Lecture 36
Data Management

During the last lecture ...
(Intelligent Systems)
• We looked at the distinguishing features of intelligent systems w.r.t. other software systems
• We looked at the role of intelligent systems in scientific, business, consumer and other applications
• We discussed several techniques for designing intelligent systems

(Artificial) Intelligent Systems:
• SW programs or SW/HW systems designed to perform complex tasks employing strategies that mimic some aspect of human thought

Not a Suitable Hammer for All Nails!
• if the nature of computations required in a task is not well understood
• or there are too many exceptions to the rules
• or known algorithms are too complex or inefficient
• then AI has the potential of offering an acceptable solution

Selected Applications:
• Games: Chess, SimCity
• Image recognition
• Medical diagnosis
• Robots
• Business intelligence

Neural Networks:
• Original inspiration was the human brain; emphasis now on usefulness as a computational tool.

Genetic Algorithms (1):
• Based on Darwin's evolutionary principle of ‘survival of the fittest’
• GAs require the ability to recognize a good solution, but not how to get to that solution

Rulebased Systems (1):
• Based on the principles of the logical reasoning ability of humans.

Fuzzy Logic (1):
• Based on the principles of the approximate reasoning faculty that humans use when faced with linguistic ambiguity

The Right Technique:
• Selection of the right AI technique requires intimate knowledge about the problem as well as the techniques under consideration
• Real problems may require a combination of techniques (AI and/or nonAI) for an optimal solution

Three exciting areas of AI applications Robotics:
• Automatic machines that perform various tasks that were previously done by humans

Autonomous Web Agents (1):
• Computer program that performs various actions continuously, autonomously on behalf of their principal!

Decision Support Systems:
• Interactive software designed to improve the decision-making capability of their users
Today's Goals: (Data Management)

- First of a two-lecture sequence
- Today we will become familiar with the issues and problems related to data-intensive computing
- We will find out about flat-files, the simplest databases
- Next time, in our 4th lecture on productivity software, we will discuss relational databases and implement a simple relational database
- Keeping track of a few dozen data items is straightforward
- However, dealing with situations that involve significant numbers of data items, requires more attention to the data handling process
- Dealing with millions - even billions - of inter-related data items requires even more careful thought

BholiBooks.com:

- Consider the situation of a large, online bookstore
- They have an inventory of millions of books, with new titles constantly arriving, and old ones being phased out on a regular basis
- The price for a book is not a static feature; it varies every once in a while
- Thousands of books are shipped each day, changing the inventory constantly
- Some are returned, again changing the inventory situation constantly
- The cost of each shipped order depends on:
  - Prices of individual books
  - Size of the order
  - Location of the customer
  - Mode of shipment
- For each order, the customer’s particulars — name, address, phone number, credit card number — are required
- Generally, that data is not deleted after the completion of the transaction; instead, it is kept for future reference
- All the transaction activity and the inventory changes result in:
  - Thousands of data items changing every day
  - Thousands of additional data items being added everyday
- Keeping track & taking care (i.e. management) of all that constantly changing and expanding data is not a trivial task and requires disciplined attention and actions for ensuring the smooth & profitable operation of the bookstore

Issues in Data Management:

- Data entry
- Data updates
- Data integrity
- Data security
- Data accessibility

Data Entry:

- New titles are added every day
- New customers are being added every day
- Some of the above may require manual entry of new data into the computer systems
- That new data needs to be added accurately
- That can be achieved, for one, by user-interfaces that prevent the input of invalid data

Data Updates:
• Old titles are deleted on a regular basis
• Inventory changes every instant
• Book prices change
• Shipping costs change
• Customers’ personal data change
• Various discount schemes are always commencing and concluding
• All those actions require updates to existing data
• Those changes need to be entered accurately
• That can also be achieved by user-interfaces that prevent the input of invalid data

**Data Security:**
• All the data that BholiBooks has in its computer systems is quite critical to its operation
• The security of the customers’ personal data is of utmost importance. Hackers are always looking for that type of data, especially for credit card numbers
• Enough leaks of that type, and customers will stop doing business with BholiBooks
• This problem can be managed by using appropriate security mechanisms that provide access to authorized persons/computers only
• Security can also be improved through:
  • Encryption
  • Private or virtual-private networks
  • Firewalls
  • Intrusion detectors
  • Virus detectors

**Data Integrity:**
• Integrity refers to maintaining the correctness and consistency of the data
  – Correctness: Free from errors
  – Consistency: No conflict among related data items
• Integrity can be compromised in many ways:
  – Typing errors
  – Transmission errors
  – Hardware malfunctions
  – Program bugs
  – Viruses
  – Fire, flood, etc.

**Ensuring Data Integrity:**
• Type Integrity is implemented by specifying the type of a data item:
  – Example: A credit card number consists of 12 digits. An update attempting to assign a value with more or fewer digits or one including a non-numeral should be rejected
• Limit Integrity is enforced by limiting the values of data items to specified ranges to prevent illegal values
  – Example: Age of person should not be negative
• Referential Integrity requires that an item referenced by the data for some other item must itself exist in the database
  – Example: If an airline reservation is requested for a particular flight, then the corresponding flight number must actually exist
  – Physical Integrity is ensured through hardware redundancy, backups, etc

**Data Accessibility:**
• If the transaction and inventory data is placed in a disorganized fashion on a hard disk, it becomes very difficult to later search for a stored data item
• What is required is that:
  – Data be stored in an organized manner
  – Additional info about the data be stored
  so that the data access times are minimized
• What if two customers check on the availability of a certain title simultaneously?
• On seeing its availability, they both order the title – for which, unfortunately, only a single copy is available
• Same is the case when two airline customers try booking the only available seat
• A solution to this concurrency control problem: Lock access to data while someone is using it
• We can write our own SW that can take care of all the issues that we just discussed
  OR
• We can save ourselves lots of time, cost, and effort by buying ourselves a Database Management System (DBMS) that takes care of most, if not all, of the issues

**DBMS:**
• DBMSes are popularly, but incorrectly, also known as ‘Databases’
• A DBMS is the SW system that operates a database, and is not the database itself
• Some people even consider the database to be a component of the DBMS, and not an entity outside the DBMS

**Database:**
• A collection of data organized in such a fashion that the computer can quickly search for a desired data item
• All data items in it are generally related to each other and share a single domain
• They allow for easy manipulation of the data
• They are designed for easy modification & reorganization of the information they contain
• They generally consist of a collection of interrelated computer files

**Example: VU Student Database:**
- Student's name
- Student's photograph
- Father’s name
- Phone number
- Street address
- eMail address
- Courses being taken
- Courses already taken & grades
- Pre-VU educational record

**Example: BholiBooks’ Customer DB:**
- Name, address, phone & fax, eMail
- Credit card type, number, expiration date
- Shipping preference
- Books on order
- All books that were ever shipped to the customer
- Book preference

**Example: BholiBooks’ Inventory DB:**
- Book title, author, publisher, binding, date of publication, price
- Book summary, table of contents
- Customers’, editors’, newspaper reviews
- Number in stock
- Number on order
- Special offer details

**OS Independence:**
- DBMS stores data in a database, which is a collection of interrelated files
- Storage of files on the computer is managed by the computer OS’s file system
- Intimate knowledge of the OS & its file system is required to provide rapid access to the data
- The DBMS takes care of those details
- It hides the actual storage details of data files from the user
- It provides an OS-independent view of the data to the user, making data manipulation and management much more convenient

**What can be stored in a database?**
- In the old days, databases were limited to numbers, Booleans, and text
- These days, anything goes
- As long as it is digital data, it can be stored:
  - Numbers, Booleans, text
  - Sounds
  - Images
  - Video

**In the very, very old days …:**
- Even large amounts of data was stored in text files, known as flat-file databases
- All related info was stored in a single long, tab- or comma-delimited text file
• Each group of info – called a record - in that file was separated by a special character; vertical bar ‘|’ was a popular option
• Each record consisted of a group of fields, each field containing some distinct data item
The Trouble with Flat-File Databases:

- The text file format makes it hard to search for specific information or to create reports that include only certain fields from each record.
- Reason: One has to search sequentially through the entire file to gather desired info, such as ‘all books by a certain author’.
- However, for small sets of data – say, consisting of several tens of kB – they can provide reasonable performance.

Consider this tabular approach ...

(same records, same fields, but in a different format)

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Publisher</th>
<th>Price</th>
<th>InStock</th>
</tr>
</thead>
<tbody>
<tr>
<td>Good Bye Mr. Bhola</td>
<td>Altaf Khan</td>
<td>BholiBooks</td>
<td>1000</td>
<td>Y</td>
</tr>
<tr>
<td>The Terrible Twins</td>
<td>Bhola Champion</td>
<td>BholiBooks</td>
<td>199</td>
<td>Y</td>
</tr>
<tr>
<td>Calculus &amp; Analytical Geometry</td>
<td>Smith Sahib</td>
<td>Good Publishers</td>
<td>325</td>
<td>N</td>
</tr>
<tr>
<td>Accounting Secrets</td>
<td>Zamin Geoffry</td>
<td>Sung-e-Kilometer Publishers</td>
<td>29</td>
<td>Y</td>
</tr>
</tbody>
</table>

Tabular Storage: Features & Possibilities:

- Similar items of data form a column.
- Fields placed in a particular row – same as a flat-file record – are strongly interrelated.
- One can sort the table w.r.t. any column.
- That makes searching – e.g., for all the books written by a certain author – straightforward.

Tabular Storage: Features & Possibilities:
5. Similarly, searching for the 10 cheapest/most expensive books can be easily accomplished through a sort
6. Effort required for adding a new field to all the records of a flat-file is much greater than adding a new column to the table

**CONCLUSION:** Tabular storage is better than flat-file storage
We will continue on this theme next time

**Today's Summary:** (Data Management)
- First of a two-lecture sequence
- Today we became familiar with the issues and problems related to data-intensive computing
- We also found out about flat-file and tabular storage

**Next Lecture:** (Database SW)
- Next time, in our 4th lecture on productivity SW, we will continue our discussion on data management
- We will find out about relational databases
- We will also implement a simple relational database