CHAPTER

3

Designing the GIS Database Schema

Database designers use the word *schema* to refer to the diagram and documents that lay out the structure of the database and the relationships that exist between elements of the database. A schema is like a blueprint for a database that tells a knowledgeable builder exactly how to construct it. Naturally, designers spend a lot of time thinking about the schema. This work comes before worrying too much about the exact content of tables and even before design concerns for the spatial data. Rushing into building a database without laying out your schema is like trying to build a house without a set of plans; it might stand up for a while, but it will not be as useful as it could be.

The tools that assist in the construction of these schema are called computer assisted software engineering (CASE) tools. These same tools are used to design the structure of complex computer programs as well as databases, and most programmers know how to use them. Many in the GIS world do not, but it is usually possible to design your database with paper and pencil, and some database designers still work this way. The ability to erase entire tables, delete relationships, add relationships, and so on, is sometimes easier with pencil and paper or on a whiteboard than mastering a new set of tools. One of the problems with GIS is that it appears to force you to develop areas of specialization and skill that you didn't have before. Sometimes it just takes too long to learn the new tools, so feel free to use simpler ones you have mastered instead of new tools that do basically the same thing.

Elements of a Schema

A schema at its simplest consists of an arrangement of tables and the relationships between them. Because organizations differ so widely in the kind of work they do and the types of data they need to do this work, it is impossible to provide a cookbook schema for every application. Software vendors that have many users of a particular sort, however, have constructed template database schema that can be customized. But even the task of customizing an existing schema in a complex organization that is planning to construct a large geographic database that integrates most of its existing data and incorporating new data tables into that schema ought to be done by highly skilled database administrators and designers. From the perspective of the users of the geographic data, who may be a minority of the total set of users, it is important that your geographic feature tables be correctly linked to the other tables you need to do your jobs. Schemas for organizations such as electric or gas utilities can take up many square feet of paper and hundreds of pages of documentation, and it is not our intention to outline that process or run through an example like that here. Rather, our focus is to ensure that you know what a schema is, why its construction is important, and how to provide input and feedback to those who are designing the entire system.

Designers of schema may be strict or loose constructionists. A strict constructionist designer would insist on a larger number of elements in a schema and a correspondingly longer time to develop one. A loose constructionist would prefer a smaller but adaptable framework before beginning to assemble data, assuming that the schema will evolve over time and use. Here we take the second approach for the practical reason that schemas can get very large and involved and thus are difficult to discuss in this format and because we do basically feel that trying to tie down every possibility in a database reaches the point of diminishing returns quickly. So as a compromise Table 3.1 shows what we consider required and optional elements for a GIS database schema.

Data Dictionary

As a relatively simple example, we present a project conducted by a consulting company for an annotated bibliography of study reports and historical documents of the Snohomish River basin in the state of Washington. The requesting state transportation agency wanted a GIS interface on this bibliography and document collection so that by identifying a feature of interest, say, a particular wetland, users could pull up all the documents related to that feature and any recommendations that had been made for that feature with respect to mitigation, restoration, water quality, and so on. The data dictionary is shown in Table 3.2. This is a minimal but adequate example; it names fields with reasonable and brief names, and it provides information about the type of data the field contains, how many columns in the table have been allotted for a field, and a description of the field.

Strict constructionists might wonder why a Char (text) data type was given to the year instead of a numeric data type and what values the Rating field in the Citations table would be allowed to take or, in the Recommendations table Preserve field, what it would mean if a value was something other than 3. These concerns can be met in the metadata. The three tables that contain information clearly represent distinctly different objects or features. In the feature tables you find the geography, or the *where,* of the particular features, wetlands, environmentally sensitive areas, and so on. We are seeing the attribute table; there is an associated table that contains the spatial information that defines the features. There will

Table 3.1 Important and Optional Elements of a GIS Database Schema

Element	*Description*	*Example*
Important elements:		
Data dictionary	A field-by-field description of each field in each table. At a minimum it must include the data type (e.g., numeric, text, date, image), the spaces it requires in the field (if appropriate for the data type), anda description of the data. It should also include explanations of the various values that a record can take for each field. (See Data_Dict Table.)	Table_Data_Dict
Primary and foreign keys	Each table must have a primary key, a field containing an identifier for each record that is unique to that record. Foreign keys are fields in one table that are primary keys in another. Primary and foreign keys are used to link tables.	FEAT_ID in the Feature Attribute tables and table DOCREC_NO in the Recommendations (see Figure 3.1)
Entity-relationship diagram	A diagram using a standard diagramming format that shows exactly how the tables are related to each other – the primary and foreign keys and the type of relationship. It must show whether relationships are one to one, one to many, or many to many and what relationships are mandatory and which optional.	Figure 3.1, 3.3
Required metadata elements	See Metadata section later in chapter.	
Optional elements:		
Work flow diagrams	A diagram of exactly how various tables are incorporated into specific routine tasks through forms, reports, maps, etc.	
Form and report designs	The layout and data sources for normal data input forms and the design and data sources for standard output reports. In a GIS database the standard reports will include map or layout templates.	
Security	Explanation of how access to schema elements is allowed or restricted to users or classes of users.	
Domains and validation rules	As part of a data dictionary, explicitly defined allowed value ranges for data and acceptable unique values for certain fields. For example, the date allowed in the field for the date a permit process began could be restricted to be only the date on which the form was filled out.	
Standard queries	Regular queries that would access tables and be input to standard reports	
Optional metadata elements	See Metadata section later in chapter.	

Table 3.2 Data Dictionary

Feature Attribute Tables (One for Each Data Type, Point, Line, or Polygon). Contains only geographic information about the features. There will be a record for each feature.

Field Name	*Type*	*Size*	*Description*
GIS FIELDS	Char	Varies	Standard fields such as length or area
FEAT_ID	Char	5	Arbitrary and unique feature ID (e.g., PL001, LN001, PT001)
LOCATION	Char	80	Text description of the feature location (e.g., Sunday Creek)

Citations Table. Contains information about each document regardless of how many features or recommendations may be referenced in the document.

Field Name	*Type*	*Size*	*Description*
DOC_NO	Char	4	Arbitrary document number (D001, D002, etc.)
AUTHOR	Char	80	Author(s) written in citation format
YEAR	Char	4	Year of publication
TITLE	Char		Document title in citation (sentence) format
SOURCE	Char	100	Document publisher/source in citation format
RATING	Char	5	Document rating (e.g., b3 r1)
ANNOFILE	Char	12	Pointer to disk (text) file containing full annotation

Information about each recommendation. There could be many recommendations in each document, so DOCREC_NO is a unique number identifying each recommendation and its associated document.

Field Name	*Type*	*Size*	*Description*
DOC_NO	Char	4	Document number (see above)
REC_NO	Char	3	Sequential number for recommendations (R01, R02, etc.)
DOCREC_NO	Char	7	Combination of document and recommendation number
PRESERVE	Char	1	3 if recommendation is for preservation
MITIGATE	Char	1	Check box if recommendation is for mitigation
RESTORE	Char	1	Check box if recommendation is for restoration
FISHHAB	Char	1	Check box if recommendation pertains to fish habitat
WATERQUAL	Char	1	Check box if recommendation pertains to water quality
WETLAND	Char	1	Check box if recommendation pertains to wetlands
FLOODING	Char	1	Check box if recommendation pertains to flood control.
DESCRIPT	Char	200	Summary of the recommendation.

Link Table. Contains all the associations between map features and recommendations.

Field Name	*Type*	*Size*	*Description*
FEAT_ID	Char	5	Arbitrary feature ID (see above)
DOCREC_NO			Combined document and recommendation number (see above)

Source: Konkel 1999. Used with permission.

be several feature tables in this database, probably one for each class of features that is involved (a set of wetland features, perhaps a set of environmentally sensitive areas, etc.). The citation table contains information about the document itself; this table closely resembles a catalogue card.

The Recommendations table comes out of the needs of the project. The clients particularly wanted to track what recommendations had been made for the various geographic features. This meant that someone had to read each document and identify the various recommendations made for which features. So a recommendation is an object separate from features or documents. This is a key issue in schema design; each table should represent a distinct class of objects and only information that relates to that type of object should be present in the table. So a document object and a recommendation object represent different feature classes and belong in different tables.

It is very easy to fall into the practice of mixing feature types, that is, putting recommendation fields in the Feature table. This quickly leads to the question of how many recommendation fields to create. You can be certain that if you design the table with space for six recommendations, you will find a feature that has had seven recommendations made for it. Then you are stuck. Generally, if you need to decide how many fields to leave for a type of information, you have mixed feature classes in a single table. Consider another example of telephone numbers and people. A telephone number is not the same thing as a person. One number can be linked to many people and certainly one person can have many telephone numbers, but you really can't know how many numbers each person will have. So a well-designed database would have a separate table for people and one for telephone numbers with links between those tables. The telephone number table could also contain a field that identified the type of number, home telephone, work phone, mobile phone, and so on. Instead, what you usually see in a personal contact table is a separate field for each possible type of telephone number. But if a person has only one type of number, you are creating a space, but there is no data. This is another clue that you are mixing feature types in a single table.

Tables and Relationships

The second critical part of a database schema, and actually the one you create first, is a diagram that shows the relationships among the various tables in the database, as shown in Figure 3.1. Relationships have a property called cardinality that describes the type of relationship. The possibilities for relationships are one to one, one to many, and many to many. Additionally, relationships may have the property of being required (mandatory) or optional. An example of a required one-to-one relationship in this figure is the relationship between the Recommendations and Citation tables (in that direction). Each recommendation must have a document number (i.e., come from a document), and that DOC_NO links to the Citation table that contains information about the document. This means that a recommendation without a document is not possible in this database. If you want to allow for that, you can, but as this database is designed, all recommendations must exist in a document.

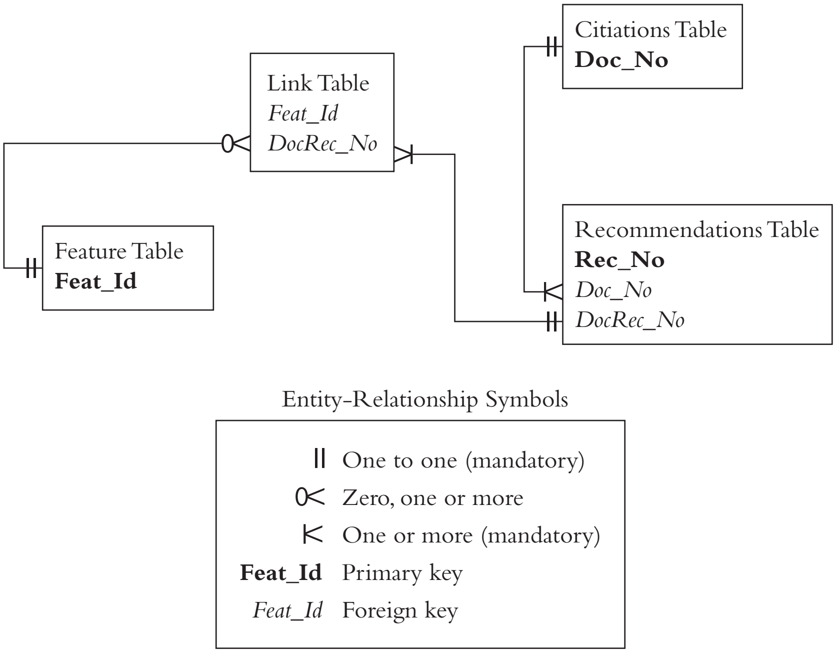

Figure 3.1 Schema diagram.

If a similar recommendation is made in another document, that is considered a different recommendation; otherwise the Recommendation-to-Citation relationship would be one to many. The relationship the other way (the Citation table to the Recommendation table) is a one-to-many mandatory relationship. Each citation must be related to at least one record in the Recommendations table but could be related to many if there were more than one recommendation in the document. Forcing at least one relationship means that all documents must contain at least one recommendation. If that were not the case, the relationship could be optional (i.e., 0, 1, or many and nonmandatory), but then there would be no link between the document (Citation table) and any features (real world geographic entities). It is only through the Recommendations table that you can tie geographic features to documents. The central relationship being modeled is between the geographic features and recommendations that have been made concerning them, and that is an optional many-to-many relationship.

To handle many-to-many relationships the schema needs a composite, or linking, table. Database programs relate tables one to one and one to many directly between the tables, but a many-to-many relationship requires the construction of an intermediate table. By establishing one-to-many relationships between each data table (the Feature and Recommendations tables) and the link or composite table, you create a many-to-many relationship between the two data tables. The one-to-many relationship between the Feature table and Link table allows features to exist in the Feature table that have not been discussed in a document and for which there are no recommendations (i.e., the feature may exist in the feature table and be seen on the map, but its Feat_Id value is not in the Link table). This means that geographic features that might be important for identifying where you are are not linked to any recommendations. Another way to handle this would be to create a dummy document and dummy recommendation that said "No recommendation has been made on this feature," possibly in the DESCRIPT field of the Recommendations table. Then the Feature table-to-Link table relationship would be a one-or-many (mandatory) relationship. You have the ability to design it either way. If a zero relationship is possible, clicking on the feature in the data (map) view will produce nothing. If you have created the dummy recommendation and citation records, the text "No recommendation has been made" would appear. If you plan it one way and change your mind, it is always possible to modify the schema, but it is better to think through questions like that at the beginning of the design process.

The reason that a schema diagram is important and not an optional element in designing a GIS should be clear from the preceding paragraphs. It is possible to document the relationships with words and descriptions, but the graphic picture of how the relationships flow is much clearer once you understand the symbols. With the tables and relationships in this schema it would be possible to click on a feature on the computer screen — a point (well), line (section of stream), or polygon (wetland) — and immediately know at least all of the following:

- Who made this recommendation and when was it made. (Table:Citation/Field:Author and Table:Citation/Field:Year)
- If this feature has had any water quality recommendations made on it and when. (Table:Recommendations/Field:Waterqual and Table:Citation/Field:Year)
- If this feature is recommended for preservation in any document. (Table:Recommendations/Field:Preserve)

Of more interest are the queries that this structure makes possible. For example, you could create a query that would show all features:

- For which a recommendation related to fish habitat was made between 1990 and 1995
- For which recommendations were made in a particular document
- That have a recommendation pertaining to wetlands and are recommended for preservation
- That have conflicting recommendations made for them in different documents

Schema Example

The schema example shown here is relatively simple. More complex databases may have dozens of tables and relationships. The development of a complete data dictionary and basic schema diagram before trying to populate it with data are vital steps. The steps to create these schemas are pretty straightforward. As an exercise we are going to work the process through for a local tax collector who wants a GIS database to collect taxes on four classes of features, land parcels, buildings, vehicles, and equipment. The key report that must be created by this database is a bill that will be mailed to an owner or owners so that taxes can be collected. The geographic features that this database must deal with are the land parcels (polygons) and buildings (polygons). The other feature classes, vehicles and equipment, have no inherent geography in this example and exist only as attribute tables.

Step 1. Identify all the possible classes of objects. We began with five object classes, land parcels, vehicles, equipment, buildings, and owners. But because this database is going to support a billing process, bills are another object class. It is important to separate the object classes so that you can design the appropriate fields for each class. A common design mistake in databases for land value assessment is to include the owner in the table for the land parcel. Owners and parcels are quite different classes of objects. They both have addresses, for example, but often not the same address. If you include the owner information as fields in the land parcel table, you run into the following problems:

- Some land parcels have multiple owners. You can allow for this with additional owner fields in the table, but most of them will be empty, and if you allow for only three owners, it is almost a given that you will find a parcel owned by six people.

- You have to store the owner's name for each parcel that he or she owns. The fundamental relationship between land parcels and owners is a many-to-many relationship. A single parcel may have more than one owner and a single owner may own more than one parcel. Storing owner information multiple times provides more opportunities for mistakes. Each owner will have only one entry in the owner table, and this removes the confusion between William J Smith, William J. Smith, and William James Smith. Perhaps this individual owns several parcels and is on the separate deeds with these slight variations on his name, but they all are the same person. If the name, rather than a parcel identification number, is part of the parcel table, it will be in the parcel table in these three slightly different variants.

- When an owner moves and changes addresses, you will have to make that change for every parcel that person owns. In a correctly designed schema each owner will be a single record in the owner table, and you will make the change once in that table. The owner's address is a property of the owner, not the parcel. If the parcel has an address, it is appropriate to have that in the parcel table, though. Addresses, although they may seem straightforward, can be rather complex things (see chapter 4 "How They Did It—Kansas Geospatial Addressing Standard.

Step 2. Sketch the relationships you will need between tables. At first, don't worry about the type of relationship, one to one, one to many, or many to many; just draw the minimum amount of needed lines. Figure 3.2 shows the six object classes and the relationships between pairs of tables.

Owners need to get bills and bills need the information from the land, vehicle, equipment, and buildings table. The relationship between the owner and the assets is one of ownership; the owner owns the asset. Viewed the other way, the asset is owned by the owner. The relationship from owner to bill is one of must pay and from bill to owner is must receive from. There are some relationships that might initially seem necessary (e.g., relationships between a bill table and the asset tables); however, they are not because they exist through the owner table. An additional relationship between buildings and land is shown in Figure 3.2. You might want to establish a relationship between the equipment and building and/or land table (is found in/on).

Step 3. Detail the key relationships first and then the secondary relationships. In this example you will need to decide whether you are going to send each owner one bill for all assets or a separate bill for each category of asset he or she owns. This is a decision that is independent of the database design; it can support either decision, but the design will differ. Making a decision to send out a single bill and later changing your mind to send a different bill for each category of asset would mean a redesign of the database. In our example, because the purpose is collecting taxes, the key relationship is between owners and bills. We have decided that we want owners who have any assets to get a single bill for all assets, but we also want the ability to include owners who own no assets at this time in the owner table. Perhaps they used to own assets but don't any longer. But if they own any assets, they are to get a single bill for all assets at once.

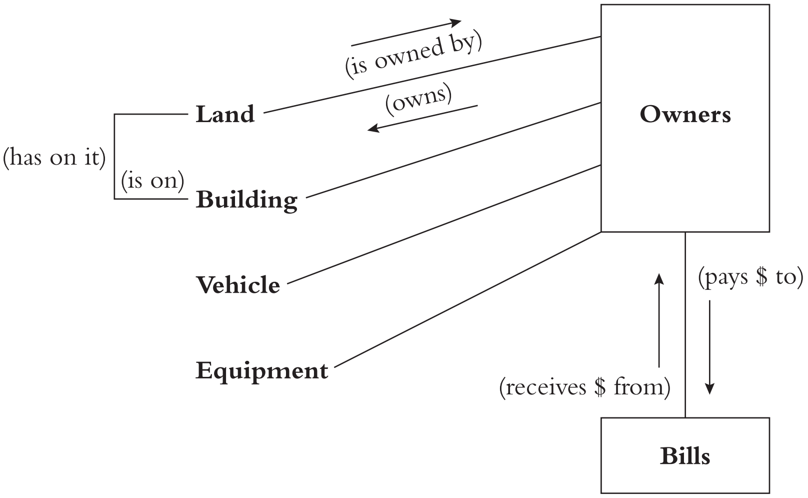

Figure 3.2 Early sketch of schema diagram.

Detailing the relationships forces you to think about primary and foreign keys for the tables. A primary key in a table is a field that contains a unique identifier that is not duplicated for any feature in the table. So, each owner has a unique ID number that might be a social security number, although there are occasionally duplicates of these numbers. Or your system could automatically create an integer Owner_Id value as owners are entered into the table. Most databases will automatically create a primary key for you, usually a sequential integer, but you may wish to create a key that contains information as well. A common primary key that could be created for the land table, for example, would be the combination of the map, block, and lot numbers of the parcel from the original paper maps. Foreign keys are fields that are primary keys in other tables. The relationships between tables are usually created between the primary key of the origin table and that field as a foreign key in destination table. For this reason keys must be formatted in exactly the same way. So if your primary key in the land parcel tables is Map-block-lot, the field must be built the same way in the asset table. If you stored it as Map/block/lot, the two tables would never join.

When joining tables, it is important to remember which is the from, or origin, table and which is the to, or destination. In Figure 3.3 consider the Owners table as the from table and the Bills table as the to table. The relationship from Owners to Bills is a one-to-many relationship because owners are going to be sent many bills through this system. Even though they will get one bill for all assets, this database will support many billing periods. Having no bills is a possibility because we wish to include people who are currently not owners but may become owners. To join these two tables the Own_Id, the primary key in the Owners table, exists as a foreign key in the Bills table. Looking at the relationship the other way, considering the Bills table as the from table and the Owners table as the to table, the relationship is a mandatory one-to-one relationship. Each bill must belong to one and only one owner. If an asset has multiple owners, only one receives the bill. This means that the Owner_Id values in the Bill_Table must be the ID values only for the primary owner. The multiple ownership is maintained in the links between the Asset_Owner and Owner tables, but the Bill_table only lists a single owner, the primary owner who receives the bill. The presumption is that that owner is responsible for seeing that the taxes are paid; if you send the same bill to all owners, you might receive multiple tax payments that would have to be refunded.

The second key relationship is between owners and assets. This relationship is many to many because a single owner may have many assets and an asset may have more than one owner. Relational database design requires a composite, or link, table in this situation, and that table is called Assets_Owner. Each asset is joined to one or more entries in the Assets_Owner table. If an asset has more than one owner, there will be multiple records in the Assets_Owner table to represent that. For these records, the Asset_Id will be the same, but the Owner_Id will be different; there will be one record in the

Assest_Owner_Table for every owner of this particular asset. This is how you get around the issue of how many owners to assign to an asset; it may have as many owners as it needs, but it must have at least one owner. An asset must have an owner, even if it is a dummy value that has an Owner_Id of 99999 that is linked to a dummy owner whose name is Unknown with an address of Unknown, and so on. Asset_ID and Owner_ID are foreign keys in this table; the primary key for Assets_Owners table, Asset_Owner_Id, is not shown but would exist. Tables whose primary key is not a foreign key in another table technically do not need to have a primary key, but it is good design to have one.

Another relationship, Owners table to Asset_Owner table, is a zero/one-to-many relationship. This means an owner may have no assets; these are the owners we wish to keep in the system even though they currently may own no assets. The relationship from the Assets_Owner table to the Owners table is a mandatory one-to-one relationship. No asset/owner combination may have zero owners but may have multiple owners.

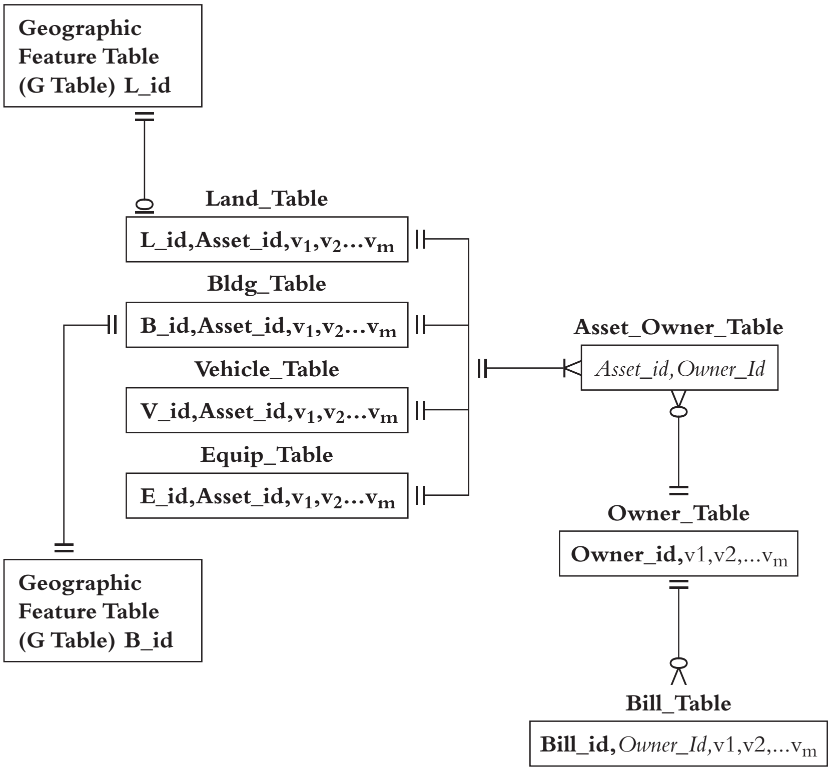
Geographic Feature Table (G Table) L_id
Land_Table
L_id,Asset_id,v_1,v_2...v_m
Bldg_Table
B_id,Asset_id,v_1,v_2...v_m
Vehicle_Table
V_id,Asset_id,v_1,v_2...v_m
Equip_Table
E_id,Asset_id,v_1,v_2...v_m
Geographic Feature Table (G Table) B_id
Asset_Owner_Table
Asset_id, Owner_Id
Owner_Table
Owner_id,v1,v2,...v_m
Bill_Table
Bill_id, Owner_Id,v1,v2,...v_m

Figure 3.3 Detailed schema diagram.

The relationships between bills and the assets themselves (land, vehicle, buildings, and equipment) are a little more complex. Clearly, owners with no assets will receive no bill, and owners with many assets will receive only one bill. So the relationships between the bill class and the asset classes are one to many and nonmandatory, but at least one of the relationships must return a value. Individually, the relationship is one bill to many assets (optional), but collectively it is one bill to at least one asset. This possibility shows that a composite table is clearly needed between owners and assets because owners may own many assets and assets may have more than one owner.

The geographic feature tables exist only for the land parcels and the building outlines. Vehicles and equipment get moved frequently, so their location is not a permanent property and is not modeled in this database. The land parcel geographic table has a zero-to-one relationship with the land parcel data table. A zero relationship means that a tax bill will not be generated for the land asset, which would be the case for properties owned by the various levels of government and some other organizations. Those land parcels will exist in the geographic table, however. The Land_Table-to-Geographic Feature Table relationship is one-to-one and mandatory. All records in the land data table are linked to one, and only one, record in the land geography table. This may not be a realistic situation; a parcel of land that is considered a single feature for tax purposes may consist of two or more separate polygon features. In that case the relationship would be one-to-many and mandatory. Every record in the land data table has a corresponding record in the land geography table. This means we know where all the taxable land is and have it digitally mapped. The relationship between the building geography and the data is simpler: one to one mandatory in both directions. Also notice that there is no link between the two geography tables, although there could be. The Land Geographic Feature Table-to_Building Geographic Feature Table relationship is a zero, one, or many relationship because a land parcel may have no building on it. The Building Geographic Feature Table-to-Land Geographic Feature Table relationship going the other way would be a mandatory one-to-many relationship because building lines can cross property lines, creating a situation where the building is associated with more than one land parcel.

Step 4. Broadly sketch out the key pieces of information you need to know about the members of each class of objects. Fields are always easy to add to tables, and spending a lot of time at the beginning of the process in detailing exactly how and what you are going to record in each field of a table is not necessary. It is important, though, to have a general idea of what kinds of information are important for each table. As a general principle, include only the information that is specific to the type of object. This is why the geographic feature tables shown for the land and building object classes are separate. They contain only the geographic information (location, area); all the other information about the features is maintained in the asset tables.

Step 5. Sketch out the detailed schema diagram. This is best done on a piece of paper where you can move tables around and erase mistakes easily. This example (Figure 3.3) uses a notation called information engineering symbols

(IES). There are other notation systems for database design (e.g., Universal Modelling Language); the database world has not settled on a single format yet. The formal design can also be accomplished using CASE tools, which sometimes come with GIS software packages. Although a complete schema will eventually include every field for each table, this can take up lots of room on a piece of paper. The example shown includes only the primary keys (show in bold) and foreign keys (shown in italic). Of course you will know other pieces of information about each feature in the tables; these are shown as v_1, v_2, and so on. The design concerns for this attribute information are covered in the next section.

A data dictionary that describes the contents of the various tables in the database and a schema are the central design elements at this stage. Tables and their resulting dictionaries can get quite large, and schemas can require large-format printer or plotters to display them. They are essential documents and contain all the information a database professional needs to understand the structure of your database. A schema can take months to complete and be much more involved than the simple example we presented here. As you go through the process, you need to keep several key issues in mind as you design the schema:

- Tables should be specific to the objects they represent and contain information only about that class of object. So a table of utility poles should not contain information about the objects that might be on the poles, (e.g., transformers). A land parcel table should not contain information about the owner; all it needs is a foreign key that can link the land parcel table to the owner table. The land parcel is a geographic feature, and that is the information you store in that table. The owner or owners are people or entities and have different properties, so they belong in different tables.
- Often a relationship is one to one, but in a few instances can be one to many. Even if the situation occurs only once in the database, it must be explicitly modeled in the schema.
- Many-to-many relationships require a composite, or link, table. These tables contain only a primary key for the table (which was missing in the example we presented) and foreign keys to the tables containing the features involved in the many-to-many relationship. Composite tables may never appear in screens that users see, but they must be in the database to maintain this most complex of relationships.

If you are implementing your GIS using a commercial RDBMS, the various types of relationships (one to one, one to many, etc.) may be referred to with different terminology. The software will probably also have tools for creating a database schema. Ultimately the schema is implemented in the database through SQL. Each table is set up with SQL statements, and the relationships are defined with SQL. The CASE tools allow you to visually draft the schema that best represents your view of the world and then automatically generates the SQL statements that will create the database and that become a textual representation of the schema. Those are usually quite technical steps, but the initial work of identifying the tables and the relationships between them can be done in a group setting with nothing more than a large piece of paper and a marker. A database professional should be able to take that and generate the necessary SQL to create the database. The visual representation of the database, as we said earlier, is a blueprint and, like blueprints, is constantly consulted during the actual construction process.

Metadata

Metadata is usually described as data about the data. It is the information you need to document your data set sufficiently so that an outsider can understand all the key issues involved in the construction of the data set, what the various values in the data set mean, what projection you are using, and so on. One analogy is to a catalogue card for a book in a library, although most metadata is considerably more involved than that. The product of the process of creating metadata is a file that describes your data set, or pieces of your database. The mountains of information available on how to produce it, what to include, how to check it against a standard, and publish it is huge, but the actual product is rather small.

The likelihood that this chapter will be out of date by the time it hits print is high because there is a lot of work being done around the world to implement standards and also because the techniques to disseminate metadata are expanding rapidly. Early implementations of geographic metadata were simple text files that could be read by word processors. The National Spatial Data Infrastructure (NSDI) initiative of the U.S. government led to a standard for implementing metadata using hypertext markup language (HTML). Currently, the push is to using extensions to HTML for producing metadata, Simple Graphics Markup Language (SGML) and Extensible Markup Language (XML). It is almost certain that there will be new high-level languages developed out of these that will have additional advantages for implementing metadata.

Since the 1990s there has been a proliferation of metadata standards, but in recent years national organizations have begun to cooperate on a set of international standards that may eventually make it easier to document and share geographic information across the world. The United States Federal Geographic Data Committee (FGDC) standards were an early version of metadata standards, and now there are groups in Europe, Australia and New Zealand (anzlic.org.au/asdi/metaelem.htm), and internationally that are working on geographic metadata. Additionally, the Dublin Core is a project that is attempting to integrate metadata efforts across disciplines and digital data types so that groups working on metadata standards for image data, for example, will have some relationship to standards developed for other types of data.

The FGDC has been at the task the longest of any of the organizations (see *How They Did It—U.S. Federal Geospatial Metadata*), and the standard is mature and well disseminated. It consists of seven information segments, three supporting sections, and major content areas (Table 3.3). Within each content area are often dozens of elements that specify the details of the content. These details are impressive; there are more than 300 elements, including 199 data entry elements. However, the standard has both required and optional elements and only certain items, fewer than 20, in the Identification and Metadata Reference sections are mandatory. Creators must fill in these sections, and some GIS software systems will read through the data set and produce a template for many of the optional items as well. For example, if the map projection for the data set is available in a file formatted for the software to read, some software will read that information and place it in the appropriate metadata location.

HOW THEY DID IT – U.S. FEDERAL CONTENT STANDARDS

How the current U.S. federal metadata standards came about is an example of the slow, but careful, process of governmental coordination and cooperation. In the late 1980s the United States Geological Survey (USGS) set some requirements for how descriptive information about digital geospatial data should be collected and even before the issuing of Office of Policy and Management (OPM) Circular A-16 in October 1990 was circulating and discussing draft metadata documentation standards in the GIS community. The OPM circular established the FGDC with 12 federal agencies on the coordinating committee and the Department of the Interior as the lead agency. This new committee (FGDC) began work in 1990 in the Department of the Interior's USGS. This unit of Interior was the logical location because of the widespread use of their digital data at several different scales by many other federal, state, and local agencies. Executive Order No. 12906 of the U.S. federal government in April 11, 1994, is the central document, and it took 4 years of coordination, negotiation, and work by the FGDC before it could be issued. It required federal agencies and organizations receiving federal funds to document their geographic information using the FGDC's Content Standard for Digital Geospatial Metadata. There were three goals for this order:

- Minimize the costs of creating geographic information. By forcing the creation and distribution of metadata, different agencies of the government are more easily able to search the resources of other agencies before they expend time and money creating a near duplicate of a data set that already exists.
- Encourage cooperative collection activities. The existence of metadata for a set of geographic data actually assists in the creation of the data (i.e., you have to be able to document the data, which forces you to consider aspects of data design you might otherwise avoid).

continued

- Establish a national framework for geographic data. This order also established the National Spatial Data Infrastructure to design a system for creating metadata-based sites on the World Wide Web for users to seek out spatial data in an organized way.

The deadlines established in the order clearly show that most of the work had been already done before the order was issued. The committee held a public forum as early as 1992 and circulated draft standards in 1993. The first standards were published in June 1994, only a few months after the executive order was issued. After that publication a considerable amount of the resources went into educating the community about the standards and establishing partnerships with state, local, tribal, and university data developers to document their data with the new standard and to set up searchable sites under the NSDI for sharing the data. These partnerships were funded with grants from the U.S. Department of the Interior, through the FDGC, and have been spread all around the country. Staff members of the FGDC were very active, going to GIS conferences around the country and giving presentations on the content standard and how its use was going to make geographic data more usable and accessible.

Adoption of the content standards, except for federal agencies who must use them, has been slow, as has the growth of the NSDI; as of May 2002 there were only 242 nodes on the NSDI, which, given the size of the GIS community, is rather small. Ten percent or more of the servers in the NSDI are likely to be down at any time, and establishing an NSDI site is not something a casual user will do. At the time of writing the position of metadata coordinator on the FGDC was vacant. The Bureau of Land Management was an early adopter and diffuser of metadata and tools to create it, but the Web site that deals with the NSDI had not been modified since September 1998.

Table 3.3 Metadata – Federal Geographic Data Committee

Major Content Areas – Federal (U.S.) Geographic Data Committee Standards	
1. Identification information	An abstract of detailed information in the other sections.
2. Data quality information	Assessment of the accuracy of the spatial and attribute data being described.
3. Spatial data organization information	Detailed documentation of the types of spatial features in the data set. The feature types, vector and raster, must correspond to the Spatial Data Transfer Standard, which was developed along with the metadata standards.
4. Spatial reference information	Coordinate system, projection, and geographic extent information, including information about elevations or depths.

Major Content Areas – Federal (U.S.) Geographic Data Committee Standards	
5. Entity and attribute information	Information about the attributes attached to the features; a detailed data dictionary explaining the values in the data fields.
6. Distribution information	How to obtain the data.
7. Metadata reference information	Description of the metadata itself and how it was produced
8. Citation information	How this information should be referenced if others use it.
9. Time period information	Period of time over which the information was prepared and whether it is updated or not.
10. Contact information	How to reach the custodians of the data.

Required Elements	
1. Identification Information	*Originator*: name of an organization or individual that developed the data set. *Publication_Date*: the date when the data set is published or otherwise made available for release. *Abstract*: narrative summary of the data set. *Purpose*: summary of the intentions for which the data set was developed. *Calendar_Date*: the year (and optionally the month or month and day) for which the data set corresponds to the ground. *Currentness_Reference:* the basis on which the time period of content information is determined. *Status:* the state of the data set. *Maintenance_and_Update_Frequency:* the frequency with which changes and additions are made to the data set after the initial data set is completed. *Theme_Keyword_Thesaurus:* reference to a formally registered thesaurus or similarly authoritative source of theme keywords. *Theme_Keyword*: common-use word or phrase used to describe the subject of the data set.

continued

Table 3.3 (*Continued*)

Required Elements	
	Access_Constraints: restrictions and legal prerequisites for accessing the data set. *Use_Constraints:* restrictions and legal constraints for using the data set after access is granted.
10. Metadata reference information	*Contact_Organization*: organization responsible for the metadata information. *Contact_Address:* four required elements of the mailing or physical address of the contact organization. *Contact_Voice_Telephone:* telephone number of the contact organization.

Minimal required metadata documentation is not the time-consuming task that many data creators think it is. Although these metadata efforts are all tied in somehow with development of the International Organization for Standardization (ISO) standards, it is not as clear how well they are tied to the activities of the Dublin Core. This is partly due to the different backgrounds of the participants; the GIS metadata standards have been developed largely by bureaucrats of national governments working either alone or together, but these professionals have principally come from the geographic data community, that is, they are users and producers of geographic data. The impetus behind the overarching efforts of the Dublin Core comes from the international librarian community, and it is being staffed and organized by professionals whose concerns are the organization, storage, and retrieval of information.

The professionals behind the Dublin Core have expressed concern that the many groups, not just producers and consumers of geographic data, working on metadata standards are going to get so widely separated from each other that they will have little in common. With those concerns, they have developed a 15-element metadata standard that is designed to accommodate many different forms of digital data, not just geographic data (Table 3.4). Although the FGDC standard has a hierarchical structure with many sub-elements and sub-sub-elements under each major section and has some strong restrictions on how data are presented, the Dublin Core is a much simpler standard. There is no complex hierarchy of elements with long names but only these 15 identifiers with suggestions to use standard lists of elements such as standard lists of data types. Examples are Multipurpose Internet Mail Extensions (MIMEs), describing the data type; Uniform Resource Locators (URLs), documenting the Internet location; and International Standard Book Number (ISBN) as a unique identifier. The Dublin Core is a much easier standard to implement and has the added advantage of being usable for all sorts of other data (e.g., documents, nongeographic databases, and

Table 3.4 Fifteen Elements of the Dublin Core (7/2/1999)

Title	*Name of the Resource*
Creator	Entity (person, organization or service) creating the resource.
Subject	Subject and keywords; they suggest using formal keyword systems.
Description	Abstract, table of contents, free-text description of the resource.
Publisher	Person, organization, or service responsible for publication,
Contributor	Entity (person, organization, or service) contributing to the resource.
Date	Typically, the date of creation or availability of the resource.
Type	The nature or genre of the resource; suggested use of the Dublin Core Type Vocabulary Collection, Dataset, Event, Image, Interactive Resource, Service, Software, Sound,Text. Geographic data would be a data set.
Format	Digital encoding of the resource. The suggested list (MIME) does not yet include any geographic data formats.
Identifier	A unique identifier such as a URL or ISBN.
Source	Reference source for the resource (i.e., where it came from).
Language	Language of the resource content.
Relation	Relation to another resource.
Coverage	Suggest use of named places and time periods rather than sets of coordinates or dates.
Rights	Information about who holds the rights to the resource.

Source: dublincore.org/documents/1999/07/02/dces/ This is a DCMI Recommendation, copyright © 2002 (Dublin Core Metadata Initiative).

images). The committees behind the Dublin Core recognize that geographic data are, after all, only digital data and can be documented using a simpler standard. GIS practitioners will recognize, however, that it fails to include necessary documentation for geographic data such as map projection information and other database documentation normally found in data dictionaries. Map projection information is easy to include in the coverage section, though, and if the nongeographic or attribute data are organized in a relational database management system, it is possible to include the data dictionary in tables in the database itself. The Dublin Core itself is probably not adequate for documenting geographic data, but with a few additions it provides a simpler, more generalized format for creating metadata.

In addition to these national and international standards in varying stages of development, many organizations (mostly governmental) have created their own formats for metadata, and some GIS software vendors have also produced formats for documenting data. In fact, it is the proliferation of these varying standards that led to the national and international attempts to standardize the standards. As the more widespread standards emerged, people began to develop tools to help users implement the standards.

In the mid- to late 1990s there was proliferation of these tools. In the United States, FGDC was pushing users and vendors to document their data, and individuals in federal agencies developed and distributed tools for users to create and validate FGDC metadata. These were sometimes software-specific but often were stand-alone programs to assist in the creation of metadata. Most of these tools were labors of love on the part of the creators and have not been maintained or updated, although there are many who are still using such tools as the metadata Arc Macro Language (AML) script written by Sol Katz in the Bureau of Land Management.

Now metadata creation tools are more likely to be built into the software used to create the spatial data. There is also a move toward the direct incorporation of metadata into the database. Previously you had to locate the metadata somewhere else and place the file somewhere in close proximity to the actual data. The reality is, though, that many users are confused about metadata, resistant to spending the time to develop it, and not certain of its utility.

The adoption of standards for metadata has been slow, and many practitioners still do not document their data or do it very poorly. It seems as though the rate of development of the standards is far exceeding the adoption of any standard. Users or organizations unwilling to document their data to a standard can use this rapidity of change as a reasonable excuse. This is unfortunate because metadata answers so many questions about data you have obtained from others and will answer their questions if you provide it to them. The problem is that metadata, prepared to almost any standard, is difficult to create, and it takes specialists to do it correctly. It has been likened to cataloguing in libraries; it requires a specialized set of tasks and not every librarian is very good at it. But organizations with insufficient staff to set aside all or part of one professional to become the metadata expert in the organization face an uphill struggle. And consulting firms who come into an organization, develop some data, and leave will have no incentive at all to spend the time to create metadata unless the requirement and resources have been explicitly included in the contract, and they often are not. The documentation of all geographic data sets to a metadata standard is a generic goal of the world of GIS practitioners, but practically it lags behind and is the last thing done.

A complete database schema will contain much more information than merely a data dictionary and table/relationships diagrams. The elements listed in Table 3.1, such as queries, reports and forms, optional metadata, and so on, are all part of a schema, but they are generated from the data dictionary, the tables, and relationships you have defined. Those items are at the core of your schema, and a well-designed database can support a very wide range of reports, input forms, queries, and workflows. It is the design of the tables and the relationships between them that is at the core of the schema, and that is why we have focused on them here.

GIS implementations almost always involve existing databases and their schemas. The decision of whether to try to bundle all the different databases together, which is something GIS is particularly well suited to, or to redesign the entire system into a single database is important. GIS implementation is always a good time to look at the structure of all your databases, and sometimes the right decision is to redesign the system completely and move the existing data into the new schema. But whether you choose to take that approach or take the approach that links the databases together, you will need to understand the schemas of the existing database. At an early point in the process someone will always say, "We don't have to reinvent the wheel here," that is, we have this perfectly acceptable database that currently meets our needs; let's just tie in a GIS. However, sometimes the wheel you have is not particularly round and runs somewhat awkwardly. In those situations it is a good idea to sit in a room for a while with a large blank piece of paper and sketch out a database that might really work for you. Some wheels are better than others, but you have to design them carefully, and that starts with understanding your existing schema or creating a new and better one.

Additional Reading

Chandler, A., D. Foley, and A. M. Hafez. 1999. Federal Geographic Data Committee (FGDC) Metadata into MARC21 and Dublin Core: Towards an Alternative to the FGDC Clearinghouse. Lafayette, LA: University of Louisiana. eeirc.nwrc.gov/pubs/crosswalk/fgdc-marc-dc.htm.

Federal Geographic Data Committee. 1998. Content Standard for Digital Geospatial Metadata. Washington, D.C.: United States Government Printing Office.

Fraser, B., and M. Gluck. 1999. "Special Section — Usability of Geospatial Metadata or Space-Time Matters." *Bulletin of the American Society for Information Science* 25 (6):24–32.

Harrington, J. L. 1998. *Relational Database Design Clearly Explained*. AP Professional: San Diego, CA.

Konkel, G. 1999. *Final Completion Report: Snohomish Basin Literature Review and GIS Data Acquisition Project*. Washington Department of Transportation Environmental Affairs Office: Olympia, WA. wsdot.wa.gov/eesc/environmental/programs/watershed/snobas/other_links/final_report.cfm.

Smits, J. 1999. "Digital Cartographic Materials." *Cataloguing and Classification Quarterly* 27(3): 321–343.

Internet Resources

Metadata Publications, FGDC:
fgdc.gov/publications/documents/metadata/metadata.html

Department of Interior, Bureau of Land Management's Metadata and WWW Mapping Homepage:
blm.gov/gis/nsdi.html

International Organization for Standardization (ISO) TC/211 – Geographic Information/Geomatics:

isotc211.org/

GIS and Metadata: Frequently Asked Questions:

standardsinaction.org/gismetadata/FAQMetadata.htm