Announcements

• Free copy of Access, Vista, etc., for educational/academic use:
  - Links on Computing page on Course Web site
  - Search for CSE or INFO to find the link on the page
  - Username is your full UW email address
  - Password is different!
    - Click on "send a reminder"
    - Check wherever your email forwards to

Announcements

• Rubrics for Quick Writes will be available next week

Announcements

• Due date for Project 3A moved to Wednesday before noon
• Project 3B will be available Monday

A Table with a View (continued)

Primary keys, normalization, and SQL

Video

• Primary Keys (5 min.)

Primary/Foreign Key

• Controlled redundancy:
  - Stores relationship between tables
  - Database tables share common attributes only to enable the tables to be linked
  - True redundancy exists only when there is unnecessary duplication of attribute values
Problem Fields (Don’ts)

- Calculated field – can be computed by mathematical calculation or text concatenation
- Waste of storage space (redundant)
- No assurance the calculated value is updated when the user changes the input field(s)
- Multipart field – contains that should be two or more fields
- Extra work when you want to analyze your data
- Multivalue field – multiple correct entries for the field
- Derived field – contents of one or more fields absolutely predicts the contents of another
- Should be dropped from the table

Video

- Redundancy and Normalization (5 min.)

Entities

- An entity defines a table
  - Name of the entity is the name of the table
  - Each attribute of that entity
    - The column heading is the attribute name

Properties of Entities

- A relational database table can be empty
- Instances Are Unordered
  - Order of the rows and columns does not matter in databases
  - Freedom to move the data is limited to exchanging entire rows or exchanging entire columns

<table>
<thead>
<tr>
<th>Island</th>
<th>Name</th>
<th>Area</th>
<th>Elevation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Isabela</td>
<td>4588</td>
<td>1707</td>
<td></td>
</tr>
<tr>
<td>Fernandina</td>
<td>642</td>
<td>1494</td>
<td></td>
</tr>
<tr>
<td>Tower</td>
<td>14</td>
<td>76</td>
<td></td>
</tr>
<tr>
<td>Santa Cruz</td>
<td>986</td>
<td>846</td>
<td></td>
</tr>
</tbody>
</table>

Figure 16.4 A table instance for the island entity.
Properties of Entities (cont'd)

- **Uniqueness**
  - No two rows can be the same
  - Two rows can have the same value for some attributes, just not all attributes

- **Atomic Data**
  - Not decomposable into any smaller parts
    - Separate fields for street, city, state, postal code
  - "Only atomic data" rule relaxed for certain types of data
    - Dates, times, currency

Database schemes

- **Database schema** - way to define a table
  - Collection of table definitions that gives the name of the table, lists the attributes and their data types, and identifies the primary key

<table>
<thead>
<tr>
<th>Island</th>
<th>Island Name</th>
<th>Area (in square kilometers)</th>
<th>Elevation (higest point on the island)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Island</td>
<td>Text</td>
<td>Number</td>
<td>Text</td>
</tr>
<tr>
<td>Primary Key: Island</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Database Tables Recap

- Tables in databases have a structure that is specified by metadata
- The structure is separate from its content
- A table structures a set of entities
  - Things that we can tell apart by their attributes
  - The entities of the table are represented as rows
  - Rows and columns are unordered
- Tables and fields should have names that describe their contents
  - Fields must be atomic (indivisible)
  - One of more attributes define the primary key

Operations on Tables

- A database is a collection of tables
- Main use of database is to look up information
  - Users specify what they want to know and the database software finds it
- We can perform operations on tables to produce new tables
- The questions we ask of a database are answered with a whole new table, or view
Select Operation

- Takes rows from one table to create a new table
  - Specify the table from which rows are to be taken, and the test for selection

Syntax: `SELECT Test FROM Table`

- Test is applied to each row of the table to determine if it should be included in result table
- Test uses attribute names, constants, and relational operators
- If the test is true for a given row, the row is included in the result table; otherwise it is ignored

```
SELECT Interest='Beach' FROM Nations
```

Animation

- A natural join

```
Australia AU Canberra 37 S 148 E Beach
Bahamas BS Nassau 25 N 78 W Beach
Barbados BB Bridgetown 13 W 59 W Beach
Belize BD Belmopan 17 N 89 W Beach
 Bermuda BM Hamilton 32 N 64 W Beach
```

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 and Logical Database

`TABLES AND VIEWS`
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
  - Because the key Student_ID is used in each table, can not only find the address for each student (Lives_At), but can also 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

\[
\text{Master_List} = \text{Student JOIN Home_Base} \\
\text{On Student.Student_ID = Home_Base.Student_ID}
\]

16-32

Figure 16.10. Attributes of the Master_List table being created from Student and Home_Base tables. Note it is a disc entity and key

Student: Student_ID
First_Name
Last_Name
Birthdate
GPA
Subject
Semester
Grade
Exams

Home_Base: Home_ID
First_Name
Last_Name
Birthdate
Phone
Address
Career