information management

- organisations depend on information
  - about their own processes
  - about what’s going on around them
  - the basis of monitoring and planning
- the real world is too hard to keep track of
  - information abstracts and summarises it
    - brings the world into alignment with some model
    - denominate the work and treat the results like equations
    - equations represent the work
      - working with the equations tells you whether and how you need to address the work!

the factors at work

- Data: more is better
  - More data means more confusion
    - More confusion means simplifying use requirements
    - Need more resources

- Complexity
  - flat files
  - hierarchical database
  - network database
  - relational database

data, database, DBMS

- data
  - a big pile of bits
- a database
  - structured collection of data
  - organised according to predefined relations
    - paper documents?
    - contact list on my Pilot?
    - world wide web?
- why bother with a database?
  - need to maintain consistency
  - don’t want to have to repeat information

data, database, DBMS

- DBMS: Data Base Management System
  - set of programs to define, update, control databases
    - this is what we often mean when we say "database"
    - Sybase, Oracle, DB2, MySQL, Postgres…
- DBMS responsibilities
  - layout out information on the disk, building indexes, getting from one piece of data to another
- your responsibilities
  - modeling the information
  - describing the relations
  - creating queries

ER modeling

- identifying entities and the relationships between them
  - not unlike OO modelling, but entirely static
- types of relationships
  - one to one
  - one to many
  - optional one to many
  - many to many
ER modeling

• things to remember
  – the simplicity of ER is useful
  – ER is a communication tool – esp. with the participants
  – you’re dealing with generic entities, not specific

the relational model

• most common (but not the only one)
• database is a set of tables
  – each table expresses a relation between data items
  – each row of the table is a record
  – each column is an attribute
• not just any table will do
  – for instance, we need a key field
  • a field (or set of fields) that uniquely identifies every record
  – other properties are enforced by normalization
  • iteratively refining the database format for efficiency

first normal form

• no repeating groups
  – essentially, normalise the record length

<table>
<thead>
<tr>
<th>Title</th>
<th>Author</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where the Action Is</td>
<td></td>
<td>$30.00</td>
</tr>
<tr>
<td>Analyzing Social Settings</td>
<td></td>
<td>$31.95</td>
</tr>
<tr>
<td>Compilers</td>
<td></td>
<td>$72.00</td>
</tr>
</tbody>
</table>

second normal form

• no non-key attributes depend on part of the key
  – essentially, break the data into many tables

<table>
<thead>
<tr>
<th>Author</th>
<th>Title</th>
<th>Price</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dourish</td>
<td>Where the Action Is</td>
<td>$30.00</td>
<td><a href="mailto:jpd@ics.uci.edu">jpd@ics.uci.edu</a></td>
</tr>
<tr>
<td>Baldi</td>
<td>Bioinformatics</td>
<td>$49.95</td>
<td><a href="mailto:baldi@ics.uci.edu">baldi@ics.uci.edu</a></td>
</tr>
</tbody>
</table>
### second normal form

- no non-key attributes depend on part of the key
  - essentially, break the data into many tables

<table>
<thead>
<tr>
<th>Author</th>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dourish</td>
<td><a href="mailto:jpd@ics.uci.edu">jpd@ics.uci.edu</a></td>
</tr>
<tr>
<td>Baldi</td>
<td><a href="mailto:baldi@ics.uci.edu">baldi@ics.uci.edu</a></td>
</tr>
</tbody>
</table>

### third normal form

- no attributes depend on other non-key attributes
  - again, break the data into many tables

<table>
<thead>
<tr>
<th>Author</th>
<th>Title</th>
<th>Price</th>
<th>Purchaser</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dourish</td>
<td>Where the Action Is</td>
<td>$30.00</td>
<td>Maria</td>
<td>12/21/00</td>
</tr>
<tr>
<td>Dourish</td>
<td>Where the Action Is</td>
<td>$30.00</td>
<td>Joe</td>
<td>1/1/01</td>
</tr>
<tr>
<td>Baldi</td>
<td>Bioinformatics</td>
<td>$49.95</td>
<td>Lisa</td>
<td>1/2/01</td>
</tr>
</tbody>
</table>

### third normal form

- no attributes depend on other non-key attributes
  - again, break the data into many tables

<table>
<thead>
<tr>
<th>Title</th>
<th>Purchaser</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Where the Action Is</td>
<td>Maria</td>
<td>12/21/00</td>
</tr>
<tr>
<td>Where the Action Is</td>
<td>Joe</td>
<td>1/1/01</td>
</tr>
<tr>
<td>Bioinformatics</td>
<td>Lisa</td>
<td>1/2/01</td>
</tr>
</tbody>
</table>

### normalisation

- what’s the point?
  - eliminate redundancy
  - eliminate opportunities for inconsistency

<table>
<thead>
<tr>
<th>Author</th>
<th>Title</th>
<th>Price</th>
<th>Purchaser</th>
<th>StudentID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dourish</td>
<td>Where the Action Is</td>
<td>$30.00</td>
<td>Maria</td>
<td>12/21/00</td>
</tr>
<tr>
<td>Dourish</td>
<td>Where the Action Is</td>
<td>$25.00</td>
<td>Joe</td>
<td>1/1/01</td>
</tr>
<tr>
<td>Baldi</td>
<td>Bioinformatics</td>
<td>$49.95</td>
<td>Lisa</td>
<td>1/2/01</td>
</tr>
</tbody>
</table>

### the transaction model

- normalisation spreads data across multiple tables
  - single action requires many updates
    - a new customer placing a new order?
  - consistency is important
  - transactions group operations into logical units
the ACID properties

• Atomicity
• Consistency
• Independence
• Durability

getting it out again

• query languages
  – SQL is most common
  • "SELECT name,id FROM grades WHERE grade='A';"

3-tier architecture

distributing databases

• managing information access needs
  – locality
  – performance

• three forms of distribution
  – distributing tables
  – distributing rows
  – replication

• two-phase commit
  – "can commit?"
  – "do commit!"

• query languages
  – SQL is most common
  • "SELECT name,id FROM grades WHERE grade='A';"
alternatives to relational

- object-oriented
  - hierarchical schemas
  - migrate code closer to data
- text databases
  - free-form indexing
  - less structure
  - but more useful for unanticipated queries
- geographical information systems
  - not a natural model for relational systems

management concerns

- information quality
  - bad information is worse than none at all
    - it’s easy to load a database with accurate information
    - it’s harder to maintain the accuracy over time
    - distribution makes this worse
    - multiplicity of information, lack of “human access control”
- accessibility
  - the point of having the information is to use it
    - availability
    - admissibility
    - but there’s a downside...
    - once you have information, you may have to disclose it
    - security! (remember the risks, from last week)

organisational perspectives

- information all comes with a point of view
  - complete information is a myth; so what is left out?
- information models encode assumptions
  - about the state of the world or the objects modeled
  - example: US Army deployment
- normalisation distributes information
  - distributed locus of power and control

summary

- key points:
  - information processing is about making the world tractable
    - amenable to summarisation, modeling & prediction
  - DBMS provides a framework for data management
    - regularised for efficiency, consistency & maintenance
  - think about where the database fits
    - technically
    - organisationally
    - politically

homework

- See the web site for details
  - two questions
    - exercise in transforming a database into 1NF, 2NF, 3NF
    - explore DNS as a distributed database
  - due at next Wednesday’s lecture

what’s coming up

- Friday
  - discussion section
  - homeworks back
- Monday
  - performance and competition
  - Alter chapter 6