Databases

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

© 2006, Lawrence Snyder

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

<table>
<thead>
<tr>
<th>ID</th>
<th>Last</th>
<th>First</th>
<th>Hire</th>
<th>Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dake</td>
<td>Nancy</td>
<td>01 May 1992</td>
<td>551 20th Ave E</td>
</tr>
<tr>
<td>2</td>
<td>Tyler</td>
<td>Andrew</td>
<td>14 Aug 1992</td>
<td>500 W Capital Way</td>
</tr>
<tr>
<td>3</td>
<td>Monier</td>
<td>Betty</td>
<td>01 Apr 1993</td>
<td>722 Moss Bay Blvd</td>
</tr>
<tr>
<td>4</td>
<td>Persons</td>
<td>William</td>
<td>03 May 1993</td>
<td>413 W Oak Ridge Blvd</td>
</tr>
<tr>
<td>5</td>
<td>Buchanan</td>
<td>Diane</td>
<td>02 Aug 1994</td>
<td>599 W Oak Road</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
- Who is president of UW? Empty Table

Putting data into a DB (set of tables) produces tables

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 Hire < 1993 From Example

<table>
<thead>
<tr>
<th>ID</th>
<th>Last</th>
<th>First</th>
<th>Hire</th>
<th>Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>David</td>
<td>Nancy</td>
<td>01 May 1993</td>
<td>502 20th Ave E</td>
</tr>
<tr>
<td>2</td>
<td>Fuller</td>
<td>Andrew</td>
<td>14 Aug 1993</td>
<td>500 W Capital Way</td>
</tr>
</tbody>
</table>

Project

Project creates a table from the columns of another table

Project Last, First From Example

<table>
<thead>
<tr>
<th>ID</th>
<th>Last</th>
<th>First</th>
<th>Hire</th>
<th>Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>David</td>
<td>Nancy</td>
<td>01 May 1993</td>
<td>502 20th Ave E</td>
</tr>
<tr>
<td>2</td>
<td>Fuller</td>
<td>Andrew</td>
<td>14 Aug 1993</td>
<td>500 W Capital Way</td>
</tr>
</tbody>
</table>

Union

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

PoliticalUnits = States + Provinces

<table>
<thead>
<tr>
<th>States</th>
<th>Capital</th>
<th>Slight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oregon</td>
<td>Salem</td>
<td>Oregon</td>
</tr>
<tr>
<td>California</td>
<td>Sacramento</td>
<td>California</td>
</tr>
</tbody>
</table>

Difference

Difference (written like subtraction) removes 1 table’s rows from another

Eastern = States - WestCoast

<table>
<thead>
<tr>
<th>States</th>
<th>Capital</th>
<th>Slight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Washington</td>
<td>Olympia</td>
<td>Washington</td>
</tr>
<tr>
<td>Oregon</td>
<td>Salem</td>
<td>Oregon</td>
</tr>
<tr>
<td>California</td>
<td>Sacramento</td>
<td>California</td>
</tr>
</tbody>
</table>

PoliticalUnits Table

<table>
<thead>
<tr>
<th>Name</th>
<th>Capital</th>
<th>Slight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alberta</td>
<td>Edmonton</td>
<td>Alberta</td>
</tr>
<tr>
<td>British Columbia</td>
<td>Victoria</td>
<td>British Columbia</td>
</tr>
<tr>
<td>Ontario</td>
<td>Toronto</td>
<td>Ontario</td>
</tr>
</tbody>
</table>

为什么要使用表格？

- 数据存储和管理
- 优化查询性能
- 提供结构化信息
- 方便数据检索和分析
Product

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

- Colors = Blues \times Reds

Column Rule: If A has \(x\) columns, B has \(y\) columns, \(A \times B\) has \(x \times y\) 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 \(x\)) if common field matches

Homes = States \bowtie Students

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