

# Database Design Concepts and Practices

Keith T. Weber  
GIS Director

Ohio State University



---

---

---

---

---

---

---

---

## Basic Steps in Database Design

- Understand and document the business' needs.
  - Problem statement
  - Business object types
  - Business relationships
- Business constraints
- Create an ERM
- Data and process inventory
- Develop tuple types
  - Tuple types to tables
  - Integrity
  - Implement the database



---

---

---

---

---

---

---

---

## Today's goal

- Provide details regarding database design using UML Visual Modeling techniques.
- Apply design concepts in the development of a "mock" database.

---

---

---

---

---

---

---

---

## Database Design

- Why spend so much time and effort?
  - Efficiency (speed, storage)
  - Client satisfaction
  - Flexibility
  - Cost savings realized

---

---

---

---

---

---

---

---

## Design Considerations

- Basic steps (described earlier)
- Data types
- Normalization
- With > 1 table, relationships must be examined

---

---

---

---

---

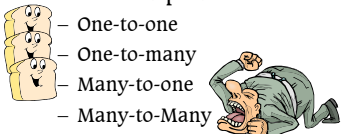
---

---

---

## Relationships

- Determine where relationships exist between tables
- Determine the type of relationship that exists



---

---

---

---

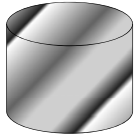
---

---

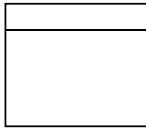
---

---

## Generic Design Symbology



= **Database**



= **Table**

---

---

---

---

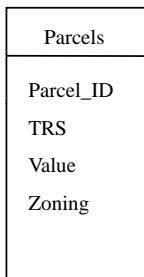
---

---

---

---

## Generic Table Symbology



- ← Table name
- ← Divider
- ← List of all attributes stored in this table as they will appear in the table

---

---

---

---

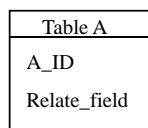
---

---

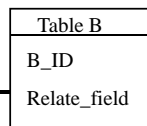
---

---

## Generic Relationship Symbology



- Draw schema of RDB
- Determine relationship fields
- Connect



---

---

---

---

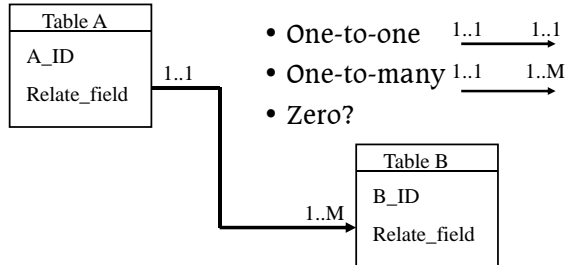
---

---

---

---

## Symbolizing Relationship Type



---

---

---

---

---

---

---

---

## Questions?

---

---

---

---

---

---

---

---

## Rational Rose Software

- What is Rational Rose?
  - CASE Tool
  - Computer Assisted Software Engineering tool
  - Performs visual modeling.
  - Object Oriented design



---

---

---

---

---

---

---

---

**Object-Oriented Fundamentals:  
How is Resiliency and Flexibility  
Achieved?**

- Encapsulation
  - Classes and objects
- Polymorphism
- Relationships between Classes
  - Inheritance
    - Superclass-Subclass relationship

---

---

---

---

---

---

---

---

**What is a CLASS?**

- A *class* is a computer construct representing a concept bound in a cohesive package
  - Some are concrete (i.e. real world)
    - Bank account
    - Rental item
    - Database item
    - File
  - Others are abstract
    - Scanner
    - Stream

---

---

---

---

---

---

---

---

**Inheritance**

- *Is-a* relationship
- Relationship between a more general class (*superclass*) and a more specialized class (*subclass*)
- Every
  - savings account is a bank account
  - DVD rental is a rental

---

---

---

---

---

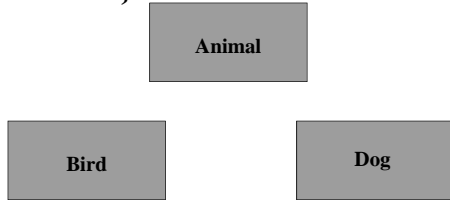
---

---

---

## A little Quiz (cont'd)...

- Class or Object?



The concept of *polymorphism!*

---

---

---

---

---

---

---

---

## Graphical Notations



- Booch
- OMT
- UML



---

---

---

---

---

---

---

---

## For More Information

- To satisfy your quest for more information on this topic you can visit:
- <http://www-st.inf.tu-dresden.de/CoFI/tutorial.pdf>
- [http://www.cs.bris.ac.uk/Teaching/Resources/COMS30203/Rationals\\_intro/quatrani\\_intro\\_UML.pdf](http://www.cs.bris.ac.uk/Teaching/Resources/COMS30203/Rationals_intro/quatrani_intro_UML.pdf)

---

---

---

---

---

---

---

---

## Modeling Languages

- XMI  
XML Metadata Interchange
- XML  
eXtensible Markup Language  
From the lineage of HTML etc.

---

---

---

---

---

---

---

---

## For More Information

- You may wish to visit:  
<http://www.dcs.ed.ac.uk/home/pxs/XMI/tutslides2up.pdf>

---

---

---

---

---

---

---

---

## Object Oriented Design

- How does it fit?



---

---

---

---

---

---

---

---

## Various CASE Diagrams

- Use Case
- Sequence
- Collaboration
- **Class diagrams**
- State-Transition
- Component
- Deployment

---

---

---

---

---

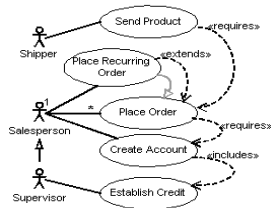
---

---

---

## Just a Few Examples...

- Use Case diagram:




---

---

---

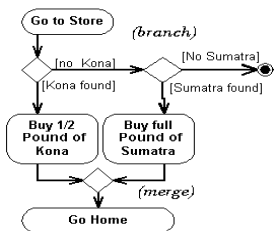
---

---

---

---

---



- State-transition (activity)

---

---

---

---

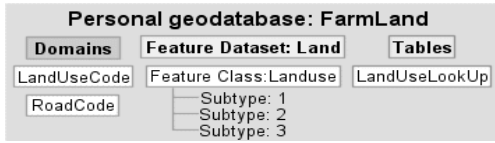
---

---

---

---

- Class diagrams




---

---

---

---

---

---

---

---

## Process



Inception



Elaboration

- Construction
- Transition

---

---

---

---

---

---

---

---

## Elaboration Exercise

- Divide into two task force teams
  - Red team
  - Green team
- Create a list of things that are:
  - Red
  - Yellow
- Brainstorm for 5 minutes




---

---

---

---

---

---

---

---

## Construction

- Use a **Framework**
- Create your visual model
- *(oh yeah...save your model frequently)*



---

---

---

---

---

---

---

---

**Let's check it out...**



---

---

---

---

---

---

---

---

## Questions?

- Your assignment
  - Form task-force teams
  - Use the exercise handout as a guideline to design a relational database

---

---

---

---

---

---

---

---