Announcements

Project 3A assigned today

Databases

Databases are collections of information; our study repeats a theme: Tell the computer the structure, and it can help you!

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

How much of your information can be in a table?

Stone Age Databases

Before relational databases (the kind we study) there were only “flat files”

- Structural information was difficult to express
- All processing of information was “special cased” -- custom programs were needed
- Information repeated; difficult to combine
- Changes in format of one file means all programs that ever process that file must be changed … adding ZIP codes

E.F. Codd of IBM invented relational databases

Relational Databases

Information is stored in tables

- Tables store information about entities -- things or stuff ... keep entities of one kind
- Entities have characteristics called attributes
- Tables are tuples (rows or records) of attributes (columns or fields)
- Every row must be unique, identified by a key
- Relationships -- associations among the data values are stored

Table structure = schema
Table contents = instance

A Table

Tables have names, attributes, tuples

Example:

<table>
<thead>
<tr>
<th>ID</th>
<th>Last text</th>
<th>First text</th>
<th>Hire date</th>
<th>Addr text</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Jones</td>
<td>Mary</td>
<td>08 May 1992</td>
<td>507 21st Ave</td>
</tr>
<tr>
<td>2</td>
<td>Fuller</td>
<td>Andrew</td>
<td>14 Aug 1992</td>
<td>560 W. Capital Way</td>
</tr>
<tr>
<td>3</td>
<td>Prado</td>
<td>Elton</td>
<td>01 Apr 1993</td>
<td>722 Moss Box Blvd</td>
</tr>
<tr>
<td>4</td>
<td>Peterson</td>
<td>Margaret</td>
<td>03 May 1993</td>
<td>4110 CL/McPherson Rd</td>
</tr>
<tr>
<td>5</td>
<td>Buchanan</td>
<td>Eliza</td>
<td>13 Apr 1994</td>
<td>13 Maple Rd</td>
</tr>
<tr>
<td>6</td>
<td>Sullivan</td>
<td>One</td>
<td>13 Apr 1994</td>
<td>13 Maple Rd</td>
</tr>
</tbody>
</table>
Redundancy Is Very Bad

Not every assembly of tables is a good database -- repeating data is bad

- Replicated data can differ in its different locations, e.g. multiple addresses can differ
- Inconsistent data is worse than no data
- Keep a single copy of any data, and if it is needed in multiple places, associate it with a key, and store key rather than the data

“You can look it up”

When looking for information, a single item might be the answer, but a table is more likely

- “Who is taking FIT100?” Table of students
- “Whose mile run time ≤ 4:00?” Runner table
- “Who won 2003 Grammy for ‘Best New Artist?’” A table containing only a single row
- “In what years has US won the World Cup?” Empty Table

Tables From Tables

There are five fundamental operations on tables to create tables:

- Select -- pick rows from a table
- Project -- pick columns from a table
- Union -- combine two tables with like columns
- Difference -- remove one table from another
- Product -- create “all pairs” from two tables

Though not primitive “Join” is usually included

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

Union

Union (written like addition) combines two tables with same attributes

- Political units = States + Provinces
### Difference
Difference (written like subtraction) removes 1 table’s rows from another

- **Eastern = States - WestCoast**

<table>
<thead>
<tr>
<th>Eastern Table</th>
<th>State</th>
<th>Capital</th>
<th>Sight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Washington</td>
<td>Olympia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oregon</td>
<td>Salem</td>
<td></td>
<td></td>
</tr>
<tr>
<td>California</td>
<td>Sacramento</td>
<td>Golden Gate</td>
<td>Grand Canyon</td>
</tr>
<tr>
<td>Arizona</td>
<td>Phoenix</td>
<td></td>
<td>Grand Canyon</td>
</tr>
<tr>
<td>Nevada</td>
<td>Carson City</td>
<td>Las Vegas</td>
<td></td>
</tr>
</tbody>
</table>

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

- **Colors = Blues x Reds**

<table>
<thead>
<tr>
<th>Product Table</th>
<th>Color</th>
<th>State</th>
</tr>
</thead>
<tbody>
<tr>
<td>Washington</td>
<td>Blues</td>
<td>Olympia</td>
</tr>
<tr>
<td>Oregon</td>
<td>Blues</td>
<td>Salem</td>
</tr>
<tr>
<td>California</td>
<td>Blues</td>
<td>Sacramento</td>
</tr>
<tr>
<td>Arizona</td>
<td>Blues</td>
<td>Phoenix</td>
</tr>
<tr>
<td>Nevada</td>
<td>Blues</td>
<td>Carson City</td>
</tr>
</tbody>
</table>

#### 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 (like $\times$) if common field matches

- **Homes = States $\times$ Students**

<table>
<thead>
<tr>
<th>Join Table</th>
<th>State</th>
<th>Capital</th>
<th>Sight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Washington</td>
<td>Olympia</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Oregon</td>
<td>Salem</td>
<td></td>
<td></td>
</tr>
<tr>
<td>California</td>
<td>Sacramento</td>
<td>Golden Gate</td>
<td>Grand Canyon</td>
</tr>
<tr>
<td>Arizona</td>
<td>Phoenix</td>
<td></td>
<td>Grand Canyon</td>
</tr>
<tr>
<td>Nevada</td>
<td>Carson City</td>
<td>Las Vegas</td>
<td></td>
</tr>
</tbody>
</table>

### 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

- SQL, the DB language we learn, is built on basic 5 operations