RE-THINKING THE DATA MODEL: GIS AS/IS A RELATIONAL DATABASE

Bill Hazelton & Yitong Wu, Troy University Geospatial Informatics

ABOUT US

- Bill Hazelton
  - Surveyor for 40+ years and licensed 35 years
  - Educator and researcher for over 30 years
  - Worked as a surveyor on 3 continents
  - Professor at Troy University
- Yitong Wu
  - Chancellor’s Fellow at Troy University
  - Student at Troy University

GIS HISTORY

- The earliest GIS were cell-based systems
  - Attribute data were stored in a gridded system
  - Location was a function of location in the grid
  - Separation of attributes from the spatial information was minimal
- The introduction of vector-based systems necessitated a hard separation of spatial and attribute data
  - The vector data were stored in one format, often a network database
  - The attribute data were stored in a separate database, commonly relational
- It was possible to store vector topological data in a relational form, but it was very inefficient to run

RELATIONAL STORAGE OF VECTOR-BASED SPATIAL DATA

- This model of topological vector-based spatial data is commonly taught in GIS courses
- It is a useful way to think about the structure of the data on a conceptual level
- But almost all vector-based GIS use a form of network database model to store the spatial data, simply for efficient operation

Lines Table

Line_ID*	From_Pt	To_Pt	Left_Poly	Right_Poly
472	31	87	6194	7073
622	54	22	3008	7073
582	54	87	7073	2247
315	31	22	7073	9462

Points Table

Point_ID*	X	Y
31	xxxxx	xxxxx
54	xxxxx	xxxxx
22	xxxxx	xxxxx
87	xxxxx	xxxxx

Polygons Table

Poly_ID*	Line_ID*
7073	472
7073	–315
7073	–582
7073	622

DEFINITIONS

DEFINITIONS

- Date defined a database system as relational “iff it supports at least the following:
  - Relational databases (i.e., databases that can be perceived by the users as tables, and nothing but tables);
  - At least the operations Select, Project and (natural) Join, without requiring any predefinition of physical access paths to support those operations.”
- The operations don’t have to be named as such
- SQL is not required (QBE, anyone?)

- The critical point is how the database appears to the user, not the underlying functionality

DEFINITIONS

- Keys are used to retrieve data from the database
- A candidate key provides a unique means of identifying each tuple in a table
- A primary key is a candidate key chosen as the sole means of accessing each tuple
- A foreign key is a set of attributes in one table that corresponds to the primary key in another table
  - Foreign keys link tables

Lines Table

Line_ID*	From_Pt	To_Pt	Left_Poly	Right_Poly
472	31	87	6194	7073
622	54	22	3008	7073
582	54	87	7073	2247
315	31	22	7073	9462

Points Table

Point_ID*	X	Y
31	xxxxx	xxxxx
54	xxxxx	xxxxx
22	xxxxx	xxxxx
87	xxxxx	xxxxx

Polygons Table

Poly_ID*	Line_ID*
7073	472
7073	–315
7073	–582
7073	622

DEFINITIONS

- Functional dependencies connect candidate keys to other attributes
- The nature of functional dependencies determines which candidate key should be the primary key for the table
- Transitive dependencies occur when there is a succession of functional dependencies within a single table
- Multivalued dependencies occur when there are repeating groups of attribute values that may occur across multiple tables

Lines Table

Line_ID*	From_Pt	To_Pt	Left_Poly	Right_Poly
472	31	87	6194	7073
622	54	22	3008	7073
582	54	87	7073	2247
315	31	22	7073	9462

Points Table

Point_ID*	X	Y
31	xxxxx	xxxxx
54	xxxxx	xxxxx
22	xxxxx	xxxxx
87	xxxxx	xxxxx

Polygons Table

Poly_ID*	Line_ID*
7073	472
7073	–315
7073	–582
7073	622

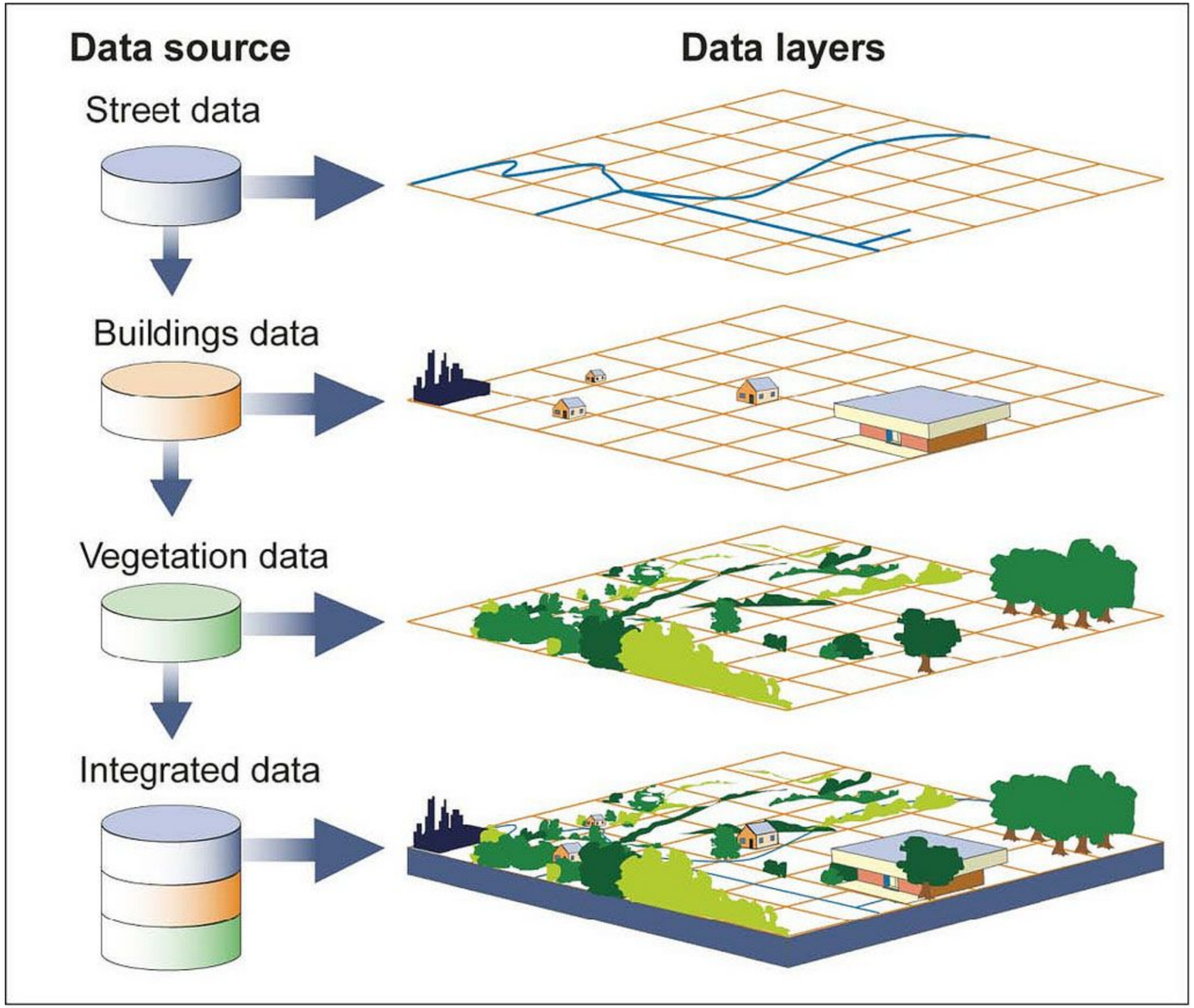

Database
Normalization

NORMALIZATION

- This is the process of reducing redundancy and ensuring more efficient operation
- All relational databases should be normalized to Third Normal Form, at least
- Where there are a number of many-to-many relationship, higher normal forms should be considered
  - This condition occurs with all topological vector spatial data
- Normalization is not obligatory, but it can make database operations a lot better
- Normalization is so named because we can consider each relation (table) to be ‘orthogonal’ to every other, i.e., independent of all the others, only linked by the appropriate keys
- Decomposition of tables to simpler forms is the essence of normalization

FIRST NORMAL FORM (1NF)

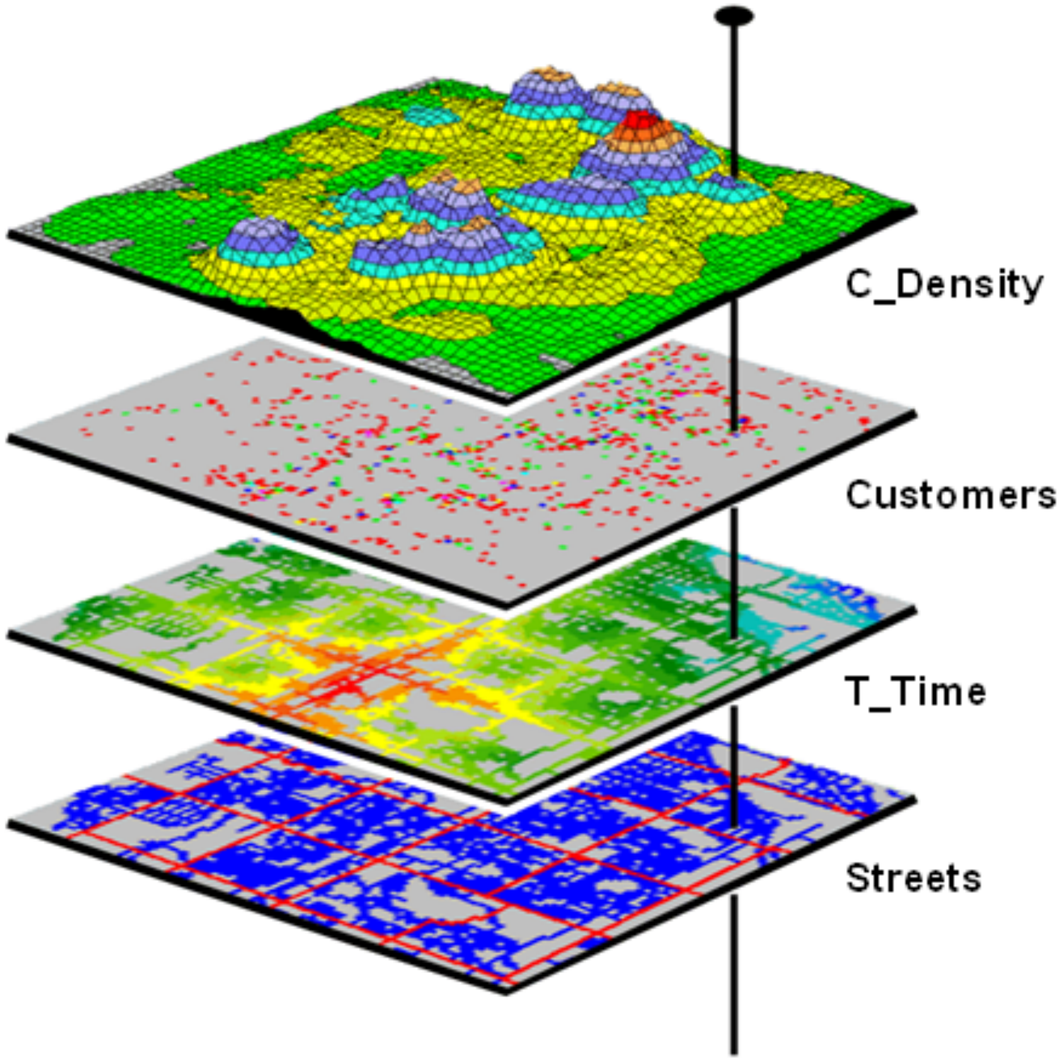
- 1NF: all attributes contain ‘atomic’ values only
  - No multi-valued attributes
- In GIS, each point has a single value for each attribute
  - Attributes are separated into layers, which equate to tables or sets of tables
- This is the same for vector and grid-cell systems



Source: GAO.

SECOND NORMAL FORM (2NF)

- 2NF: in all tables, every non-key attribute is fully dependent on the primary key
- In GIS, all attribute data is directly accessible via the primary key: location
  - Location is always the overall primary key
  - This does not preclude foreign keys
- This is the same for vector and grid-cell systems



THIRD NORMAL FORM (3NF)

- 3NF has evolved, through Boyce-Codd Normal Form, but deals with avoiding transitive dependence in tables
- 3NF: all attributes are fully functionally dependent on the primary key and the primary key is the determinant of each table
- All GIS spatial data tables are in 3NF
- This is the same for vector and grid-cell systems

Lines Table

Line_ID*	From_Pt	To_Pt	Left_Poly	Right_Poly
472	31	87	6194	7073
622	54	22	3008	7073
582	54	87	7073	2247
315	31	22	7073	9462

Points Table

Point_ID*	X	Y
31	xxxxx	xxxxx
54	xxxxx	xxxxx
22	xxxxx	xxxxx
87	xxxxx	xxxxx

Polygons Table

Poly_ID*	Line_ID*
7073	472
7073	–315
7073	–582
7073	622

FOURTH NORMAL FORM (4NF)

- 4NF deals with repeating groups of data within tables
- Splitting tables (the right way) to avoid multi-valued dependencies can avoid problems with spurious groups of data after a join
- The Points table may have multiple appearance of a given X or Y value, but the X and Y values are fully functionally dependent on the primary key, Point_ID
- The Lines table can have all the non-key attributesviewed as a single composite attribute
- Grid-cell systems are 4NF as well

Lines Table

Line_ID*	From_Pt	To_Pt	Left_Poly	Right_Poly
472	31	87	6194	7073
622	54	22	3008	7073
582	54	87	7073	2247
315	31	22	7073	9462

Points Table

Point_ID*	X	Y
31	xxxxx	xxxxx
54	xxxxx	xxxxx
22	xxxxx	xxxxx
87	xxxxx	xxxxx

Polygons Table

Poly_ID*	Line_ID*
7073	472
7073	–315
7073	–582
7073	622

FIFTH NORMAL FORM (5NF)

- 5NF deals with join dependency, where a table cannot be split into two tables without loss of information, or creation of spurious information when the two tables are joined
  - Databases are in 5NF when join dependency is eliminated
- By considering the non-key attributes in the Lines table as a single composite attribute, decomposition cannot be carried further without disruption
- This set of tables is in 5NF
- Grid-cell systems are in 5NF

Lines Table

Line_ID*	From_Pt	To_Pt	Left_Poly	Right_Poly
472	31	87	6194	7073
622	54	22	3008	7073
582	54	87	7073	2247
315	31	22	7073	9462

Points Table

Point_ID*	X	Y
31	xxxxx	xxxxx
54	xxxxx	xxxxx
22	xxxxx	xxxxx
87	xxxxx	xxxxx

Polygons Table

Poly_ID*	Line_ID*
7073	472
7073	–315
7073	–582
7073	622

SIXTH NORMAL FORM (6NF)

- 6NF effectively means breaking tables down to a primary key and one dependent attribute
- 6NF was developed to deal with problems with temporally-referenced data and data in large data warehouses
- The GIS topological vector tables cannot be broken down into a simpler form without compromising operational effectiveness, at least with current GIS (which don't natively handle time), so are in 6NF for current GIS
- Grid-cell systems are in 6NF

Lines Table

Line_ID*	From_Pt	To_Pt	Left_Poly	Right_Poly
472	31	87	6194	7073
622	54	22	3008	7073
582	54	87	7073	2247
315	31	22	7073	9462

Points Table

Point_ID*	X	Y
31	xxxxx	xxxxx
54	xxxxx	xxxxx
22	xxxxx	xxxxx
87	xxxxx	xxxxx

Polygons Table

Poly_ID*	Line_ID*
7073	472
7073	–315
7073	–582
7073	622

OTHER NORMAL FORMS

- A database in 6NF cannot be decomposed any further
  - Therefore there are no further normal forms possible
- However, there have been a few intermediate normal forms proposed over the years to deal with very specific problems
- None of these have any real impact on spatial data in GIS

DISCUSSION IN GIS LITERATURE

- In the regular GIS textbooks, normalization of spatial data in GIS is rarely discussed
  - Laurini and Thompson's 1992 text seems to be one of the few to consider it
- Most texts discuss normalization of attribute data
- Some discuss normalization only in terms of removing bias from attribute data and adjusting discrete attribute values within specific ranges
- There are advantages to discussing normal forms and normalization using spatial data as a way to teach both a way of thinking about spatial data and to cover normalization
- Future GIS that deal natively with time and 'big data' will need this knowledge among practitioners

FUTURE DEVELOPMENTS IN GIS DATABASES

- More complex attribute data (see discussion on Augmented Reality)
- Increasing amount of spatial data collected over time will necessitate GIS handling time natively, along with space
  - Need to change reference frames and locations over time
  - Need to analyze data over time
    - 6NF spatio-temporal databases in data warehouses
- Non-traditional databases may become more relevant, e.g., object-oriented and non-1NF databases, along with interoperability
- Need for a deeper understanding of underlying data structures and models among geospatial professionals

QUESTIONS?

THANK YOU!