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:
      - Users see only what they need
      - Users see only what they have permission to see

Physical vs. Logical

- 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

The Idea of Relationship

- 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_Af)
  - Find the student for each address (Home_Of)
- Relationship examples

Defining Logical Tables

- Constructing a View Using Join
  - Match on the common field of Student_ID
    Master_List = Student JOIN Home_Base
    On Student.Student_ID = Home_Base.Student_ID

Designing a Database

Hands on in Access and on paper
Athletes and Teams

- “Business Rules”
  - What the database is about
  - What things are important
  - How things relate

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.

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.

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

- student_id
- student_result_at_event
- student_points_to_date
- student_points_at_event
- student_first_name
- letters_sport_code
- student_middle_name
- letters_awarded_date
- student_last_name
- team_gender
- student_date_of_birth
- event_location

- team_name
- student_gender
- team_description
- student_address
- coach_name
- student_other_details
- team_other_details
- division_description
- sport_description
- event_name
- event_start_date
- event_end_date
- event_other_details