Data design

* A successful GIS is built on a well-designed collection of geographic data that effectively models your world and makes for simple and efficient display, editing, and analysis.

\* Data modeling tools allow you to do the following:

1. Set up user or system defined rules
2. Automate data processes
3. Clean your data through comprehensive checks

Data design types

Fields

Fields are the components that provide structure for a table. Tables form the foundation of geographic data and are the fundamental building blocks of any data model. Tables are a collection of information, such as a list of building owners, employees, or customer information. Layers in a map, such as city streets, buildings, or address points, are essentially tables that include information about the geometry and location of those features.

Tabular information consists of rows and columns. In ArcGIS, rows are referred to as records, and columns as fields. Each field in a table can store a specific type of data, such as a number, date, or piece of text. Fields form the attribute information for each layer in the map or stand-alone table.

A screenshot of a computer

Description automatically generated

Domains

Attribute domains are rules that describe the available values of a field type. They are used to constrain the values allowed in any particular attribute for a table or feature class. They provide a method for enforcing data integrity by limiting what can be placed on a field to a valid list or range of choices. If the features in a feature class or nonspatial objects in a table have been grouped into subtypes, you can assign different attribute domains to each of the subtypes. Whenever a domain is associated with an attribute field, only the values within that domain are valid for the field. In other words, the field will not accept a value that is not in that domain.

You can share attribute domains across feature classes, tables, and subtypes in a geodatabase. For example, a feature class for water mains and a feature class that stores water laterals can use the same domain for the ground surface type field.

Each row in the view is an existing domain and they all share common properties such as a name, description, field type, domain type, and split and merge policies.

When a domain is created in an enterprise geodatabase, the current connected user becomes the owner of the domain. Only the owner or the geodatabase administrator can modify the domain properties.

* The field type is the type of attribute field with which the domain can be associated.
* Short—Short integer (16-bit)
* Long—Long integer (32-bit)
* Big integer—Big integer (64-bit)
* Float—Single-precision (32-bit) floating-point number
* Double—Double-precision (64-bit) floating-point number
* Text (Coded domains only)—Alphanumeric characters
* Date—Date and time value  
  Date only—Date values only, with no time value

Time only—Time values only, with no date value

Domains on date field types support only whole second-based precision.

Domain type

There are two types of attribute domains:

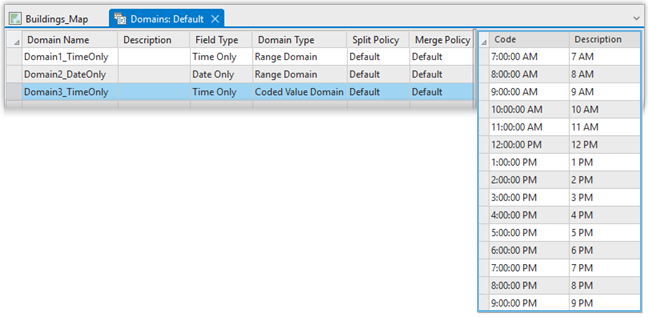
2- Range Domains—A range domain specifies a valid range of values for a numeric or date attribute data type. When creating a range domain, you provide a minimum and maximum valid value. You can apply a range domain to short integer, long integer, big integer, float, double, date, date-only and time-only field types.

Domain1\_TimeOnly—A range domain created for a Time Only field type, where any time value between 7:00:00 AM - 9:00:00 PM would be valid.

Domain2\_DateOnly—A range domain created for a Date Only field type, where any date values between 4/20/2023 - 5/16/2023 would be valid. A screenshot of a computer

Description automatically generated

2- Coded Value Domains—A coded value domain can apply to any attribute data type, be it text, numeric, date, and so forth. Coded value domains specify a valid set of values for an attribute.

Domain3\_TimeOnly—A coded value domain created for a Time Only field. Once this domain is applied to a Time Only field, only these hours appear in the list for a user to select and would be valid.  
Split and merge policies:  
  
Often, when editing data, a single feature is split into two features or two separate features are combined, or merged, into a single feature. For example, in a landbase database, a land parcel may be split into two separate land parcels due to rezoning. Similar zoning changes may require two adjacent parcels to be merged into a single parcel.

While the results of these types of edit operations on the feature's geometry are predictable, their effects on the attribute values are not. The behavior of an attribute's values when a feature is split, if that attribute has a domain applied, is controlled by the domain's split policy. When two features are merged, any attribute with a domain applied will have its value controlled by the domain's merge policy.  
  
  
Each attribute domain has both a split policy and a merge policy. When a feature is split or merged, the geodatabase looks to these policies to determine what values the resulting feature or features have for a particular attribute.  
 Split policies

Default value—The attributes of the two resulting features take on the default value for the attribute of the given feature class or subtype.

Duplicate—The attribute of the two resulting features takes on a copy of the original object's attribute value.

Geometry ratio—The attributes of resulting features are a ratio of the original feature's value. The ratio is based on the ratio in which the original geometry is divided. If the geometry is divided equally, each new feature's attribute gets one-half of the value of the original object's attribute. Geometry ratio policies only apply to domains for numeric field types.

Merge policies

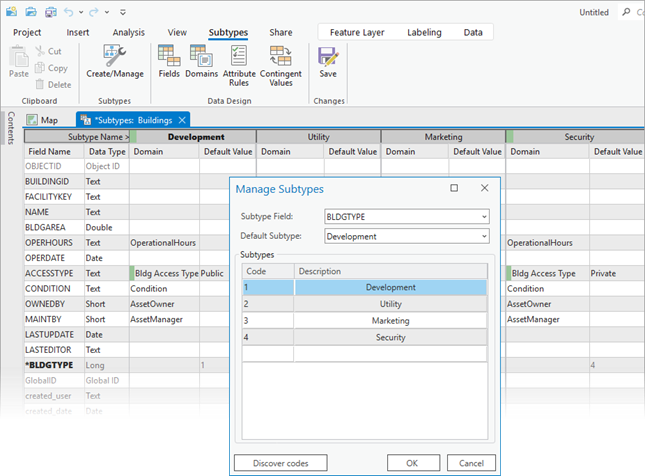
Default value—The attribute of the resulting feature takes on the default value for the attribute of the given feature class or subtype. This is the only merge policy that applies to nonnumeric fields and coded value domains.

Sum values—The attribute of the resulting feature takes on the sum of the values from the original features' attributes.

Geometry weighted—The attribute of the resulting feature is the weighted average of the values of the attributes from the original features. This average is based on the original feature's geometry.

Subtypes

Subtypes are a subset of features in a feature class, or objects in a table, that share the same attributes. They are used as a method to categorize your data. For example, the streets in a city streets feature class could be categorized into three subtypes: local streets, collector streets, and arterial streets.



Subtypes allow you to do the following:

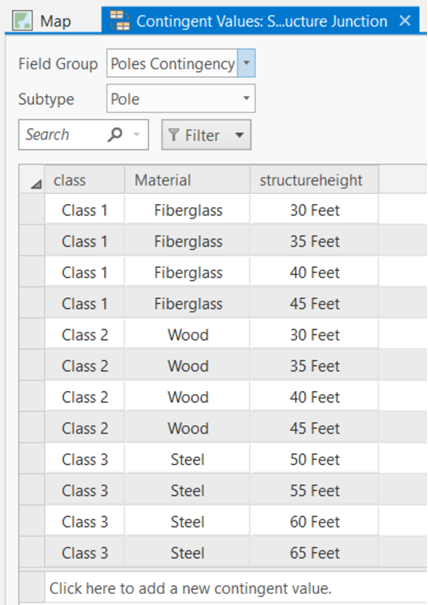
* Set default values on fields in each of the subtypes that will automatically apply when creating new features.
* In the above example, the local streets subtype could have the default value for a speed limit attribute set to 25 miles per hour, while the default value for the arterial streets subtype could be 35 miles per hour. Whenever a local street is added to the streets feature class, its speed limit attribute will automatically be set to 25 miles per hour. Whenever an arterial street is added, the speed limit will automatically be set to the default of 35 miles per hour.
* Apply coded or ranged domains to the fields of a subtype, enabling you to constrain input information to a valid set of values.  
  For example, in a feature class for water mains, a range domain for water pressure can be implemented. The subtype representing transmission water mains can have a pressure between 40 and 100 psi, while the subtype for distribution water mains can have a pressure between 50 and 75 psi.

* Each subtype may have different connectivity, relationship, or topology rules associated with it.
* Increase the performance of the geodatabase by representing a variety of real-world objects as a subset of features in a given feature class instead of creating new feature classes for each object.

A screenshot of a computer

Description automatically generated

contingent values

Contingent values, sometimes referred to as contingent attribute values, are a data design feature that allows you to make values in one field dependent on values in another field. Attribute domains allow you to restrict the valid entries in a field to a set list or range of values. Contingent values extend this model by making your choice for a value on one field further constrain the domain values that can be placed on another field. In this way, contingent values enforce data integrity by applying additional constraints to reduce the number of valid field inputs. This can be useful when modeling real world parts and assets in a GIS, particularly for utility companies. This can also be useful for creating decision trees or survey questions when making a choice will determine the next set of valid choices.  


**Contingent Values provides the following buttons to create and manage contingent values:**

New—Adds a row to create contingent values on the fields from an active field group in the view.

Import—Imports field groups and contingent values from a comma-separated values (.csv) file.

Export—Exports field groups and contingent values to a .csv file.

Retire Value—Allows you to set the highlighted row as a retired contingent value. Retired values are still visible in other places in the user interface, such as a fields value in the Attribute pane, but they are not selectable.

Toggle Value IDs—Displays or hides the Value ID value in the data grid.

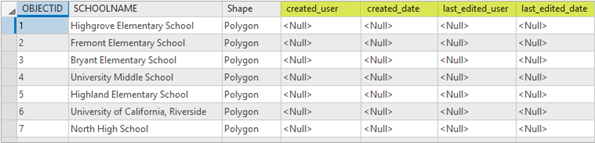
Toggle Code/Description—Switches the display between domain code or description in the data grid.

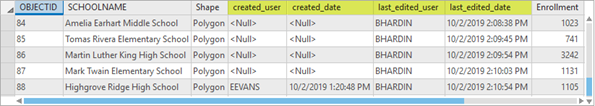
Save—Saves all changes made in the Contingent Values view. Changes are kept in memory until you click Save. A screenshot of a computer

Description automatically generated  
  
  
  
Editor Tracking  
Editor tracking provides a setting on feature classes and tables that allows you to automatically record information about any inserts and updates that are made. It maintains a record of the editor who created or modified the data and a time stamp of when the edit occurred. Many organizations find editor tracking helpful to maintain accountability and transparency as well as to enforce quality control standards

When editor tracking is enabled in ArcGIS Pro on a dataset in a file, mobile, or enterprise geodatabase, the attribute table is updated to include fields designated to track editing information. When a feature is created, editor tracking records the name of the user who created the feature along with the date and time of creation. When existing features are updated, the name of the user who made the edit, along with the date and time the edit was made, will be recorded.

* When editor tracking is enabled using default settings, the following four field names and field data types are added automatically:
* created\_user (Text)
* created\_date (Date)
* last\_edited\_user (Text)
* last\_edited\_date (Date)





UTC (recommended)—When editor tracking is enabled on a dataset using predefined settings, the date and time recorded when either the feature or record is created or updated is set to record in UTC by default.

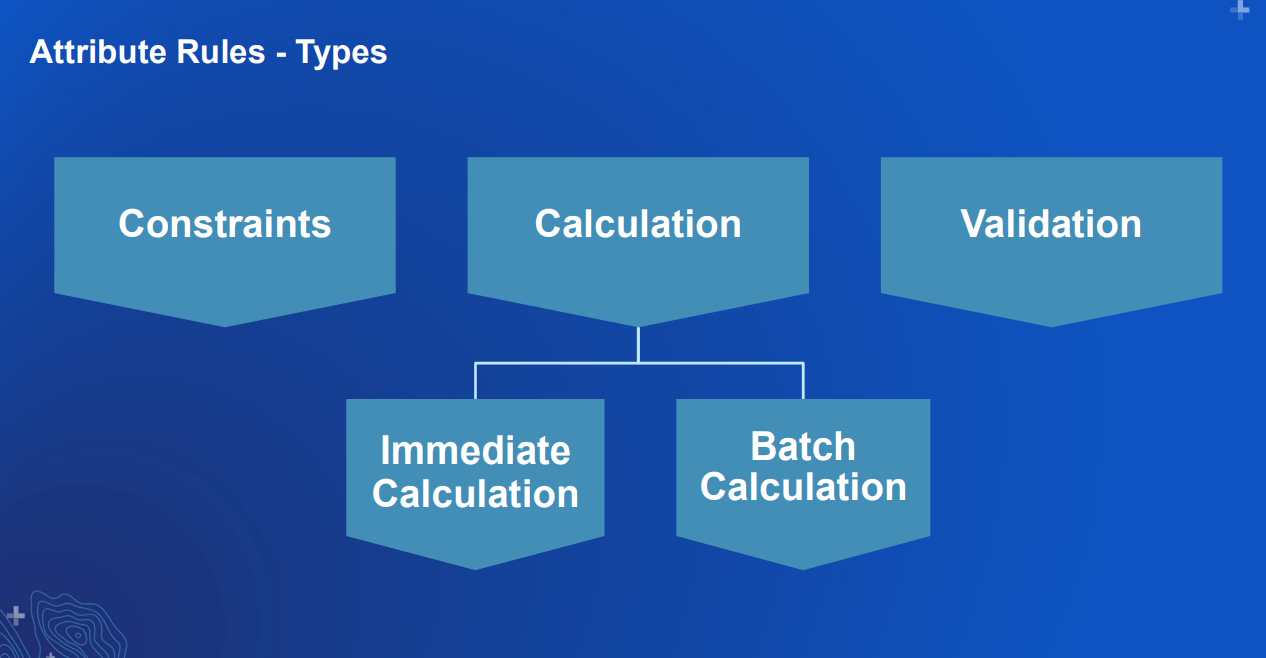
Attribute Rules

Attribute rules enhance the editing experience and improve data integrity for geodatabase datasets. They are user-defined rules that can be used to automatically populate attributes, restrict invalid edits during edit operations, and perform quality assurance checks on existing features.  
  
Attribute rules are complementary to existing rules used in the geodatabase, such as domains and subtypes. For example, domains can be assigned to an attribute field to aid in the data collection process by providing a list of valid values for editors. Additionally, an attribute rule can be used to restrict values for an attribute field that are not part of the domain when performing a field calculation. After rules are added to a dataset, they can be evaluated as edits take place or at a later time.

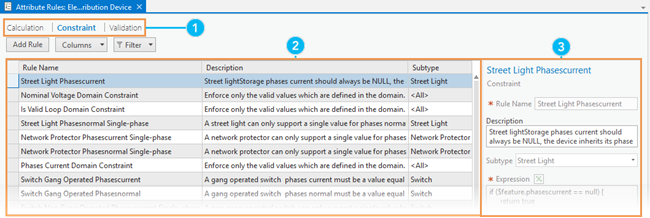
The following are examples of how attribute rules can be used in a geodatabase:

* To save time during editing, specific attribute values are automatically populated for newly created features. For example, when a pole is created, the assetID attribute is generated based on a sequence.
* To help enforce data integrity, edits can be blocked based on a user requirement. For example, when creating or updating a single phase transformer, the phasescurrent attribute must be A, B, or C.
* Populate specific attribute values for existing features at a user-specified time. For example, the fulladdress field for existing features is populated by concatenating several text fields.
* Review existing features to ensure that user requirements are maintained after field edits are made. For example, existing features can be evaluated to ensure that when a pole is taller than 65 feet, the material must be steel. Features that don't pass the evaluation are flagged as error features to be reviewed.

Rule types  
  
When you create an attribute rule, you must specify the rule type to use. The attribute rule type chosen depends on the task and at what point in the editing process the rule needs to be evaluated.

- Review the following rule type details to determine the applicable rule type for your workflow:  
  
Calculation—To increase efficiency in data collection and editing, immediate calculation rules automatically populate attributes based on editing patterns. This enhances the editing experience as new features are created or as changes are made to existing data. Batch calculation rules are evaluated at a user-specified time on existing data.

Constraint—To ensure quality data is entered into the GIS, constraint rules control data entry by user-defined requirements. This prevents invalid data entry while editing to ensure data integrity. To find data inconsistencies of existing data, use validation rules.

Validation—To review features that violate specified attribute or geometry requirements, validation rules highlight errors in existing data. This brings awareness to affected features to improve data integrity.  
  


**Manage attribute rules:**

Delete Rule—Delete the selected rule in the view.  
Import Rules—Import attribute rules from a .csv file.  
Export Rules—Export attribute rules to a .csv file.  
Enable Rule—Enable the selected rule in the view.  
Disable Rule—Disable the selected rule in the view.  
Save—Commit edits made in the view.

A screenshot of a computer

Description automatically generated

ArcGIS Data Reviewer checks  
ArcGIS Data Reviewer checks are tools that allow you to evaluate your data based on specific conditions. Some checks allow you to search for conditions, such as polygon slivers or cutbacks, while other checks allow you to search for features that have spatial relationships with each other. For example, you can find cutbacks on route features that impact measurement values in pipes or find buildings that have been mistakenly placed on water features.

**The following tables describe the Data Reviewer checks that support constraint and validation workflows:**

Attribute checks

Domain

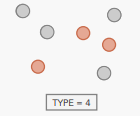
Finds attribute values that do not comply with coded value or range domains that are associated with an attribute field

A diagram of a company

Description automatically generated with medium confidence

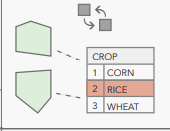
Query Attributes

Finds records based on a WHERE clause run against row attributes



Regular Expression

Finds features and rows that contain text values that do not match a defined patter



Relationship

Finds rows in feature classes and stand-alone tables that violate cardinality or relationship rules defined in a relationship class

A screenshot of a computer screen

Description automatically generated

Subtype

Finds features in a subtype with improper or null attribute values

A screenshot of a computer

Description automatically generated

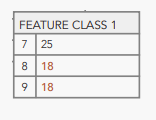
Table to Table Attribute

Finds features or rows that contain attribute values that meet a relationship you define and compares them to values in another feature class or stand-alone table

A screenshot of a table

Description automatically generated

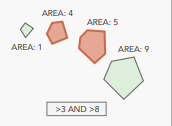
Unique Field Value

Finds rows in feature classes and stand-alone tables that contain nonunique values in an editable field or a list of fields  


POLYGON CHECKS

Evaluate Polygon Perimeter and Area

Finds polygon features based on the area or perimeter of the entire polygon or its individual parts or segments



Polygon Gap is Sliver

Finds gaps between polygon features that are below a specified thinness ratio

A colorful geometrical figure

Description automatically generated with medium confidence

Polygon Overlap is Sliver

Finds overlaps between polygon features that are below a specified thinness ratio

A colorful hexagons with different angles

Description automatically generated with medium confidence

Polygon Sliver

Finds polygon features with a thinness ratio below a specified threshold

A group of yellow and red shapes

Description automatically generated

Unclosed Polygon

Finds unclosed rings in polygon features

A yellow and red hexagon

Description automatically generated

Unnecessary Polygon Boundaries

Finds polygon features that share a common boundary and contain identical attribute values

A yellow hexagon with red line

Description automatically generated

Find Polygons with Holes

Finds polygon features that have holes

A group of colorful shapes

Description automatically generated

POLYLINE CHECK

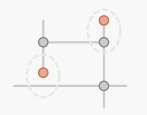
Evaluate Polyline Length

Finds polyline segments, parts, or features that have a line length within a specified tolerance



Find Dangles

Finds polylines that have dangles within a specified tolerance



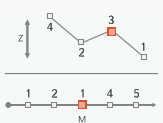
Find Disconnected Polylines

Finds polyline features that are not connected to other features in the same or other data source

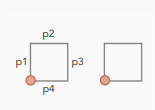
A diagram of a network

Description automatically generated

Monotonicity

Finds vertices in z-enabled or m-enabled polylines that are not strictly increasing or decreasing in value or are trending based on specified conditions  
  


Polyline or Path Closes on Self

Finds paths or lines on polyline features that close themselves  
  
  


Unnecessary Nodes

Finds polyline features that share a node and have matching attribute values

