

Database Architecture Models and Design

Ian Horrocks
Carole Goble
John Sargeant
Simon Harper

Contact Details & Resources

- Ian Horrocks
 - Room 2.75 <Horrocks@cs.man.ac.uk>
- Carole Goble
 - Room 2.28 <Carole@cs.man.ac.uk>
- Simon Harper (lab)
 - Room 2.120a <sharper@cs.man.ac.uk>
- John Sargeant
 - Room 2.69 <jsargeant@cs.man.ac.uk>
- <http://www.cs.man.ac.uk/~horrocks/cs2312/>

Objectives

- Raise awareness about databases, database design and database management systems
- Enable you to design and use a database to support an application
- To understand the implications of your design
- To realise that designing databases is non-trivial and requires imagination, flexibility and thought

CS2311 Course Structure

- Introduction and motivation
- A database model
 - Relational database model
 - Integrity
 - Oracle DBMS
 - SQL and Embedded SQL
- Database design
 - Logical design
 - Conceptual design
 - Mapping conceptual to logical
 - Pragmatic issues
 - Physical design
 - Integrity and correctness
- Architecture of DBMS
 - Client-server
 - Open architectures
- Transaction Processing
 - Multi-user & Concurrency
 - Resilience & Recovery

Laboratory Exercise: Sweeney Tours

- Use, design and build a database to support a holiday booking system
- Oracle 8i DBMS
- Example classes and labs integrated
- Exercise 1 & 2 individually
- Exercise 3 & 4 as pairs



IMPORTANT ATTENDANCE POLICY

- You are required to attend Lab Classes, if you attend you will be given an Automatic Extension to submit at the start of the following weeks example class. If you do not attend you will be given NO extension and you will be expected to submit by the end of the Lab Class and no extension will be forthcoming. There may be exceptional circumstances as to why you did not attend the Lab Class or Hand in by the Automatic Extension date. These cases will be examined individually BUT we are very strict on attendance and handing in deadlines. You should note that we find there is a definite correlation between the people who attend Lab Classes and Examples Classes and those that pass their exams.

Recommended Texts

- Elmasri and Navathe
 - Fundamentals of Database Systems
 - 3rd Edition, Addison Wesley
- Atzeni, Ceri, Paraboschi and Torlone
 - Database Systems: Concepts, Languages and Architectures
 - MacGraw Hill
- Ullman and Widom
 - A First Course in Databases
- Garcia-Molina, Ullman and Widom
 - Database Systems: The Complete Book

What is a database?

a *structured* collection of information captures the *semantics* of an application

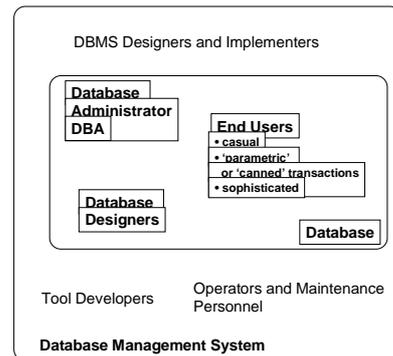
- Logically coherent—so it makes sense
- Inherent meaning—information vs data
- Specific purpose—intended user group
- Representation of the real world—changes in the real world reflected in the database

What is a Database Management System?

a collection of programs and tools to create & maintain a database

- Defining specifying *types* of data
- Constructing storing & populating
- Manipulating querying, updating, reporting

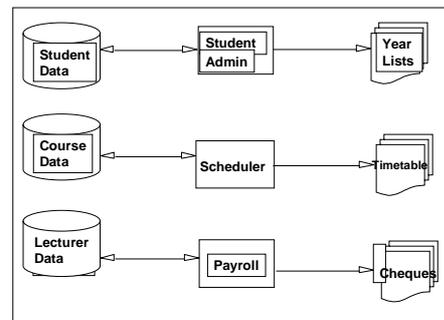
Actors



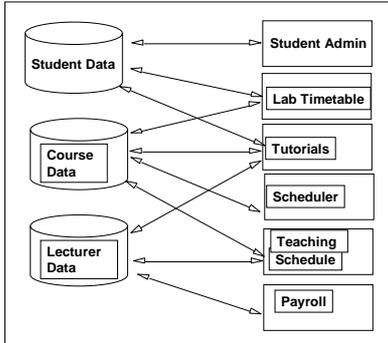
Characteristics of a Database

- Structure
 - data types
 - data behaviour
- Persistence
 - store data on secondary storage
- Retrieval
 - a declarative query language
 - a procedural database programming language
- Performance
 - retrieve and store data quickly
 - Correctness
- Sharing
 - concurrency
- Reliability and resilience
- Large volumes

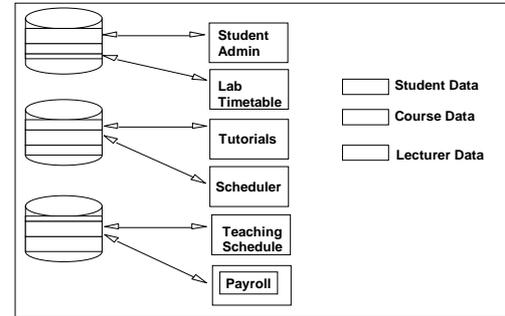
File Management Systems: a physical interface



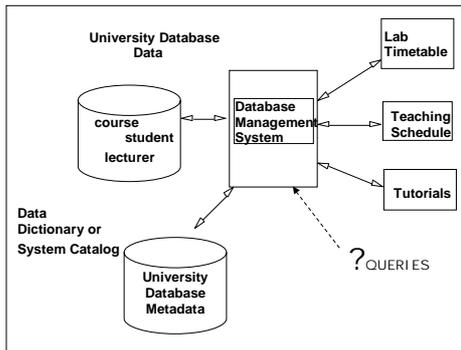
File Management Systems: Sharing data and operations



Sharing data: replication—redundancy



DBMS: A Logical Interface



File Management Systems

- Uncontrolled redundancy
- Inconsistent data
- Inflexibility
- Limited data sharing
- Poor enforcement of standards
- Low programmer productivity
- Excessive program maintenance
- Excessive data maintenance

Database Management System Approach

- Controlled redundancy
 - consistency of data & integrity constraints
- Integration of data
 - self-contained & represents semantics of application
- Data and operation sharing
 - multiple interfaces
- Services & Controls
 - security & privacy controls
 - backup & recovery
 - enforcement of standards
- Flexibility
 - data independence
 - data accessibility
 - reduced program maintenance
- Ease of application development



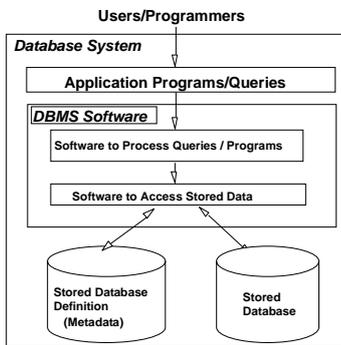
However....

- more expensive
- more complex
- general
- simple
- stringent real-time
- single user
- static

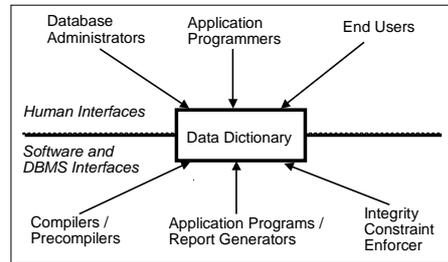
Summary:

- In a file management system data is PHYSICALLY accessed and UNINTEGRATED
- In a database management system data is LOGICALLY accessed and INTEGRATED
 - a data dictionary
 - a query language

Simplified Environment (from Elmasri & Navathe)



Interfaces to a Data Dictionary



Models and Schemas

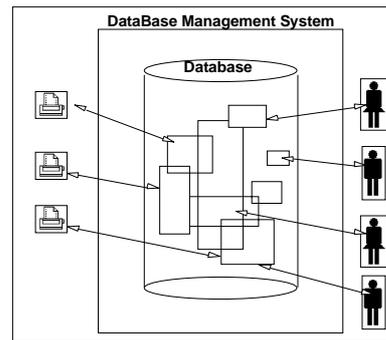
- Model**
 - A structure that demonstrates all the required features of the parts of the real world which is of interest to the users of the information in the model.
 - Representation and reflection of the real world (Universe of Discourse)
- Data Model**
 - A set of concepts that can be used to describe the structure of a database: the data types, relationships, constraints, semantics and operational behaviour.
 - It is a tool for data abstraction
 - A model is described by the *schema* which is held in the *data dictionary*.

Student (studno, name, address)

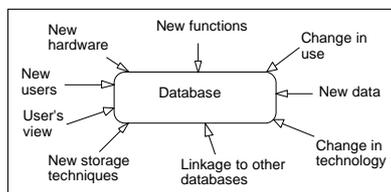
Course (courseno, lecturer) ← Schema

Student (123, Bloggs, Woolton)
(321, Jones, Owens) ← Instance

Sharing—Multiple views of data



Data Independence



- Logical data independence**
 - change the conceptual schema without having to change the external schemas
- Physical data independence**
 - change the internal schema without having to change the conceptual schema

Ansi-Sparc Database Architecture

