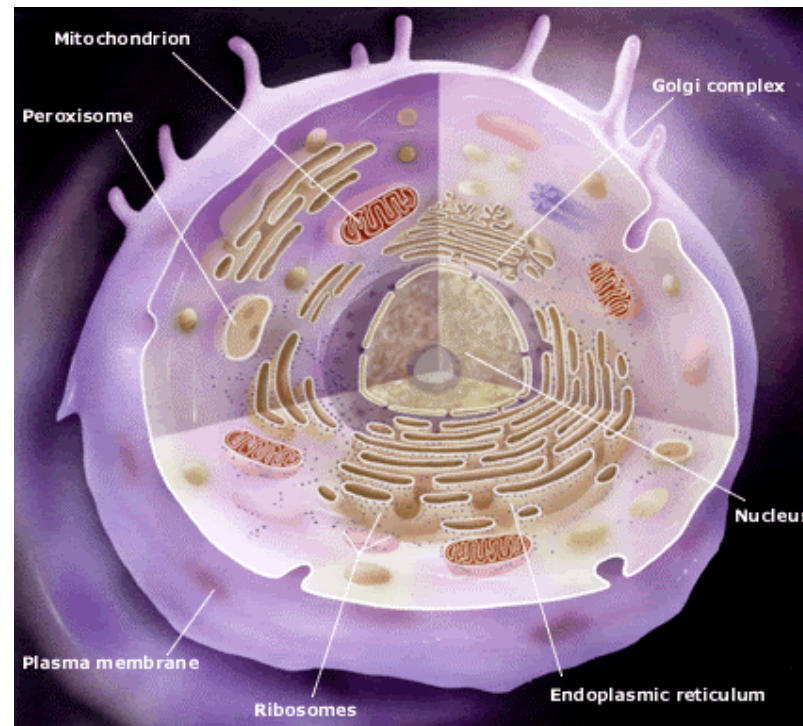
Caratteristiche dei dati SS

- La struttura è irregolare o parziale.

```
<html>
<head><title>La divina commedia</title></head>
<body>
<h1>La Divina Commedia</h1>
<h2>Canto I</h2>
Nel mezzo del cammin di nostra vita<br/>
...
<img src=“img/illustrazione1_1.jpg”>
...
</body>
</html>
```

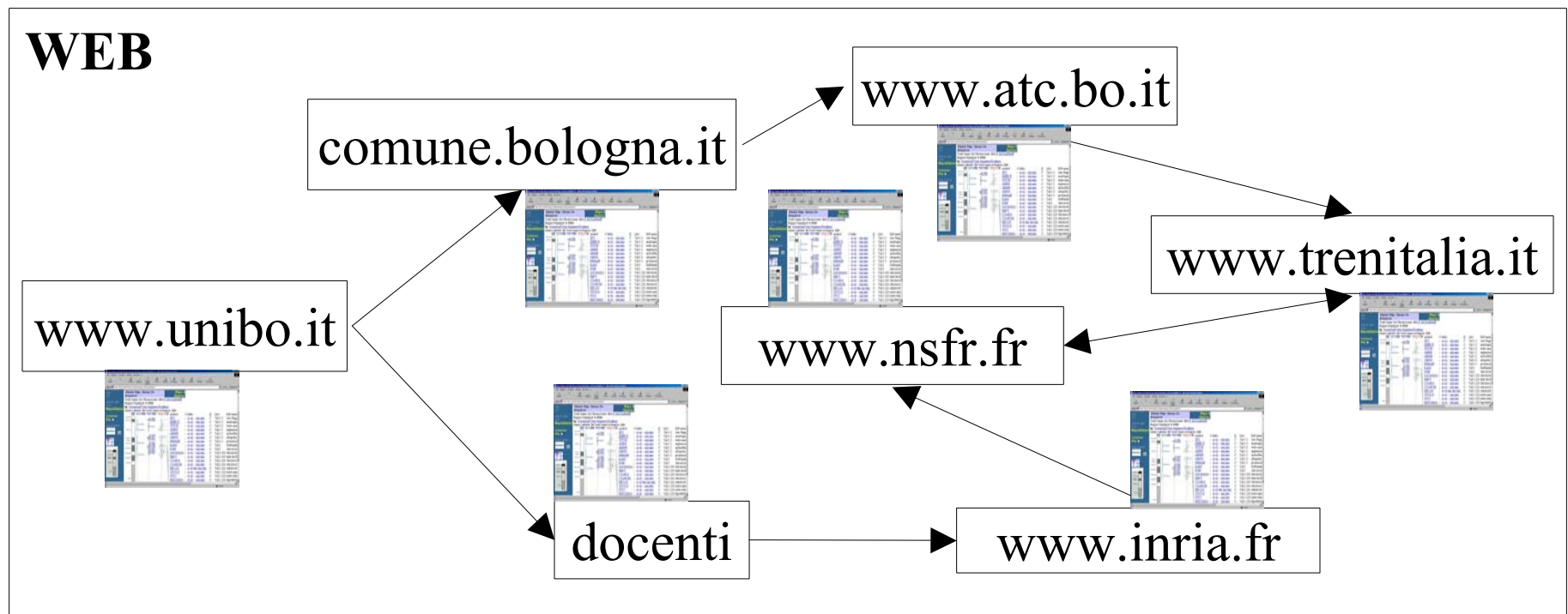
Caratteristiche dei dati SS

- Lo schema è costruito a posteriori (*data guide*).



Caratteristiche dei dati SS

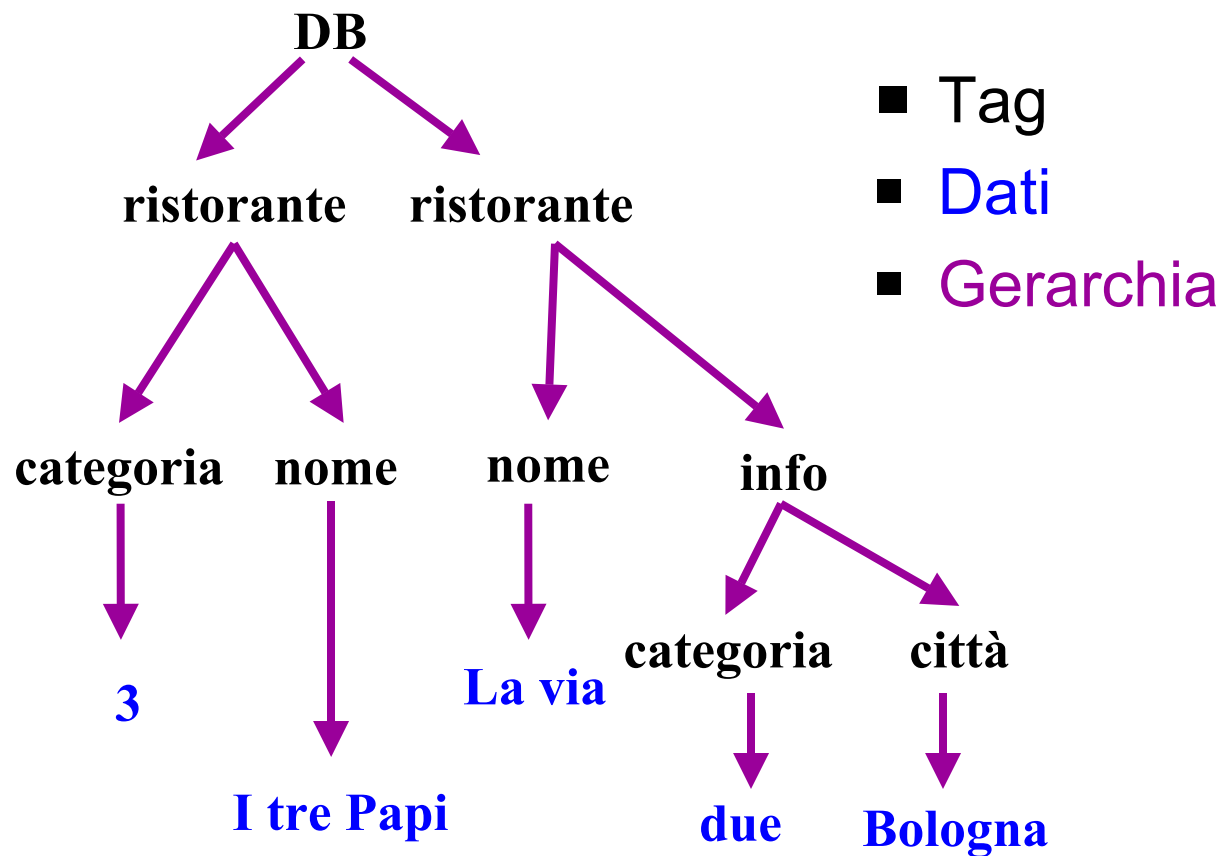
- Lo schema è molto ampio.



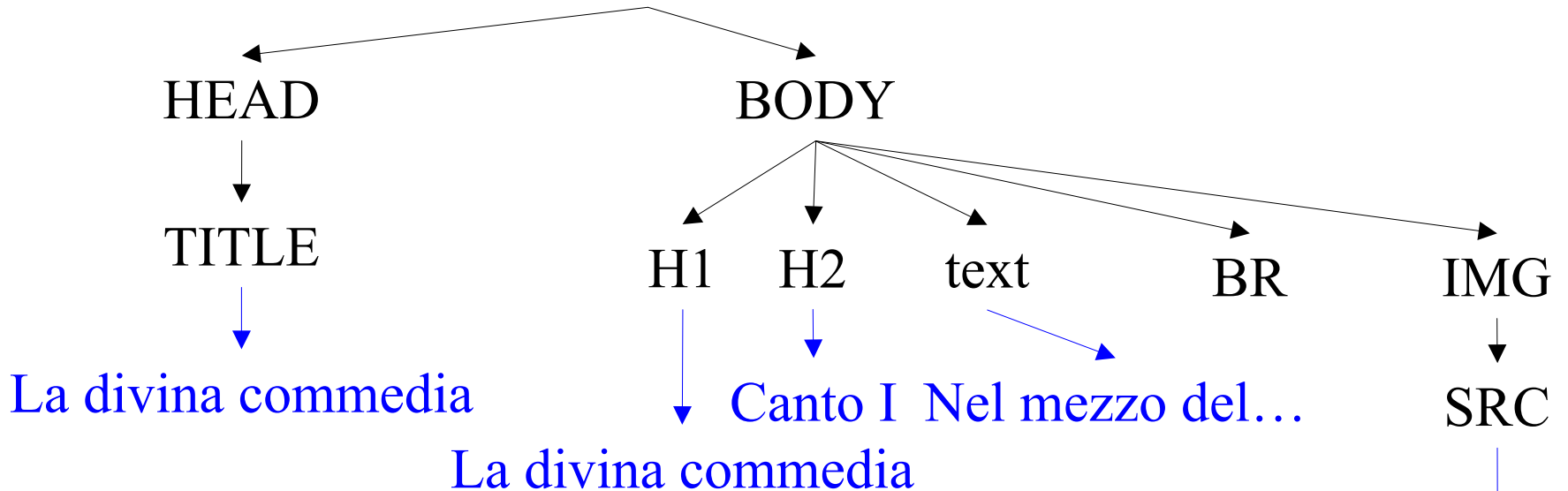
Riassumendo

- La struttura è irregolare o parziale.
- Lo schema è costruito a posteriori (*data guide*).
- Lo schema è molto ampio.
- Lo schema evolve rapidamente.
- Le differenze tra schema e dati non sono significative.
 - Lo schema viene modificato.
 - Lo schema viene comunicato insieme ai dati.
 - Lo schema non impone vincoli inappellabili.
 - Le interrogazioni riguardano anche lo schema.

Strumenti di memorizzazione

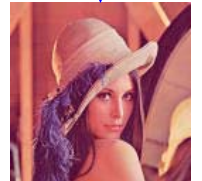
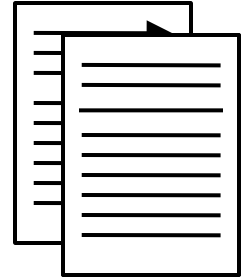
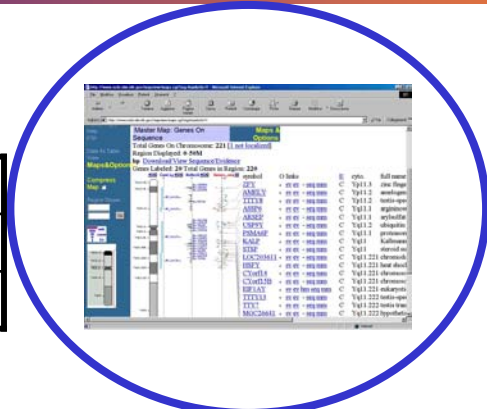


Applichiamo il modello HTML



ID	Nome	Cognome
00000001	Maria	Rossi

ID	telefono
00000001	051 123456
00000001	333 333333



Applichiamo il modello

TEXT

Titolo

Testo

Autore

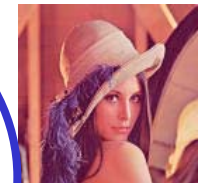
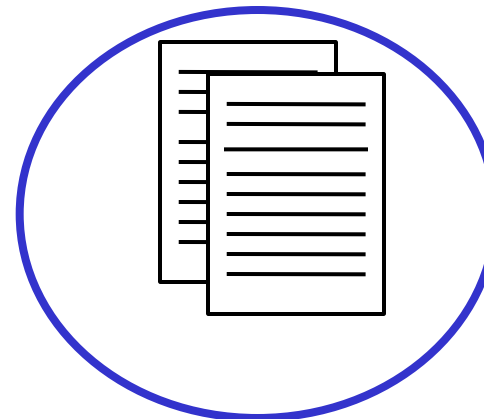
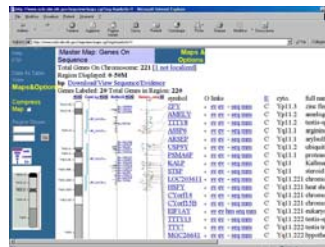
Tra un fiore colto e l'altro donato
l'inesprimibile nulla

L'ETERNO

G.Ungaretti

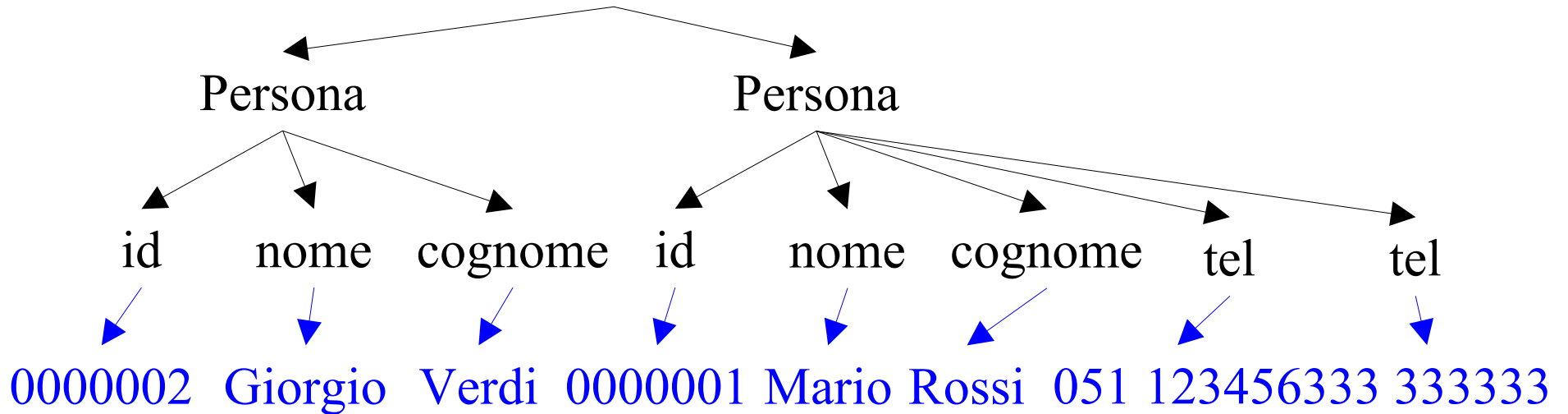
ID	Nome	Cognome
00000001	Maria	Rossi

ID	telefono
00000001	051 123456
00000001	333 333333



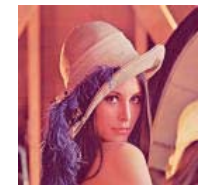
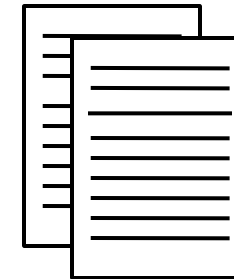
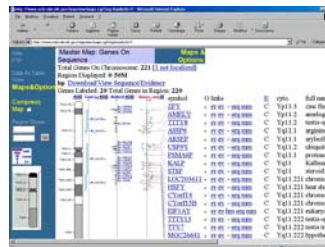
Applichiamo il modello

DB



ID	Nome	Cognome
00000001	Mario	Rossi
00000002		

ID	telefono
00000001	051 123456
00000001	333 333333



Riassumendo

- Molti dati non sono facilmente rappresentabili tramite il modello relazionale.
- In alcuni di questi, detti *semi-strutturati*, le differenze tra dati e schema sono sfumate.
- Un modello per dati semi-strutturati si può basare su **tag**, **dati** e **gerarchie**, con l'eventuale aggiunta di **attributi**, come in XML.
- Un tale modello può descrivere anche dati strutturati.

Bibliografia

- XML Data Management
A.B.Chaudhri, A.Rashid, R.Zicari
Addison-Wesley (Capitoli 1 e 10)
- XML Corso di programmazione
H.M.Deitel, P.J.Deitel, T.R.Nieto, T.M.Lin,
P.Sadhu
APOGEO (Capitolo 5)

Webliografia (1)

Tutorial:

<http://www.html.it/xml/>

<http://www.xml.com/pub/a/98/10/guide0.html>

Home Page del consorzio W3C

<http://www.w3.org>

Specifica XML commentata (inglese)

<http://www.xml.com/axml/testaxml.htm>

Webliografia (2)

- Articolo introduttivo su XML/SQL:

<http://www.acm.org/sigmod/record/issues/0206/standard.pdf>

- Articolo su SQL/XML e XQuery (IBM):

<http://www.research.ibm.com/journal/sj/414/reinwald.pdf>