

Database Systems

Introduction

H. Turgut Uyar Şule Öğüdücü

2002-2010

1 / 31

License



©2002-2010 T. Uyar, Ş. Öğüdücü

You are free:

- ▶ to Share — to copy, distribute and transmit the work
- ▶ to Remix — to adapt the work

Under the following conditions:

- ▶ Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).
- ▶ Noncommercial — You may not use this work for commercial purposes.
- ▶ Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same or similar license to this one.

Legal code (the full license):

<http://creativecommons.org/licenses/by-nc-sa/3.0/>

2 / 31

Topics

Data Processing

Introduction
Record Files

Database Management Systems

Introduction
Client/Server
SQL

3 / 31

Processing Data

- ▶ *problem:*
effectively storing and processing large amounts of data
- ▶ processing data
 - ▶ adding new data
 - ▶ changing existing data
 - ▶ deleting data
 - ▶ querying: planned - ad hoc

4 / 31

Data Types

- ▶ *persistent data:*
data that must be stored due to the nature of the information
- ▶ *temporary data*
 - ▶ *output data:* data that can be derived from persistent data (query results, reports, etc.)
 - ▶ *input data:* data which just entered the system
 - ▶ can be added to persistent data
 - ▶ can cause changes in the persistent data
 - ▶ can be deleted without processing

5 / 31

Roles

- ▶ *end users:*
people who work on the data
 - ▶ assumed not to have any technical knowledge
- ▶ *application programmers:*
people who develop the programs that the end users use

6 / 31

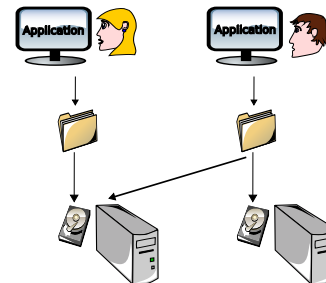
Application Example

Example (Student data)

- ▶ Student Affairs:
name, number,
department, courses taken,
internships, etc.
- ▶ Library:
name, number,
department, books taken,
etc.
- ▶ common data:
name, number,
department, etc.
- ▶ application specific data:
courses taken, internships,
books, etc.

7 / 31

Record Files



- ▶ each application has its own data
- ▶ the application keeps its data in the files that it manages itself

8 / 31

Redundancy

- ▶ the same data is kept in multiple places
 - ▶ waste of disk space

Example

- ▶ the name, number and department of the student is kept both in the Student Affairs and in the Library

9 / 31

Inconsistency

- ▶ if there are multiple copies of the same data, they can become different

Example

- ▶ the name of the student can be recorded as "Victoria Adams" in Student Affairs and "Victoria Beckham" in the Library

10 / 31

Integrity

- ▶ records can refer to invalid data
 - ▶ especially in coded information

Example

- ▶ the department of the student can be recorded as "Control and Computer Engineering" but there might not be such a department anymore

11 / 31

New Applications

- ▶ same work has to be done for each new application

Example

- ▶ a new application will be developed for the Scholarship Office

12 / 31

Policy Gaps

- ▶ no standards in the applications of the institution
 - ▶ differences in paradigms, methods, programming languages, etc.
 - ▶ data transfer between applications
- ▶ each department considers only its own requirements

13 / 31

Security

- ▶ hard to define detailed security arrangements
- ▶ security depends only on the operating system

14 / 31

Data Dependence

Definition

data dependence:

the application code depends on the organization of the data and the access method

- ▶ hard to make any changes in the code

15 / 31

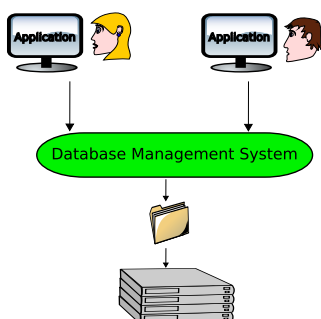
Data Dependence

Example

- ▶ the student number is a string in Student Affairs but a number in the Library
- ▶ the Student Affairs application keeps a B-tree index on the student number and uses B-tree search algorithms in queries

16 / 31

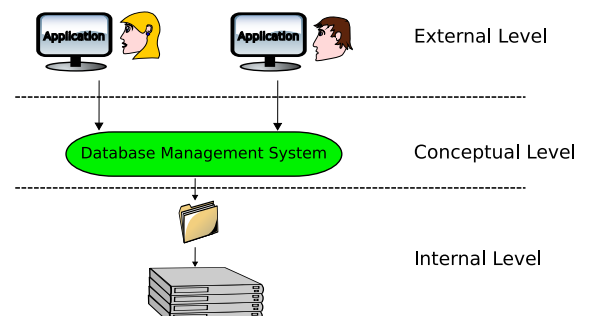
Database Management Systems



- ▶ data is kept in a shared system
- ▶ applications access data over a common interface

17 / 31

ANSI/SPARC Architecture



18 / 31

External Level

- ▶ from the end user's perspective:
 - ▶ the data needed by that end user
 - ▶ the interface of the application that she is using
- ▶ from the application programmer's perspective:
 - ▶ the programming language she uses
 - ▶ the extensions to this language for database operations:
data sublanguage

19 / 31

The Conceptual Level

- ▶ the entire data
- ▶ where the data independence is achieved
- ▶ **catalogue:**
definitions that describe the content of the data
 - ▶ databases
 - ▶ data types, integrity constraints
 - ▶ users, privileges, security constraints

20 / 31

The Internal Level

- ▶ the physical level
- ▶ how the data is represented
 - ▶ files, records
- ▶ how the data is accessed
 - ▶ pointers, indexes, B-trees

21 / 31

Conversions

- ▶ conversions between levels for abstraction

Example (conceptual - external)

- ▶ give the student number as a string to the Student Affairs application and as a number to the Library application

Example (conceptual - internal)

- ▶ generate an index on the student number

22 / 31

Administrator Roles

- ▶ *data administrator:*
makes the decisions
 - ▶ which data will be stored?
 - ▶ who can access which data?
- ▶ *database administrator:*
applies the decisions
 - ▶ defines the conceptual-external/internal conversions
 - ▶ adjusts system performance
 - ▶ guarantees system availability

23 / 31

DBMS Functions

- ▶ data definition language
- ▶ data manipulation language
- ▶ checking whether data manipulation requests conform to integrity and security constraints
- ▶ processing simultaneous requests properly
- ▶ performance

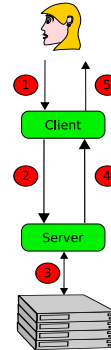
24 / 31

Client/Server Architecture

- ▶ **server:**
provides the DBMS functions
- ▶ **client:**
provides interaction between the user and the server
 - ▶ vendor supplied tools (query processors, report generators, etc.)
 - ▶ applications developed by application programmers

25 / 31

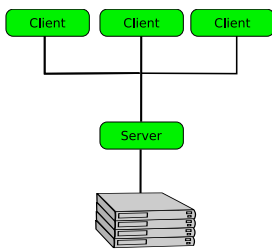
Architecture



- ▶ the client and the server can be on the same computer

26 / 31

Multiple Clients / Single Server



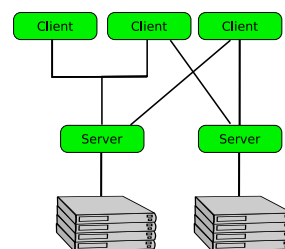
- ▶ many clients can connect to a single server

Example (Bank)

- ▶ a server in the computer centre
- ▶ clients in branches

27 / 31

Multiple Clients / Multiple Servers



- ▶ the servers can also be distributed

Example (Bank)

- ▶ each branch is the server (and client) for its own accounts
- ▶ a client for other branches' accounts

28 / 31

SQL

- ▶ *Structured Query Language*
 - ▶ data definition language
 - ▶ data manipulation language
 - ▶ interaction with general purpose programming languages
- ▶ history
 - ▶ started by IBM in the 1970s
 - ▶ standards: 1992, 1999, 2003

29 / 31

SQL Products

- ▶ Oracle
- ▶ IBM DB2, Progress, MS-SQL, Sybase
- ▶ open source: PostgreSQL, MySQL, Firebird
- ▶ embedded: SQLite, MS Access

30 / 31

References

Required text: [Date](#)

- ▶ Chapter 1: An Overview of Database Management
 - ▶ 1.4. [Why Database?](#)
 - ▶ 1.5. [Data Independence](#)
- ▶ Chapter 2: [Database System Architecture](#)