Introduction

Example databases

Overview of concepts

Why use database systems

Example Databases

u University

6 Data: departments, students, exams, rooms, ...

6 Usage: creating exam plans, enter exam results, create statistics, build timetables, …

6 Web-based access common

u Bank

6 Data: clients, accounts, credits, funds,…

6 Applications: accounting, transfers (> 50,000 per day), risk management

6 Web-based access for some parts of the application

u Airline

6 Data: flights, passengers, employees, airplanes,…

6 Applications: reservation, booking, creating flight schedules, querying flight schedules

6 Web-based access for some parts of the application

Example Databases (contd.)

u Genetics

6 Data: DNA-sequences (> 1 Mio. bases for simple bacteria), proteins

6 Applications: search similar sequences, predict protein structure, keep track of experiments…

u Online bookstore

6 Data: all what is sold at the bookstore, customer information

6 Applications: keyword search, booking, maintaining shopping basket, checkout, …

u Global Change Research

6 Data: topographic maps, satellite data (NASA Earth Observation System: 1 Terabyte per day)

6 Applications: Analysis and prediction of climate and environmental changes (e.g., greenhouse effect, waldsterben), visualization of data

u Stock market, libraries, etc.

The basic Terminology

u **Database**: collection of data modeling a real world enterprise

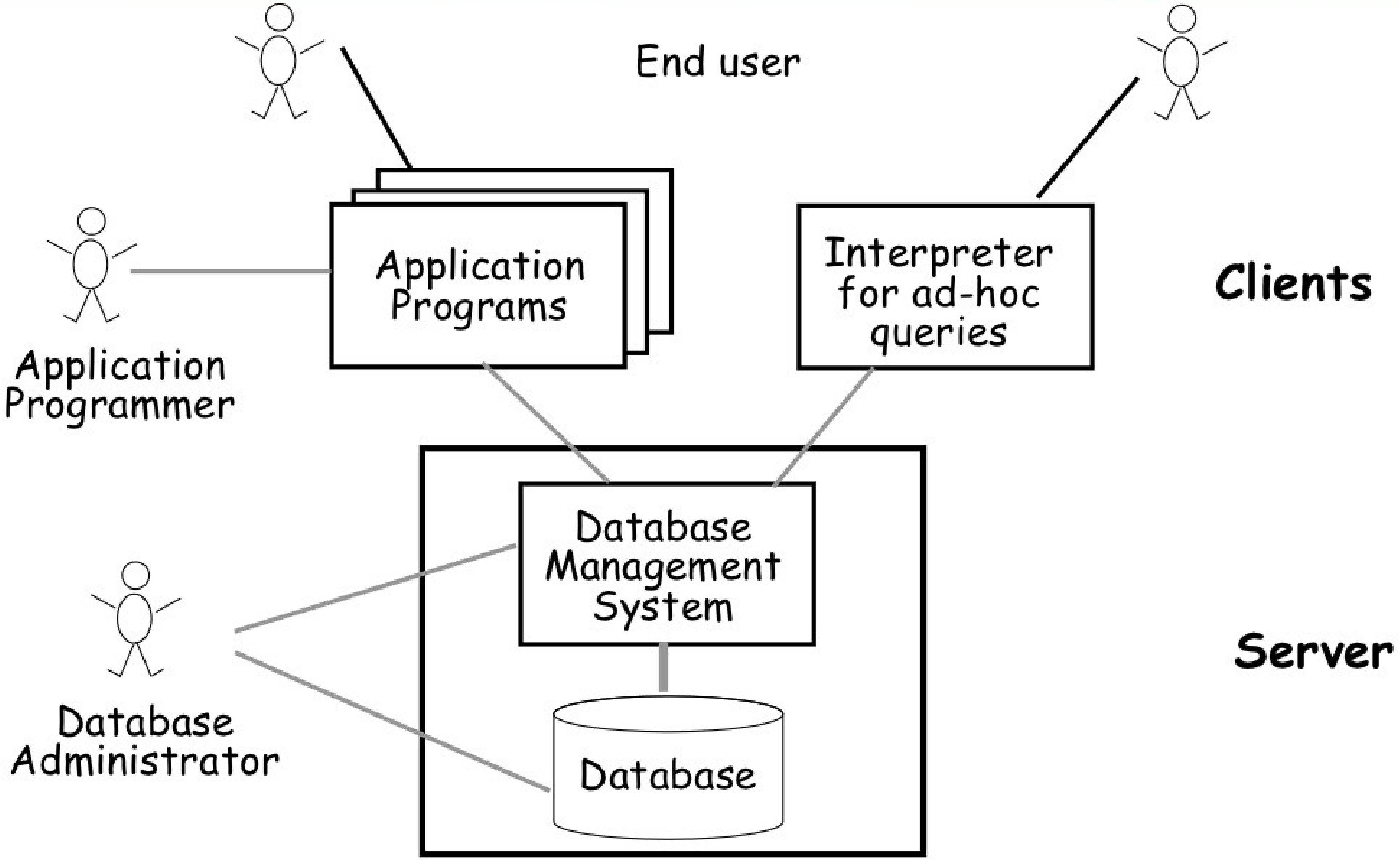
u **Database management system (DBMS) or database system (DBS):** the software package to store and manage the data

u **Application programs**: the software to access and process the data (implements business logic, e.g., enter exam grades, purchase book)

u **Information system** (within this course): database + DBMS + application programs

  6 There exist other types of information systems, based on information-retrieval systems or knowledge-based systems

Client/Server-Architecture



Example DBMS

- Relational DBMS companies like Oracle
- IBM offers its relational DB2 system. With IMS, a non-relational system, IBM is by some accounts the largest DBMS vendor in the world
- Microsoft offers SQL-Server, plus Microsoft Access for the cheap DBMS on the desktop, answered by "lite" systems from other competitors
- There exist several widely used open-source DBMS: PostgreSQL and MySQL being the best known of them
- XML database systems
- Object-oriented database systems
- Embedded database systems

Who wants to study databases?

- “What is the use of all the courses I have taken so far?”
  - This course shows very concrete how Information System is used in the outside world
  - This courses uses a lot of the basics introduced in the 200/300 level courses
- “I want to work in an interdisciplinary environment”
  - Be an application developer building solutions for all kinds of fields working with people from other areas (e-commerce, science, administration, law, etc. etc.)
- “I love the internals of how computers and systems work”
  - Be a database administrator or a DBS developer: a DBMS is an entire operating system and more
- “I am more a theoretical person”
  - Database systems have a very sound theoretical foundation and there are many exciting open problems
- “I want to work with computer languages, human-computer interaction, multimedia, logic, communication, distributed systems, knowledge management,….
  - It’s all there
- “I want to make a lot of money”
  - E-commerce, banks and business: here you are
- “I am not interested in databases”
  - You will have to use them anyway

Data Models

- A **data model** is a collection of concepts for describing data
- A **schema** is a description of a particular collection of data, using a given data model
- Most DBMS are based on the **relational data model**
  - Main concept: **relation**, basically a table with rows and columns
  - Every relation has a **schema**, which describes the columns (also called attributes or fields)
  - Close to how the DBMS stores the data

Students

sid	*name*	*login*	*age*	*gpa*
31	Raja	raj	22	3.4
34	Rani	ran	21	3.3

- *Schemas are defined using a data definition language (DDL)*
- *Data is modified using a data manipulation language (DML)*

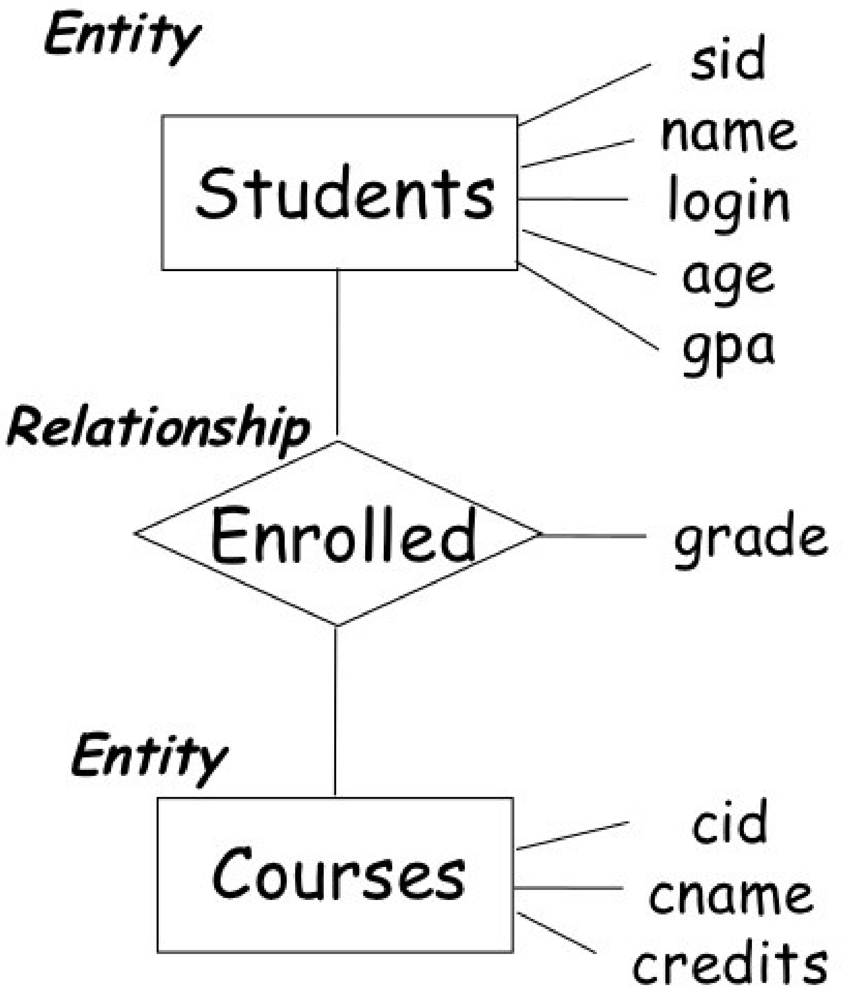
Data Models (contd.)

u A **Semantic Data Model** provides abstract, high-level constructs with which it is “easy” to develop an initial description of the data (schema) in an enterprise

  6 that is, develop a schema using the semantic data model and then translate it into a schema based on the data model provided by the DBMS

  6 for instance, the entity-relation model (ER) allows us to pictorially denote entities and relationships among them

Example: University Database



Students

Relation

sid	name	login	age	gpa

Enrolled

Relation

sid	cid	grade

Courses

Relation

cid	cname	credits

Querying Data

u Querying the database:

  6 Student record: Give me all courses a student XYZ has taken. Show all the grades he/she got plus the class average.

u Inserting data into the database

  6 A student registers for a course

u Updating existing data

  6 The instructor enters the grades for a course

u Specific query languages for DBMS: best known is SQL and OQL

Data Storage and Data Access

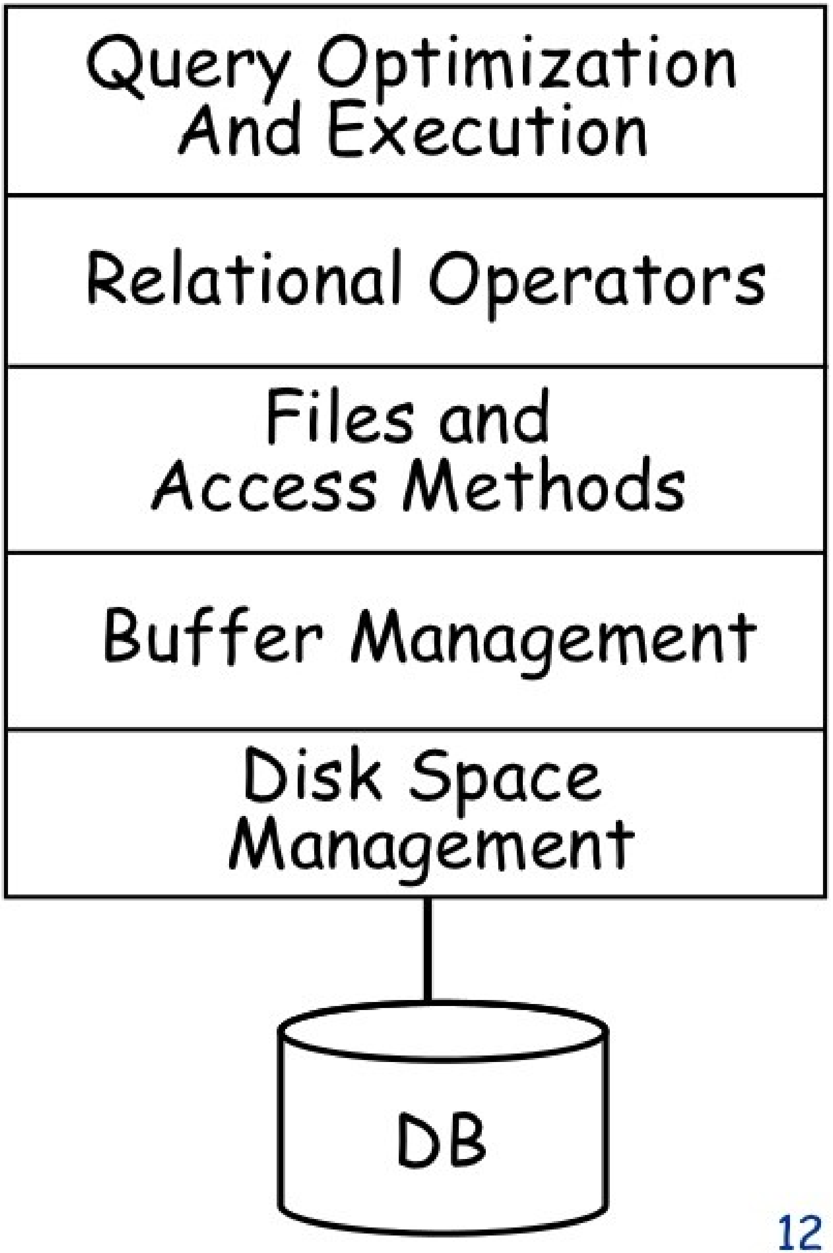
u A typical DBMS has a layered architecture

u Disk space management, buffer management, files and access methods represent (nearly) an operating system on top of the operation system

u Fast and sophisticated data retrieval requires

  6 special index structures

  6 query optimization techniques



Transaction Management: Controlling the Database Access

u Key concept is <u>**transaction**</u>, which is an atomic sequence of database actions (read and write operations on data items)

u A transaction represents a <u>logical unit</u> of operations (from the application point of view)

  6 Often "user program" = transaction

  6 For instance: transfer transaction = debit of account X, credit on account Y

Transactional Properties

- **Atomicity**: all or none of the operations of a transaction should succeed (all-or-nothing property)
  - All operations succeed = commit
  - None of the operations succeed = abort (in the case of failure: undo all operations executed so far)
- **Durability**: the changes of a committed transaction must be persistent even in the case of failures
  - write changes to disk before commit
- **Recovery:** when restarting a failed site, recovery brings the database back to a consistent state
  - Exactly the updates of all committed transactions must be in the database and nothing else
- **Isolation**: don't mess up the database when running several transactions at the same time

Why use a DBMS?
(instead of files)

- appropriate data models (helps to make a good design)
  - offer more than records, arrays and basic data types
- easy definition of data (declarative and set-oriented)
  - define data once with simple constructs instead of spreading definitions over various program modules
- easy access to data
  - query language allows for sophisticated data retrieval using simple query statements; simple creation, deletion and modification of data
- efficient access to data
  - good index structures provided
- data independence
  - application programs receive abstract view of data and are independent of how data is stored and accessed
- persistent data storage
  - Gigabytes of data do not fit in main memory and require special file support
  - guarantee that changes to data are on stable storage

Why use a DBMS? (contd.)
(instead of files)

- 6 data integrity
  - q simple to define constraints to keep data consistent (account must always be above zero)
- 6 security and authorization
  - q very flexible access control and execution control
- 6 good basis to allow different applications to work on the same data
  - q centralize data management
  - q provide unified interface
- 6 uniform data administration
  - q good tools for tuning, upgrading, monitoring, ...
- 6 concurrent access, recovery from crashes
  - q comes for free
- 6 ...

Reduced application development time

Reduced application maintenance

§ Safer

Emphasis of the Course

u How to organize, maintain and retrieve information using a DBMS

  6 **design** of databases

  6 **usage** of DBMS

  6 Understand **how** DBMS **work** in order to use them appropriately

Levels of Abstraction

u Single **conceptual (logical) schema** defines logical structure

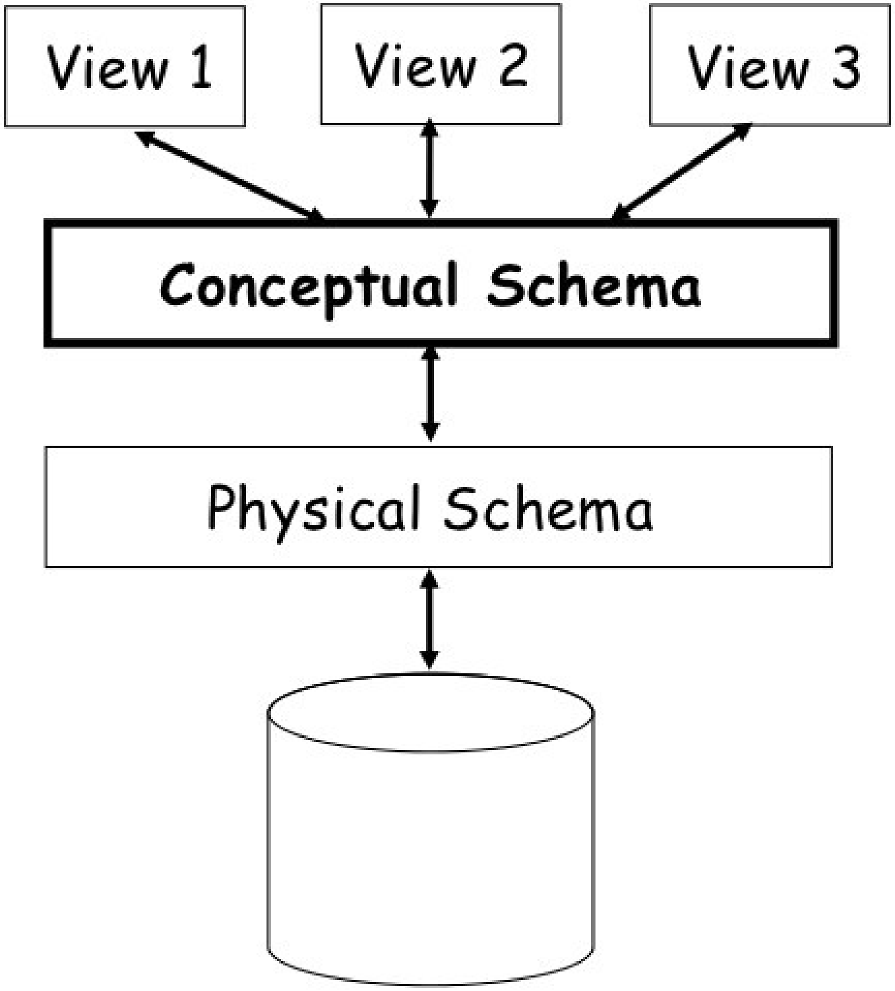
6 Conceptual database design

u **Physical schema** describes the files and indexes used

6 Physical database design

u Different **views** describe how users see the data (also referred to as external schema)

6 generated on demand from the real data



Example: University Database

- Conceptual schema:
  - Students(sid:string, name:string, login:string, age:integer, gpa:real)
  - Courses(cid:string, cname:string, credits:integer)
  - Enrolled(sid:string, cid:string, grade:string)
- Physical schema:
  - Relations stored as unordered files
  - Index on first column of Students
- External schema (view):
  - Course_info(cid:string, enrollment:integer)

Data Independence

- Applications isolated from how data is structured and stored
- **Logical data independence**: protection from changes in logical structure of data
- **Physical data independence**: protection from changes in the physical structure of data