Database trends: XML data storage
DRC Students

- If any student in the class requires a special accommodation for test taking or other assignment, please contact me
  - In person, or via email, ejw@cs.ucsc.edu
  - If you don’t contact me, I will not know you need this accommodation
  - The DRC office no longer sends notifications out about this
Midterm #1

- Wednesday, April 27, in class
- A review session will be held Tuesday 3-5pm, Engineering 2, room 215 (2nd floor)
- Test will be mostly short answer type questions, and questions similar to homework #2
- Closed book, closed note
- Will cover all material in class, up to and including today’s lecture

- Reminder: lecture notes available from class website:
  - [http://www.soe.ucsc.edu/classes/cmps010/Spring11/](http://www.soe.ucsc.edu/classes/cmps010/Spring11/)
- Homework #2 solutions also are on website
  - Go “Assignments” -> “Homework #2” -> “Solutions”
Potential Exam Topics

- As Univ. of California students, you are expected to be able to assess complex material and make judgments concerning its relative importance.
- That said, it can be helpful to have some input from the Professor to help focus studying activity.
- The following are questions/topics that are likely, but not guaranteed to appear on the exam.
- Anything covered in class or in the assigned readings may appear, even if not explicitly mentioned today.
Potential exam topics/questions

- What is computer science?
  - what can be accomplished using computers, and
  - how to construct software to do these things
- What are the negative qualities of having humans perform complex computations?
- What two computing machines did Charles Babbage develop, and during what time period?
- What was the key contribution of the analytical engine?
  - Abstracting the instructions for a computation away from the physical device that realizes them
- Who was Ada Lovelace? What “first” is she credited with?
- What was the crisis facing the census of 1890?
- What did Herman Hollerith invent? How did this solve the census crisis?
- How did typical punched card computation work? What additional capabilities were required to perform scientific and engineering computation? How did this lead to the development of the card programmable calculator?
Potential exam topics/questions

- What was ENIAC? Where was it developed? Who were the two main inventors?
- How long did it take to set up the ENIAC for a problem?
- What are the key elements of the von Neumann architecture?
  - Computer includes an instruction set
  - Computer memory can include either data or program instructions
  - Computer fetches an instruction from memory, decodes & executes it, then fetches the instruction in the next memory location, etc.
- What is the fetch-execute cycle?
- What was UNIVAC? What “first” is it credited with?
- Know the relative chronology of ENIAC and UNIVAC
Potential exam topics/questions

- Know the various different uses of computers (notes from Lecture 3, April 1)
  - Given a description of a particular use of computing, be able to describe which use area it belongs to
  - Be able to compare/contrast the different areas
- Know the process for converting the real world into data
  - Real world \(\rightarrow\) model \(\rightarrow\) data
- Be able to describe the process of abstraction
  - **Focus** on aspects of the real world that are important to the problem. Add those elements to your model
  - **Omit** elements of the real world that aren’t relevant
- Be able to describe why the same physical world situation/scenario can be modeled in different ways
  - Different problems lead to different models of the same situation
Potential exam topics/questions

- Know the difference between a floating point number and an integer
- Know the difference between a character and a string
- Know what values a boolean can take
- Know the difference between an array, list, stack, and queue
  - All of these can represent a set
  - But, have different pros/cons
- Be able to perform operations on these basic data types (similar to the second homework assignment)
- Know the difference between a graph and a tree
  - Know what each are good at modeling
- Be able to perform data modeling scenarios, like in the second homework assignment
Potential exam topics/questions

- In class modeling (object-modeling), know that inheritance models the “isa” relationship (also called parent-child relationship)
- Know that children inherit data fields from their parent
- What is a Turing Machine? Who invented it? Did he ever build a physical version?
- What are the components of the Turing Machine
- What was the goal of the Principia Mathematica?
- What was the relationship of the Decidability (Entscheidungsproblem) to the goals of the Principia Mathematica? How does the computability of numbers relate?
- What was the relationship of the Decidability Problem and the Turing Machine?
- Today, what is the utility of the Turing Machine?
  - A general model of computation, permits theoretical examination of what is, and is not, computable
- Post’s Correspondence Problem is an example of what?
  - It is an uncomputable problem in the general case.
Potential exam topics/questions

- What is an algorithm?
- Know the key building blocks of algorithms
- What is a condition?
- How does an if .. then .. else block work?
- What is iteration?
- What is recursion?
- What is a qubit?
- Can quantum computers solve any problems that a Turing machine cannot solve?
- What is the main advantage of quantum computing?
  - Can solve some problems much faster than traditional computers.
- What did Rey Johnson invent that was a “game changer” for storage and processing of data?
  - What was the game-changing aspect? Permitted **random access** to stored data.
Potential exam topics/questions

- How did database management systems differ from sequential data processing applications?
  - Central store of data (the database, stored on disk)
  - Many applications interact with the same data
  - Database is now at the center, and applications are around it
  - In sequence data processing, application is at center, data flows through it

- Who developed the relational data model?

- What are key elements of relational data model:
  - Data are stored in tables (relationships)
  - Separation of logical content of data from physical representation
  - Database is responsible for executing a query

- What is SQL? What do you do with it?
Potential exam topics/questions

- How does structured data (e.g., tables in a database) differ from semi-structured data (e.g., XML data)?
- What services does a database provide?
- What are some typical functions inside an organization that use databases?
  - Payroll, inventory management, accounting...
- Know the organization of a database
  - Database contains tables, tables contain rows of data, each table has an associated schema
- What is a database schema?
- How is a schema similar to a class model?
- What is a unique identifier? What is the “unique” property it holds?
- Know the functions of the 4 main parts of a SQL query
  - For example, what is the difference between the WHERE clause and the ORDER BY clause?
- What is XML?
Semi-structured data

- What if you were given the task of representing the contents of a book in data?
  - A book has a title, author, copyright data
  - It contains many chapters
  - Each chapter has a title, may contain sub-headings, and contains many paragraphs
  - A paragraph might contain a bulleted list, a numbered list, an equation, or just a sequence of text
  - The text will have some areas emphasized, and others bolded
- This description shows that a book does have some structure
  - Book contains chapters contain text, etc.
- But, the structure is somewhat **variable**
  - How many chapters? This varies
  - How much text per chapter? Varies
  - How many bulleted lists per chapter? Varies
- As a result, data like the representation of a book is called **semi-structured**
Semi-structured data and databases

- Database tables do not perform well when representing semi-structured data

- Why?
  - The fact that some data items are often not present, or present in varying amounts, means that a database table will have many blank cells

- Better to use a representation that can handle missing elements, or elements that can be present zero, one, or multiple times

- XML – extensible markup language
  - Most widely used standard for representing semi-structured data
The underlying model is a **tree**

The tree is composed of **elements**

An element contains:
- (optional) **Attributes**
- A sequence of characters (character data)
- Other elements

```
contains -- an element contains other elements

element

contains
character data

contains
attributes
```
Example: a book in XML

- A book is modeled as a **book** element
  - Book element contains a series of **chapter** elements
    - Each chapter element contains a sequence of text, bulleted_list, numbered_list, and equation elements
    - Text elements have character data
    - Bulleted and numbered lists have **li** elements (list item)
XML in text

- An XML element has a:
  - Begin tag
  - End tag
  - A **begin tag** has the name of the element in between < > brackets
    - `<element_name>`
  - An **end tag** has the same name between </ and > brackets
    - `</element_name>`

- In between the start and end tag for an element you can have:
  - Characters
  - Other elements

- Attributes come after the element name in the begin tag
  - `<element_name attribute_1=value attribute_2=another_value …>`
Once upon a time, there was an intelligent computer scientist named Grace Hopper. She worked on the following projects:

- UNIVAC programming
- COBOL language

Over time, she was promoted to Rear Admiral, and had a destroyer named after her. You never know where computer science will take you!
Textual representation of XML

- XML is represented using a text-based notation

Benefits:
- Is somewhat human readable
- Is easy to exchange between different computers
- Is easy to extend over time
- Is relatively robust when there are errors

Drawbacks
- Is space inefficient as compared with other ways of representing the same data
- Is better for representing text data than numeric data
- Represents lists and trees well, but does not represent graphs very well.
XML data storage

- Over the past 10 years, several companies have developed database systems that work with XML data
  - These are called XML databases
- Two main types
  - XML-enabled
    - Map XML to a relational database schema inside the database
  - Native XML
    - Internal model of the database is XML, and XML data models are used throughout the database
- All provide strong services for storing, updating, and searching data stored as XML
Rich set of standards around XML

- There is now a rich set of standards supporting XML
  - XPath
    - For identifying parts of XML documents
  - XQuery
    - For searching through collections of XML documents
  - XSchema
    - For describing the kind of data held in each element
  - XSLT
    - A language for transforming one XML document into another

- ... and many other
XML as building block

- XML is increasingly used as a building block for creating new standards
  - It takes care of the problem of representing data in an extensible, machine-readable way that can be transported across machines
  - Since it supports UNICODE, also handles internationalization well
- Today, many new network protocols use XML to represent data sent across the wire
- Is a core technology used in many networking and **cloud computing** technologies
  - For example, a Zynga game like CityVille exchanges data in XML format across the network