The Two Types of Data in GIS

**Spatial data**: Describing *where* things are

**AND**

**Attribute data**: Describing *what* things are

• Example: A point specified by UTM coordinates
  • Easting = 50,000 m
  • Northing = 5,000,000 m
  • Zone = 17

• This specifies the **location** of a point of the ground
• The nature of the real-world feature located at this point would be recorded in the **attribute data**

• Traditionally, geographic data and attributes were recorded on paper too (maps), and these had the same problems as a phone book
GIS Database Models

• The approaches used for constructing GIS database management systems have depended upon the development of DBMS in computer science. This dates back to the 1970s when data entry used punch cards, and it has come a long way since then …

• The first successful GIS Arc/INFO was really the marriage of two separate components:
  • The Arc spatial data processing component
  • The INFO relational database management system
Some Database Definitions

• **Database** – an integrated set of data on a particular subject

• **Geographic (~spatial) database** - database containing geographic data of a particular subject for a particular area

• **Database Management System (DBMS)** – software to create, maintain and access databases
Advantages of Databases over Files

• DBs avoid *redundancy* and duplication
• DBs reduce data maintenance *costs*
• Applications are separated from the data
  – Applications *persist* over time
  – Support multiple concurrent applications
• DBs facilitate better *data sharing*
• Security and standards can be defined and enforced using DBs
Disadvantages of Databases over Files

- **Expense** of databases
- **Complexity** of databases
- **Performance** of databases – especially with complex data types (including spatial data)
- **Integration** with other systems can be difficult, especially if those systems don’t use the same data model
Characteristics of DBMS (1)

- Data model support for **multiple data types**
  - e.g. MS Access: Text, Memo, Number, Date/Time, Currency, AutoNumber, Yes/No, OLE Object, Hyperlink, Lookup Wizard
- **Load data** from files, databases and other applications
- **Indexed** for rapid retrieval
Characteristics of DBMS (2)

- Query language – SQL provides a **structured way** to ask questions of the data
- Security – **controlled access** to data
  - Multi-level groups etc.
- Controlled **update** using a transaction manager manages the updating process
- **Backup and recovery** of data for when the unthinkable happens …
- DBA tools for optimizing performance
  - Configuration, tuning
Characteristics of DBMS (3)

• Applications that help make good use the data
  – CASE tools to help in **formalizing logic**
  – Forms builder to make **structured input** easy
  – Report writer to create **customized output**
  – Internet Application Server to serve data on the Web

• Programmable API (application programming interface) to allow **customization**
The Role of DBMS in GIS

System

Geographic Information System

Database Management System

Data

Task

• Data load
• Editing
• Visualization
• Mapping
• Analysis

• Storage
• Indexing
• Security
• Query
Types of DBMS Models

- Hierarchical
- Network
- Relational - RDBMS
- Object-oriented - OODBMS
- Object-relational - ORDBMS

Our focus
Hierarchical Data Model

• Suppose we are designing a model for faculty & staff data:
Hierarchical Data Model

• Now, suppose we are creating a model for places in the USA.
Disadvantages of the Hierarchical Data Model

- The database is defined in terms of a tree structure which is inflexible and has trouble dealing with exceptions (i.e. all records need to follow the same uniform structure):

  1. We **cannot define new linkages** between records once the hierarchical tree is established.
  2. We **cannot define linkages laterally or diagonally** in the tree, only vertically.
  3. The only geographical relationships which can be **encoded easily** are “is contained within” or “belongs to”.
Relational Data Model

The **relational model** organizes data in a series of two-dimensional tables, each of which contains records for one kind of entity.

<table>
<thead>
<tr>
<th>Fields</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PID #</td>
<td>Name</td>
<td>Major</td>
<td>Phone #</td>
<td>…</td>
</tr>
<tr>
<td>1010789</td>
<td>John</td>
<td>Geog.</td>
<td>555-4321</td>
<td>…</td>
</tr>
<tr>
<td>1021384</td>
<td>David</td>
<td>Comm.</td>
<td>555-6789</td>
<td>…</td>
</tr>
</tbody>
</table>

This model is a **revolution** in database management. It replaced almost all other approaches in database management because it allows more **flexible relations** between kinds of entities.
Relational DBMS (1)

• In a RDBMS, data is stored as **tuples** (pronounced tup-el), and is conceptualized as tables

• **Table** – contains data about a class of objects
  – Two-dimensional list (array)
  – Rows = objects
  – Columns = object states (properties, attributes)
A Table

Row = object

Column = property

Table = Object Class

Object classes which have Geometry encoded are called feature classes
Relational DBMS (2)

• **Most popular** type of DBMS
  – Over 95% of data in DBMS is in RDBMS

• Commercial systems:
  – IBM DB2
  – Informix
  – Microsoft Access
  – Microsoft SQL Server
  – Oracle
  – Sybase
Relation Rules (Codd, 1970)

- Only one value in each cell (intersection of row and column)
- All values in a column are about the same subject
- Each row is unique
- No significance in column sequence
- No significance in row sequence
Normalization

• This is the process of converting tables to conform to Codd’s relational rules

• Split tables into new tables that can be joined at query time
  – The relational join

• Several levels of normalization
  – Forms: 1NF, 2NF, 3NF, etc.

• Normalization creates many expensive joins

• De-normalization is OK for performance optimization
We use the relational join operation because
- We are using tables that have been transformed by normalization
- Data created/maintained by different users, but integration needed for queries
- We want to combine data to ask questions that can only be answered by using the data together

Table joins use common keys (column values) filled with the same identifiers

The table (attribute) join concept has been extended to geographic cases
Relational Join

• Take two tables full of different data (e.g. registrar info & parking data), and *join* them:

```
<table>
<thead>
<tr>
<th>PID #</th>
<th>Name</th>
<th>Major</th>
<th>Phone #</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1010789</td>
<td>John D.</td>
<td>Geog</td>
<td>555-4321</td>
<td></td>
</tr>
<tr>
<td>1021384</td>
<td>David Q.</td>
<td>Comm</td>
<td>555-6789</td>
<td></td>
</tr>
</tbody>
</table>
```

```
<table>
<thead>
<tr>
<th>PID #</th>
<th>Name</th>
<th>Parking Lot</th>
<th>License Plate</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1010789</td>
<td>John D.</td>
<td>Friday Ctr.</td>
<td>PNT3465</td>
<td></td>
</tr>
<tr>
<td>1021384</td>
<td>David Q.</td>
<td>Airport Rd.</td>
<td>JRS4089</td>
<td></td>
</tr>
</tbody>
</table>
```

The tables are joined through a **common key** which has a unique value for each record.
Spatial Relations

• In addition to relations that join tables based on an identical common key, we can evaluate relations between the spatial characteristics of features:
  • Equals – same geometries
  • Disjoint – geometries share common point
  • Intersects – geometries intersect
  • Touches – geometries intersect at common boundary
  • Crosses – geometries overlap
  • Within – geometry within
  • Contains – geometry completely contains
  • Overlaps – geometries of same dimension overlap
  • Relate – intersection between interior, boundary or exterior
## Contains Relation

<table>
<thead>
<tr>
<th>Comparison Geometry</th>
<th>Base Geometry</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Comparison Geometry" /></td>
<td><img src="image" alt="Base Geometry" /></td>
<td>No containment relationship possible</td>
</tr>
<tr>
<td><img src="image" alt="Comparison Geometry" /></td>
<td><img src="image" alt="Base Geometry" /></td>
<td>No containment relationship possible</td>
</tr>
<tr>
<td><img src="image" alt="Comparison Geometry" /></td>
<td><img src="image" alt="Base Geometry" /></td>
<td>No containment relationship possible</td>
</tr>
</tbody>
</table>
# Touches Relation

<table>
<thead>
<tr>
<th>Base Geometry</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="No touch relation possible" /></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Comparison Geometry</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="No touch relation possible" /></td>
<td><img src="image" alt="No touch relation possible" /></td>
<td><img src="image" alt="No touch relation possible" /></td>
</tr>
<tr>
<td><img src="image" alt="No touch relation possible" /></td>
<td><img src="image" alt="No touch relation possible" /></td>
<td><img src="image" alt="No touch relation possible" /></td>
</tr>
<tr>
<td><img src="image" alt="No touch relation possible" /></td>
<td><img src="image" alt="No touch relation possible" /></td>
<td><img src="image" alt="No touch relation possible" /></td>
</tr>
<tr>
<td><img src="image" alt="No touch relation possible" /></td>
<td><img src="image" alt="No touch relation possible" /></td>
<td><img src="image" alt="No touch relation possible" /></td>
</tr>
<tr>
<td><img src="image" alt="No touch relation possible" /></td>
<td><img src="image" alt="No touch relation possible" /></td>
<td><img src="image" alt="No touch relation possible" /></td>
</tr>
<tr>
<td><img src="image" alt="No touch relation possible" /></td>
<td><img src="image" alt="No touch relation possible" /></td>
<td><img src="image" alt="No touch relation possible" /></td>
</tr>
<tr>
<td><img src="image" alt="No touch relation possible" /></td>
<td><img src="image" alt="No touch relation possible" /></td>
<td><img src="image" alt="No touch relation possible" /></td>
</tr>
</tbody>
</table>

David Tenenbaum – GEOG 070 – UNC-CH Spring 2005
Spatial Methods

• We can also take features from two tables, and use them to calculate distances, or new features that are formed by operating a **method** on the features:
  • **Distance** – shortest distance
  • **Buffer** – geometric buffer
  • **ConvexHull** – smallest convex polygon geometry
  • **Intersection** – points common to two geometries
  • **Union** – all points in geometries
  • **Difference** – points different between two geometries
  • **SymDifference** – points in either, but not both of input geometries
Convex Hull and Difference Methods
Indexing

• Used to locate rows quickly, speed up access
• RDBMS use simple 1-d indexing
• Spatial DBMS need 2-d, hierarchical indexing to allow features in a given vicinity to be found quickly, using a variety of methods:
  – Grid
  – Quadtree
  – R-tree
  – Others
• Hierarchical in the sense that multi-level queries are often used for better performance
Grid Index (multi-level)
Point and Region Quadtrees
R-tree

Branching factor $M = 4$
Minimum Bounding Rectangle