Ekonomická fakulta
Faculty of Economics
Jihočeská univerzita v Českých Budějovicích
University of South Bohemia in České Budějovice

Geographic Information Systems 1
Lecture 6: Databases, Geodatabase

Renata Klufová
University of South Bohemia, Faculty of Economics

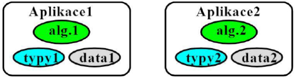
April 2021

- **Database** is a shared collection of logically ordered data (and descriptions of that data - **metadata**) that is designed to meet the needs of the user.
- Original name - databank: a specific set of organized information (data)
  - Corporate archive,
  - Library card filing system,
  - Information about the company's customers,
  - Information about students and fields of study of the faculty.
- **Aim**: the most effective way of finding information about embedded phenomena,
- Computer databases only make it possible to process data more efficiently.
  - Interactive ordering and sorting of data, searching for specific text, searching for data meeting a complex set of conditions ...

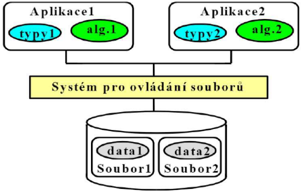
- **Database**: set of all user data stored in the database**Datová základna**: soubor všech uživatelských dat uložených v databázi
- **Database system** = data + tools for working with data
  - Access
  - FoxPro, dBase
  - Paradox
  - Oracle
  - MySQL
  - and other.

- establishment of records,
- filling with data,
- data editing,
- add additional monitored data,
- delete data,
- write new data,
- calculate other data,
- data sorting,
- selection of data,
- forms,
- export/import,
- macros, modules.

- 1950s - **agenda data processing** - data was stored as one or more computer files that could be accessed using software specialized for this purpose.

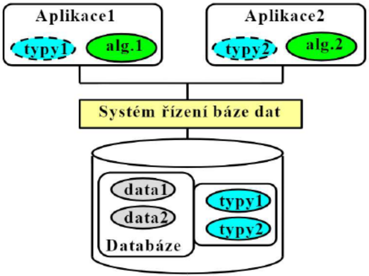


- 1960s ⇒ **file control system** (supported by the host operating system), emergence of programming languages for working with files - classic "mass data processing"



disadvantages: data redundancy, risk of inconsistency, data access problems for unplanned (ad hoc) queries, data isolation, data security problems (limited access), integrity problems

- 2nd half of the 60s - database technology - database management systems (DBMS, DBMS - Database Management System)



The basic benefit of database technology - achieving a certain **independence of data** on user programs and vice versa.

Database (DB) - persistent data, used by application systems of the institution (in the classic database **structured**).

Persistent data - data with a lifetime exceeding the application program runtime and computer shutdown.

other properties of the database data:

- **integrated** - can be understood as unification of several data sets with removal of redundancy (complete or partial),
- **shared** - typically multi-user access with possible view restriction,
- **secure** - easier to implement data access rights restrictions,
- easier to ensure **data integrity** (implementation of integrity constraints).

Integrity of data - correctness of data in terms of meeting constraints that exist in the real world - e.g. the specified branch must exist, the birth number must meet the divisibility condition 11 ...

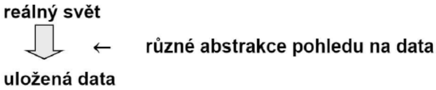
Data consistency - if it is violated, the data is inconsistent - e.g. a client's address with a different value is stored in the database twice, after transferring the amount from account A to account B the sum of both accounts is different than before the transfer ...

Database Management System - program layer dealing with DB operations - goal: shielding the user (application) from technical details - operations: creating DB, tables, searching, inserting, ...

Database system (DBS) - a system that in a broader sense includes:

- technical tools,
- data - DB,
- software - DBMS, development tools, libraries, ...
- DB users.

One of the important tasks of a DBS is to provide users with an abstract view of the data (the details of data storage and management are hidden).



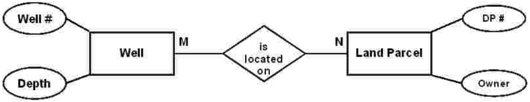
basic levels of data abstraction:

- **physical (internal)** - describes the data as it is actually stored,
- **conceptual (logical)** - describes what data is actually stored in the database and what relationships exist between them,
- **view level (external)** - describes what data is visible to individual users, i.e. generally only the part of the database that represents data representing real-world objects visible to individual users.

Data Model - a collection of conceptual tools for describing the oibjects of reality, or the data representing them, the relationships between them, semantics and integrity constraints.

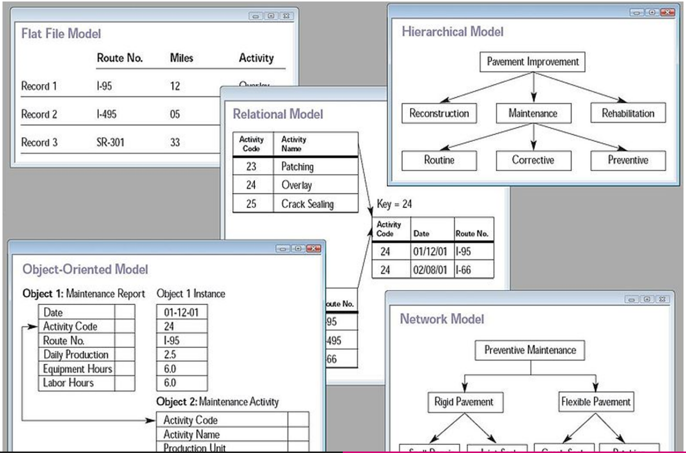
according to the level of modelling:

- **logical models** - describe data at the conceptual and view level,
  - models for modelling real world objects (ER model, OO model, functional model) - **conceptual modelling**;
  - database models defining the logical organization of data in the database (relational, network, hierarchical, OO, object-relational, ...);

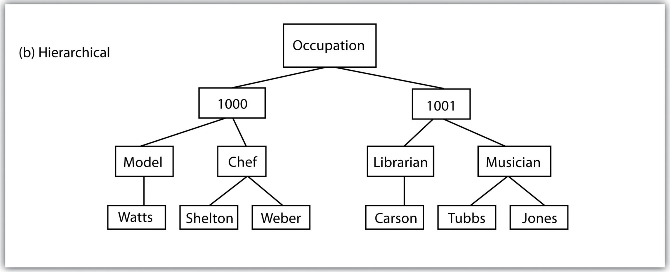


- **physical data models** - describe data at the physical level.

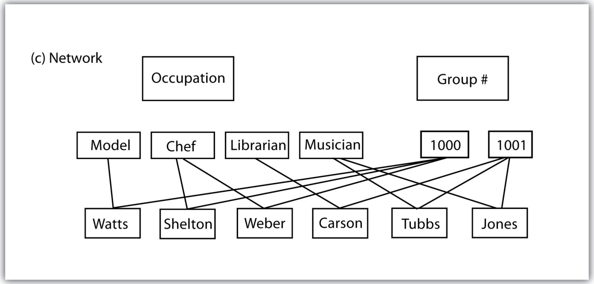
The evolution of database systems has moved from the network model, through the hierarchical model, to the most widely used relational model, to today's evolving object-oriented model:



- **hierarchical model** - 2nd half of the 60s and onwards - entity types strictly mapped on the example of the father-son relationship - the structure is formed as a system of connections of all records in the form of a tree, the top of the hierarchy is the root. The hierarchical model allows only 1:1 and 1:n relationships between entity types; disadvantages: high redundancy (repeated storage of the same data in different places) and a specified query procedure, given by the hierarchy, which is difficult to change.

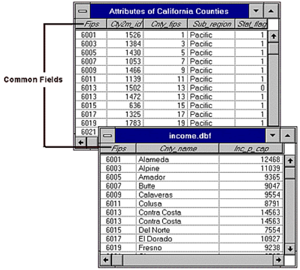


- **network model** - it abandons strict hierarchy and there can exist even m:n type relations - it can be seen as a kind of generalization of the hierarchical model - more flexible and less redundant structure - used e.g. when searching for the best connection in a communication network.



Note: Hierarchical and network structures are sometimes referred to as navigational structures because the connections within them are constructed using pointers, which are also used in the design of application programs.

- **relational model** - 1970 - theory, 1975 System R (IBM) - at the conceptual level data structured into tables (so called normalized - values in tbl. must be atomic in terms of meaning), no hierarchy of fields inside the record, each field can be a key to access data in another table, all possible associations (1:1, 1:n, m:n) between entity types are represented by pointers:
  - serve as lookups between different sessions, i.e. they join a table with another table,
  - used for unambiguous identification of entities - so called primary keys (must have two properties: it is unambiguous, it is minimal - no attribute can be omitted in order not to violate the rule of uniqueness).



Attributes of California Counties

Fips	Cty2m_id	Cnty_fips	Sub_region	Stat_flag
6001	1526	1	Pacific	1
6003	1384	3	Pacific	1
6005	1430	5	Pacific	1
6007	1053	7	Pacific	1
6009	1466	9	Pacific	1
6011	1139	11	Pacific	1
6013	1502	13	Pacific	0
6013	1472	13	Pacific	1
6015	636	15	Pacific	1
6017	1325	17	Pacific	1
6019	1783	19	Pacific	1
6021				

income.dbf

Fips	Cnty_name	Inc_p_cap
6001	Alameda	12468
6003	Alpine	11039
6005	Amador	9365
6007	Butte	9047
6009	Calaveras	9554
6011	Colusa	8791
6013	Contra Costa	14563
6013	Contra Costa	14563
6015	Del Norte	7554
6017	El Dorado	10927
6019	Fresno	9238

Operations that can be performed on sessions are divided into two basic groups:

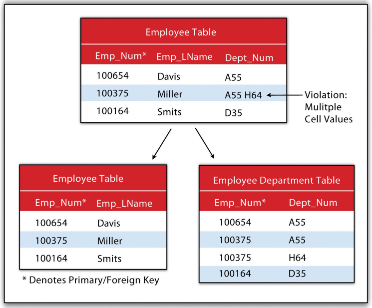
- relational algebra,
- relational calculus.

E-R diagram - for designing and writing relationships between DB entities - Peter Pin Shan Chen (1976)

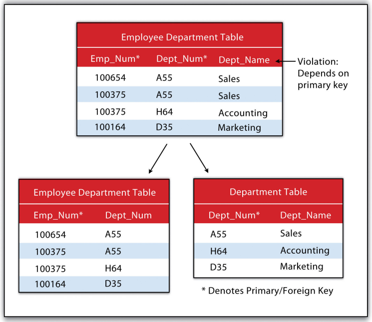
Jednoduchý E-R diagram

Normal forms - used for better database system design. In general, the higher the normal form of a table, the better the table is designed:

- **0. NF**: a table is in zero normal form if there is at least one field that contins more than one value.
- **1. NF**: the table is in the first normal form if only a simple data type can be inserted into each field (they are indivisible).



- **2.NF**: the table is in the second normal form if it is in the first normal form and, in addition, there is a key and all non-key fields are functions of the whole key (and thus not just parts of it).



Employee Department Table

Emp_Num*	Dept_Num*	Dept_Name
100654	A55	Sales
100375	A55	Sales
100375	H64	Accounting
100164	D35	Marketing

Violation: Depends on primary key

Employee Department Table

Emp_Num*	Dept_Num
100654	A55
100375	A55
100375	H64
100164	D35

Department Table

Dept_Num*	Dept_Name
A55	Sales
H64	Accounting
D35	Marketing

* Denotes Primary/Foreign Key

- **3.NF**: a table is in the third normal form if every non-key attribute is not transitively dependent on any schema key (see Figure 1), or if it is not in the second normal form and at the same time there is no single dependency of the non-key columns of the table.

Department Table

Dept_Num*	Dept_Name*	Mgr_Num†	Mgr_Name‡
A55	Sales	100152	Brown
H64	Accounting	100987	Bird
D35	Marketing	100026	Hart

Department Table

Dept_Num*	Dept_Name	Mgr_Num
A55	Sales	100152
H64	Accounting	100987
D35	Marketing	100026

Employee Table

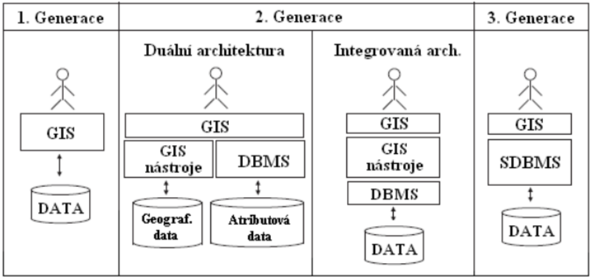
Emp_Num*	Emp_LName
100654	Davis
100375	Miller
100164	Smits
100152	Brown
100987	Bird
100026	Hart

* Denotes Primary/Foreign Key
† Depends on Primary Key
‡ Depends on Non-Primary Key

- **4. NF**: a table is in the fourth normal form if the columns (attributes) contained in it describe only one fact or one context.
- **5. NF**: a table is in the fifth normal norm if it is in the fourth and it is not possible to add a new column (group of columns) to it so that it breaks down into several subtables due to hidden dependencies.

And what about spatial data storage?

Since both spatial (geometric) and non-spatial (attribute) data are stored in a database, the problem of data organization in GIS is considered a "database" problem. Spatial data is represented in GIS by storing geometry and its associated attributes. Different GIS systems differ considerably in terms of how data is stored and how the attributes and the spatial (geometric) part of the geographic database are linked.



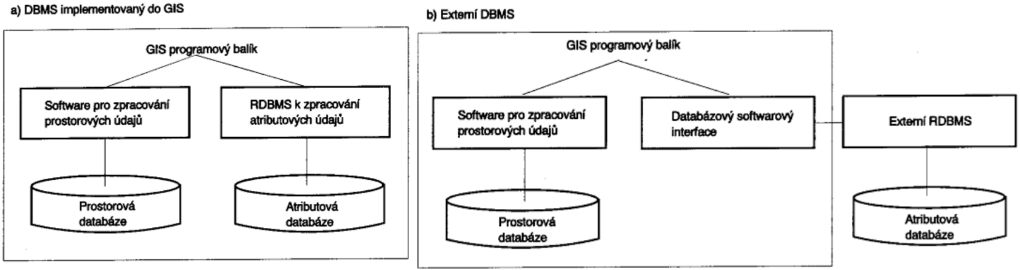
- **"first generation" GIS** (without DBMS)
  (a) systems without attribute file - pure raster approach does not allow separation of localization and attribute data,
  (b) **flat** file systems - separate storage of geometric and spatial data in separate environments and their merging only when needed.

(a) Flat File

Name	Group #	Occupation
Watts	1000	Model
Shelton	1000	Chef
Weber	1000	Chef
Tubbs	1001	Musician
Jones	1001	Musician
Carson	1001	Librarian

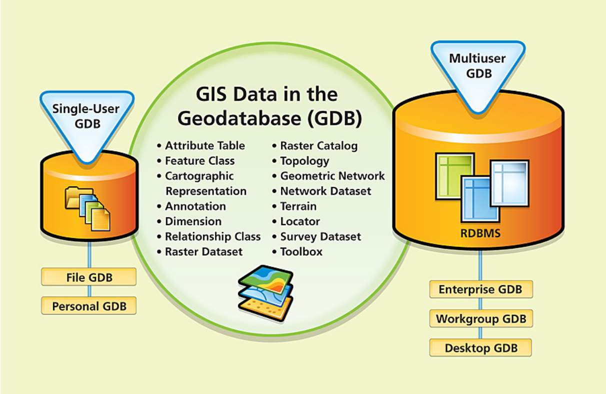
- **"second generation" GIS** (with DBMS)
  (a) **dual/hybrid systems** - non-spatial data stored in a relational database, spatial objects in a file system - e.g. ARC/INFO, MGE and Geo/SQL;
  (b) **integrated systems** - spatial and non-spatial data stored in one database structure, spatial data types are not available for storing spatial data, so data are stored as BLOB; functionality for manipulation of these data incorporated into so-called middleware - e.g. ArcSDE, GeoMedia, SYSTEM 9;

dual/hybrid systems

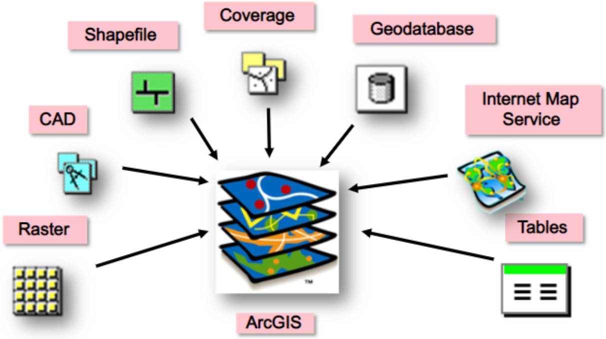


- **"third generation" GIS** - object model - two types of approaches:
  (a) **object** - one real entity corresponds to one database object, data (spatial + non-spatial) stored together with object methods;
  (b) **object-relational** - spatial data types including corresponding operations and functions are integrated into a relational DBMS - e.g. Oracle Spatial, PostGIS, Geodatabase.

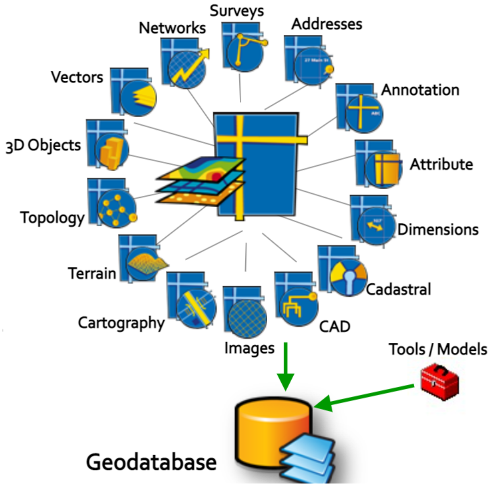
= ArcGIS data storage and management environment - can be used for desktop, server and mobile environments



patial data in many formats:

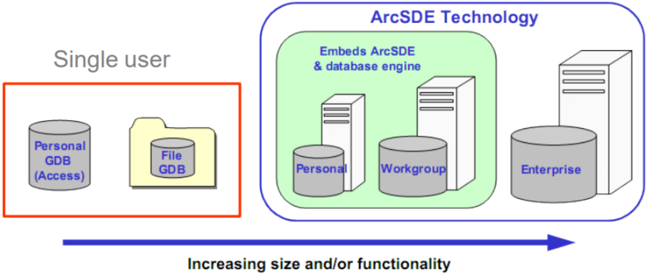


integration of data from many sources:



ArcGIS - three main types of geodatabases - they differ in the technologies used to create and manage them:

- Personal Geodatabase
- File Geodatabase
- ArcSDE Geodatabase



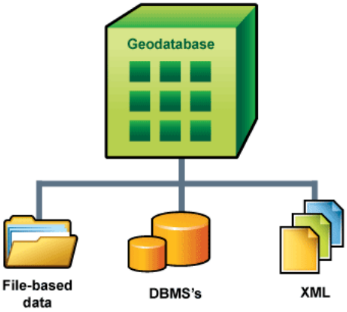
designed for individual work within desktop GIS:

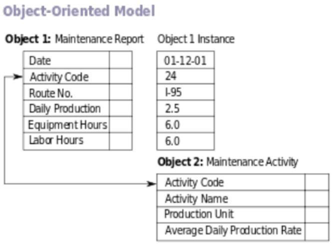
	File	Personal
Storage Technology	Uses local file structure	Microsoft Access (Jet Engine)
Licensing	ArcGIS for Desktop Basic, Standard, and Advanced	ArcGIS for Desktop Basic, Standard, and Advanced
Differentiating Characteristics	No versioning support 1 TB per table size limit (default)	No versioning support Max. of 2 GB of data

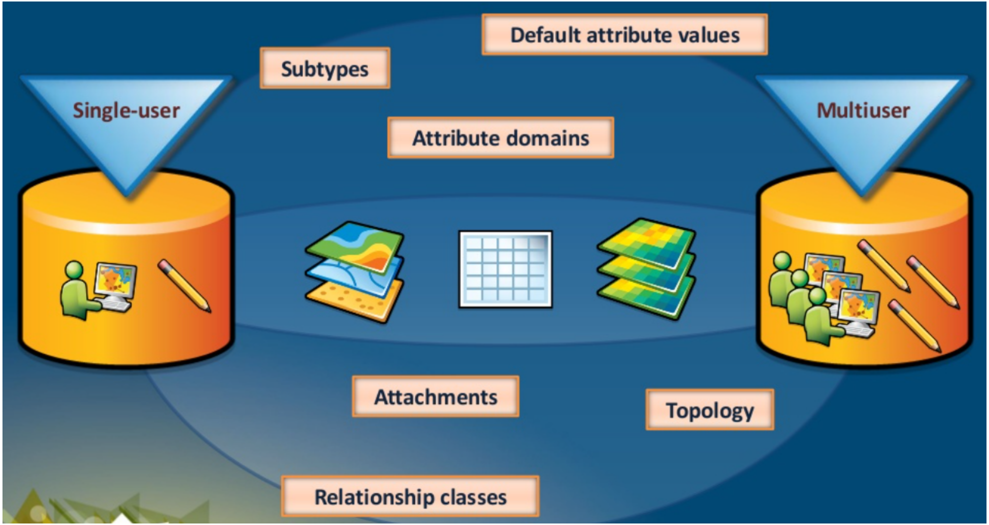
- = a relational database that stores geographic data,
- common storage of spatial and attribute data and the relationships that exist between them,
- advantages:
  - support for two-, three-and four-dimensional vector data,
  - the ability to classify elements within a single class using subtypes,
  - the ability to define spatial relationships between data using topology rules,
  - offline editing,
  - data exchange (import, export) in XML format,
  - and other ...

ESRI geodatabase
architecture

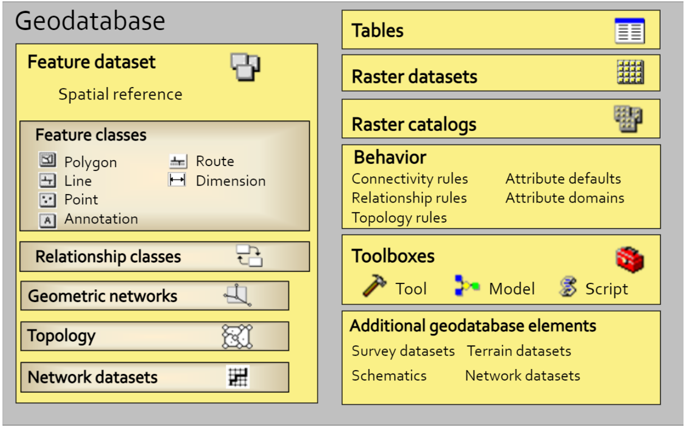
gdb - multilayer application architecture with developed logic and behavior - various DBMS, files, XML - **object-relational model**





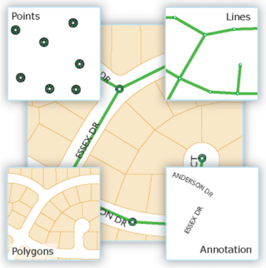

Default attribute values
Subtypes
Single-user
Multiuser
Attribute domains
Attachments
Topology
Relationship classes

basic elements of the geodatabase:



three key components:

- **Feature class** = set of features of the same geometric type (point, line or polygon) and attributes expressed in the same coordinate system - example: all restaurants in a city can be stored in geodatabase as one feature class. Geometrically, the restaurants on the map would be represented as a point whose coordinates would be expressed in the chosen coordinate system. A "non-spatial" table would then store for each restaurant information about its opening hours, capacity, etc. Feature classes can exist in the geodatabase as stand-alone classes, or they can be part of a feature dataset.

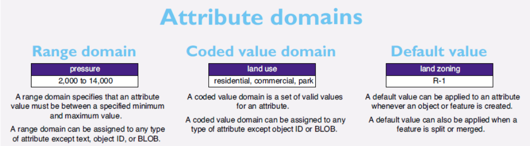


three key components:

- **Feature dataset** = collection of feature classes. All feature classes within a feature dataset must have the same coordinate system. Feature dataset is primarily used to store feature classes that have topological relationships with each other, such as such as adjacency. To be able to define the use of topological rule between feature classes, these classes must be part of a single feature dataset.
- **Nonspatial Tables** = "non-spatial" tables contain attribute data that can be associated with feature classes. These tables contain only attribute data, they do not contain geometric descriptions of the elements - which distinguishes them from so-called feature class tables, which contain at least one column with a geometric description of the features.

other elements of personal geodatabase:

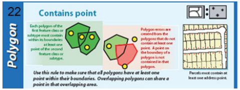
- **Anotations** = map text containing the characteristics of how it should be the text should be displayed,
- **Domains** - domains = special case of annotation. Domains prevent errors when entering data into the geodatabase. They are also used to control attribute values in existing data. Domains define a set of of allowable values that can be inserted into an attribute. A domain is defined by either a coded value domain or a range domain).



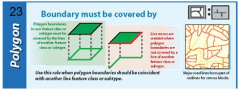
other elements of personal geodatabase:

- **Topology** (topology) - spatial relationships between elements - defining topological rules is necessary if, for example modelling river basins. Then it is desirable that the elements interact with each other (watercourses) of this network are related to each other, etc., and this continuity and other properties can be ensured by defining appropriate topological rules.

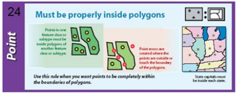
Between a polygon and a point feature class

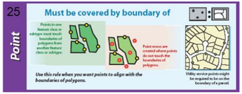


Between a polygon and a line feature class



Between a point and a polygon feature class



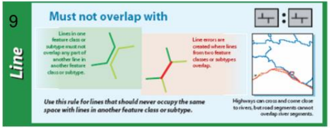


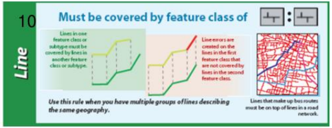
All lines and polygons

Note: numbers are mine, not ESRI's!

other elements of personal geodatabase:

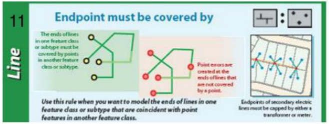
Between two line feature classes

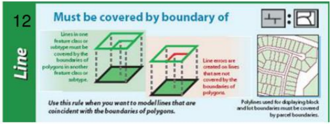




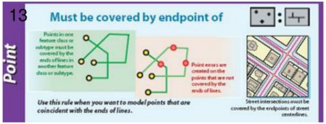
Between a line and a point feature class

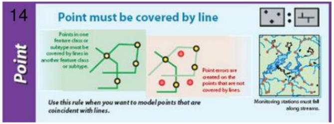
Between a line and a polygon feature class





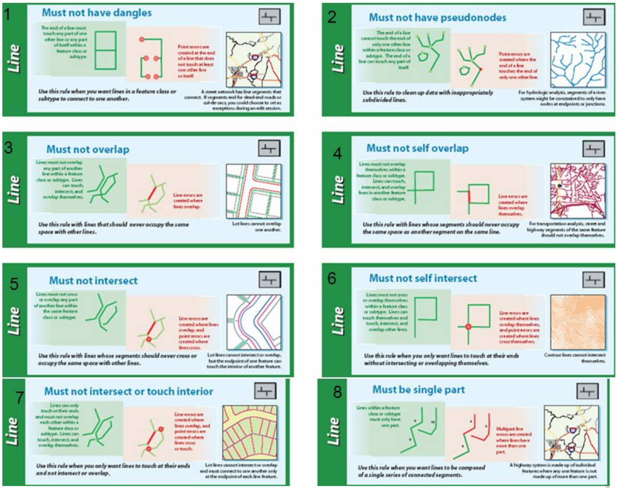
Between a point and a line feature class





other elements of personal geodatabase:

Within <u>one</u> line feature class



Three types of topologies are possible in the geodatabase:

- geodatabase topology,
- map topology,
- topology created for a geometry network topology).

Geodatabase Topology
ArcGIS contains over 20 topological rules that can be used to to model spatial relationships between features and "force" their compliance. All feature classes involved in the geodatabase topologies (in other words, the elements of these feature classes are subject to some topological rule) must be contained in the same feature dataset.

Geometric network - geometric network. It is probably not difficult to imagine a figurative road network on the territory of the city of Pilsen, for example. Using geodatabase we can model this situation. In this case, our model would contain a point theme (intersections, ...) and linear theme (roads) in the form of several feature classes collected together in one feature dataset. Such a geometric network would then allow us to answer questions such as:

- What is the shortest route from point A to point B (between two intersections)?
- How long is the journey from M1 to M2?

Relationship classes - relationships between real world objects. In the geodatabase we have relationship classes represent a way to model the relationships that exist between real-world objects. An example of such a relationship might be the relationship parcel - building. We can have a specific building in the feature class building that stands on a a specific piece of land, represented as a parcel in the feature class parcel. If I have a relationship defined in the geodatabase in the form of a relationship class between elements from the feature class of the parcel and the feature class of the building, then, for example, when removing a given parcel from the map, the corresponding of the building that stands on the given parcel.

- **Raster Data** - just as we can work with vector data in the geodatabase, we can also work with raster data. There are two types of raster objects we can create in a geodatabase - raster dataset and raster catalog.
- **Raster dataset** is created from one or more separate rasters. In case we create a raster dataset from multiple raster, these data are merged into one seamless dataset. In this case, the input rasters must have the same coordinate system, uniform cell size and data format. An .img file (ERDAS IMAGINE file) is created for each raster dataset.
- **Raster catalog** - contains a collection of rasters that do not have to be related to each other, can be stored in different formats and have different differences. Raster catalog is defined in the geodatabase as a table that can be viewed in ArcCatalog as any other table. Each raster (within the catalog) corresponds to one row in raster catalog table.

The personal geodatabase format provides extensive functionality and offers many advantages for GIS users:

- **Personal geodatabase** is a relational database that stores geographic data.
- There are two types of ESRI geodatabase format - **personal** and **multiuser**.
- The key components of personal geodatabase are - **Feature class, Feature dataset and Nonspatial tables**.
- In the geodatabase we can define **topology** and **relationships between elements** (relationship).
- Two types of raster objects can be created in the geodatabase - raster dataset and raster catalog. Multiuser geodatabase directly stores raster data, while a personal geodatabase references rasters.

Thank you for your attention.