Database Structure

Chapter 16

Structure Of A Database

- We want to arrange the information in a database in a way that users see a relevant-to-their-needs view of the data that they will use continually.

Physical vs. Logical Databases

- The point of the two-level system is to separate the management of the data (physical database) from the presentation of the data (logical view of the database).

Physical Databases

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

Logical Database

- Creating specialized versions/views of the data for different users' needs
- Creating a new copy from the single data each time
Queries

- A query is a specification using operations that define a table from other tables
- SQL (Structured Query Language)
  - Seen in last lecture
  - Standard database language to write queries

Defining Physical Tables

- Database schemas
  - Metadata specification that describes the database design

Connecting Database Tables by Relationships

- Different tables can have different security access restrictions based on their data
  - For example, some can access Home_Base data without having access to more sensitive data in Students
- Separate tables but not independent
  - Student_ID connects (establishes a relationship) between the two tables

The Idea of Relationship

- A relationship is a correspondence between rows of one table and the rows of another table
- Because the key Student_ID is used in each table, can find the address for each student (Lives_At) AND can also find the student for each address (Home_Of)

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

- Create a Master Table which combines 2 tables.

  - Construction Using Join
    - Match on the common field of Student_ID
      
      ```
      SELECT *
      FROM Students INNER JOIN Home_Base
      ON Students.Student_ID = Home_Base.Student_ID
      ```

Join Resulting Attributes

<table>
<thead>
<tr>
<th>Student_ID</th>
<th>First_Name</th>
<th>Middle_Name</th>
<th>Last_Name</th>
<th>Birthday</th>
<th>On_Probation</th>
<th>Street_Address</th>
<th>City</th>
<th>State</th>
<th>Country</th>
<th>Postal_Code</th>
</tr>
</thead>
</table>

Practical Construction Using QBE

- Query By Example
  - Given a template of a table we fill in what we want in the fields

The Dean's View

- Storing the Dean's Data
  - Top_Scholar is information of interest only to the dean
Join Three Tables into One

- Join using Top_Scholar, Student, and Home_Base tables matching on the Student_ID attribute across all three tables.

- Trim the Table
  - Project – retrieve certain columns

Creating A Dean's View


Exercise- Designing a DB

- Create a Database for administrative services in UW to manage courses and student info.

  - We need to store
    - Student’s Basic information
    - Courses offered (term, faculty, location etc)
    - Basic Information about courses(course no, department, name, etc)

Data to Capture

- Student id
  - first_name
  - middle_name
  - last_name
  - gpa
- Course id
  - department
  - course_name
  - course_description
  - SLN
  - term
  - location
  - when_meet
  - faculty
To find out courses taken by the Student.

- Need another Table
  - course_offering_id (matches with CourseOffering.id)
  - student_id (matches with StudentInfo.id)
  - grade_received