Practical Database Design

Today's discussion

• What is a database?
• Which application should I use?
• Steps in designing a relational database
• Database terminology and concepts

• Live Demo
• Common design problems
• Redesigning the database

So, what exactly is a database?

• Word labels
• Excel Worksheet
• Rolodex
• Filing cabinet
• Telephone book
• Access database
• Word labels
• Excel Worksheet
• Tree-like structure
• Access database
• SQL database
• Research data
• Patient information
• Student information

Reasons for creating a computerized database

• Rapid access to information
• Eliminate or minimize duplication information
• Simplify data entry and reporting
• Accommodate expansion
• Maintain integrity of data
• Prevent access to unauthorized persons
• Provide statistics for analysis
• Provide summary data for papers, presentations

Top Down Approach

• Data model that is well organized
• Details can be easily overlooked

Bottom Up Approach

• Helps insure no data overlooked
• Overall organization may not be so apparent

• Database Designed
• Relationships Determined
• Data Entities Identified
• Proposed Database
• Information Model
• Systems Planning

Manual/ Non-computerized

Flat/ hierarchal

Relational
## Combined Approach

Cross check for both completeness and good organization.

- Systems Planning
- Information Model
- Database Designed
- Relationships Determined
- Proposed Database
- Data Entities Identified

## What is the purpose of your database?

- Write out statement of purpose
- How do you collect data?
  - Who collects it
  - Who enters it into the database, what is their experience level (with data and computers)
- What’s the expected lifetime of the project?
- Can the same form be used for multiple projects – either now or in the future
- What kinds of analysis are you going to do?
  - Who will analyze the data
  - Who will want reports
- Brainstorm questions about database
- Sketch out workflow - reports
- Gather current forms and reports

## Which application should I use?

### Spreadsheet Strengths

1. Ease of data entry
2. Calculations and statistical functions available
3. Charting capabilities
4. Pivot table reports

### Spreadsheet Weaknesses

1. Hard to customize data entry for complicated situations
2. Not good at managing large amounts of data with complex relationships (patient records with multiple visits)
3. Hard to sort or select for specific subsets of data
4. Complex statistics are usually beyond its reach

## Which application should I use?

### Excel

1. Single table of data, fairly small amount
2. Numbers and financial data
3. Sophisticated what if models and cost benefit analyses
4. Charting and data bars
5. Pivot tables
6. Familiarity/preference

### Good Excel Example

Avoid wrapped text in headings.
## Practical Database Design

### Poor Example

<table>
<thead>
<tr>
<th>Company Name</th>
<th>Inventory/Asset Cost of Goods Sold Analysis</th>
<th>Unit Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Which application should I use?

#### Relational Database Management System Strengths
1. Manage and track large amounts of data
2. Powerful structure to develop queries to find specific information
3. Relatively easy to create data entry forms and reports in complex situations
4. Fairly easy to make relatively good, if not publication quality charts
5. Allows multiple user access at the same time
6. Relatively easy to import/export data from/to other applications

#### Relational Database Management System Weaknesses
1. Limited statistical capability largely restricted to basic descriptive measures
2. Charting and graphing capability is often limited
3. Learning curve

### Which application should I use?

#### Access Database
1. Store and manage lots of data, especially repeated data
2. Keep track of actions or events?
3. More than one table related to each other
4. Want to create data entry forms
5. More complex filters/queries
6. Generate variety of reports
7. Multiple users entering data at the same time

### Access 2007

Making a relational database work

- Tables
- Forms
- Queries
- Reports
- Macros
- Modules
**Practical Database Design**

**Rooms “Tables” where you store your data**

**Furniture & Possessions “Fields” What you keep in the rooms – “records”**

**Windows “Queries” Ways of looking at what is in the house, from different perspectives**

**Entrances, doors “Forms & Reports” Ways of getting data in and out of the house**

---

**Common Design Problems**

---

**Structural Design Flaws**

- Additions, Deletions, and Updates behave unpredictably
- Inability to retrieve information efficiently
- Redundant information
- Inconsistent information
- Loss of information (zip codes)
- Doesn’t export easily

---

**Preventable Data Entry Errors**

- These are data entry errors that the system could either prevent or provide a warning if the right data validation rules are in place.
  1. Data out of range
  2. Using the wrong type of variable, entering mixed character and numeric data – e.g., 98.6F
  3. Loss of information due to rounding errors

---

**Operator Errors**

- These are data entry errors that could not be caught by the system. Some examples would be:
  1. Inconsistent spellings or terms for the same thing
  2. Reversed letters and/or numbers
  3. Wrong codes
  4. Inaccurate entry within a variable’s acceptable range

---

**How to avoid errors**

- Three-pronged approach at table design and forms design level
  1. Formal model constrains what can be done and insures data integrity
  2. Data validation rules
  3. Basic design principles, psychology
Practical Database Design

Steps in Designing a Database

- Purpose of your database
- Table(s) you'll need
- Fields in each table
- Relationships
- Refine your design
- Add data and create other database objects

Identify Facts about your Database

Lab Tests
- TestID
- Date
- RatID
- TestIDCode
- Description
- Application

Rats
- RatID#
- Weight
- M/F
- DOB

Primary Key Field

- Uniquely defines a row in a table
- Minimal size
- Stable – should not change
- Simple and familiar

Foreign field

- Primary key in one table, used to relate (join) records in another table
- Has the same type and properties of the primary field (doesn’t have to have the same name)
Other Keys

Compound key
Composed of two or more columns, such as DateDone and TestName (if a specific test is done only once per day)

Surrogate key
Doesn’t come from the data, such as autonumber

Determine Relationships

One-to-one relationship — inner join

One-to-many relationship — outer join
This is where a row in one table (the parent table) has one or more matching rows in a second (child) table.
This is probably the most common relationship;

Many-to-many relationship
Row in table 1 has many matching rows in table 2, and a row in table 2 has many matching rows in table 1.
Requires a third intermediate table

Sample Relationship Screen
Refine your design
Enter sample data
Many empty fields?
Duplicate information in your tables?
Good primary key for each table?
Forget any fields?
More on Relationships and Referential Integrity

Add data and other database objects

Small number of records initially

Continue to refine

Create forms, queries, reports

Test to see if you're getting the desired results

Other considerations

• Updating and modifying data
• Indexes needed?
• Security
• Backup and recovery
• Documentation

Did you inherit someone else’s database?

Tools > Analyze > Documenter

Sample Database Scenarios

• Birthday cards
• Home music library
• Home inventory
• Employee equipment inventory
• Research data on rat experiments

Database terms

Hierarchical database
Relational database
Database objects
1. Tables - tblName
2. Fields
3. Queries - qryName
4. Forms - frmName
5. Reports - rptName
6. Properties

Relationships
1. Primary keys
2. Foreign keys

Referential integrity

Validity

Normalizing

Tools > Analyze > Documenter
These are probably not appropriate for a research audience. Depending on time, we may not have time for this. I think more effective to show good and bad examples.

Klaersen, 10/19/2010
Building a Database
Live Demo
Tom Reardon

Where to Get More Information

- Detailed handout and recording of today’s session at RUG web page — www.unmc.edu/its/292.htm
- Training sessions- ITS Web page www.unmc.edu/its - classes and training link
- On-line resources
  - http://office.microsoft.com
  - DVD tutorials – 6th floor circulation desk of the Library
  - Books, articles; electronic sources
  - The Missing Manual for Access 2007 – Matthew McDonald
- ITS Customer Support – 559-7700
- Academic programming – Tom Reardon – 9-5671
- College courses