Announcements

- Informatics major
  - Information session tomorrow
    - 12:30-1:30 in MGH 420
  - Informatics video
Announcements

- Due Friday at 10pm
  - Vote for the Best GoPosters WebQ
    - 5 points XC for everyone who answers
    - Ten winners will also receive 10-40 points XC
  - Weekly GoPost discussion
Announcements

- Quiz in lab on Wednesday/Thursday
- Topics on GoPost
Database Tables, Views, and Design

D.A. Clements
Physical and Logical Database

TABLES AND VIEWS
Structure of a Database

- Physical database and logical database
  - Physical database is the files, records in any order, no logical organization other than tables
  - Logical database is a view of database that shows only the rows and fields needed by the users
  - Solves Information Overload and Security:
    - Users see only what they need
    - Users see only what they have permission to see

D.A. Clements, MLIS, UW iSchool
Figure 16.15  Structure of a database system. The physical database is the permanent repository of the data; the logical database, or view of the database, is the form of the database the users see. The transformation is implemented by the query processor, and is based on queries that define the logical database tables from the physical database tables.
Physical Database

- Designed by database administrators
  - Fast to access
  - No redundancy/duplicating information
    - Multiple data can lead to inconsistent data
  - Backup copies in case of accidental data deletion or disk crash
Logical Database

- Creating specialized views of the data for different users' needs
  - Creating a new “result set” from the current data each time
    - Fresh
    - Accurate
Defining Physical Tables

- Database schemes (schema)
  - Metadata specification that describes the database design
Figure 16.16 Table declarations from Microsoft Access 2007: (a) Home_Base table declaration shown in the design view; and (b) students table declaration. Notice that the key is specified by the tiny key next to Student_ID in the first column.
A **relationship** is a correspondence between rows of one table and the rows of another table.

- **key Student_ID** is used in each table,
  - Find address for each student (*Lives_At*)
  - Find the student for each address (*Home_Of*)

**Relationship examples**
Figure 16.17 The Relationships window from the Microsoft Access database system; the 1-to-1 Lives_At and Home_Of relationships are shown between Home_Base and Students.
Defining Logical Tables

● Constructing a View Using \textit{Join}

● Match on the common field of \textit{Student\_ID}

\texttt{Master\_List} = \texttt{Student JOIN Home\_Base}\n\texttt{On Student.Student\_ID = Home\_Base.Student\_ID}

\begin{verbatim}
Student\_ID
First\_Name
Middle\_Name
Last\_Name
Birthdate
On\_Probation
Street\_Address
City
State
Country
Postal\_Code
\end{verbatim}

\textbf{Figure 16.18} Attributes of the \texttt{Master\_List} table. Being created from \texttt{Student} and \texttt{Home\_Base} allows \texttt{Master\_List} to inherit its data types and key (\texttt{Student\_ID}) from the component tables.
Designing a Database

Hands on in Access and on paper

12/11/2009

D.A. Clements, MLIS, UW iSchool
Athletes and Teams

• “Business Rules”
  • What the database is about
  • What things are important
  • How things relate
Athletes and Teams

- A Database for an athletics department at a high school
- Storing details of:
  - Teams with
    - division,
    - gender,
    - coach
  - Student Athletes
- Individuals are selected for a team.
- Keep track of the points awarded to each student for participating in a sport for the awarding of school letters.
- The Database has to keep track of student Athletes over five years with any given Athlete participating in multiple sports in a given year.
Athletes and Teams

- THINGS of Interest, include:
  - Athletes
  - Events
  - Points earned for success
  - Teams

- These THINGS are related as follows:
  - A Student Athlete can participate in zero, one or many TEAMS.
<table>
<thead>
<tr>
<th>column</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>student_id</td>
<td></td>
</tr>
<tr>
<td>student_result_at_event</td>
<td></td>
</tr>
<tr>
<td>student_points_to_date</td>
<td></td>
</tr>
<tr>
<td>student_points_at_event</td>
<td></td>
</tr>
<tr>
<td>student_first_name</td>
<td></td>
</tr>
<tr>
<td>letters_sport_code</td>
<td></td>
</tr>
<tr>
<td>student_middle_name</td>
<td></td>
</tr>
<tr>
<td>letters_awarded_date</td>
<td></td>
</tr>
<tr>
<td>student_last_name</td>
<td></td>
</tr>
<tr>
<td>team_gender</td>
<td></td>
</tr>
<tr>
<td>student_date_of_birth</td>
<td></td>
</tr>
<tr>
<td>event_location</td>
<td></td>
</tr>
<tr>
<td>team_name</td>
<td></td>
</tr>
<tr>
<td>student_gender</td>
<td></td>
</tr>
<tr>
<td>team_description</td>
<td></td>
</tr>
<tr>
<td>student_address</td>
<td></td>
</tr>
<tr>
<td>coach_name</td>
<td></td>
</tr>
<tr>
<td>student_other_details</td>
<td></td>
</tr>
<tr>
<td>team_other_details</td>
<td></td>
</tr>
<tr>
<td>division_description</td>
<td></td>
</tr>
<tr>
<td>sport_description</td>
<td></td>
</tr>
<tr>
<td>event_name</td>
<td></td>
</tr>
<tr>
<td>event_start_date</td>
<td></td>
</tr>
<tr>
<td>event_end_date</td>
<td></td>
</tr>
<tr>
<td>event_other_details</td>
<td></td>
</tr>
</tbody>
</table>
Design the Database

- Divide into teams of three or four:
  - Design the Students and Teams database:
    - Decide what tables you would build.
    - Decide what fields you would put in each table.
    - List table names and attributes.
    - Choose primary keys.
    - List foreign keys in the foreign table.

- You have 15 minutes