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

Physical vs. Logical

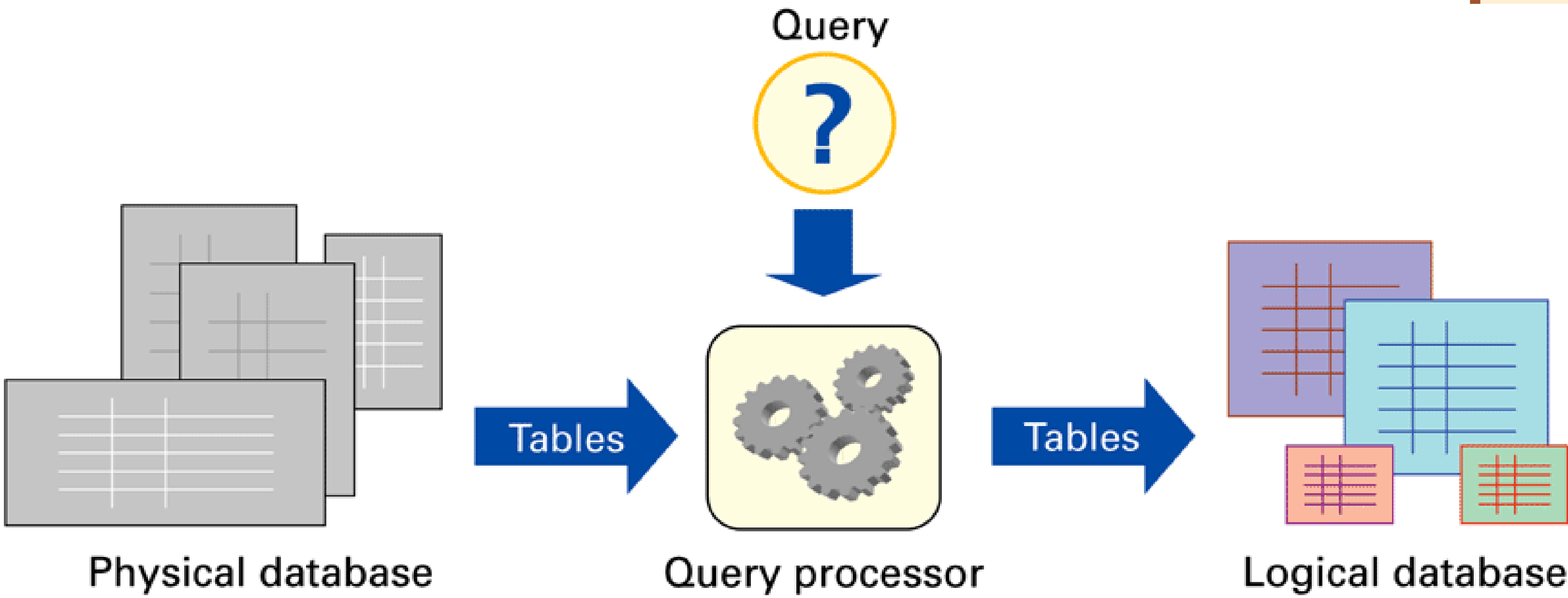


Figure 16.15 Structure of a database system. The physical database is the permanent repository of the data; the logical database, or view of the database, is the form of the database the users see. The transformation is implemented by the query processor, and is based on queries that define the logical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

Physical Database

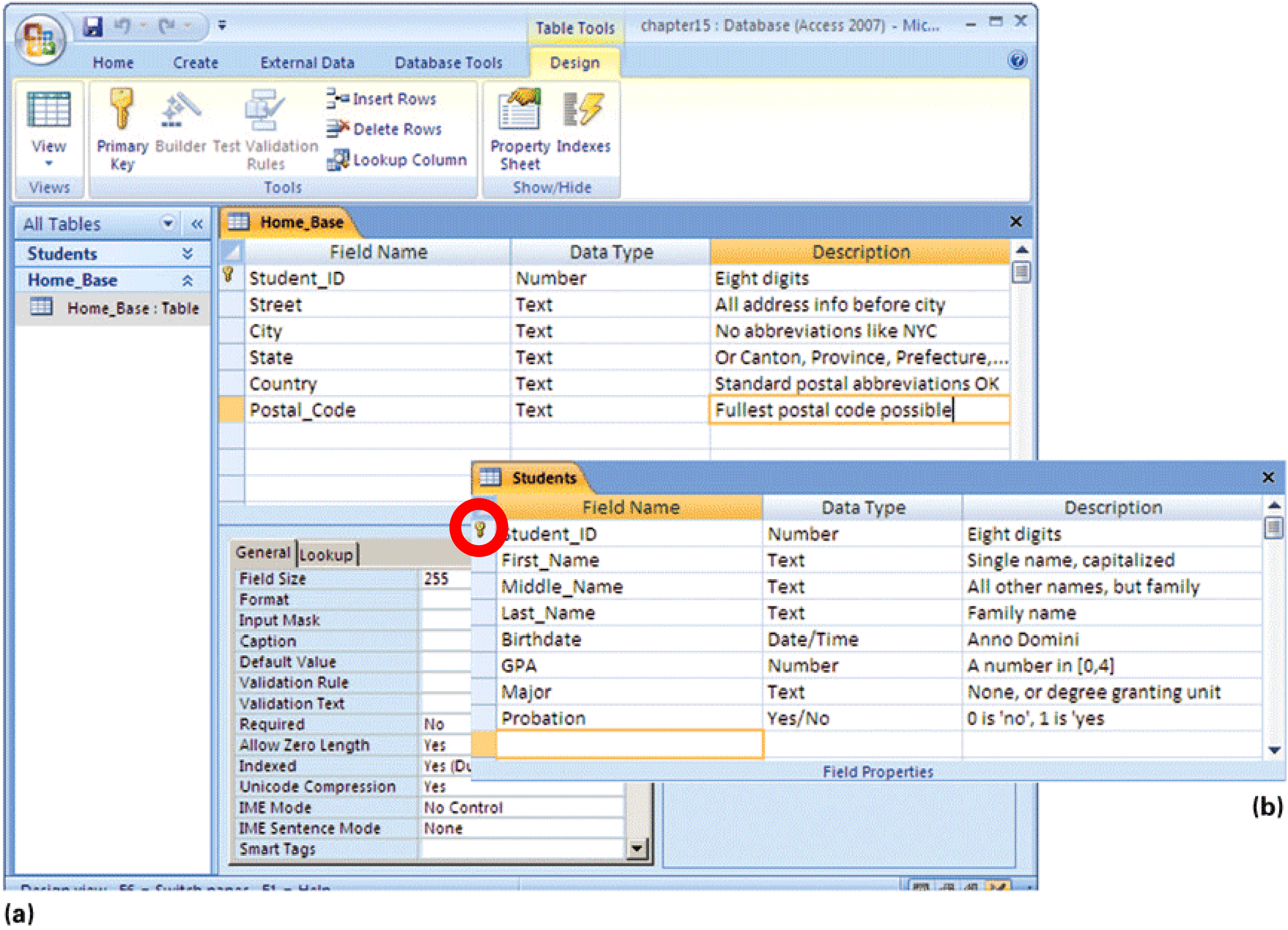
- Designed by database administrators
  - Fast to access
  - No redundancy/duplicating information
    - Multiple data can lead to inconsistent data
  - Backup copies in case of accidental data deletion or disk crash

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



Home_Base

Field Name	Data Type	Description
Student_ID	Number	Eight digits
Street	Text	All address info before city
City	Text	No abbreviations like NYC
State	Text	Or Canton, Province, Prefecture,...
Country	Text	Standard postal abbreviations OK
Postal_Code	Text	Fullest postal code possible

Field Size	255
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	Yes (Du
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

Students

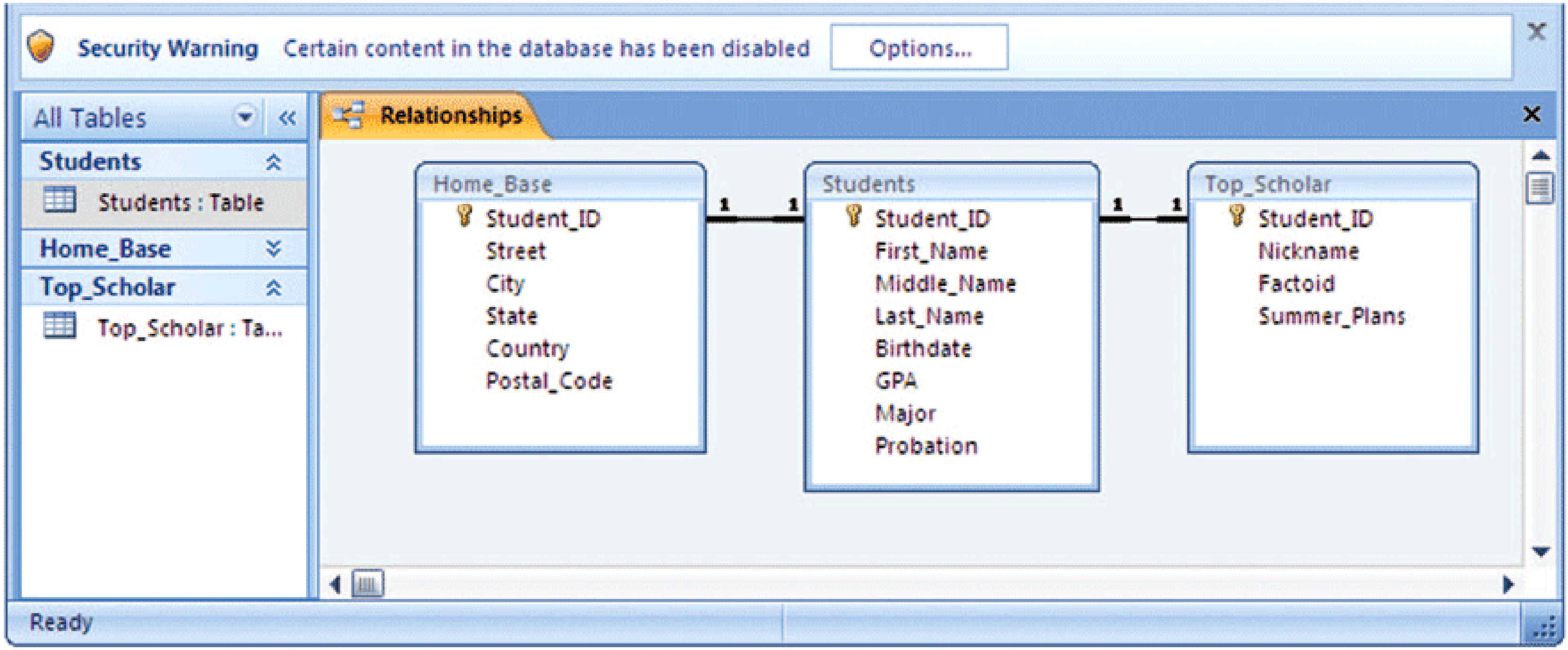
Field Name	Data Type	Description
Student_ID	Number	Eight digits
First_Name	Text	Single name, capitalized
Middle_Name	Text	All other names, but family
Last_Name	Text	Family name
Birthdate	Date/Time	Anno Domini
GPA	Number	A number in [0,4]
Major	Text	None, or degree granting unit
Probation	Yes/No	0 is 'no', 1 is 'yes

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.

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

Relationships in Practice



Security Warning Certain content in the database has been disabled

Home_Base
- Student_ID
- Street
- City
- State
- Country
- Postal_Code

Students
- Student_ID
- First_Name
- Middle_Name
- Last_Name
- Birthdate
- GPA
- Major
- Probation

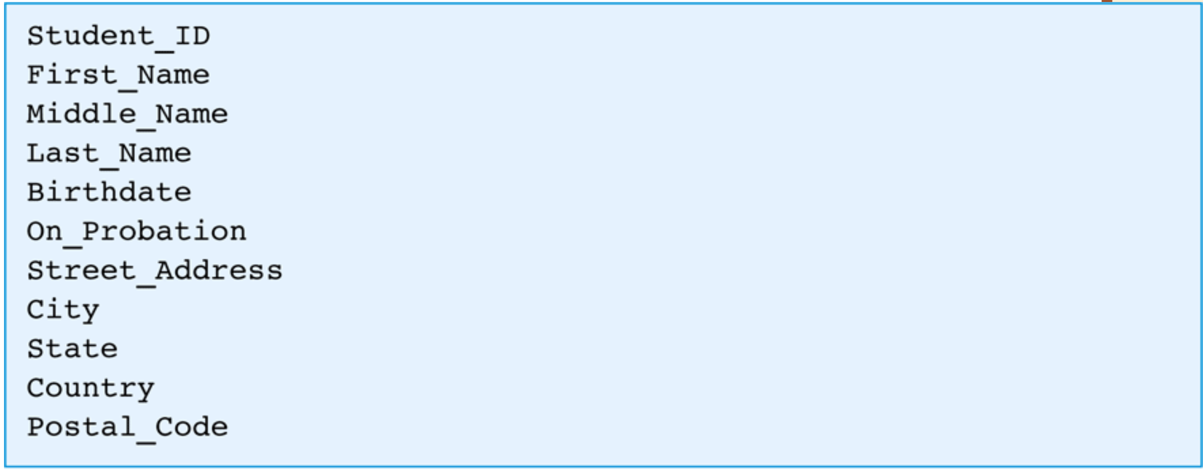
Top_Scholar
- Student_ID
- Nickname
- Factoid
- Summer_Plans

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 `Join`
  - Match on the common field of `Student_ID`

```
Master_List = Student JOIN Home_Base
  On Student.Student_ID = Home_Base.Student_ID
```



```
Student_ID
First_Name
Middle_Name
Last_Name
Birthdate
On_Probation
Street_Address
City
State
Country
Postal_Code
```

Figure 16.18 Attributes of the `Master_List` table. Being created from `Student` and `Home_Base` allows `Master_List` to inherit its data types and key (`Student_ID`) from the component tables.

15 min.

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

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.

Athletes and Teams

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

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