ALMA MATER STUDIORUM - UNIVERSITÀ DI BOLOGNA

Modeling and Processing of Multimedia Data

International Second cycle degree programme (LM) in
Digital Humanities and Digital Knowledge (DHDK)
University of Bologna

Data Types Classification

Home page: http://www-db.disi.unibo.it/courses/DMMMDB/
Electronic version: 0.02.DataTypes.pdf
Electronic version: 0.02.DataTypes-2p.pdf

1

Outline

- Basics on structured data
  - Relational databases
  - Practical exercises on SQL queries
- Basics on semi-structured data
  - Practical examples
- Unstructured data
- Comparative analysis

2

Let's keep in mind our goal!

- *Facilitate* and *improve* the "access" to documentary data repositories for general users, conjunctively exploiting:
  - dedicated users manually provided *metadata* → **structured data**
  - *low level features* (e.g., document keywords) → **unstructured data**
  - semi-automatically provided annotations → **semi-structured data**

Archivio Storico Fiat

- **Trimotore Fiat G212**
- Data: **1947**
- Collezione: **Tema di cultura industriale**
- Tipologia: **Immagine**
- **Aereo**, **Motore**, **Ali**

Cineteca

- **Das Cabinet des Dr. Caligari**
- Data: **1920**
- Nazione: **Germania**
- Regista: **Robert Wiene**
- Genere: **Horror**
- **Espressionismo**, **Ipnosi**, **Sonnambulismo**

Archivio Artistico

- **La Gioconda**
- Sito: **Museo Louvre, Parigi**
- Secolo: **XVI**
- Autore: **Leonardo da Vinci**
- Periodo: **Rinascimento**
- Data: **1503**
- **Dipinto**, **Ritratto**, **Sorriso**

Recall on structured data

- **Structured data** base on a predefined **schema** able to describe the content of the document collection (…what is known as "**traditional metadata**" in the cultural heritage domain)
- A database (DB) can be seen as a *collection of objects representing some information of interest in a structured way* (i.e., through a schema)
- A *relational database management systems* (*RDBMS* or just *DBMS)* is a *software system able to "manage" collections of objects* which can be very *large* (Giga-Tera byte and more) and *shared* by different applications in a *persistent* way (even in presence of faults)
  - "*manage*" = obtain, elaborate, maintain, produce, distribute
- Examples of DBMSs: Oracle, *IBM (DB2 UDB), Microsoft (SQL Server), Sybase, mySQL, PostgreSQL, InterBase*

Relations as tables

- DBMSs use the *relational model* (Codd, 1970) to *describe the data, that is the information* is organized in tables ("*relations*")
- The rows of table corresponds to *records*, while the columns correspond to *attributes (schema)*
- The language to *store/retrieve* information from such tables is the *Structured Query Language* (SQL)
- Example: if we want to create a *table with employees records, so that we can store their employee number, name, age and salary*, we can use the following SQL statement:

```
create table EMPLOYEE (
  empN integer PRIMARY KEY;
  name char(50);
  age  integer;
  salary float );
```

An alternative, "compact way", to represent the schema of table **EMPLOYEE** is:

EMPLOYEE(empN,name,age,salary)

empN	name	age	salary

Populating and querying tables

- Tables can be populated with the SQL `insert` command, e.g.,:

```
insert into EMPLOYEE values (
  123, 'Smith, John', 30, 38000.00);
insert into EMPLOYEE values (
  456, 'Johnson, Tom', 25, 55000.00);
```

empN	name	age	salary
123	Smith, John	30	38000.00
456	Johnson, Tom	25	55000.00

- We can retrieve information using the `select` command. E.g., *if we want to find all the employees with salary less that 50000*, we use the query:

```
Select *
From   EMPLOYEE
Where  salary <= 50000.00
```

Query execution

- In absence of *access methods* (e.g., an *index*), the DBMS will perform a sequential scanning, *checking the salary of each and every employee record against the desired threshold of 50000!!!*
- To accelerate queries execution, we can create an index (usually a *B-tree* index, as we will see in few minutes) with the command `create index`
- E.g., to build an index on the employee's salary, we would issue the SQL statement:

```
create index salIdx on EMPLOYEE(salary)
```

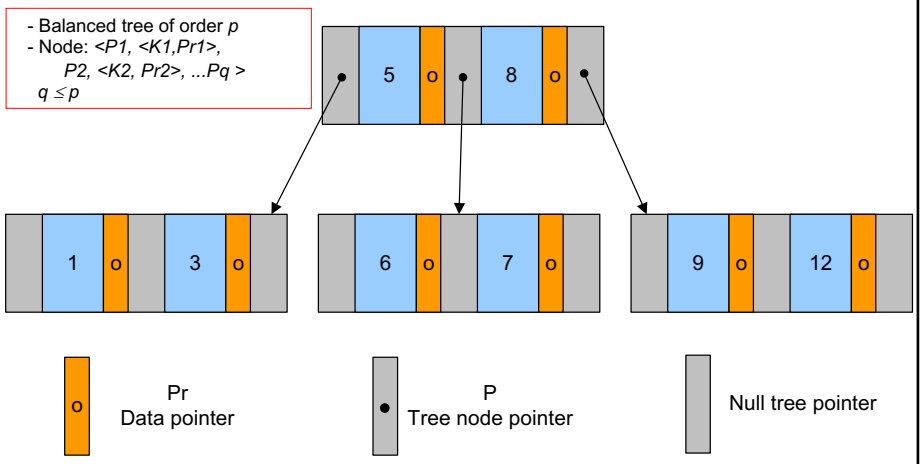
- In general the DBMS relies on an "optimizer" component to decide which is the more efficient way to execute a given query
  - sequential vs. index-based evaluation
  - which index is the most appropriate
  - …

Storage hierarchies

- First level is typically *main memory or core or RAM*
  - Fast (access time of *micro-seconds* or faster), small, expensive
- Second level (*secondary store*) is typically magnetic disk
  - *Much slower* (*5-10 msec.* access time)*, but much larger and cheaper*
- Database researchers has focused on "large databases" that *do not fit in main memory and thus have to be stored in secondary memory*
- *Secondary store is organized* into block (= pages)
  - The reason is that, accessing data from the disk involves the mechanical move of the read/write head of the disk above the appropriate track on the disk
- Because these moves ('*seeks*') are slow and expensive, every time we do a disk read we bring into main memory a whole disk block (of the order of *1KB - 8 KB*)
- *So, it makes a huge difference of performance if we mange to group similar data in the same disk blocks*!!

B-tree

- Access methods, like *B-tree*, try exactly to achieve good clustering of data in order to minimize the number of disk-reads

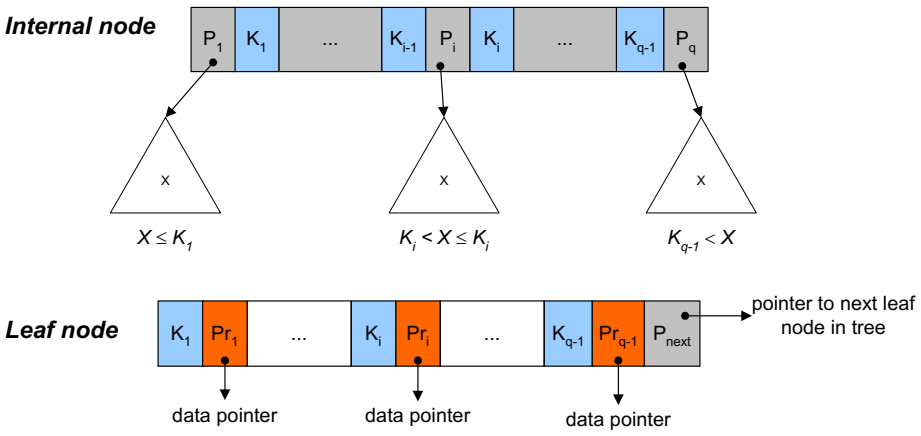


9

B+-tree

- B-tree variant
  - more commonly used than B-tree

- Data pointers only at the leaf nodes
- All leaf nodes linked together
  - ° allows ordered access! ☺



10

What else…

- The relational model and SQL provide a large number of additional features, such as:
  - the ability to retrieve information from several tables ('joins'); the matching is based on values!
  - the ability to perform aggregate operations (e.g., sums, averages, etc.)
- For time reasons, we restrict our discussion to the few features which are the essential ones for our purposes

- Now we focus on a set of practical exercises based on real relational schemas, at increasing complexity
  - In doing this, we follow the "learning by doing" methodology
  - This is possible also thanks to our LIM that represents the easiest way to draw both teachers and students closer to a new teaching and learning method, transforming the traditional frontal teaching method into a cooperative learning system

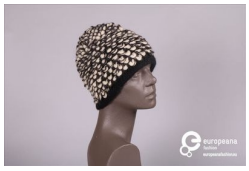
Database FASHION

- Here we provide an example of a "toy" database schema that simulates the real life of a little fashion company selling clothing…

CUSTOMER(CODE_CUST,NAME,ADDRESS,E_MAIL,AGE)

CLOTHING(CODE_CLOTHES,TYPE,BRAND,PRICE,DESCRIPTION)

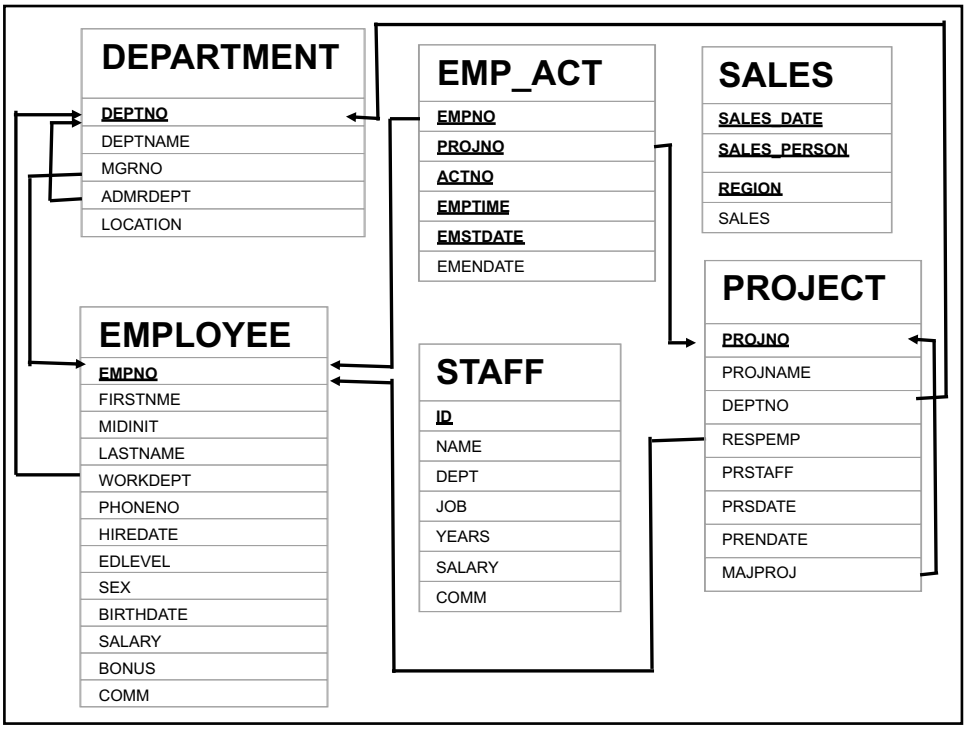
ORDER(CODE_CUST,CODE_CLOTHES,NUMBER_OF_ITEMS)
Foreign Key: CODE_CUST REFERENCES CUSTOMER
Foreign Key: CODE_CLOTHES REFERENCES CLOTHING



Database COMPANY

- Here we provide an example of a real database schema that shapes the real life of a small/medium-size company to show how things can become complex…

13



14

Queries database FASHION

1. “Distinct descriptions of clothing of type "hat" and brand “Armani” costing less than € 100”

2. “Clothing codes of orders related to customers whose code belongs to the set (‘X’, ‘Y’, ‘Z’, ‘T’)”

3. “E-mails of customers whose name starts with ‘BA’”

15

Queries database FASHION

4. “Clothes codes, and relative number of orders, for clothing that have been ordered at least 3 times”

5. “Clothes codes, and relative total number of ordered items, by ordering the result (in descendent way) by total number of items and then by clothes code”

6. “Customer codes and total number of ordered clothing for those customers that have ordered at least 500 clothes”

7. “Clothing codes, types, brands, and prices of orders related to customers whose code belongs to the set (‘X’, ‘Y’, ‘Z’, ‘T’)”

16

Queries database COMPANY

1. “Woman employees of the 'D11' department with an annual salary exceeding $ 22,000”
2. “Ordered list of the names of the departments ending in 'S' and that refer to the master department 'D01'”
3. “Names of staff people who earn over $ 20,000 but who are not managers”
4. “Employees with 16 years of school education in descending order of department code”
5. “Names of departments that do not have a reference manager”

Queries database COMPANY

6. “Identification and name of persons belonging to the staff who have not null commission and who have a seniority of service between 3 and 5 years”
7. “Surname and name (alphabetic order 'A-Z' on both) of employees who made sales in 'Quebec' on 1996-04-01”
8. “Data relating to projects whose managers are male, have a school education greater than or equal to 18 years and whose department of affiliation is that of 'PLANNING'”

Queries database COMPANY

9. "Region name and total sales for each region where the total sales is greater than 30, sorted by total sales in descending order"

10. "Name of projects and number of distinct employees working on each project in descending order of number of employees (select only the first 5 records)"

Recall on semi-structured data

- **Semi-structured data** are partially described by means of hierarchical or graph-based models
- Among relevant models:
  - XML,
  - RDF,
  - OWL,
  - …
- For a complete treatment on the subject, please refer to the courses:
  - "*Knowledge Organization and Digital Methods in Cultural Heritage Domain*"
  - "*Information Modelling and Web Technologies*", and
  - "*Knowledge Representation and Extraction*"
    - for Bologna's students
- and to the course:
  - "*Humanities Computing*"
    - for Ravenna's students

XML by example

- XML document example:

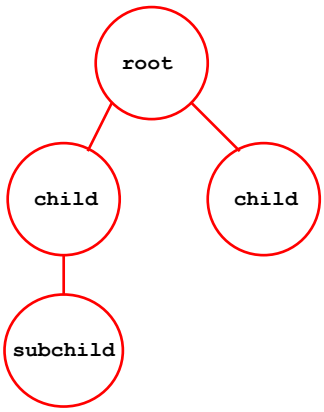
```
<Article>
        <Author>
                <FirstName>Bob</FirstName>
                <Surname>Smith</Surname>
        </Author>
        <Abstract>This paper concerns.... </Abstract>
        <Section n="1">
                <Title>Introduction</Title>
                <Para>...
        </Section>
</Article>
```

- Specific languages (e.g., *XQuery*, *XPath*) are used for querying

XML: from physical too logical representation

- There is a direct correspondence between the physical representation of an XML document a its logical representation (or *document tree*)

```
<root>
   <child>
       <subchild>
            ...
       </subchild>
   </child>
   <child>
   ...
   </child>
</root>
```



Unstructured data

- **Unstructured data** are data **without a model/schema able to describe them or to assign a specific semantic**
- E.g., color distribution of an image, shape of an object, etc.

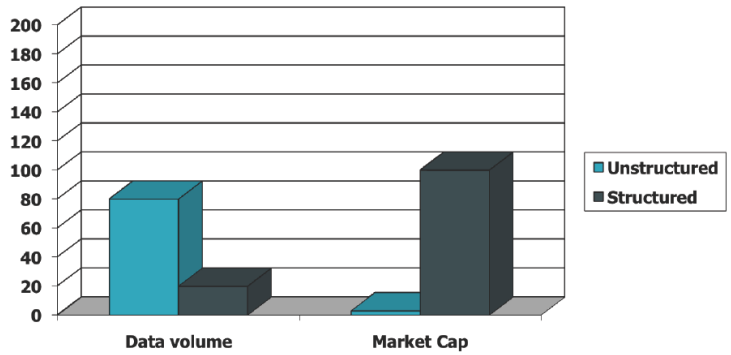
- Relevant "rich" examples of unstructured data can be found in data repositories such as:
  - World Wide Web (WWW);
  - Multimedia digital libraries;
  - Digital museum;
  - Multimedia digital archives;
  - …
  - any other suggestion? ☺

Why unstructured data are that important?

- On the basis of studies conducted in nineties, users preferred to receive information by other people rather than using an information retrieval system
  - E.g., travel booking
- The trend has been reversed in the last 10 years thanks to the success of Web technologies and Web search engines
  - E.g., already in 2004 the 92% of the population considered the Web a suitable source for the daily retrieve of useful information

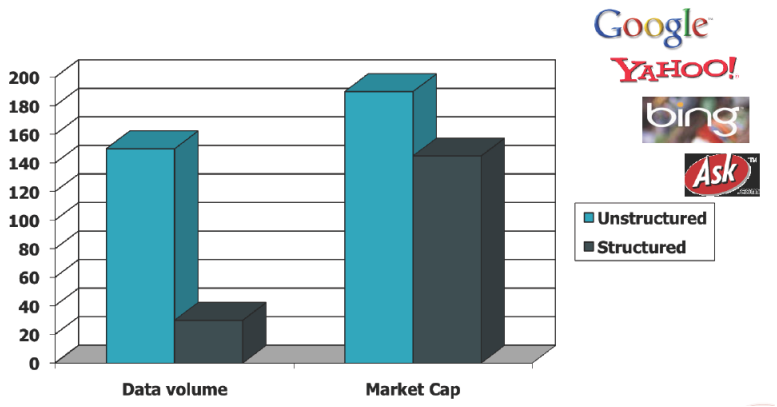
- Let's keep in mind that:
  - "85% of all stored data is held in unstructured formats"
  - "80% of business is conducted on unstructured data"
  - Unstructured data double every 3 months

Structured vs. unstructured data: in 1996



32

Structured vs. unstructured data: in 2009



33

Free exercise 1.A

- Make a note of all the different media and combinations of media you are exposed to in the course of a single day
- Figurate some concrete examples of MM applications (like the ones just illustrated) by separately describing, in **natural language**, relevant **structured**, **semi-structured**, and **unstructured data**

Archivio Storico Fiat

- **Trimotore Fiat G212**
- Data: **1947**
- Collezione: **Tema di cultura industriale**
- Tipologia: **Immagine**
- **Aereo**, **Motore**, **Ali**

what else? …

Free exercise 1.B

- Starting from **descriptions in natural language** of relevant **structured**, **semi-structured**, and **unstructured data** selected for your examples of Exercise 1.A, **model the data** according to:
  - **relational model** (for **structured data**), and
  - **XML model** (for **semi-structured data**)
- Provide a **definition accurate as much as possible of the low-level features** you chose for describing the "content" of involved MM data (**unstructured data**)

Free exercise 1.B: students to do

- Prepare an electronic version of your proposals
  - .ppt file

  similarly, to the examples proposed during lectures

Low-level features description per unstructured data
- Distribuzione colore dell'immagine
- Forma degli oggetti nell'immagine
- …