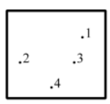
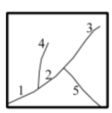
Attribute Data Input & Management

- Georelational & object-oriented data models
  - Geometry + attributes
  - Objects (including geometric and attribute objects)
- (R)DBMS
  - INFO, MS Access, Oracle, Informix, SYBASE…

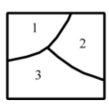
Geometry and Attribute Data



Point-id	Field 1	Field 2	. . .
1			
2			
3			
4			



Line-id	Field 1	Field 2	. . .
1			
2			
3			
4			
5			



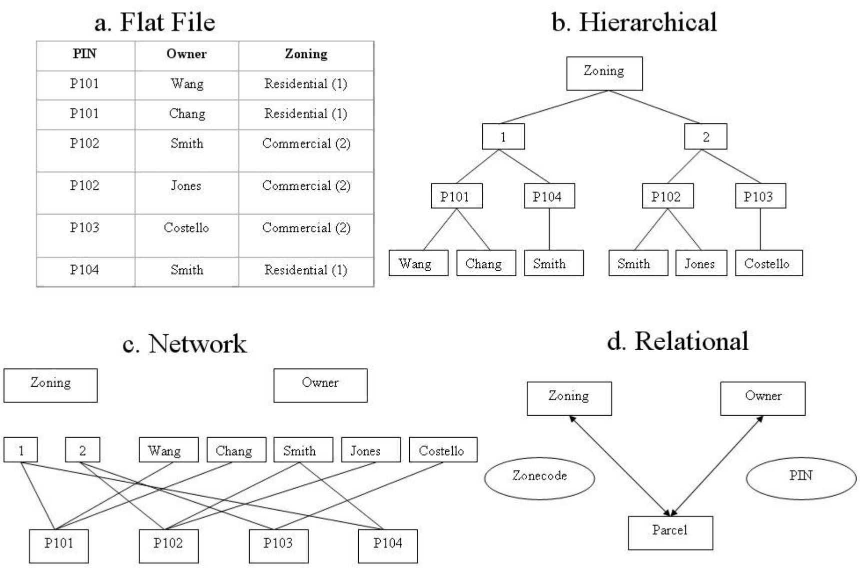
Polygon-id	Field 1	Field 2	. . .
1			
2			
3			

Types of Attribute Data

- Categorical data
  - Nominal
  - Ordinal
- Numerical data
  - Interval
  - Ratio
- Data types
  - Integer, float, double, string…
- Measurement scale and cartographic symbology
- Selection of data type for attribute data
  - Measurement scale, efficiency, applications

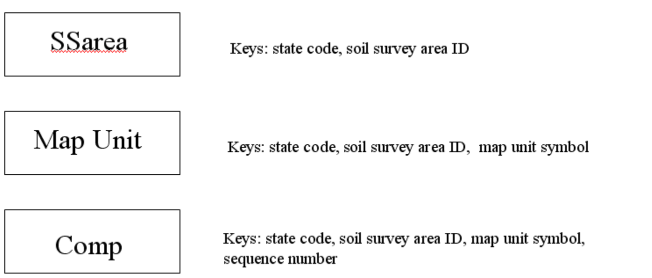
Pay Attention to Data!!!

Database Structure



Relational Database

- Keys
  - **Primary** key: identifies each record in a table.
  - **Foreign** key: an attribute that completes a relationship by identifying the parent entity.
  - **Composite** key: a primary that made up of more than one attribute.
    - NRCS Soil Attribute Data: Map Unit Interpretation Record (MUIR)



DB Normalization

- Avoid redundancy
- Improve DB maintenance efficiency
- Maintain entity independence

Database Normalization

- 1NF

TABLE 6.2 First step in normalization

PIN	Owner	Owner address	Sale date	Acres	Zone code	Zoning
P101	Wang	101 Oak St	1-10-98	1.0	1	residential
P101	Chang	200 Maple St	1-10-98	1.0	1	residential
P102	Smith	300 Spruce Rd	10-6-68	3.0	2	commercial
P102	Jones	105 Ash St	10-6-68	3.0	2	commercial
P103	Costello	206 Elm St	3-7-97	2.5	2	commercial
P104	Smith	300 Spruce Rd	7-30-78	1.0	1	residential

- 2NF (has no composite keys)

Parcel table

PIN	Sale date	Acres	Zone code	Zoning
P101	1-10-98	1.0	1	residential
P102	10-6-68	3.0	2	commercial
P103	3-7-97	2.5	2	commercial
P104	7-30-78	1.0	1	residential

Address table

Owner name	Owner address
Wang	101 Oak St
Chang	200 Maple St
Jones	105 Ash St
Smith	300 Spruce Rd
Costello	206 Elm St

Owner table

PIN	Owner
P101	Wang
P101	Chang
P102	Smith
P102	Jones
P103	Costello
P104	Smith

Parcel Table

PIN	Sale date	Acres	Zone code
P101	1-10-98	1.0	1
P102	10-6-68	3.0	2
P103	3-7-97	2.5	2
P104	7-30-78	1.0	1

Owner Table

PIN	Owner
P101	Wang
P101	Chang
P102	Smith
P102	Jones
P103	Costello
P104	Smith

Address Table

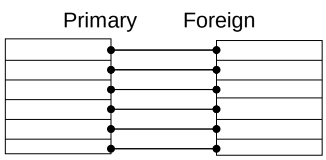
Owner name	Owner address
Wang	101 Oak St
Chang	200 Maple St
Jones	105 Ash St
Smith	300 Spruce Rd
Costello	206 Elm St

Zone Table

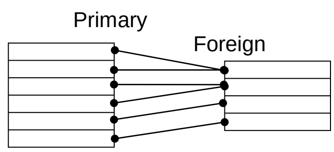
Zone code	Zoning
1	residential
2	commercial

Separate tables after normalization.

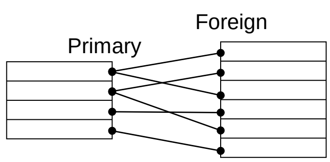
Types of Data Relationship (cardinalities)



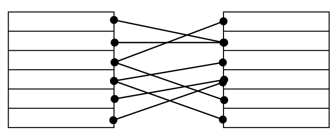
One-to-one relationship



Many-to-one relationship



One-to-many relationship

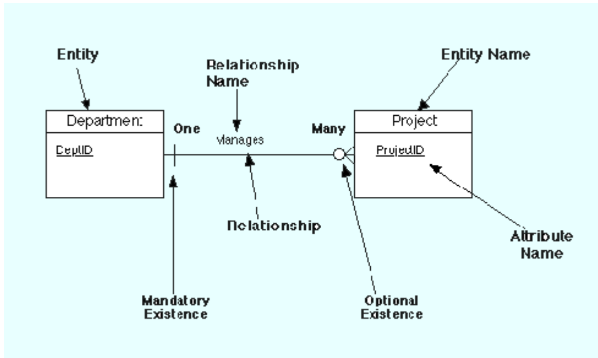


Many-to-many relationship

- Type of relationship and data display

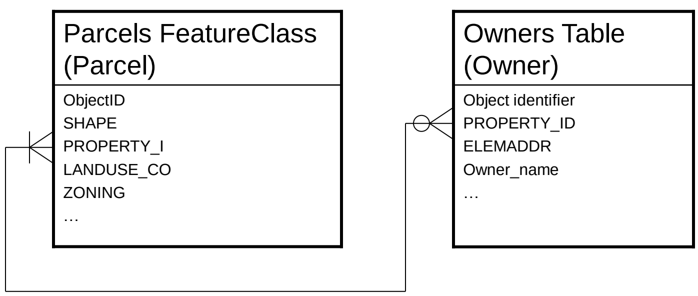
ER Diagram Notation

- Entity – Relation – Attribute
- Cardinality (one, many)
- Existance (optional, mandatory)



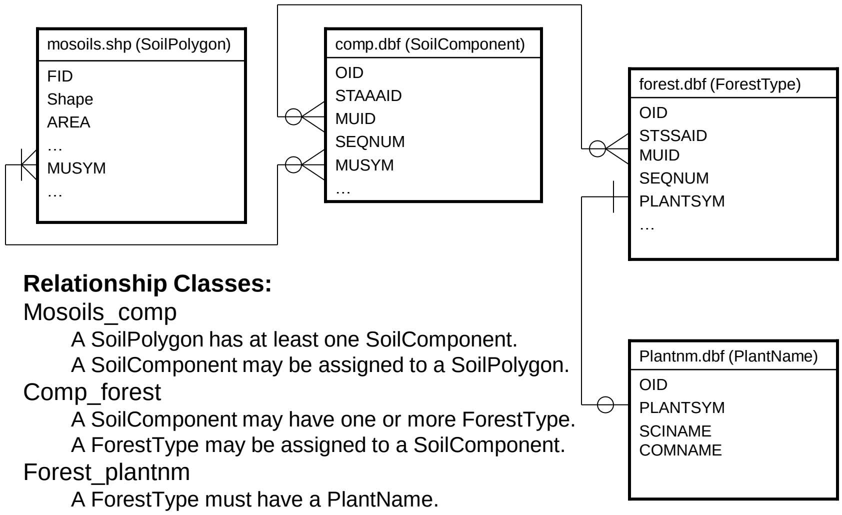
- http://www.utexas.edu/its/windows/database/datamodeling/dm/erintro.html

Parcels & Owners



- An Owner may own one or more Pacels.
- A Pacel is owned by at least one Owner.

SURRGO & MUIR



Relationship Classes:

Mosoils_comp

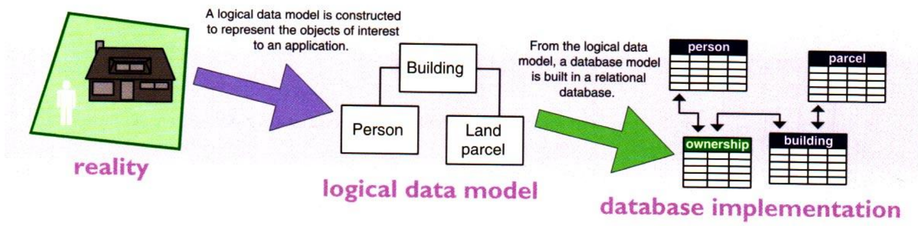
A SoilPolygon has at least one SoilComponent.
A SoilComponent may be assigned to a SoilPolygon.

Comp_forest

A SoilComponent may have one or more ForestType.
A ForestType may be assigned to a SoilComponent.

Forest_plantnm

A ForestType must have a PlantName.
A PlantName may be assigned to a ForestType.



- Reference:

  Introduction to Data Modeling

  http://www.utexas.edu/its-archive/windows/database/datamodeling/index.html

Exercise

Wednesday, April 30th

Conference Program

~1~

Wetland Education through Maps and Aerial Photography (WET-MAPP) — Seminar for Educators

Moderator: Lawrence R. Handley, *U.S. Geological Survey, National Wetlands Research Center*

Catherine Lockwood, *Chadron State College*

9:15 am to 5:00 pm

Room A103

This seminar is designed for educators to increase their ability to promote student awareness of and interest in wetland issues. This seminar will explore wetlands using aerial photography, satellite imagery, and wetland maps, and will introduce traditional mapping techniques into the classroom.

Technical Sessions
9:15 am to 10:45 am

~2~

Soil Applications

Moderator: Erik Strandhagen, GISP, *Integral Consulting Inc.*

Room B118

Modeling Bare Soil Exposure in a Semiarid Ecosystem using Remote Sensing and Geographic Information Systems

Jacob Tibbitts, *Idaho State University GIS Training and Research Center*

Nancy Glenn and Keith Weber

Soil Erosion Calculation using Remote Sensing and GIS

Alejandra M. Rojas-González, *University of Puerto Rico*

Modeling and Analyzing Mass and Volume of DDx Contamination in Sediment for Environmental Remediation

Erik Strandhagen, GISP, *Integral Consulting Inc.*

David G. Livermore, R.G. and Eron Dodak, R.G.

~5~

Lidar — General I

Moderator: Ralph A. Haugerud, *U.S. Geological Survey*

Room B112

A Consumer's Perspective on Lidar Data Quality

Ralph A. Haugerud, *U.S. Geological Survey*

Diana Martinez

Toward Automatic Generation of Digital True Orthophoto by using Dense Lidar Data

Keinan Eran, *Mapping and Geo-Information Engineering, Faculty of Civil and Environmental Engineering*, Israel

Doytsher Yerach

~6~

Hyperspectral Remote Sensing

Moderator: Kaiguang Zhao, *Spatial Sciences Lab*

Room B110

Using a Modified Gaussian Model to Predict Concentrations of Blue-green Algal Pigments in Eutrophic Indiana Reservoirs

Anthony Robertson, *Department of Earth Sciences, Indiana University - Purdue University*

Lin Li, Lenore Tedesco, Jeffrey Wilson, and Emmanuel Soyeux

Bayesian Learning with Gaussian Processes for Classification of Hyperspectral Data

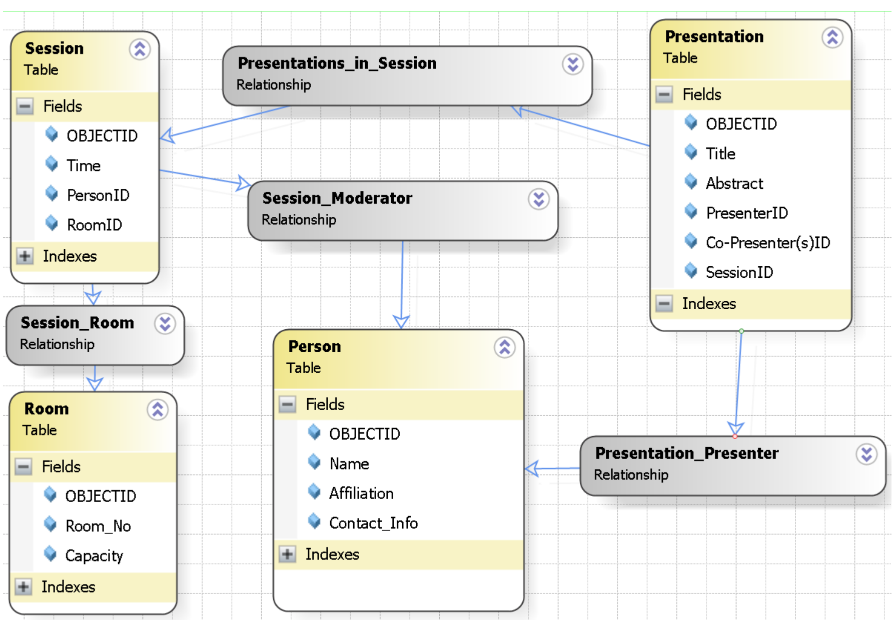
Kaiguang Zhao, *Spatial Sciences Lab*

Sorin Popescu and Xuesong Zhang

Estimating Spatial Variations in Soil Organic Carbon using Hyperspectral Data and Map Algebra

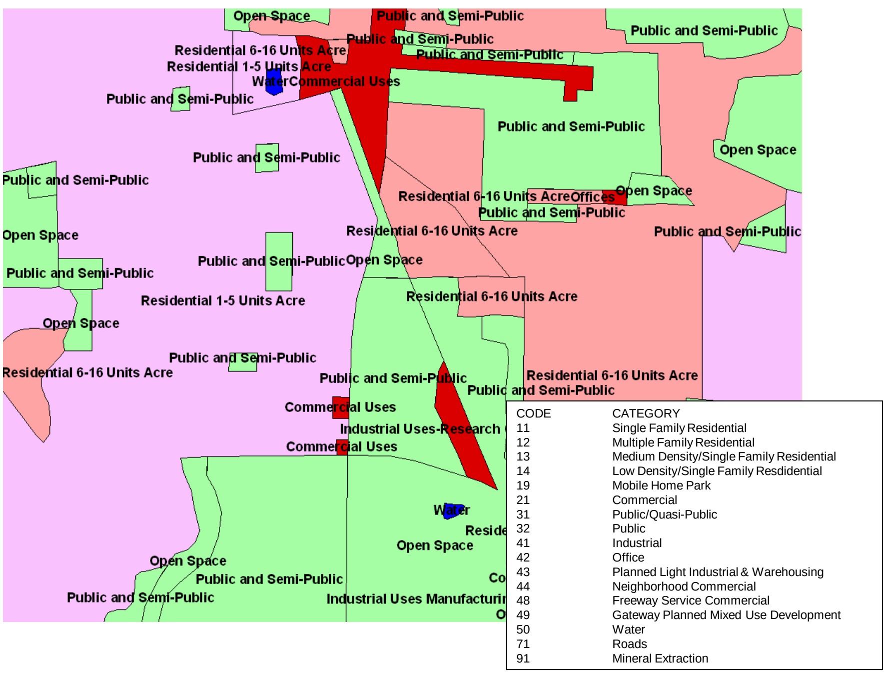
Salahuddin M. Jaber, *Department of Water Management and Environment, Hashemite University*, Jordan

Christopher L. Lant



Attribute Data Entry

- Data entry: type, OCR, import, data logger
  - Verification
    - Visual inspection
    - Validation rules (attribute domain)
    - Cross-checking
- Data classification
  - Simplification, isolation, ranking
- Data computation



Data Exploration

- Exploratory data analysis
  - Descriptive statistics
  - Graphs
  - Dynamic graphics (brushing)
  - Spatial data exploration
- Data query
  - Boolean expressions and Boolean connectors
  - Subset manipulation
  - SQL Select

SQL Select

- SELECT *fieldlist*
  FROM *table*
  [WHERE *selectcriteria*]
  [GROUP BY *groupfieldlist*]
  [HAVING *groupcriteria*]
  [ORDER BY*fieldlist*]

```
SELECT *
FROM Employees;
```

```
SELECT LastName, Salary
FROM Employees
WHERE Salary > 21000;
```

SQL Select (cont.)

```
SELECT *
FROM Orders
WHERE ShippedDate = DateValue('5/10/96');

SELECT COUNT(EmployeeID)
AS HeadCount FROM Employees;

SELECT CategoryID,
Sum(UnitsInStock)
FROM Products
GROUP BY CategoryID
HAVING Sum(UnitsInStock) > 100;
```

Join Tables in SQL

```
SELECT Employees.Department, Supervisors.SupvName
FROM Employees, Supervisors
WHERE Employees.Department = Supervisors.Department;
```

- **INNER JOIN**
- **OUTER JOIN (LEFT JOIN, RIGHT JOIN)**

Employees Table

ID	Department	Age
1	B	40
2	J	38
3	K	29
4	L	55

Supervisors Table

SupvName	Department	Building
Brown	B	CH
Johnson	J	USB
Smith	S	SB1
Young	Y	SB1

INNER JOIN

Department	SupvName
B	Brown
J	Johnson

LEFT JOIN

Department	SupvName
B	Brown
J	Johnson
K	
L	

RIGHT JOIN

Department	SupvName
B	Brown
J	Johnson
	Smith
	Young

Inner Join, Left & Right Joins

```
SELECT field1, field2, field3
FROM first_table
INNER JOIN second_table
ON first_table.keyfield = second_table.foreign_keyfield;


SELECT field1, field2, field3
FROM first_table
LEFT JOIN second_table
ON first_table.keyfield = second_table.foreign_keyfield;


SELECT field1, field2, field3
FROM first_table
RIGHT JOIN second_table
ON first_table.keyfield = second_table.foreign_keyfield;
```

Join and Relate

- Join: 1-1, M-1
- Relate: 1-1, M-1, 1-M, M-M
- Relationship class: a Relate defined in a Geodatabase instead of in a map.

Spatial Database: ST_Gemetry storage type

- Geodatabases in DB2, Informix, Oracle, or PostgreSQL use the ST_Geometry SQL data type. This data type can be used within the geodatabase. It provides spatially-enabled SQL access to feature class geometry for third-party applications.
- ST_Geometry implements the OGC and ISO SQL Multimedia Specification for Spatial. (The OGC reference is OpenGIS Implementation Specification for Geographic information - Simple feature access - Part 2: SQL option. The ISO reference is ISO/IEC 13249-3 SQL multimedia and application packages - Part 3: Spatial.)
- Using SQL functions with ST_Geometry
  - test spatial relationships
  - perform spatial operations
  - return properties of a geometry
  - create spatial data or perform spatial transformations

ST_Geometry SQL Functions: Test Spatial Relationships

- ST_Contains
- ST_Crosses
- ST_Disjoint
- ST_Equals
- ST_Intersects
- ST_Overlaps
- ST_Relate
- ST_Touches
- ST_Within

Example

- Tables:
  - bfp (building_id int, footprint st_geometry);
  - lots (lot_id int, lot st_geometry);
- SELECT DISTINCT (building_id) FROM bfp, lots WHERE st_intersects (lot, footprint) = 't' AND st_contains (lot, footprint) = 'f';

ST_Geometry SQL Functions: Perform Spatial Operations

- ST_Buffer
- ST_ConvexHull
- ST_Difference
- ST_Intersection
- ST_SymmetricDiff
- ST_Distance

Example

- Tables:
  - sensitive_areas (id int, zone st_geometry);
  - hazardous_sites (id int, location st_geometry);

- SELECT sa.id, hs.id, st_area (st_union (st_buffer (hs.location, .01), sa.zone)) FROM hazardous_sites hs, sensitive_areas sa;

ST_Geometry SQL Functions: Return Properties of a Geometry

- ST_Area
- ST_EqualSRS (PostgreSQL only)
- ST_Is3d (Oracle only)
- ST_IsClosed
- ST_IsEmpty
- ST_IsMeasured (Oracle only)
- ST_IsRing
- ST_IsSimple
- ST_OrderingEquals

ST_Geometry SQL Functions: Create Data or Perform Transformations

- ST_PointFromText
- ST_Polygon
- ST_Geometry
- ST_Point
- …

Example

- Table:
  - Point_test (pt1 st_geometry);

- INSERT INTO point_test VALUES (st_point (10.01, 20.03, 0));