Measuring Data: Bits and Bytes

- Bit = 0 or 1

- One byte is 8 bits.
  - example: 01101100

- Other common units:
  
<table>
<thead>
<tr>
<th>name</th>
<th>approximate size</th>
<th>exact size</th>
</tr>
</thead>
<tbody>
<tr>
<td>kilobyte (KB)</td>
<td>1000 bytes</td>
<td>$2^{10} = 1024$ bytes</td>
</tr>
<tr>
<td>megabyte (MB)</td>
<td>1 million bytes</td>
<td>$2^{20}$ bytes</td>
</tr>
<tr>
<td>gigabyte (GB)</td>
<td>1 billion bytes</td>
<td>$2^{30}$ bytes</td>
</tr>
</tbody>
</table>

- Scientists are starting to generate data collections measured in:
  - terabytes: $2^{40}$ or approx. $10^{12}$ bytes
  - petabytes: $2^{50}$ or approx. $10^{15}$ bytes
    - equivalent to the text in one billion books!
Storing Data: Memory

- Used to store programs and other data that are currently in use.
- Values stored in memory are read into the CPU to be operated on.
- The results of operations performed by the CPU can be written back to memory.
- Advantage of memory: short access times
  - can read from/write to memory in nanoseconds (10⁻⁹ sec)
- Disadvantages:
  - relatively expensive
  - contents are lost when the power is turned off

Storing Data: Secondary Storage

- Used to store programs and other data for later use.
  - examples: hard disks, floppy disks, CD/DVD drives, tape drives
- Advantages of hard disks:
  - relatively inexpensive
  - contents are retained when the power goes off
- Disadvantage: long access times
  - roughly 10 ms (10⁻³ sec)
  - in that time, a modern CPU can perform millions of operations!
  - it's important to minimize the number of times that the disk is accessed
What is a Database?

• A collection of data
  • it does not need to be on a computer.
    • example: the paper card catalogs that libraries maintained

• A given database may be divided into subcollections (tables)
  • should be related in some way
  • example: a university database
    • possible subcollections?

Database vs. Database Management System

• A database is a collection of data. It is not a piece of software.

• A database management system (DBMS) is the software that manages one or more databases.
Key Functions of a DBMS

1. efficient storage
2. providing a logical view of data
3. query processing
4. transaction management

• Let's look at each of them in turn.

1. Efficient Storage

• Recall: accessing the disk is very inefficient.

• A DBMS organizes the data on disk in ways that allow it to reduce the number of disk accesses.

• Example:
  • a database with 100,000 records
  • a given record is between 64-256 bytes long

• An inefficient approach:
  • give each record 256 bytes, even though it may not need it
  • scan through the database to find a record
  • may require thousands of disk reads!
1. Efficient Storage (cont.)

- A more efficient approach:
  - give each record only as much space as it needs
  - use a special *index structure*
    - allows the DBMS to locate a particular record *without* looking at every record
  - can dramatically reduce the number of disk accesses
    - as few as 1-3!

- A DBMS can also spread a database over multiple disks.
  - allows for larger collections of data
  - the disks can be accessed in parallel, which speeds things up
  - another advantage of using multiple disks?

2. Providing a Logical Representation of Data

**Logical representation (tables, fields, etc.)**

<table>
<thead>
<tr>
<th>id</th>
<th>name</th>
<th>address</th>
<th>class</th>
<th>dob</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345678</td>
<td>Jill Jones</td>
<td>Warren Towers 100</td>
<td>2007</td>
<td>3/10/85</td>
</tr>
<tr>
<td>25252525</td>
<td>Alan Turing</td>
<td>Student Village A210</td>
<td>2010</td>
<td>2/7/88</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Physical representation**
- disks
- (bytes on disk blocks, index structures, etc.)

- The DBMS takes care of translating between the representations.
  - makes the user's job much easier!

- This is an example of *abstraction*.
  - hide low-level details behind a simpler representation
  - an important concept in computer science
3. Query Processing

- A DBMS has some type of *query language*.
  - example: SQL
  - includes commands for:
    - adding new records
    - modifying or deleting existing records
    - retrieving data according to some criteria

- The DBMS determines the best way to execute a query-language command.
  - which index structures to use
  - if multiple operations are needed, the order in which they should be carried out

4. Transaction Management

- A *transaction* is a sequence of operations that is treated as a single logical operation.

- Example: balance transfer of $50 from blue to pink
  - remove $50 from blue
  - add $50 to pink
  ![Diagram of piggy banks with $450 and $300]
4. Transaction Management

- A transaction is a sequence of operations that is treated as a single logical operation.

- Example: balance transfer of $50 from blue to pink
  - remove $50 from blue
  - add $50 to pink

- Without a transaction, bad things could happen!

By using a transaction for the balance transfer, we ensure that all of the steps happen, or none do.

- all or nothing!

remove $50 from blue

*** CRASH ***

Money is lost!

remove $50 from blue

*** CRASH ***

restore original state
4. Transaction Management (cont.)

- Other examples:
  - making a flight reservation
    select flight, reserve seat, make payment
  - making an online purchase
  - making an ATM withdrawal

- Ensure that operations by different users don’t overlap in problematic ways.
  - example: what’s wrong with the following?
    
    ```
    user's balance transfer
    remove 500 from blue
    read blue balance
    read pink balance
    if (blue + pink < minimum)
      charge the user a fee
    add 500 to pink
    bank's check for clients below minimum balance
    ```

Database Applications

- Users often use a database application.
  - a separate piece of software that interacts with the DBMS
- Provide easier access the database.
  - don't need to know the query language
- Examples:
  - the software that runs on ATMs for a bank
  - a web interface to a library database
Desktop Database System

• Combines the functions of a database application and a DBMS.
  • examples: Microsoft Access, Filemaker Pro

• Includes tools/wizards for building the databases, forms, etc.

• Less flexible and less powerful than a full-fledged DBMS.
  • doesn't support all possible operations
  • doesn't support multi-user applications
  • doesn't scale well to very large databases

Looking Ahead

• The logical representation that a DBMS uses is based on some type of data model.

• There are a number of different models that can be used for this purpose.

• The most prevalent one is the relational model.

• We'll look next at the key features of this model.

• Reminder: complete Lab 0 by the first lab