

# Understanding RDBMS

Keith T. Weber  
GIS Director

ISU GIS Training and Research Center



---

---

---

---

---

---

---

---

## RDBMS

- Relational Database Management System

---

---

---

---

---

---

---

---

## Databases

- The "I" in GIS

---

---

---

---

---

---

---

---

## Database software...

- Light Duty
- Medium Duty
- Heavy Duty

---

---

---

---

---

---

---

---

## Database software...

- Light Duty
- Medium Duty
- Heavy Duty



---

---

---

---

---

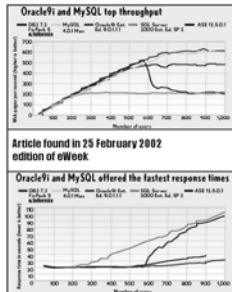
---

---

---

## IBM DB2 UDB

- The GIS Center's heavy hitter- IBM DB2, Universal Database



---

---

---

---

---

---

---

---

## Spreadsheets vs. Databases

- Integrity!
- Structure

---

---

---

---

---

---

---

---

## RDBMS Concepts and Terms

---

---

---

---

---

---

---

---

## Independence

- Physical
- Logical



---

---

---

---

---

---

---

---

## Integrity

- Important for consistency and transaction management.
- Types:
  - Domain
  - Redundancy
  - Constraint
  - Entity
  - Referential
    - Cascading or non-cascading

---

---

---

---

---

---

---

---

## Key Fields



- Unique Identifiers (?)
  - Primary key
  - Foreign key
- AKA- Relate fields.

---

---

---

---

---

---

---

---

## RDBMS Structure

---

---

---

---

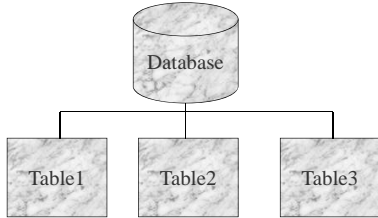
---

---

---

---

## Database Tables




---

---

---

---

---

---

---

---

## Table Structure

	Column 1 (Field 1)	Column 2 (Field 2)
Row 1 (Record 1)	value	
Row 2 (Record 2)		

---

---

---

---

---

---

---

---

## Data Value Types

Type Name	Storage Occupied/ data value	Valid Domain Range
Short Integer	2 bytes	-32768 to 32767
Long Integer	4 bytes	-2147483648 to 2147483647
Float	4 bytes	Any number from $n^{-45}$ to $n^{38}$
Double	8 bytes	Any number from $n^{-124}$ to $n^{108}$
Text (string)	10 + max. length = bytes	Any alphanumeric characters
Date	8 bytes	Jan 1, 100 to Dec. 31 9999
LOB (variant)	22 + max. length = bytes	Any alphanumeric characters

---

---

---

---

---

---

---

---

## 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
- Populate the database

---

---

---

---

---

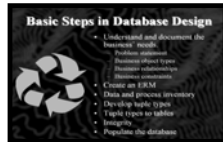
---

---

---

## A Scenario...

- Develop a GIS-Based Tourism database for Southeast Idaho.



---

---

---

---

---

---

---

---

## Document the business needs

- What problem or issue is this database going to address?
- This is a **business statement**



---

---

---

---

---

---

---

---

## The Preliminary ERM

- Symbolized.
  - Standard Representation
  - Attribute Representation
  - Entity Instance Representation

DINING
K Restaurant Number: 126
Name: Burger King
Type of food: Fast



---

---

---

---

---

---

---

---

## Relationships

- Determine the relationships between your entity types.
- Add these to the ERM

---

---

---

---

---

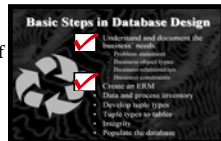
---

---

---

## Define the List

- Database Dictionary
  - Restaurant\_Name • The name of the restaurant
  - Food\_Type • Categories of food (e.g., 1 = Continental, 2 = Fast food, etc.)
  - Cost\_Mean • The average cost of all regular menu items.



---

---

---

---

---

---

---

---

## Develop Tuple Types

- Use your ERM with relationships
- Perform a “Walk-through” exercise
  - Simulate information is being added/used in your database.
- Symbolize using Attribute Representation

---

---

---

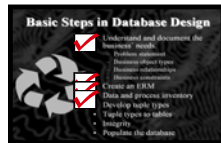
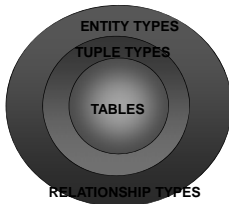
---

---

---

---

## Tuple Types to Tables



---

---

---

---

---

---

---

## Normalization

- First-Fifth Form Normal (1FN, 2FN,...5FN)
- Academic
- Applied

---

---

---

---

---

---

---

### 1FN

- All values are atomic
- Single cell contains single data value

---

---

---

---

---

---

---

### 2FN

- Satisfy 1FN and...
- Redundant data must be eliminated
  - How?
  - Example: Puppy\_ID, Trick\_ID, Trick\_Name

---

---

---

---

---

---

---

### 3FN

- Satisfy 1NF and 2FN and...
- No non-key attributes are dependent on other non-key attributes.
  - Example: Appointment\_ID, Name, Date, Time, Species

---

---

---

---

---

---

---

## After Normalization

- New tuple types will be created.
- New tables will be planned.
- Many-many relationships will be handled using associative tables (bridge tables).



---

---

---

---

---

---

---

---

## De-Normalization

- What? Is this heresy?

---

---

---

---

---

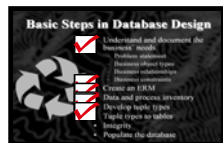
---

---

---

## Designing the Actual RDBMS

- Visual modeling based upon your ERM and Tuple type model.
- Implementation of integrity rules based upon your business constraints.



---

---

---

---

---

---

---

---

## Populate...

- Questions and concerns to revisit
  - Null data
  - Reporting discrepancies and variations
  - Measuring or estimating methods
  - Client utility/efficiency

---

---

---

---

---

---

---

---

## The Last Step?

**Validation!**

---

---

---

---

---

---

---

---

## Questions?

---

---

---

---

---

---

---

---

**Let's take a Break!**

---

---

---

---

---

---

---

---

**Reading a Business Statement**

IT4GIS  
Keith T. Weber, GISP  
GIS Director, ISU



---

---

---

---

---

---

---

---

**Identify Candidate Classes**

- A *candidate* class may or may not remain a class throughout the design process
- A *candidate* class may or may not become a table
- Do not think about tables and relationship classes at this point

---

---

---

---

---

---

---

---

## Think Object-Oriented

- Classes are nouns
- A noun is a “person, places, and things”

**SCHOOLHOUSE ROCK**



---

---

---

---

---

---

---

---

## And now...Verbs

- *Candidate* methods are verbs
  - They show action
  - They are behaviors

**SCHOOLHOUSE ROCK**



---

---

---

---

---

---

---

---

## Methods

- Identifying *candidate* methods allows us to better understand how the business operates and how the Enterprise uses GIS data.
- A method is a behavior...a relationship between classes
- The candidate methods will describe an inheritance, aggregation, or dependency relationship

---

---

---

---

---

---

---

---

### **And Now a Little Practice**

- Complete the quick exercise that is being handed out...
- Bring it to class next week

---

---

---

---

---

---

---

---