



2

Creating a geodatabase

Tutorial 2-1

Building a geodatabase

Designing a geodatabase can be a long and drawn-out process, but it's only after that phase is completed that the data structure can be created in ArcCatalog. A good design will greatly simplify the creation phase and make it go more smoothly and quickly.

Learning objectives

- *Work with ArcCatalog*
- *Create a geodatabase*
- *Build a database schema*

Introduction

A good database design is essential to smooth creation of a geodatabase. It is best to think through the entire design, documenting your needs and addressing them with the logical model. Once that is completed, the creation phase can begin.

Databases can be very sensitive to change, and in fact, some elements, once created, cannot be changed. They can only be deleted, and then re-created. It is therefore very important when going through the creation tools to pay attention to what has been designed. Creating a feature class without regard to the data type will result in wasted time since the data type cannot be altered later. The same will hold true for field names and null allowances. Once created, these things cannot change.

The order of creating components is not necessarily as critical. Feature classes created outside a feature dataset can usually be moved into a feature dataset with a simple drag-and-drop operation. Field names left out of tables can usually be added later, or subtypes can be set up after the fact.

But even though some alterations can be made to the data structure later, it is best to have a complete logical model prepared and to follow it closely when creating the geodatabase. At this point, all opinions should have been heard and all aspects of the design completed. Major alterations to a data schema after the fact may require exporting the data to new feature classes, which could end in errors and confusion.

Creating the data structure

Scenario The City of Oleander has passed along completed geodatabase design forms and asked that you build the data structure for them. The forms show feature datasets, feature classes, fields, domains, and subtypes. They have some existing data that will be loaded into this schema later.

Data Print out the design forms provided. These are the completed geodatabase logical model diagrams from the previous tutorial.

Tools used ArcCatalog:

- New Personal Geodatabase
- New Feature Dataset
- New Feature Class
- Create a Domain
- Table to Domain

As you go through the creation process, pay close attention to all the steps involved. Sometimes setting even one option wrong can result in having to delete the entire piece and start over. The order of creation is not as critical. Some people may prefer to make all the domains first, then create all the tables, and follow that up with subtype creation. Others will create one feature class, its table, domains, and subtypes, and then move on to the next table. As long as the components are understood and created correctly, the order is not important.

Create the data structure

You are starting with a blank canvas and will be creating everything from scratch. The first step will be to create the geodatabase, which will contain all the rest of the components. This could be an SDE geodatabase or file geodatabase depending on your particular setup, but for ease and clarity, this tutorial will create a personal geodatabase.

- 1** Print the geodatabase design forms from tutorial 1-1. If you did not complete the tutorial, or if you are not confident in your answers, a completed set of forms can be found in the Data folder called Tutorial 1-1 GDB Design.

Geodatabase name		LandRecords
Feature dataset name		PropertyData
Feature classes:		
Type	Feature class name	Alias
POLY	Parcels	Property Ownership
L	LotBoundaries	Lot Boundaries

- 2 Start ArcCatalog and connect to the folder where you installed the tutorial data (for example, C:\ESRI\Press\GIST3\). Right-click the MyAnswers folder, select New, and click File Geodatabase.
- 3 Type the name of the database that's on the design form, **LandRecords**.

Create a feature dataset

From the design form, you can see that there will be a feature dataset in this geodatabase, with two feature classes. The geodatabase itself only needed a name to exist. The feature dataset needs two things: a name and a spatial reference.

The spatial reference is what ties your data to a location on the globe. You may need to decide if your data needs to be stored as projected or unprojected data, and you will need to define the spatial extent of your dataset. This dataset needs to cover the City of Oleander in North Central Texas and will use a spatial reference that is typical for the area. For more information on selecting the spatial reference for other datasets, see the section in ArcGIS Desktop Help titled “An overview of spatial references.”

- 1 Right-click the geodatabase, select New, and click Feature Dataset.
- 2 Type the name of the feature dataset as **PropertyData** and click Next.

The next screen is for setting the spatial reference. Oleander is in North Central Texas, and you will scroll through the lists for projected data to find the correct spatial reference. The one used for Oleander is Texas State Plane, NAD 1983 (feet), for North Central Texas.

- 3 Double-click Projected Coordinate Systems, then State Plane, then NAD 1983 (Feet), and then scroll down and find Texas North Central. Click Next and accept the default values for the next two screens, clicking Finish to complete the feature dataset.



Create the feature classes

Inside this feature dataset, you will create two feature classes. A feature class, at the minimum, needs three things: a name, a spatial reference, and a geometry type. You will provide the name from the design form. The spatial reference has been set at the feature dataset level, and the feature class will inherit this spatial reference. The final thing will be to set the geometry type. Pay close attention to set this correctly as it cannot be changed later.

- 1 Right-click the **PropertyData** feature dataset. Select **New**, and click **Feature Class**.
- 2 Type the name and alias. Then click the data type drop-down arrow, and click **Polygon Features**. Click **Next**, and then click **Next** again to accept the default option for **Configuration Keyword**. The groundwork for the feature class has been created. Next is to enter all the fields where the attributes will be stored. Refer to the tables worksheet for the names and configurations of the fields. As you enter each field, check the worksheet to get all the settings correct. Then highlight the row and double-check all the entries before moving on to the next field. This will help you avoid mistakes, which could cause you to have to delete the feature class and start over.

Tables worksheet						
Feature class or table name	Field name	Field type	Alias	Nulls (Y/N)	Default value	Domain name or subtype field (D) or (S)
Parcels	SubName	Text	Subdivision Name	No		
	Blk	Text	Block Designation	No		
	LotNo	Text	Lot Number	No		
	Pre_Type	Text	Prefix Type	Yes		
	Pre_Dir	Text	Prefix Direction	Yes		
	House_Num	Text	House Number	No		
	Street_Name	Text	Street Name	No		
	Street_Type	Text	Street Type	No		(D) StTypeAbbrv
	Suffix_Dir	Text	Suffix Direction	Yes		
	ZIPCODE	LI	ZIP Code	No		
	UseCode	Text	Land Use Code	No		(D) ParcelUseCodes
	Georeference	Text	Georeference	No		
	PlatStatus	SI	Plat Status	No		1 (S) PlatSubtype

2-1

2-2

- 3 On the first blank line, type the field name **SubName** and set the data type to Text. In the Field Properties pane, type the alias **Subdivision Name** and use the drop-down arrow to set “Allow NULL Values” to No. Be careful and methodical as you work your way through the field entry process. It is desirable to do this in one pass through the dialog box to avoid getting confused or forgetting to set a value. Once the Finish button is pressed, only the alias can be later changed. Mistakes must be deleted and the files constructed again.



- 4 On the next blank line, type the field name **BLK** and set the field type to Text. In the Field Properties pane, type the alias **Block Designation**, and use the drop-down arrow to set “Allow NULL values” to No.

YOUR TURN

Work your way down the list of fields from the tables worksheet and enter all the values for field name, data type, alias, and Allow NULL values. At this point, disregard the default value and domain/subtype entries. If you run out of blank lines for new fields, use the slider bar at the left to reveal more. When all the fields have been entered, click Finish.

The feature class has been created and added to the feature dataset. There's one more feature class to create for the lot boundaries. The process will be the same, but this feature class only has one attribute field, so it will go much faster.



- 5 Right-click the **PropertyData** feature dataset. Select **New** and click **Feature Class**.
- 6 Type the name and alias as shown on the design form. Then click the data type drop-down arrow and select **Line Features**. Click **Next**, and then click **Next** again to accept the default setting for **Configuration Keyword**. Refer to the tables worksheet for the names and configurations of the fields.

Tables worksheet						
Feature class or table name	Field name	Field type	Alias	Nulls (Y/N)	Default value	Domain name or subtype field (D) or (S)
LotBoundaries	LineCode	Text	Line Code	No		(D) ParcelLineCodes

- 7 On the first blank line, type the field name **LineCode** and set the field type to **Text**. Add the alias of **Line Code**, and use the drop-down arrow to set “Allow NULL values” to **No**. Click **Finish**.

Create the domains

Now you have both feature classes, with all their fields. The next step will be to introduce data integrity rules for the feature classes. This will start with creating the domains and then assigning them to the fields in the feature classes. Domains are stored at the geodatabase level, which means that all the required domains for all the feature classes residing in the geodatabase will be stored there. You may be working on a utility project, but there may be others with their own feature datasets and feature classes requiring dozens of domains. Because of this, there are two critical rules when working with domains.

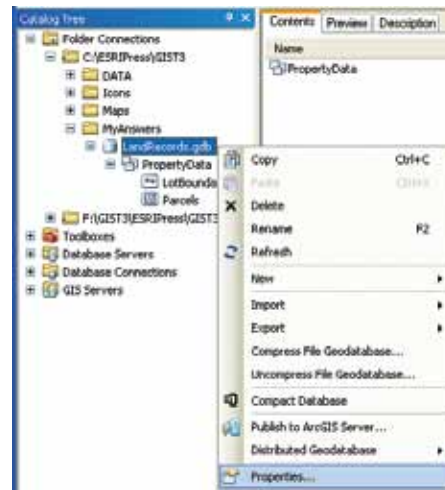
The first rule is to give your domain a very specific name, even if it means making it rather long. A domain called “Type” may be confusing, because it doesn’t describe its purpose very well. Is this a type of road, sign, tree, or something else? A better name might be “TypeOfSignpost,” if it were used to constrain the entry of signposts to certain types such as metal or wood.

The second rule is to never alter someone else’s domain to fit your needs. You may find a domain called “Material” with a list of wood, steel, fiberglass, and concrete. The domain you want may be the exact same list but without concrete, so you delete that entry and use it. The next time the dataset for which that domain was originally created is used, all the features with a material of concrete will be invalid with respect to the material domain because the entry is missing. If it gets changed back to make that data valid, the new dataset will become invalid when referencing that domain. To prevent this from happening, it’s best to always create your own domains.

The domains worksheet will be the guide for creating domains, so get that ready.

Database	Description	Field	Domain	Coded values	Code	Value
ParcelUseCodes	Use Codes for Parcels	Text	Coded values	A1	Single Family Detached	
				A2	Mobile Homes	
				A3	Condominiums	
				A4	Townhouses	
				A5	Single Family Limited	
				B1	MultiFamily	
				B2	Duplex	
				B3	Triplex	
				B4	Quadplex	
				CITY	Developed City Property	
				CITY	Recreational City Property	
				CITY	Water Utility Property	
				CH	Church	
				ESMT	Essement	
				F1	Commercial	
				F2	Industrial	
				GOV	Government	
				POS	Public Open Space	
				PRK	Park	
				PRW	Private Right-of-way	
				RW	Right-of-way	
				SCH	School	
				UTIL	Utility	
				VAC	Vacant	

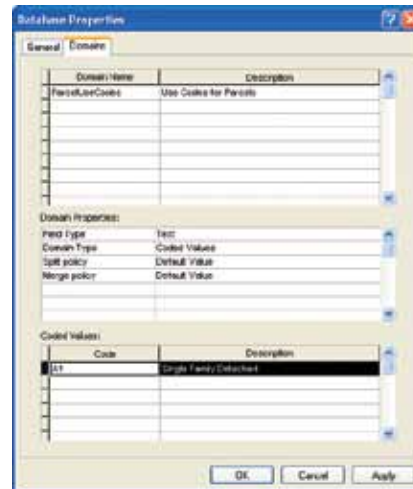
- 1 Right-click the geodatabase LandRecords, and then scroll down and click Properties.



- 2 Click the Domains tab. This is where domains are created. On the first empty line, type the domain name **ParcelUseCodes** and add the description **Use Codes for Parcels**.

- 3 In the Domain Properties pane, set the field type to **Text**. Remember that there are two types of domains. The first is coded values, which contains a list of values. When you set the field type to Text, the only choice is Coded Values, so it is set automatically. Numeric fields can also be used with the Coded Values choice. The other type of domain is a range. This lets you set a start and end value within which the entered data must fall, but it is only valid with numeric field types.

- 4 Next, move down to the Coded Values pane. On the first line in the Code column, enter the first value of **A1**. Next to it in the Description column, enter the value **Single Family Detached**. Click Apply to save the entry.



2-1

2-2

YOUR TURN

Using the domains worksheet, enter the rest of the values for code and description. Periodically, click Apply to save your work and make sure there are no errors in the domain. When you have completed the list, click OK.



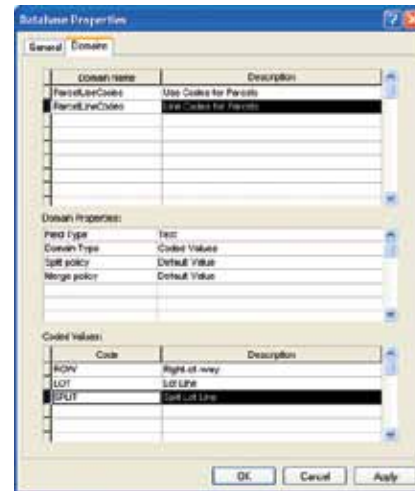
There are two more domains to make. One will use the same process as the use codes domain, and the other will use a process to import the values from an existing database. Create the parcel line codes domain first, since it uses the same dialog as the previous example.

5 Right-click the geodatabase LandRecords, select Properties, and click the Domains tab. You'll see the domain created earlier, as well as any other domains that anyone else may have made in this geodatabase. Click the first empty line and type the new domain name **ParcelLineCodes** as well as the description **Line Codes for Parcels**.

6 Move to the Domain Properties pane and set the field type to Text. The domain type will automatically set to coded values. In the Coded Values pane, enter the codes and descriptions for the domain:

- ROW = Right-of-way
- LOT = Lot Line
- SPLIT = Split Lot Line

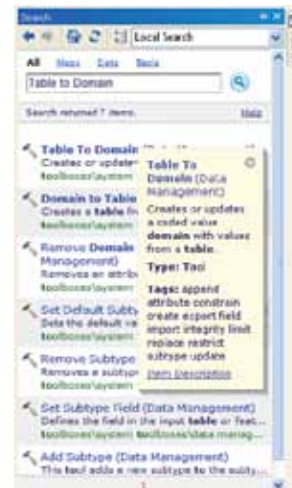
When all the values are entered, click Apply, and then click OK.



Create a domain from a table

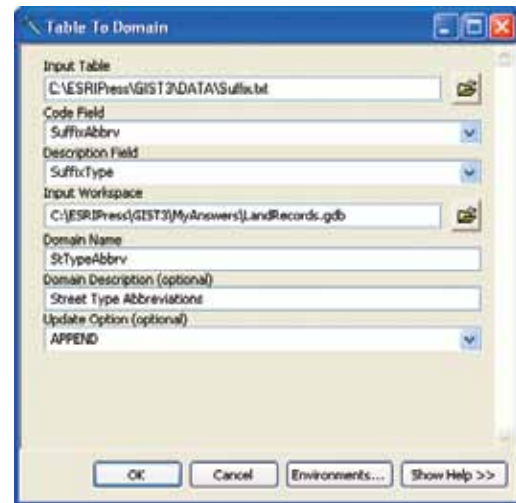
The next method of domain creation will use an existing database of street suffixes to fill in all the values. In the ArcToolbox interface, there is a tool that will read the table and transfer it into the domain you created in the geodatabase. The first step will be to find the tool.

1 In ArcCatalog, use the Search tool to locate the Table to Domain tool.

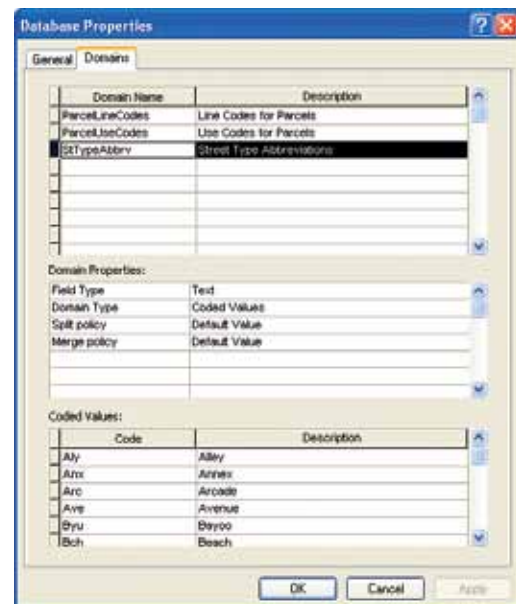


2 Double-click Table to Domain to start the tool. In the dialog box, click the Browse button next to Input Table. Browse through the C:\ESRIPress\GIST3\Data folder for the table Suffix.txt. Click the table, and then click Add.

- 3 In the Code Field input box, click the drop-down arrow and select SuffixAbbrv.
- 4 In the Description Field input box, select SuffixType.
- 5 Use the Browse button to set Input Workspace to the LandRecords geodatabase, and then click Add.
- 6 Finally, type the domain name as **StTypeAbbrv** and description as **Street Type Abbreviations**. When your dialog box matches the accompanying image, click OK. The domain was created with all the suffix values that are used by the U.S. Postal Service. This is a quick and easy option for creating domains that come from other data sources.



- 7 Open Properties of the geodatabase in ArcCatalog and examine the results.



Assign the domains

Now that all the domains are created, they need to be assigned to the fields that they will help control. The tables worksheet shows that these domains will be assigned to two fields in the Parcels feature class and one field in the LotBoundaries feature class.

Tables worksheet						
Feature class or table name	Field name	Field type	Alias	Nulls (Y/N)	Default value	Domain name or subtype field (D) or (S)
Parcels	SubName	Text	Subdivision Name	No		
	Blk	Text	Block Designation	No		
	LotNo	Text	Lot Number	No		
	Pre_Type	Text	Prefix Type	Yes		
	Pre_Dir	Text	Prefix Direction	Yes		
	House_Num	Text	House Number	No		
	Street_Name	Text	Street Name	No		
	Street_Type	Text	Street Type	No		(D) StTypeAbbrv
	Suffix_Dir	Text	Suffix Direction	Yes		
	ZIPCODE	LI	ZIP Code	No		
	UseCode	Text	Land Use Code	No		(D) ParcelUseCodes
	Georeference	Text	Georeference	No		
	PlotStatus	SI	Plot Status	No		1 (S) PlotSubtype
LotBoundaries	LineCode	Text	Line Code	No		(D) ParcelLineCodes

- 1 In ArcCatalog, open the properties of the Parcels feature class and click the Fields tab. Click the field Street_Type.
- 2 In the Field Properties pane, click the blank space next to Domain. Use the drop-down arrow and click StTypeAbbrv. Click Apply. Notice that when you select the domain, the drop-down list is by name and not description. This is another reason to give domains very explicit names instead of something less descriptive or generic like “Material” or “Type.”
- 3 Next, click the field UseCode, move down to the Domain box, and use the drop-down arrow to click ParcelUseCodes. Click Apply, and then click OK to close Feature Class Properties.

YOUR TURN

Apply the domain ParcelLineCodes to the LineCode field in the LotBoundaries feature class.

Set up subtypes

It was recognized earlier that there would be a benefit to having subtypes in the Parcels feature class. A subtype can show the plat status and allow updates to be done more efficiently. If you are still unsure of the role of subtypes, the next section will demonstrate how they can be used to good effect.

Refer to the subtypes worksheet for the information that will be included in the subtype entry screen.

Subtypes worksheet			PRESET DEFAULTS		
Subtype name	Code	Description	Field	Domain name	Default value
PlatSubtype	1	Platted Property			
	2	Unplatted Property			
	3	Plat Pending			

- 1** In ArcCatalog, right-click the new feature class Parcels and select Properties. Click the Subtypes tab. Click the drop-down arrow for Subtype Field and select PlatStatus.
- 2** Move down to the Subtypes input area. Replace code 0 with a 1 and type the description of **Platted Property**.
- 3** Fill in the remaining entries according to the subtypes worksheet:
 - 2 = Unplatted Property
 - 3 = Plat Pending

When your dialog box matches the image to the right, click Apply, and then click OK.



In this example, there were no additional default or domain values to set up unique to the subtype setting. Pressing Apply was a good way to check if there were any problems with your entries. If an error message had been returned, it would have instructed you how to correct the error and proceed.

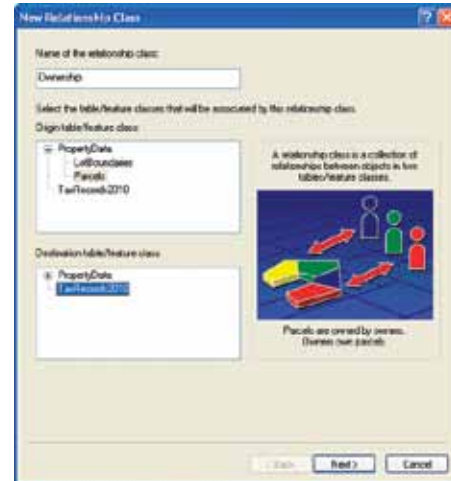
Define the relationship class

You designed a relationship class earlier that will link the parcel polygons to an external table of ownership data. This relationship class will be built in the geodatabase for land records, and it can only access data stored within this geodatabase. Review the design form for relationship classes before proceeding.

- 1 Import the table `TaxRecords2010` from the install folder (for example, `C:\ESRIPress\GIST3\Data\`) into your new `LandRecords` geodatabase by right-clicking the geodatabase name, pointing to `Import`, and clicking `Table (single)`.
- 2 Set `Input Rows` to `TaxRecords2010` and `Output Table` name to the same, saving it in your `MyAnswers` folder. Click `OK`.

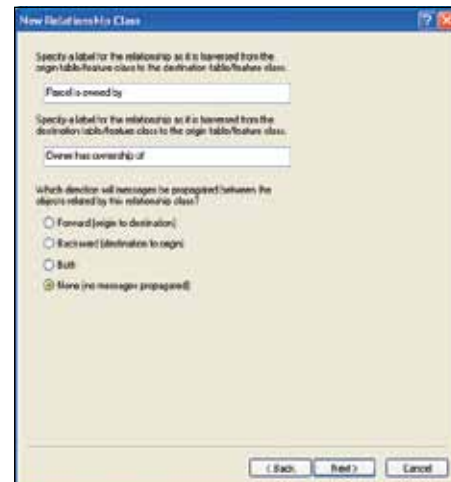
- 3 Start the creation of the relationship class by right-clicking the `LandRecords` geodatabase, selecting `New`, and then clicking `Relationship Class`.

- 4 Using the information from the geodatabase worksheets, enter the name of the relationship class as **Ownership**, select **Parcels** as the origin feature class, and **TaxRecords2010** as the destination table. Click Next.



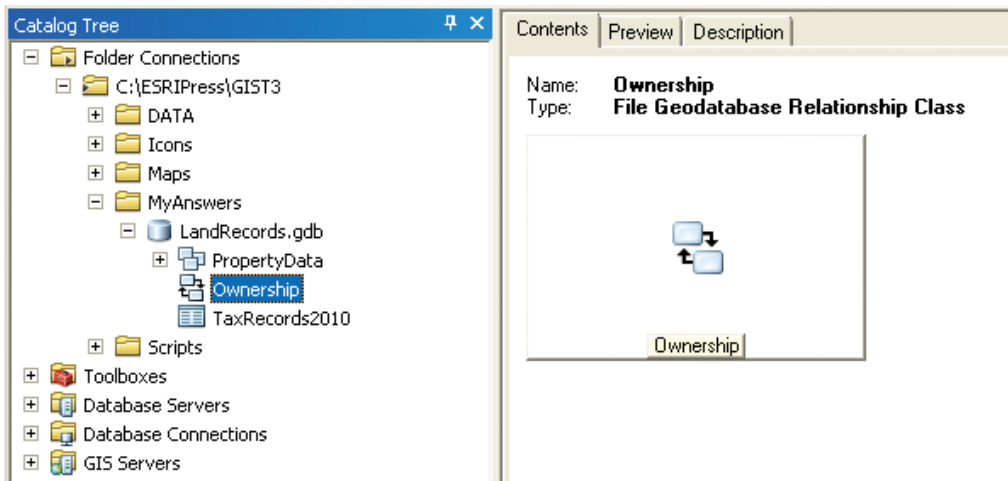
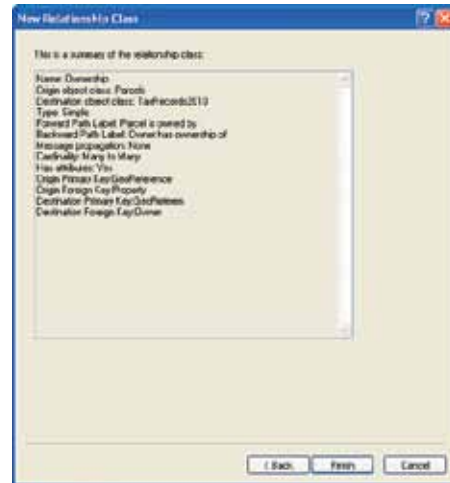
- 5 Set the relationship type to Simple (peer to peer). Click Next.

- 6 Type the labels that will describe the two relationships—**Parcel is owned by** for the forward path (as you traverse from the origin table), and **Owner has ownership of** for the backward path (as you traverse from the destination table). Set the message condition to None. Click Next.



- 7 Set the relationship cardinality to M-N (many to many). Click Next.
- 8 Select the option indicating that you want to have attributes for this relationship. Click Next.
- 9 Type the new field name **PercentOwn** with a data type of Float, as recorded on the design form. Set Field Properties to allow for null values. Click Next.

- 10** Use the two drop-down lists to set the origin and destination primary key fields to **Georeference** and **GeoReferen**, respectively. Then type the labels for the foreign key fields from the design form (see tutorial 1-1)—**Property** for the origin table and **Owner** for the destination table. Click **Next**.
- 11** The last screen is a summary of the values you have entered. Review them for completeness, and then click **Finish**. The new relationship class has now been created in your geodatabase. This links the feature class **Parcels** with the table **TaxRecords2010** so that the information from the table can be used for analysis and symbolized through the parcels. As new tax rolls are released, more relationship classes can be built allowing for comparisons of different tax years to be displayed in the **Parcels** feature class.



- 12** Close ArcCatalog.

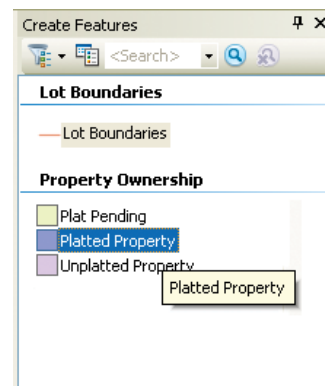
Test the subtypes

To get a better idea of the value of subtypes and the other data integrity rules, it would be beneficial to test them in ArcMap.

- 1 Start an ArcMap session and open Tutorial 2.1.mxd.
- 2 Click the Add Data button, browse to the LandRecords geodatabase, click the PropertyData feature dataset, and then click Add. When a feature dataset is added to a map document, all the feature classes in the feature dataset are added. Since these feature classes are empty, no features will be displayed in the map area. However, the table of contents will show some interesting things.

The two feature classes Lot Boundaries and Property Ownership now appear in the table of contents. Normally when a feature class is added to a map document, it is symbolized with a single symbol. In the case of a feature class with a subtype, however, the data is automatically classified according to the subtype field. Notice that the Property Ownership layer shows the three categories defined as subtypes.

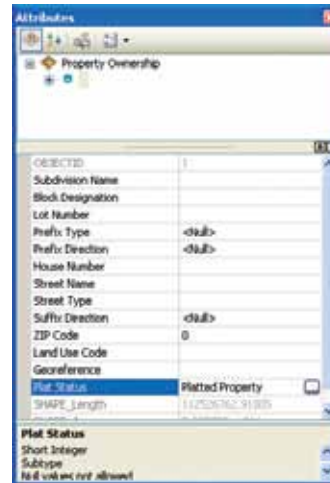
- 3 From the Editor toolbar, select Start Editing. When the edit session is started, ArcMap creates a new editing menu called the Create Features template. All the feature classes will appear in this window shown with their classifications. To edit, simply click the feature you wish to work with, select an editing tool, and then begin working in the map area.
- 4 In the Create Features template, select Platted Property.



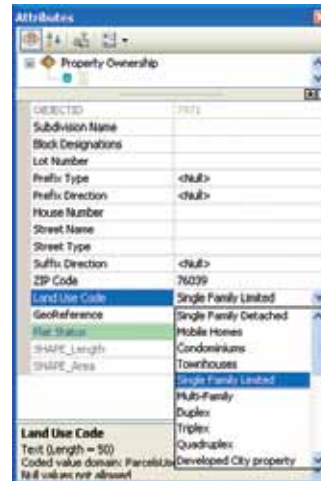
2-1

2-2

- 5 Try drawing a few polygons in the map area, and then click the Attributes button on the Editor toolbar. Notice that only the Plat Status field is populated.



- 6 Click the empty box next to the Land Use Code field. Notice the drop-down list. Its values come from its domain. Also look at the domain for Street Type. With Platted Property selected in the Create Features template, all new features drawn will automatically be put in the Platted Property subtype and all the unique data integrity rules you set up will be in effect. Without subtypes, you would have to go into the attribute table and change each entry as it is drawn.



The next tutorial will build more feature classes with data integrity rules. These will be tested to give you an even better idea of the role of subtypes and domains.

- 7 Close ArcMap without saving your edits or the map document.

Exercise 2-1

The tutorial showed how to create a geodatabase and all its components from the logical model. This included a feature dataset, feature classes, domains, and subtypes.

In this exercise, you will repeat the process using the logical model for zoning data created in exercise 1-1.

- Start ArcCatalog and locate the MyAnswers folder.
- Use the geodatabase worksheet to create the geodatabase.
- Use the tables worksheet to create the feature classes with the correct fields.
- Use the domains worksheet to create and apply the domains.
- Use the subtypes worksheet to create the subtypes.

2-1

2-2

WHAT TO TURN IN

If you are working in a classroom setting with an instructor, you may be required to submit the materials you created in tutorial 2-1.

Screen capture of ArcCatalog showing all the components of the completed geodatabases

Tutorial 2-1
Exercise 2-1

Tutorial 2-1 review

Any model of reality is only as good as the foundation on which it rests. This tutorial illustrated the development of a geodatabase using a systematic approach to design that documents the uses and relationships of the geographic features of parcels in the City of Oleander.

The tutorials in this book deal mainly with the file geodatabase, but there are other types of geodatabases that you should be aware of. The **file geodatabase** was introduced in ArcGIS 9.2 and is stored as a system of folders containing files. Perhaps the biggest advantages to a file geodatabase are that it is cross-platform compatible; the available file sizes can extend up to one terabyte; and the database can be processed faster. The **personal geodatabase** uses a Microsoft Access database, and consequently, tables can be read in Microsoft Access or Excel. Personal geodatabases are typically used in smaller organizations, where versioning is not required; there are fewer users of the data; and file sizes do not exceed two gigabytes. One disadvantage to the personal geodatabase is that because it is based on Microsoft Access, it can only be used in a Microsoft environment. The **SDE**

geodatabase is cross-platform compatible and can be operated using a variety of DBMS formats. (Check ArcGIS Desktop Help for a current list of supported formats.) Typically used in larger organizations, an SDE geodatabase can support many editors editing the geodatabase simultaneously and efficiently manage these edits through versioning. The SDE geodatabase requires ESRI ArcSDE software to help translate geoprocessing commands to the database. The logical model that you design can be used to build any of these types of geodatabases.

Through the use of **geodatabase design forms** to design your geodatabase, specific representations and necessary relational behavior among geographic features can be clearly represented. The use of the design forms provides a **logical model** of the geodatabase and focuses the developer's planning throughout the development process to ensure adequate representation of these features inside the geodatabase. As such, the development of a logical model is a central aspect of any geodatabase planning phase.

ArcCatalog was used to organize data in **feature datasets**, **feature classes**, and **subtypes** within the geodatabase. You can picture the hierarchical relationships of the geodatabase, feature datasets, and feature classes as a filing cabinet, where the geodatabase is the drawer of the filing cabinet, the feature dataset is the folder that resides in the drawer, and the feature classes are the individual pieces of paper within the file folder. You can assume that an organized relationship exists among these components in a filing cabinet if you ever want to find a document filed away. You can also assume that documents that are similar to each other, or that must support each other, are filed in the same folders. Remember, the rules of the geodatabase require that all feature classes supporting a topology or network must exist within the same feature dataset.

Data integrity and consistency issues within the LotBoundaries and Parcels feature classes of the geodatabase were addressed through the development of **domains** for several features. Proper use of domains in the beginning of design can save hundreds of hours of data cleanup from mistakes made during the data entry phase and countless hours of interpreting nonstandardized entries of data. Although domains are specifically used by feature classes, they are established at the geodatabase level to ensure that one central place will contain all domains and be available for all feature classes. Once a domain is created for a field within a feature class, the likelihood of making mistakes when entering data into this field is reduced drastically, thus improving data integrity and consistency issues within the geodatabase.

During the initial planning phase of geodatabase development, you should examine the relationships among all feature classes and the features they contain with respect to their common behaviors and characteristics. Typically, you want to optimize (minimize) the number of feature classes contained in your geodatabase, while still capturing all the unique characteristics of each feature. This opportunity for optimizing your geodatabase is one of the advantages of using **subtypes** within a feature class. Creating subtypes for a feature class based on certain defining characteristics of features within the feature class essentially allows you to build in the flexibility of an additional feature class without having to build and store one within your geodatabase. The City of Oleander used subtypes to classify the parcels it manages according to the current status of the plat. The status of a parcel with respect to platting is critical to supporting the city's development activities.

You could have created a separate feature class for each plat status. You would simply create a feature class for unplatted parcels, a separate feature class for platted parcels, and a third feature class for plat pending parcels. However, by recognizing that platted, plat pending, and unplatted characteristics of a parcel are simply a specific state of the evolution of the parcel in the land development process, you created a geodatabase with a subtype that would differentiate each parcel on the basis of its status. All three of these subtypes are still types of parcels. However, now they have been categorized in the geodatabase by a subtype. Regardless of what features you intend to store in a geodatabase, as long as the primary behaviors among features are consistent, considering a subtype instead of creating a separate feature class is a good choice.

2-1

2-2

This tutorial used ArcCatalog to develop subtypes for the Parcels and LotBoundaries feature classes and illustrated another advantage of using subtypes when presenting your data to the user. When you displayed your Parcels feature class in ArcMap, each feature class was automatically classified by these subtypes. Again, rather than adding three separate feature classes to display information on plat status within Oleander, ArcMap used the subtypes from this one feature class to automatically classify features based on subtype. This is a more efficient way of working with and viewing the data. As you will see in later tutorials, the creation of subtypes gives you additional advantages when editing your data. Subtypes created within a feature class will allow each subtype to be a separate target feature during an edit session. This gives the user the ability to use unique default values for each subtype and select unique values for each subtype from the supporting domain values. As you can see, a little thinking ahead regarding the use of subtypes and domains can help not only organize the behavior and characteristics of the features contained in your geodatabase, but also provide many advantages when displaying, editing, and populating the geodatabase with data.

STUDY QUESTIONS

1. List three advantages of using domains in your geodatabase. At what level within the geodatabase are these domains established, and why are they created at this level?
2. Why was it necessary to create separate feature classes for the parcels and the lot boundaries? Could this characteristic have been represented by a subtype?
3. How does the establishment of default values within the table of a feature class benefit the operation of the geodatabase?
4. How can importing domains for a feature class benefit the geodatabase developer? Provide an example of suitable information that could be imported into a domain.

Other real-world examples

The use of domains and subtypes to refine the management of spatial data within a geodatabase can be beneficial to anyone needing to develop a geodatabase that accurately portrays geographic features and their associated behaviors. State and local agencies use domains for roadways to help standardize data entry and enforce data integrity for specific types of paving materials, which could include a choice between asphalt and concrete. Water utilities use domains to help standardize entries for fields that contain pipe size and pipe materials.

Subtypes can be used as a data management tool to help refine the classification of features. A roadway feature class, for example, can be organized to optimize maintenance planning according to the type of material. Subtypes can be used to create a default classification of the roads based on asphalt or concrete construction that would be important to help determine different maintenance programs for each type of road. Default values and domains can then be tailored to the unique demands of each roadway subtype. Using subtypes helps to create more definition in the features without having to create additional feature classes.

Power utilities can create subtypes within the geodatabase for different types of power poles—whether metal or wood. Although each type of pole has the same primary behavior of supporting power lines, transformers, and so forth, they vary by material and capacity, and consequently these subtypes will have implications for maintenance and planning.

Tutorial 2-2

Adding complex geodatabase components

2-1

2-2

The logical model included several techniques for simplifying data entry and managing data integrity. These are included in the geodatabase when it is built in ArcCatalog. If time is taken to design it well in the beginning, the long-term benefits will be great.

Learning objectives

- Work with ArcCatalog
- Create a geodatabase
- Build a database schema

Introduction

Tutorial and exercise 2-1 worked with polygon feature classes. These have unique characteristics in their database design. The use of domains and subtypes helped build in certain data integrity rules.

This tutorial will also focus on setting up feature classes with domains and subtypes but will work with linear and point data types. The process of setting up and using them is very similar to the polygons used previously, so the tutorial will move rather quickly. When everything is set up, however, a more extensive demonstration of the purpose of the data integrity rules will be presented.

Adding to the data structure

Scenario You've completed designing and building the database for the parcel and zoning data for the City of Oleander. Now it's time to create the geodatabase structure for the sewer line designs that were also written earlier.

- 3** Go to the Wastewater feature dataset and start the creation process for the SewerLines feature class, being very careful to set the feature type correctly. In the field name dialog box, enter all the fields from the design form. Pay close attention to the Allow Nulls setting and any default values.

Tables worksheet						
Feature class or table name	Field name	Field type	Alias	Nulls (Y/N)	Default value	Domain name or subtype field (D) or (S)
SewerLines	PipeSize	SI	Pipe Size	No		(D) SewerPipeSize
	Material	SI	Pipe Material	No		(S) SewerLineMaterial
	YearBuilt	LI	Year Built	Yes		
	Description	Text	Owner	No		

Remember that the domains and subtypes cannot be assigned yet, because they have not yet been created. In the next feature class, the fields are the same as in the SewerLines feature class. There is an import feature that will make filling out the fields dialog box very easy.

Import field definitions

The next feature class on the design form is for interceptors.

Tables worksheet						
Feature class or table name	Field name	Field type	Alias	Nulls (Y/N)	Default value	Domain name or subtype field (D) or (S)
Interceptors	PipeSize	SI	Pipe Size	No		
	Material	Text	Pipe Material	No		
	YearBuilt	LI	Year Built	Yes		
	Description	Text	Owner	No		

- 1** Go to the Wastewater feature dataset again and begin creating the Interceptors feature class. Enter the name and description from the design form and click Next twice. Instead of typing in all the field names, click the Import button to retrieve the fields from another file.

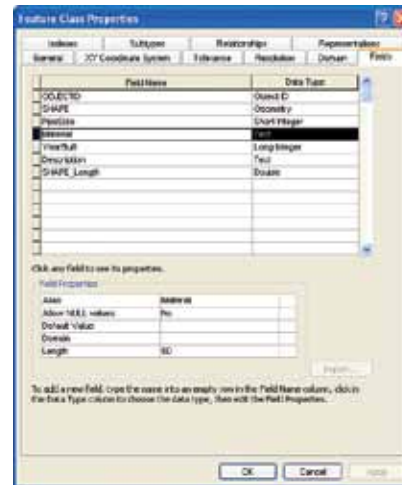


2-1

2-2

2 Browse to the SewerLines feature class, select it, and click Add.

3 Consulting the design forms, you will see that the field type for Material is Text. Change the field type from Short Integer to Text. Check that the alias and null settings are correct, and then click Finish to complete the feature class. Once again, the Import button was a great shortcut. You saw, however, that one of the field types was different from the imported data structure. This can be changed while you are working in the feature class creation screens, but once you apply the changes, they cannot be changed again. It is interesting to note that field structures can be imported.



Continue the creation process

Now move on to the feature class for the sewer fixtures.

Tables worksheet						
Feature class or table name	Field name	Field type	Alias	Nulls (Y/N)	Default value	Domain name or subtype field (D) or (S)
SewerFixtures	FixType	SI	Fixture Type	No		(S) SewerFixType
	FlowLine	Float	Flow Line Elevation	Yes		
	RimElev	Float	Rim Elevation	Yes		
	Depth	Float	Depth from Surface	Yes		
	YearBuilt	LI	Year Built	Yes	2010	
	Description	Text	Owner	No	Oleander	

1 In the Wastewater feature dataset, create the SewerFixtures feature class. Add the fields and their parameters from the design worksheet. Properties of a field include its data type, alias, nulls flag, default values, and domains. When everything is correct, click Finish.

The final feature class to create is for the interceptor fixtures. Notice that it has four of the same fields as the sewer fixtures.



Tables worksheet						
Feature class or table name	Field name	Field type	Alias	Nulls (Y/N)	Default value	Domain name or subtype field (D) or (S)
InterceptorFix	FlowLine	Float	Flow Line Elevation	Yes		
	RimElev	Float	Rim Elevation	Yes		
	Depth	Float	Depth from Surface	Yes		
	YearBuilt	LI	Year Built	Yes	2010	

2 Once again, go to the Wastewater feature dataset and create a new feature class for InterceptorFix. In the Field Name dialog box, click Import and transfer all the fields from the feature class SewerFixtures.

3 Two of the fields are not necessary. Remove them by clicking the gray box to the left of their name and pressing Delete. Once they are both removed, click Finish. Now is a good time to go back and check your work. Did you create all the feature classes as the correct feature types? Did you get all the Allow Null Values settings correct? Did you spell all the field names correctly? If not, now is the time to delete the feature class and start again as none of these things can be changed. Also, check field alias names and required default value settings. If any of these are incorrect, they can be changed now.



2-1

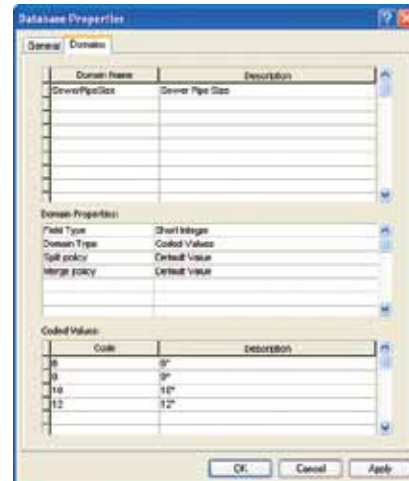
2-2

Create domains

Move on to the domains worksheet. There is only one to create. Notice that the field type is Short Integer and the codes are integers. But the descriptions are text with inch marks included. Both of these formats will be saved in the domain, and both are available for use in a map document. For example, when a label is created to show pipe size, you will have the choice of using 12 or 12".

Domains worksheet							
Domain name	Description	Field type	Domain type	Coded values / Range			
				Code	(Min)	Desc	(Max)
SewerPipeSize	Sewer Pipe Size	SI	Coded values	6		6"	
				8		8"	
				10		10"	
				12		12"	

1 Open the properties of the Utility Data geodatabase and click the Domains tab. On the first blank line, enter the name of the domain and its description and adjust the Domain Properties. Proceed to the Coded Values pane and enter the other parameters and codes for the domain. Click Apply, and then click OK. This pipe size domain will be assigned to the PipeSize field in the SewerLines feature class.



2 Open the properties of the SewerLines feature class and click the Fields tab. Click the PipeSize field and in the Field Properties pane, use the Domain drop-down arrow to assign the SewerPipeSize domain. Click Apply, and then click OK.



Create subtypes

The last of the data integrity rules involve the subtypes. Review the subtypes worksheet.

Subtypes worksheet			PRESET DEFAULTS	
Subtype name	Code	Description	Field	Default value
SewerLineMaterial	1	P.V.C.	PipeSize	SewerPipeSize
			Description	Oleander
	2	HDPE	YearBuilt	2010
			PipeSize	SewerPipeSize
			Description	Oleander
3	DI	YearBuilt	2010	
		PipeSize	SewerPipeSize	
		Description	Oleander	
4	Conc	YearBuilt	2010	
		Description	Oleander	
5	Clay	YearBuilt	2010	
		Description	Oleander	
SewerFixType	1	Manhole		
		Cleanout		

The second subtype is less complex, so for learning purposes, you should create that one first. Then a more detailed description will follow to create the more complex subtype.

- 1 Right-click the **SewerFixtures** feature class, click **Properties**, and then click the **Subtypes** tab. Use the drop-down arrow to set the subtype field to **FixType**.
- 2 Change the first code from 0 to **1** and enter the description **Manhole**. Add a second code of **2** with a description of **Cleanout**. Click **Apply**, and then click **OK**.

Now to tackle the tougher subtype. This one has codes and descriptions like the other subtype but also includes a domain and default setting for each subtype code.



- 3 Right-click the **SewerLines** feature class, click **Properties**, and then click the **Subtypes** tab. Use the drop-down arrow to set the subtype field to **Material**.
- 4 Change the first code from 0 to **1** with a description of **P.V.C.** Then in the **Default Values and Domains** pane, enter the default values of **8** for **PipeSize**, **2010** for **YearBuilt**, and **Oleander** for **Description**. Click **Apply** to check for errors. If there are any, correct them before moving on.
- 5 On the next blank line, enter a code of **2** with a description of **HDPE**. Enter the default values of **10** for **PipeSize**, **2010** for **YearBuilt**, and **Oleander** for **Description**. Click **Apply**.
- 6 Continue by entering a code of **3** and a description of **DI** on the next blank line. Enter the default values of **12** for **PipeSize**, **2010** for **YearBuilt**, and **Oleander** for **Description**. Click **Apply**.

- 7** The last two subtypes do not have any additional default or domain parameters. On the next two lines, enter codes **4** and **5** with descriptions of **Conc** and **Clay**, respectively. Click **Apply**, and then click **OK**. This wraps up the database creation process. All the components of the logical model have been entered. On very complex datasets, it's a good idea to run back through the logical model and check the entries against the feature classes to make sure nothing was missed. If you were methodical in the process, everything should be OK.

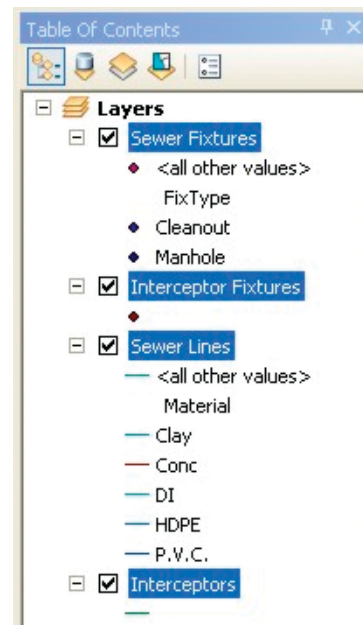


- 8** Close ArcCatalog.

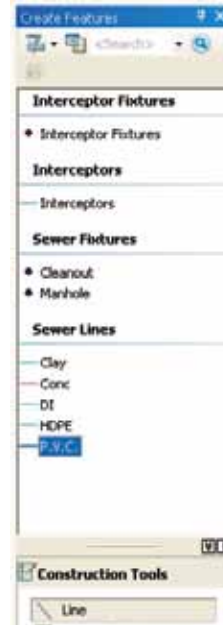
Test the rules

As in the last tutorial, it would be advisable to test the features in ArcMap before proceeding. This quick pilot study will determine if any of the rules are not working and will give you the chance to go back into the data structure and correct them if necessary. Once data is loaded into the feature classes, it would be much riskier to try and fix problems, especially if this involved deleting and re-creating a field.

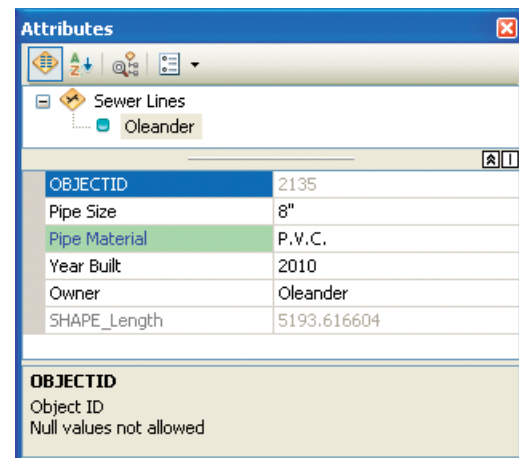
- 1** Start ArcMap and open Tutorial 2-2.mxd. Add the feature dataset Wastewater from the Utility Data geodatabase to the table of contents, and zoom to the bookmark Starting Extent. All the feature classes in the Wastewater feature dataset should be added to the table of contents. Notice that the layers Sewer Fixtures and Sewer Lines have already been classified by their subtype values. ArcMap does this automatically, although if this is not the desired classification, it can be changed manually.



- 2** Go to Editor > Start Editing. The Create Features template will display all the features that can be edited, grouped by feature class and subtype. Select P.V.C. from the Sewer Lines listing. Notice that the list of selections under Sewer Fixtures and Sewer Lines comes straight from the subtypes you defined earlier.



- 3** Use the default construction tool in the map area to draw a sample line, complete it by double-clicking the last point, and open the Attributes dialog box from the Editor menu. Notice in the Attributes dialog box that all the fields have already been filled in! These are the default values that you defined in the layer properties. For the sewer line type of P.V.C., the default size is 8", the year built is 2010, and the description regarding ownership of the line is set to Oleander.



YOUR TURN

Draw each of the types of sewer lines and notice the attribute values assigned to each one. Remember that the defaults apply only to new items drawn. Since concrete and clay pipes are rarely installed as new, there are no defaults set for these. They need to appear in the database, however, to accommodate existing pipes made of these materials. If they were not in this list, they would test as invalid when the database is checked.

2-1

2-2

- 4** When you have finished looking at the various sewer lines, close ArcMap without saving edits or the map document. This was a quick test of the data structure. If any errors were found, you could go back and correct them in ArcCatalog, but be sure to close ArcMap first before making any corrections. If you wish to delete incorrect entries from the Attribute table, start an edit session in ArcMap, highlight the bad entries, and press Delete. Remember that once data is added to these feature classes, it will be more difficult to change the data structure after the fact.

Exercise 2-2

The tutorial showed how to create a geodatabase and all its components from the logical model. This included a feature dataset, feature classes, domains, and subtypes.

In this exercise, you will repeat the process using the logical model for a storm drain system created in exercise 1-2.

- Start ArcCatalog and locate the MyAnswers folder.
- Use the geodatabase worksheet to create the geodatabase.
- Use the tables worksheet to create the feature classes with the correct fields.
- Use the domains worksheet to create and apply the domains.
- Use the subtypes worksheet to create the subtypes.

WHAT TO TURN IN

If you are working in a classroom setting with an instructor, you may be required to submit the materials you created in tutorial 2-2.

Screen capture of ArcCatalog showing all the components of the completed geodatabases

Tutorial 2-2

Exercise 2-2

Tutorial 2–2 review

This exercise built on the concepts of creating unique **attribute domains** and **subtypes** for linear features. The City of Oleander is managing its sewer network in a geodatabase. As you can see, the same process that was used for setting up domains and subtypes for parcels was used for the sewer system. By using the standardized planning process and filling out the forms ahead of time, you were able to simplify creation of the geodatabase for the sewer system.

ArcCatalog provides all the necessary tools to easily create your geodatabase; however, careful thought was necessary to help establish the behaviors and relationships for these features. As you could see, separate feature classes could have been created for the different materials of pipes and fixtures. However, regardless of the material of pipe or fixture, each one has the same basic function and could be managed more efficiently through the creation of **subtypes** based on material. Using subtypes to define categories of features in this manner is common in the industry to distinguish behavior, attributes, and access properties of features. A domain was created for the size of the sewer pipes to help with the standardization of these values in order to enforce data integrity.

As you have worked through the tutorials, you have probably noticed that there has always been a step to test each component that you built into your geodatabase to ensure that it is representing the features as the needs of the project dictate. This QA/QC exercise is a critical step that should be conducted as often as possible throughout the development project. QA/QC of your database development project should not be skipped because of time constraints, or for any other reason. There is nothing worse than completing the construction of a project and realizing that it does not adequately meet the needs of your client because of a logical oversight or because an error was made early in the development process. Typically, these situations can surface in two different ways that you can help control with thorough planning: Either the features, attributes, behaviors, access restrictions, and so forth, built into the model do not adequately support the workflow process they were designed to support; or errors were made in the execution of the plan. It's really pretty simple. When a failure occurs, you either have a bad model or poor execution. (Of course, this is assuming that the workflow process itself is designed adequately to achieve the necessary results. Most likely, you will have no control of the process that you are hired to support).

To help you succeed with your database design project, make sure to do the following: (1) meet with and document your interviews with project sponsors to discern the necessary requirements for the project, (2) have the project sponsor review and approve the project requirements document that you compiled, (3) use the geodatabase design forms to carefully think through an appropriate approach that reflects the approved results from your requirements session, (4) review your design with the project sponsor and get his or her approval of your proposed design, (5) carefully execute your plan of developing your geodatabase, (6) conduct testing to ensure that your geodatabase meets your requirements, and finally (7) implement your geodatabase.

2-1

2-2

Many of the fastest ways to fail in a GIS database development project can be avoided simply by carefully checking your work, having a colleague check your work, and then rechecking your own work one final time. One of the advantages of using the design forms during the planning phase is that you create a documented paper trail of your logic that you can refer to if a failure occurs and you need to troubleshoot your design. The ESRI geodatabase provides many of the tools you will need to help build an efficient, flexible, accurate, and consistent model of reality. In the last two chapters, you have learned how to organize your geographic features in a geodatabase and how to incorporate attribute domains and feature subtypes into the geodatabase. These tools, as well as the many others that you will be learning in upcoming tutorials, will help you become successful, but remember that the most critical component of any GIS is *the capable mind of a thinking operator!*

STUDY QUESTIONS

1. Explain the rationale for creating subtypes for the sewer lines. What other approach could have been used and how? What situations prohibit the use of subtypes? What advantages do subtypes offer during data editing?
2. What are the primary advantages of creating domains for the sewer pipes? How will this step save you time later?
3. Explain the relationship between feature datasets and feature classes. What are the advantages and disadvantages of grouping your feature classes into several feature datasets?
4. List three mistakes that could be made when creating a feature class that would cause you to have to re-create the entire feature class. How can these mistakes be avoided?
5. What is the most critical component of any GIS?

Other real-world examples

Domains and subtypes are a handy way to increase efficiency when populating your geodatabase and are used in a variety of situations. Consider the use of subtypes and domains in the case of infrastructure management at an electric utility. An electric utility manages its customer power distribution network through a set of power generators, wires, transformers, power poles, and meters. Subtypes can be used to categorize different types of power poles. Wooden power poles can support up to three transformers, and metal power poles can support up to five. The power poles essentially perform the same function in supporting the transformers and the wires; however, when data is being entered, a wooden power pole domain is created to limit the transformer load to three transformers, and a metal power pole domain is created to limit the transformer load to five transformers. Simply establishing these domains to restrict data entry values to established rules for each subclass of power pole will prevent inaccurate data from being entered into the geodatabase, and will allow each type of pole to be symbolized using different symbols on a map (see chapter 8, “Developing labels and annotation”).

Another example of the use of domains and subtypes can be found in the water utility industry. Consider valves along the water transmission pipeline, for example. Valves are used to regulate water within the pipelines. There are two primary types of water valves: maintenance valves and control valves. Maintenance valves are used primarily in pipeline maintenance activities and water flow efficiency regulation. Control valves can be used to restrict or prevent water from flowing within the pipe system. Subtypes can be created for each valve type to designate a maintenance or control function. Within each of these valve subtypes, there are different varieties of valves such as blow-off or air valves in the maintenance valve subtype, and butterfly or gate valves in the control valve subtype. Domains can be created for each subtype to ensure that maintenance valve attribute fields are populated only with blow-off or air valve values and that control valves are populated strictly with butterfly or gate valve types. Such a process will not only streamline data entry, but also enhance data entry accuracy.

2-1

2-2

References for further study

There are a number of additional resources to help create a geodatabase. **ArcGIS Desktop Help** lets you use keywords to search for a title or topic. You can access ArcGIS Desktop Help in ArcMap or ArcCatalog by clicking Help on the main menu. Then click the Search tab, type a keyword, and click Ask. Use the keywords provided here to search for additional learning resources on many of the concepts taught in this chapter.

ESRI has developed a number of online courses on a wide variety of pertinent GIS topics. **ESRI Self-Study (Virtual Campus) courses** are an excellent resource for students that will supplement information covered in this course. ESRI Virtual Campus courses can be accessed at <http://training.esri.com/gateway/index.cfm>.

ArcGIS Desktop Help search keywords

geodatabase, attribute domains, feature class, feature datasets, working with geodatabases, building geodatabases, feature class basics, subtypes, data integrity, topology, projections, spatial reference, coded values, creating subtypes

ESRI Self-Study (Virtual Campus) courses

The following ESRI Virtual Campus classes may be helpful:

1. Basics of the Geodatabase Data Model
2. Creating, Editing, and Managing Geodatabases for ArcGIS Desktop
3. Working with Geodatabase Subtypes and Domains
4. Creating and Editing Geodatabase Topology with ArcGIS Desktop

