GEOG 413/613

LECTURE 9

1

Geospatial Databases

- A database can be thought of as an integrated set of data on a particular subject
  - Data are related and represent a specific aspect of the world (sometimes referred to as a miniworld)
  - Data are for a specific purpose (users and applications) to describe an organization or domain

2

Geospatial Databases

- Databases can be physically stored in files or by using specialist software programs called Database Management Systems (DBMS)
- Geospatial database is a collection of geographic (or spatial) data
  - Has entities (house, river, lake, road…)
  - Has attribute of these entities (location, size, type, name…)
  - Has spatial relationships (distances between entities, adjacency…)

3

Merits of Geospatial Databases

- Data stored at a single location reduces redundancy
  - Consider cadastral data needed by different levels of govt or departments
- Maintenance costs decrease
- Multiple applications and users can use the same data
  - Data are not dependent on software
- Data sharing is easier
  - Multiple interfaces and operations
- Data security and standards

4

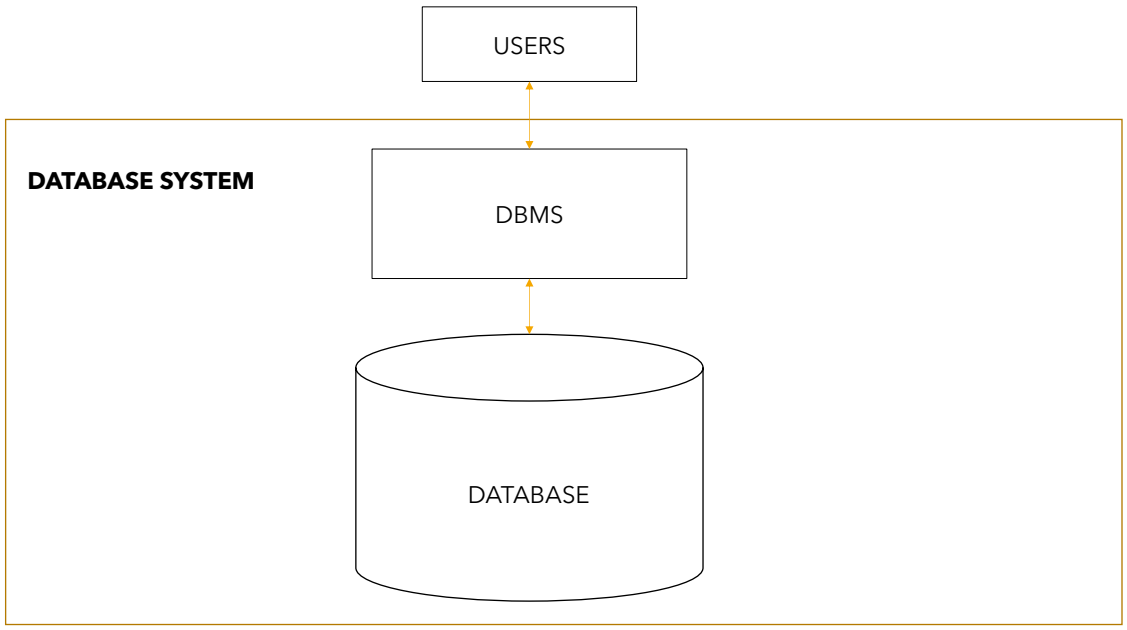
4

Disadvantages Geospatial Databases

- The cost of acquiring DBMS software can be quite high
- A DBMS can add unnecessary complexities for data management in small projects
- Single-user GIS will often be better for files rather DBs

5

5



6

6

Database Management Systems

- Small DBs for a handful of users can be stored on a hard disk
- However, large, complex databases with hundreds of users require specialist DBMS software to ensure data integrity and longevity.
  - A DBMS is a software application designed to organize the efficient and effective storage and access of data. It is used to define, construct, and manipulate a database.

7

7

Functions of a DBMS

- A data model:
  - a mechanism used to represent real-world objects digitally in a computer system, including standard data models suitable for representing several object types (e.g., integer and floating-point numbers, dates, and text)
  - Data model describes the structure of the DB
    - the data types
    - the relations between the data items
    - the constraints that should hold on the data
  - Geospatial DBMSs support geographic (spatial) object types

8

8

Functions of a DBMS

- Data input capability
  - tools to load data into databases in well-structured formats
- Indexing
  - An index is a data structuring used to speed up searching. All databases include tools to index standard database data types
- A query language
  - a standard data query/manipulation language called SQL (Structured/Standard Query Language).

9

9

Functions of a DBMS

- Security
  - DBMSs provide controlled access to data
  - data access rights (read, write, modify, delete) for users to parts or all of DB
- Backup and recovery
  - to protect system from failure and incorrect (accidental or deliberate) update
- Database administration tools
  - Setting the structure of a database (the schema), indexing, backup and recovery, access control

10

10

Functions of a DBMS

- Applications
  - General-purpose tools for creating, using, and maintaining databases (e.g., forms and reports)
- Application programming interfaces (APIs) for further customization

11

11

Types of DBMSs

- Three main types of DBMSs have been used in GI systems:
  - relational (RDBMS)
    - comprises a set of tables, each a two-dimensional list (or array) of records containing attributes about the objects
    - most of the DBMSs are built on relational DB concepts
  - object (ODBMS)
    - RDBMSs were focused primarily on business applications such as banking, human resource management, and stock control and inventory, they were never designed to deal with rich data types, such as geographic objects, sound, and video
    - ODBMSs can store objects persistently
  - object-relational (ORDBMS)
    - An ORDBMS is an RDBMS engine with some additional capabilities for dealing with objects.
      - Provide object description (attributes such as color, size, and age)
      - Provide object methods or functions such as drawing instructions, query interfaces, and interpolation algorithms)
      - Examples: IBM's Informix, Microsoft's SQL Server, Oracle's Oracle DBMS and PostgreSQL (open source development)

12

12

Geographic DBMS Extensions

- Several ORDBMS have spatial database extension to provide core support for geographic data types and functions
  - The open-source DBMS PostgreSQL has the PostGIS extension which supports spatial types and functions
- Typically spatial extensions provide
  - Indexing; Storage management; Transaction services; Query language; DB replication services; Query parser; Query optimizer

13

13

Storing Geospatial Data in DBMS Tables

- Databases users interact with an object class
  - The Object Class is also what is referred to as a layer or feature class (data on a particular these)
  - Object classes are stored in standard database tables
  - Database tables are designed along the following principles
    - There is only one value in each cell at the intersection of a row and column.
    - All values in a column are about the same subject.
    - Each row is unique
    - There is no significance to the sequence of columns
    - There is no significance to the sequence of rows

14

14

Storing Geospatial Data in DBMS Tables

- A database table is a two-dimensional array of rows and columns
- Each object class is stored as a single database table in a DBMS.
  - Table rows contain objects (instances of object classes, e.g., data for a single polygon)
  - Table columns contain object properties (the attributes)
  - Geographic database tables have a geometry column (sometimes called the shape column. The coordinate values may be stored in a highly compressed format.

15

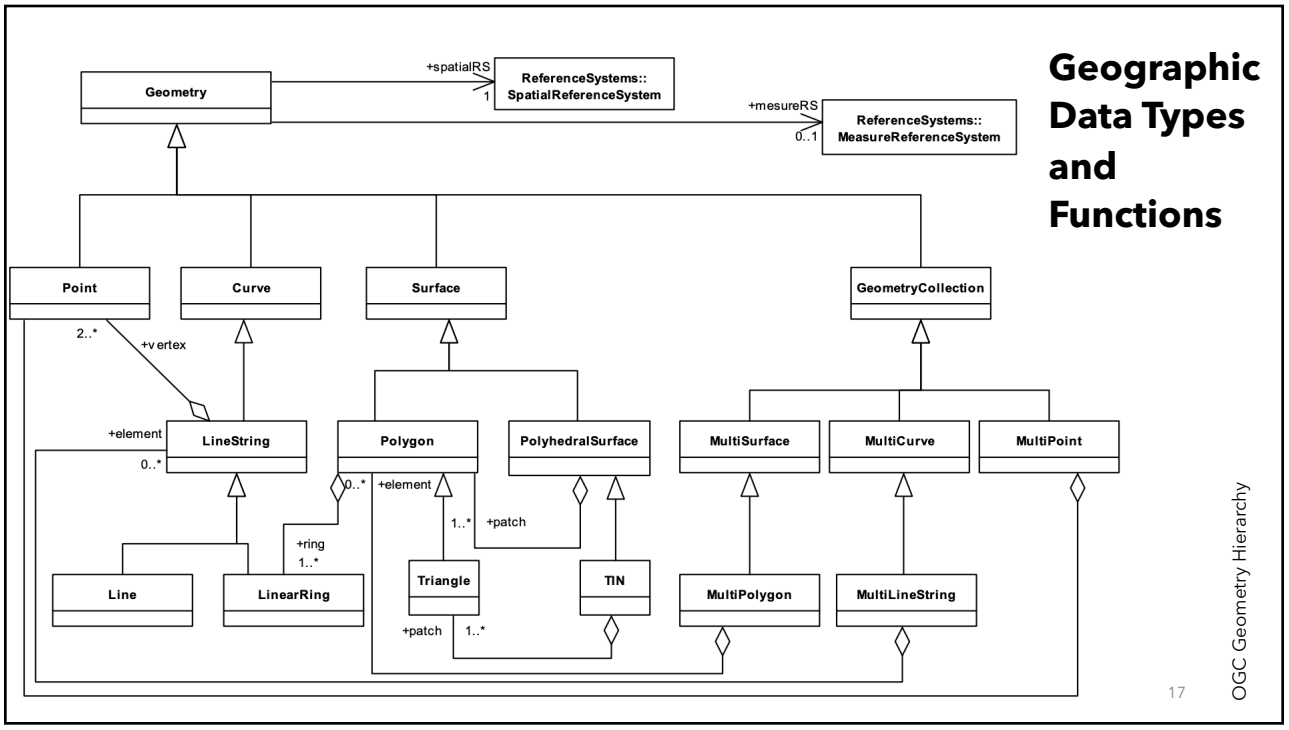
15

SQL

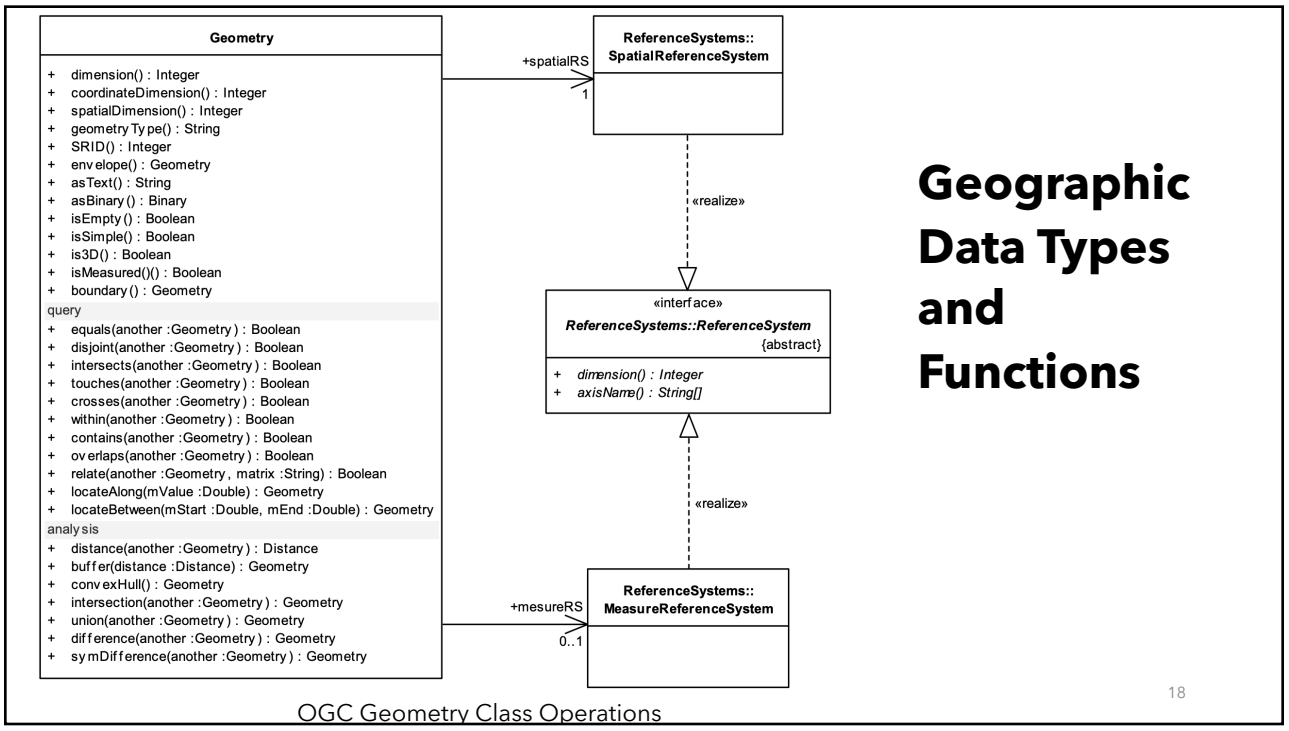
- SQL (Structured Query Language) programming language designed to retrieve sets (row and column combinations) of data from relational databases
- It is the standard database query language it has geographic capabilities
- Some DBMS can have proprietary SQL extensions that are usually only used on their system

16

16


Geographic Data Types and Functions
Geometry
+spatialRS
1
ReferenceSystems:: SpatialReferenceSystem
+mesureRS
0..1
ReferenceSystems:: MeasureReferenceSystem
Point
Curve
Surface
GeometryCollection
2..*
+vertex
+element
0..*
LineString
Polygon
PolyhedralSurface
MultiSurface
MultiCurve
MultiPoint
+ring
1..*
+patch
Line
LinearRing
Triangle
TIN
MultiPolygon
MultiLineString
17
OGC Geometry Hierarchy

17


Geometry
+ dimension() : Integer
+ coordinateDimension() : Integer
+ spatialDimension() : Integer
+ geometryType() : String
+ SRID() : Integer
+ envelope() : Geometry
+ asText() : String
+ asBinary() : Binary
+ isEmpty() : Boolean
+ isSimple() : Boolean
+ is3D() : Boolean
+ isMeasured()() : Boolean
+ boundary() : Geometry
query
+ equals(another :Geometry) : Boolean
+ disjoint(another :Geometry) : Boolean
+ intersects(another :Geometry) : Boolean
+ touches(another :Geometry) : Boolean
+ crosses(another :Geometry) : Boolean
+ within(another :Geometry) : Boolean
+ contains(another :Geometry) : Boolean
+ overlaps(another :Geometry) : Boolean
+ relate(another :Geometry, matrix :String) : Boolean
+ locateAlong(mValue :Double) : Geometry
+ locateBetween(mStart :Double, mEnd :Double) : Geometry
analysis
+ distance(another :Geometry) : Distance
+ buffer(distance :Distance) : Geometry
+ convexHull() : Geometry
+ intersection(another :Geometry) : Geometry
+ union(another :Geometry) : Geometry
+ difference(another :Geometry) : Geometry
+ symDifference(another :Geometry) : Geometry
+spatialRS
1
ReferenceSystems:: SpatialReferenceSystem
«realize»
«interface»
ReferenceSystems::ReferenceSystem
{abstract}
+ dimension() : Integer
+ axisName() : String[]
«realize»
+mesureRS
0..1
ReferenceSystems:: MeasureReferenceSystem
Geographic Data Types and Functions
18
OGC Geometry Class Operations

18

Geographic Database Design

- A GIS will normally be associated with a specific domain e.g. water, forestry, land management
- A database design is the process of modelling the data for this specific domain.
  - This design includes:
    - specification of all data types and relationships
    - the actual database configuration required to store them
- The database design process involves three stages: conceptual, logical, and physical

19

19

DB Design: Conceptual Model (high level)

1. Model the User's View
   - Identify organization functions
   - Determine data require for these functions
   - Organize the data into groups for management
- One can present this conceptual model as as report with accompanying tables

20

20

DB Design: Conceptual Model

2. Define the Objects and their Relationships
   - Specify object types (classes) and functions
   - Specify relationships between object types

- Object models and diagrams are used to describe a set of object classes and the relationships between them

21

21

DB Design: Conceptual Model

3. Select Geographic Representation
   - discrete object or continuous field
   - It is possible to change between representation type later but it can be complex and expensive with a risk of information loss

22

22

DB Design: Logical Model (medium level)

1. Match to Geographic Database Types
   - Match object types to specific data types (point, line, area, georaster, etc.)
     - This could be implemented in Oracle, PostGIS,…
2. Organize Geographic Database Structure
   - Define topological rules and relationships
   - Define coordinate systems

- Hides details of database structure but outlines how data is organised

23

23

DB Design: Physical Model (low level)

1. Final stage before the implementation of the DB
   - Details of how data is structured in a database to minimise storage requirements and improve performance
   - Choices on hardware would be made at this stage

24

24

Database Schema and Instances

- A data models make a clear distinction between the content and its description
  - the description of the database is called *database schema*
  - the data items that reside in a database at a specific point in time form the *database instance*
  - The database schema is specified during the database design process
    - and is not expected to change frequently
    - each modification on the schema may affect the database relation

25

25

Database Schema

Owners

Field Name	Data Type
OID	Large Number
LASTNAME	Short Text
FIRSTNAME	
DOB	
ST_ADDRESS	
CITY	
PROVINCE	
POSTALCODE	
COUNTRY	

Titles

Field Name
TID
PID
OID
PURCHASE_DATE
LASTUPDATE

Parcels

Field Name	Data Type
PID	Large Number
OID	Large Number
Address	Short Text
Area	Short Text

26

26

Database Schema

```
CREATE SCHEMA cadastre;
create table cadastre.parcels (
    PID BIGINT NOT NULL,
    OID BIGINT NOT NULL,
    ADDRESS VARCHAR (50) NOT NULL
    AREA INT NOT NULL,
    PRIMARY KEY (PID)
);
```

27

27

Database Instance

OID	LASTNAME	FIRSTNAME	DOB	ST_ADDRESS	CITY	PROVINCE	POSTALCODE	COUNTRY
12345678	Abe	Ed	11/01/80	123 West St	Sunset	DD	VVVVVV	
23456789	Bud	Fiona	13/02/81	123 East Ave	Meadows	GG	YYYYYY	
34567890	Cage	Gayle	15/03/82	780 Run Cr	Springfield	EE	UUUUU	

TID	PID	OID	PURCHACE_DATE	LAT_UPDATE
2097541	5667890256	12345678	11/01/22	123 West St
5689014	2677908390	23456789	13/02/21	123 East Ave
2678902	2342567123	34567890	15/03/22	780 Run Cr

PID	OID	ADDRESS	AREA
5667890256	12345678	123 Pine St	0.3
2677908390	23456789	123 Queens Ave	0.23
2342567123	34567890	780 Mayor Cr	0.65

28

28

Entity-Relationship Model

- The Entity-Relationship model (ER-model) is a diagrammatic representation of the *miniworld* into a set of entities and their relationships.
  - An entity is a unit with a real existence e.g. a company, a school, parcel
  - Attribute is a property of entity e.g. company name, school population, parcel ID
  - Attributes describe entities
  - An instance of an entity instance comprises the values assigned to its attributes

29

29

Entity-Relationship Model

- Attributes
  - An attribute can be simple or composite
    - Simple e.g. last name
    - Composite contains other attributes e.g. address contains multiple items
  - An attribute can be
    - single valued (e.g., the date of birth)
    - or multivalued (e.g., a person's telephone numbers)
  - An attribute can be
    - either stored in the database, e.g., a person's date of birth
    - or derived after processing the database content, e.g., the current person's age
- An attribute which identifies an entity instance is called a key attribute (e.g., a person's SIN, a parcel's id)

30

30

Key Attributes

- The *key* is an attribute or a group of attributes whose values can be used to uniquely identify an individual entity in an entity set.
  - Candidate key: each attribute or combination of attributes that identifies the row in a relation. For instance, the tuples in the relation of owners can be identified either through the SIN (candidate key 1) or the combination of attributes SURNAME-NAME-DoB (candidate key 2), assuming that there are no two owners in the database sharing the same combination of values in these three attributes.
  - A candidate key is called simple, when it consists of a single attribute (e.g., the candidate key 1) or composite, when it comprises more than one attributes (e.g., the candidate key 2)

31

31

Key Attributes

- The *key* is an attribute or a group of attributes whose values can be used to uniquely identify an individual entity in an entity set.
  - Primary key: the candidate key chosen to identify the rows in a relation. For practical reasons, it is the one with the fewest attributes. For example, the SIN (candidate key 1) is an ideal key for a table of owners.
  - The names of the primary key attributes are underlined in the relational schema.

32

32

Key Attributes

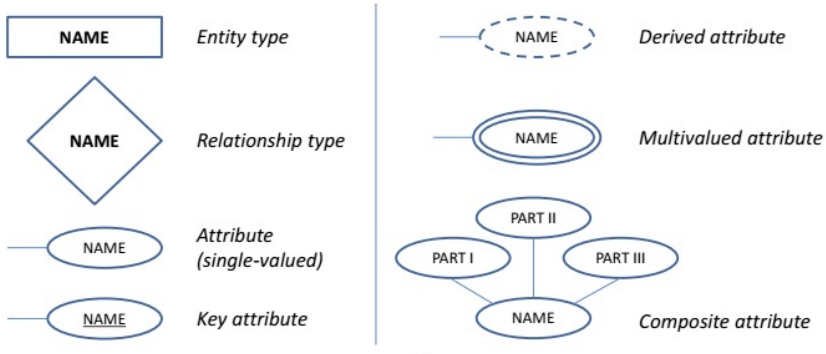
- The *key* is an attribute or a group of attributes whose values can be used to uniquely identify an individual entity in an entity set.
  - Foreign key: is an attribute in a table that references the primary key in another table
  - Both foreign and primary keys must be of the same data type.

33

33

Entity-Relationship Model

- The representation of an entity type in an ER-diagram
  - rectangle denotes the name of the entity type
  - ellipses denotes the attributes assigned to an entity type



34

34

Entity and Relationship Types

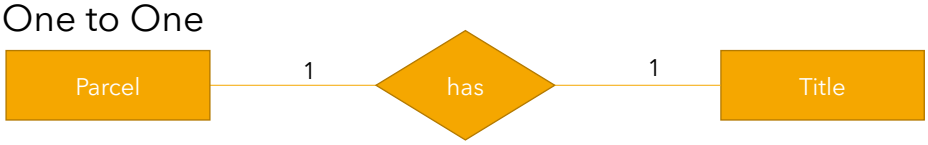
- The entities of one or more types may be related through relationships.
  - A relationship type *R* links entity types E_1, E_2, …, E_n in a structured manner
  - There are three types of relationship that may apply to a database
    - They are one-to-one relationships
    - one-to-many relationships
    - many-to-many relationships
  - These three types of relationship are also referred to the *cardinality*

35

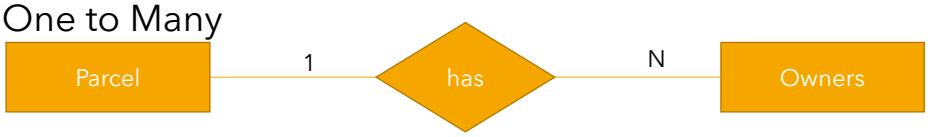
35

Entity and Relationship Types

- One to One



- One to Many



- Many to Many

36

36

Relationship Types

- **One-to-one relationship**
  - One occurrence of an entity relates to exactly one occurrence of another entity.
  - One row in a specific table that relates to one row in another table
    - E.g. Each student is assigned one student ID

37

37

Relationship Types

- **One-to-many relationship**
  - One occurrence in an entity relates to many occurrences in another entity.
  - one row in a specific table that relates to multiple rows in a different table
    - E.g. One customer ID links to Many Order IDs

38

38

Relationship Types

- **Many-to-many relationship**
  - Multiple occurrences in one entity relate to multiple occurrences in another entity.
  - Several rows in a specific table that relate to several rows in another table
    - E.g. Book titles and Authors:
      - A book can be linked to more than one author and an author can be linked to more than one book title

39

39

Normalization

- The process minimizing redundancy in a database
  - Characterizes the level of redundancy in a relational schema
  - Provides mechanisms for transforming schemas in order to remove redundancy
  - Data normalization follows certain rules which are categorized as "normal forms". There are 6 normal forms but we'll briefly look at only 3
- In principle, any information that can be applied to more than one record should be moved to its own table.
  - Each successive normal form applied must meet the rules of the previous form

40

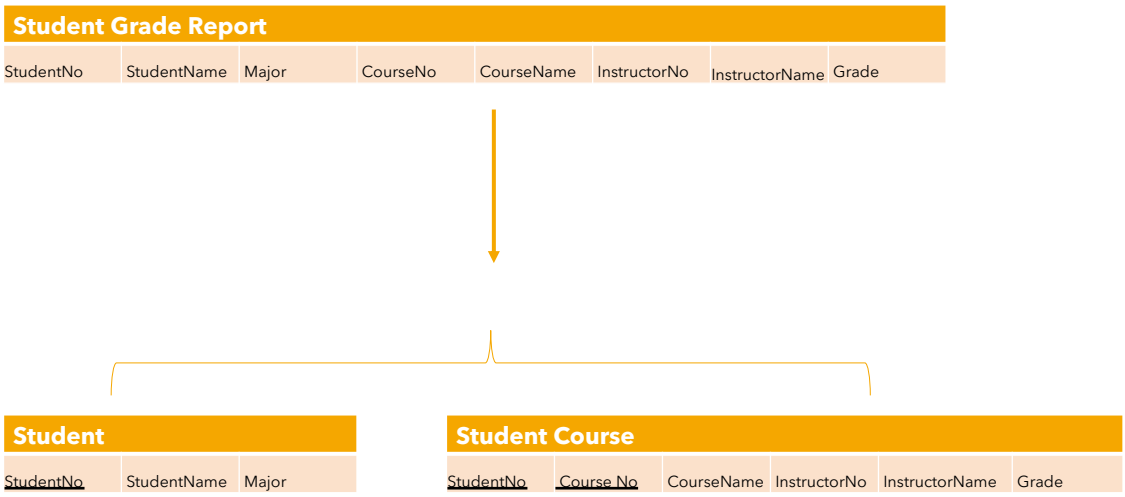
40

Normalization

- First Normal Form (1NF)
  - Eliminates repeated data entries by giving a single value for each cell
  - To normalize a relation that contains a repeating group, remove the repeating group and form two new relations.
  - It creates unique records for each data set and uses a primary key to identify data sets.
  - These primary keys help to organize data that would otherwise need multiple fields.
    - An example of this process could be a student grade report.
    - The repeating group is the course information because a student can take many courses

41

41



42

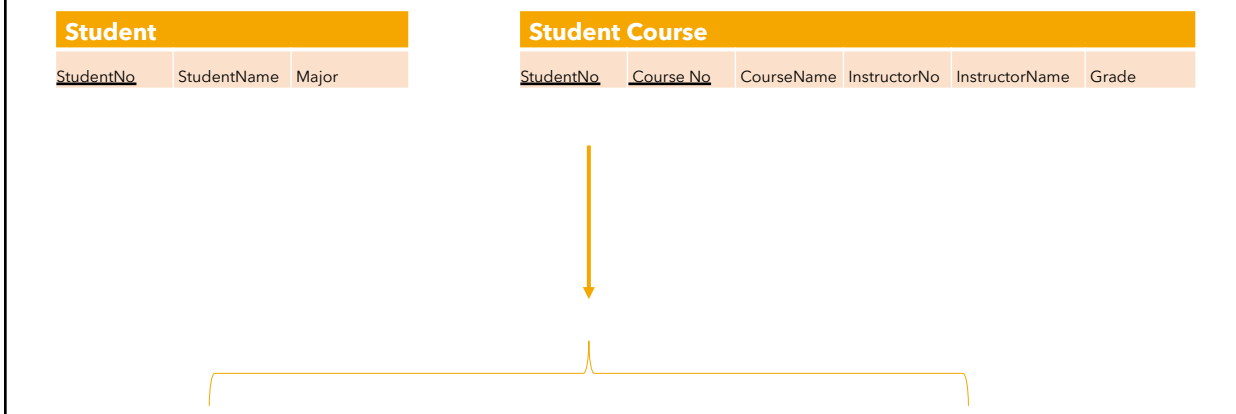
42

Normalization

- Second Normal Form (2NF)
  - For 2NF the relation must first be in 1NF
    - Relation is automatically in 2NF if, and only if, the Primary Key comprises a single attribute.
  - Used to break data into multiple rows and separate tables
  - adds a distinct foreign key to a data set that corresponds with a value in the first normal groupings
  - If the relation has a composite PK, then each non-key attribute must be fully dependent on the entire PK
    - Student table is already 2NF
    - For the course inform not all attributes are filly depend on the Primary Key. The grade is fully dependent on the primary key

43

43



44

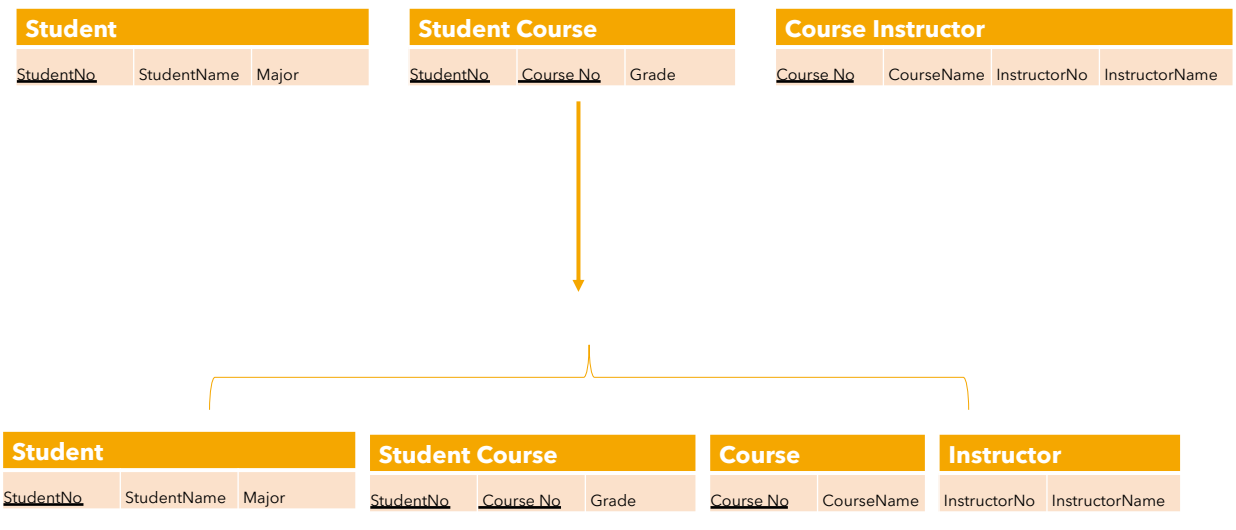
44

Normalization

- Third Normal Form (3NF)
  - focuses on eliminating any fields not dependent on the key. It is used most effectively for information that changes often.
  - If you change the primary key through this step, you must also move all related data into a different table.

45

45



46

46

Indexing Geographic Information

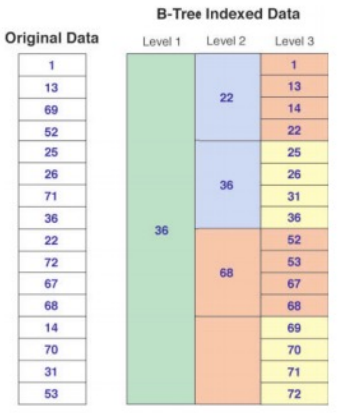
- Geographic databases tend to be very large and geographic queries computationally expensive
  - A database uses indexing to find data quickly
  - A database index is, conceptually speaking, an ordered list derived from the data in a table much like a book index
  - Using an index to find data reduces the number of computational tests that have to be performed to locate a given set of records
  - Full table scans are avoided when an index is created and stored in a table
- *A database index is a special representation of information about objects that improves searching*

47

47

Indexing B-tree example

- Consider B-Tree indexing found in many DBMSs
  - Sort the original data into an ordered list
  - splits the ordered list into buckets of a given size (in this example it is four and then two)
  - the upper value for the bucket is stored
  - To find a specific value, such as 72, using the index involves a maximum of six tests: one at Level 1 (less than or greater than 36), one at Level 2 (less than or greater than 68), and a sequential read of four records at Level 3
  - Of course the larger the dataset, the more effective indexes are in retrieval performance



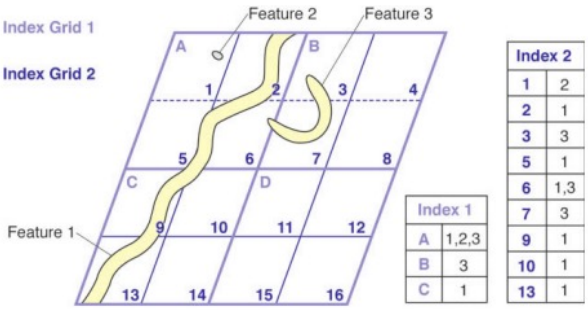
Longley, Goodchild, et al 2015. Geographic Information Science and Systems. Wiley.

48

48

Grid Indexing

- A grid index is similar to a mesh placed over a layer of geographic object
  - The highest (coarsest) grid (Index 1) splits the layer into four equal-sized cells.
  - Cell A includes parts of Features 1, 2, and 3;
  - Cell B includes a part of Feature 3; and Cell C has part of Feature 1.
  - There are no features on Cell D.
- The same process is repeated for the second-level index (Index 2).



Index 1	
A	1,2,3
B	3
C	1

Index 2	
1	2
2	1
3	3
5	1
6	1,3
7	3
9	1
10	1
13	1

Longley, Goodchild, et al 2015. Geographic Information Science and Systems. Wiley.

A query to locate an object searches the indexed list first to find the object and then retrieves the object geometry or attributes for further analysis (e.g., tests for overlap, adjacency, or containment with other objects on the same or another layer). These two tests are often referred to as primary and secondary filters. Secondary filtering, which involves geometric processing, is much more computationally expensive.

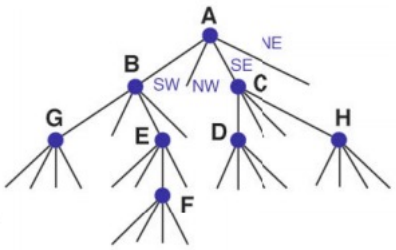
49

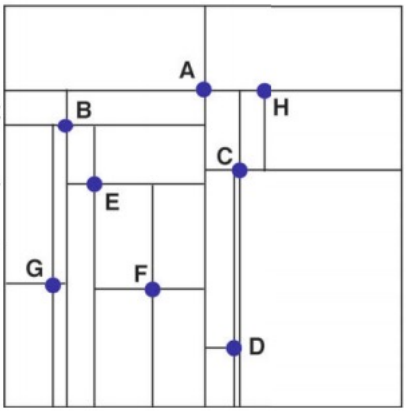
49

Quadtree Indexes

- In a point quadtree, space is divided successively into four rectangles based on the location of the points. The root of the tree corresponds to the region as a whole. The rectangular region is divided into four usually irregular parts based on the (x,y) coordinates of the first point. Successive points subdivide each new subregion into quadrants until all the points are indexed.

Quadtrees are used for both indexing and compressing geographic database layers. The many types of quadtrees can be classified according to the types of data that are indexed (points, lines, areas, surfaces, or rasters), the algorithm that is used to decompose (divide) the layer being indexed, and whether fixed or variable resolution decomposition is used.



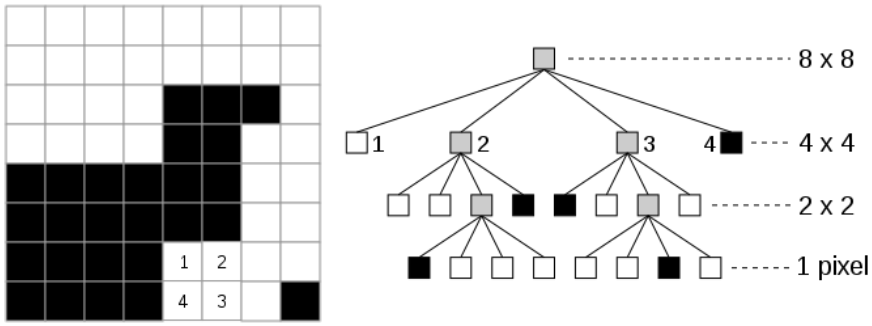


Longley, Goodchild, et al 2015. Geographic Information Science and Systems. Wiley.

50

50

Quadtree Indexes



http://wiki.gis.com/wiki/index.php/Quadtree

51

51

R-Tree Indexes

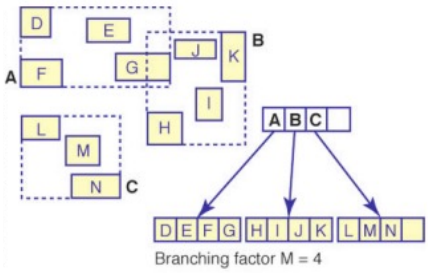
- R-trees group objects using a rectangular approximation of their location called a minimum bounding rectangle (MBR)
  - Groups of point, line, or area objects are indexed based on their MBR
- Objects are added to the index by choosing the MBR that would require the least expansion to accommodate each new object.
- If the object causes the MBR to be expanded beyond some preset parameter, then the MBR is split into two new MBRs.

52

52

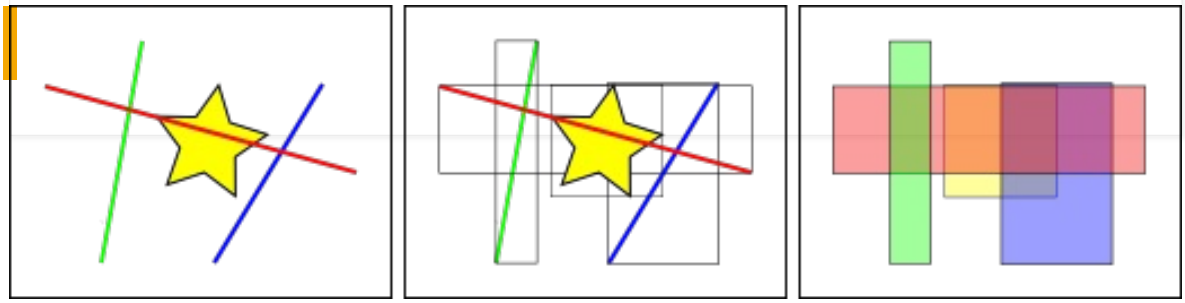
R-Tree Indexes

- The lowest level contains three "leaf nodes";
- the highest has one node with pointers to the MBR of the leaf nodes.
- The MBR is used to reduce the number of objects that need to be examined in order to satisfy a query.
- R-trees are popular methods of indexing geographic data because of their flexibility and excellent performance.



53

53



R-Tree Indexes

- The number of lines that intersect the yellow star is **one**, the red line. But the bounding boxes of features that intersect the yellow box is **two**, the red and blue ones.
- The way the database efficiently answers the question "what lines intersect the yellow star" is to first answer the question "what boxes intersect the yellow box" using the index (which is very fast) and then do an exact calculation of "what lines intersect the yellow star" **only for those features returned by the first test**.

54

54

Editing and Data Maintenance

- A general-purpose geographic database will require tools for
  - geometry and attribute editing
  - database maintenance
  - creating and updating indexes and topology,
  - importing and exporting data
  - georeferencing objects

55

55

Editing and Data Maintenance

- Access to the database must be carefully managed to ensure continued security and quality.
- Edits must be stored persistently in the database
- The mechanism for managing edits to a database is called a *transaction*
- Editing for multiple users will require
  - concurrent read, write and query access
  - avoid database corruption from multiple concurrent edits

56

56

Transactions

- A transaction is a group of changes that are made to a database as a coherent group. All the changes that form part of a transaction are either committed, or the database is rolled back to its initial state.
- Many databases are multiuser and transactional
  - they have multiple users performing update operations at the same time

57

57

Transactions

- Many databases have short transactions (less that 0.01 seconds) e.g. editing bank records
  - Multiuser editing is handled by locking (preventing access to) affected database records during the course of the transaction
- Geospatial databases have long transaction times
  - locking a database for a long transaction is impractical
  - If system failure occurs during a long transaction, work may be lost unless there is a procedure for storing updates in the database

58

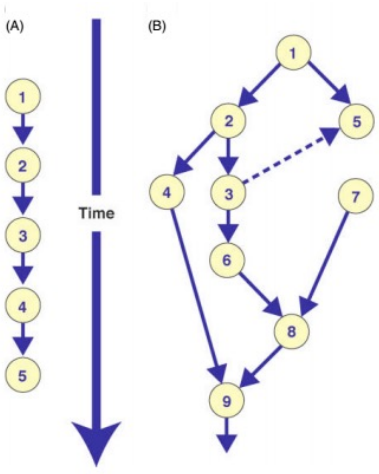
58

Versioning

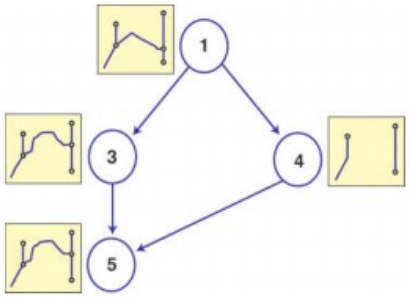
- Versioning allows multiple users to update a database at the same time
  - Assumes that conflicts from concurrent edits are very unlikely to occur and can be used to resolve them

59

59



Database transactions: (A) linear short transactions; (B) branching version tree.



Version reconciliation. For Version 5 the user chooses via the GUI the geometry edit made in Version 3 instead of 1 or 4

Source: Longley, Goodchild, et al Geographic Information Science and Systems. Wiley

60

60

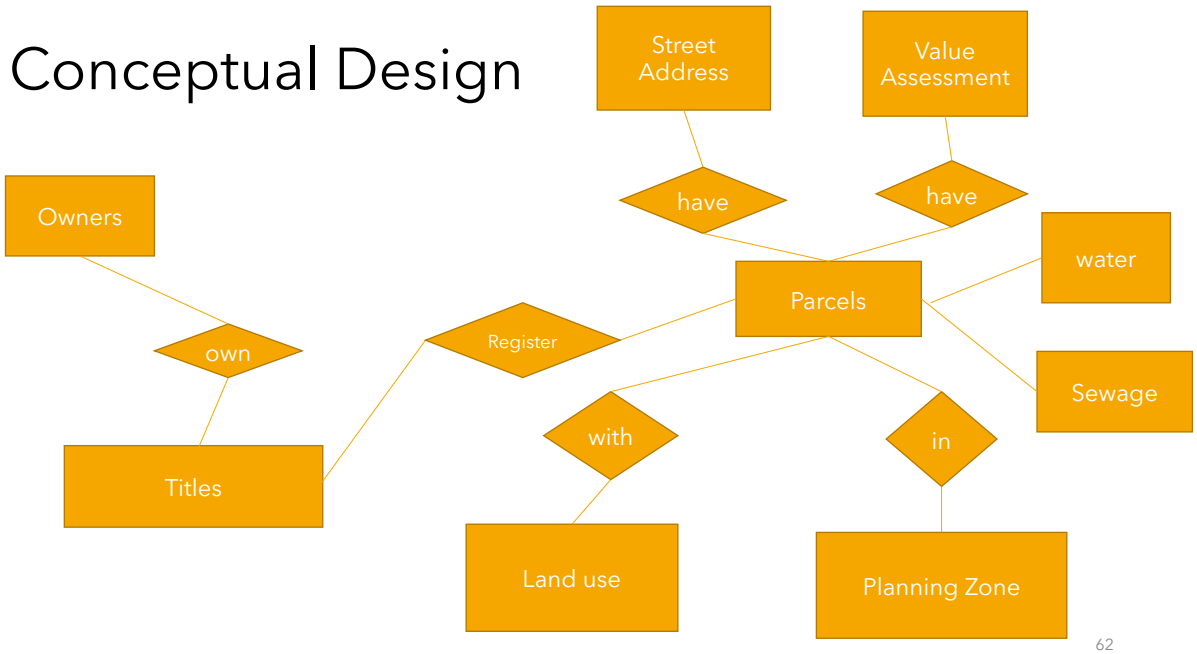
In Class Exercise

- Database Design Exercise
- Consider a contrived case of a database cadastral parcel information
  - Needed by city planners
  - Needed by taxation authorities
  - Needed by public for records search

61

61

Conceptual Design



62

62

Conceptual Design

Owners
- Name
- Phone
- OID
- Address
- PID

Parcels
- PID
- Size
- Geometry
- Address
- OwnerID

Titles
- TID
- OwnerID
- PID
- IssueDate
- LastUpdate

63

63

Conceptual Design

LandUse
- LUICODE
- LUNAME
- Geometry
- MuniAuthority

PlanningZone
- ZNID
- ZNNAME
- Geometry
- ASSIGNDATE

Utilities
- UTID
- Geometry
- ServiceType
- Provider
- Length

64

64

Logical Design

65

Implementation

66