BUSINESS DATABASE DEVELOPMENT

Software Development Lifecycle

- Define Requirements
- Design
- Implement
- Test

Let's apply this to **database applications**

1. Define Requirements

- Understand problem to be solved
- Document database requirements by creating
  - Draft E-R model
  - Draft Data dictionary
  - Specifications for data entry forms and reports

Data Dictionary

- Defines entities and attributes unambiguously

Contestant

A contestant is a student who participates in a contest.

Attribute	Data Type	Definition	Example
ContestantID	int*	Sequentially generated identifier that uniquely identifies a contestant	253
Name	Char(40)*	Contestant full name	Fred Jones
Classification	Integer*	Contestant's classification in school	Values: •31 - Freshman •32 - Sophomore •33 - Junior •34 - Senior
Status	Char(1)*	Contestant's registration status	Values: •R - Requested •A - Approved

2. Design System

- Finalize E-R model and data dictionary
- Design data entry forms and reports

3. Implement System

- Create schema from E-R model
- Create database and tables
- Populate with initial data
- Create views to support reports
- Create forms and reports using appropriate tools
  - Microsoft Access?
  - Traditional programming language?

4. Test

- Verify system meets specifications and solves user's problems

Database Requirements Analysis

Database Requirements Analysis

- Problem:
  - Gather and analyze requirements for a database

Requirements Analysis: How To

- Step One: Gather Information
  - Interview client
  - Obtain existing forms and reports and descriptions of existing processes
- Step Two: Analyze Information
  - Perform top-down and bottom-up analysis
  - Draft E-R model and data dictionary
- Step Three: Review and Finalize Requirements Documents

Requirements Analysis

- Two overall approaches:
  - Top-down
  - Bottom-up

Top-Down Approach

- Interview user to understand problems to be solved
- Define requirements for database to solve these problems

Bottom-Up Approach

- Inspect existing client processes and paperwork
- Define requirements based on analyzing these artifacts

Best Practice

- Combine both approaches
- Important to understand user's problems
  - Helps us build functionality the user needs
  - Helps us avoid building functionality user doesn't need
- Examining existing processes and paperwork helps ensure we don't miss important requirements

Creating E-R Model

Creating E-R Model

1. Start by identifying entities
2. Create initial E-R model with entities only
3. Refine to add attributes and relationships
4. Test and refine the model

Step 1: Identifying Entities

- Make a list of key nouns found in notes from client interviews
  - Focus on nouns that you suspect are candidates for entities
- Consolidate nouns into unique concepts
  - Identify and eliminate synonyms
- Start data dictionary by defining each noun in the consolidated list

Class Exercise

1. Make a list of nouns that appear to represent entities for ABC Game Rental
2. Review list to identify synonyms for the same idea
3. Refine list to eliminate synonyms

Refined Noun List

- Customer – someone who rents games
- Game – a game title; we have several copies for rent
  - Synonyms: Title
- Copy – a physical game that can be rented
- Rental – _____
  - Synonymns: Order
- Gift Certificate

Step 2: Start E-R Model

- Using consolidated list of nouns, create initial E-R diagram
  - Place each noun on the E-R diagram as an entity rectangle
- Add obvious relationships

Step 3: Add Attributes and Relationships

- Add attributes and relationships to diagram
- Remember not to include foreign keys on model

Class exercise: Add relationships to E-R diagram

Step 4: Test and Refine Model

- Review notes and other documentation provided by client
  - Especially forms and reports
- Ensure that model includes all information needed
- Ensure that model is consistent with information