  - Noted structural problems
  - Number of taps by type (fitting or break-in)

Stormwater System Applications Data

- Attribute data required for hydraulic modeling (e.g., inlet capacities)
- Watersheds
- Drainage areas
- Landfills
- Parks
- Recreation areas
- Open lands
- National Pollution and Discharge Elimination System compliance data
  - Roadways
    - Sweeping schedule
    - Debris/litter removal program designations and responsibilities
    - Drainage structures
  - Construction sites
    - Standard Industrial Classification code
    - Owner/occupant information
    - Inspection schedule
    - Inspection dates
    - Comments
  - Field data
  - Dry weather testing data
  - Wet weather testing data
  - Land use
- Identify potential sources of stormwater pollution and drainage defects
  - TV inspections
    - Attributes same as for wastewater system listed above
  - Manhole inspections
  - Smoke testing
  - Dye testing

Stormwater Management Applications Data

- Attribute data required for hydraulic modeling (e.g., runoff curve numbers and stream cross sections)

- Land use
- Stream centerline
- Cross-section location
- Field surveys
- Benchmarks
- Watershed boundaries
- Subwatershed / subbasin boundaries
- Photograph locations
- Outfall locations
- Hydraulic obstructions (culverts and bridges, etc.) and their capacities
- Floodplain boundaries
    - Existing
    - Future
- Improvement locations

DATABASE DESIGN STANDARDS

Certain design standards should be followed in creating a GIS database. For example, the design should adhere to computer industry standards. The standards set forth guidelines on system interoperability and integration, which are critical for the success of a GIS application project. There are four important standards for modern GIS software (ESRI, 2000):

1. Microsoft Windows for interface
2. Structured Query Language (SQL) for data access
3. Component Object Model (COM) for tools
4. Transmission control protocol/Internet protocol (TCP/IP) and hyper text transfer protocol (HTTP) for network data transfer

Interoperability is not a reality today because different standards are advocated by various groups. The major players are Microsoft, with its ActiveX standards, and the open standards bodies, which are promoting the use of Common Object Request Broker Architecture (CORBA) standards. The software industry has recognized the need to provide seamless interoperability between products developed by different vendors. Mechanisms are required to allow components developed by different vendors to communicate, wherever those components reside, be it on the same device or distributed across a local or wide area network.

There are significant benefits in this "component-based" approach, but it requires the adoption of a common standard for the interoperability of component objects manufactured by different vendors. The computer industry standards used in GIS software procurement and integration should also be used for the database design. Additional database design standards and requirements are described in the next section.

GIS Data Models

For data storage and manipulation, a database management system (DBMS) uses a data model, such as a hierarchical, network, or relational data model. In the late 1990s, a new object-oriented (OO) data model was introduced. This model, which can store spatial data inside a relational DBMS (RDBMS), is described at the end of this chapter.

The relational model is the most widely used data model. An RDBMS is a software program that is used to create, maintain, modify, and manipulate a relational database. An RDBMS is also used to create the applications that will enable users to interact with the data stored in the database (Hernandez, 1997). It allows for easy data entry and manipulation, provides fast query and display, and maintains data integrity and security (EPA, 2000). Relational database systems have become the commercial de facto standard because of ease of use and implementation, ability to be modified, and flexibility.

Some applications use information from several different databases or tables. For example, a smoke testing application might require information from manhole and customer account databases. It is inefficient and cumbersome to enter the data from different tables in one table. Relational database tables allow information to be accessed from different tables without joining them together physically.

A relational database stores information in records (rows) and fields (columns). An RDBMS conducts searches by using data in specified fields of one table to find additional data in another table. To accomplish this, there should be at least one "key" or "common" field in each table that uniquely identifies the records. The common field can be used to link the GIS database tables to virtually any external database table. This linkage capability allows the GIS to make effective use of existing databases without requiring new data entries in the GIS database. Once the GIS and external database tables have been linked, the external data can be queried or mapped from within the GIS. Figure 6-1 shows how a GIS table for valves can be linked to an external table containing valve maintenance history. In this example, the "Valve ID" field is the key field common to both the GIS and the external database tables. Figure 6-2 shows the linkage of a manhole theme table in ArcView with an external manhole inspection table.

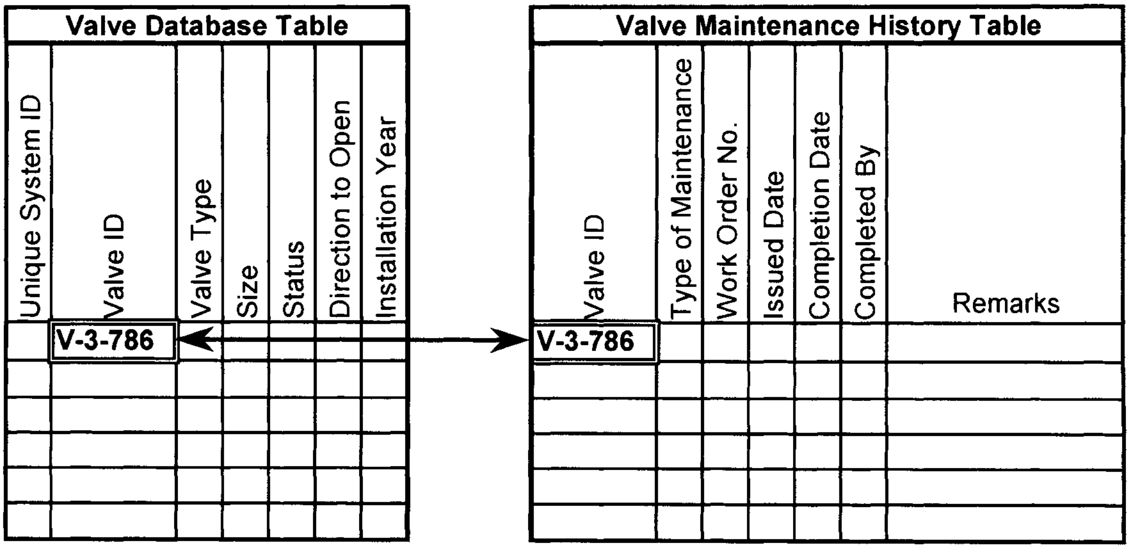


Figure 6-1. Linking GIS and External Tables in Relational Databases

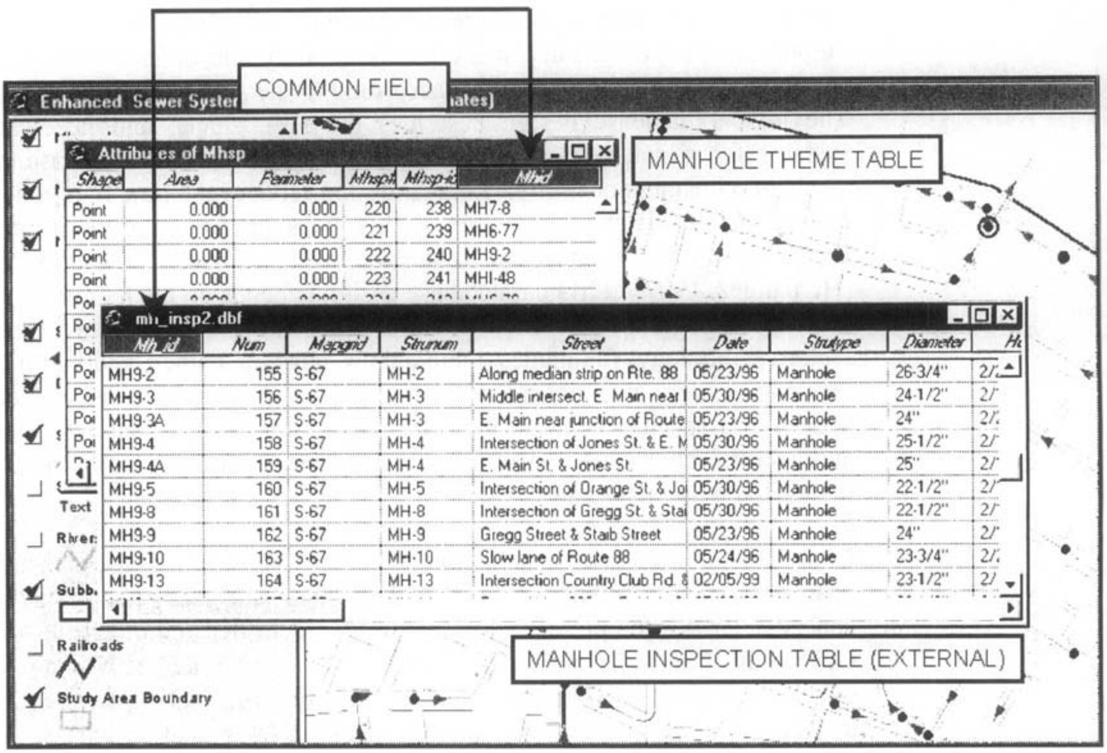


Figure 6-2. Linking Manhole Theme Table and External Manhole Inspection Table in ArcView

The linkage shown in Figure 6-2 is accomplished using the fields "Mhid" of the manhole theme table and "Mh_id" of the manhole inspection table. It shows that when manhole "MH9-2" is selected in the theme table, the corresponding records in the manhole inspection table are also selected.

The traditional file system (hybrid) database design approach saves the results of spatial queries, such as an overlay operation, in a new file. This approach requires a GIS manager to keep all derivative files (such as a GIS overlay) synchronized with their source data. Any change in the source coverage would also require updating the overlay map—a laborious process! The RDBMS approach saves the query rather than the results of the query. Thus, an update of source data would automatically update the query view and there would be no need to change the overlay map (Lowe, 2000a).

DATABASE DESIGN STEPS

Database design involves three steps: (1) conceptual design, (2) logical design, and (3) physical design. Conceptual design does not depend on hardware or software. It defines graphic features, attributes, labels, data format, reports, feature placement rules and guidelines, and database output. Logical design depends on software only and defines the logical structure of the database elements according to the software requirements. The logical database design describes the size, shape, and necessary systems for a database; it addresses the informational and operational needs of a business or utility. Logical design reduces the time required to implement the database structure. This frees up time for developing applications that will be used to interact with the data in the database (Hernandez, 1997). Physical database design depends on the hardware. It specifies system-specific file structure, database structure, data layers, feature symbology, file names, table names, attribute names, and key IDs. Physical design also provides specifications on memory, disk space, access, and speed. You can think of the logical database design as the architectural blueprints and the physical database implementation as the completed home (Hernandez, 1997).

Conventional database design methods involve three phases: needs analysis (or assessment), data modeling, and normalization. Needs analysis clarifies the project's specific needs; identifies and quantifies the GIS needs of an organization and its stakeholders; and defines how a GIS will benefit an organization by relating specific organizational resources and needs to specific GIS capabilities (Wells, 1991). Needs analysis is analogous to strategic planning. Careful needs assessment is critical to successful GIS implementation. Before a systemwide implementation, the initial database design and data conversion methodology should be evaluated for a small pilot area and fine-tuned if necessary. Additional needs analysis and pilot project testing is provided in Chapter 9 (Wastewater System Applications).

Depending on the design goals, basic database design may be divided into three categories: standalone, giant, and joint (most complex). Today, the trend is to use standard database management systems, such as Oracle, Informix, DB2, or SQL Server, to store both attribute and geometrical data. The dependence between the basic data model and applications can be reduced by using simple storage structures in a standard DBMS (Bernhardsen, 1999). A detailed discussion of database design phases and categories is outside the scope of this book. Please read one of the database books (e.g., Hernandez, 1997) listed in Chapter 12 (GIS Resources) for more information.

Each feature of a database must have a unique identifier or "structure ID." Each feature should also have a "source ID" attribute in the database. This attribute represents an identification number for the source material from which the feature was digitized. The source IDs should be explained in a separate table called a "source table" that accompanies the database design report. This table should have information about the source title, number, creator, date, system (water, wastewater, or stormwater), and comments. The columns and sample entries for a typical source table are shown in Table 6-5.

Table 6-5. Typical Source Table for a Database Design Report

Column	*Sample Entry*
Source ID	1001
Source title	River Park Water Service Extension
Creator	ABC Engineers
Date created	January 15, 1959
Date modified	October 21, 1960
Drawing No.	2252-24-59-13
Sheet No.	13
System	Water
Comments	River Park is now called the Reagan Park

Once you have created tables, set up table relationships, and established the appropriate levels of data integrity, your database is complete. Now you are ready to develop applications that will allow you to interact easily with the data stored in your database, and you can be confident that these applications will provide you with timely and—most important—accurate information.

The database design should also support the intended applications, such as work order management. The assets and applications database design should be fully compliant with the data requirements of the facilities management software, such as Cityworks® (more information in Chapter 3). Each feature type should contain certain mandatory attribute fields that facilitate some function within the application software. In addition to these, user-defined attribute fields should be allowed, which can be added to or subtracted from the design. For example, database design for a layer of combined sewer overflow (CSO) events is presented in Table 6-6.

Table 6-6. Database Design for Combined Sewer Overflow (CSO) Events

This layer describes CSO events and supports links to the CSO database.		
Layer Name: CSOEVENT.PAT		
Layer Type: POINT		
Attribute Name	*Definition*	*Description*
Arc#	10 10 I	System-generated ID
Csoevent#	10 10 I	System-generated ID
Csoevent_id	10 10 I	User-specified ID
Structure code	3 3 I	Code representing structure type
Source ID	9 12 F	Source drawing
X_coord	8,20,F,5	State plane *x* coordinate
Y_coord	8,20,F,5	State plane *y* coordinate
Address	32 32 C	Address nearest event
City	32 32 C	City where event occurred
Zip	10 10 C	Zip code where event occurred
Parcel_ID	48 48 I	Tax parcel ID number of event location
County	32 32 C	County where event occurred
Type	32 32 C	Type of event
CSO_Num	16 16 C	CSO event number
Comments	64 64 C	Data converter comments

Typical structure codes referenced in Table 6-6 are shown in Table 6-7. The second column (definition) of the above table consists of three elements:

- Item width: the number of bytes to store the item
- Output width: the number of columns to display the item value
- Item type: the type of data stored in the item

  B = Binary
  C = Character
  F,n = Floating with n decimal places
  I = Integer

Table 6-7. Typical Sewer System Structure Codes

Structure Code	*Structure Description*
101	Manhole
102	Buried manhole
103	Lamphole
104	Cleanout
105	Force main valve
106	Air release valve
107	Pump
108	Diversion chamber
109	Manhole overflow
110	Combined sewer overflow
111	Sanitary sewer overflow
112	Pump station overflow
113	Grease trap
114	Grinder pump
115	Catch basin
116	Head wall
117	End wall
118	Meter pit

Data Dictionary

The first step in designing a database is the development of a data dictionary, which is defined as a catalog that explains the data. It can be a paper or computer document. A data dictionary delineates the specific categories of descriptive information required for each map feature. Data that do not come with a data dictionary may not be usable (ESRI, 1997). Each unique category of descriptive information is called an attribute.

The database design provides a detailed definition of the structure and content of the database. The database design is presented with the help of a data dictionary, which documents the logical and physical structure of the layers of the GIS. It includes descriptions of individual layers, specifications of all tables associated with the layers, physical definitions of the items contained within those tables, listings of the valid codes or codes associated with those items, and diagrams graphically showing the logical relationships between the items and tables.

Table 6-8 shows a portion of a data dictionary for a sewer system documenting pipe material codes. Table 6-9 shows a sample database for a water system (ESRI, 1996a).

Table 6-8. Sample Data Dictionary for a Sewer System

Data Layer	*Sanitary Sewer Pipes*
Name	SANSEWER
Source	Public Works Department
Date	1985
Projection	State Plane
Item	PIPEMAT
Code = ACP	Asbestos cement pipe
Code = CIP	Cast iron pipe
Code = CMP	Corrugated metal pipe
Code = DIP	Ductile iron pipe
Code = PVC	Polyvinyl chloride
Code = RCP	Reinforced concrete pipe
Code = TCP	Terra cotta pipe
Code = VCP	Vitrified clay pipe

Table 6-9. Sample Water System Database

Department	*Layer*	*Abbreviation*	*Theme*	*Features*
Public Service (PS)	WATERPOT	WP	Potable water system	Lines Nodes Points
PS	WATERFACI	WF	Water facilities	Polygons
PS	WATERSERV	WS	Water service areas	Polygons
PS	WATERSV	WV	Water services	Lines Nodes
PS	WATERLEAK	WL	Water system leaks	Points

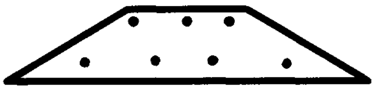
The data dictionary provides the physical description of each layer using three types of documents (ESRI, 1996a):

1. A layer fact sheet that provides basic information about the layer. Figure 6-3 provides a sample fact sheet for the WATERLEAK layer of the sample database shown in Table 6-9.

2. A data diagram that illustrates the logical relationships among the layer's feature attribute tables (FATs) and related tables. Figure 6-4 provides a sample data diagram for the WATERLEAK layer.

3. A data dictionary template(s) that define(s) the FATs and related tables associated with the layer. Table 6-10 gives a sample FAT template for the WATERLEAK layer.

FACTSHEET
WATERLEAK



Category: Water
Theme: Water Distribution System Leaks
Library: FACET

Description	The WATERLEAK layer shows leak locations in the City's water system. Job information for water leaks is maintained in the general water job file (WPJOB.DAT). The link is affected through the job number (WL_JOB_NO).
Feature Class	Points
Entities	Leak points
Associated Data Sets	None
Precision	Double
Responsible Agency	Public Service

Figure 6-3. Data Dictionary Fact Sheet for Water Leak Layer

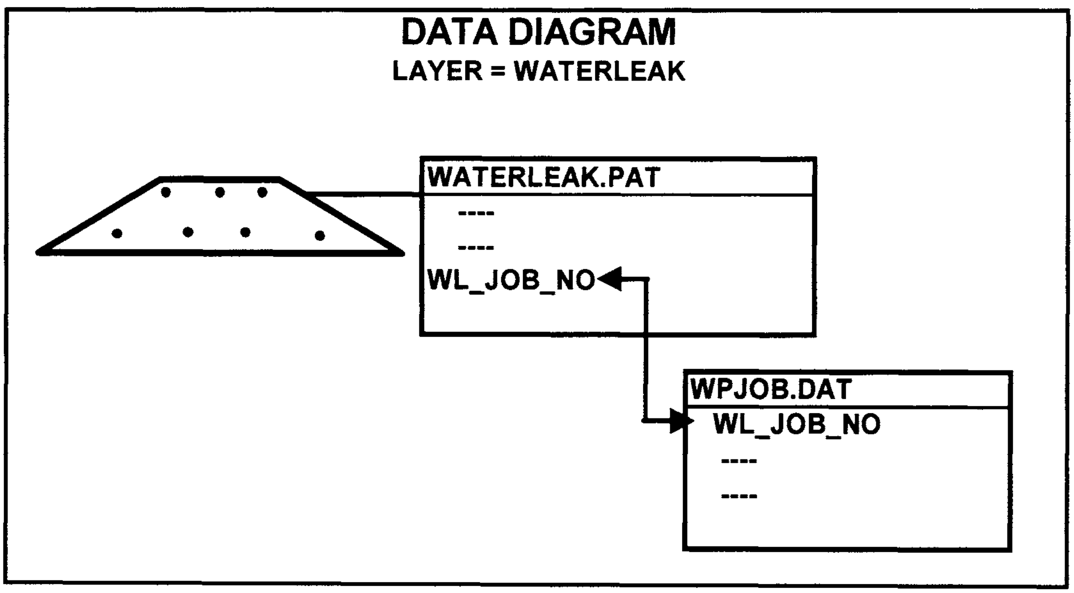


Figure 6-4. Data Dictionary Data Diagram for Water Leak Layer