BBM371 - Data Management
Lecture 1: Course policies, Introduction to DBMS
11.10.2018

Today
- Introduction
  - About the class
  - Organization of this course
- Introduction to Database Management Systems (DBMS)

About the class

Reference Book - 1
Database Management Systems, Raghu Ramakrishnan, McGraw-Hill Education
Reference Book - 2
Database System Implementation, Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom

Communication
The course web page will be updated regularly throughout the semester with lecture notes, announcements and important dates.
http://web.cs.hacettepe.edu.tr/~bbm371

Course Work and Grading
- 1 midterm exam (25 points)
  - Closed book and notes
- Pop quizzes (25 points)
  - Closed book and notes
- Final exam (50 points)

Course Overview (Tentative)

<table>
<thead>
<tr>
<th>Date</th>
<th>Topic</th>
</tr>
</thead>
<tbody>
<tr>
<td>11/10/2018</td>
<td>Course Policies, Introduction to Data Management</td>
</tr>
<tr>
<td>18/10/2018</td>
<td>Storage Devices</td>
</tr>
<tr>
<td>25/10/2018</td>
<td>Basic File Concepts 1</td>
</tr>
<tr>
<td>01/11/2018</td>
<td>Basic File Concepts 2</td>
</tr>
<tr>
<td>08/11/2018</td>
<td>Index Types</td>
</tr>
<tr>
<td>15/11/2018</td>
<td>Hash Based File Organization</td>
</tr>
<tr>
<td>22/11/2018</td>
<td>Hash-based Indexing</td>
</tr>
<tr>
<td>06/12/2018</td>
<td>Treebased Indexing 1</td>
</tr>
<tr>
<td>13/12/2018</td>
<td>Midterm Exam</td>
</tr>
<tr>
<td>20/12/2018</td>
<td>Treebased Indexing (continued)</td>
</tr>
<tr>
<td>27/12/2018</td>
<td>External Sorting</td>
</tr>
<tr>
<td>05/01/2019</td>
<td>External Sorting (continued)</td>
</tr>
<tr>
<td>10/01/2019</td>
<td>Spatial Data Management</td>
</tr>
</tbody>
</table>
Introduction to Database Management Systems

What is Data?

► Data: Almost any kind of unorganized fact(s).

► Examples:
  » You throw a dice for a million times. Results are your data.
  » Anything you see in this classroom.
  » Music on a CD.
  » A computer file.

What is Signal?

► Signal is the encoding of the data that is needed for transmission.

  » Analog
  » Digital

What is Information?

► Data becomes information when it is processed and organized and thereby it becomes useful.
How to represent Data?

- Text
- Number
- Image
- Audio
- Video

How to represent Complex Data?

- Relational
- Graph
- Structured etc.

What is Management?

Management: The process of dealing with things (or people)!

- Initiation/Setting Objectives
- Planning
- Design and Implementation
- Execution
- Monitoring and Control

Finally – What is Data Management?

in this class...

- We will be interested in the following two concepts of data management:
  - Storage
  - Query Processing
What is a DBMS?

- A very large, integrated collection of data.
- Models real-world enterprise
- A Database Management System (DBMS) is a software package designed to store and manage databases
- Information about:
  - Entities such as students, faculty, courses
  - Relationships between entities for example a student is enrolled to a course

Data-Centric Applications

- Applications in which data plays an important role
  - Airline reservation systems
  - Banking applications
  - Hospital systems
  - University systems

- Data: aircrafts, flights, flight attendants, passengers, etc.
- Data: clients, deposits, withdraws, etc.
- Data: patients, physicians, diagnosis, prescriptions, etc.
- Data: students, teaching staff, courses, enrollments, etc.

History of DBMS

- Even from the early days of computers, data must be stored for applications
- Late 1960 IBM’s Information Management System (IMS) for airline reservations.
- 1970s Edgar Codd proposed a relational data model
- 1980s database query language SQL was standardized
- 1990s Data warehouses, consolidating data from multiple data stores for analysis
- 2000s Web applications
- Now – Even larger volumes of data NoSQL databases

Files vs. DBMS

- Imagine writing a program for a bank
  - Customers, Accounts, Money Transfers
  - More than 500 GB (does not fit in memory)
- Application must stage large datasets between main memory and secondary storage (500GB RAM is not still cheap!)
- Must protect data from inconsistency (update in ATM should be consistent with bank branch)
- Crash recovery
- Security and access control
- Concurrency (Transaction management)
Why Use a DBMS?

- Data independence and efficient access
- Reduced application and development time
- Data integrity and security
- Uniform data administration
- Concurrent access
- Recovery from crashes

Example of a Traditional Database Application

Suppose we are building a system to store the information about:

- students
- courses
- professors
- who takes what, who teaches what

Can we do it without a DBMS?

Sure we can! Start by storing the data in files:

students.txt courses.txt professors.txt

Now write C or Java programs to implement specific tasks

Doing it without a DBMS...

- Enroll “Mary Johnson” in “CSE444”:

  Write a C/Java program to do the following:

  Read 'students.txt'
  Read 'courses.txt'
  Find & update the record “Mary Johnson”
  Find & update the record “CSE444”
  Write “students.txt”
  Write “courses.txt”
Problems without an DBMS...

- System crashes:
  - What is the problem?
- Large data sets (say 50GB)
  - Why is this a problem?
- Simultaneous access by many users
  - Lock `students.txt` – what is the problem?

Why Study Databases?

- Shift from computation to information
  - Low-end users: Web Applications needs to organize information (a mess will not be effective)
  - High-end users: Scientific applications now have data management problems!
- Datasets increasing in diversity and volume
  - Digital libraries, interactive video, Human Genome project etc.
- DBMS encompasses most of CS
  - OS, languages, AI, multimedia etc.

DBMS

“Client-server”

Why Study Databases?

- Shift from computation to information
  - Low-end users: Web Applications needs to organize information (a mess will not be effective)
  - High-end users: Scientific applications now have data management problems!
- Datasets increasing in diversity and volume
  - Digital libraries, interactive video, Human Genome project etc.
- DBMS encompasses most of CS
  - OS, languages, AI, multimedia etc.

Data Models

- A **data model** is a collection of concepts for describing data. (high-level)
- A **schema** is a description of a particular collection of data, using the given data model
- The **relational model of data** is the most widely used model today.
  - **Main concept**: relation, basically a table with rows and columns
  - Every relation has a **schema**, which describes the columns, or fields.
  - Schema is defined by: name of schema, the name of each field (or attribute or column) and type of each field

  e.g.
  ```
  Student(sid: string, name: string, login: string, age: integer, gpa:real)
  ```

Database Management Systems 3ed, R. Ramakrishnan and J. Gehrke
Entity: Student

- Students (sid: string, name: string, login: string, age: integer, gpa: real)

<table>
<thead>
<tr>
<th>Sid</th>
<th>name</th>
<th>login</th>
<th>age</th>
<th>gpa</th>
</tr>
</thead>
<tbody>
<tr>
<td>53666</td>
<td>Jones</td>
<td>jones@cs</td>
<td>18</td>
<td>3.4</td>
</tr>
<tr>
<td>53688</td>
<td>Smith</td>
<td>smith@ee</td>
<td>18</td>
<td>3.2</td>
</tr>
<tr>
<td>53650</td>
<td>Smith</td>
<td>smith@math</td>
<td>19</td>
<td>3.8</td>
</tr>
</tbody>
</table>

Attributes (field or column)

Integrity Constraints: We can define the field sid to be unique or age to be larger than 0. Rules for records to satisfy

Levels of Abstraction

- Unlike programmers of early systems, programmer of relational system does not need to implement lower level details.
- Many views, single conceptual (logical) schema and physical schema.
- Views describe how users see the data.
- Conceptual schema defines logial structure
- Physical schema describes the files and indexes used.

Database Management Systems 3ed, R. Ramakrishnan and J. Gehrke
Physical View
- The DBMS must know
  - exact physical location
  - precise physical structure

Employee record
A.B.C. De Silva | 222, Galle Road, Colombo |
Name (20 characters) | Address (40 characters) |
650370690V | Senior Lecturer |
NID (10 char) | Designation (15 char) |

Conceptual Layer
- The conceptual model is a logical representation of the entire contents of the database.
- The conceptual model is made up of base tables.
- Base tables are “real” in that they contain physical records.

External View
- The user/application see
  - authorised data
  - own format

Lecturer
Name
A.B.C. De Silva
Department
Dept. of Computer Science
Designation
Senior Lecturer
Age 35

External View cont.
- External views allow to
  - hide unauthorised data
    - e.g. salary, dob
  - provide user view
    - e.g. view employee name, designation, department data taken from employee and department files
  - derive new attributes
    - e.g. age derived from dob
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 insulated 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 physical structure of data.

- One of the most important benefits of using a DBMS!

Concurrency Control

- Concurrent execution of user programs is essential for good DBMS performance.
  - Because disk accesses are frequent and relatively slow, it is important to keep the CPU humming by working on several user programs concurrently.
  - Interleaving actions of different user programs can lead to inconsistency.
  - DBMS ensures such problems don’t arise.

- Users can pretend they are using a single-user system.

Transaction Example 1

- Two users performing operations on a joint account at the same time.
  - If one reads before the other writes back, the first to write will be cancelled.
  - It will work ok if read and insert is atomic (not interrupted).
  - To make sure, we can lock the account.
Transaction Example 2

- A prepaid mobile phone user will transfer 10 credits to User 2.
- This operation needs two steps:
  - If trying to remove 10 credits from User 1 fails for some reason, we have added 10 credits to U2 out of the blue.
  - If we perform the operation in a transaction, we can roll-back the changes.

Structure of a DBMS

- A typical DBMS has a layered architecture.
- The figure does not show the concurrency control and recovery components.
- This is one of several possible architectures; each system has its own variations.
End of the first lecture...