Lecture 1: Overview of CSCI 485

Shahram Ghandeharizadeh
Director of USC Database Lab
Computer Science Department
University of Southern California

Instructor Details
Dr. Shahram Ghandeharizadeh
Office: SAL 208
E-mail: shahram@usc.edu
Phone: 213-740-4781
Office Hours:
- Mon: 9 to 10 am, 2-3 pm
- Wed: 2 to 4 pm
Class URL: http://dblab.usc.edu/csci485

Logistics
- Required text book:
  - Please start to read Chapter 7 and Sections 2.1 to 2.3 for the next lecture
- Pre-req for the course:
  - CS201: Data Structures
  - Knowledge of an object-oriented programming language such as C++, Java, C#

TA Details
Jason Yap
Office: TBA
E-mail: jyap@usc.edu
Office Hours: TBA

Outline
- A brief history of relational DBMSs.
  - OSs 1960/70 1980+
  - Database DBMS
- Overview of CSCI 485.
- Grading: Assignments and projects.

Before Computers
Future Databases

BEFORE DBMS: 1960/70s

TERMINOLOGY
- Computers represent data as a sequence of zero and ones, termed bits:
  0101111110011010101010110000000000…..00000
- A byte is eight contiguous bits:
  0101111110011010101010110000000000…..00000
  1024 bytes is one Kilobyte.
  1024 KB is one Megabyte.
  1024 MB is one Gigabyte.
  1024 GB is one Terabyte.
  1024 TB is one Petabyte.
  1024 PB is one Exabyte.
  1024 EB is one Zetabyte.

AFTER DBMS: Late 1970+

DATABASE & DBMS
- Database: An integrated collection of data, usually stored on secondary storage, typically describing the activities of one or more related organizations.
- Database management system (DBMS): A collection of software/programs designed to assist in maintaining and utilizing large collections of data.

SQL
- SQL as a “What”-oriented language.
  - DBMS is responsible for “How” to retrieve data.
  - Example: Retrieve student-id and name of all those CS485 students with a GPA higher than 3.5:
    ```sql
    SELECT s.id, s.name
    FROM students s
    WHERE s.enrolled-in='CS485' and s.gpa > 3.5
    ```
WHY A DBMS?

1. Reduced application development time
2. Data independence: Application programs not dependent on data representation and storage details
3. Data sharing: data is better utilized (discovered and reused), redundancy of data is minimized
4. Data integrity and consistency: one may enforce consistency constraints on data, e.g., number of seats sold ≤ number of seats on the plane
5. Centralized control: DBA tunes the database to balance user's needs
6. Security: mechanisms to prevent unauthorized access. These mechanisms are based on content instead of file-oriented approach.
7. Concurrency control: avoids undesirable race conditions that arise with simultaneous access/updates to data
8. Crash recovery: ensures the integrity of data in the presence of failures

DBMS ARCHITECTURE

User 1

User n

DBMS

DB

Physical data

Conceptual schema

Data Models

Conceptual

Logical

Physical

Data Models (Example)

Build a database to keep the medical records of U.S. residents

Data Models

Conceptual

Logical

Physical

Challenges

Conceptual

Logical

Physical

Abstraction, Inheritance, Encapsulation

Reduction to tables with minimal data duplication, potential for data loss and update anomalies

Effective use of a DBMS, management of mismatch between tables and OO constructs, index structures, CC & Crash recovery, Optimization techniques

© 2013 Shahram Ghandeharizadeh
Challenges

- **Conceptual**
  - E-R data model
  - 5 Normal Forms

- **Logical**
  - Relational Data model: SQL, Relational Algebra

- **Physical**
  - Different kinds of storage, A DBMS architecture, Index structures, CC & Crash recovery, Optimization techniques, Techniques to build a DBMS

Relational DBMS

- **Why?**
  - Performance!
  - Reduced application development time
  - Use of SQL makes access to data more uniform:
    - Software modularity,
    - Extensibility

Challenge 2

- Two ways to teach this course:
  - How to **implement** a DBMS?
    - Protocols to realize atomic property of transactions
  - How to **use** a DBMS?
    - Setup a web server with a database and build a shopping bag

- Key difference: discussion at both the logical and physical levels
- Both require use of OO constructs

Grading

- Midterm 1: 35%
- Midterm 2: 35%
- Assignments: 30%