Timeless Theory vs. Changing Users:

Reconsidering Database Education

Purpose of the Session

- Demonstration of subject matter mastery, teaching skills
  - But theme topic required
- Focus on my two divergent roles
  - Database and application consultant
  - Historian of corporate computing
- What insights do these bring to database education

Structure of Talk

1. The Timeless Principles (4)
2. Sources of the Principles
3. Trends in Practice (5)
4. Possible Responses

Assumptions For Talk

- Am assuming familiarity with
  - Basic database concepts
  - Current application technologies
- Please stop and ask question if appropriate

4 Pillars of Database Education

- DBMS Concept
- Relational Model
- SQL
- Entity Relationship Modeling

- All “Timeless Principles”
- All 70s ideas, commercialized in 80s

Strengths of Approach

- Gives principles, not technical skills
  - Relational model is now ubiquitous
  - SQL is lingua-franca of databases
- Many principles of good design are universal
  - ERM forces students to think before making tables
  - Normalization is a very powerful idea
  - Data-centric way of thinking is very different from procedural way

Sources of the Principles

1: Database Management System

- Key concepts from CODASYL Database Task Group (1971)
- DBMS as software layer between data, users
  - Different interfaces, languages for
    - Programs & programmers
    - Ad-hoc managerial reporting
    - Data definition and maintenance

2: The Relational Model

- E. F. Codd, 1970
- Simple, elegant, mathematically grounded
  - Abstracts data from underlying representations
  - Relations specified by query, not by DDL

3: SQL

“Looking back on it, I don't think the problem we thought we were solving was where we had the most impact. What we thought we were doing was making it possible for non-programmers to interact with databases.”

Don Chamberlin

– System R SQL Language group,

4: Entity Relationship Modeling

- Formulated by Chen, 1976
- Links database entities to real-world functions and processes
- Easy to convert to relational design

Trends in Practice

Database Technology – 1980s

- Databases and servers (mainframe/ large mini) are
  - Expensive
  - Centralized
  - Run by expert staff
- Database and applications are separate
  - Applications are monolithic, self contained

Database Methodology – 1980s

- Focus is on design of system from scratch
- Construction of database is separate from, comes before, applications that it supports
- Structure of database reflects real world entities

5 Trends in Practice:

1. Diversity of Scale
2. Diverging Uses and Users
3. Merging of DB and application platforms
4. Integrating Database and Application Development
5. Proliferation of Existing Databases

1: Diversity of Scale

- Enterprise (Data Warehouse, ERP)
- Departmental
- Workgroup
- Desktop
- Handheld

2: Diverging Uses & Users

- World's leading
  - Programming Language: Visual Basic
  - DBMS/Application Platform: Microsoft Access
- Powerful relational tools in the hands of end users
- Most IS departments lack resources/mindset to support

3: N-tier Architectures

- Database and application platforms merging
  - Oracle now includes file system, Java language and Web support
  - Close ties between ColdFusion, ASP, etc. and DBMS
  - Business logic is migrating to DMBS

4: Application Development

- Database now at the heart of all corporate applications
  - And behind every serious web site
- So database and applications must be developed together
  - Yet very different software engineering methodologies apply to
    - departmental applications,
    - enterprise systems,
    - web projects etc.

5: Database Proliferation

- Most applications are now purchased, not developed
  - Often have to build links to vendor-supplied database
  - Problem is integration into other systems
- How to incorporate data from warehouse or datamart?
- Some or most of data often already in local database
  - Clean? Combine? Discard?

- Educational Implications & Responses

ERM is a Tool

- Should be justified as part of broader methodology
  - Is hybrid – object oriented data but little support for business rules
- Mapping of real-world to ERM is non-deterministic
  - Choice of model reflects tradeoffs, demands of application
  - Requirements analysis is non-trivial, non-mechanical

ERM – Use Must Be Justified

- “It's a good thing these folks are book writers and academics and do not design databases for a living…. The text is filled with Entity Relationship diagrams that must add 200% to the cost of their designs…. if you're already designing databases, this methodology will drive you up the wall.”
- Amazon.com user review of Connolly & Begg textbook

Responses: Education Structure

- Integrate use of database technology into other curriculum areas
- Involve core courses in shared project
  - systems analysis,
  - user interface design
  - application architecture
  - E-business
- Include exposure to real databases and situations

Response: More Cases

- "How Not To"
  - We learn from mistakes, ideally those of other people.
  - Expose students to real databases
  - More "case based" teaching

Response: Give Guidelines, not Commandments

- Admit that different styles of database development are appropriate for different situations
  - Eg when NOT to normalize!
  - When to keep application meta-data in database

Response: More Context

- Discuss roles, situations in which databases are developed
  - Role of database expert in application development team
  - …as management analyst
  - …as end user
- Mention organizational/political aspects of databases

In Short

- Supply not just the technical and conceptual **tools**
- But an idea of when and why to use them