DB generalities : What is a database?

Chris Date:

Database = computer-based record keeping system

R.W. Engles: "A Tutorial on DB Organisation" (1974)

collection of stored operational data used by the applications system of a particular enterprise

enterprise: hospital, university, bank, company etc

operational data:

data on products, accounts, patients etc

typically **persistent** cf conventional program IO data

DB generalities : Why use a database?

Case-study: Banking (after Korth &Silberschatz Chap. 1

? How to meet needs using a traditional file-processing system supported by a conventional OS

Files: permanent records of customers, accounts

Applications programs (APs): enable user to modify files

- to credit or debit an account
- to add a new account
- to find the balance in an account
- to generate monthly statements

APs written by systems programmers as required

new requirements ⇒ new files + new programs

Original context for data modelling 1

- 1970s style applications
  - unsophisticated computer ***users***
    - batch mode interaction
      - modest response times
      - no visualisation or GUI
    - modest expectations for ease-of-use
      - programming perceived as technical
  - simple computing ***infrastructure*** and environment
    - no PC, web etc
    - no live feeds of data
    - textual interaction the norm

Original context for data modelling 2

1970s style applications

Business context

simple business model, limited automation, access etc

low volume of data

not initially distributed

Computing context

- existing/emerging DB proposals unconvincing
- computers not very powerful
- human and computing resources very expensive

Summary of issues for data management

Problems of data management for file systems
that DBs were originally intended to address:

- Data redundancy and inconsistency
- Difficulty in accessing data
- Data isolation
- Concurrent access anomalies
- Security problems
- Integrity problems

DB generalities : +’s and –’s of DB use

Conventional file systems have certain characteristics

… will review the key issues for data management:

+ indicates a positive impact of using a database

– indicates a potentially negative impact of using a DB

DB generalities: Issues for data management

Problems that DBs were originally intended to address:

- Data redundancy and inconsistency
- Difficulty in accessing data
- Data isolation
- Concurrent access anomalies
- Security problems
- Integrity problems

DB generalities: Issues for data management

Data redundancy and inconsistency

- each programmer potentially uses different format file, develops at different stage in history of enterprise
- data duplicated

+ in a DB rationalise and standardise data

[rationalise: conceptually create an authentic shared source for data]

… rationalise doesn't necessarily mean *centralise*

DB generalities: Issues for data management

Data redundancy and inconsistency

- each programmer potentially uses different format file, develops at different stage in history of enterprise
- data duplicated

– compromises are needed,

where users suit themselves, can get efficient results

vs no perfect data organisation to suit all users

duplication can give insurance against info loss

DB generalities: Issues for data management

Difficulty in accessing data

- have to respond to unforeseen requests, hence new functionality
- in file-processing environment, have to write new programs, and possibly devise new data structures

+ in a DB, simplify access and manipulation by intelligent organisation of data cf. modelling approach to requirements, as e.g. in use of UML in OOSE

DB generalities: Issues for data management

Data isolation

- data has to be retrieved from many sources when APs written

+ in DB, aim to hide the source and form of physical data by viewing the data at a higher level of abstraction

– automation decreases the amount of human interaction with data

risk of corrupted data passing between integrated files is greater

DB generalities: Issues for data management

Concurrent access anomalies

- would like multiple access for efficiency and faster response time e.g. simultaneous withdrawal

+ concurrency can't be managed without a form of overall control

DB generalities: Issues for data management

Security problems

- would like to restrict access to authorised users for confidential info

+ security can't be managed without a form of overall control

– issue as to whether this control is most easily exercised inside or outside computer system e.g. non-trivial problem to determine what can be inferred from responses to queries that aren't explicit

DB generalities: Issues for data management

Integrity problems

- data in file system must satisfy integrity constraints constraints may arise dynamically: difficult to modify programs to cope with this; also hard to guarantee integrity if data is stored in different files

+ automated management demands some form of overall control

– automation reduces scope for human intervention / interpretation

DB generalities: Issues for data management

Conclusion from above discussion ...

For many commercial applications (as in enterprises above) good solution is offered by a **database management system** (DBMS). A DBMS is an unconventional OS operating over a structured file system. The +’s above indicate some of the positive benefits of the use of a DBMS.

DB generalities: Issues for data management

For many commercial applications (as in 1970s-style enterprises above) good solution is offered by a **database management system** (DBMS).

A DBMS is an unconventional OS operating over a structured file system.

Generalities of DBs: the DBMS concept

Motivating idea:

devise an abstract model of the entire corpus of operational data that simplifies the data processing activity, so that

- simple queries can be handled without writing new application programs
- where applications programs must be written, the task of accessing and manipulating operational data consistently and efficiently is greatly simplified

DB generalities: the ingredients of a database

Data

- integrated
- shared
- possibly distributed

Hardware

- primary storage + secondary storage

Software

- database management system: DBMS
- protects users from hardware level detail
- serves the needs of all users

DB generalities: the ingredients of a database

Users

end-user:

- non-specialist accessing data via a query language
- naïve user accessing data via a special-purpose interface

performs data retrieval and update (extend / modify)

applications programmer:

- writes programs that use the DB by embedding queries to the DB in a HLL
- develops interfaces for the naïve user

DB generalities: the ingredients of a database

Users

- Database Administrator (DBA):
  - responsible for overall control
  - decides what data is to be stored
  - designs the conceptual scheme
    - used to represent the operational data
  - implements authorisation checks
  - decides strategy for backup and recovery
  - monitors performance
  - oversees modification to suit user requirements

DB generalities: data abstraction in a database

Data abstraction addresses issues of **use**, **design**, **management** and **implementation** in a database

The data model serves to describe in a formal manner the way in which data is viewed at three different levels of abstraction:

physical level

conceptual level

view level

DB generalities: data abstraction in a database

- **physical** level:

  how is the data actually represented in the hardware?

  bits, bytes

- **conceptual** level:

  what meaningful relationships are expressed by the physical data?

  entities, and relationships between entities

- **view** level:

  what particular relationships are required by users?

  more abstract because partial typically very high-level knowledge constitutes the view

DB generalities: data abstraction in a database

Illustrating data abstraction:

Data base stores the date of birth of a client as a bit string. When we identify the senior citizens, we find all clients aged over 65.

Representations at different levels of abstraction

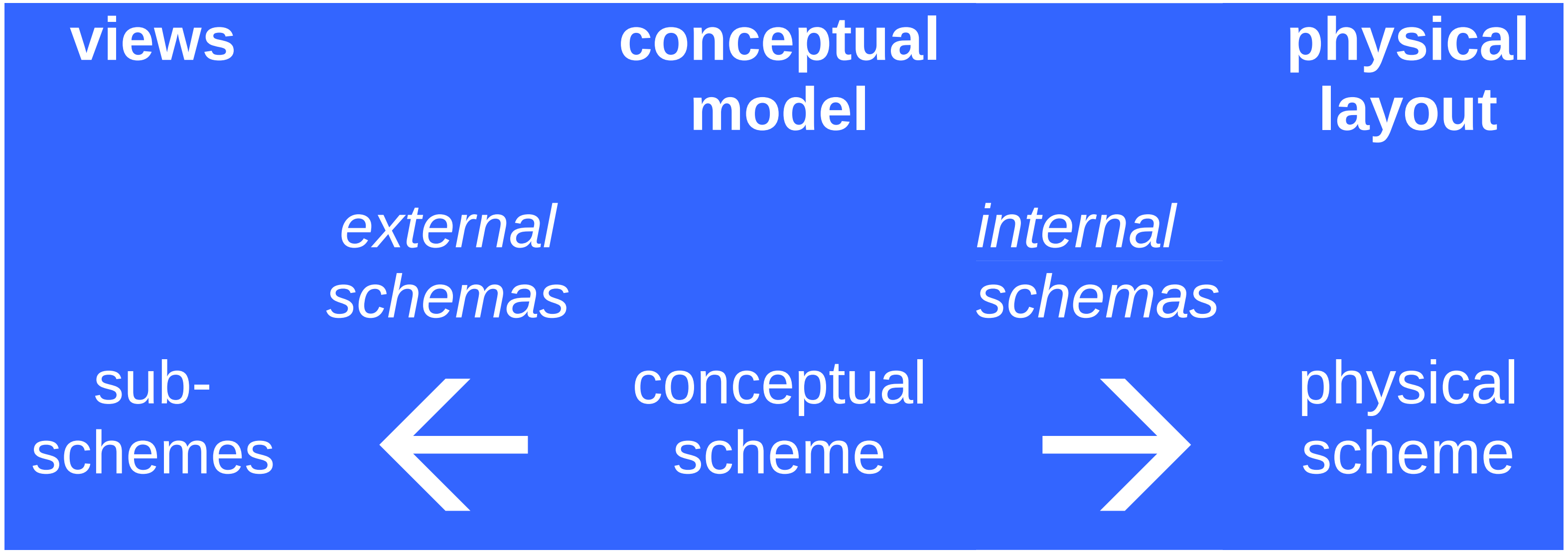
- **conceptual** date of birth of a client
- **physical** the bit string that records this information
- **view** refers to age, which isn't stored in the DB.

DB generalities: data abstraction in a database

USE

DESIGN
&
MANAGEMENT

IMPLEMENTATION



views

conceptual model

physical layout

external schemas

internal schemas

sub-schemes ← conceptual scheme → physical scheme

DB generalities: data abstraction in a database

- The DBA conceives the database in terms of the conceptual model.
- Users and application programs access the physical data via the conceptual model.
- **physical data independence**: protecting the conceptual model from change when the physical organisation changes
- **logical data independence**: protecting the user from the need to change views when the conceptual model changes

Recall - Generalities of DBs: the DBMS concept

Motivating idea:

devise an abstract model of the entire corpus of operational data that simplifies the data processing activity, so that

- simple queries can be handled without writing new application programs
- where applications programs must be written, the task of accessing and manipulating operational data consistently and efficiently is greatly simplified

DB generalities: data models for a database

Many different paradigms have been proposed for developing abstract data models for databases

There are two principal kinds of abstract data model:

- object-based models
- record-based models

The earliest database systems were record-based

- this reflects the file system culture that they displaced

DB generalities: data models for a database

Object-based models

The main models in this category are

- entity-relationship models
- object-oriented data models

Others include semantic and functional data models.

E-R model widely used to model data abstractly

OO model gaining acceptance in practice: effectively represents data + operations on data.

DB generalities: data models for a database

Record-based Logical Models

Used at the conceptual and view levels. Specify both

- overall logical structure of the database
- higher-level description of the implementation.

Record-based because uses records in fixed-format of several types. This simplifies implementation: cf. trend towards richness and variety in structures used to implement OODBs

DB generalities: data models for a database

Varieties of record-based logical model

- hierarchical model

  records & links organised as a family of trees

- network model

  records & links organised as a family of graphs

- relational model

  uses tables to record relationships between data

DB generalities: data models for a database

Physical Data Models

There are also models of data at the lowest level of abstraction, concerned with physical organisation. These are not our primary concern in this module.

Relevant issues for relational databases include:

- are data tables stored using hashing?
- how are data tables indexed?
- how are entries in data tables encoded and ordered?
- what algorithms are used to retrieve and update?

DB generalities: classical database features

Instances and Schemes

State of a DB changes over time: distinguish structure of DB from current state as defined by the data in it.

overall design of DB	=	database scheme
current content of DB	=	instance of the DB

Useful analogy with procedural variables:

database scheme	*type definition for variable*
instance of database	*value of the variable*

DB generalities: classical database features

Data Definition Language (DDL)

database scheme is defined using a DDL

compiling the DDL description creates a Data Dictionary

the storage and access methods used by the DB

are specified in a storage and definition language

Implementation details for storage are usually hidden from users

DB generalities: classical database features

Data Manipulation Language (DML)

data manipulation means accessing DB to

retrieve, insert, delete, or modify data

most common use of DML is for data retrieval: informally described as "querying the DB"

retrieval component of DML = query language

(and by abuse, sometimes use term ‘query language’ as synonym for DML)

DB generalities: classical database features

Varieties of Data Manipulation Language

There is a tension between

- efficiency at physical level
- intelligibility / ease of use at higher level

Have both procedural and non-procedural DMLs

procedural: requires knowledge of data implementation

non-procedural: need only specify what data is needed

DB generalities: classical database features

Data Manipulation Languages for typical data models

object-based, hierarchical, network models have procedural DMLs

user can take explicit responsibility for optimising queries, but needs knowledge of data organisation

relational models use non-procedural DMLs

can formulate queries without knowledge of data organisation, but implementation has to be optimised