INFO2120/2820: Database Systems I
COMP5138: Database Management Systems

Week 1: Introduction to Database Systems

Dr. Uwe Röhm
School of Information Technologies

Outline

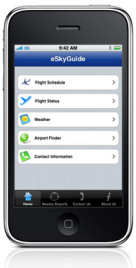
- Introduction
- File Systems vs. DBMS
- Overview of Core Database Functionalities
  - Data Independence
  - Declarative Querying
  - Transactions
- Metadata

Based on slides from Kifer/Bernstein/Lewis (2006) “Database Systems” and also including material from Fekete and Röhm.

(Online Demo)
How many Databases did I just use?

- At least 1
- At least 2
- At least 3
- At least 4
- More?!?

Databases Touch all Aspects of our Life!



Databases on the Internet…

- Ebay (in 2005)
  - More than 100 back-end databases
  - ca. 5 billion SQL/day
- Wikipedia: (as of Oct. 2006 - http://stats.wikimedia.org/EN/ChartsWikipediaEN.htm)
  - ca. 350 servers
  - 249 languages, millions of articles
    (engl.: 1.5M with 5GB data, 4.1 million updates/month)
  - Behind each language at least one database cluster (2+ dbms)

    cf. *Brion Vibber "Scaling and Managing LAMP at Wikimedia", Santa Clara, 2008*.
- May 2008: Yahoo! claims record with 1 Petabyte Database

Data-Intensive Scientific Discovery

- Scientific Research as it evolves over time:
  - **Experimental** (thousands of years ago)
  - **Theoretical** (few hundred years ago)
  - **Computational** (last few decades)
  - **Data-Intensive** (termed by the late *Jim Gray*)

  Cf.: Tony Hey, et al (ed.):
  The Fourth Paradigm: Data-Intensive Scientific Discovery,
  Microsoft Research, 2009.

$$E = mc^2$$

- eScience: "IT meets scientists"
  - Modern scientific instruments allow to automatically collect Terabytes of scientific results and data that is shared world-wide
  - At the essence this means: **Data-intensive research**
    To base theories and results purely on the analysis of this data.
- **eScience depends on effective data management**

Proof-of-Concept: SDSS SkyServer

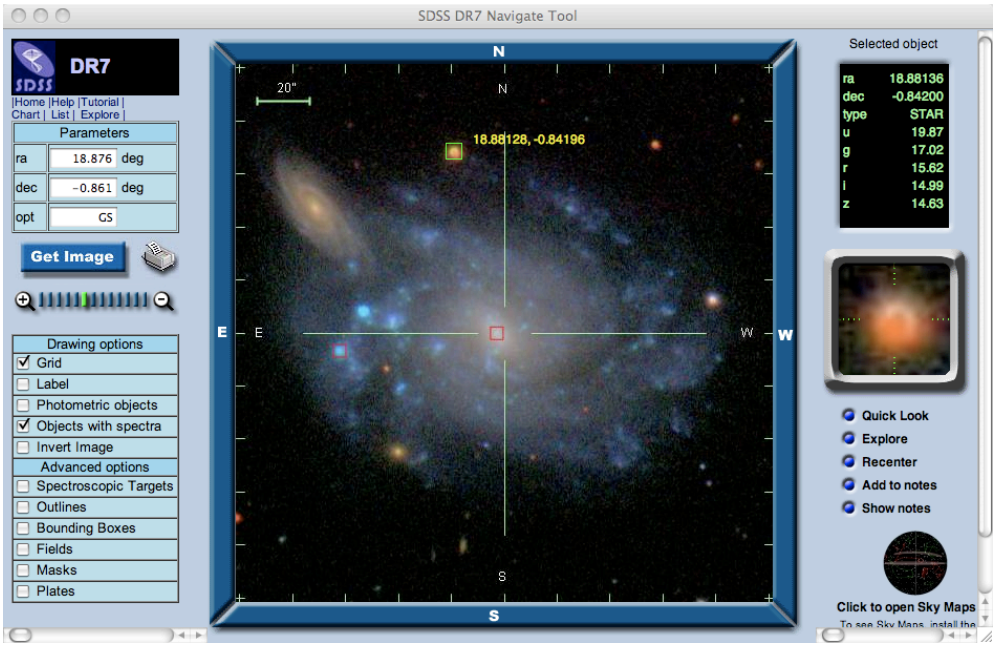
- Website to access data from the Sloan Digital Sky Survey (SDSS)
  - astronomy survey aimed at creating a map of a large part of the universe
  - used a 2.5-meter telescope in New Mexico to automatically take images of about 1/4 of the night sky.

Jim Gray standing next to the main SDSS telescope at Apache Point Observatory, New Mexico.

- SDSS Data Release DR7:
  - 350 million celestial objects
  - 1.51 million spectra of stars, galaxies and quasars
  - over 70TB of raw and processed data

Example: SkyServer

A 'Virtual Telescope' over the SDSS data, backed by database systems, allowing to share and analyse the data using – SQL.



[Screenshot of SkyServer website for SDSS DR7; Source: www.sdss.org]

What is a Database?

- Collection of data central to some enterprise / organisation
- Essential to operation of enterprise
  - Contains the only record of enterprise activity
- An asset in its own right
  - Historical data can guide enterprise strategy
  - Of interest to other enterprises
- State of database mirrors state of enterprise
  - Database is persistent
  - Shared:
    all qualified users have access to the same data for use in a variety of activities.

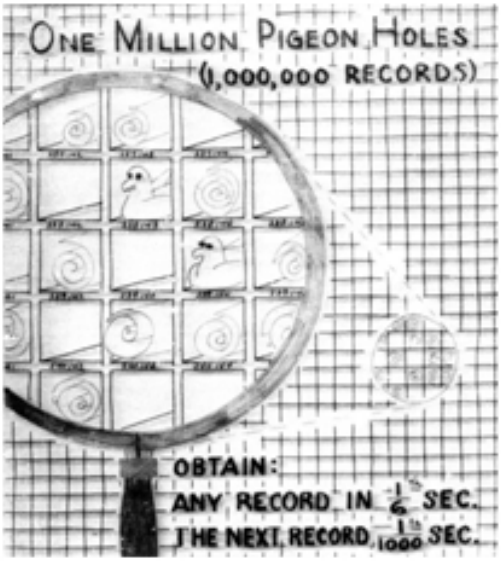
What is a DBMS?

- A **Database Management System (DBMS)** is a program that manages a database:
  - Stores the database on some mass storage providing fail safety (backup / recovery)
  - Supports a high-level access language (e.g. SQL)
    - Application describes database accesses using that language.
    - DBMS interprets statements of language to perform requested database access.
  - Provides transaction management to guarantee correct concurrent access to shared data

[Source: Disney, FL]

The Age of DBMS?

- How old do you think are database systems?



[Source: ACM Blog / Charles W. Bachmann]

The 50th Anniversary of DBMSs

- General Electric's "Integrated Data Store" (IDS)
  - designed and developed in 1962/1963
  - by a team around Charles W. Bachmann => Turing Award 1973
- The Problem
  - Every programmer had to implement their own data layouts and access methods
- The Solution
  - Centralise these functions in a system that allows to integrate programs that access shared data
- Full story at http://wp.sigmod.org

Charles W. Bachmann
[Source: Wikipedia]

This Idea is still Valid: File System vs. DBMS: Data Storage

File System:

```
class Customer {
  Integer cid;
  String  name;
  String  city;
  Float   rebate;

  boolean checkRebate() {…}
}

class Book { …   }
```

Data definitions repeated in each program…

Database System:

```
CREATE TABLE Customer (
  cid     INTEGER CHECK (cid>0),
  name    VARCHAR(20),
  city    VARCHAR(20),
  rebate  FLOAT
);

CREATE TABLE Book ( … );
```

Data definitions once in the central data dictionary of a DBMS.
=> Same for all applications

Central idea:
Database keeps metadata about its content and state

File vs. DBMS: Data Access

File System:

```
VAR o : order;
p : product;
fh1, fh2: file_handle;
fh1 = Open(order_file);
WHILE NOT EOF(fh1) DO
o := getnext(fh1);
IF o.month = 10 AND o.day >= 19 THEN
   fh2 = Open(product_file);
   WHILE NOT eof(fh2) DO
     p := getnext(fh2);
     IF p.pnr=o.pnr AND p.stock<100 THEN
       WriteCard(o.pnr, o.quantity);
     END;
    END;
    close(fh2):
END;
END; close(fh1);
```

You have to program it by hand, over and over and over …

Database System:

```
SELECT pnr, quantity
  FROM Orders o JOIN Products p
 WHERE o.month = 10 AND o.day >= 18
   AND p.stock < 100
```

Declarative queries
- easy to read and maintain
- evtl. usable from different applications
- efficient evaluation due to automatic optimization

Disadvantages of File Processing

- Program-Data Dependence
  - All programs contain full descriptions of each data file they use
- Data Redundancy (Duplication of data)
  - Different systems/programs have separate copies of the same data
  - No centralized control of data => **Integrity problems!**
- Limited Data Sharing
  - Required data in several, (potentially incompatible) files.
- Lengthy Development Times
  - Programmers must design their own file formats
  - For each new data access task, a new program is required.
- Excessive Program Maintenance
  - 80% of information systems budget

Problems with Data Dependency

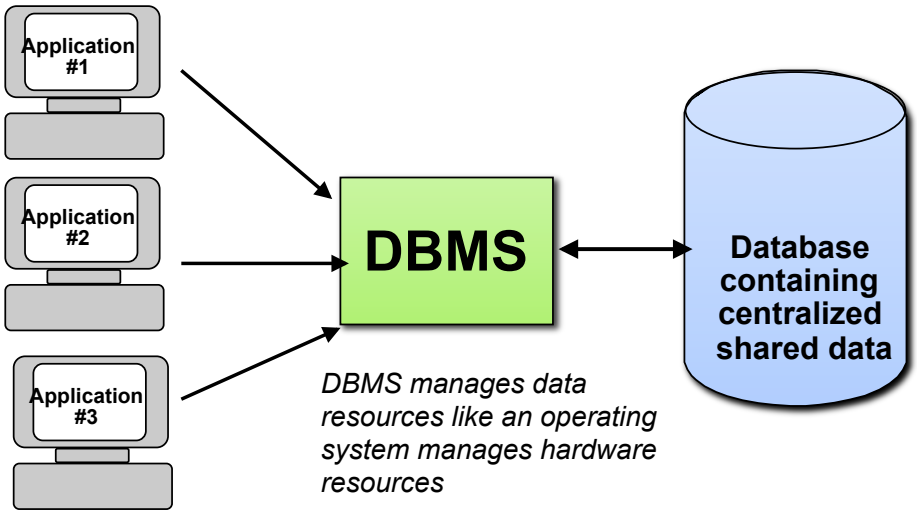
- Each application programmer must maintain their own data
- Each application program needs to include code for the metadata of each file
- Each application program must have its own processing routines for reading, inserting, updating and deleting data
- Lack of coordination and central control
- Non-standard file formats

Problems with Data Redundancy

- Waste of space to have duplicate data
- Causes more maintenance headaches
- The biggest Problem:
  - When data changes in one file, could cause inconsistencies
  - Compromises **data integrity**

Solution: The Database Approach

- Central repository of shared data
- Data is managed by a DBMS
- Stored in a standardized, convenient form



Own DB Programming Experience?

Central DBMS Services

- Data Independence
- Declarative Querying
- Transaction Management & Concurrency Control

Data versus Information

- **Data**: stored representations of raw facts or meaningful objects such as images and sounds that relate to people, objects, events, and other entities.
  - Structured: numbers, text, dates
  - Unstructured: images, video, documents

- **Information** refers to data that has been processed in some form (filtering, formatting, summarizing).
  It has been rendered appropriate for decision making or other kinds of use in particular contexts

- **Metadata**: Data that describes data

Main Advantages of Databases

- **Program-Data Independence**
  - Metadata stored in DBMS, so applications don't need to worry about data formats
  - Data queries/updates managed by DBMS so programs don't need to process data access routines
  - Results in:
    - Reduced application development time
    - Increased maintenance productivity
    - Efficient access

- **Minimal Data Redundancy**
  - Leads to increased data integrity/consistency

…and a couple more; cf. next slide / printout

Advantages of Databases (cont'd)

- Improved Data Sharing
  - Different users get different views of the data
  - Efficient concurrent access
- Enforcement of Standards
  - All data access is done in the same way
- Improved Data Quality
  - Integrity constraints, data validation rules
- Better Data Accessibility/ Responsiveness
  - Use of standard data query language (SQL)
- Security, Backup/Recovery, Concurrency
  - Disaster recovery is easier

Short History of Data Models

- **Early Database Applications:**
  - The Hierarchical and Network Models (CODASYL) were introduced in mid 1960's and dominated during the seventies. A bulk of the worldwide database processing still occurs using these models (legacy systems).
- **Relational Model based Systems:**
  - The model that was originally introduced in 1970 by **E. Codd** (another Turing award). It was researched and experimented with in IBM and universities. Relational DBMS products emerged in the 1980's and are now the norm since the 1990s.
- **Object-oriented Applications:**
  - OODBMSs were introduced in late 1980's and early 1990's to cater to the need of complex data processing in CAD and other applications. They sought to integrate the class definitions in an OO language (eg C++) with the way the data is stored persistently. Their use has not taken off much…
- **Data on the Web and E-commerce Applications:**
  - To express many different sorts of data that need to be exchanged between cooperating businesses, the recent standard is **XML (eXtended Markup Language)**. Main features: data is *semi-structured* and *self-describing.*

Relational Databases

- Stores data as rows with multiple attributes
- Rows of the same format ('type') form a table
- A relational database is a collection of such tables (which typically are related to each other by key attrbutes; more on that next week)
- Example:

Student				
sid	name	email	gender	address
5312666	Jones	ajon1121@cs	m	123 Main St
5366668	Smith	smith@mail	m	45 George
5309650	Jin	ojin4536@it	f	19 City Rd

Queries in a DBMS

- DBMS provides a specialized language for accessing data
  - **Query Language**
  - Can be further distinguished between
    - DML - Data Manipulation Language
    - DDL - Data Definition Language
    - DCL - Data Control Language
- Standard for relational DBMS: **SQL**
  - Based on formal query languages: Relational Algebra and Relational Calculus
- Queries are evaluated as efficient as possible
  - Huge influence by physical design

SQL Example

- The *working-horse* command: **SELECT – FROM – WHERE**
- retrieves data (rows) from one or more tables of a relational database that fulfil a search condition
- Example 1:

```
SELECT name, email
  FROM Student
 WHERE sid=5312666
```

- Example 2:

```
SELECT *
  FROM Student
```

- Example 3:

```
SELECT COUNT(*)
  FROM Student
 WHERE gender='f'
```


Declarative Queries: "What" not "How"

- It is convenient to indicate declaratively *what* information is needed, and leave it to the system to work out *how* to process through the data to extract what you need
  - Programming is hard, and choosing between different computations is hard
- Users should be offered a way to express their requests declaratively
  - A query language can be based on logic
  - Select…where…

We want to update data too…

=> Transaction Concept

- When an event in the real world changes the state of the enterprise, a ***transaction*** is executed to cause the corresponding change in the database state
  - With an on-line database, the event causes the transaction to be executed in real time
- A transaction is an application program with special properties - discussed later - to guarantee it maintains database correctness

Metadata

- A key idea in DBMSs is for the database itself to store descriptions of the format of the data
- This is stored in the “System Catalogue” or “Data Dictionary”
  - as part of the SQL standard: INFORMATION_SCHEMA
- Eg, you can find that each employee has
  - Identifier which is integer
  - Name which is a string of up to 30 characters
  - Address which is string of up to 60 characters
  - Salary which is integer
- This sort of information is often called *meta-data*

Metadata as Data

- The schema, and other meta-data, is essential to working with the data
  - Data is useless if one can't interpret it
  - Eg Uwe Roehm, 162,47,447
    - Is 162 weight in pounds, salary in $/hrs, height in cms, room number, or something else?
  - Over time, the people may leave who know what a value or string means
- Meta-data should be stored with the data it describes
  - And there should be some facilities for asking about and updating the meta-data

=> Another core idea of database systems

Example: Data

- Consider this list of facts:

Baker, Kenneth D.	324917628
Doyle, Joan E.	476193248
Finkle, Clive R.	548429344
Lewis, John C.	551742186
McFerran, Debra R.	409723145
Sisneros, Michael	392416582

Example: Data in Context

- Context helps users understand data.

Class Roster

Course: MGT 500 Business Policy Semester: Spring 200X

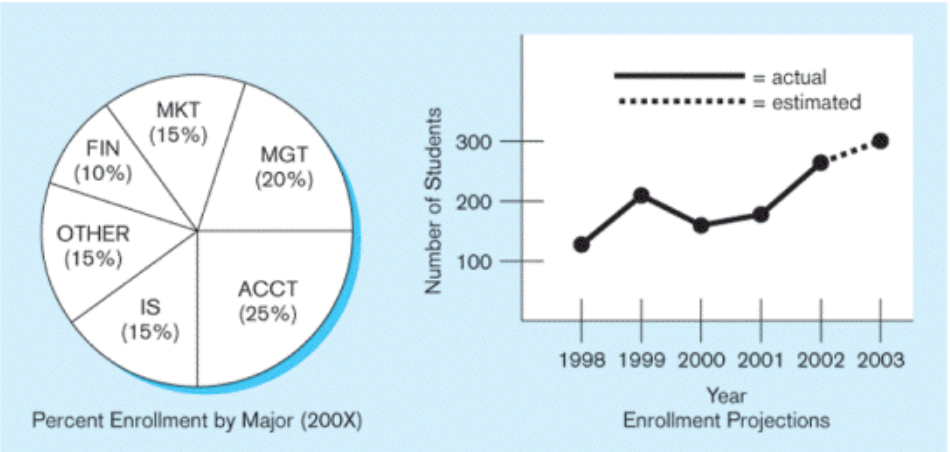
Section: 2

Name	ID	Major	GPA
Baker, Kenneth D.	324917628	MGT	2.9
Doyle, Joan E.	476193248	MKT	3.4
Finkle, Clive R.	548429344	PRM	2.8
Lewis, John C.	551742186	MGT	3.7
McFerran, Debra R.	409723145	IS	2.9
Sisneros, Michael	392416582	ACCT	3.3

But large volume of facts are difficult to interpret or make decisions based on...

Converting Data to Information

- Graphical displays turns data into useful information that managers can use for decision making and interpretation



Roles in Design, Implementation and Maintenance of a Information System

- System Analysts
  - specifies system using input from customer; provides complete description of functionality from customer's and user's point of view
  - Conceptual database design
- Database Designer
  - specifies structure of data that will be stored in database (logical & physical databse schemas)
- DB Application Programmer
  - implements application programs (transactions) that access data and support enterprise rules
- Database Administrator (DBA)
  - maintains database once system is operational: space allocation, performance optimization, database security, deals with failures and congestion
- End-Users
  - often unaware that they are dealing with data in a DBMS
- DBMS Vendor's Software Engineers

Summary

- DBMS used to maintain & query large datasets that are shared by many application programs/users
- Some powerful ideas:
  - Program-Data Independence
  - Controlled Data Redundancy
  - Declarative Queries
  - Transactions
- Databases are one of the broadest and most useful areas in CS and IS
  - Every 'knowledge worker' or scientists needs database know-how, as do all IT experts (application developers, software engineers, system analysts, …)

Next Week

- Conceptual Database Design using the
  - Entity Relationship Model
  - Database Design with UML
- Introduction of the DB Project Task
- Readings:
  - Kifer/Bernstein/Lewis book, Chapter 4
  - Ramakrishnan/Gehrke (Cow book), Chapter 2
  - Ullman/Widom, Chapter 4

References

- Kifer/Bernstein/Lewis (2nd edition)
  - Chapters 1.1-1.3, 2.1, 2.2, 3.1, 3.2
  - Missing: comparison with file-based info system
- Ramakrishnan/Gehrke (3rd edition)
  - Chapters 1.1-1.6, 1.9, 3.1
- Ullman/Widom (3rd edition)
  - Chapters 1.1, 2.1, 2.2
  - Missing: comparison with file-based info system, roles of workers
- Silberschatz/Korth/Sudarshan (5th edition)
  - Chapters 1.1-1.5, 1.12, 2.1
- Tony Hey et. Al. (Ed.): *The Fourth Paradigm: Data-Intensive Scientific Discovery*, Microsoft Research, 2009
  - http://research.microsoft.com/en-us/collaboration/fourthparadigm/