XML e basi di dati

1

XML e Database: il problema

- Problema:
  - è possibile/necessario memorizzare documenti XML in un DBMS?
  - Quale tecnologia è necessaria a questo scopo?
- Risposta:
  - è certamente possibile memorizzare e gestire documenti XML in un DBMS
  - la tecnologia necessaria a questo scopo dipende dal perché vogliamo gestire documenti XML in un DBMS

2

Tipologie di documenti XML

- Due possibili usi per documenti XML:
  - *Data Centric*: i documenti possono rappresentare lo strumento con il quale dati tradizionali (es. relazionali) vengono trasferiti su Web
    - XML come veicolo per trasporto di dati
    - Esempio: ordini di vendita, scheduling di voli, menù
  - *Document Centric*: l'informazione è rappresentata dal documento in sé
    - XML come modello per la rappresentazione dei dati
    - Esempio: libri, documenti in genere

3

Documenti Data Centric

- Struttura regolare
- livello di dettaglio piuttosto fine
- contenuto omogeneo
- l'ordine con cui gli elementi allo stesso livello appaiono è ininfluente
- Utilizzati per "machine consumption"
- Esempi: ordini di vendita, scheduling di voli, menù,…

4

Esempio: ordini di vendita

```
<Orders>
   <SalesOrder SONumber="12345">
     <Customer CustNumber="543">
       <CustName>ABC Industries</CustName>
       ...
     </Customer>
     <OrderDate>981215</OrderDate>
     <Line LineNumber="1">
       <Part PartNumber="123">
         <Description>
               Turkey wrench: Stainless steel, one piece...
         </Description>
         <Price>9.95</Price>
       </Part>
       <Quantity>10</Quantity>
     </Line>
     <Line LineNumber="2">
       ...
     </Line>
   </SaleOrder>
</Orders>
```

5

Documenti Document Centric

- Struttura irregolare
- Livello di dettaglio meno fine
- contenuto eterogeneo
- l'ordine degli elementi allo stesso livello è significativo
- in genere progettati per "human consumption"
- Esempi: libri, email, …

6

Product Description

```
<Product>
<Name>Turkey Wrench</Name>
<Developer>Full Fabrication Labs, Inc.</Developer>
<Summary>Like a monkey wrench, but not as big.</Summary>
<Description>
<Para>The Turkey wrench, which comes in both right- and left-
  handed versions ....</Para>
<Para>You can:</Para>
<List>
  <Item><Link URL="Order.htm">Order your turkey
  wrench</Link></Item>
  <Item><Link URL="Wrench.html">Read about wrenches</Link></Item>
  <Item><Link URL="catalog.zip">Download the
  catalog</Link></Item>
</List>
  ....
</Description>
</Product>
```

7

XML e DBMS

- Ciascuna tipologia di documenti richiede una particolare tecnologia per la sua gestione

data ⟺ Relational/object-oriented DB

document ⟺ DB basato su XML (XML è il modello dei dati)

8

XML e DBMS

- Due categorie di DBMS:
  - XML-Native DBMS:
    - comprendono un insieme di nuovi sistemi la cui architettura è stata progettata per supportare totalmente le funzionalità necessarie alla gestione di documenti XML
    - tecnologia non ancora matura
    - utili per Document Centric
    - Esempio:eXcelon
  - XML-Enabled DBMS:
    - comprendono tutti i DBMS che mantengono integra la propria architettura estendendola con funzionalità necessarie alla gestione di documenti XML
    - sono tipicamente Object-Relational (DB2, Oracle8i,…)
    - utili per Data Centric e parzialmente per Document Centric

9

XML e DBMS

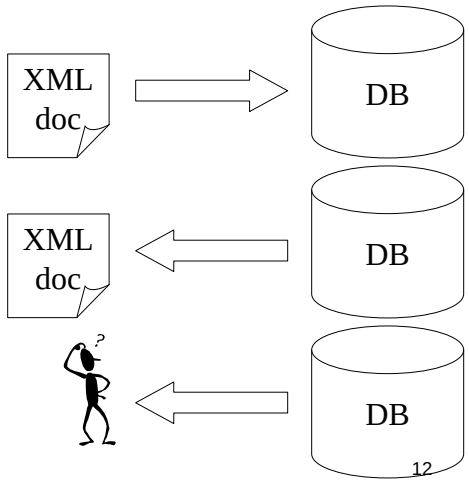
- Nel seguito.
  - Problematiche relative alla gestione di documenti Data Centric e Document Centric in XML-Enabled DBMS

10

XML-Enabled DBMS e documenti Data Centric

Problematiche per Data Centric

- Tre problematiche di base:
  - come rappresentare i dati contenuti nei documenti XML nel DBMS
  - come generare documenti XML partendo dai dati contenuti nel DBMS
  - come interrogare i dati estratti da documenti XML



Rappresentazione dati

- È necessario definire un mapping tra la struttura dei documenti XML e lo schema del DB
  - ➨ Per memorizzare i dati contenuti in un documenti XML in un DB, deve esistere una o più tabelle con lo schema richiesto dal mapping
- rappresentazione strutturata
- Vantaggi:
  - approccio piuttosto semplice
  - i dati sono facilmente interrogabili
- Svantaggi:
  - Scarsa flessibilità: la tabella deve essere conforme al documento
  - il documento di partenza non è più recuperabile

13

1. DBMS relazionale

- Un documento XMl viene rappresentato come una singola tabella o un insieme di tabelle
- la struttura del documento XML è simile alla seguente:

```
<database>
  <table>
    <row>
      <column1>...</column1>
      <column1>...</column1>
            ...
    </row>
    ...
  </table>
  ...
</database>
```

- approccio tipico per DBMS relazionali, object-relational

14

Esempio

Documento XML

```
<clienti>
  <row>
    <numero> 7369 </numero>
    <nome> PAUL </nome>
    <cognome> SMITH </cognome>
  </row>
  <row>
    <numero> 7000 </numero>
    <nome> STEVE </nome>
    <cognome> ADAM </cognome>
  </row>
</clienti>
```

Tabella Clienti

Numero	Nome	Cognome
2000	MIKE	SCOTT
7369	PAUL	SMITH
7000	STEVE	ADAM

15

Esempio

Documento XML

```
<clienti>
  <row>
    <numero> 7369 </numero>
    <lista_clienti>
      <cliente>
        <nome> PAUL </nome>
        <cognome> SMITH </cognome>
      </cliente>
      <cliente>
        <nome> STEVE </nome>
        <cognome> ADAM </cognome>
      </cliente>
    </lista_clienti>
  </row>
</clienti>
```

Tabella Lista_Clienti

Numero
2000
7369

Tabella Clienti

Numero	Num_cliente	Nome	Cognome
2000	1	MIKE	SCOTT
7369	2	PAUL	SMITH
7369	3	STEVE	ADAM

16

2. DBMS object relational

- Il documento può sempre essere mappato in una singola tabella, utilizzando campi strutturati

17

Esempio

Documento XML

```
<clienti>
  <row>
    <numero> 7369 </numero>
      <cliente>
        <nome> PAUL </nome>
        <cognome> SMITH </cognome>
      </cliente>
  </row>
  <row>
    <numero> 7000 </numero>
      <cliente>
        <nome> STEVE </nome>
        <cognome> ADAM </cognome>
      </cliente>
  </row>
</clienti>
```

Tabella Clienti

Numero	Cliente
2000	nome: MIKE; cognome: SCOTT
7369	nome: PAUL; cognome: SMITH
7000	nome: STEVE; cognome: ADAM

18

Esempio

Documento XML

```
<clienti>
  <row>
    <numero> 7369 </numero>
     <lista_clienti>
       <cliente>
         <nome> PAUL </nome>
         <cognome> SMITH </cognome>
       </cliente>
       <cliente>
         <nome> STEVE </nome>
         <cognome> ADAM </cognome>
       </cliente>
     </lista_clienti>
   </row>
</clienti>
```

Tabella Clienti

Numero	Cliente
2000	nome: MIKE, cognome: SCOTT
7369	CLIENTE: nome: PAUL, cognome: SMITH CLIENTE: nome: STEVE, cognome: ADAM

19

Interrogazione dati

- Poiché i dati vengono rappresentati secondo il modello supportato dal DBMS (es. relazionale), è possibile utilizzare i linguaggi supportati dal DBMS per l'interrogazione dei dati memorizzati
- approccio template-based:
  - la query viene rappresentata nel documento XML
  - necessità di middleware

20

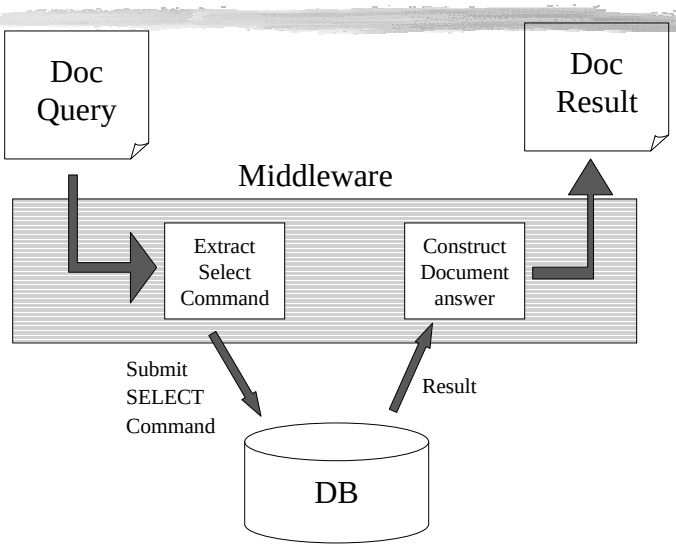
Flight Information

```
<?xml version="1.0">
<FlightInfo>
   <Intro>The following flights have available seats:</Intro>
   <SelectStmt>
     SELECT Airline, FltNumber, Depart, Arrive FROM Flights
   </SelectStmt>
   <Conclude>We hope one of these meets your needs</Conclude>
</FlightInfo>
```

```
<?xml version="1.0">
<FlightInfo>
   <Intro>The following flights have available seats:</Intro>
   <Flight>
     <Row>
        <Airline>ACME</Airline><FltNumber>123</FltNumber>
        <Depart>Dec 12, 1998 13:43</Depart><Arrive>...<Arrive>
     <Row>
   </Flight>
   <Conclude>We hope one of these meets your needs</Conclude>
</FlightInfo>
```

21

Interrogazione dati



22

Generazione documenti XML

- Problema: fornire una rappresentazione XML ai dati recuperati tramite query dal DBMS
- si utilizza il mapping inverso rispetto a quello utilizzato per la memorizzazione
- operazione importante per attribuire un formato standard ai dati ritrovati, prima di inviarli sulla rete

23

Esempio

```
SELECT nome, cognome
FROM Clienti
WHERE Numero = "7369
```

Tabella Clienti

Numero	Nome	Cognome
2000	MIKE	SCOTT
7369	PAUL	SMITH
7000	STEVE	ADAM

Documento XML

```
<clienti>
  <row>
    <nome> PAUL </nome>
    <cognome> SMITH </cognome>
  </row>
</clienti>
```

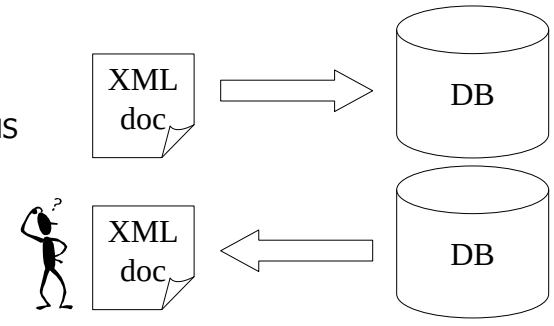
24

XML-Enabled DBMS e documenti Document Centric

25

Problematiche per Document Centric

- Due problematiche di base:
  - come rappresentare i documenti XML nel DBMS
  - come interrogare i documenti XML



26

Rappresentazione

- Permette di mantenere integro il documento XML
- Due approcci:
  - rappresentazione non strutturata
    - documento come unico oggetto
  - rappresentazione ibrida
    - documento parzialmente rappresentato secondo la rappresentazione strutturata e parzialmente secondo la rappresentazione non strutturata

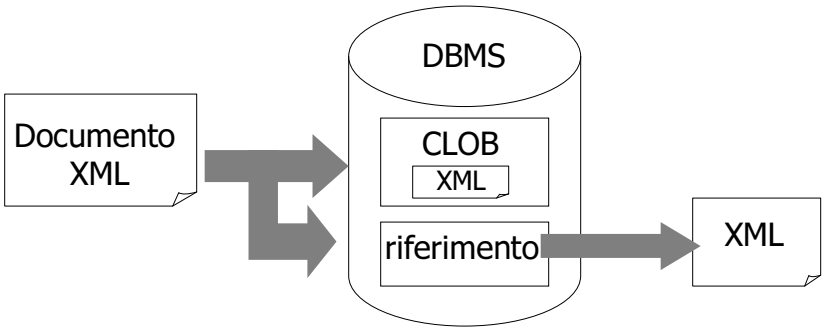
27

Rappresentazione non strutturata

- Il documento viene tipicamente mappato in un singolo campo di una tabella di tipo:
  - CLOB (Character Large Object): il documento è fisicamente contenuto nel campo della tabella
    - alcuni DBMS (IBM DB2) supportato tipi ad hoc: XMLVARCHAR
  - riferimento: il campo contiene il riferimento al documento, memorizzato altrove, sul file system
- Vantaggi:
  - flessibile
- Svantaggi:
  - i dati sono non strutturati
  - interrogazione più complessa
  - la tabella può contenere documenti eterogenei (diversi DTD)

28

Rappresentazione non strutturata



29

Esempio

Documento XML

```
<clienti>
   <row>
     <numero> 7369 </numero>
     <nome> PAUL </nome>
     <cognome> SMITH </cognome>
   </row>
   <row>
     <numero> 7000 </numero>
     <nome> STEVE </nome>
     <cognome> ADAM </cognome>
   </row>
</clienti>
```

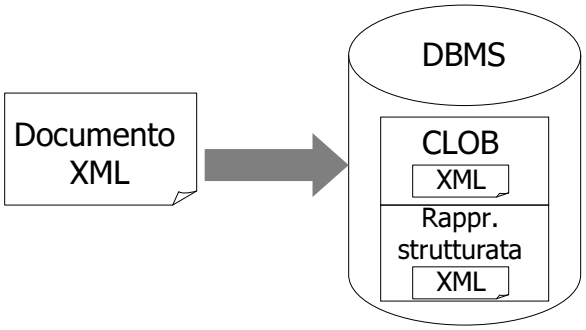
Tabella Clienti

Id	Documento_XML
10	<clienti> <row> <numero> 7369 </numero> <nome> PAUL </nome> <cognome> SMITH </cognome> </row> <row> <numero> 7000 </numero> <nome> STEVE </nome> <cognome> ADAM </cognome> </row> </clienti>

30

Rappresentazione ibrida

- Rappresentazione che combina rappresentazione strutturata e non strutturata



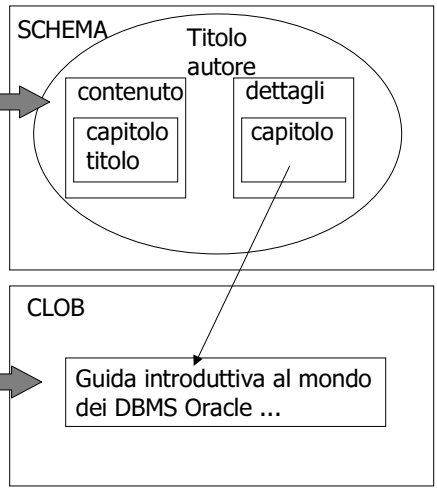
31

Esempio

Documento XML

```
<libro>
   <titolo> Oracle Guide </titolo>
   <autore> M. Abbey  </autore>
  <contenuto>
     <capitolo n='1'>
        <titolo> Introduzione </titolo>
        …
     </capitolo>
     …
  </contenuto>
  <dettagli>
     <capitolo n='1'>
       <sezione n='1'>
         Guida introduttiva al mondo
               dei DBMS Oracle …
       </sezione>
     </capitolo>
  </dettagli>
 …
</libro>
```

Tabella LIBRO



32

Interrogazione documenti

- Dal punto di vista del DBMS, un documento memorizzato in modo non strutturato non è che un documento di testo
- in genere i DBMS supportano strumenti per ritrovare i documenti in base al contenuto
- nel caso di documenti XML, mettono a disposizione operatori avanzati da utilizzare in statement SQL per recuperare documenti XML in base al contenuto

33

Interrogazione in Oracle

- È possibile utilizzare un particolare motore di ricerca per testi
  - Intermedia Text (ne parleremo nel contesto Multimedia)
- utilizzando questo strumento è possibile abilitare ricerche sui vari elementi ed attributi di un documento XML, tramite un meccanismo di indicizzazione
- SQL viene esteso in modo da supportare predicati ad hoc per la ricerca in documenti XML

34

Interrogazione in Oracle

- Nuova funzione:
  - `CONTAINS(XML_COLUMN, QUERY_TAG)`
    - `XML_COLUMN`: colonna (attributo) in cui sono contenuti i documenti XML
    - `QUERY_TAG`: predicato che permette di specificare condizioni sui documenti XML
  - `QUERY_TAG::= <tag_value> WITHIN <tag_name> | <attribute_value> WITHIN <tag_name@attribute_name>| …`
  - `CONTAINS` restituisce un valore maggiore di 0 se la condizione è verificata

35

Esempio

- LISTA_CLIENTI(NUMERO, DOCUMENTO_XML)
- SELECT * FROM LISTA_CLIENTI
  WHERE COND >0;
- COND = CONTAINS (DOCUMENT_XML, 'PAUL WITHIN NOME')
  - determina tutti i documenti contenuti nel campo DOCUMENT_XML che contengono un tag NOME con valore PAUL
- COND = CONTAINS (DOCUMENT_XML, '1 WITHIN NUM@Cliente')
  - determina tutti i documenti contenuti nel campo DOCUMENT_XML che contengono un elemento Cliente con un attributo NUM di valore 1

36

XML e Oracle 8i

- XML-enabled
- supporta rappresentazione strutturata, non strutturata in campi CLOB e BFILE, e ibrida
- interrogazione rappresentazione non strutturata tramite Intermedia Context
- generazione documenti XML a partire dal contenuto DB

37

XML e IBM DB2

- XML enabled
- supporta rappresentazione strutturata, non strutturata in campi ad hoc, e ibrida
- Nuovi tipi di dato:
  - XMLVARCHAR: documenti XML memorizzati come VARCHAR
  - XMLCLOB: documenti XML memorizzati come CLOB
  - XMLFILE: riferimento ad un documento XML, memorizzato su file system
- interrogazione rappresentazione non strutturata tramite:
  - operatori specifici, che permettono di navigare la struttura del documento
  - text extender, che supporta funzionalità aggiuntive di analisi del contenuto (ne parleremo nel contesto Multimedia)
- generazione documenti XML a partire dal contenuto DB

38