Database Intro

INFO/CSE 100, Spring 2006
Fluency in Information Technology

http://www.cs.washington.edu/100
Readings and References

• Reading
  » Fluency with Information Technology
    • Chapter 13, Introduction to Database Concepts

• References
  » Access Database: Design and Programming
    • by Steve Roman, published by O'Reilly
Why Study Databases?

• Some of us want to compute, but all of us want information …
  • Much of the archived information is in tables
  • Databases enhance applications, e.g. Web
  • Once you know how to create databases, you can use them to personal advantage
  • Databases introduce interesting ideas

The Internet Movie Database
Visited by over 20 million movie lovers each month!
Welcome to the Internet Movie Database, the biggest, best, most award-winning movie site on the planet.
Relational Databases

- Information is stored in tables
  - Tables store information about \textit{entities}
  - Entities have characteristics called \textit{attributes}
  - Each row in a table represents a single entity
    - Each row is a set of attribute values
    - Every row must be unique, identified by a \textit{key}
  - Relationships -- associations among the data values are stored

Table structure = \textit{schema}
Table contents = \textit{instance}
A Table in a Database

Tables have names, attributes \{fields\}, entities \{rows\}

<table>
<thead>
<tr>
<th>ID</th>
<th>Last</th>
<th>First</th>
<th>JobID</th>
<th>Hire</th>
<th>Street</th>
<th>City</th>
<th>State</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Davelino</td>
<td>Nancy</td>
<td>0</td>
<td>5/1/1992</td>
<td>507 20th Ave E</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>2</td>
<td>Fuller</td>
<td>Andrew</td>
<td>3</td>
<td>8/14/1992</td>
<td>908 W. Capital Way</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>3</td>
<td>Wooster</td>
<td>Berton</td>
<td>1</td>
<td>4/1/1993</td>
<td>722 Moss Bay Elvd</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>4</td>
<td>Peacock</td>
<td>Margaret</td>
<td>2</td>
<td>5/3/1993</td>
<td>4110 Old Redmond Rd</td>
<td>Kirkland</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>5</td>
<td>Buchanan</td>
<td>Steven</td>
<td>3</td>
<td>10/17/1994</td>
<td>13 Garrett Hill</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>6</td>
<td>Sullimani</td>
<td>Okan</td>
<td>2</td>
<td>12/12/1994</td>
<td>Coventry House</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
</tbody>
</table>

Schema for Example table:

- **ID**: number, unique number (Key)
- **Last**: text, person’s last name
- **First**: text, person’s first name
- **JobCode**: number, current position
- **Hire**: date, first day on job

...
Two tables in a database
Redundancy in a database is Very Bad

- Not every assembly of tables is a good database
- Repeating data is a bad idea
  - Replicated data can differ in its different location
    - Multiple addresses can differ
      - Inconsistent data is worse than no data
      - Cut down on the typos and mis-keyed entries
  - Keep a single copy of any data
    - Reduces memory and data processing costs
    - If it is needed in multiple places, associate it with a key and store key rather than the data
  - Effort to update is high
Relationships between tables

The diagram illustrates relationships between tables by showing how data from one table can be linked to data from another table. The `Perms` table contains employee information, while the `JobCodes` table contains job title information. The `Relationships` diagram connects these tables, indicating how a job ID from the `Perms` table can be matched with a job title from the `JobCodes` table.
“You can look it up”

- When looking for information, a single item might be the answer, but a table is more likely
  - Which employees live in Kirkland?
    - Table of employees
  - Who is taking INFO/CSE 100?
    - Table of students
  - Whose mile run time ≤ 4:00?
    - Table of runners

Query to a database (set of tables) produces a new table
Relational Algebra: Tables From Tables

- There are five basic “algebraic” operations on tables:
  - Select -- pick rows from a table
  - Project -- pick columns from a table
  - Union/Join -- combine two tables w/like columns
  - Difference -- remove one table from another
  - Product -- create “all pairs” from two tables

From this basis, many more complicated operations can be built up
Select Operation

• Select creates a table from the rows of another table meeting a criterion

Select_from Example On Hire < 1993
Project

• Project creates a table from the columns of another table

**Project** Last, First From Example

This is a projection from 9 dimensions to 2 dimensions
### Union

- **Union combines two tables with *same attributes***

All employees = perms UNION temps

<table>
<thead>
<tr>
<th>ID</th>
<th>Last</th>
<th>First</th>
<th>JobID</th>
<th>Hire</th>
<th>Street</th>
<th>City</th>
<th>State</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dvalino</td>
<td>Nancy</td>
<td>0</td>
<td>01-May-92</td>
<td>507 20th Ave E</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>2</td>
<td>Fuller</td>
<td>Andrew</td>
<td>3</td>
<td>01-May-92</td>
<td>906 W. Capital Ave</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>3</td>
<td>Wooster</td>
<td>Barton</td>
<td>1</td>
<td>03-May-92</td>
<td>722 Moss Bay Dr</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>4</td>
<td>Peacock</td>
<td>Margaret</td>
<td>2</td>
<td>03-May-92</td>
<td>4110 Old Redmond Ave</td>
<td>Kirkland</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>5</td>
<td>Buchanan</td>
<td>Steven</td>
<td>3</td>
<td>10/17/94</td>
<td>13 Barrett Hill</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>6</td>
<td>Sullimani</td>
<td>Okan</td>
<td>2</td>
<td>12/12/94</td>
<td>4034 NW 50th St</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID</th>
<th>Last</th>
<th>First</th>
<th>JobID</th>
<th>Hire</th>
<th>Street</th>
<th>City</th>
<th>State</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Soggy</td>
<td>Peter</td>
<td>0</td>
<td>6/1/2004</td>
<td>1300 20th Ave W</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>2</td>
<td>Morken</td>
<td>Xavier</td>
<td>3</td>
<td>9/1/2003</td>
<td>100 Eastlake Dr</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>3</td>
<td>Wilshire</td>
<td>Bruce</td>
<td>1</td>
<td>3/1/1998</td>
<td>34 15th Ave NE</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>4</td>
<td>Brazely</td>
<td>Tanya</td>
<td>2</td>
<td>3/3/2002</td>
<td>103 25th Ave NW</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>5</td>
<td>Compton</td>
<td>Sarah</td>
<td>3</td>
<td>11/17/99</td>
<td>4034 NW 50th St</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
<tr>
<td>6</td>
<td>Zanzy</td>
<td>Ovid</td>
<td>2</td>
<td>1/12/1999</td>
<td>4502 NW 52nd St</td>
<td>Seattle</td>
<td>WA</td>
<td>USA</td>
</tr>
</tbody>
</table>
Difference

• Difference (written like subtraction) removes 1 table’s rows from another
  • Eastern = States - WestCoast
Product

• Product (written like multiplication) combines columns and pairs all rows

  Colors = Blues x Reds

  Column Rule: If A has $x$ columns, B has $y$ columns, $A \times B$ has $x+y$ columns

  Row Rule: If A has $m$ rows, B has $n$ rows $A \times B$ has $mn$ rows
Join

- Join (written like a bow tie) combines rows if common field matches

Employee List = Perms "<" JobCodes
DB Operations

• The five DB Operations can create any table from a given set of tables
  • All modern database systems are built on these relational operations
  • Join is not primitive, but can be built from 5
  • Join, select and project are used most often
  • The operations are not usually used directly, but are used indirectly from other languages

• Structured Query Language (SQL) is the language that we talk to the database in

SQL, the DB language we learn, is built on basic 5
Database Structure

• A database contains one or more *tables*
  » Tables include *entities* with *attributes*
  » There are *relationships* defined between the entities in the various tables
  » Retrieve information from the tables using *queries*
  » Create GUI front ends (*forms* and *reports*) for users

• First, design the database or create the schema
  » What are the entities?
  » What are the attributes of each entity?
  » What are the relationships between tables?
entity-relationship diagram for Library database
Create a new database

![Image of Microsoft Access interface showing the New Database window with file name set to "little.mdb" and save as type set to Microsoft Access Databases (*.mdb).]
Create a new table in the database
Creating a table in Design view
Entering Table Data
Build another table
Add publisher ID to books
Create the link between the tables
Hey presto, we have a database!
Two tables with a relationship
Create a query
The query produces a new (virtual) table
Project (select particular columns)
Select particular rows
SQL behind the scenes

```
SELECT books.*, publishers.*
FROM publishers INNER JOIN books ON publishers.ID=books.PubID;
```

```
SELECT books.ISBN, books.Title, publishers.Name
FROM publishers INNER JOIN books ON publishers.ID=books.PubID;
```

```
FROM publishers INNER JOIN books ON publishers.ID = books.PubID
WHERE (((books.Price)>15));
```