Databases

Rolf Fagerberg

DM505, Spring 2006, 4th Quarter

Course

Literature

Database Management Systems, Ramakrishnan and Gehrke, 3rd edition.

Exam

Oral exam, 13-scale.

Project

Database design and implementation project (DB: PostgreSQL).

Hours

Four/two lectures a week
Two/four exercise lessons a week.

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 purpuse 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. (Turing prize 1981).

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.
- Database modeling.
- 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
- Concurrency control (transactions)
- A little on query parsing, optimization, execution (if time).

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

Issues NOT covered in course

The following database subjects are also interesting, but we have no room for them:

- Details of query parsing, optimization, execution
- Crash recovery
- Database access control
- Distributed databases
- Object-oriented databases
- Deductive databases
- Data warehousing
- Spatial databases

Note

- Database Systems Course $\neq$ SQL
- Database Systems Course $\neq$ Oracle, IBM, MySQL, PostgreSQL,. . .

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