Acknowledgement:
Most slides for this course have been adapted from slides made available by Addison Wesley to accompany Elmasri and Navathe’s textbook.
LECTURE OUTLINE

- Introduction
- An Example
- Characteristics of the Database Approach
- Actors on the Scene
- Workers behind the Scene
- When Not to Use a DBMS
WEALTH OF DATA

- Traditional database applications
  - Store numeric short textual information
  - Typically for managing enterprises
- Text and multimedia databases
  - Store documents, digital images, audio, and video streams
- Geographic information systems (GIS)
  - Store maps, weather data, and satellite images
  - For route-finding, agriculture, and natural resource management
- Data warehouses and online analytical processing (OLAP) systems
  - Store historical business information
  - For business analytics and decision support
- Real-time and active database technology
  - Store process models, constraints, and key performance indicators
  - Control industrial and manufacturing processes
TERMINOLOGY

- **Database**
  - Collection of related data (logically coherent)
  - Known facts that can be recorded and that have implicit meaning
  - Represents some aspect(s) of the real world (miniworld)
  - Built for a specific purpose

- **Examples of large databases**
  - Amazon.com, Canadian Census, The Bay’s product inventory, data collection underlying Quest
Database management system (DBMS)
- Collection of programs
- Enables users to create and maintain a database
- Allows multiple users and programs to access and manipulate the database concurrently
- Provides protection against unauthorized access and manipulation
- Provides means to evolve database and program behaviour as requirements change over time

Examples of database management systems
- IBM’s DB2, Microsoft’s Access and SQL Server, Oracle, MySQL, SAP’s SQL Anywhere
TERMINOLOGY (CONT'ED.)

- **Defining** a database
  - Specifying the data types, structures, and constraints of the data to be stored
  - Uses a *Data Definition Language*

- **Meta-data**
  - Database definition or descriptive information
  - Stored by the DBMS in the form of a *database catalog* or *data dictionary*

- Phases for designing a database:
  - **Requirements specification and analysis**
  - **Conceptual design**
    - e.g., using the *Entity-Relationship model*
  - **Logical design**
    - e.g., using the *relational model*
  - **Physical design**
TERMINOLOGY (CONT'D.)

- **Populating** a database
  - Inserting data to reflect the miniworld

- **Query**
  - Interaction causing some data to be retrieved
  - uses a *Query Language*

- **Manipulating** a database
  - Querying and updating the database to understand/reflect miniworld
  - Generating reports
  - Uses a *Data Manipulation Language*

- **Application program**
  - Accesses database by sending queries and updates to DBMS

- **Transaction**
  - An atomic unit of queries and updates that must be executed as a whole
    - e.g., buying a product, transferring funds, switching co-op streams
Figure 1.1
A simplified database system environment.
Movie database
• Information concerning movies, actors, awards

Data records
• Film
• Person
• Role
• Honours

Define structure of each type of record by specifying data elements to include and data type for each element
• String (sequence of alphabetic characters)
• Numeric (integer or real)
• Date (year or year-month-day)
• Monetary amount
• etc.
Populate MOVIES database
• Store data to represent each film, actor, director, award, role

<table>
<thead>
<tr>
<th>Film</th>
<th>title</th>
<th>genre</th>
<th>year</th>
<th>director</th>
<th>runtime</th>
<th>budget</th>
<th>gross</th>
</tr>
</thead>
<tbody>
<tr>
<td>The Company Men</td>
<td>drama</td>
<td>2010</td>
<td>John Wells</td>
<td>104</td>
<td>15,000,000</td>
<td>4,439,063</td>
<td></td>
</tr>
<tr>
<td>Lincoln</td>
<td>biography</td>
<td>2012</td>
<td>Steven Spielberg</td>
<td>150</td>
<td>65,000,000</td>
<td>181,408,467</td>
<td></td>
</tr>
<tr>
<td>War Horse</td>
<td>drama</td>
<td>2011</td>
<td>Steven Spielberg</td>
<td>146</td>
<td>66,000,000</td>
<td>79,883,359</td>
<td></td>
</tr>
<tr>
<td>Argo</td>
<td>drama</td>
<td>2012</td>
<td>Ben Affleck</td>
<td>120</td>
<td>44,500,000</td>
<td>135,178,251</td>
<td></td>
</tr>
<tr>
<td>Fire Sale</td>
<td>comedy</td>
<td>1977</td>
<td>Alan Arkin</td>
<td>88</td>
<td>1,500,000</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Person</th>
<th>name</th>
<th>birth</th>
<th>city</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ben Affleck</td>
<td>1972</td>
<td>Berkeley</td>
<td></td>
</tr>
<tr>
<td>Alan Arkin</td>
<td>1934</td>
<td>New York</td>
<td></td>
</tr>
<tr>
<td>Tommy Lee Jones</td>
<td>1946</td>
<td>San Saba</td>
<td></td>
</tr>
<tr>
<td>John Wells</td>
<td>1957</td>
<td>Alexandria</td>
<td></td>
</tr>
<tr>
<td>Steven Spielberg</td>
<td>1946</td>
<td>Cincinnati</td>
<td></td>
</tr>
<tr>
<td>Daniel Day-Lewis</td>
<td>1957</td>
<td>Greenwich</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Honours</th>
<th>movie</th>
<th>award</th>
<th>category</th>
<th>winner</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lincoln</td>
<td>Critic's Choice</td>
<td>actor</td>
<td>Daniel Day-Lewis</td>
<td></td>
</tr>
<tr>
<td>Argo</td>
<td>Critic's Choice</td>
<td>director</td>
<td>Ben Affleck</td>
<td></td>
</tr>
<tr>
<td>Lincoln</td>
<td>Screen Actors Guild</td>
<td>supporting actor</td>
<td>Tommy Lee Jones</td>
<td></td>
</tr>
<tr>
<td>Lincoln</td>
<td>Screen Actors Guild</td>
<td>actor</td>
<td>Daniel Day-Lewis</td>
<td></td>
</tr>
<tr>
<td>Lincoln</td>
<td>Critic's Choice</td>
<td>screenplay</td>
<td>Tony Kushner</td>
<td></td>
</tr>
<tr>
<td>Argo</td>
<td>Screen Actors Guild</td>
<td>cast</td>
<td>Argo</td>
<td></td>
</tr>
<tr>
<td>War Horse</td>
<td>BMI Flim</td>
<td>music</td>
<td>John Williams</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Role</th>
<th>actor</th>
<th>movie</th>
<th>persona</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ben Affleck</td>
<td>Argo</td>
<td>Tony Mendez</td>
<td></td>
</tr>
<tr>
<td>Alan Arkin</td>
<td>Argo</td>
<td>Lester Siegel</td>
<td></td>
</tr>
<tr>
<td>Ben Affleck</td>
<td>The Company Men</td>
<td>Bobby Walker</td>
<td></td>
</tr>
<tr>
<td>Tommy Lee Jones</td>
<td>The Company Men</td>
<td>Gene McClary</td>
<td></td>
</tr>
<tr>
<td>Tommy Lee Jones</td>
<td>Lincoln</td>
<td>Thaddeus Stevens</td>
<td></td>
</tr>
<tr>
<td>Alan Arkin</td>
<td>Fire Sale</td>
<td>Ezra Fikus</td>
<td></td>
</tr>
<tr>
<td>Daniel Day-Lewis</td>
<td>Lincoln</td>
<td>Abraham Lincoln</td>
<td></td>
</tr>
</tbody>
</table>
Manipulation involves querying and updating

Examples of queries:
- List the cast of characters for *Lincoln*.
- Who directed a *drama* in 2012?
- Who directed a film in which he or she also played a role?
- What awards were won by *War Horse*?

Examples of updates:
- Record that *Argo* won a Golden Globe award for best picture.
- Add another $395,533 to the gross earnings for *Lincoln*.
- Change the birthplace for *Daniel Day-Lewis* to *London*.
- Delete *Fire Sale* from the database.
TERMINOLOGY (CONT'D.)

- **Reorganizing** a database
  - Changes the metadata rather than the data
  - More drastic than data updates
    - May require massive changes to the data
    - May require changes to some application programs
  - Uses the *Data Definition Language again*

- **Examples:**
  - Rename *gross* to be *domestic earnings* and add a new column for *foreign earnings*.
  - Move *director* from FILM to a separate relation DIRECTOR with columns for *person* and *movie*
  - Change *birth* from *yyyy* to *yyyy/mm/dd*
  - Split name in PERSON to separate *surname* from *given names*.
  - Include column *movieID* in FILM (to accommodate remakes and other duplications of film title) and update other relations accordingly.
PRE-DBMS DATABASES

- Used traditional file processing
  - Each user defines and implements the files needed for a specific software application
  - As the application base grows
    - many shared files
    - a multitude of file structures
    - a need to exchange data among applications
- Eventually recognized that data is a critical corporate asset (along with capital and personnel)
DATABASE APPROACH

- Single repository maintains data that is defined once and then accessed by various users
- Addresses a variety of problems
  - redundancy: multiple copies
  - inconsistency: independent updates
  - inaccuracy: concurrent updates
  - incompatibility: multiple formats
  - insecurity: proliferation
  - inauditability: poor chain of responsibility
  - inflexibility: changes are difficult to apply
CHARACTERISTICS OF THE DATABASE APPROACH

- Programs isolated from data through abstraction
  - Does not expose details of how (or where) data is stored or how operations are implemented
  - Data sharing through multiple views
- Multiuser transaction processing
  - Encapsulates sequence of operations to behave atomically
- Data is self-defining
  - Database system contains complete definition of structure and constraints as meta-data
  - Database catalog used by:
    - DBMS software
    - Database users who need information about database structure
### DATABASE CATALOG

#### RELATIONS

<table>
<thead>
<tr>
<th>Relation_name</th>
<th>No_of_columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>STUDENT</td>
<td>4</td>
</tr>
<tr>
<td>COURSE</td>
<td>4</td>
</tr>
<tr>
<td>SECTION</td>
<td>5</td>
</tr>
<tr>
<td>GRADE_REPORT</td>
<td>3</td>
</tr>
<tr>
<td>PREREQUISITE</td>
<td>2</td>
</tr>
</tbody>
</table>

#### COLUMNS

<table>
<thead>
<tr>
<th>Column_name</th>
<th>Data_type</th>
<th>Belongs_to_relation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Character (30)</td>
<td>STUDENT</td>
</tr>
<tr>
<td>Student_number</td>
<td>Character (4)</td>
<td>STUDENT</td>
</tr>
<tr>
<td>Class</td>
<td>Integer (1)</td>
<td>STUDENT</td>
</tr>
<tr>
<td>Major</td>
<td>Major_type</td>
<td>STUDENT</td>
</tr>
<tr>
<td>Course_name</td>
<td>Character (10)</td>
<td>COURSE</td>
</tr>
<tr>
<td>Course_number</td>
<td>XXXXNNNN</td>
<td>COURSE</td>
</tr>
<tr>
<td>Prerequisite_number</td>
<td>XXXXNNNN</td>
<td>PREREQUISITE</td>
</tr>
</tbody>
</table>

*Note: Major_type is defined as an enumerated type with all known majors. XXXXNNNNN is used to define a type with four alpha characters followed by four digits.*
ACTORS ON THE SCENE

- **Database administrator (DBA)** responsible for:
  - Authorizing access to the database
  - Coordinating and monitoring its use
  - Acquiring software and hardware resources
  - Tuning the DBMS for best performance

- **Database designer** responsible for:
  - Identifying the data to be stored
  - Choosing appropriate structures to represent and store this data
ACTORS ON THE SCENE (CONT'D.)

- **End users**
  - Those whose jobs require access to the database
    - *Naive or parametric end users*
      - canned queries and updates
    - *Casual end users*
      - occasional, special-purpose access
    - *Sophisticated end users*
      - deep knowledge of database design and DBMS facilities
  - **Standalone users**
    - users of personal databases

- **System analysts**
  - Determine requirements of end users

- **Application programmers**
  - Implement complex specifications (*business logic*) as programs
WORKERS BEHIND THE SCENE

- **DBMS system designers and implementers**
  - Design and implement the DBMS modules and interfaces as a software package

- **Tool developers**
  - Design and implement tools

- **Operators and maintenance personnel**
  - Responsible for running and maintenance of hardware and software environment for database system
WHEN NOT TO USE A DBMS

- More desirable to use regular files for:
  - Simple, well-defined applications with no expected changes at all
  - Small variety of data and/or small amount of data
  - Stringent, real-time requirements that cannot afford DBMS overhead
  - Only single (personal) access to data

- Unlikely that any of these apply to corporate data management.
  - In fact, corporations often maintain many databases across many database systems.
LECTURE SUMMARY

- **Database**
  - Collection of related data (recorded facts)
- **DBMS**
  - Generalized software package for implementing and maintaining a computerized database
  - Provides many services to manage data resources
- Several categories of database users