# Database Systems

Rolf Fagerberg

DM26, Fall 2005

# Course

Literature

Database Management Systems, Ramakrishnan and Gehrke.

Exam

Oral exam, 13-scale

Project

Database design and implementation project (DB: PostgreSQL).

Hours

Two lectures a week
Two exercise lessons a week (for each of two classes).

# Need for Databases

- Corporate data (payrolls, inventory, sales, customers, accounting, documents,…)

- Banking systems

- Stock exchanges

- Airline systems

- University data (students, grades)

- Hospitals

- Scientific data

- Website backends

- Personal data (CD collection, addresses,…)

# Database Desiderata

- Ease of use

- Flexible searching

- Efficiency

- Centralized storage, multi-user access

- Scalability (large amounts of data)

- Security and consistency:
  - Concurrency issues
  - System crashes
  - Access control
  - Integrity constraints on data

- Abstraction (implementation hiding)

- Good data modeling

# Current Systems

- DBMS = Database Management System

- Many vendors (Oracle, IBM DB2, MS SQL Server, MySQL, PostgreSQL,… ).

- All rather similar.

- Very big systems. Surprisingly easy to use.

Common features:

- Relational model

- SQL as query language

- Server-client architecture

# History

Early 60's

Integrated Data Store, General Electric. First general purpose DBMS. Network data model.

Late 60's

Information Management System, IBM (still in use!) Hierarchical data model.

1970

E. Codd: Relational data model, relational query languages.

# History (Cont.)

Mid 70s

    First relational DBMSs (IBM System R, UC Berkeley Ingres,. . . ).

80s

    Almost all commercial systems now based on relational model. SQL standardized.

90s

    Additional features added to DBMS: richer data types (large objects, OO-features), tools for manament, report generation, business analysis, data mining. Object-oriented DBMS appear, but not dominant.

# Why Study DBs?

- Very widely used.

- Part of many software solutions.

- DB expertise is a career asset.

- Interesting:
  - Mix of many different requirements
  - Mix of many different methodologies
  - Real world application

On the downside:

Real world applicability has top priority. Heuristics more than theory, few theorems.

# DB People

- End users (zillions)

- Application programmers (billions)

- DBMS administrators (millions)

- DBMS suppliers (thousands)

# DB development

- Requirement specification (not covered here)
- Data modeling (conceptual design)
- Database modeling (database design)
- Application programming (interface for end users)
- Database tuning

# Issues covered in course

Using DBs:

- E/R-model for data modeling

- Relational model (data model, relational query languages, normal forms)

- SQL syntax

- Application programming

- DB tuning

# Issues covered in course (Cont.)

DBMS implementation principles:

- Physical data storage

- Index structures

- Query parsing, optimization, execution

- Concurrency control (transactions)

- Crash recovery

Important background knowledge for efficient use of DBs, in particular for tuning.

# Note

- Database Systems Course $\neq$ SQL

- Database Systems Course $\neq$ Oracle, IBM, PostgreSQL,…

Database Systems Course = *principles* of use and implementation of (relational) database systems.