COMP67321

Introduction to Database Systems and Architectures

Goran Nenadic

School of Computer Science
University of Manchester

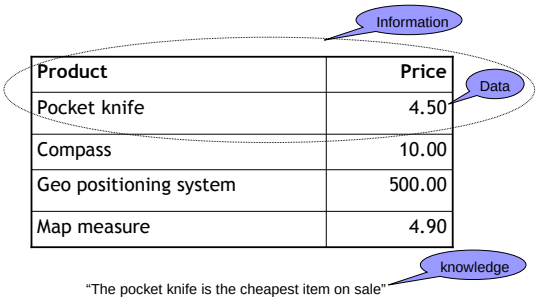
1

Aims

- Identify the roles of databases in information management
- Provide a summary of database technologies
  - databases, DBMS
  - advantages, disadvantages
- Provide an overview of database concepts and architectures

2

Data, information & knowledge



Product	Price
Pocket knife	4.50
Compass	10.00
Geo positioning system	500.00
Map measure	4.90

"The pocket knife is the cheapest item on sale"

3

continued

Data, information & knowledge

- Data
  - Raw, unsummarized, and unanalyzed facts
- Information
  - Data processed and modeled into a meaningful form
- Knowledge
  - Knowing what the information means and how to use it

4

Data in Information Systems

- An information system is a model about some phenomena in the world (application domain) that
  - records facts pertinent to the application domain
  - retrieves, updates and deletes these facts
  - derives other facts from existing ones
- Software Engineering
  - designing and implementing an IT-based IS

5

Examples of application domains

- Purchases from the supermarket
- Purchases using your credit/debit card
- Booking a flight at the travel agents
- Renting a book
- Taking out a mortgage
- Buying from the Internet
- Studying at a university

6

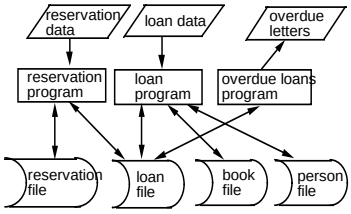
File-based approach

- Store data in (separate) files
- Data is distributed and replicated across files
- Each program defines and manages its *own* data
  - files dedicated and linked to application programs

7

Library example

- Files and application programs for book reservations, loans, book information and person information
- *ALL* these need to 'talk' to each other



8

Limitations of file-based approach

- Separation and isolation of data
  - each program communicates with a file (or sets of files) and maintains its own set of data
  - users of one program may be unaware of (potentially useful) data held by other programs
- Duplication of data
  - same data is held by different files accessed by different programs
  - *wasted space* and more importantly *potentially different values* and/or *different formats* for the same item(s)

9

continued

Limitations of file-based approach

- Data dependence
  - data storage/file structure is defined in the program code
- Incompatible file formats
  - programs are written in different languages, and so cannot easily access each other's files.
- Fixed queries/proliferation of application programs
  - programs are written to satisfy particular functions
  - any new requirement needs a new program and may require a new set of files

10

Important problem

- Updating one file may require updating others
  - Consistency issues
    - Do ***all*** files reflect current situation?
  - Responsibility issues
    - Who is responsible for updating files?
    - If different people, then back to 'consistency issues'
  - Checking issues
    - Can all users check before making any request or operation that all files are correct, up to date and consistent?

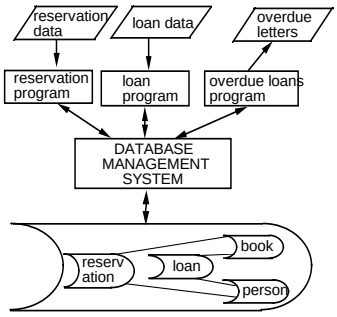
11

File-based approach

- Summary
  - Definition of data is embedded in application programs, rather than being stored separately and independently.
  - Difficult to re-use, update and check consistency: data redundancy and inconsistency
  - No control over access and manipulation of data beyond that imposed by application programs.
- Solution:
  - Databases and Database Management Systems

12

Database approach



13

What is a database (DB)?

- A database is a collection of related data that is used by application systems in a given organisation
- Contains/represents *known facts* and *relations* between entities of interest
- Provides description of data to enable program–data independence

14

continued

What is a database (DB)?

- DB is a shared collection of logically related data (and a description of this data), designed to meet the information needs of an organization.
  - Logically related data comprises entities, attributes, and relationships
- Database management system (DBMS) is a software system that enables users to define, create, and maintain the database and that provides controlled access to this database.

15

Properties of databases

- *Structured* collection of data
- *Logically coherent* — so it makes sense
- *Inherent meaning — information* vs. data
- *Specific purpose* — intended user group(s)
- *Representation of the real world (mini-world)* — changes in the real world reflected in the database

16

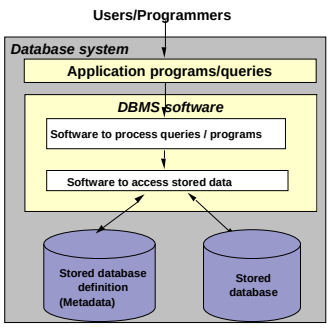
Database system

- Four major components
  - **data**
  - **hardware**
  - **software**: database management system (DBMS)
    - resides between the physical database and the users
    - enables users to create and maintain the DB
  - **users**
    - application programmers, end users, database administrators
    - other applications

17

continued

Database system



18

Database scheme and instances

- DB scheme: the description and definition of DB
  - **meta-data**: description of schema objects (constructs) and constraints
  - need to be specified during design
- Database instances
  - real data that is stored in a given DB at a given point in time

19

Characteristics of DB approach

- Self-describing nature
  - meta-data: complete definition and description of structure of data and constraints
  - includes type and format of data
  - stored in the **database catalog**
- Insulation between programs and data, and data abstraction
  - program-data independence
  - **data model**: conceptual representation of data (no details on implementation)

20

continued

Characteristics of DB approach

- Multiple views of data
  - different perspectives of the DB
  - virtual data
- Sharing data and multi-user transaction processing
  - multiple users can access DB at the same time
  - concurrency control and transactions

21

An Overview of Data Models

22

Data models

- Provide **data abstraction** and **description**
- A set of concepts used to describe the structure of a database
  - data types, relationships, constraints, semantics and operational behaviour
- An abstract, self-contained, logical definition of the objects, operations, etc, that together constitute the *abstract machine* with which users interact
  - objects allow us to model the *structure* of data
  - operations allows us to model its *behaviour*

23

Data models – brief history

- 1964/65: hierarchical and network data models
- 1970: proposal of relational data model
- 1976: entity-relationship (ER) data model
- 1982: advent of relational DBMs
- Early 90's: Object Oriented data model
- Mid 90's: WWW (with HTML)
- Late 90's: XML

24

Data models – categories

- **high-level** or **conceptual** data models
  - how most of end users perceive data
- **low-level** or **physical** data models
  - how data is stored
- **representational** or **implementation** data model
  - models that fit between the two
  - understood by end users but reflects organisation of data in the DB

25

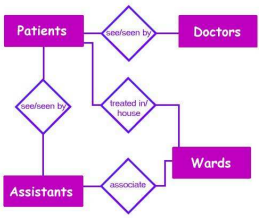
Example: ER data model

- ER = entity-relationship
- Example of high-level data model
- Conceptual data model that views the "real world" as *entities* and *relationships*
- Main concepts
  - entities and attributes
  - relationships

26

ER data model – diagrams

- used to visually represents the model



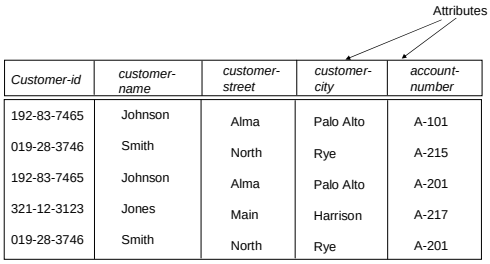
27

Relational data model

- An example of representational data model
- Data is represented as **rows** in **tables**, interpreted as true propositions
- Operators are provided that directly support the process of inferring additional true propositions from given ones
  - joins, averages, etc.

28

Relational model – example



Customer-id	*customer-name*	*customer-street*	*customer-city*	*account-number*
192-83-7465	Johnson	Alma	Palo Alto	A-101
019-28-3746	Smith	North	Rye	A-215
192-83-7465	Johnson	Alma	Palo Alto	A-201
321-12-3123	Jones	Main	Harrison	A-217
019-28-3746	Smith	North	Rye	A-201

29

Object data model

- OODBs – integration of two technologies: DBs and OO paradigm
- Support for
  - abstract data types
  - encapsulation of operations
  - inheritance
  - better integration with programming languages
- Main elements: objects and classes, encapsulation, interface, polymorphism

30

Database management systems and architectures

31

DBMS

- DBMS = Database Management System
- Collection of programs that enables creation, usage and maintenance of a database
- Bridges the gap between the users and data
- General-purpose and complex software
- Most widely used (besides operating systems)

32

DBMS functions

- **Defining** DB: specify data types, structures, etc.
- **Constructing** DB: storing the data
- **Manipulating** DB: querying, searching, retrieval and updates
- **Sharing** DB: multiple users and processes
- **Protecting** DB: mulfunctioning and security
- **Maintaining** DB: evolving the database

33

DBMS languages

- Data definition language (DDL)
  - Permits specification of data types, structures and any data constraints.
- Data manipulation language (DML)
  - General enquiry facility (query language) of the data.

➢ *We will see examples of DBMS languages as part of the course unit* (SQL, ODL, OQL)

34

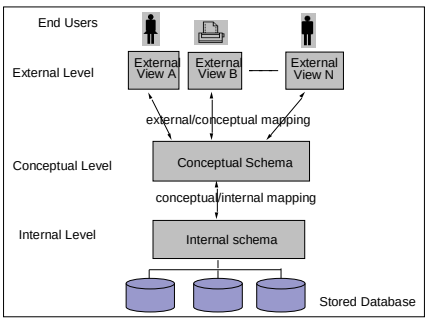
ANSI/SPARC DBMS architecture

- High-level, three schema architecture
- DBMS is divided into three levels (schemas)
  - **internal** (*physical*) level concerns the physical storage of the data
  - **external** (*user logical*) level concerns the way the data is seen by end users
  - **conceptual** (*logical*) level is a level of indirection between the other two

35

continued

ANSI/SPARC DBMS architecture



36

continued

ANSI/SPARC DBMS architecture

- internal schema (HOW)
  - uses a physical data model
  - describes details of data storage and access
- conceptual schema (WHAT)
  - describes the structure of DB
  - hides details of physical storage
- external schemas
  - describes the part of DB that a specific user views

37

Views

- Not everybody needs to see everything and not everybody should be allowed access to everything
  - access control, security, privacy
- Allow each user to have his or her own view of the database
- A view is essentially a subset of the database

38

Benefits of views

- Reduced complexity
  - User is shielded from the complexity of a large database
- Provide a level of security
- Provide a mechanism to customise the appearance of the database
- Present a consistent, unchanging picture of the structure of the database, even if the underlying database is changed

39

Characteristics of DBMS

- Providing storage for efficient query processing
  - indexes
  - query processing and optimisation
- Controlled redundancy
  - avoid unnecessary duplication of data
  - for: consistency, update and storage efficiency
- Back-up and recovery
  - recovery from hardware and software failures

40

continued

Characteristics of DBMS

- Access control
  - security and authorisation subsystems
- Multiple user interfaces
  - query languages, APIs, forms, GUIs
- Representation of complex relationships
  - interrelated data
- Enforcing various integrity constraints
  - “semantics” of data
  - primary and foreign keys

41

continued

Advantages of DBMS

- Enforcing standards
  - data presentation, formats, terminology
- Flexibility
  - evolutionary changes of DB
- Reduced application development
  - when the same data needs to be used by many
- Economies of scale
  - reduces overall costs of operation and management

42

Designing Databases

43

Overview of DB design process

- Two main activities:
  - data (and database) design
  - applications design
- Database design
  - designing a conceptual schema for use by database applications
- Applications design focuses on the programs and interfaces that access the database
  - generally considered part of software engineering

44

DB development lifecycle

- DB planning
  - integrating DB with the overall enterprise plans and IS strategy
- DB definition
  - DB scope and boundaries, users, etc.
- Requirement collection and analysis
  - fact-finding: description of data and processes that use the data
  - specification of database and functional requirements

45

continued

DB development lifecycle

- DB design
  - bottom-up, top-down, inside-out, mixed
  - data modelling: conceptual, logical, physical
- DBMS selection
  - select appropriate DBMS
- Application design
  - user interface, application programs, transaction implementation, etc.

46

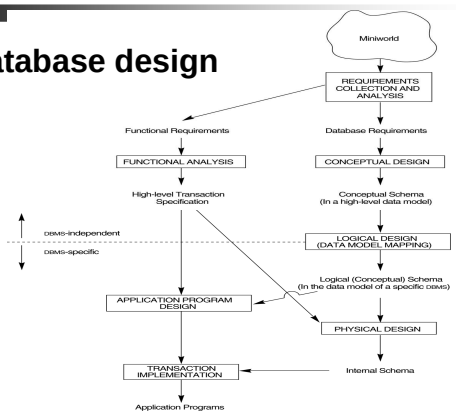
continued

DB development lifecycle

- Prototyping
  - working model
- Implementation
  - physical realisation
- Data conversion and loading
  - load existing data
- Testing
- Operational maintenance

47

Database design



48

Summary
– the role of databases

49

Role of databases

- A database is a central/critical component of any contemporary organization's memory system
- DBMS are the most widely available technology (besides operating systems)
- Rapid expansion primarily because of the increase in Web applications and other emerging application areas

50

Advantages of Databases

- Control of data redundancy
- Data consistency
- More information from the same amount of data
- Sharing of data
- Improved data integrity
- Improved security
- Enforcement of standards
- Economy of scale

51

continued

Advantages of Databases

- Balance conflicting requirements
- Improved data accessibility and responsiveness
- Increased productivity
- Improved maintenance through data independence
- Increased concurrency
- Improved backup and recovery services

52

Disadvantages of Databases

- Complexity
- Size
- Cost of DBMS
- Additional hardware costs
- Cost of conversion
- Performance
- Higher impact of a failure

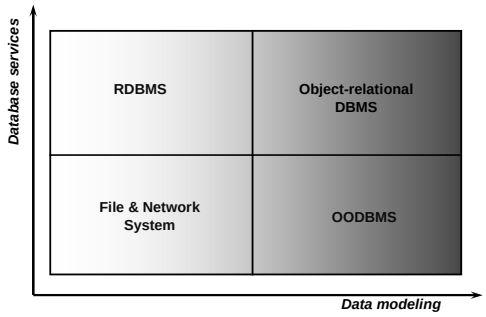
53

Database market

- Database research and technology are more than three decades old.
- Current market value is about $20 billion per year worldwide.
- Average growth of 20% per year since 1965.
- Projected growth of about 20% for the immediate future (?)

54

Current status



55

Reading for this lecture

- Chapters 1 and 2 in [Elmasri & Navathe]
- Chapters 1 and 2 in [Connolly & Begg]

56