Lab 5: Designing an ArcGIS geodatabase

50 points

The past few releases of ArcGIS software have included many additions and improvements to the data storage capabilities of the geodatabase file structure. There are new techniques to control interaction with the data, assign a behavior to it, and define relationships among datasets. In the design process, it is important to understand these techniques in order to build the most efficient database possible.

The goal of designing a geodatabase is to model the reality it is intended to represent. Many characteristics, or behaviors, of the data can be included in a geodatabase using various techniques. As the data modeler, it is your job to explore the capabilities of ArcGIS to make the most efficient and flexible database possible. The time spent at the start of a project in designing the geodatabase will make the data easier to use and edit, presenting a better representation of reality.

Objective

- Become familiar with designing an ArcGIS geodatabase data model.
- Practice the process and special considerations for spatial entities and relationships when designing a spatial DB.
- Learn how to document your DB design for feature classes (spatial entities), relationships, subtypes, and domains etc.

Tasks

- Complete both problems in **Part I** of this lab. The spreadsheet for one problem of your choosing needs to be turned in as your submission for Lab 5, **Part II**.
- Problem 1 and Problem 2 will be used in Lab 6, so make sure to save your work.

Part I: Exercise Tutorial: Creating a geodatabase – building a geodatabase data model

Problem 1: A geodatabase for land parcel management system

I. Begin the conceptual design for the geodatabase

The main component of this geodatabase will be polygons representing every piece of **property** in the city. *Each piece of property is assigned certain data by the city, such as subdivision name, block designation, lot designation, street address, and a land-use code.* The geodatabase is the framework in which other components are built. It may contain feature classes, tables, relationship classes, feature datasets, and many other components. You will design the geodatabase and its components using the ***geodatabase design forms*.**

1. Download '*GDB design forms.xls*'. You may work in Excel with ***five geodatabase design forms*** (GDB, Datasets, Domains, Subtypes, and Relationships worksheets). The first two worksheets (GDB and Datasets) will be used for these first few steps, but the other worksheets will be used in the later steps of this tutorial.
2. On the first line of the first worksheet (***GDB***), write the name **Land Records** for the *geodatabase name*.
3. On the same worksheet (the ***GDB worksheet***), add "**PropertyData**" as the *feature dataset name*. (Note: A **feature dataset** is a collection of related feature classes that share a common coordinate system. Feature datasets are used to spatially or thematically integrate related feature classes.)
4. Add a new feature class named **Parcels**. Note its type as **POLY** (polygon) and give it an alias of **Property Ownership**. The alias is one of the first characteristics of the geodatabase that will be assigned. This alias will be shown in the table of contents when the layer is added to *ArcMap*.

Geodatabase design forms

Geodatabase name	**Land Records**	
Feature Dataset	FeatureClass Type	Name	Alias
PropertyData	POLY	**Parcles**	**Property Ownership**

4. This feature class will have **fields** to store data, and these are recorded on the second design worksheet. On the ***Datasets worksheet***, write the name of the new feature class on the first line (**Parcels**). Under field name, enter the first field as **SubName**. Note its type as **String with the length as 50**. Add another field for **Blk** as **String with the data length as 25,** and **LotNo** as **String with the length as 10.**

Datasets worksheet: Descriptions of all dataset

Featureclass or table name	Field name	Field type	Length
Parcels	SubName	String	50
	Blk	String	25
	LotNo	String	10

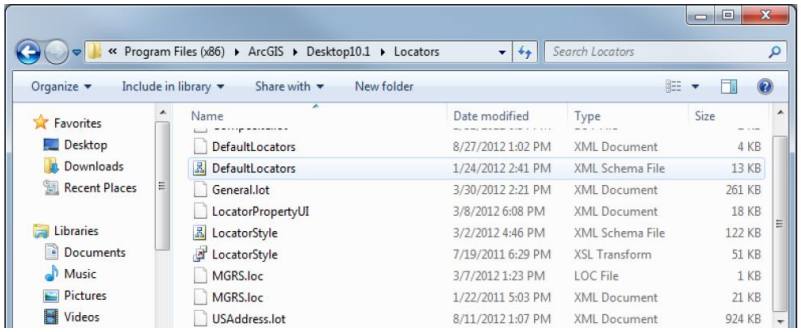
5. Next to the field length, write the description **Subdivision Name** as the alias for the field ***SubName***. Then write the alias **Block Designation** for ***Blk*** and **Lot Number** for *LotNo*. The field alias will be shown in many of the ArcGIS tools, the attribute table, any classification schemes, and many more places when the data is accessed. You can copy the alias of each field as the descriptions for those fields (Note: you can add more detailed descriptions for the fields in your real-world DB design to make them interpretable).

Datasets worksheet: Descriptions of all dataset

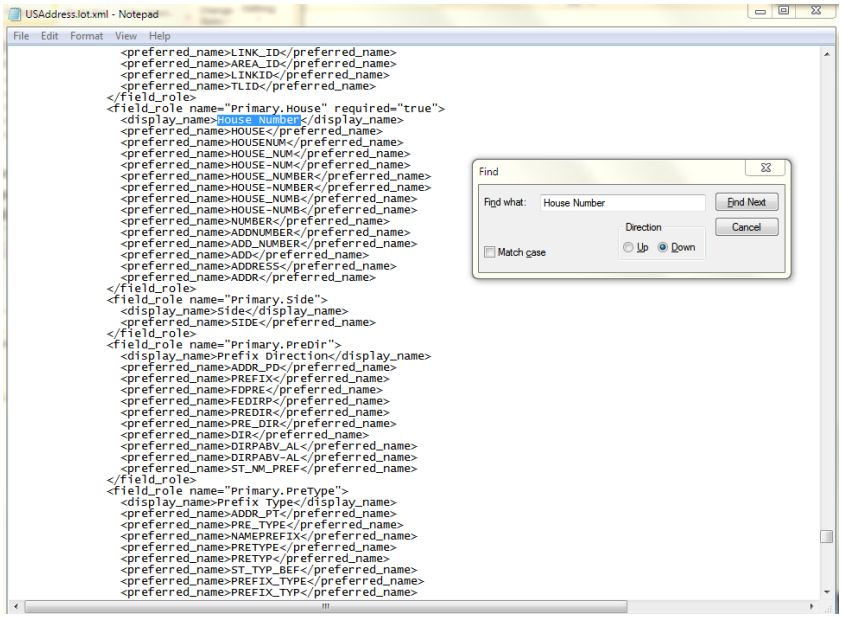
Featureclass or table name	Field name	Field type	Length	AliasName	Description	DomanName	DefaultValue
Parcels	SubName	String	50	Subdivision Name	Subdivision Name		
	Blk	String	25	Block Designation	Block Designation		
	LotNo	String	10	Lot Number	Lot Number		

The next information is address information for each parcel. This could be entered as a single field, but if you ever want to geocode against this dataset, it would be better to have each component of the address in separate fields. *The common fields for **geocoding** are street prefix type, prefix direction, address number, street name, street type, suffix direction, and ZIP code.* One interesting thing is that if the fields are given certain names that ArcMap recognizes, they will be filled in automatically when you make an address locator. An address locator is a special file that ArcGIS builds using your dataset that will allow addresses to be found easily ***when geocoding or using the Find tool***. A list of preferred field names for each field is stored in the address locator style file. You can then open the file in the \ArcGIS\Locator folder and view the list by searching for the phrase "Preferred Field Names".

6. Open a Windows Explorer window. Navigate to the folder containing your ArcGIS installation (e.g., *C:\Program Files\ArcGIS* or *C:\Program Files(x86)\ArcGIS\Desktop10.1*) and open the Locators folder.



Scroll down to the file ***USAddress.lot.xml***, right-click, and select Open with--> Notepad or if given the option, select "Choose the program to open this file," and then select Notepad. Use the Find tool or scroll down to the area labeled ***House Number***.



As you can see, there are ten or more acceptable field names for each of the fields necessary for address geocoding. Look over the list to see what the choices are.

7. On the ***Dataset worksheet***, write the following field names, field types, aliases, and description:

- ***Pre_Type***, String, 3, Prefix Type, Prefix type with possible values as INT, HWY, RMP, CON, LOC
- ***Pre_Dir***, String, 2, Prefix Direction, Prefix direction with possible values as N.S, E, W, NE, NW, SE, SW
- ***House_Num***, String, 5, House Number, House number
- ***Street_Name***, String, 25, Street Name, Street Name
- ***Street_Type***, String, 3, Street Type, Street type with possible values as Ave, ST, etc
- ***Suffix_Dir***, String, 3, Suffix Direction, Suffix Direction
- ***ZIPCODE***, String, 9, ZIP Code, ZIP Code

These fields will add a lot of functionality to the dataset that may be valuable later. You could select all the parcels in a certain subdivision, all the parcels that front a certain street, or use the ***House_Num*** field to put address labels on the map. The last bit of data is the land-use code. This is a two- to nine-letter alphanumeric code denoting how the land is currently being used.

8. On the dataset worksheet, add the field **UseCode** with the field type **String** with **length** as **5** and the alias **Land Use Code**.

9. The identity of the *property owner* is not stored in the parcel's attribute table but is stored in an external table. You will need to add a field (to serve as a foreign key) to your data structure that will allow you to set up a relationship between it and the external table. The procedure is discussed later, but for now you will add a field to accommodate the relationship. On the datasets worksheet, add the field **Georeference** (serves as an unique ID for each piece of property), with the field type and length as **String, 254,** and the alias **Georeference**.

II. Design for data integrity

Imagine what will happen when people start putting data in the table. If they leave the SubName field blank, there would be no way to identify the legal record of a piece of property. What about address number or land-use code? Leaving these blank could create gaps in the data. On the other hand, not every street will have a value for prefix type, so there will be instances where a space can be left blank and still be correct.

One way to build data integrity into your table is to *set the flag for allowing **nulls**, or no value, for a field*. If nulls are not allowed, ArcMap will produce an error for any records entered without all the necessary values being provided. Perhaps, the person entering the data accidentally skips the field during data entry or tries to enter data before all the information is known. Either way, it could cause problems with your data. The solution is to mark in the design table which fields are allowed to have nulls and which must have a value entered.

1. On the Datasets worksheet in column **Nulls (Y/N)**, mark the following fields to allow null values by putting a "Y":

   - Pre_Type
   - Pre_Dir
   - Suffix_Dir

2. Mark the remaining fields as not allowing null values.

Another data integrity component is the ***domain***. A domain allows you to define a list of values for any text field or a range of values for a numeric field. When data is entered, it is matched against the domain to see if it is a valid value. This helps eliminate typos or inventive abbreviations. Imagine ten data entry clerks all coming up with unique abbreviations for the land-use code Vacant. It might be entered as VAC, V, Vcnt, or any number of spellings. If a domain is applied to the field *Use_Code* with all the correct abbreviations, it would be impossible for anyone to enter a value that wasn't in the domain.

3. On the ***Datasets worksheet***, note on the line of Field **UseCode** the **Domain name** that will contain the acceptable values for this field **UseCode**. Call it **ParcelUseCode** for domain name.

Datasets worksheet: Descriptions of all dataset

Featureclass or table name	Field name	Field type	Length	AliasName	Description	DomanName	DefaultValue	IsNullable(Y/N)	Subtype
Parcels	SubName	String	50	Subdivision Name	Subdivision Name			N	
	Blk	String	25	Block Designation	Block Designation			N	
	LotNo	String	10	Lot Number	Lot Number			N	
	Pre_Type	String	3	Prefix Type	Prefix type with possible values as INT, HWY, RMP, CON, LOC			Y	
	Pre_Dir	String	2	Prefix Direction	Prefix direction with possible values as N.S, E, W, NE, NW,			Y	
	House_Num	String	5	House Number	House Number			N	
	Street_Name	String	25	Street Name	Street Name			N	
	Street_Type	String	3	Street Type	Street Type			N	
	Suffix_Dir	String	3	Suffix Direction	Suffix Direction			Y	
	ZIPCODE	String	9	ZIPCODE	ZIPCODE			N	
	UseCode	String	3	Land Use Code	Land Use Code	ParcelUseCode		N	
	Georeference	String	254	Georeferences	Georeferences for land parcel			N	

4. Now turn to the ***Domains worksheet*** and write the domain name **ParcelUseCodes**, a description of **Use Codes for Parcels**, a field type of **String**, and the type of domain as **Coded values**.
5. In the Code column, write **A1** with a description of **Single Family Detached** in the Desc column. Under A1 write **A2** with a description of **Mobile Homes**. Continue down the form entering the rest of the Use Code values from the accompanying list. (Note: to make your life easier, you only need to enter five Use Codes in the excel table, and the geodatabase).

A3	Condominiums
A4	Townhouses
A5	Single Family Limited
B1	Multifamily
B2	Duplex
B3	Triplex
B4	Quadruplex
CITY	Developed City Property
CITYV	Vacant City Property
CITYW	Water Utility Property
CRH	Church
ESMT	Easement
F1	Commercial
F2	Industrial
GOV	Government
POS	Public Open Space
PRK	Park
PROW	Private Right-of-way
ROW	Right-of-way
SCH	School
UTIL	Utility
VAC	Vacant

Domains worksheet

Domain name	Description	Field type	Length	Domain type	Coded values/Range Code (Min)	Desc(Max)
ParcelUseCodes	User codes for Parcels	String	5	Coded Value	A1	Single Family Detached
					A2	Mobile Homes
					A3	Condominiums
					A4	Townhouses
					A5	Single Family Limited
					B1	Multifamily
					B2	Duplex
					B3	Triplex
					B4	Quadruplex
					CITY	Developed City Property
					CITYV	Vacant City Property
					CITYW	Water Utility Property
					CRH	Church
					ESMT	Easement
					F1	Commercial
					F2	Industrial
					GOV	Government
					POS	Public Open Space
					PRK	Park
					PROW	Private Right-of-way
					ROW	Right-of-way
					SCH	School
					UTIL	Utility
					VAC	Vacant

Name: Name of domain
Description: Description of domain
Field type: Must match the field to which this domain will be applied
Domain type: Coded values (text) or range (numbers)
Code (or minimum): Code used to represent value OR low end of numeric range allowed
Desc (or maximum): Description of code OR high end of numeric range allowed

There is a large number of acceptable street type abbreviations, and not only would you not want to have to list them all on the domains form, but you also wouldn't want to type them into a domain. Fortunately, there is a command to take a file listing of street types and read them into a domain. That process will be demonstrated later, but for now, the information about the file name can be recorded on the design worksheet.

6. On the Dataset worksheet, for the field **Street_Type**, write **StTypeAbbrv** as the domain name.
7. Next, go to the ***Domains* worksheet** and write the name **StTypeAbbrv**, a description of **Street Type Abbreviation**, a field type of **String with a length of 3**, and a domain type of **Coded values**. Under Code, write the file name **Data\Suffix.txt** to identify the file holding the domain values. This file was created from a list on the US Postal Service Web site, and it will be added into the domain later.

Domains worksheet

Domain name	Description	Field type	Length	Domain type	Coded values/Range Code (Min)	Desc(Max)
StTypeAbbrv	Street Type Abbreviati	String	3	Coded values	Data\Suffix.txt	

A ***subtype*** is a way to subdivide data within the same feature class, and then apply different data integrity rules to each category. The data can be separated into logical categories and be given data integrity rules for each category, but keep the convenience of being edited and managed in a single feature class. You'll also see later how subtypes can be used to set default values, establish unique attribute domains, set connectivity rules, and establish relationship rules for each subcategory created.

8. On the ***Datasets worksheet***, write in a new field on the bottom called **PlatStatus**, make its field type **SI**, give it an alias of **Plat Status**, and don't allow for null values. Since most new property being added to the dataset will be platted, make its default value 1. Finally, write the name **PlatSubtype** for the 'Subtype'.
9. Next, go to the ***subtypes worksheet***. Write the name of the subtype as **PlatSubtype** and add the three codes described previously:

    - 1 = Platted Property
    - 2 = Unplatted Property
    - 3 = Plat Pending

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

III. Extend the data model

When the Parcels feature class' polygons are symbolized, they can each have a solid fill and a line style for their perimeter. When the maps are made, however, the boundaries of the parcels will be to be symbolized differently. The edge of the parcel that fronts a street will be drawn with a thicker line; the edges representing property lines between properties will need to be a thinner line; and in the event that someone owns two adjacent pieces of property, the line between them should be dashed.

A set of lines can be created that will duplicate the boundaries of each parcel. Then these lines can be symbolized as described. A behavior will need to be set up between the polygons representing property and the lines representing their boundaries. If the shape of any polygon is modified, the lines will need to automatically adjust to coincide. This type of relationship is called ***topology***.

Feature datasets are another way to segregate data inside a geodatabase. If any behavior is to be built for a feature class, such as topologies, network databases, geometric networks, relationships, or terrains, they will have to reside in a feature dataset.

1. On the ***GDB worksheet***, in the **PropertyData** feature dataset, add a new feature class named **LotBoundaries**. Give it a feature type of **L** (for line) and an alias of **Lot Boundaries**.

Geodatabase design forms

Geodatabase name		Land Records	
Feature Dataset	FeatureClass Type	Name	Alias
PropertyData	POLY	Parcles	Property Ownership
	L	LotBoundaries	Lot Boundaries

2. Next, go to the ***Datasets worksheet*** and write the name of the new feature class as **LotBoundaries**. Then write **LineCode** for its Field name. Give it a field type of **String with the length as 10**, add an alias of **Line Code**, and do not allow nulls. Note that there is a domain for this field and name it **ParcelLineCodes**.
3. Finish by filling in the information for the domain. On the ***Domains worksheet***, add the name of the domain **ParcelLineCodes**, a description of **Line Codes for Parcels**, a field type of **String with the length as 5**, and note the domain type as **Coded values**. Then write the three domain values described previously:

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

Domains worksheet

Domain name	Description	Field type	Length	Domain type	Coded values/Range Code (Min)	Desc(Max)
StTypeAbbrv	Street Type Abbreviati	String	3	Coded values	Data\Suffix.txt	
ParcelLineCodes	Line Codes for Parcels	String	5	Coded values	ROW	Right-of-way
					LOT	Lot Line
					SPLIT	Split Lot Line

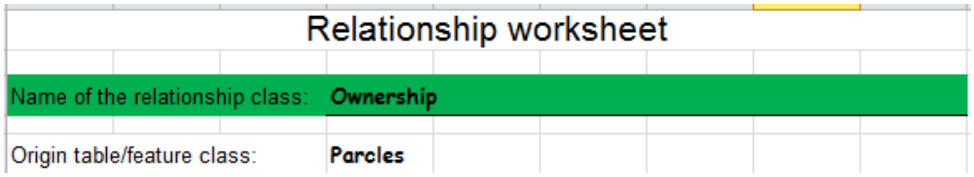
IV. Design a relationship class

Not all the data you will need for this model is in the form of points, lines, and polygons. The design will also need to include ***tabular data*** that is provided by an outside agency. For each parcel, there is ownership and value information that comes from a county appraisal agency. This data would be valuable for analysis if it were associated with the parcel data.

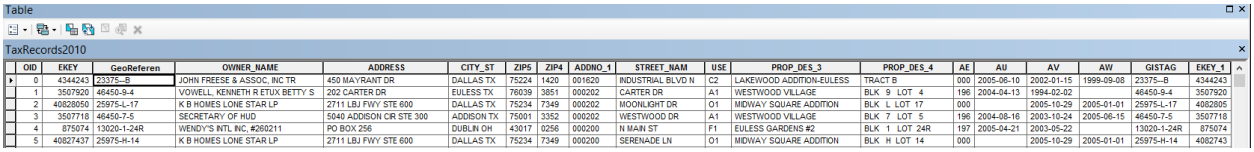
A ***relationship class*** has many of the benefits of a simple join in ArcMap but also provides a mechanism for controlling edits in the related table. If the graphic features were altered in an edit session, rules in the relationship class could also alter the related table and maintain the relationship.

The final consideration is the ***cardinality*** of the relationship. If each parcel has one and only one match in the appraisal table, and vice versa, the cardinality is said to be one-to-one (1:1). If one parcel can have several matches in the appraisal table, such as the case of a single parcel being owned by more than one person, the cardinality is said to be one-to-many (1:M). If the opposite were also true- that is, an owner can also own several pieces of property- the relationship is said to be many-to-many (M:N).

1. On the ***relationships worksheet***, name the relationship class **Ownership**. Record the origin table as **Parcels** and the destination table as **TaxRecords2010**.



   Note: ***TaxRecords2010*** is non-spatial Cadastral table, which does not contain parcel geometry information. However, it includes more details about land parcel properties such as, owner, shown as below.



Table

TaxRecords2010

	OID	EKEY	GeoReferen	OWNER_NAME	ADDRESS	CITY_ST	ZIP5	ZIP4	ADDNO_1	STREET_NAM	USE	PROP_DES_3	PROP_DES_4	AE	AU	AV	AW	GISTAG	EKEY_1
▸	0	4344243	23375--B	JOHN FREESE & ASSOC, INC TR	450 MAYRANT DR	DALLAS TX	75224	1420	001620	INDUSTRIAL BLVD N	C2	LAKEWOOD ADDITION-EULESS	TRACT B	000	2005-06-10	2002-01-15	1999-09-08	23375--B	4344243
	1	3507920	46450-9-4	VOWELL, KENNETH R ETUX BETTY S	202 CARTER DR	EULESS TX	76039	3851	000202	CARTER DR	A1	WESTWOOD VILLAGE	BLK 9 LOT 4	196	2004-04-13	1994-02-02		46450-9-4	3507920
	2	40828050	25975-L-17	K B HOMES LONE STAR LP	2711 LBJ FWY STE 600	DALLAS TX	75234	7349	000202	MOONLIGHT DR	O1	MIDWAY SQUARE ADDITION	BLK L LOT 17	000		2005-10-29	2005-01-01	25975-L-17	4082805
	3	3507718	46450-7-5	SECRETARY OF HUD	5040 ADDISON CIR STE 300	ADDISON TX	75001	3352	000202	WESTWOOD DR	A1	WESTWOOD VILLAGE	BLK 7 LOT 5	196	2004-08-16	2003-10-24	2005-06-15	46450-7-5	3507718
	4	875074	13020-1-24R	WENDY'S INTL INC, #260211	PO BOX 256	DUBLIN OH	43017	0256	000200	N MAIN ST	F1	EULESS GARDENS #2	BLK 1 LOT 24R	197	2005-04-21	2003-05-22		13020-1-24R	875074
	5	40827437	25975-H-14	K B HOMES LONE STAR LP	2711 LBJ FWY STE 600	DALLAS TX	75234	7349	000200	SERENADE LN	O1	MIDWAY SQUARE ADDITION	BLK H LOT 14	000		2005-10-29	2005-01-01	25975-H-14	4082743

   If this is the real-world database design for land parcel management, you will have to add the information for table into the GDB *worksheet (with* the type as Tabular, and Alias as Tax Records 2010*), and include its attributes to the Datasets worksheet.* However, since this is a practice, we will ignore this step.

2. The relationship class can be used to add or delete records, but since the related table will be managed by another source, the relationship type should not allow records to be deleted, making it a simple peer-to-peer relationship. Mark ***Simple*** (peer to peer) as the relationship type and write the labels **Parcel is owned by** for Origin to destination and **Owner has ownership of** for Destination to origin.

3. Mark ***M-N*** for cardinality on your design form, circle ***Yes*** under Attributes, and write the name of the table as **Ownership_Rel**. This table will be used to store what percentage of ownership can be attributed to each owner. Set the Origin foreign key and Destination primary key fields as **Georeference**.

Relationship worksheet

Name of the relationship class:				
Origin table/feature class:		Parcels		
Destination table/feature class:		TaxRecords2010		
Relationship type:	Simple (peer to peer)		Composite	
Labels:				
	Origin to destination:	Parcel is owned by		
	Destination to origin:	Owner has ownership of		
Cardinality:	1-1	1-M	M-N	
Attributes:	No	Yes - Table name:	Ownership_Rel	
			Add to the datasets worksheet	
		Primary key field	Foreign key name	
Origin table/feature class:			Georeference	
Destination table/feature class:		Georeference	Owner	

4. On the ***Datasets worksheet***, add a new table called **Ownership_Rel** with a field called **PercentOwn** and a field type of **Float**. Write the alias as **Percentage Owned**, allow for nulls, and set the default value of **100**.

5. *Save* your work as ***"tutorial_1-1_YourInitials*.xls"**.

Problem 2: A geodatabase for sewer system

In this tutorial, you will design a set of feature classes to store data for a sewer system.

I. Begin the geodatabase design process

1. Open another set of design worksheets. On the ***GDB worksheet***, write the name of the new geodatabase as **Utility Data**. Add a feature dataset named **Wastewater**.

2. On the ***GDB worksheet***, add a new feature class in **Wastewater** feature dataset. Give it the name **SewerLines** with a feature class type of **L** and an alias of **Sewer Lines**.

Geodatabase design forms

Geodatabase name		Utility Data	
Feature Dataset	FeatureClass Type	Name	Alias
Wastewater	L	SewerLines	Sewer Lines

3. On the ***Datasets worksheet***, write the name of the feature class, **SewerLines**. Then write in the Field name **PipeSize** with a Field type of **SI**, **Material** with a Field type of **Text,** and **YearBuilt** with a Field type of **LI**. Add the aliases of **Pipe Size, Pipe Material**, and **Year Built**, respectively.

In Oleander, there are sewer lines that run through the city that belong to other agencies. Some are the pipes of other cities that are headed for the treatment plant, and some belong to the regional utility that handles all the wastewater treatment for local cities. They will all

need to be included in the dataset, and on the maps, to prevent accidentally digging into them. The owner of the line needs to be recorded, so you'll add a field called ***Description*** to store the name of the owner of each pipe.

4. In the ***Field name*** column, add a field called **Description**. Write a field type of **Text** with an alias of **Owner**.

II. Data integrity issues

With this data, it is important that *every pipe have an entry for size and material*. Year of construction, however, may not be known for some of the older, existing pipes. Because of this, do not accept null values for the fields **PipeSize** and **Material**, but allow nulls for **YearBuilt**. Also, the ownership (**Description**) of every pipe must be known, so don't allow for nulls.

1. For each field, write **N** next to the aliases Pipe Size, Pipe Material, and Description in the Nulls column. Write **Y** next to Year Built in the same column.

Datasets worksheet: Descriptions of all datasets

Featureclass or table name	Field name	Field type	Length	AliasName	Description	DomainName	DefaultValue	IsNullable(Y/N)	Subtype
SewerLines	PipeSize	SI		Pipe Size		SewerPipeSize		N	
	Material	SI		Pipe Material				N	SewerLineMaterial
	YearBuilt	LI		Year Built				Y	
	Description	String		Owner				N	

2. On the ***Datasets worksheet***, write the name of the domain for the field ***PipeSize*** as **SewerPipeSize**. Then on the ***Domains worksheet***, write the same name **SewerPipeSize**. Add a description of **Sewer Pipe Size**, set the field type as **SI**, and write the domain type as **Coded values.** Enter the values as shown below and their corresponding descriptions:

- 6 = 6"
- 8 = 8"
- 10 = 10"
- 12 = 12"

Domains worksheet

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

Notice that although the field stores integers, and the code must be an integer, the associated description can be text. This will be useful in labeling the text later, as the inch marks will be visible on the labels that ArcMap generates.

You will use subtypes to segregate data, so that there will be separate domains and defaults for each subset of data. In this scenario, the data will be separated by material. (the new polyvinyl chloride (PVC) pipes going in are 8 inches, the new high-density polyethylene (HDPE) pipes are 10 inches, and the ductile iron (DI) pipes are 12 inches). The only exception would be the interceptors (pipes owned by a regional utility that handles all the wastewater processing). These are typically larger than 12 inches, but the size and material change for each situation. Instead, you will put them in their own feature class. As long as this feature class is in the same feature dataset as **SewerLines**, they can be edited simultaneously and participate in any networks.

3. Add the name of the new linear feature class as **Interceptors** on the ***GDB worksheet*** and give it an alias of **Interceptors**. Be sure to fill in its type as **L**.

4. Write the name of the feature class on the ***Datasets worksheet*** and duplicate all the fields and properties (except Domain name) from the **SewerLines** feature class.

5. On the Datasets worksheet, for the feature class ***SewerLines***, erase the field type for Material and enter **SI**. Also, add the name **SewerLineMaterial** in the subtype column.

Datasets worksheet: Descriptions of all datasets

Featureclass or table name	Field name	Field type	Length	AliasName	Description	DomainName	DefaultValue	IsNullable(Y/N)	Subtype
SewerLines	PipeSize	SI		Pipe Size		SewerPipeSize		N	
	Material	SI		Pipe Material				N	SewerLineMaterial
	YearBuilt	LI		Year Built				Y	
	Description	String		Owner				N	
Interceptors	PipeSize	SI		Pipe Size				N	
	Material	Text		Pipe Material				N	
	YearBuilt	LI		Year Built				Y	
	Description	Text		Owner				N	

6. On the ***subtypes worksheet***, write the subtype name **SewerLineMaterial**. Write the code as **1** and the description as **P.V.C.** In the field column, write the name **PipeSize**, note its domain as **SewerPipeSize**, and its default value of **8**. Note that the default value does not have the inch marks next to it. Remember that its field type is short integer, so the value entered in the database must be a short integer. But the inch mark was stored in the domain description, which can be used for labeling if necessary.

7. Now, write the names of the other fields from the ***SewerLines*** table and their default values. For the field named ***Description***, write a default value of **Oleander**. For the ***YearBuilt*** field, write a default value of **2010**.

8. On the next blank line of the subtypes worksheet, write the code of **2** with a description of **HDPE**. In the field column, write the name **PipeSize**, note its domain as **SewerPipeSize**, and it default value as **10**. As before, write a default value of **Oleander** for the Description field, and **2010** as the default value for the ***YearBuilt*** field.

Subtypes worksheet

				PRESET DEFAULTS	
Subtype name	Code	Description	Field	Domain name	Default value
SewerLineMateria	1	P.V.C.	PipeSize	SewerPipeSize	8
			Description		Oleander
			YearBuilt		2010
	2	HDPE	PipeSize	SewerPipeSize	10
			Description		Oleander
			YearBuilt		2010
	3	DI	PipeSize	SewerPipeSize	12
			Description		Oleander
			YearBuilt		2010

9. Fill in the information for the material type **ductile iron**, or **DI**. It will have a default size of **12** inches with the same domain as the other sizes, as well as a default of Oleander and default year built of **2010**.

10. On the next blank line, write a code **4** with a description of **Conc** and a code **5** with a description of **Clay**. No defaults or domains are required for these, since these material types are no longer installed new.

This completes the design for linear features. Next will be to investigate the point features associated with the sewer lines. At each intersection of the sewer lines, and at various locations along their length, manholes are constructed for maintenance. At the ends of the lines, a smaller access port called a cleanout is added to accommodate the mechanical device that is run down the pipes to clean out clogs.

11. On the GDB feature classes form, write the name of the new point feature class as **SewerFixtures**. Give it a feature class type of **PNT** (for point) and an alias of **Sewer Fixtures**.
12. On the ***Datasets worksheet***, write the name of the new feature class **SewerFixtures** and add the following names with their field types, aliases, and null value allowances:

- **FixType**, SI, Fixture Type, No
- **Flowline**, Float, Flow Line Elevation, Yes
- **RimElev**, Float, Rim Elevation, Yes
- **Depth**, Float, Depth from Surface, Yes
- **YearBuilt**, LI, Year Built, Yes

- **Description**, Text, Owner, No

13. On the Datasets worksheet, write the subtype name **SewerFixType** for the FixType field. Add a default YearBuilt value of **2010** and a default Description value of **Oleander**.

Datasets worksheet: Descriptions of all datasets

Featureclass or table name	Field name	Field type	Length	AliasName	Description	DomainName	DefaultValue	IsNullable(Y/N)	Subtype
SewerLines	PipeSize	SI		Pipe Size		SewerPipeSize		N	
	Material	SI		Pipe Material				N	SewerLineMaterial
	YearBuilt	LI		Year Built				Y	
	Description	String		Owner				N	
Interceptors	PipeSize	SI		Pipe Size				N	
	Material	Text		Pipe Material				N	
	YearBuilt	LI		Year Built				Y	
	Description	Text		Owner				N	
SewerFixtures	FixType	SI		Fixture Type				N	SewerFixType
	Flowline	Float		Flow Line Elevation				Y	
	RimElev	Float		Depth from Surface				Y	
	Depth	Float		Depth from Surface				Y	
	YearBuilt	LI		Year Built			2010	Y	
	Description	Text		Owner			Oleander	N	

14. On the subtypes worksheet, write the name of the subtype as **SewerFixType**. Give it a code of **1** for the description **Manhole** and a code of **2** for **Cleanout**.

Subtypes worksheet

			PRESET DEFAULTS		
Subtype name	Code	Description	Field	Domain name	Default value
SewerLineMateria	1	P.V.C.	PipeSize	SewerPipeSize	8
			Description		Oleander
			YearBuilt		2010
	2	HDPE	PipeSize	SewerPipeSize	10
			Description		Oleander
			YearBuilt		2010
	3	DI	PipeSize	SewerPipeSize	12
			Description		Oleander
			YearBuilt		2010
	4	Conc			
	5	Clay			
SewerFixType	1	Manhole			
	2	Cleanout			

15. Add a new feature class to the GDB feature class worksheet. Name it **InterceptorFix**, with a feature type of **PNT** and an alias of **Interceptor Fixtures**. The interceptors will also have associated fixtures, but they are all manholes.
16. Finally, fill in the ***Datasets worksheet*** for the new feature class **InterceptorFix** with these field names, field types, aliases, and null values:

- **FlowLine**, Float, Flow Line Elevation, Yes
- **RimElev**, Float, Rim Elevation, Yes
- **Depth**, Float, Depth from Surface, Yes

- **YearBuilt**, LI, Year Built, Yes (default value of 2010)

17. Save your work as **"Tutorial_1-2_*YourInitials*.xls".**

Part II: Lab Assignment

1. Choose one of the problems, follow the spatial database design instructions. Submit the excel file including your database design worksheet. (50 pts)