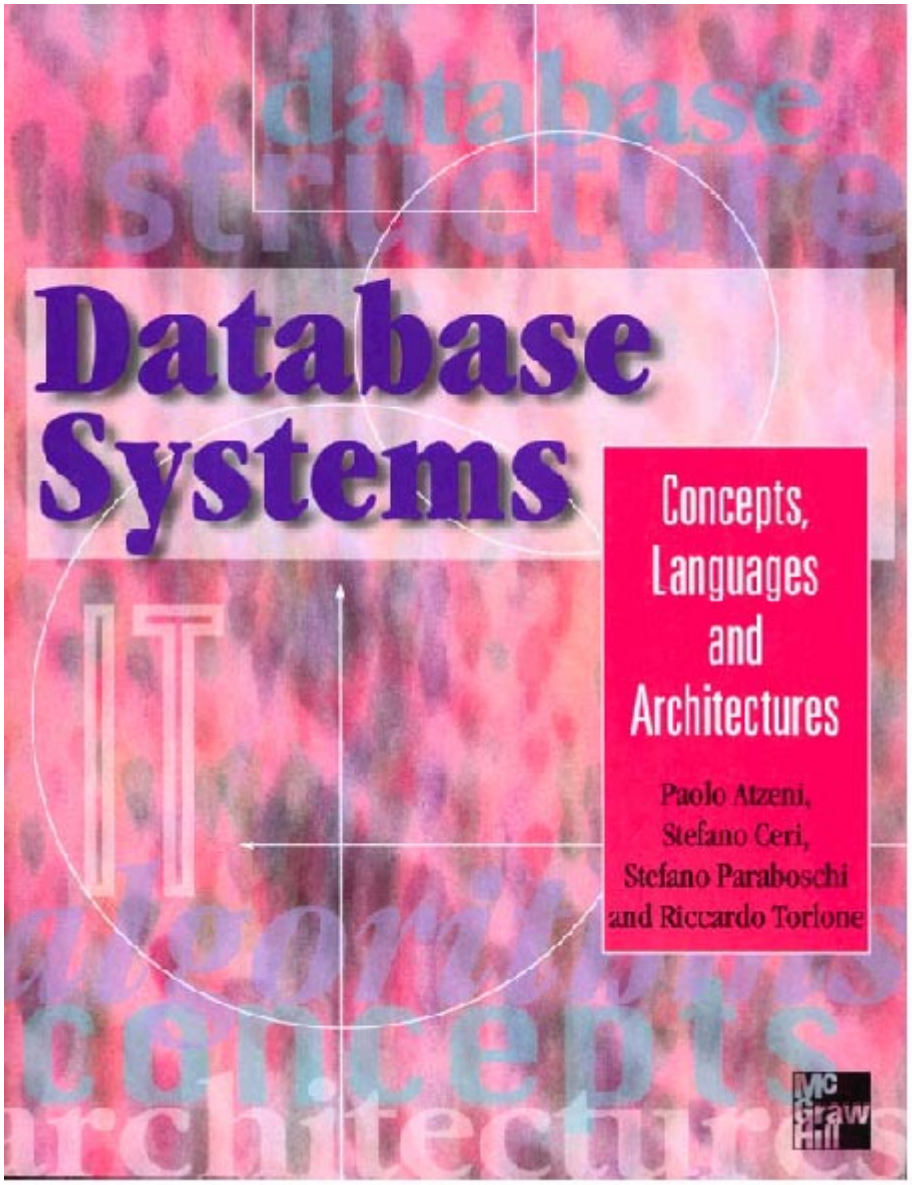


Chapter 1

Introduction

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

Information system

- Component of an organization that manages (gets, processes, stores, communicates) the information of interest
  - each organization has an information system, possibly not made explicit in its structure
  - usually, the information system operates in support to other components of teh organization

- The very notion of information system is partly independent of its computerization; however, we are mainly interested in information systems that are, to a large extent, computerized

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

Management of information

- Information is handled and recorded according to various techniques:
  - informal ideas
  - natural language (written or spoken)
  - drawings, diagrams,
  - numbers
  - codes

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

Structured information

- As activities become systematized, appropriate forms of organization and codification for information have been devised
- Look at information about people
  - in most countries a structure for the name has been introduced in the last few centuries
  - later, it was realized that it could be useful to keep track of birthdate and birthplace (and use them in order to identify people, together with the name)
  - more recently, social security numbers (or tax codes) have been introduced in order to obtain unique identification

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

Information and data

- In most computer-based systems (as well as in many other places) information is represented by means of data
  - **data** raw facts, to be interpreted and correlated in order to provide **information**
- An example:
  - "John Smith" and 25755 are a name (or, better, a string) and a number: two pieces of data
  - if they are provided as a reply to a request: "Who is the dept head, and which is his/her extension," then we get information out of them

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

Why data?

- This is the best that can be done to a large extent
- In most cases data are a valuable resource, with a very long life-cycle: banking applications have had data with the same structure for centuries, well before computers were invented!

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

Database

(generic definition)

- **a collection of data, used to represent information of interest to an information system**

(more technical definition)

- **a collection of data, managed by a DBMS**

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

DataBase Management System — DBMS

- software system able to manage **collections of data** that are
  - **large** (bigger, often much bigger, than the main memory available)
  - **shared** (used by various applications and users)
  - **persistent** (with a lifespan that is not limited to single executions of the programs that use them)

  and to ensure their reliability (so preserving the database in case of hardware or software failure) and privacy (controlling accesses and authorizations). Like any software product, a dbms must be **efficient** (using the appropriate amount of resources, such as time and space) and **effective** (supporting the productivity of its users).

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

Sharing

- Most organizations have a structure (departments, divisions, …) and each component is interested in a portion of the information system
- The data of interest of the various components often overlap
- A database is an **integrated resource**, shared by various components
- Integration and sharing allow a reduction of **redundancy** and the consequent possibility of **inconsistency**
- Since sharing is never complete, DBMS provide support for privacy of data and access authorizations
- Sharing also requires that multiple accesses to data are suitably organized: **concurrency control** techniques are used

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

DBMS vs file systems

- The management of large and persistent sets of data can be done by means of simpler tools: file systems
- File systems provide also rough support for sharing
- There is no sharp line between DBMSs and non-DBMSs: DBMSs provide many features, that extend those of file systems
- The crucial issue is **effectiveness**, take advantage of these features

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

DBMS vs file systems (2)

- In traditional programs that make use of files, each program includes a description of the organization of the file, which is often just a stream of bytes; there are chances of incoherence between the file and its description (or descriptions, if teh file is shared)
- In DBMSs, there is a portion of the database (called the **dictionary** or **catalogue**) that describes teh database itself, which is shared

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

Data model

- set of constructs used to organize data
- basic feature: **structuring mechanism** (or **type constructor**), as in programming languages; in Pascal we have array, record, set, file constructors
- in the **relational database model** we have the **relation** constructor, to organize data as sets of homogeneous records

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

Tables: representation of relations

COURSES

Course	Tutor	Room
Databases	Smith	DS3
Systems	Black	N3
Networks	Brown	N3
Theory	Brown	G

ROOMS

Code	Building	Floor
DS1	Ex-OMI	Ground
N3	Ex-OMI	Ground
G	Science	Third

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

Schemas and instances

In a database we have:

- the **schema**, rather stable over time, that describes the structure (**intensional** component);

  in the example, the headings of the tables
- the **instance**, the actual values, which vary, even very rapidly (**extensional** component);

  in the example, the bodies of of the tables

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

Two main types of models

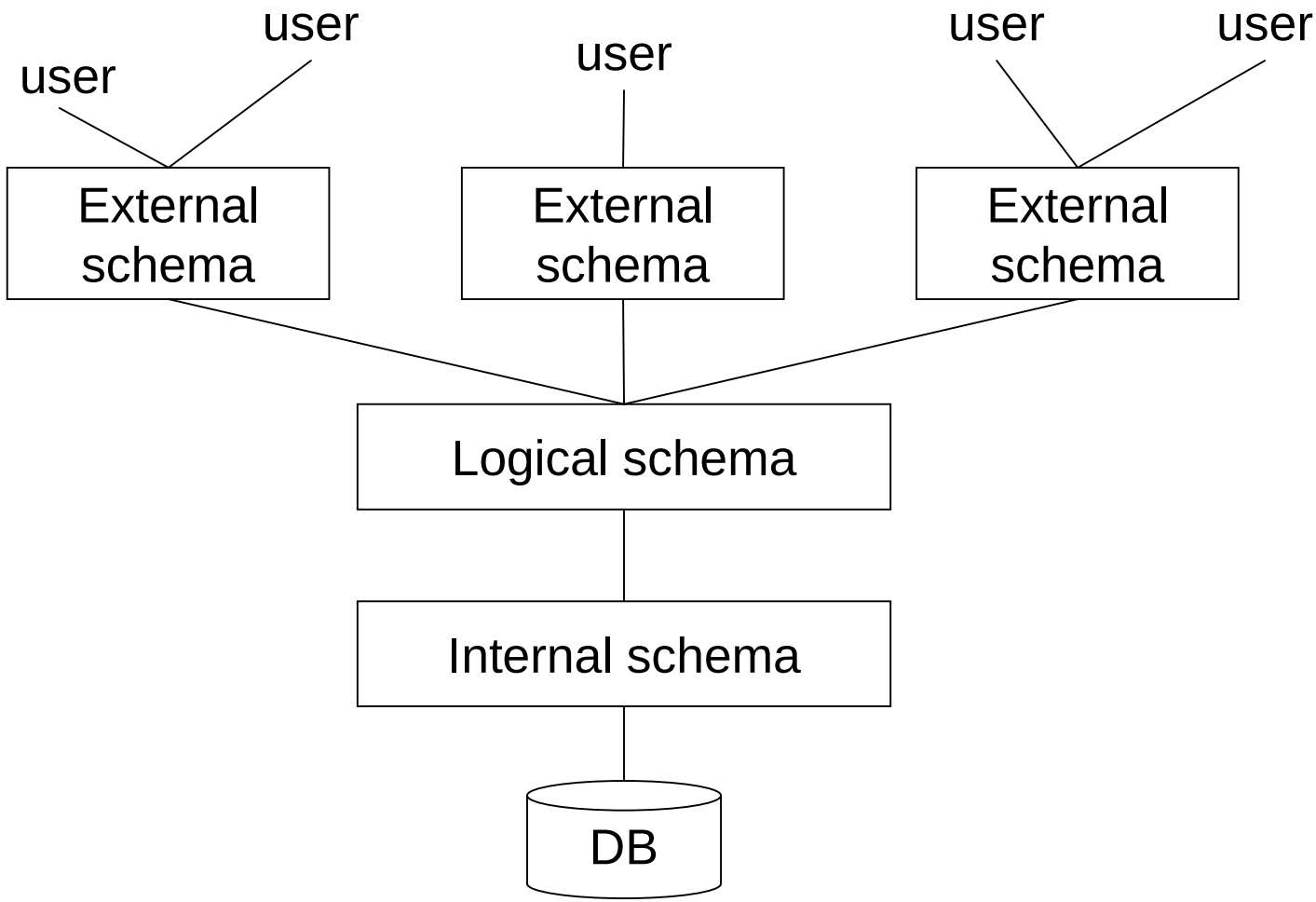
- **Logical models:** used in DBMSs for the organization of data at a alevel that abstracts from physical structures

  examples: relational, network, hierarchical, object
- **Conceptual models**: used to describe data in a way that is completely independent of any system, with the goal of representing the concepts of the real world; they are used in the early stages of database design

  the most popular is the **Entity-Relationship** model

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

Standard (ANSI/SPARC) three-level architecture for a DBMS



© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

ANSI/SPARC architecture: schemas

Logical schema: description of the whole database by means of the logical model adopted by the dbms

External schema: description of a portion of the database in a logical model ("views," possibly in differnt models)

Physical schema: description of the implementation of the logical schema by means of physical storage structures

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

Data independence

Guaranteed by the multilevel architecture (which allows access only via the external level; could coincide with the logical one)

Two forms of independence

Physical: the logical and external level are independent of the the physical one; a relation is referred to always in the same way, regardless of its physical implementation (which could even varyu over time)

Logical: the external level is independent of the logical one

- addition of (or changes to) views do not require changes to the logical schema
- changes to the logical schema need not affect the external schemas (provided that the definition of mappings are adjusted)

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

Database languages

- Various forms (a contribution to effectiveness)
  1. Interactive textual languages, such as SQL
  2. Interactive commands **embedded** in a **host** language (Pascal, C, Cobol, Java, etc.)
  3. Interactive commands **embedded** in a **ad-hoc** development language, usually with additional features (for the production of forms, menus, reports, ...)
  4. By means of non-textual **user-friendly** interfaces

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

SQL, an interactive language

```
SELECT Course, Room, Floor
FROM Rooms, Courses
WHERE Code = Room
AND Floor=”Ground"
```

Course	Room	Floor
Networks	N3	Ground
Systems	N3	Ground

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

SQL embedded in Pascal

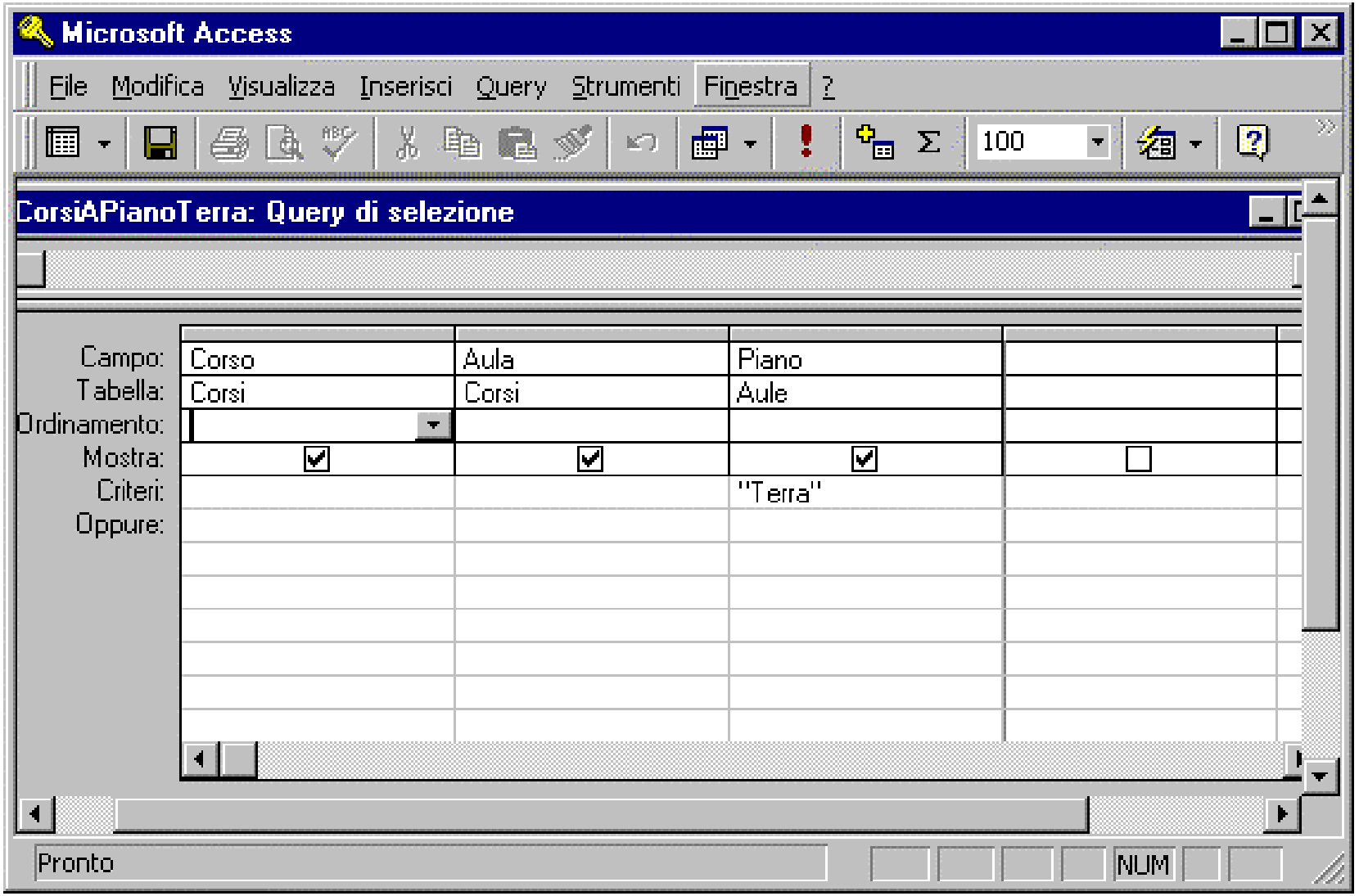
```
write('city name"?'); readln(city);
EXEC SQL DECLARE E CURSOR FOR
      SELECT NAME, SALARY
      FROM EMPLOYEES
      WHERE CITY = :city ;
EXEC SQL OPEN E ;
EXEC SQL FETCH E INTO :name, :salary ;
while SQLCODE = 0 do begin
    write(employee:', name, 'raise?');
    readln(raise);
    EXEC SQL UPDATE PERSONE SET SALARY = SALARY + :raise
                WHERE CURRENT OF E
    EXEC SQL FETCH E INTO :name, :salary
  end;
EXEC SQL CLOSE CURSOR E
```

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

SQL embedded in a ah-hoc language (Oracle PL/SQL)

```
declare  Sal number;
begin
     select Salary into Sal
     from Employee
     where Code = ’575488’
     for update of Salary;
     if Sal > 30 then
        update Employee set Salary = Salary * 1.1 where Code = ’575488’;
     else
        update Employee set Salary = Salary * * 1.15 where Code = ’575488’;
     end if;
     commit;
  exception
     when no_data_found then
       insert into Errors
        values(‘No employee has given code',sysdate);
  end;
```

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999



© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

A useful distinction (cfr. the separation data vs programs)

data definition language (DDL) :
used to define the logical, external and physical schemas and access authorizations

data manipulation language (DML) :
used for querying and updating database instances

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

People

- DBMS designers and implementors
- database designers and database administrators (DBA)
- application designers and developers
- users:
  - end users, who use predefined **transactions** (such as a flight reservation or a bank operation)
  - **casual users**, who issue queries by means of interactive languages or interfaces

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999

DBMSs: advantages and disadvantages

Pros

- data can be handled as a common resource, the database is a model of the real world
- centralized management and economy of scale
- availability of integrated services
- reduction of redundancies and inconsistencies
- data independence (an atout in the development and maintenance of applications)

Cons

- cost of the product (and associated tools) and of the migration
- difficulty in separating features and services (with possible lack of efficiency)

© McGraw-Hill and Atzeni, Ceri, Paraboschi, Torlone 1999