Database Basics

Good database design will get you through poor programming **better than good programming** will get you through poor database design….

Relational Database

- Definition:
  - Data stored in tables that are associated by shared attributes (keys).
  - Any data element (or entity) can be found in the database through the name of the table, the attribute name, and the value of the primary key.

Relational Database Definitions

- Entity: Object, Concept or event (subject)
- Attribute: a Characteristic of an entity
- Row or Record: the specific characteristics of one entity
- Table: a collection of records
- Database: a collection of tables

The Relational Database model

- Developed by E.F. Codd, C.J. Date (70s)
- Table = Entity = Relation
- Table row = tuple = instance
- Table column = attribute
- Table linkage by values
- Entity-Relationship Model

The Relational Model

- Each attribute has a unique name within an entity
- All entries in the column are examples of it
- Each row is unique
- Ordering of rows and columns is unimportant
- Each position (tuple) is limited to a single entry.

Data Model: What's a model?

- A data model is a representation of reality
- It's used to define the storage and manipulation of a data base.
- Data Models have two components:
  - Structure: the structure of the data stored within
  - Operations: Facilities for manipulation of the data.

Relational Database Systems

Most popular DBMS model for GIS

Flexible approach to linkages between records comes the closest to modeling the complexity of spatial relationships between objects.

CRUD !

- Refers to the most common Database Operations:
  - Create
  - Read
  - Update
  - Delete
- Operations occur at all levels: Tables, Records, Columns

Database Tables

- Tables represent entities
- Tables are always named in the singular, such as: Vehicle, Order, Grade, etc.
- Tables in database jargon are “flat files”, dBase or Spreadsheet like..

Attributes

- Characteristics of an entity
- Examples:
  - Vehicle (VIN, color, make, model, mileage)
  - Student (SSN, Fname, Lname, Address)
  - Fishing License (Type, Start_date, End_date)

Database Table Example

Students : Table

	Name	DOB	Address	Phone Number
	Adam Glasnost	1/3/84	11 Price Lane, Spline, 2988	(222) 489-2019
	James Smith	5/9/82	27 Lexington Dr, Ester, 2980	(270) 298-0928
	Clarissa Dubai	12/11/82	123 Dress Dr, Cranton, 2987	(222) 494-2222
	James Smith	5/9/82	5/15 Coventry, Spline, 2988	(278) 509-3981
*				

Record: 2 of 4

Figure 1: A simple – and flawed – table design.

Students II : Table

	StudentID	Lastname	Firstname	DOB	Street address	Suburb	Postcode
	89120	Glasnost	Adam	1/3/84	11 Price Lane	Spline	2988
	90109	Dubai	Clarissa	12/11/82	123 Dress Dr	Cranton	2987
	91082	Smith	James	5/9/82	27 Lexington Dr	Ester	2980
	93007	Smith	James	5/9/82	5/15 Coventry	Spline	2988
▶							

Record: 5 of 5

Figure 2: An improved database table..

Database Views

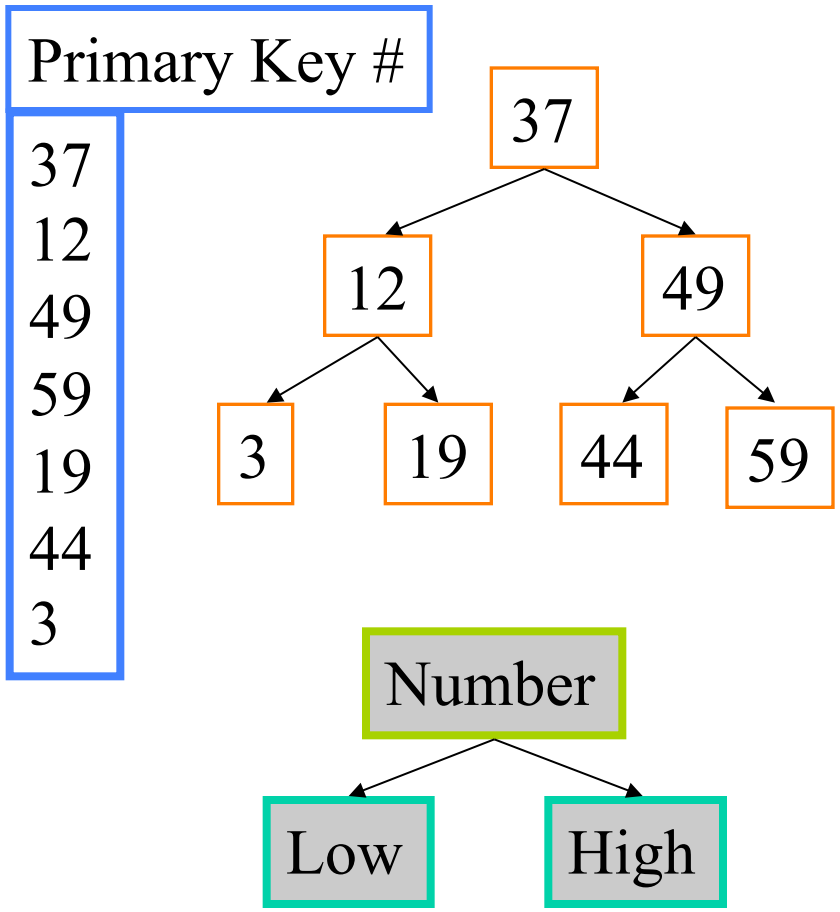
- A View is an individual's picture of a database. It can be composed of many tables, unbeknownst to the user.
  - It's a simplification of a complex data model
  - It provides a measure of database security
  - Views are useful, primarily for READ-only users and are not always safe for CREATE, UPDATE, and DELETE.

Table Indexing

- An Index is a means of expediting the retrieval of data.
- Indexes are “built” on a column(s).
- Indexes occupy disk space; occasionally a lot.
- Indexes aren’t technically necessary for operation and must be maintained by the database administrator.

B-Tree Index Example

- Commonly used with “attribute” tables as well as “graphic-attribute” tables (CAD data structures)
- Binary coding reduces the search list by streaming down the “tree”.
- A “balanced” tree is best.



Database Relationships

- How is one entity related to another entity?
- Real-world sources:
  - Ownership
  - Parentage
  - Assignment
  - Regulation

Database Table Keys

Definition:
A key of a relation is a subset of attributes with the following attributes:

- Unique identification
- Non-redundancy

Types of Keys

PRIMARY KEY

- Serves as the row level addressing mechanism in the relational database model.
- It can be formed through the combination of several items.

FOREIGN KEY

- A column or set of columns within a table that are required to match those of a primary key of a second table.

These keys are used to form a RELATIONAL JOIN - thereby connecting row to row across the individual tables.

Relational Database Management System (RDBMS)

Table A

Name	Address	Parcel #
John Smith	18 Lawyers Dr.	756554
T. Brown	14 Summers Tr.	887419

Table B

Parcel #	Assessed Value
887419	152,000
446397	100,000

Database Keys

- Primary Key - Indicates uniqueness within records or rows in a table.
- Foreign Key - the primary key from another table, this is the **only** way join relationships can be established.
- There may also be *alternate* or *secondary* keys within a table.

Constructing Join Relationships

- One-to-many relationships include the Primary Key of the ‘one’ table and a Foreign Key (FK) in the ‘many’ table.

Other common terms

- Cardinality: one-to-one, one-to-many, many-to-many relationships
- Optionality: the relationship is either mandatory or optional.

Ensuring Database Integrity

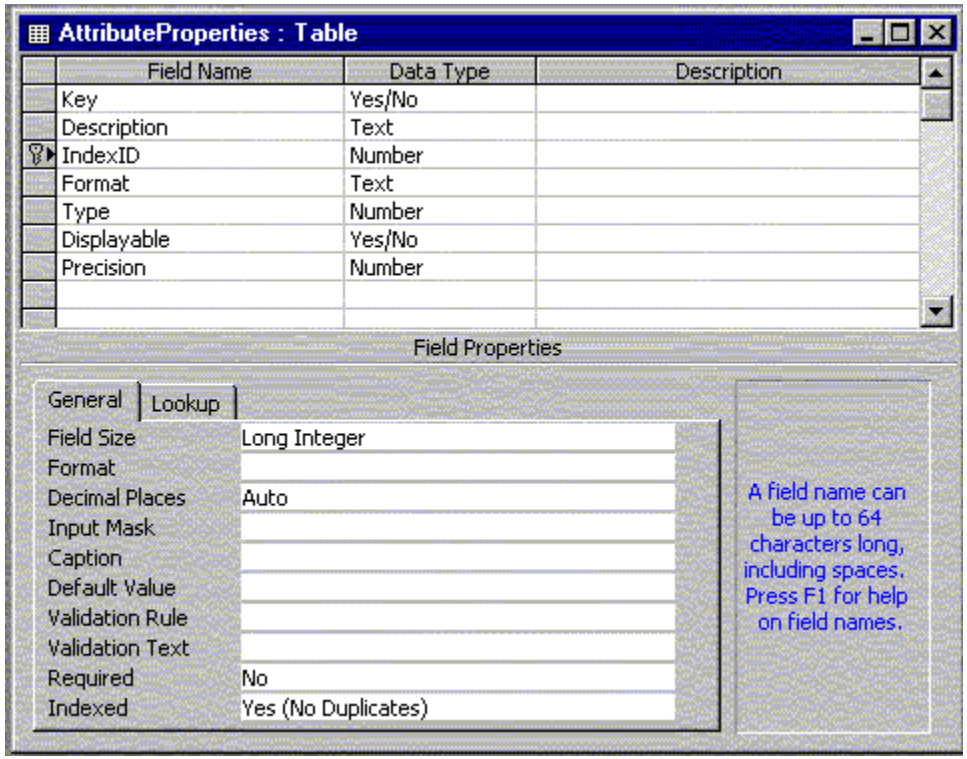
- Database integrity involves the maintenance of the logical and business rules of the database.
- There are two kinds of “DB Integrity” that must be addressed:
  - Entity Integrity
  - Referential Integrity

Strategies for managing Integrity

- You could ignore it, but it costs you time.
- Place the Burden on your customer or user.
- Have the programmers “fix the problem”
- Place the burden on the Database Management System (DBMS)
- Temporal integrity is one of the key challenges of Address Database management.

Entity Integrity

- Entity integrity deals with within-entity rules.
- These rules deal with ranges and the permission of null values in attributes or possibly between records



Examples of Entity Integrity

- Data Type Integrity: very common and most basic. Checks only for "data type" compatibility with DB Schema, such as: numeric, character, logical, date format, etc.
- Commonly referred to in GIS manuals as:
  - Range and List domains
    - Ranges - acceptable Numeric ranges for input
    - List - acceptable text entries or drop-down lists.

Enforcing Integrity

- Not a trivial task!
- Not all database management systems or GIS software enable users to “enforce data integrity” during attribute entry or edit sessions.
- Therefore, the programmer or the Database Administrator must enforce and/or check for “Integrity.”

Referential Integrity

- Referential integrity concerns two or more tables that are related.
- Example: IF **table A** contains a foreign key that matches the primary key of **table B** THEN values of this foreign key either match the value of the primary key for a row in **table B** or must be *null*.

Functions of a Database Management System

- Data Storage, Retrieval and Update (CRUD)
- Catalog or Data Dictionary
- Shared Update Support
- Backup and Recovery Services
- Security Services
- Integrity Services
- Data Independence - independent from programs
- Various Data Manipulation Utilities

CRUD

- Four basic functions, for a given entity they should all be performed with few exceptions, in your system:
  - CREATE
  - READ
  - UPDATE
  - DELETE

Using SQL- Structured Query Language

- SQL is a standard database protocol, adopted by most ‘relational’ databases
- Provides syntax for data:
  - Definition
  - Retrieval
  - Functions (COUNT, SUM, MIN, MAX, etc)
  - Updates and Deletes

SQL Examples

- CREATE TABLE SALESREP
  - Item definition expression(s)
    - {item, type, (width)}

- DELETE table
  - WHERE expression

Data Retrieval

- **SELECT** list **FROM** table **WHERE** condition
- list - a list of items or * for all items
  - WHERE - a logical expression limiting the number of records selected
  - can be combined with Boolean logic: AND, OR, NOT
  - ORDER may be used to format results

UPDATE tables

- SET item = expression
- WHERE expression
- INSERT INTO table
- VALUES …..

Database Normalization

- **Normalization:** The process of structuring data to minimize duplication and inconsistencies.
- The process usually involves breaking down a single Table into two or more tables and defining relationships between those tables.
- Normalization is usually done in stages, with each stage applying more rigorous rules to the types of information which can be stored in a table.

Normalization

- Normalization: a process for analyzing the design of a relational database
  - Database Design - Arrangement of attributes into entities
- It permits the identification of potential problems in your database design
- Concepts related to Normalization:
  - KEYS and FUNCTIONAL DEPENDENCE

Ex: Database Normalization (1)

- Sample Student Activities DB Table
- Poorly Designed
  - Non-unique records
    - John Smith
- Test the Design by developing sample reports and queries

Activities Table

Student	Activity1	Cost1	Activity2	Cost2
John Smith	Tennis	$36	Swimming	$17
Jane Bloggs	Squash	$40	Swimming	$17
John Smith	Tennis	$36		
Mark Antony	Swimming	$15	Golf	$47

Ex: Database Normalization (2)

- Created a unique "ID" for each Record in the Activities Table
- Required the creation of an "ID" look-up table for reporting (Students Table)
- Converted the "Flat-File into a Relational Database

Students Table

Student	ID*
John Smith	084
Jane Bloggs	100
John Smith	182
Mark Antony	219

Activities Table

ID*	Activity1	Cost1	Activity2	Cost2
084	Tennis	$36	Swimming	$17
100	Squash	$40	Swimming	$17
182	Tennis	$36		
219	Swimming	$15	Golf	$47

Ex: Database Normalization (3)

- Wasted Space
- Redundant data entry
- What about taking a 3rd Activity?
- Query Difficulties - trying to find all swimmers
- Data Inconsistencies - conflicting prices

Students Table

Student	ID*
John Smith	084
Jane Bloggs	100
John Smith	182
Mark Antony	219

Activities Table

ID*	Activity1	Cost1	Activity2	Cost2
084	Tennis	$36	Swimming	$17
100	Squash	$40	Swimming	$17
182	Tennis	$36		
219	Swimming	$15	Golf	$47

Ex: Database Normalization (4)

- Students table is fine
- Elimination of two columns and an Activities Table restructuring, Simplifies the Table
- **BUT**, we still have Redundant data (activity fees) and data insertion anomalies.

Students Table

Student	ID*
John Smith	084
Jane Bloggs	100
John Smith	182
Mark Antony	219

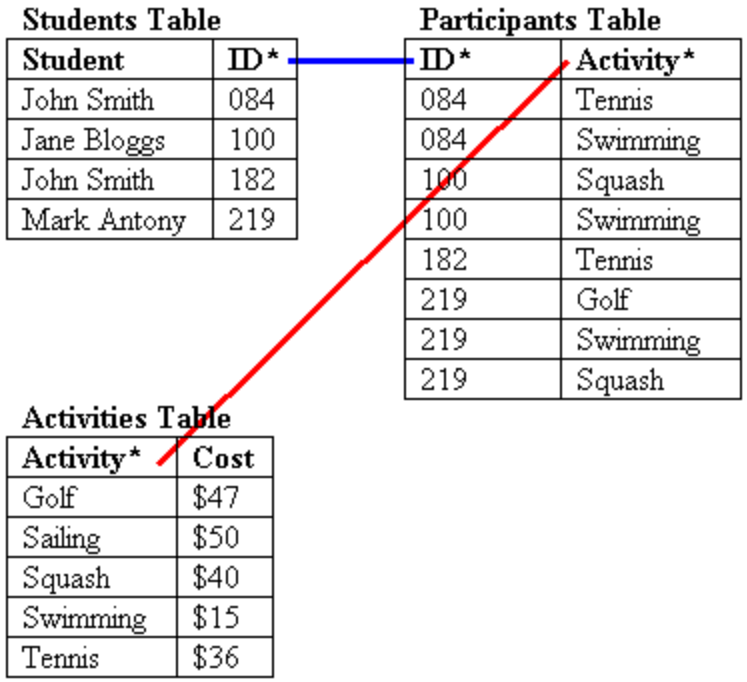
Activities Table

ID*	Activity*	Cost
084	Swimming	$17
084	Tennis	$36
100	Squash	$40
100	Swimming	$17
182	Tennis	$36
219	Golf	$47
219	Swimming	$15
219	Squash	$40

Problem: If student #219 transfers we lose all references to Golf and its price.

Ex: Database Normalization (5)

- Modify the Design to ensure that "every non-key field is dependent on the whole key"
- Creation of the Participants Table, corrects our problems and forms a union between 2 tables.



Students Table

Student	ID*
John Smith	084
Jane Bloggs	100
John Smith	182
Mark Antony	219

Participants Table

ID*	Activity*
084	Tennis
084	Swimming
100	Squash
100	Swimming
182	Tennis
219	Golf
219	Swimming
219	Squash

Activities Table

Activity*	Cost
Golf	$47
Sailing	$50
Squash	$40
Swimming	$15
Tennis	$36

This is a Better Design!

Database Design: Basic Steps

- Step 1: Determine the entities involved and create a separate table for each type of entity (thing, concept, event, theme) and name it.
- Step 2: Determine the Primary Key for each table.
- Step 3: Determine the properties for each entity (the non-key attributes).
- Step 4: Determine the relationships among the entities

Design Example: Music CD collection

- Entities: the CD, Tracks, Composer
- Attributes:
  - CD (ID, title, musician, cost, etc.)
  - Track (song title, length, order number)
  - Composer (name, genre, DOB, DOD)
- Relationships: CD to Track, Composer to Track

Table Design Example

Students : Table

	Name	DOB	Address	Phone Number
	Adam Glasnost	1/3/84	11 Price Lane, Spline, 2988	(222) 489-2019
	James Smith	5/9/82	27 Lexington Dr, Ester, 2980	(270) 298-0928
	Clarissa Dubai	12/11/82	123 Dress Dr, Cranton, 2987	(222) 494-2222
	James Smith	5/9/82	5/15 Coventry, Spline, 2988	(278) 509-3981
*				

Record: 2 of 4

Figure 1: A simple – and flawed – table design.

Students II : Table

	StudentID	Lastname	Firstname	DOB	Street address	Suburb	Postcode
	89120	Glasnost	Adam	1/3/84	11 Price Lane	Spline	2988
	90109	Dubai	Clarissa	12/11/82	123 Dress Dr	Cranton	2987
	91082	Smith	James	5/9/82	27 Lexington Dr	Ester	2980
	93007	Smith	James	5/9/82	5/15 Coventry	Spline	2988
▶							

Record: 5 of 5

Figure 2: An improved database table..

Step1: Creating a Data Model

- Identify Candidate Entities
- Identify Relationships
- Define Entities & Relationships
- Review Entity-Relationship Model

Step 2: Defining an Attribute Model

- List Candidate Attributes for each Entity
- Add KEYS to model
- Attribute & Normalize Model
- Define Attributes
- Review Logical Model

Step 3: Identify & Capture Business Rules

- Review & Verify Cardinalities
- Define Referential Integrity
- Identify Business Domains
- Identify Attribute Default Values

Step 4: Define Physical Model

- Select Target DBMS
- Name Tables & Columns
- Name & Define Indexes
- Define Columns
- Verify/Update Triggers
- Generate Reports & Document Design

Step 5: Review Final Design

- Verify Entities & Definitions
- Verify Relationships & Definitions
- Verify Attributes & Definitions
- Verify Business Constraints
- Approve Schema Design

A Review of the Advantages of Database Processing

- Lower cost… (relative it what?)
- More Information from same amount of data
- Data Sharing is easier
- Controlled or elimination of redundancy
- Consistency, Integrity, Security
- Increased Productivity

Some Disadvantage of Database Processing

- Greater Complexity
- Possibly a greater impact of a failure
- Recovery is more difficult
- Although these are all debated issues, opportunities for complete failure are often reduced with the latest database products, but reliability results in higher investment costs.