Database Design Process

IST 210, Section 1
Todd S. Bacastow
January 2004
Welcome to Your New Home!
Key points

- Database design must reflect the information system of which the database is a part.
- Information systems undergo evaluation and revision within a framework known as the Systems Development Life Cycle (SDLC).
- Databases also undergo evaluation and revision within a framework known as the Database Life Cycle (DBLC).
- There are two general design strategies exist:
  - top-down vs. bottom-up design
  - centralized vs. decentralized design
Changing Data into Information

- Data
  - Raw facts stored in databases
  - Need additional processing to become useful

- Information
  - Required by decision maker
  - Data processed and presented in a meaningful form

- Transformation
The Information System

- **Database**
  - Carefully designed and constructed repository of facts
  - Part of an information system

- **Information System**
  - Provides data collection, storage, and retrieval
  - Facilitates data transformation
  - Includes people, hardware, and software
    - Software: Database(s), Application programs, and Procedures
The Information System (Con’t.)

- **System Analysis**
  - Establishes need and extent of an information system
    - Refer to *Recommended Requirements Gathering Practices*
    - We are NOT DOING A SYSTEM REQ’T ANALYSIS!!

- **Systems development**
  - Process of creating information system

- **Database development**
  - Process of database design and implementation
  - Creation of database models
  - Implementation
    - Creating storage structure
    - Loading data into database
    - Providing for data management
Systems Development Life Cycle

- System Analysis
- Database Organization (IST 210)
Database Lifecycle (DBLC)

Phase 1
Phase 2
Phase 3
Phase 4
Phase 5
Phase 6

Database initial study
Database design
Implementation and loading
Testing and evaluation
Operation
Maintenance and evaluation

Database Organization (IST 210)
Phase 1: Database Initial Study

- Purposes
  - Analyze company situation
    - Operating environment
    - Organizational structure
  - Define problems and constraints
  - Define objectives
  - Define scope and boundaries
Initial Study Activities

- Analysis of the company situation
  - Company objectives
  - Company operations
  - Company structure
- Definition of problems and constraints
- Database system specifications
  - Objectives
  - Scope
  - Boundaries
Phase 2: Database Design

- Most Critical DBLC phase
- Makes sure final product meets requirements
- Focus on data requirements
- Subphases
  - I. Create conceptual design
  - II. DBMS software selection
  - III. Create logical design
  - IV. Create physical design
Two Views of Data

Manager's view
- What are the problems?
- What are the solutions?
- What information is needed to implement the solutions?
- What data are required to generate the desired information?

Designer's view
- How must the data be structured?
- How will the data be accessed?
- How are the data transformed into information?
I. Conceptual Design

- Data modeling creates abstract data structure to represent real-world items
- High level of abstraction
- Four steps
  - Data analysis and requirements
  - *Entity relationship modeling and normalization*
  - *Data model verification*
Data analysis and Requirements

- Focus on:
  - Information needs
  - Information users
  - Information sources

- Data sources
  - Developing and gathering end-user data views
  - Direct observation of current system
  - Interfacing with systems design group

- Business rules
## Entity Relationship Modeling and Normalization

<table>
<thead>
<tr>
<th>STEP</th>
<th>ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identify, analyze, and refine the business rules.</td>
</tr>
<tr>
<td>2</td>
<td>Identify the main entities, using the results of Step 1.</td>
</tr>
<tr>
<td>3</td>
<td>Define the relationships among the entities, using the results of Steps 1 and 2.</td>
</tr>
<tr>
<td>4</td>
<td>Define the attributes, primary keys, and foreign keys for each of the entities.</td>
</tr>
<tr>
<td>5</td>
<td>Normalize the entities.</td>
</tr>
<tr>
<td>6</td>
<td>Complete the initial E-R diagram.</td>
</tr>
<tr>
<td>7</td>
<td>Have the main end users verify the model in Step 6 against the data, information, and processing requirements.</td>
</tr>
<tr>
<td>8</td>
<td>Modify the E-R diagram, using the results of Step 7.</td>
</tr>
</tbody>
</table>
E-R Modeling is Iterative

Diagram:

1. Database initial study
2. DBLC processes and database transactions
3. Data analysis, user views, and business rules
4. Initial E-R model
5. Verification
6. Normalization
7. Attributes
8. Final E-R model
Concept Design: Tools and Sources

Information sources
- Business rules and data constraints
- Data flow diagrams (DFD*)
- Process functional descriptions (FD)* (user views)

Design tools
- E-R diagram
- Normalization
- Data dictionary

Conceptual model
- ERD
  - Definition and validation
Data Model Verification

- E-R model is verified against proposed system processes
  - End user views and required transactions
  - Access paths, security, concurrency control
  - Business-imposed data requirements and constraints
- Reveals additional entity and attribute details
# E-R Model Verification Process

<table>
<thead>
<tr>
<th>STEP</th>
<th>ACTIVITY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Identify the E-R model’s central entity.</td>
</tr>
<tr>
<td>2</td>
<td>Identify each module and its components.</td>
</tr>
</tbody>
</table>
| 3    | Identify each module’s transaction requirements:  
  | Internal: Updates/Inserts/Deletes/Queries/Reports  
  | External: Module interfaces |
| 4    | Verify all processes against the E-R model. |
| 5    | Make all necessary changes suggested in Step 4. |
| 6    | Repeat Steps 2 through 5 for all modules. |
Iterative Process of Verification
II. DBMS Software Selection

- DBMS software selection is critical
- Advantages and disadvantages need study
- Factors affecting purchasing decision
  - Cost
  - DBMS features and tools
  - Underlying model
  - Portability
  - DBMS hardware requirements
III. Logical Design

- Translates conceptual design into internal model
- Maps objects in model to specific DBMS constructs
- Design components
  - Tables
  - Indexes
  - Views
  - Transactions
  - Access authorities
  - Others
IV. Physical Design

- Selection of data storage and access characteristics
  - Very technical
  - More important in older hierarchical and network models
- Becomes more complex for distributed systems
- Designers favor software that hides physical details
Phase 3: Implementation and Loading

- Creation of special storage-related constructs to house end-user tables
- Data loaded into tables
- Other issues
  - Performance
  - Security
  - Backup and recovery
  - Integrity
  - Company standards
  - Concurrency controls
Phase 4: Testing and Evaluation

- Database is tested and fine-tuned for performance, integrity, concurrent access, and security constraints
- Done in parallel with application programming
- Actions taken if tests fail
  - Fine-tuning based on reference manuals
  - Modification of physical design
  - Modification of logical design
  - Upgrade or change DBMS software or hardware
Phase 5: Operation

- Database considered operational
- Starts process of system evaluation
- Unforeseen problems may surface
- Demand for change is constant
Phase 6: Maintenance and Evaluation

- Preventative maintenance
- Corrective maintenance
- Adaptive maintenance
- Assignment of access permissions
- Generation of database access statistics to monitor performance
- Periodic security audits based on system-generated statistics
- Periodic system usage-summaries
DB Design Strategy Notes

- Top-down
  1) Identify data sets
  2) Define data elements

- Bottom-up
  1) Identify data elements
  2) Group them into data sets
Top-Down vs. Bottom-Up
Centralized vs. Decentralized Design

- Centralized design
  - Typical of simple databases
  - Conducted by single person or small team

- Decentralized design
  - Larger numbers of entities and complex relations
  - Spread across multiple sites
  - Developed by teams
Decentralized Design