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:
      - 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

---

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

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

---

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	team_name
student_result_at_event	student_gender
student_points_to_date	team_description
student_points_at_event	student_address
student_first_name	coach_name
letters_sport_code	student_other_details
student_middle_name	team_other_details
letters_awarded_date	division_description
student_last_name	sport_description
team_gender	event_name
student_date_of_birth	event_start_date
event_location	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