第 1 章 Geodatabase

1、1 Geodatabase 概述

Geodatabase 概述

ArcGIS 操作基于 GIS 文件格式和存储于 geodatabases 中的地理信息。Geodatabase 是 ArcGIS 的本地数据结构，是用于编辑和数据管理的基本数据格式。

Geodatabase 能将空间数据存储在文件、MDB 文件或者大型 DBMS 中。以上三种存储方式的区别在于可存储数据量的不同以及可支持的并发用户数量不同，能够实现从小数据量、单用户的文件数据库到大数据量、多用户并发编辑的企业级 DBMS 的不同层次的应用。

理解 geodatabase

简介：geodatabase 是大量不同类型的地理数据集的集合。在这一章节可以学习 geodatabase 的基础，这些概念能够为学习和有效使用 geodatabase 打下坚实的基础。

Geodatabase 定义：geodatabase 是大量不同类型的地理数据集的集合，这些地理数据集可以存储在普通的文件中、微软的 Access 数据库中或者多用户的关系数据库(比如 Oracle, Microsoft SQL Server, 或者 IBM DB2）。

Geodatabase 中主要数据集：数据集是 geodatabase 中的一个重要概念。它是 ArcGIS 用来组织和运用地理信息的基本机制。Geodatabase 包含三种主要的数据集类型：要素类、栅格数据集和表。

在创建 Geodatabase 时，首先生成不同的 Dataset 类型，然后添加或者扩展 Geodatabase 基本要素的能力，例如添加拓扑、网络、子类以实现 GIS 行为建模、维护数据完整性以及建立空间关系。

Geodatabase 的存储

Geodatabase 的存储不仅包括简单的空间坐标和属性数据的表格，还包括这些地理数据集的模式和规则。Geodatabase 的三种基础数据集（要素类，属性表和栅格数据集）和其他的 geodatabase 元素都以表格的形式存储。在地理数据集中空间表示或者以矢量要素的形式存储，或者以栅格数据存储。几何对象和传统的属性字段一起存储在表的列中。Geodatabase 将地理要素以表格的形式存储，每行记录代表一个要素。下表是多边形要素，在 Shape 列为每个要素存储了多边形类型，值 Polygon 用于描述 Shape 列的几何坐标和几何形状，表示在每一行定义了一个 polygon。

Geodatabase 的一个关键策略就是利用 RDBMS 来管理从简单要素集到海量数据集，多用户并发操作的 GIS 数据集。二维表为几何数据集提供了基本的存储机制。SQL 语言具有强大的查询和操作表的功能，geodatabase 正是基于想利用这些功能而设计的。

空间数据在下列数据库中支持 Sql 访问空间信息：

可带或不带 oracle spatial 和 oracle locator 的 oracle

IBM DB2

IBM Informix

内在的 SQL ApI 是基于 标准 Sql Spatial 和 OGL 的简易特征 SQL 规范的，这一规范对空间向量类型以标准 Sql 扩展。

用高级数据类型扩展要素类、栅格数据集和属性表

大量的 geodatabase 元素用来扩展简单的表，要素类、栅格数据集，包括添加丰富的行为，数据完整性、数据管理等能力。Geodatabasse 的模式包括定义、完整性规则和行为。具体包括 coordinate systems, coordinate resolution, feature classes, topologies, networks, raster catalogs, relationships, domains 等等。模式信息存储在 DBMS 的 geodatabase 元数据表的集合

中，这些表定义了数据的完整性和行为。

Geodatabase 元素

不管 ArcGIS 用户使用何种操作系统，他们都操作三种的基本数据集类型。即一系列的要素类（如 ESRI 的 shapefiles），或者大量的属性表（如 dBase 文件、 Microsoft Access 表、Excel 电子表格、DBMSs 等等），还有大量的影像和栅格数据集。

原则上，所有的 geodatabase 都包含相同类型的内容，设计 GIS 数据库的第一步就是生成不同类型的数据集。

实际上，用户需要扩展他们的数据模型以支持特定的能力，geodatabase 有大量数据元素和数据集类型用来扩展基本的数据集类型。

Geodatabase 的事务处理和版本管理

ArcSDE geodatabases 支持版本和长事务处理

Geodatabase 支持丰富的数据类型，如 annotation, topology, networks, terrains, and address locators 等等，这些类型都可以应用于海量、高性能的数据库。Geodatabase 同时还支持长事务框架，以支持多种数据管理工作流和操作。

大多数情况下需要多用户并发编辑

以 Check out 和 check in 方式更新

多个复制品之间由共享只供改变更新的多个同步拷贝可以是 DBMS 类型的任何数字 (譬如 Oracle 和 SQL 服务器) 并且不需要被连接。

创建、管理和使用历史归档数据

1、2 表基础

表基础

Geodatabase 中的属性存储在二维表中，建立在以下一系列简单但根本的关系数据概念：

表包含行。

表中所有的行含有相同的列。

每一列具有一个数据类型，如 integer, decimal number, character, and date。

一系列的相关函数和操作符（如 SQL）可以用于操作表和表的数据元素。

和在传统的数据库中的应用一样，表和关系在 ARCGIS 中同样扮演着重要的角色。表中的行可以用于存储地理对象的所有属性，包括在 Shape 列存储和管理要素的几何图形。

下表解释了两个表如何通过使用一个公共的字段使它们的记录相互关联。

Geodatabase 支持的属性数据类型

在 geodatabase 中支持很多数据类型存储和管理属性列，包括很多数值类型、文本类型、日期类型、二进制大对象类型和全球唯一标识码（GUIDS）。

在 Geodatabase 的数据表支持的数据类型包括：

Number：包括 short integers, long integers , single-precision floating-point numbers (通常称为单精度浮点类型)和 double-precision floating-point numbers (通常称为双精度浮点类型)。

Text：任何一定长度的文字数字式字符的集合。

Date：存储日期和时间数据。

BLOBs：二进制大对象用于存储和管理二进制信息，如符号和 CAD 几何图形。

Global Identifiers：GlobalID 和 GUID 数据类型存储用波形括号括起来的由 36 个字符组成的 registry style 字符串。

XML 列类型也可以通过程序接口支持 ，XML 列可以包含任何格式化 XML 内容（如元数据 XML）。

扩展表

在 geodatabase 中表为要素类、栅格数据集和传统的属性表提供了详细的信息描述，用户也可以实现对表的关系操作。

在 geodatabase 中有以下一些可选的功能用于扩展表的能力，它们包括以下几种：

1、3 要素类基础

要素类基础

要素类是具有相同几何类型和属性的要素的集合。在 geodatabase 中常用的要素类有四种：点、线、多边形和注记。

在下面的图示中，点、线、多边形和注记用于表示同一地区的四种数据集。（1）入孔井盖用点表示（2）下水道用线表示（3）地块用多边形表示（4）街道名称用注记表示。

在这个图示中，也许已经注意到一些潜在的对高级要素属性建模的要求。比如，下水道管线和入孔井盖组成了一个雨水管线网络。同时，相邻的地块共享了公共的边界，大多数地块用户通过使用 topology 来保持在数据集中共享要素的完整性。

Geodatabase 中要素类的类型

矢量要素是最常用的表达地理数据的数据格式，它适合于表达离散的有明显边界且稳定存在的地理对象。如井、街道、河流、州和地块。一个要素是一个简单对象，它存储其几何图形，它可以是一个点、线或者面，作为这条记录中的一个属性或者字段。在 ArcGIS 中，要素类是具有相同空间参考和属性设置的相同要素的集合。例如，用一个线要素来表达道路中心线。

注意：当在 geodatabase 中创建一个要素类时，需要设置要素类型以定义要素类的类型（点、线、多边形等等）。

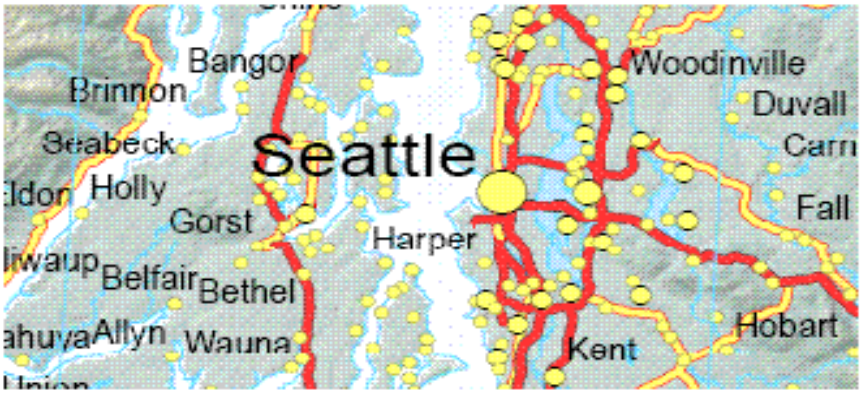
通常，要素类是点、线或多边形的集合，具体有 7 种要素类型。

1、点－用来表达那些很小且不能用线或多边形来表示的地理要素（如 GPS 观测站）。

2、线—用来表达那些长条形的，非常狭窄也不能用多边形表示的地理要素，如街道中心线和溪流。线也可以表达那些有长度却没有面积的要素，如轮廓线和边界。

3、多边形－是一个封闭的图形，用来表示均质要素的形状和位置，如州、县、土地、土壤类型或者土地利用类型区等。

4、注记－地图文本，包括文本如何组织的属性；例如，除了每个注记的文本字符串。还包括其他属性，如放置文字形状，字体，字体大小和其他显示属性。注记也可以是 feature-linked，也可以包含子类。



5、维度—一种特殊的注记，它显示具体的长度或距离；例如，为了显示一个建筑物或一块地的一条边的长度，或者两个不同要素之间的距离。维度经常用于 GIS 设计、工程和工具应用中。

6、多点—由不止一个点组成的要素。Multipoints 通常用于管理大量点集合的数组，如激光雷达点串，它们可以包含几十亿个点。使用一个单独的列表示这样的点几何是不可行的。聚类这些多点行可以使 geodatabase 处理大块点集合。

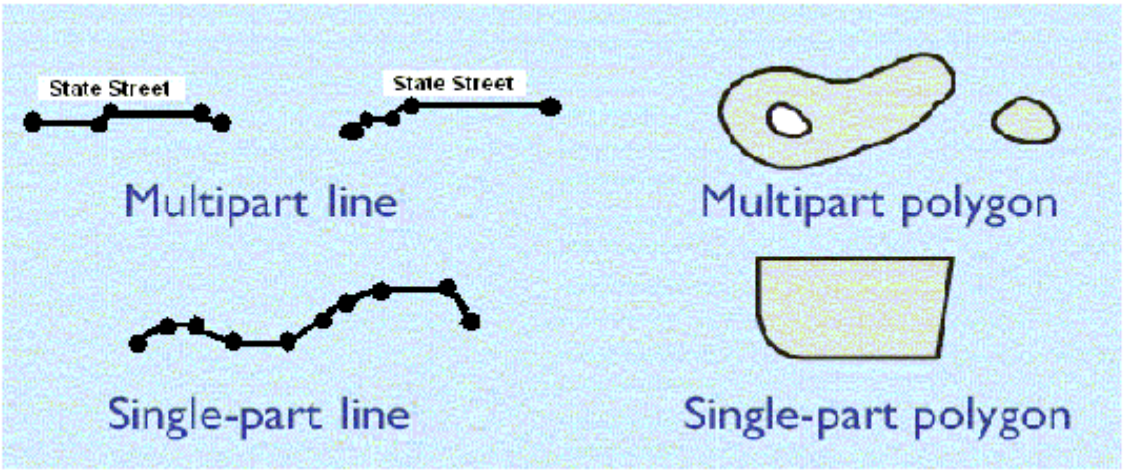
7、Multipatches—一个 3D 几何图形用于表达要素的外部表面或者外形，这个要素占用一个离散的二维面积或者三维空间的体积。 Multipatches 由平面的 3D 环和三角形组成。Multipatches 可以被用于表达任何简单到复杂的对象，如球体和立方体到表面和建筑物。

要素的几何特征系统

要素类不仅包含每一要素的几何形状，还包含描述要素的属性。一个要素可以用点、线或多边形三种要素类型来定义，但是其他的地理属性也可以定义，如要素是单部件或多部件，可以有 3D 定点，可以有线性测量（称为 m－值），可以曲线。这一部分对这些功能作了简单介绍。

单部件和多部件的线，多边形

Geodatadase 中线和多边形要素可以有单部件或多部件组成。如，一个州可以包含多部件（夏威夷岛），但是通常认为它是一个单独的要素。



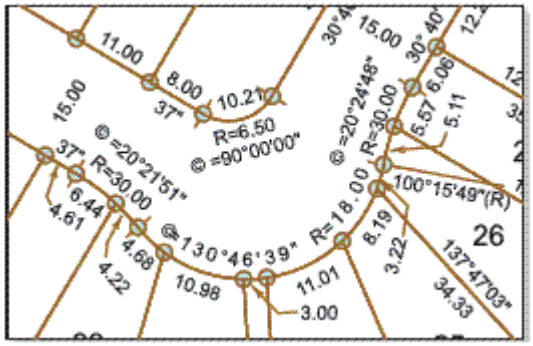
顶点、线段、高程值和测量值

要素几何图形主要是由坐标顶点组成的。线和多边形要素中的线段跨多个顶点。线段可以是直线边线或者是参数定义的曲线。要素中的顶点也可以包含表达高程值的 Z 值和表达沿着线要素的度量值的 M 值。

线段类型

线段和多边形可以由两个关键元素定义（1）定义线段或多边形的几何形状的一系列有序定点（2）连接每对定点之间的线段类型。因此，每一条线段或一个多边形可以认为是由一系列可以连接的有序定点形成的几何形状。还有一种表达线或多边形的方法是：每一条线段或一个多边形是一系列有序的可以连接的线段，每一线段可以是以下类型中的一种：直线

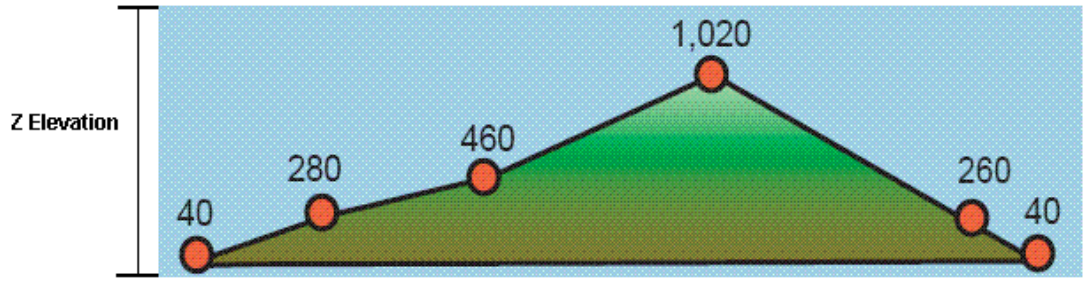
段、圆弧、椭圆弧或 Bezier 曲线。



默认的线段类型是直线段，但是，当需要定义 curves 或 parametric shapes 时，可以有三种其他线段类型选择：圆弧、椭圆弧或 Bezier 曲线。这些几何形状常常用于表达建筑环境，如地块边界和车道线。

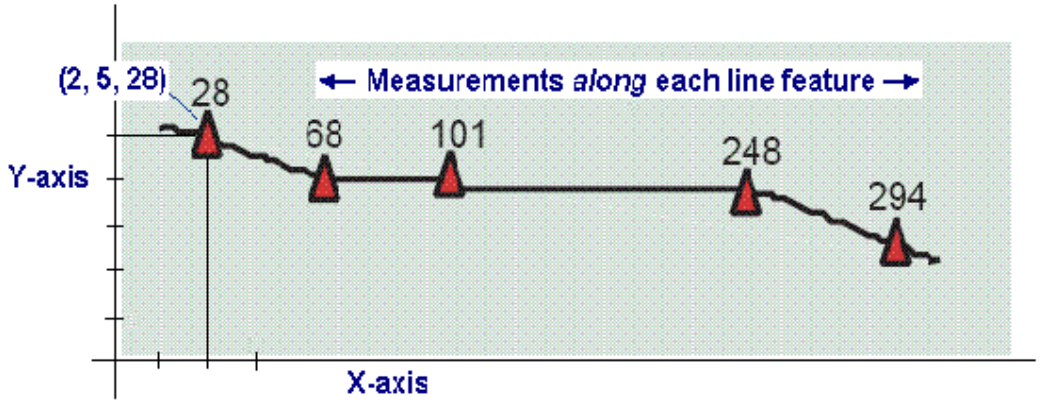
用 Z 值作垂直量度

要素的几何坐标可以包括 x、y 或 x、y、z。Z 值大多数情况下表示高程值，但它们也可以用来表示其他的属性，如降雨量等或空气质量值。



用 m 值作线性量度

线性要素的每一个顶点也可以指定一个 M 值或线性量度。一些 GIS 应用利用线性量度方法沿着线性要素内插距离，如道路、溪流和管线。可以给点、多点、折线或多边形的每一个点指定一个 M 值或线性量度。线性度量方法的一个典型例子就是公路或运河沿线的里程标志或码头。



含线性量度的顶点的坐标可以是（x，y，m）或者（x，y，z，m）。

支持这些数据类型通常称为线性参照（Linear Referencing）。沿着这些量度系统的发生的地理定位事件的过程称为动态分割（Dynamic Segmentation）。

在 ArcGIS 的线性参照的实现中，术语 route 引用任何线性要素，如一个城市街道、高

速公路、河流或者管道，它具有唯一一个标识符和沿着每个线性要素的一个通用度量系统。用一个通用度量系统的 routes 集合建立在一个线性要素类上，如下所示：

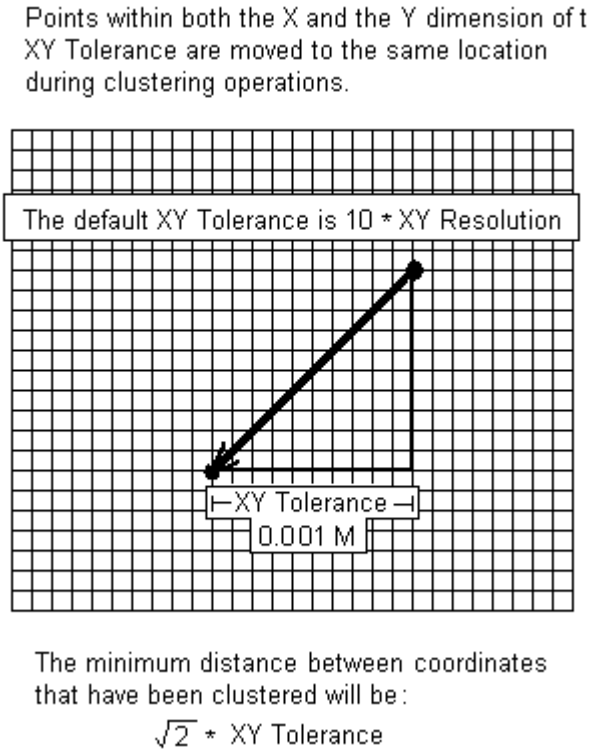
要素容差

在各类 ArcGIS 操作中，使用一些关键的地理属性来处理和管理要素坐标。这些属性在每个要素类或者要素数据集创建的时候被定义。

X,y tolerance

当创建一个新的要素类时，将被提示设置 x,y tolerance。x,y tolerance 被用于在聚类操作中设置坐标之间的最小距离，如拓扑验证、缓冲区生成和多边形叠加等编辑操作。

要素处理操作是被 x,y tolerance 影响的，它决定了在这些操作中所有要素坐标的最小距离。通过定义，它也定义了在聚类操作中一个坐标在 x 或 y 方向（或者两者）可以移动的距离。



x,y tolerance 是及其小的距离。它用于在聚类操作中解决坐标之间不精确的相交位置。当使用几何操作处理要素类时，如拓扑验证、多边形叠加、缓冲分析和裁剪等，坐标之间的 x 距离和 y 距离落在 x、y tolerance 内时，则被认为是一致的点。因此，聚类坐标被移动到一个公共的位置。通常，较低精确的坐标被移动到较高精度坐标的位置，或者一个新的位置在聚类的坐标之间作为一个加权平均值距离被计算。在这种情况下，加权平均值距离是基于聚类坐标的精度级别的。

聚类处理通过移动整个地图内的落入 x、y tolerance 的坐标来进行。ArcGIS 使用这个运算法则发现、清除和管理要素之间的共享几何图形。这意味着这些坐标被认为是一致的。这是许多 GIS 操作和概念的根本。

在这些操作中一个坐标可以移动到新位置的最大距离是√2 倍 XYTolerance。

默认的 x、y tolerance 是 0.001 meters（地理坐标系统）。例如，如果是在 state plane feet 坐标系统中，默认的 x、y tolerance 是 0.0003281 feet (0.003937 inches)。如果是 latitude-longitude 坐标系统，默认的 x、y tolerance 是 0.000000008982995 degrees。

x、y tolerance 的默认值是 10 倍的默认 x,y resolution，这是在大多数情况下推荐使用的。

注意：x、y tolerance 不是打算用于生成几何形状。相反，它是用于在拓扑操作中集成线和边界的。这意味着集成彼此落入非常小距离的坐标。

有用的提示：

一般地，使用 10 倍 x,y resolution 的一个 x,y tolerance；

为了保持较小的移动，保持 x,y tolerance 很小。但是，一个太小的 x,y tolerance 可能不能准确地集成一致性边界的线元素和坐标；

相反，如果 x,y tolerance 太大，要素坐标可能相互覆盖。这可能降低要素边界表达的准确性；

x,y tolerance 应该永远不要接近 data capture resolution；

在拓扑中，可以设置每个要素类的 coordinate rank。

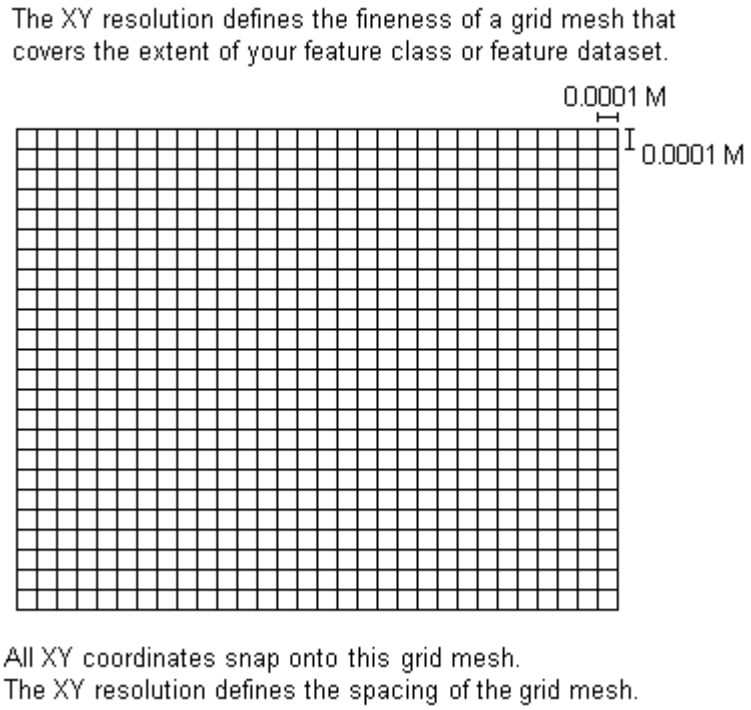
X,y resolution

要素类或者要素数据集的 x,y resolution 是用于存储 x、y 坐标值的数值精度。精度对准确的要素表达、分析和制图是很重要的。

x,y resolution 定义了小数点后可以保留的小数位数，或者存储要素坐标的有效数。坐标值在 ArcGIS 以 integers 被精确存储和操作。

默认的要素类的 x,y resolution 在 ArcGIS9.2 中是 0.0001 meter（地理坐标）。如果要素是被存储在 State Plane feet 中，默认的 x,y resolution 是 0.0003281 feet (0.003937 inches)。如果是 latitude-longitude 坐标系统，则默认的 x,y resolution 是 0.000000001 degrees。

以下是一个关于 x,y resolution 的概念视图的图表。



如果必要，可以不用管默认的 x,y resolution 值，设置要素类或者要素数据集的另一个 x,y resolution 值。设置一个较小的 x,y resolution 值，相比一个较大的 x,y resolution 值，可能潜在地增加了数据存储和处理的时间。

设置 X,y tolerance 和 X,y resolution

当在 ArcCatalog 中建立一个新的要素类，将被询问设置新要素类的 x,y tolerance。数据集坐标系统的默认 x,y tolerance 值如下面板所示：



在大多数情况下，默认的 x,y tolerance 是最好的选择。

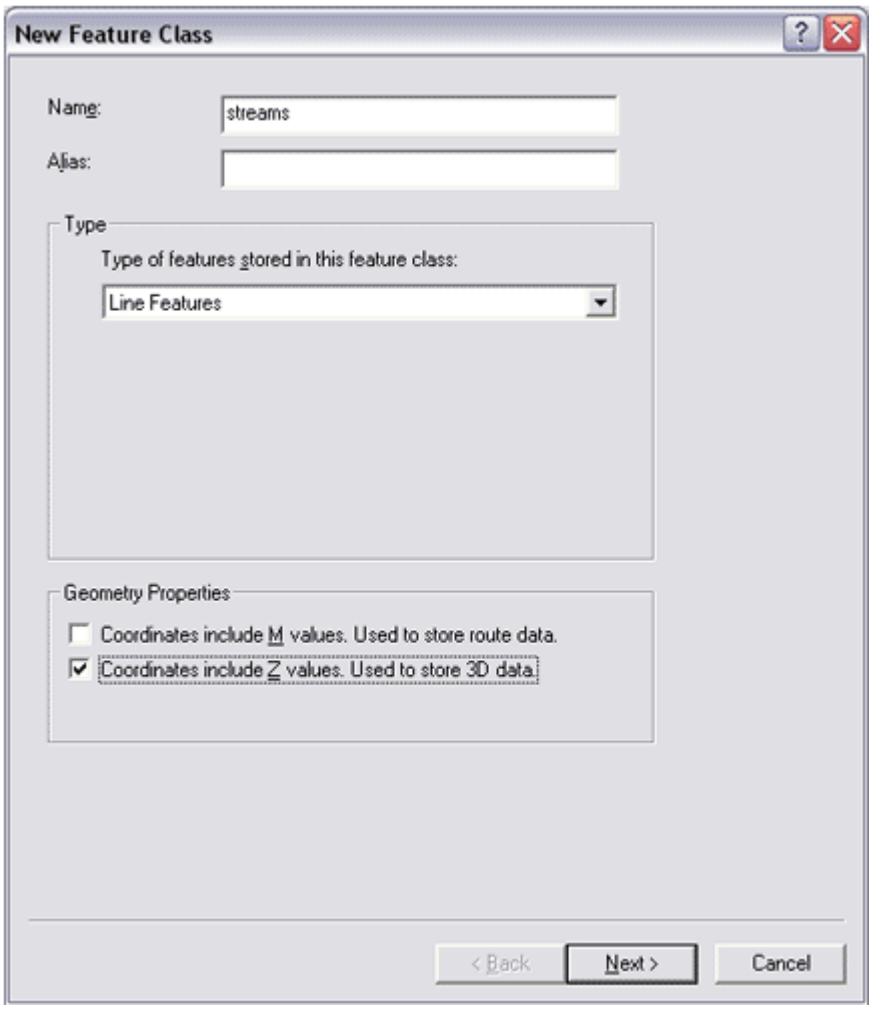
在这个面板上，也可以选择接受默认的 x,y resolution，它是 0.0001 meters 或者与坐标系统单位相等的量。默认的 x,y resolution 在大多数情况下，是合适的，除非需要比测量和土木工程更好的精度存储坐标值。

为了设置自己的 x,y resolution，取消选择上面面板中的 Accept default resolution 复选框，到下一个面板中设置需要的 x,y resolution 值。

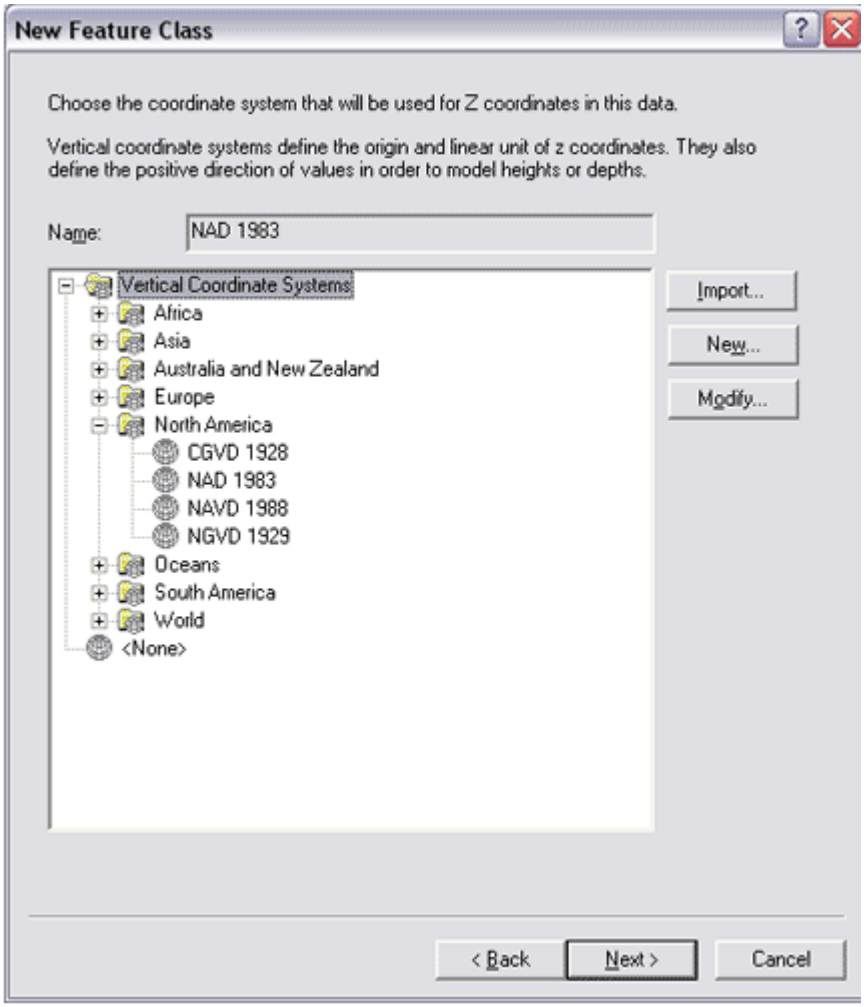
Z-resolution, z-tolerance 和垂直坐标系统

Z-coordinates 主要用于包含高程的 GIS 要素中。但是，Z-coordinates 也可以被用于其他垂直度量，如空气污染观察数据、温度和其他度量值。

当建立一个新要素类时，需要包含 Z-coordinates，可以选择 Coordinates include z values 复选框：



当建立了一个含 Z-coordinates 的要素类，将随意设置一个垂直坐标系统(VCS)：



指定一个垂直坐标系统的主要目标是数据集文档化，在 VCS 中定义的 z-units 将被用于 ArcGIS 的 3D 处理操作。

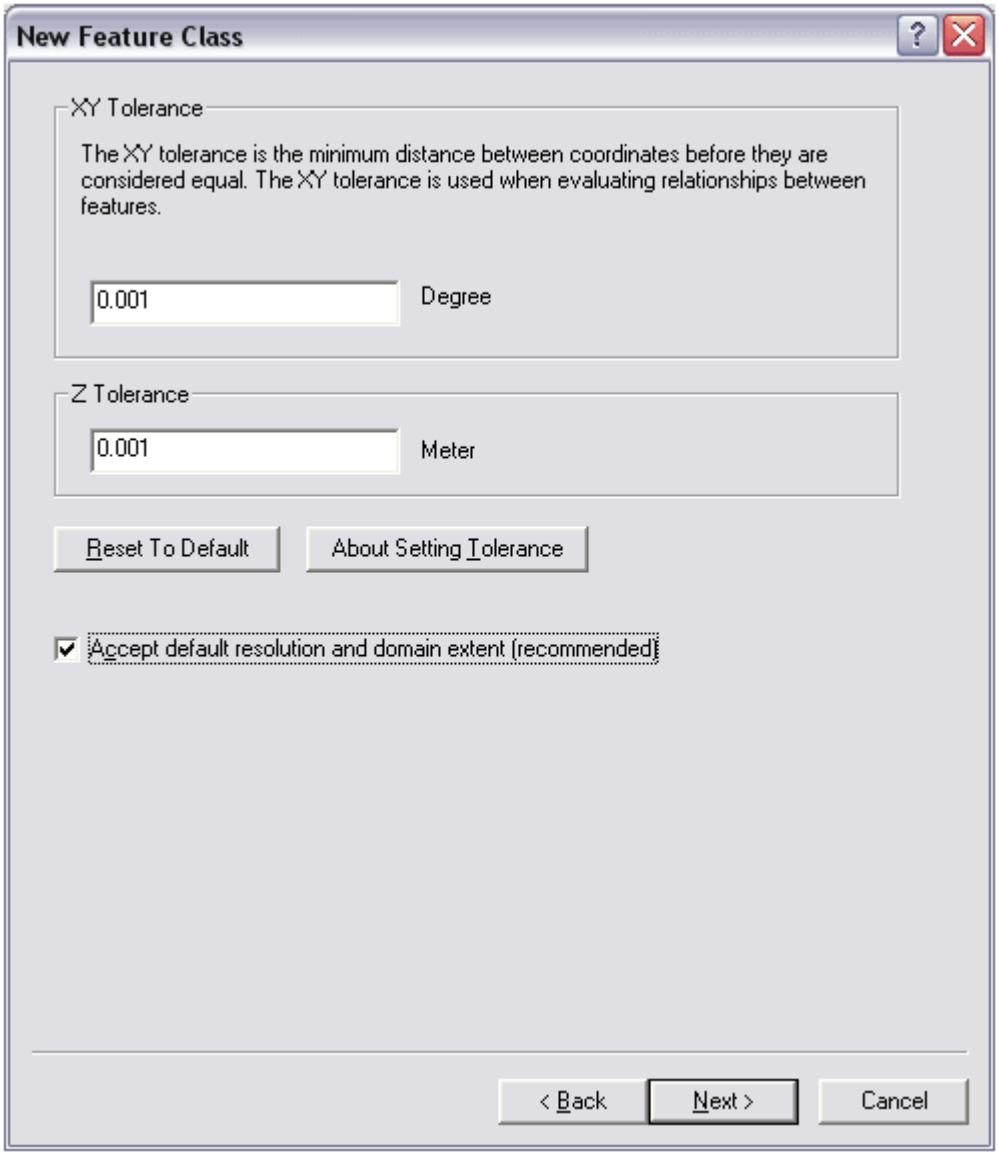
大多数 z-units 值以 x,y 坐标相同单位来度量海平面以上或者以下的高度，但是，还有一些其他的 z 度量，它可能需要一个不同的 z-tolerance 和 z-resolution 值。

Z-tolerance 和 z-resolution

z-tolerance 定义了处理过程中聚类 z-values。默认的 z-tolerance 是 0.001 meters 或者定义的 VCS 的单位的相等的量。

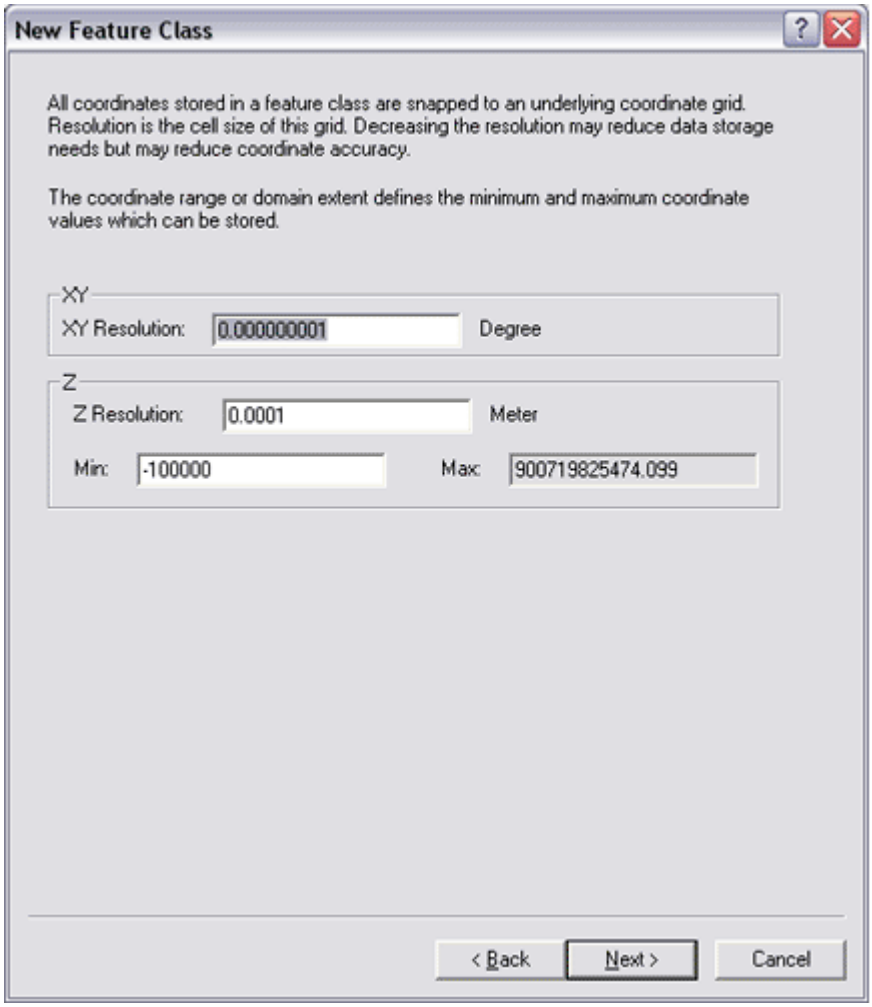
如果坐标是一致的，它们的 z-values 落入了 z-tolerance，它们的 z-values 将被设置为相同。

当建立一个新的要素类时，可以接受默认的 z-tolerance 的值，或者选择设置自己的值。



在大多数情况下，默认的 z-tolerance 能很好工作，除非 z-coordinate units 与 x,y coordinates 有很大的不同。

在与上面相同的一个面板上，可以选择接受默认的 z-resolution 值。如果选择修改 x,y resolution、z-resolution 或者两者，取消选择 Accept default resolution and domain extent 复选框，将出现一个面板，在其中可以看到默认的 z-resolution 和 z-values 的最大和最小值。



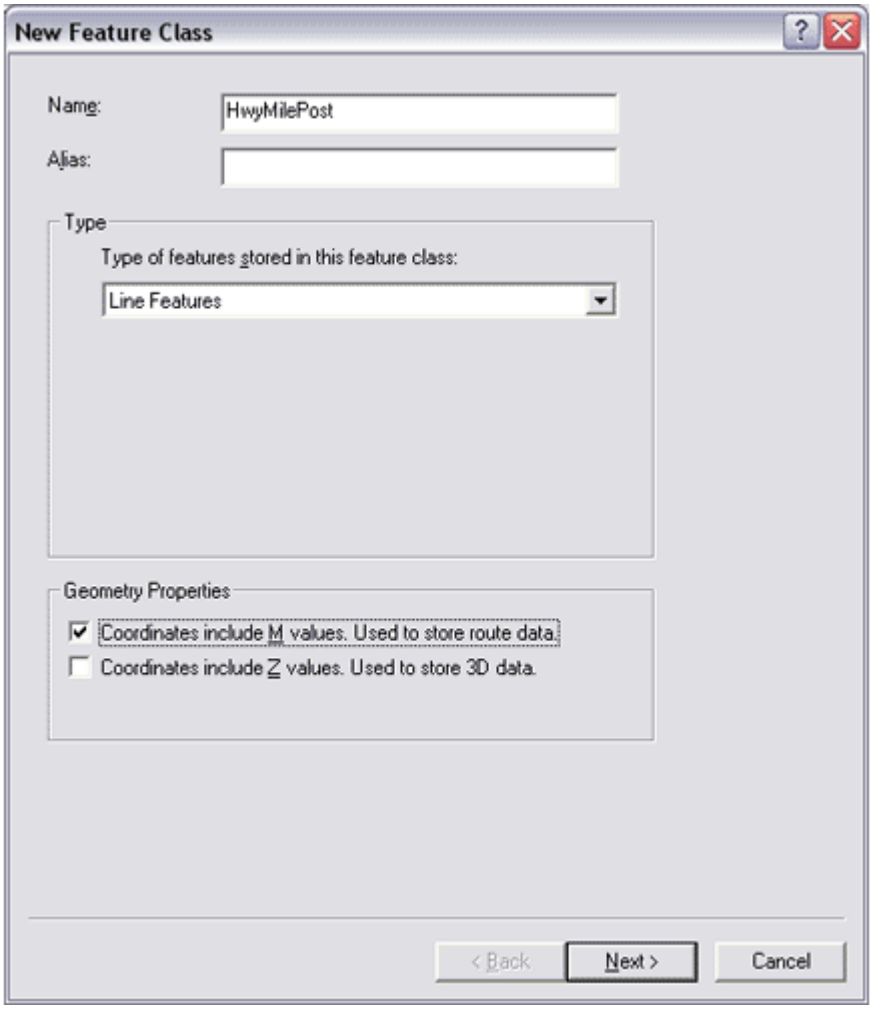
对于 z-tolerance，在大多数情况下，接受默认的 z-resolution 和 z-range 将能很好地工作。

Z-resolution 定义了 z-values 的 resolution。默认的 z-resolution 是默认的 z-tolerance1/10 倍。

线性参照的 M-resolution 和 m-tolerance

当建立含 m-measures 的要素类，可以设置 m-measures 值的两个属性：

在 ArcCatalog 的 New Feature Class 对话框上，通过选择 Coordinates include M values 复选框指定含 m-measures 的要素类。



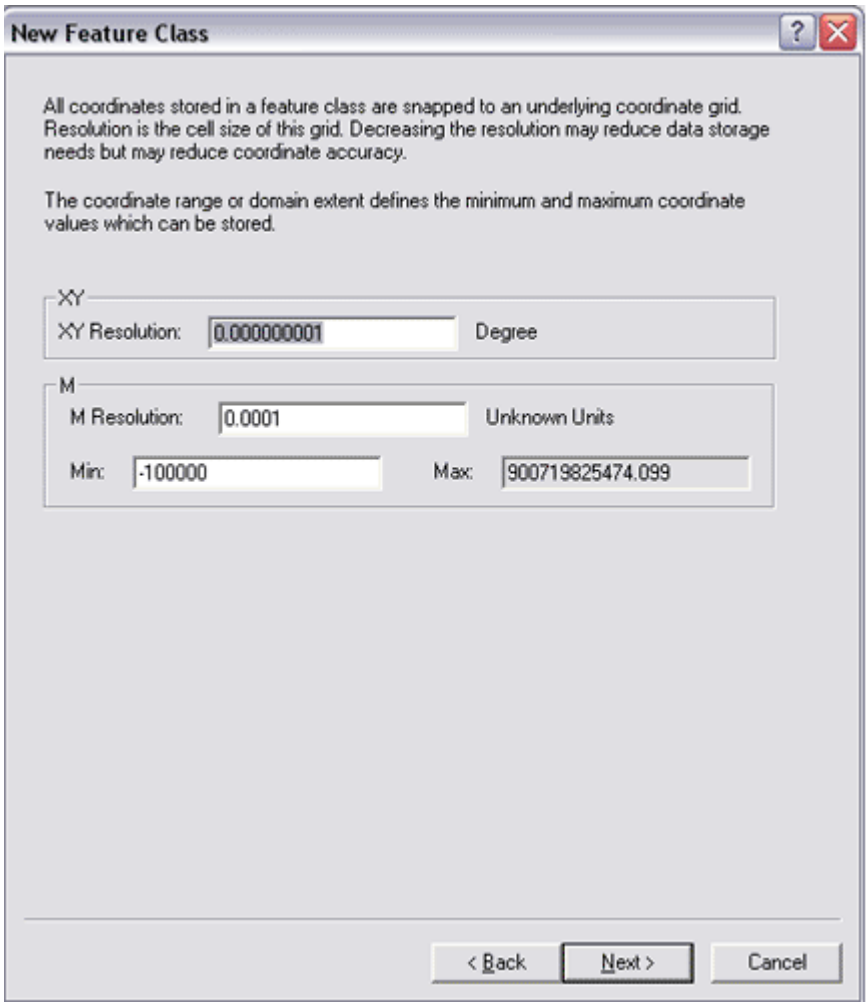
在接下来的面板中，可以设置 m-tolerance 或者接受默认值。m-tolerance 定义了处理过程中 m-values 的 tolerance。它默认值是 0.001 units。如果坐标的 x 和 y 一致，它们的 m-values 落入 m-tolerance，那么它们的 m-values 将被设置为相等。



大多数情况下，接受默认的 m-tolerance 能满足工作。但是，如果 m-measure 值相当小，需要保证 m-measure 不聚类，可以指定自己的 m-tolerance。

在同一个面板中也可以选择修改默认的 m-resolution：取消选择 Accept default resolution and domain extent 复选框。如下所示：

在接下来的面板中可以设置一些其他的 m-coordinate 的属性：

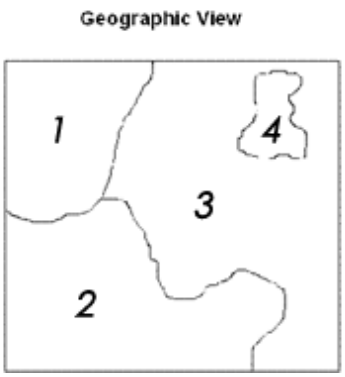


在此，M-resolution 定义了自己 m-values 的 resolution。它的默认值是 0.0001 units。

通过设置 m-values 最小值(Min)和最大值(Max)来设置 m-values 的范围

geodatabase 中的要素类存储

在 geodatabase 中，每个要素类在一个单独的表中被管理。每行中的 Shape 列用于存储每个要素的 geometry 或 shape。



Tables View

Object ID	Shape	Name	LV Code	Management Agency
1		Shady Pines	20	Private
2		Pinewood Village	30	Pinewood Village Association
3		Sarah Park	80	City Park Board
4		Town Park	99	City Park Board

在要素类表中：

每个要素类是一个表；

单独的要素被存储为行；

Shpelie 存储每个要素的 geometry (point, line, polygon)；

Object ID 列存储每个要素的唯一标识符。

在 ArcSDE geodatabases 中，关系数据库存储每个要素类为一个表。有三种 DBMSs(Oracle, DB2, 和 Informix)提供了 SQL 访问 geodatabase 中的要素的 geometry。

扩展要素类

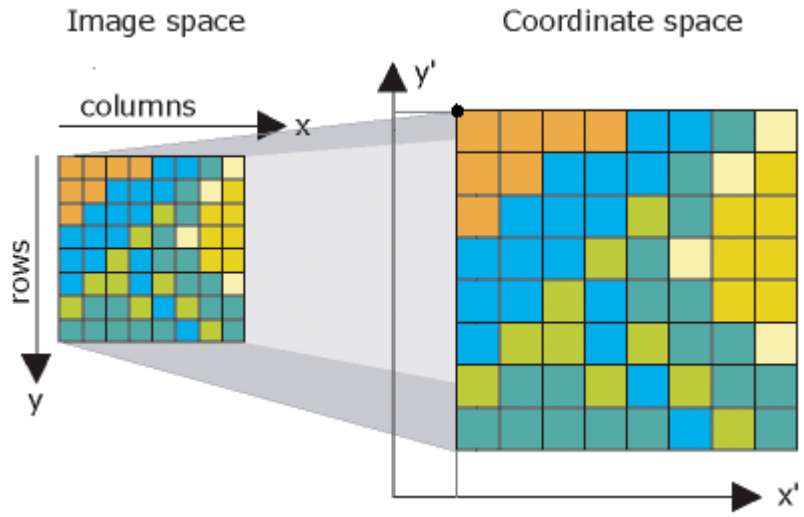
Geodatabase 中的每个要素类是具有相同几何形状、相同属性和相同空间参考的地理要素的集合。要素类可以根据需要被扩展。以下是一些用于要素类扩展的方法：

Use:	If you need to:
Feature Dataset	Hold a collection of spatially related feature classes or build topologies, networks, and terrains.
Subtypes	Manage a set of feature subclasses in a single feature class. This is often used on feature class tables to manage different behaviors on subsets of the same feature types.
Attribute Domains	Specify a list of valid values or a range of valid values for attribute columns. Use domains to help ensure the integrity of attribute values. Domains are often used to enforce data classifications (such as road class, zoning codes, and land-use classifications).
Relationship Classes	Build relationships between feature classes and other tables using a common key. For example, find the related rows in a second table based on rows selected in the feature class and so on.
Topology	Model how features share geometry. For example, adjacent counties share a common boundary. Also, county polygons nest within and completely cover states.
Network Dataset	Model transportation connectivity and flow. (Must have Network Analyst installed to use)
Geometric Network	Model utilities networks and tracing.
Terrain	Model triangulated irregular networks (TINs) and manage large lidar and sonar point collections.
Address Locator	Geocode adressess.
Linear Referencing	Locate events along linear features with measurements.
Cartographic Representations	Manage multiple cartographic representations and advanced cartographic drawing rules.

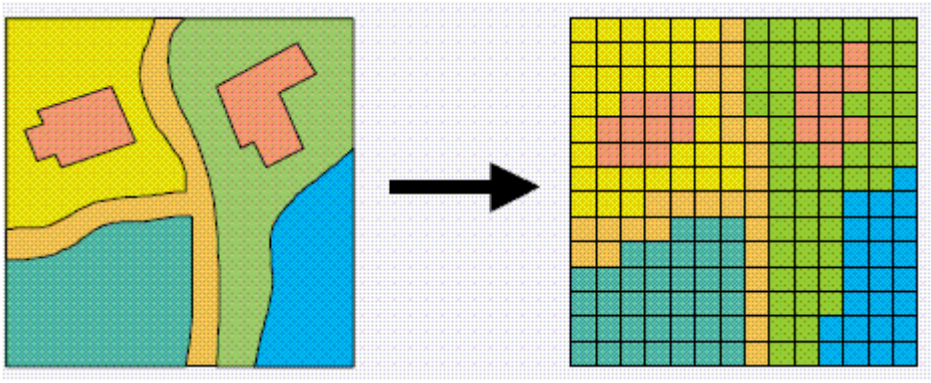
1、4 栅格基础

栅格基础

栅格数据集是采用规则的格网来表示地理要素的，每个规则格网单元用于表达这个位置的一些属性。



栅格数据集通常用于表示和管理影像、数字高程模型和大量其他的现象。通常栅格作为一个表达点、线和多边形要素的方法而使用。在下面的例子中，可以看到一系列多边形被表达为一个栅格数据集。



栅格被感兴趣是基于两个原因：一，它们可以被用于表达所有的地理信息（要素、影像和表面），二，它们具有一套丰富的分析处理运算符。因此，栅格作为一种通用的在 GIS 中存储影像的数据类型，也被用于表达用于基于栅格模型和分析的要素。

Geodatabase 中的栅格

一个栅格就是按行和列分布的一组格网单元，是被经常用于 GIS 的一种数据集。用户通常使用大量栅格文件，但是许多用户需要在一个 DBMS 中同时管理地理信息和栅格数据。Geodatabase 提供了一个非常有效的方法在 file 和 ArcSDE geodatabase 中管理栅格数据。

栅格数据集管理策略

两种栅格数据集管理策略是重要的：

Raster provisioning。

Rasters in the geodatabase。

栅格数据的地理属性

通常有四个地理属性为栅格数据集存储。这些属性在地理参考和帮助解释栅格数据文件如何结构化方面很有用的。

栅格数据集用一个特殊的方法定义地理位置。一旦格网单元或者象素可以被精确引用，然后获得一个栅格中所有格网单元值的一个顺序列表是容易的。这意味着每个栅格数据集通常有一个 header record 存储它的地理属性，数据的主要部分是格网单元值的一个顺序列表。

一个栅格的地理属性通常包括：

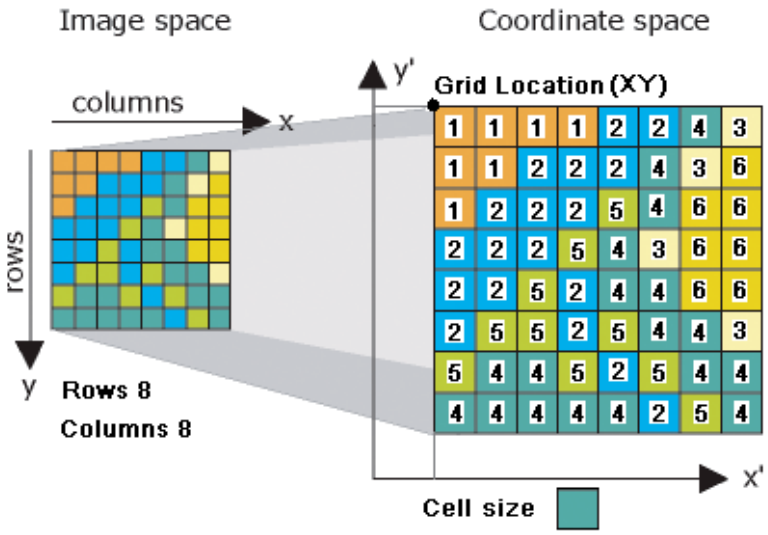
坐标系统；

参考坐标系统或者 x，y 位置（通常指左上角或者左下角）；

一个格网的大小；

行和列的数量。

这个信息可以被用于找到任何具体的格网单元的位置。获得这个信息后，栅格数据结构容易地列出所有的格网单元值，从左上角开始，沿着每一行，然后到最右下角。如下图所示：



List of cell values

[1111224311222436122254662225436622524466255254435445254444444254]

Geodatabase 中的 raster block table

栅格数据通常在大小上比要素要大很多，需要一个 side table 存储。例如，一个正常的正射影像有 6700 行*7600 列。

为了取得这些大栅格数据集的高性能，一个 geodatabase 栅格被分割为较小的片（称为 blocks），通常为 128 行*128 列或者 256 行*256 列。这些较小的 blocks 然后存储在 side table 中。如下图所示：

Geographic View

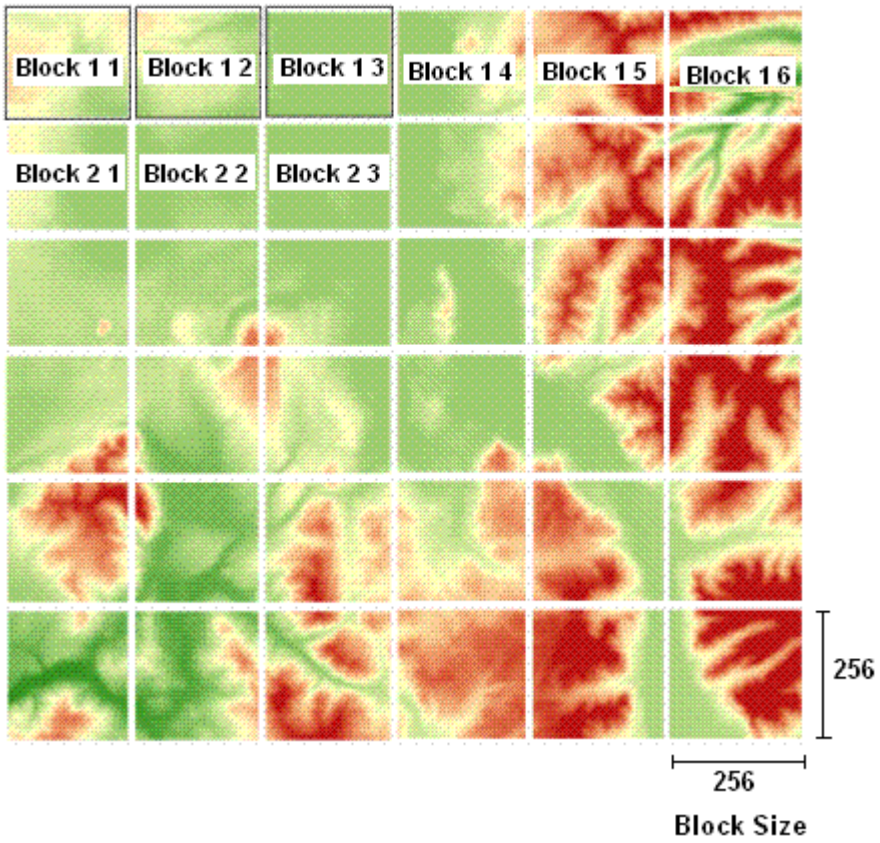


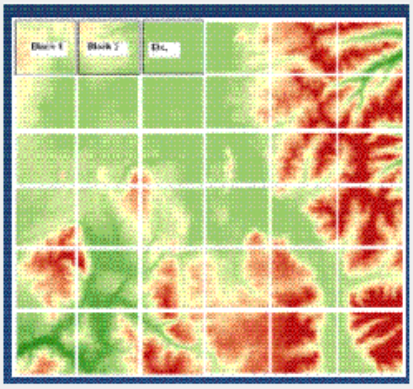
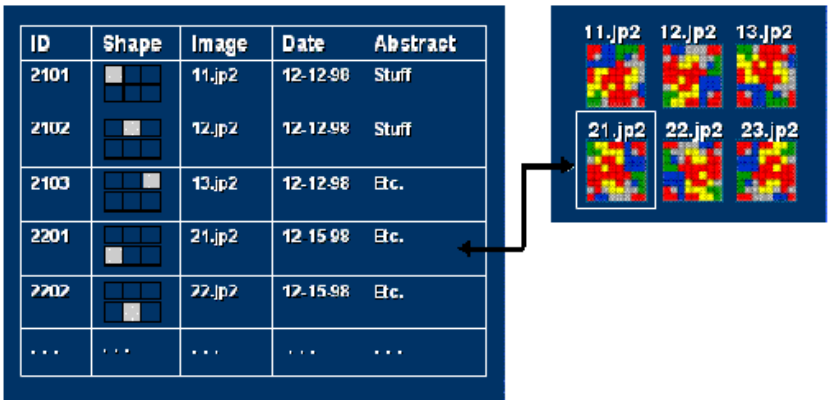
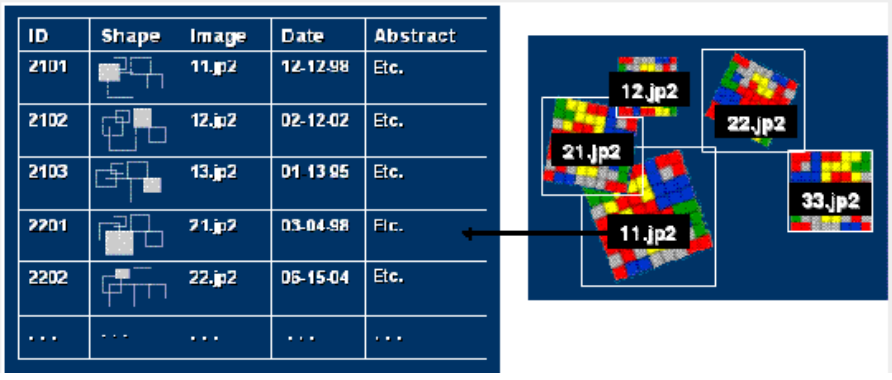
Table View

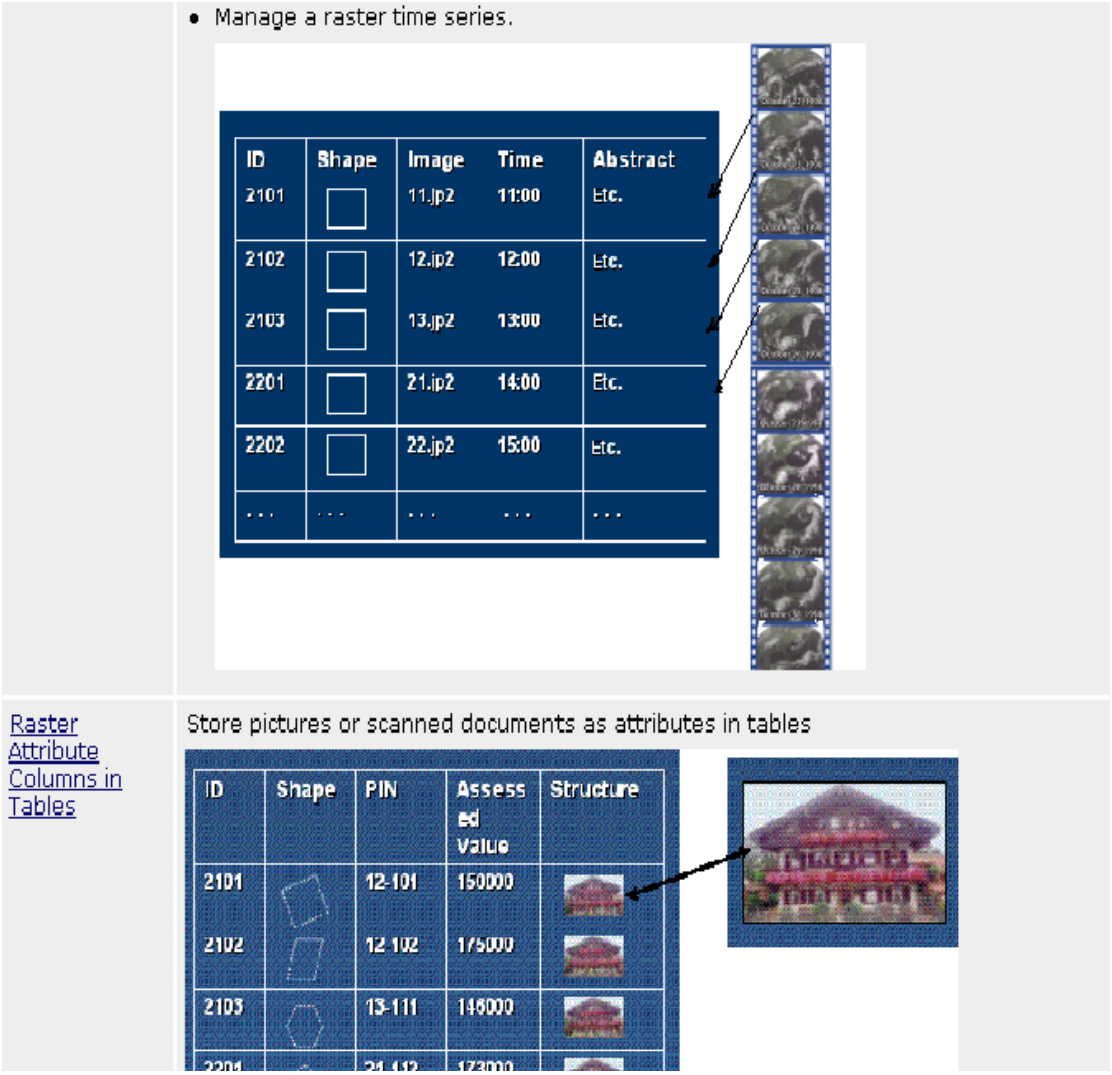
BAND	RRD	ROW	COL	BLOCK DATA
1	4	1	1	
1	4	1	2	
1	4	1	3	
1	4	1	5	
1	4	1	5	
1	4	2	1	
1	4	2	2	
Etc.				

扩展栅格数据

栅格数据集被日益广泛地用于 GIS 应用。Geodatabase 可以管理栅格为：一个单独的数据集、数据记得逻辑集合和表中的图片属性。

大量 geodatabase 功能允许用户扩展如何管理这些栅格信息，如下表所示：

Use	If you need to
Raster Datasets	Manage very large, continuous image datasets and image mosaics. 
Raster Catalogs	Accomplish a number of purposes, including • Manage a tiled image layer, where each tile is a separate image. 
	• Manage any series of images in a DBMS. 



1、5 Geodatabase 类型

geodatabase 类型

Geodatabase 是一个容器用于存储数据集集合。它有三种类型：

1、File Geodatabases—以文件系统中的文件夹存储。每个数据集以一个文件被存储，可达到 TB 级。与 personal geodatabases 相比，推荐使用这种类型。

2、Personal Geodatabases—所有的数据集被存储在一个 Microsoft 的 Access 数据文件中，它的容量限制为 2GB。

3、ArcSDE Geodatabases—使用 Oracle, Microsoft SQL Server, IBM DB2,IBM Informix 存储在一个关系数据库中。这个多用户的 geodatabase 需要 ArcSDE 的使用，在容量和用户数量上没有限制。

三种类型 geodatabase 的比较：

Key Characteristics	ArcSDE Geodatabase	File Geodatabase	Personal Geodatabase
Description	A collection of various types of GIS datasets held as tables in a relational database This is the recommended native data format for ArcGIS stored and managed in a relational database.	A collection of various types of GIS datasets held in a file system folder This is the recommended native data format for ArcGIS stored and managed in a file system folder.	Original data format for ArcGIS geodatabases stored and managed in Microsoft Access data files This is limited in size and tied to the Windows operating system.
Number of Users	Multiuser Many readers and many writers ArcSDE can be licensed for use at three levels: • Personal ArcSDE • Workgroup ArcSDE • Enterprise ArcSDE	Single user and small workgroups Some readers and one writer per feature dataset, standalone feature class or table. Concurrent use of any specific file eventually degrades for large numbers of readers.	Single user and small workgroups with smaller datasets Some readers and one writer. Concurrent use eventually degrades for large numbers of readers.
Storage Format	• Oracle • Microsoft SQL Server • IBM DB2 • IBM Informix	Each dataset is a separate file on disk A file geodatabase is a file folder that holds its dataset files.	All the contents in each personal geodatabase are held in a single Microsoft Access file (.mdb).
Size Limits	Up to DBMS limits	One TB for each dataset. Each file geodatabase can hold many datasets Each feature class can scale up to hundreds of millions of vector features per dataset.	Two GB per Access database Effective limit before performance degrades is typically between 250 and 500 MB per Access database file.
Versioning Support	Fully supported across all DBMSs; includes cross-database replication and updates	Not supported	Not supported
Platforms	Windows, Unix, Linux, and direct connections to DBMSs that can potentially run on any platform on the user's local network	Cross-platform	Windows only
Security and Permissions	Provided by DBMS	Operating file system security	Windows file system security
Database Administration Tools	Full DBMS functions for backup, recovery, replication, SQL support, security, and so on	File system management	Windows file system management
Notes	Requires the use of ArcSDE	Allows you to optionally store data in a read-only compressed format to reduce storage requirements	Often used as an attribute table manager (via Microsoft Access). Users like the string handling for text attributes.

File geodatabases 和 personal geodatabases

File 和 Personal geodatabase，对所有的 ArcGIS 用户都是免费使用的，支持 geodatabase 的全部信息模型。这包括拓扑、栅格目录、网络数据集、表面数据集、地址定位器等。File 和 Personal geodatabase 被设计为由一个单独的用户编辑，不支持 geodatabase 版本。对于 file Geodatabase，可以有多个编辑者同时进行编辑，只要是他们编辑不同的要素数据集、独立要素类或者表。

File geodatabase 是在 ArcGIS9.2 版本中新发布的一种新的 geodatabase 类型。它的目标是：

提供所有用户一个广泛可用、简单和可伸缩的 geodatabase 解决方案；

提供一个跨操作系统的简便的 geodatabase；

提高处理大数据集的能力；

提供了优越的性能和可伸缩性；

使用一个有效的数据结构，它对数据存储和性能是最优化的；

Personal geodatabases 从 ArcGIS8.0 版本开始就被 ArcGIS 使用，使用 Microsoft 的 Access 数据结构（mdb file）。它们支持的 geodatabase 容量限制在 2GB 或者小于 2GB。但是，有效的数据库大小是比较小的，大概在 250MB－500 MB 之间，超过这个范围之后，数据库性能将开始降低。Personal geodatabases 也仅在 Microsoft 的 Windows 操作系统上被支持。

由于多个原因，ArcGIS 将继续支持 personal geodatabases。但是，它也高度推荐使用 file geodatabase。对于基于文件数据集的 GIS 项目使用 file geodatabase 是理想的，对于比较小的工作组，使用 personal geodatabases 比较理想。

ArcSDE geodatabases

当需要一个可以由多个用户同时编辑和使用的多用户 geodatabase，ArcSDE geodatabase 是一个好的解决方案。它添加了管理一个共享、多用户 geodatabase，以及基于版本的 GIS 工作流的能力。利用企业级的关系数据库的功能是 ArcSDE geodatabase 的最大优势。

ArcSDE geodatabase 使用多种 DBMS 存储模型（IBM DB2, Informix, Oracle, 和 SQL Server）。ArcSDE geodatabases 主要被用于工作组、部门和企业范围。它们利用基本的 DBMS 体系支持：

及其大、连续的 GIS 数据库；

多个并发用户；

长事务和版本工作流；

支持 GIS 数据管理的关系数据库；

通过许多 ArcSDE geodatabase 的实现，已经发现 DBMSs 在加载和删除大二进制对象类型的数据很有效。另外，GIS 数据库的容量和支持的用户数量比 GIS 文件系统大得多。

在 ArcGIS 中访问和使用 ArcSDE 有三种级别

ArcSDE geodatabases 包括 Personal ArcSDE geodatabase、Workgroup ArcSDE geodatabase 和 Enterprise ArcSDE geodatabase 三个级别。ArcSDE geodatabase 的功能可以在以下的 ESRI 软件产品中获得：

Personal ArcSDE included with ArcEditor and ArcInfo:从 ArcGIS9.2 开始，ArcEditor 和 ArcInfo 包括免费的 Microsoft SQL Server Express database。这些软件也包括 ArcSDE 的功能支持同时有三个连接的 Personal ArcSDE geodatabase－其中一个可以编辑数据。

SQL Server Express 限制在一个 CPU 上运行，使用 1GB 的内存。SQL Server Express 最大的数据库限制在 4GB。

Personal ArcSDE geodatabase 提供了在 ArcEditor 和 ArcInfo 中使用 SQL Server Express 管理 ArcSDE geodatabases 的功能。

Workgroup ArcSDE included with ArcGIS Server for Workgroups: ArcGIS Server 的 Workgroups 级别包括支持 SQL Server Express。在这个级别上的 ArcSDE，可以在任何数量的用户和编辑者上使用 SQL Server Express，只要他们被配置所支持。

对于 personal ArcSDE，使用 ArcEditor 或者 ArcInfo 创建、执行和管理 workgroup ArcSDE geodatabases。可以使用 SQL Server Express 在 ArcCatalog 中建立和管理这些 workgroup ArcSDE geodatabases。不需要额外的数据库管理技术。

在这里，可以认为 ArcGIS Server 的 Workgroups 级别是 ArcEditor 或者 ArcInfo 的扩展，帮助管理和服务 workgroup ArcSDE geodatabases。

Enterprise ArcSDE included with ArcGIS Server for Enterprises:这是传统的 ArcSDE 技术，它运行在 Oracle, SQL Server, IBM DB2, 和 IBM Informix，可达到任何容量和任何数量用户的数据库，运行在任何大小和配置的计算机上。

ArcSDE 在 DBMS 的事务框架上提出长和短事务处理

ArcSDE geodatabase 包括对一个多用户 geodatabase 的编辑和更新的高级支持。随着 GIS 日益增加的用户和对来自一组传感器数据的管理的需要，GIS 对事务管理的需要变得更加重要。在 GIS 中，长事务的需求更多。

通常，GIS 用户具有特殊的事务处理需求，其中一个就是跨长时间的（不仅是几秒或者几分钟，有时几个小时，甚至几天和几个月）。

另外，GIS 中一个单独的编辑会话可能包括多个表中的多条记录的变化。用户需要 undo 和 redo 操作。用户在他们提交变化时，需要对待每个编辑会话为一个单独的事务。甚至，编辑通常必须是在一个与中央、共享数据库断开的环境中执行的。

在一个多用户的数据库中，GIS 事务必须与 DBMSs 的短事务框架协调。ArcSDE 通过管理高层次、复杂的 GIS 事务在这些操作中扮演一个关键角色。

ArcSDE 以 delta records 在数据库中存储这些变化的信息；使用版本分离多个编辑会话；支持复杂事务、自动历史数据归档和历史查询。

1、6 常见 geodatabase 的任务

创建和操作 geodatabase 的主要任务包括以下：

设计一个 geodatabase；

建立一个 geodatabase；

加载数据集和栅格到 geodatabase 中；

用高级数据类型扩展要素类；

用域、关联和子类型扩展表；

扩展栅格数据集；

加载和维护 geodatabase 中每个数据集的数据；

管理 geodatabase 的更新和事务；

管理 File 或者 Personal geodatabase；

管理 ArcSDE geodatabase。

第2章 Geodatabase 体系结构

2、1 Geodatabase 体系结构

用户通常认为 geodatabase 是地理信息的物理存储，但从根本来说，是使用 DBMS 或文件来存储的。除了作为数据集集合的一个物理实例，每个 geodatabase 还有其他关键方面特征。

1、 geodatabase 有一个广泛的信息模型来表达和管理地理信息。这个信息模型的实现是通过一系列简单数据表，这些表是存储在要素类、栅格数据集和属性表中的。另外，高级的 GIS 数据对象通过增加 GIS 行为、规则来管理空间完整性，以及增加工具来处理要素、栅格和属性表的大量的空间关系。

geodatabase 的 software logic 提供了通用的 application logic，贯穿整个 ArcGIS，用于访问和处理各种数据格式的地理数据。无疑，它是支持 geodatabase 的，并且支持 shapefiles, CAD 文件, TIN's, grids, CAD 数据, imagery, 还有其他大量的 GIS 数据源。

geodatabase 具有一个事务处理模型来管理 GIS 的数据工作流。

2、2 Geodatabase 基于关系原则存储

用户倾向于认为 DBMS 本质上是开放的，这是因为关系数据模型的简易性和灵活性，使得它能支持广泛的应用。

Geodatabase 的存储模型是以 DBMS 原则为基础，利用了一系列简单却基本的关系数据库概念。DBMS（File geodatabase 的文件系统）提供了简单而又合适的数据模型存储和操作表。

包括以下的关键概念：

数据被组织为表

表包含行

表中所有的行具有相同的列

每一列具有一个类型，如 integer, decimal number, character, date 等等。

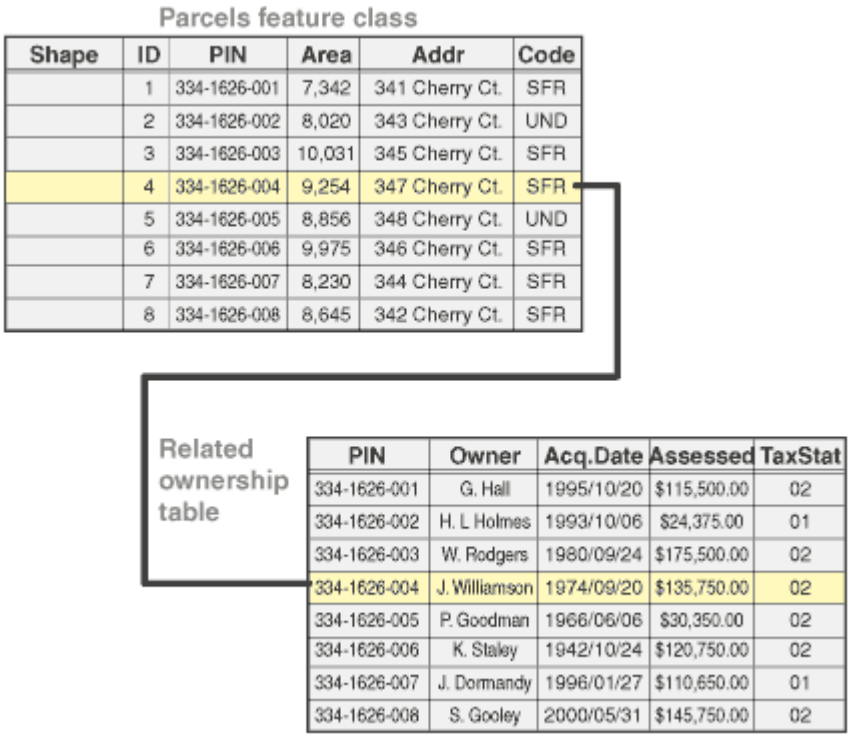
关系类用于一个表中的行和另一个表中的行发生关联，这是以每个表中都有一个公共的列为基础的。

关系完整性规则存在于所有表中。如，每一行总是有相同的列，一个域为每一列指定了有效值或者值范围等等。

对 ArcSDE geodatabases 来说，还有大量其他的 DBMS 功能可以应用：

SQL，一系列关系函数和操作符可以用于操作表和表中的元素。

SQL 操作符同时被设计为可以处理普通的关系数据类型，如 integers, decimal numbers, dates, 和 characters。



Parcels feature class

Shape	ID	PIN	Area	Addr	Code
	1	334-1626-001	7,342	341 Cherry Ct.	SFR
	2	334-1626-002	8,020	343 Cherry Ct.	UND
	3	334-1626-003	10,031	345 Cherry Ct.	SFR
	4	334-1626-004	9,254	347 Cherry Ct.	SFR
	5	334-1626-005	8,856	348 Cherry Ct.	UND
	6	334-1626-006	9,975	346 Cherry Ct.	SFR
	7	334-1626-007	8,230	344 Cherry Ct.	SFR
	8	334-1626-008	8,645	342 Cherry Ct.	SFR

Related ownership table

PIN	Owner	Acq.Date	Assessed	TaxStat
334-1626-001	G. Hall	1995/10/20	$115,500.00	02
334-1626-002	H. L Holmes	1993/10/06	$24,375.00	01
334-1626-003	W. Rodgers	1980/09/24	$175,500.00	02
334-1626-004	J. Williamson	1974/09/20	$135,750.00	02
334-1626-005	P. Goodman	1966/06/06	$30,350.00	02
334-1626-006	K. Staley	1942/10/24	$120,750.00	02
334-1626-007	J. Dormandy	1996/01/27	$110,650.00	01
334-1626-008	S. Gooley	2000/05/31	$145,750.00	02

例如，一个要素类以 DBMS 表的形式存储。每一行代表一个要素，每一行中列描述了该要素的各种特征或属性，表中的某一列存储了该要素的几何形状（如 point, line 或 polygon coordinates）。在上面的实例中，shape 字段存储了 polygon 形状。

在 DBMS 中大量的列类型用于 shape 字段。可以是典型的 binary large object (BLOB)类型或者是扩展的空间类型，它们能够被一些 DBMS 所支持。如，ESRI 提供了一个空间列类型用于在 ArcSDE geodatabases 存储要素，支持这种类型的关系数据库有 Oracle, IBM DB2, 和 Informix。

SQL 操作表中的行、列。这些列类型（numbers, characters, dates, BLOB's, spatial types 等等）在 SQL 代数中被认为是对象。DBMS 管理这些简单的数据类型和表，同时其他应用逻辑实现更复杂的对象行为和完整性约束。

在关系数据库管理系统中实现更高层次的对象和行为

开发人员为了实现更高层次的具有行为和逻辑的对象，就需要写应用程序代码去实现它。如，一个组织可能实现如下名为 EMPLOYEES 的表：

Last Name	First Name	Hire Date	Salary
Brown	Ben	10-10-2001	$10,000.50
Jones	Betty	06-14-1998	$22,000.00
Smith	Jason	08-23-1999	$44,000.75

上面的表是一个简单的关系数据表，包含行和列。每一列的数据都隶属一个独特的数据类型，如 character、date 和 number，DBMS 就是在这个数据类型层次上操作信息的。

然而，简单的给一个 DBMS 增加这些信息，不会使 DBMS 变成一个 payroll 或 employee 管理信息系统。增加一列名为“Dollars”用于存储带两位小数的数值，并不意味着将一个 DBMS 变为一个帐务清算系统。要实现这些需要更高层次的应用逻辑。

实现支持雇佣行为的逻辑例子是雇用、实现薪水增长、员工辞职、晋升和奖金管理。业务对象被建模为雇员和他们的姓名、薪水和雇用日期，为了实现这些对象的行为和完整性需要更复杂的和聚焦的应用逻辑。

相似的业务对象也普遍应用于 GIS。如，topologies, networks, linear referencing systems, raster catalogs, annotations, terrains, map layers 等等，都是高级对象用于实现 GIS 的行为。对于其他的 DBMS 应用，具有空间属性类型的表是对他们自己的 GIS 应用是不够的。简单的 DBMS 列类型和 geodatabase 应用对象如 topologies 对建立地理信息系统是必须的。

2、3 应用逻辑属于什么？

用户可以用很多方法实现这种高层次的逻辑。例如，应用逻辑可以按如下被实现：

DBMS 中的 Stored procedures 和 database triggers；

扩展 DBMS 的数据类型；

建立一个单独的应用逻辑层用于操作表中的行和列类型。

在过去的二十年中不计其数的 DBMS 实现已经证明使用应用逻辑层对于高级应用是一个明智的选择。如，通用的 customer information systems (CIS), enterprise resource planning (ERP) systems 和帐务清算包都是在应用逻辑层实现了高级的应用，它提供了更多的开放性和可扩展性，高性能操作，丰富的工具包和大大的灵活性。

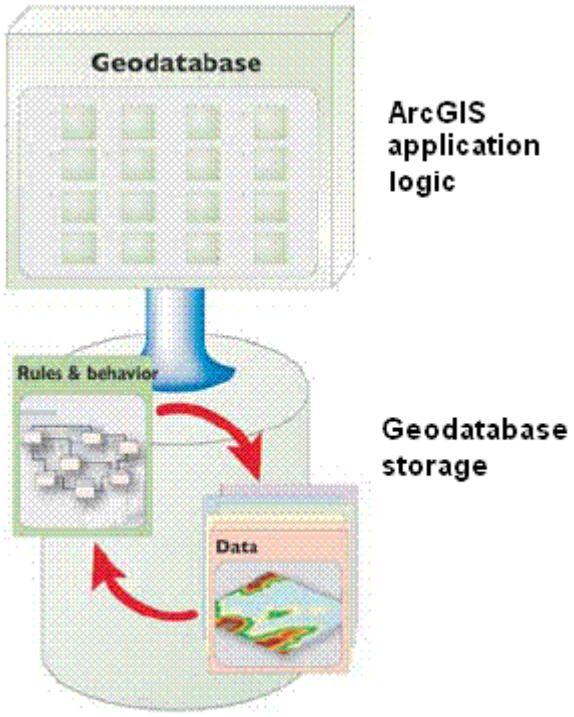
用户通过应用逻辑层与系统交互，执行版本管理，实现大量的操作，仅在 focused activities 中使用 SQL。

分离的应用逻辑层置于数据层之上允许相同的逻辑被应用于 DBMS，files 或其他的数据存储格式。这使得这种体系结构更开放。如，在 ArcGIS 中 geodatabase 的应用逻辑层也可以用于读取和操作所有的地理数据源 CAD data, shapefiles, MapInfo data, Intergraph GeoMedia files 等等。

2、4 Geodatabase 是 object-relational 的结构

Geodatabase 采用两层结构：数据存储层和应用层。数据存储层是将 GIS 数据存储为 File、XML、DBMS 等多种格式，而应用层则是维护数据的高级逻辑和行为，例如 Feature Classes、Raster Dataset、Topology、Network、Address Locators 等等。

多层的 geodatabase 体系结构有时被称为 object-relational 模型。



Geodatabase 管理地理数据集的功能由 ArcGIS 软件和 DBMS 共同实现。管理地理数据集功能的某些方面，如 disk-based 存储，属性类型的定义，查询处理及多用户版本处理都被

委托给了 DBMS。GIS 功能，如定义具体 DBMS 的 schema，用于表达各种各样的 datasets 和 domain－specific 逻辑，以保持基本记录的完整性和有效性。

实际上，DBMS 作为一种存储地理数据集的实现机制而被使用。但是，DBMS 并没有完全定义地理数据的语义。因此可以认为：geodatabase 采用两层体系结构，数据存储层和应用层，在数据存储层实现数据存储和检索，在应用层实现高层数据完整性和信息处理。

在其他高级 DBMS 应用中 geodatabase 也可以通过这种两层应用结构而实现。Geodatabase 对象在 DBMS 表中以行的形式存储，要素行为由 geodatabase 应用逻辑提供。

所有的 ArcGIS 应用与 geodatabase 的这种通用的 GIS 对象模型通信，而不是实际的基于 SQL 的 DBMS instance。Geodatabase 软件组件实现了这种通用模型的行为和完整性规则，并转换数据请求到合适的物理数据库设计。ArcSDE 软件提供了 ArcGIS 和 DBMS 之间的一个通用网关。

2、5 Geodatabase 在关系数据库存储

在 geodatabase 的核心是标准的关系数据库模式(一系列 DBMS 表、列类型、索引等等)。这种简易的物理存储与宿主在应用逻辑层的高层应用相协调，并受它们的控制，应用层可以是 ArcGIS Desktop、嵌入式 ArcGIS engine logic、或者 ArcGIS server。

这些应用中都包含用于定义通用 GIS 信息模型的 geodatabase 对象，并且为所有的 GIS 应用程序和用户所共享。Geodatabase 对象的目的就是暴露高层 GIS 应用模型给客户端，选择合适的存储模型存储应用模型的具体实现细节，如存储在标准的 DBMS 表中、file geodatabases, 和 XML 。

Geodatabase 的存储包括基于每个地理数据集的 schema 和 rule，还有简单的空间数据和属性数据的表存储。

Geodatabase 的 schema 包含定义、完整性规则和地理数据集的行为。这些具体包括要素的 properties、topologies, networks, raster catalogs, relationships, domains 等等。Schema 保存在 DBMS 的 geodatabase 元数据表集合中，这些元数据定义了地理信息的完整性和行为。

空间表达要么以矢量要素或栅格数据集的形式用传统的二维表来存储。例如，一个 DBMS 表可以存储一个要素类，表中每一行代表一个要素。每一行的 Shape 列都保存了 geometry 或者是要素的 shape，shape 列通常保存的 geometry 是这两种列类型中的一种：

BLOB column type

spatial column type，如果 DBMS 支持它

要素类是相同类型要素的集合，具有相同的几何类型，如点、线或多边形，具有相同的属性列集合，要素类通常可以由一个独立的表管理。

栅格和影像数据集同样也由关系表存储和管理。栅格数据通常比较大，存储的时候需要一个 Side Table 来存储，将栅格数据集切为小片，或者 Blocks，存储在 Raster 表中，通过 Side 表的每一行记录去记录栅格中的每一个 Block。

列类型存储矢量和栅格的 geometry 随着数据库的不同而不同。当一个 DBMS 支持空间类型扩展时，geodatabase 可以使用这些类型来存储空间几何形状。ESRI 也参与了很多扩展空间 SQL，如作为 SQL 3 MM Spatial 和 OGC Simple Features SQL specifications 的主要创作者。ESRI 不仅关注支持独立的 Oracle Spatial types，同时也支持这些类型，使用 DBMS 的标准来存储 geodatabase。

目前，支持 geodatabase 的含有空间类型的 DBMS 有三种：

使用 ESRI 的 spatial type 的 Oracle 或 Oracle Spatial type

使用 Spatial Extender Geometry Object 的 IBM DB2

使用 Spatial DataBlade Geometry Object 的 Informix。

2、6 Geodatabase 的事务管理

事务是对数据库变动的记录块。GIS 数据库，像其它数据库应用一样，必须支持执行数据完整性和应用行为的更新事务。在很多情况下，用户可以使用 DBMS 的事务处理框架来管理数据编辑和 geodatabase 的更新。

但是，GIS 用户普遍有具体的专业事务处理要求，其中最重要的就是这些事务要跨越很长的时间（有时是几天和几个月）。

另外，大多数 GIS 编辑包括协调（orchestrating）多个表中的多行变化和以单独的统一的事务管理这些变化，用户需要能够撤销或重做这些变化。编辑过程可以跨越几个小时甚至几天。通常编辑过程必须是在与中央共享的数据库断开的状态下进行操作的。

由于 GIS 工作流过程可以跨越几天或几个月，GIS 数据库必须为日常的操作保持持续地可用，每个用户可能有共享 GIS 数据库的个人视图或状态。在一个多用户的数据库中，GIS 事务必须由 DBMS 的短事务处理框架管理。ArcSDE 在简单 DBMS 事务处理框架管理高层次、复杂的 GIS 事务的操作中，扮演着关键的角色。

在很多情况下，GIS 用户选择长事务工作流是必须的，这样才有可能多用户的 DBMS 的使用和用 ArcSDE 来管理中央数据库的更新。下面是一些需要使用长事务处理模型的 GIS 数据编辑工作流的例子：

Multiple edit sessions－一个单独的 GIS 数据库更新需要大量的变动，这些变动跨越多个编辑过程，可能是几天或几个星期。

Multi-user editing－多个用户通常需要同时编辑更新相同的空间要素。每个用户都需要操作私人化的数据库状态，只关注个人的更新情况而忽略其他用户的更新。最后，各个用户提交各自的更新并与其他用户协调更新以确认更新，解决所有冲突。

Checkout/check-in transactions－给特定区域或地区的某个计算机签出（check out）数据库的一部分通常是必要的，它在与数据库断开的状态下进行信息的更新操作，这种情况可能持续几天甚至几个星期的， 这些更新最后必须提交到中央数据库。

History－有时候保存一个 GIS 数据库每一个要素的历史版本是很有利的，甚至是在一个特定版本已经被更新后，保存 retired 和 changed 要素的复本在一个历史文档数据中或追踪一个独立要素的历史，例如，在一个国家地图数据库中 parcel lineage 的更新。

Transfer of change-only updates－企业级的数据库和空间数据基础设施的信息被组织内或跨组织共享的，它们合作实现更新操作通过互联网以定义明确的 XML 达到共享。

Distributed geographic database replicas－一个区域数据库可能是某一特定地理区域的中央 GIS 数据库的部分拷贝。定期地，两个数据库必须通过交换更新以达到同步。

Loosely coupled replication across DBMS's－通常地，GIS 数据必须在一系列数据库复本（replicas）中保持同步，每一个复本操作其本地的数据库进行更新。这些数据库只有定期地通过互联网连接。更新按照预定的原则从一个数据库复本转移到其他的数据库复本，保持它们内容的同步。大多数情况下，DBMS 是不同的，例如，可以在 SQL Server, Oracle, 和 IBM DB2 进行复制数据集。

Geodatabase 的事务模型－－版本

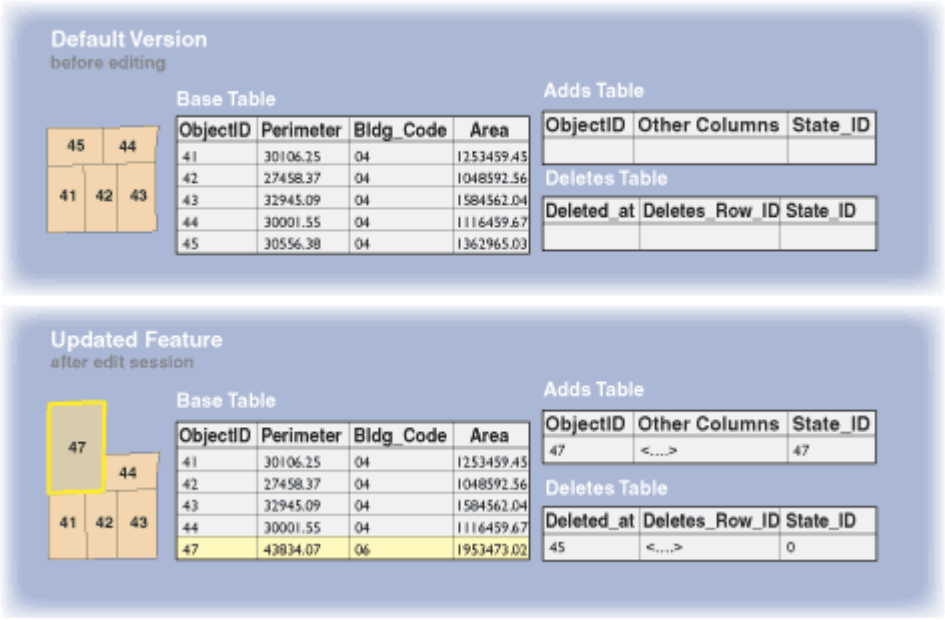
Geodatabase 管理一些 GIS 工作流的机制就是保持多个状态，并且保证地理信息、规则和行为的完整性。从字面意思来理解，版本就是记录独立要素或对象的修改、增加和删除的各种状态。每一个版本就是明确记录了每一个要素或对象的行状态，同时还包括一些重要的事务处理信息。任何数量的用户可以同时操作和管理多个版本。

版本可以使所有的事务处理随着时间以数据库的一系列变化而记录。这意味着各个用户都可以操作 geodatabase 的多个视图或状态。它的目标就是开放的、高性能的多用户访问。例如，系统必须高速运行，并且必须支持含有成千上万的记录的数据集在数千个用户并发访问时的使用。

Geodatabase 基于版本的事务模型相对简单－－更新被记录在变化的表中。

版本在两个 delta tables 明确地记录了 geodatabase 对象状态：Adds table 和 Deletes table。简单的查询用于对任何想要的 geodatabase 状态进行视图；例如，实时对一个点的数据库状态进行视图或特定用户的当前的编辑版本。

ArcSDE 在版本化的 geodatabase 应用中扮演着关键的角色，用于提供对同一或不同的 DBMS 的长事务管理。



上面是一个例子，一个 parcel（number 45）被更新为 parcel（number 47）。利用版本，原来的 parcel 被保存在 Deletes table 中，新的 parcel 被保存在 Adds table 中。其他的版本信息被记录到 meta tables 中，如时间和更新序列、版本名称、每一个更新的 state ID。每一个版本都有其安全措施和访问优先权。

大量的更新会不利于每个版本的，当用户编辑数据时，他们需要连接操作更新的版本。当用户准备与其他的用户共享更新时，就需要执行冲突协调和提交编辑操作，保存在更新版本中的编辑将被提交到默认（Default）的版本，在对冲突的解决方案中确认和协调那些潜在的冲突。

2、7 Geodatabase XML

Geodatabase XML

Geodatabase XML 代表了 ESRI 的 geodatabase 和外部系统之间的开放的信息交换机制。ESRI 以 XML specification 公开了全部的 geodatabase 的 schema 和 content，并且提供了实现例子解释用户如何在不同类型的系统中共享数据更新。

地理信息的 XML 交换很简单地使用了 geodatabase XML specification。外部应用程序可以接受的 XML data streams 包括以下几种：

交换完全无损数据集；

交换简单要素集（如 shapefile 的交换）；

通过使用 XML streams 在 geodatabase 和外部的数据结构之间传递更新和变化，实现 change－only 记录的交换；

Geodatabase XML 不仅是所有 ArcGIS 用户，也是外部用户共享数据的最基本的交换机制。

在 ArcGIS 中可以创建三种类型的 XML documents：a Workspace document, a RecordSet

document, 和 a Data Changes document。

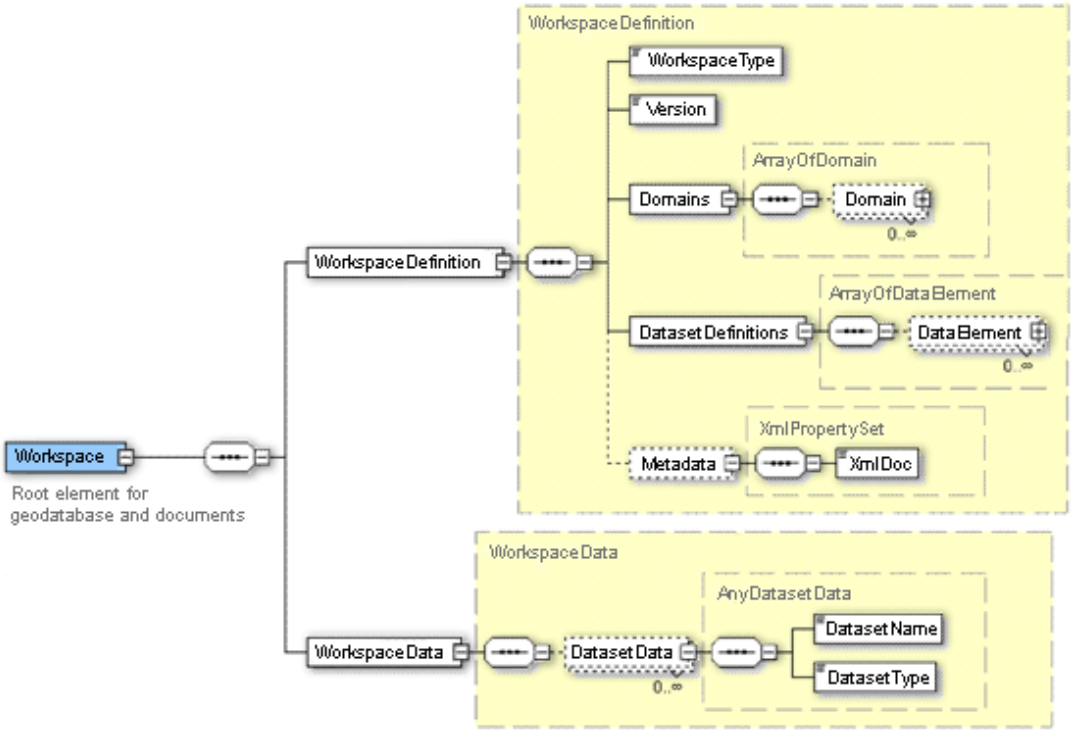
Geodatabase 的 XML workspace document

XML workspace document 保存了一个 geodatabase 中所有内容或其子集。XML workspace document 包含了所有的 schema 信息，数据是随意的。

XML workspace document 用于共享 schema 信息，交换数据集和它们的所有内容，交换数据集和与它们相关的所有数据元素。

Schema、relationship 和 behavior 信息作为 WorkispaceDefinition section 的部分导出，包括简单的用户要素数据、网络和拓扑、网络连接性和拓扑规则、简单的复合关系类和其他任何与地理数据集相关联的信息。因此，所有与 geodatabase 相关联的行为被存储，当 XML documents 被导入时被再创建。

XML workspace document 包含两个元素：WorkspaceDefinition 和 WorkspaceData。



使用 WorkspaceDefinition 和 WorkspaceData，XML workspace document 可以包含 geodatabase 的 schema。与实际数据分开的定义提供了一些好处。一个软件代理仅能读取 workspace document 的定义部分了解它所包含的，使数据部分可选，workspace document 仅能用于转换 schema 信息。

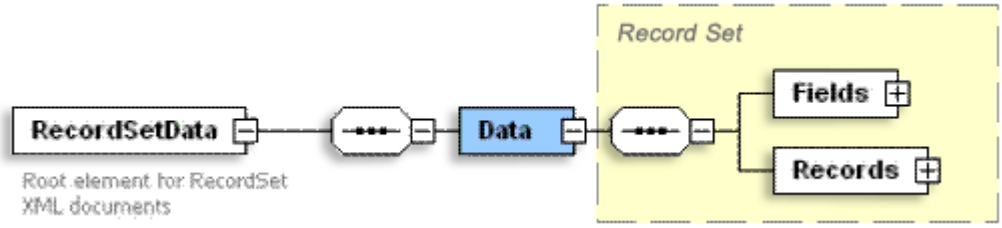
对表和要素类的来说，在定义部分中数据元素的名称和数据部分中的表数据名称是一致对应的。其他数据集，如 topologies, geometric networks, 或 要素数据集，仅在定义部分中出现（它们不能导出数据）。

Geodatabase 的 RecordSet document

Geodatabase 的 RecordSet document 用于从一个独立要素类或表中以简单要素或表记录导出为行集合。

导出到一个 RecordSet document 相当于导出到一个 shapefile 文件。行集合以记录被导出，并且没有附加的 geodatabase-related 信息被写到输出文件。

例如，topologies 和其他要素数据集信息将不会被导出，关系类也不会被导出。



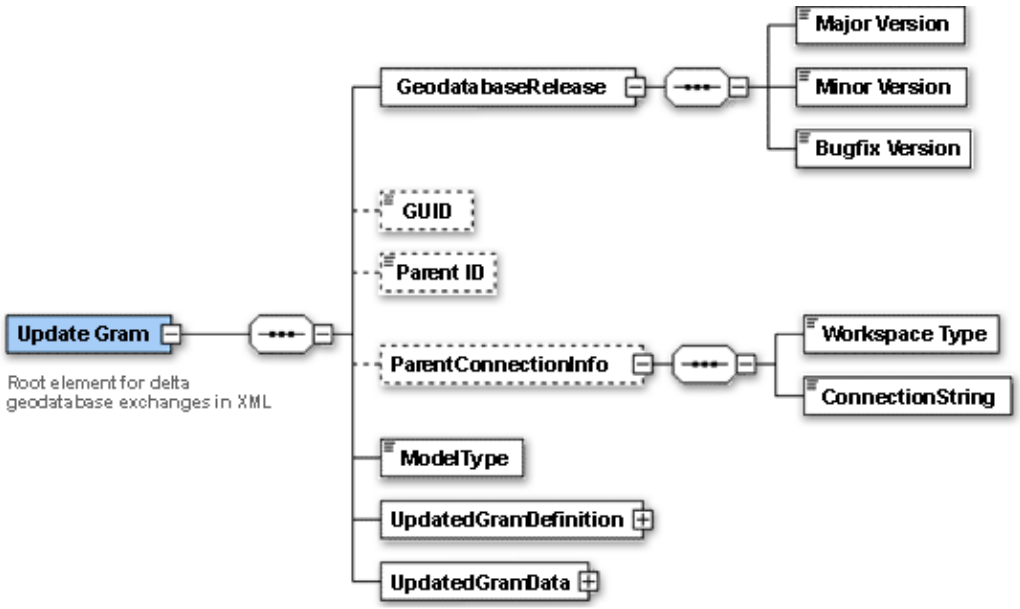
Geodatabase 的 XML Data Changes document

作为 geodatabase 事务和版本管理的一部分，用户需要共享和操作 change－only 记录集。XML Data Changes document 用于共享 geodatabase 之间、geodatabase 和外部系统之间的变化和更新。要做到如此仅能通过分发变化，可以通过 geodatabase 的版本追踪这些变化。

ArcGIS 的离线编辑框架允许将数据库中的数据签出到一个单独的 geodatabase 中，然后在与父数据库断开连接的状态下对数据进行编辑。一旦编辑完成，就可以导出变化（并不是所有的数据）到 XML file。这个 XML file 可以随后用于签入变化到父数据库中。

这也是 geodatabase replication 关键的一部分。

XML Data Changes document 的基础信息是 UpdateGram，在 UpdateGram 中只有变化可以作为 XML document 被传递。



在 ArcCatalog 和 Geoprocessing 中使用 geodatabase XML

在 ArcGIS 中，ArcCatalog 和 Geoprocessing 框架包含大量的功能，用于导入和导出地理数据。在 ArcGIS 中，用户可以用 Geodatabase XML 执行的操作包含以下几种：

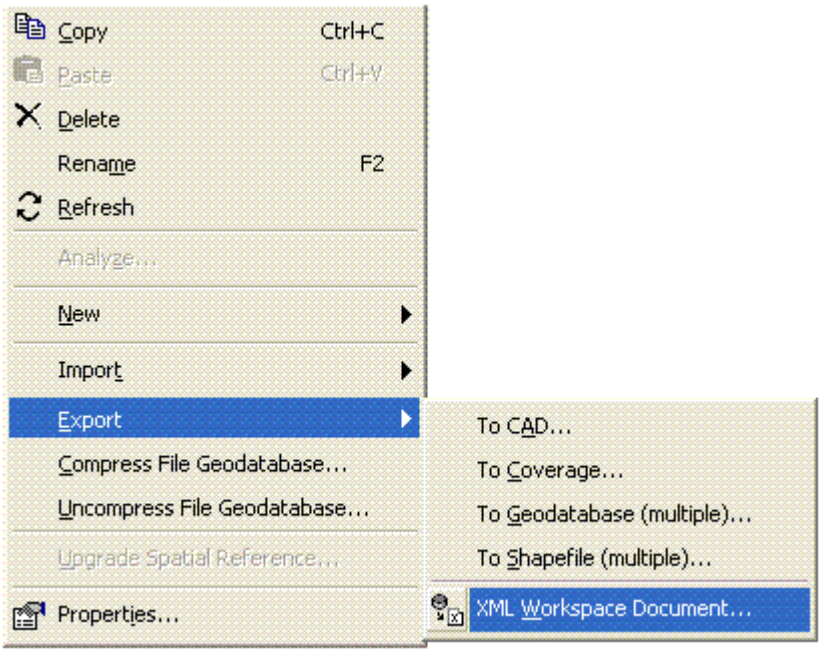
共享 geodatabase 的 schemas；

复制整个 geodatabase 或其子集，如要素数据集和所有相关的信息集合；

复制独立的要素类，表或栅格数据；

跨多个 geodatabase 复本同步其内容。

例如，使用 ArcCatalog，通过在一个 geodatabase 或要素类上单击右键，创建一个 XML Workspace Document，这个 Document 可以用于共享一个 geodatabase 的 schema 或者复制其内容、规则和行为到另外一个 geodatabase。



在 ArcGIS 中，使用 geoprocessing，也可以执行一些操作实现跨 geodatabase 同步和共享变化。

第 3 章 设计 Geodatabase

3、1 geodatabase 设计概要

geodatabase 设计概要

Geodatabase 设计是以基本的 GIS 设计步骤为基础。因此，了解 GIS 设计的目标和方法是很重要的。

GIS 设计包括将地理信息组织为一系列专题数据－层（layers）。因此，geodatabase 设计以确定使用哪些数据专题为开始，然后指定每个专题层的内容和表达方式。具体包括如下的定义：

地理要素如何表达为每一个专题层（如点、线、多边形或栅格数据集），和它们的属性表；

数据如何被组织为数据集如要素类、属性表、栅格数据集等等；

为了完整性规则、实现丰富的 GIS 行为（如 topologies, networks, 和 raster catalogs）、定义数据集之间空间和属性的关系，哪些其他的空间和数据库元素是必需的。

Representation

每个 GIS 数据库设计都是以决定为每个数据集采取什么地理表达为基础的。单独的地理实体可以表达为以下几种形式：

要素类：（Feature classes ）；

影像和栅格：（Imagery and rasters）；

连续表面：（Continuous surfaces ）它可以使用要素（轮廓），栅格数据集（数字高程模型 DEM），或者不规则三角网（TINs）来表达；

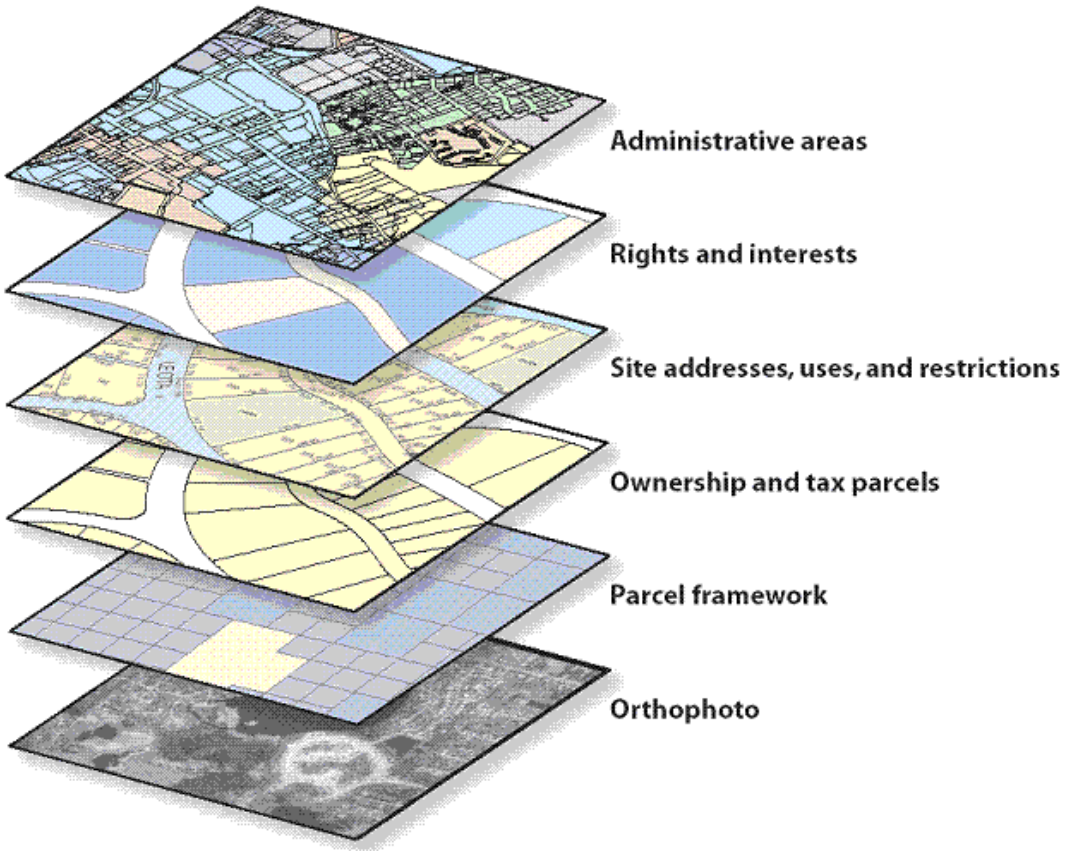
用于描述数据的属性表。

Data themes

地理现象的表达被组织为一系列的数据专题（有时称为专题图层）。在 GIS 中一个关键的概念就是某一个数据图层或专题。一个数据专题就是相同地理要素的集合如一个道路网路，地块边界的集合，土地类型、某一特定时间的卫星遥感影像，井位置等等。

专题图层的概念是早期 GIS 概念之一。早期的从业人员考虑如何将地图上的地理信息区分为逻辑信息层－－单个对象的随机集合（如一条道路、一座桥、一座山、一座房子、一个半岛）。这些早期的 GIS 工作者以专题图层的形式组织地理信息，这些图层描述了某一现象如何分布，以及在这一地理范围中如何描绘它。这些图层也提供了一个协议组织地理表达（如要素集、栅格图层、属性表等等）。

在 GIS 中，专题图层是 GIS 数据库设计主要数据组织原则之一。



每个 GIS 对某一地理区域都包含多个专题层。每一个专题可以被管理为与其他专题独立的信息集。每一个都有它们自己的表达方式（点、线、多边形、表面、栅格等等）。由于各种独立的专题在空间上被作为参照，它们可以叠加到另一个专题上，并可以在地图中合成显示。另外，GIS 分析操作，如叠加可以在不同专题间融合信息。

GIS 数据集是一个数据专题的表达集

在一个 GIS 数据库中，地理数据集可以以要素类或基于栅格的数据集来表达。

很多专题以相同要素的单独集合来表示，如土地类型多边形要素，井点要素集合。其他专题，如一个运输组织被表达为多个数据集（如街道、十字路口、桥、道路岔口等空间上相关的要素类集合）。

栅格数据用于表达连续的表面，如高程、斜坡和表面，同时也用卫星影像、航空照片和其他的栅格数据（如土地覆盖和蔬菜种类）。

欲打算使用方式和现存的数据源都影响 GIS 中的数据表达。当设计一个 GIS 数据库时，用户会在头脑中形成很多的应用集合。定义这些应用可以帮助决定每一个专题的内容描述以及如何以地理的形式表达它们。例如，表达地面高程有很多的选择：以轮廓线和位置高度值

（如山顶高度）、以连续的地形表面（TIN）或者阴影地貌，每一个表达方法都与一个特定的 GIS 数据库设计相关联的。欲将数据做何种使用会帮助决定那一种表达方式是需要的。

经常地，地理数据的表达将在一定程度上被可以利用的数据源预先决定。如果欲先存在的数据源以特定的比例尺和表达存储，使用这些数据适应设计是必要的。

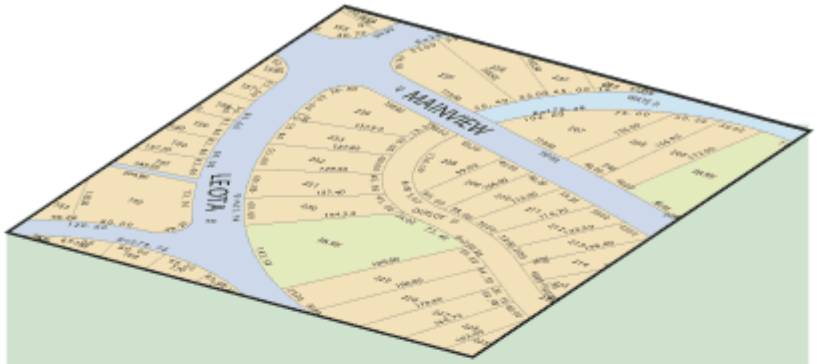
3、2 geodatabase 设计步骤

Geodatabase 设计以专题图层开始。

首先，为了特定的应用和信息需求，确定所需要的专题图层。然后，详细定义每一个专题图层。每一个专题图层的描述就可以导出标准 geodatabase 数据元素的描述，如要素类, tables, relationship classes, raster datasets, subtypes, topologies, domains 等等。

在设计时确定专题图层，应尽力描述每个专题的特征，依据它的实际表现、在 GIS 种它的预期使用、它可能的数据源和精度水平。例如，需要在多大比例尺和范围使用这些信息，在每个比例尺下如何表达这些元素？这些特征帮助描述预期从每个专题中获得的内容。

下面是一个在 cadastral application 中对 ownership parcels 的描述。



Layer	**Ownership parcels**
Map use	Parcels define land ownership and are used for taxation
Data source	Compiled from land ownership transactions at local government
Representation	Polygons in survey-aware feature classes and related annotation
Spatial relationships	Parcel Polygons cannot overlap and are covered by boundary lines
Map scale and accuracy	Typical map scales are 1:1200 and 1:2400
Symbology and annotation	Parcels often drawn using boundary features and related annotation

一旦已经确定了关键的专题图层，接下来就要在物理数据库基础上描述每个专题的内容。

列出地图比例尺和范围；

对每个专题确定如何表达这些地理要素（以点、线、多边形、栅格和属性表等）；

将数据如何组织为要素类、表和关系类？

如何使用空间的和数据库的完整性规则实现 GIS 行为？

下面给出的 11 个步骤大致概述了一个普通 GIS 数据库设计的过程。1 到 3 步的初始设计帮助确定和描绘每一个专题层，4 到 7 步，开始采用详细的描述方式、关系，基本上形成 geodatabase 元素和它们的属性。8 到 9 步，可以定义 data capture procedures 和分派 data

collection responsibilities。在最后几步（10 到 11 步），通过一系列初始化执行测试，进而完善设计。同时，需要将设计文档。

Geodatabase 设计的 11 个步骤：

确定在 GIS 中将创建和管理的信息产品。

GIS 数据库设计应该反映所在组织的工作。考虑编辑和保存地图产品的目录、分析模型、web 地图应用、数据流、数据库报表、关键职责、3D 视图和其他基于责任的需求。列出在工作中当前使用的数据源。利用这些资源开始数据设计。

基于信息需求确定关键的数据专题。

比较全面地定义每个数据专题的一些关键方面。决定如何使用每个数据集－－为编辑、GIS 建模还是分析，表达业务流、制图还是 3D 显示。具体描述地图的用途，数据源，各个比例尺的空间表达；二维视图和三维视图的数据精确性和指导方针；专题如何显示，考虑其 symbology, text labels, 和 annotation。考虑每个地图图层如何与其他图层以集成的样式显示。对于建模和分析，考虑信息如何与其他数据集一起使用（如它们如何合并和集成）。这会帮助确定一些关键的空间关系和数据完整性规则。应该保证这些显示和分析内容在数据库设计中是不可缺少的一部分。

3、详细说明每个比例的数据专题的比例范围和空间表达。

数据编辑是在一个具体的地图比例尺范围内的，应该将地图比例尺与地理表达相联系起来。地理数据表达会经常在不同地理比例尺之间变化（如从多边形到线或点）。在很多情况下，需要在一个相对小的比例尺下生成要素表达。栅格数据使用影像金字塔重建模。

4、将每个地理表达分解为一个和多个地理数据集。

离散的要素通常被建模为要素点、线和多边形的要素类。可以考虑用高级数据类型，如 topologies, networks, 和 terrains，来建立同一专题层不同元素之间以及跨数据集之间的关系类。

对栅格数据集来说，mosaics 和 catalog collections 是用于管理大数据集合的选择方式。Surface 可以用要素建模，如轮廓，也可以用栅格和表面。

5、为属性表定义表结构和行为。

确定属性字段和列类型。

表同样可能包括属性域、关联和子类型。定义有效值、属性范围和分类。使用子类型控制行为。确定表关联和关联类间的联接。

6、为数据集定义空间行为和完整性规则。

可以使用 topologies, address locators, networks, terrains 为要素增加行为和功能。例如，使用 topologies 可以模仿共享几何图形的空间关联，强化完整性规则。使用 address locators 可以支持地理编码。对栅格数据，可以根据需要决定选择 raster dataset 还是 raster catalog。

7、提出一个 geodatabase 设计。

在设计中为每个数据专题定义所需要的 geodatabase 元素集合。研究现存的设计，开始自己的设计工作。

8、设计编辑工作流程和地图显示属性。

定义编辑流程和完整性规则（如，所有的街道在交叉处被切开，街道段在端点处连接）。设计编辑工作流程可以帮助实现完整性规则。

定义地图和 3D 视图的显示属性，这可以用于定义地图图层。

9、为创建和维护数据层分派职责。

决定分派谁来做数据的维护工作，是由本组织还是其他组织的人员来做。理解这些角色是重要的。需要设计数据如何变换和转换，用于和你的伙伴组织进行数据导入和导出工作。

10、创建一个工作原型不断完善设计。

测试原型设计。使用一个 file, personal, 或者 ArcSDE Personal geodatabase 给已提出的设计创建一个实例 geodatabase。通过创建地图，运行关键的应用程序，进行编辑操作来测试设计的功能。在原型测试结果的基础上，修改并完善设计。一旦有了一个作业方案，加载数据集合（如加到一个 ArcSDE geodatabase 中）来验证产品、性能、伸缩性和数据管理工作流。

这是很重要的一步。在开始装载 geodatabase 前先决定如何设计它。

11、将 geodatabase 设计文档化。

各种方法可以用于描述数据设计和结果。如使用 drawings, map layer examples, schema diagrams, simple reports, 和 metadata documents。

有的用户喜欢使用 UML，但是，UML 本身功能有限。UML 不能表达所有的地理属性，同时 UML 也不能传递 GIS 设计的关键概念，如专题组织、拓扑规则、网络连接，UML 在设计中不能提供空间视图。

很多用户喜欢用 Visio 来创建 geodatabse 模式的地理表达，如那些发布的 ESRI 数据模型图。ESRI 提供了工具来帮助使用 Visio 获得数据模型元素的这些图形。

3、3 使用 ArcGIS 数据模型设计

ESRI 和它的用户已经花费了大量的时间开发一系列的 geodatabase 数据模型的模板，这些模板提供了一个快速开始的 geodatabase 设计。这些设计已经被详细描述和文档化，在 http://support.esri.com/datamodels 可以看到。

在这个网站，可以发现已存的 geodatabase 模板，还有一些有用的文档，这都是关于在很多行业和应用中的 geodatabase 设计。这些模型都是很好的开端。很多用户都是使用这些设计模板开始的，然后不断完善和扩展自己的设计，以达到自己具体的需要和要求。

一旦找到了相关的数据模型，可以从这个网站上下载一个 geodatabase 模板，用于快速开始的设计。可以建立一个测试的 geodatabase，加载一些数据到其中，然后测试和完善设计。

使用 ArcGIS 数据模型步骤作为设计的基础的步骤

使用 ArcGIS 数据模型包含的步骤与如何导入和修改一个现存的 geodatabase 设计是非常相似的。

从 ESRI Support Center http://support.esri.com/datamodels 下载合适的数据模型。

创建一个空的测试的 file geodatabase。

导入 schema，建立合适的空间参照。

加载一些现存的数据集到空的新 geodatabase 中。

测试并完善设计。

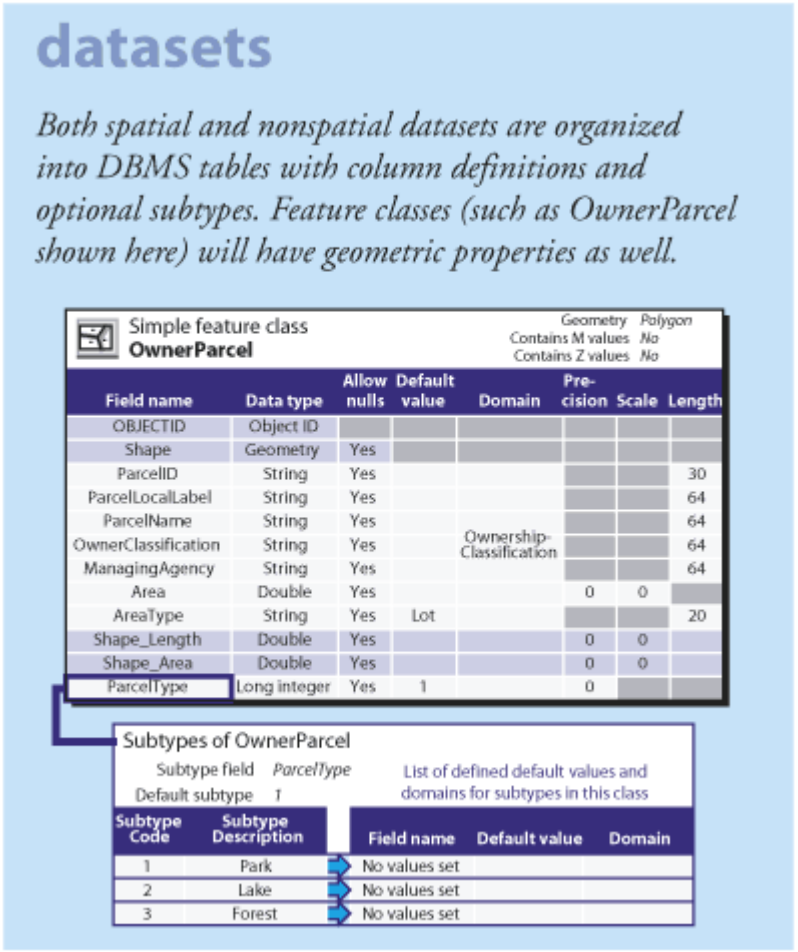
3、4 文档化 geodatabase 设计

文档化 geodatabae 设计

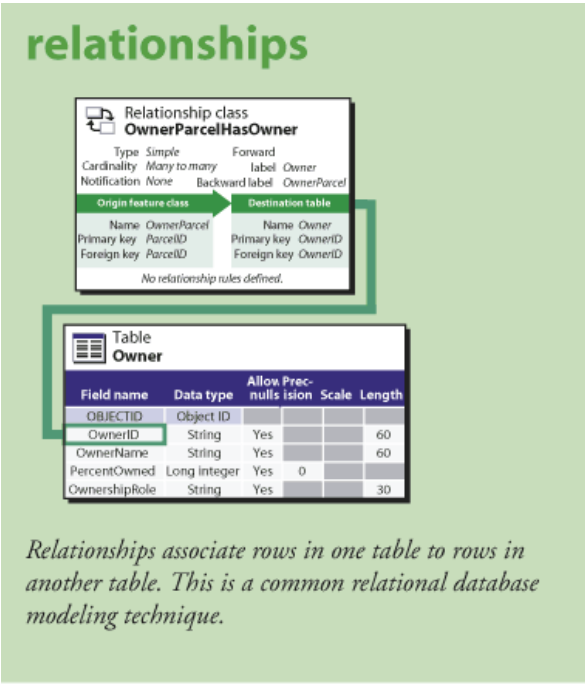
Geodatabase 设计文档化是很重要的。在 ArcGIS 数据模型网站（http://support.esri.com/datamodels），一系列模型图用来表达关键的设计概念，文档化每个数据模型模板的 geodatabase 元素规格、元数据和地图图层。这部分简单地概括了网站上人如何表达各种各样的 geodatabase 元素，这对自己的设计文档或许是有帮助的。

在 geodatabase 设计中有 5 个关键的元素来表达设计的内容，具体包括：

1、**datasets**—它们是如何记录要素类，栅格和属性表的属性的规范，还有每个表的列集合的规范。对于空间表达可以看到一些几何属性（如点、线和多边形以及坐标系类型）。通常，还可以看到对子类型的说明。这部分的 schema diagram 常用蓝色显示。

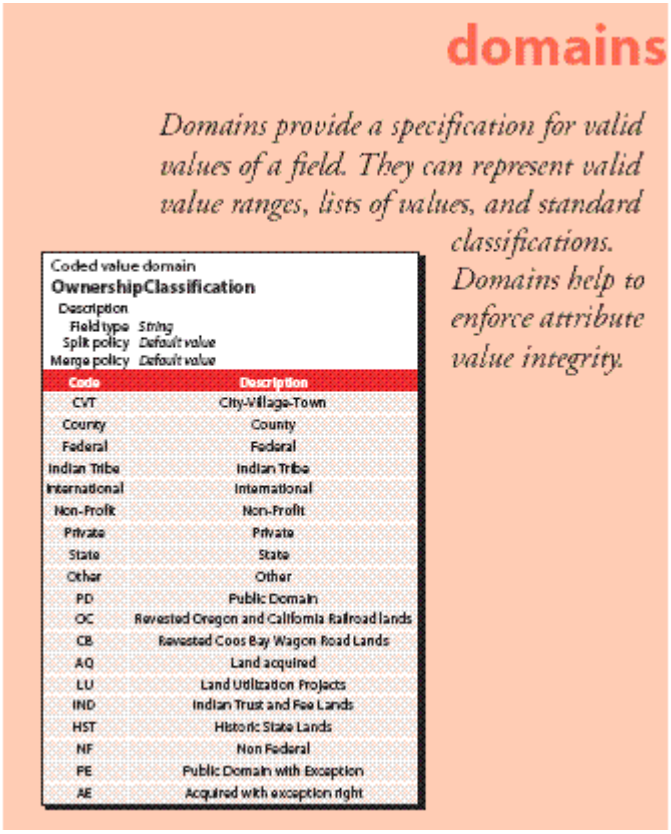


2、**Relationship classes—**属性关联广泛地应用于 GIS，正如它们在 DBMS 中的应用。它们定义了一个表中的行如何与另外一个表中的行相关联。关联是具有方向和其他一些属性（如，这是一个一对一，一对多还是多对多的关联）关联和它们的属性用绿色显示。



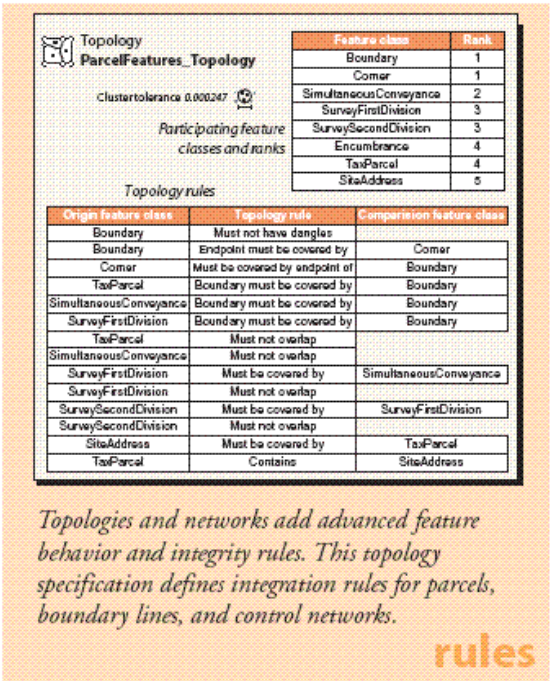
3、**Domains—**它们表达了属性列的有效值范围。这些规则控制系统如何维持某些属性

列的数据完整性。域用红色显示。



4、Spatial Rules—大量可以用于 geodatabase 的高级数据建模功能。例如，数据元素，如拓扑和它们的属性用于模拟要素如何与其他要素共享几何图形。Topologies，及 network datasets, address locators, terrains, cartographic representations, geometric networks，还有许多其他的高级 geodatabase 类型，提供了关键和广泛地使用 GIS 的机制，在 GIS 数据库中实现空间行为，强化完整性。这些规则如 networks 用橙色显示。

理解 geodatabase 中如何文档化和描述这一套扩展的数据类型最好的方法就是描述它们的规则和行为。下面是一个拓扑如何被文档化的例子。



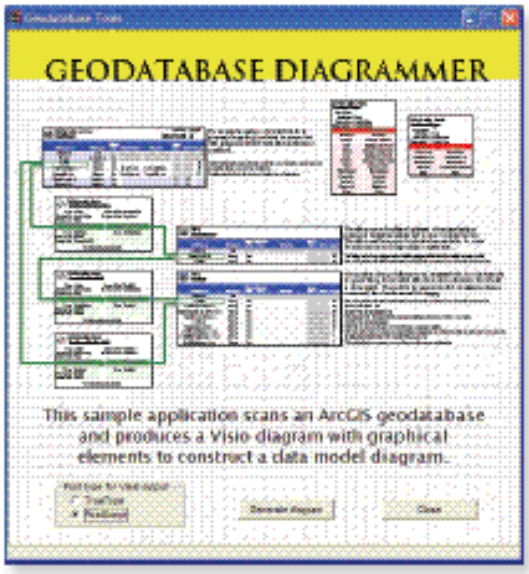
5、**Map Layers—**GIS 包括交互式地图和其他的视图。每个数据集的关键部分就是对如何符号化和排列地图显示的详细规范。在 ArcMap 中这些被定义为层属性，它详细描述了如何给要素选取地图符号（颜色、填充样式、线和点符号）和文字注记。数据图层并不在 geodatbase 中管理，但在 geodatabse 模式中帮助定义关键数据集的属性是很重要的。数据图层描述一般用黄色显示。数据图层可以存储为.lyr 文件或者作为 ArcMap 文档（.mxd）的元素。

使用 Microsoft 的 Visio 和 Geodatabase Diagrammer 工具

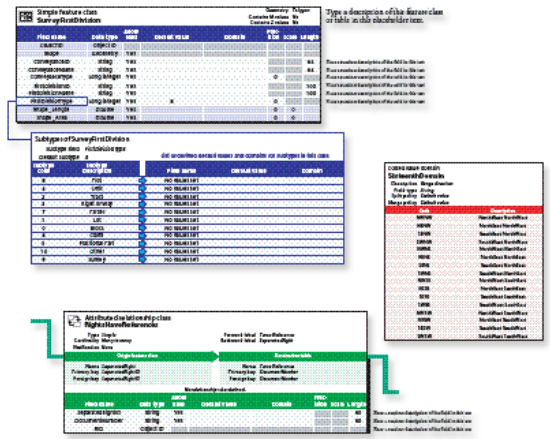
ESRI 提供了一个图表工具，想生成与这些相似图表的用户可以下载。也可以下载一个工具，Geodatabase Diagrammer，它可以生成数据集和 geodatabase 元素的一系列 Visio 图，在 http://arcscripts.esri.com 上可以搜索 Geodatabase Diagrammer。

这个工具可以用于在 Visio 中创建 geodatabase 内容的图表元素。可以容易地 cut 和 paste 图表从 Visio 到 Word、PowerPoint 和任何其他应用程序，它可以读取.wmf 文件。

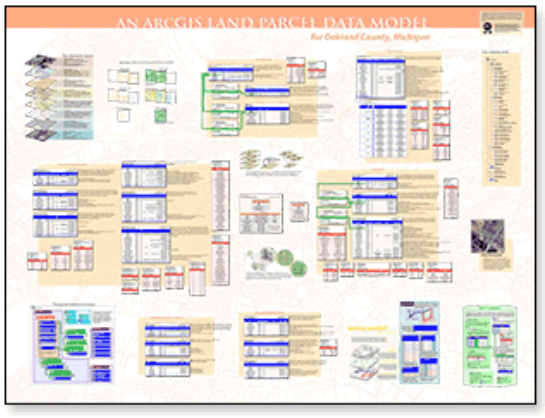
文档化 geodatabase 设计的其他属性。



In ArcCatalog, select a geodatabase and start the Geodatabase Diagrammer command.



Geodatabase Diagrammer launches Microsoft Visio and interrorgates the structure of the geodatabase. Schema graphics are generated that you can insert in your own documentation.



You will derive great benefit from diagramming your data model—the active participation of stakeholders in examining your data model poster, the perspective you will gain on data model patterns, and the assurance that your schema is correct. The act of diagramming drives changes that will improve your data model.

文档化 geodatabase 设计的其他属性

其他关键的 geodatabase 设计的属性应该考虑和文档化包括：

每个数据集的坐标系统和空间属性的定义；这包括 map projection, coordinate system, spheroid, datum, x,y units, vertical coordinate system，z 值和 m 值的使用。

每个数据集的 tolerances 和 coordinate resolution；

数据源和数据编辑工作流；这包括 translation scripts, geoprocessing 和 transformation models，还有用于创建和维护数据集的工作流。

每个数据集的元数据文档。

3、5 要素类建模

以下是 geodatabase 的要素类建模的设计技巧：

Task 1.设计简单的要素类

几乎没有例外，任何 geodatabase 都包含要素类。或许仅需要只包含简单要素类集合的 geodatabae 设计。但是，大多数用户需要设计一个更为复杂的数据模型，这个模型添加了高级的 geodatabase 元素。可以根据系统的目标和实际需要来决定是否扩展简单要素类的设计；可以扩展设计来支持一般的 GIS 功能和行为。这一部分就是介绍许多要素类的功能

从定义一个简单要素类的普通属性开始。可以稍后添加需要的，但是在第一步还是集中在基础设计方面。

一个要素类是具有相同几何类型（如点、线或多边形），相同属性，相同坐标系统的地理要素的集合。

ArcGIS 中的要素类例子：

Feature Class	Representation	Notes
Street centerlines	Line	Street segments split at each intersection; usually contain address ranges and network properties
Wells	Point	
Soil types	Polygon	Usually have many descriptive attributes in related tables
Parcels	Polygon	Topologically integrated with parcel boundaries and corners
Parcel boundaries	Line	Has coordinate geometry and dimension attributes; participates in a topology with parcels and corners
Parcel corners	Point	Surveyed corners of parcels; participates in a topology with parcel polygons and boundaries
Parcel annotation	Annotation	Provides text labels for lot dimensions, taxation, and legal description information
Building footprints	Polygon	Contains outlines of buildings and structures

一旦决定了要素类的列表，就要开始为每个要素类定义以下的内容：

挑选一个几何类型（或称要素类型）如点、线、多边形或注记。需要为每个要素类中的所有要素定义相同几何类型。

决定属性字段和列类型。

决定几何属性。是否含有 z 值？M 值？用何种 coordinate resolution？对于线和多边形要素类使用何种 line segments？通常情况下，用户只需要默认值就可以，即简单的 straight-line segments。但是，有时需要 curved segments，例如，表达 cul-de-sacs 和 roads 时。

定义要素类的坐标系统

是否需要该数据集在多级比例尺下使用？在每个地图比例尺下这些地理表达是如何变化的？你会发现在另一个比例尺范围下需要改变要素类表达方式。因此，在这种情况下，你需要考虑每个比例尺范围表达同一数据专题的其他要素类。

有时，你直接将数据加载到 GIS 中。在这种情况下，你不需要按照下面的设计任务去做。但是，评价给你的 geodatabase 中要素添加一些 GIS 功能的优点，这是重要的。从长远来看，这些附加的功能可以潜在地使你的数据应用和维护变得容易多。它们将帮助你维护空间信息的完整性；帮助你以多种方式利用数据；最重要的是，帮助你认识你的数据对满足你的需要信任度。

扩展简单要素数据模型的理由：

在导入数据集前需要验证其有效性（为了保证数据集遵守一系列的空间完整性规则）；

需要编辑数据和维护其空间完整性；

需要使用要素类做高级的 GIS 工作，如建模和分析。

Task2：组织相关要素类为要素数据集

使用**要素数据集**组织空间上相关的要素类为一个普通的要素数据集。如果需要做以下的步骤，则要素数据集是必需的：

添加 topology；

添加 network dataset（必须有 Network Analyst 使用）；

添加 geometric network；

添加 terrain dataset。

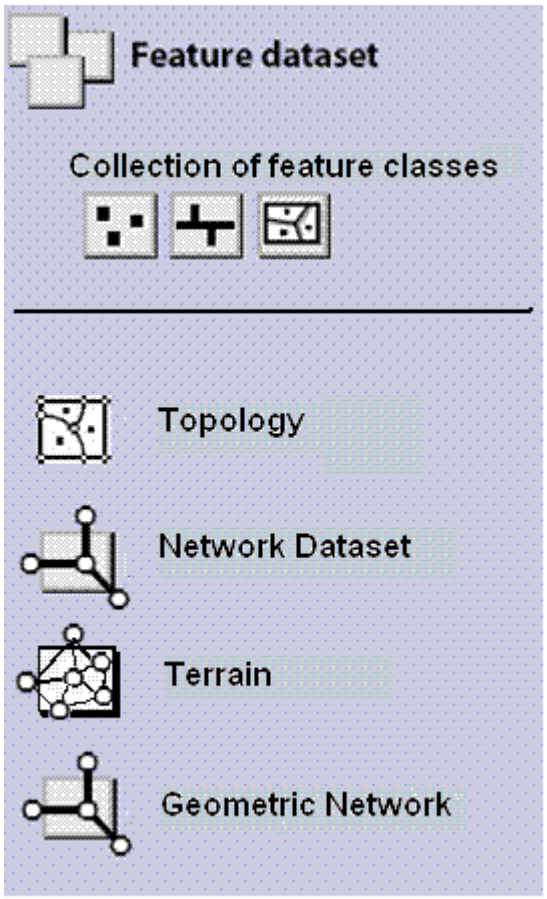
要素数据集是共享同一个坐标系统的空间上或者主题上相关的要素类的集合。要素数据集被用于存储参与共享拓扑、网络数据集、几何网络或者表面的要素类。

有时候，用户对同一专题的要素集合组织为一个独立的要素数据集。如，用户可能有一

个 Water 的要素数据集，包含 Hydro Points（dams, bridges, 和 intakes），Hydro Lines (streams, canals, rivers), 和 Hydro Polygons (lakes, catchment areas, watersheds 等等)。

在一些情况下，用户使用要素数据集作为一个文件夹存储简单要素类集合。这种技术主要用于组织用户如何共享数据集。但是，对编辑来说，这不是一个有利的数据结构。

如果需要，你可以跳跃到 task 3 和 4 决定一个最终的设计，在每个要素数据集中如何组织要素类。



要素数据集在为数据编辑建立许可中扮演着关键的角色。在一个要素数据集中的所有要素类具有相同的许可。这意味着用户可以对要素数据集建立许可，以确定各个组织或团体维护其许可的内容。如果需要对每个 feature class 建立不同的许可，那么要素类就需要被组织为一个独立要素数据集（或要素类），每个含有其自己的许可设置。在这种情况下，extract, transform, load (ETL)或 Import/Export 可以用于各个数据集之间移动数据更新。

何时使用要素数据集：

当在空间上、专题上集成相关要素类时，需要使用要素数据集。主要目的就是建立 topology, network dataset, terrain dataset, 或 geometric network.。

如果要素类参与以下 geodatabase 功能中的任何一个，你必须使用要素数据集来存储要素类的集合。

Topology

Network dataset

Terrain

Geometric network

Task3：添加 geodatabase 元素便于数据编辑和管理数据完整性

Geodatabase 包含一些可选的数据建模功能，即为 GIS 添加完整性规则和编辑行为。这

些功能可以使你的数据管理工组和完整性验证更加自动化。

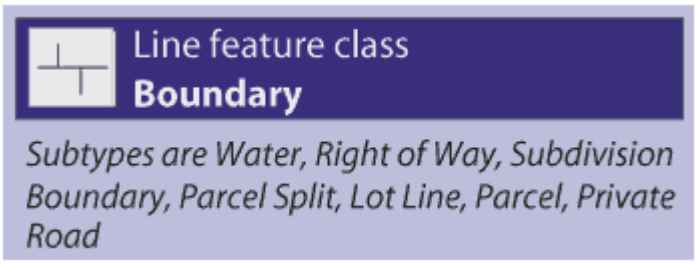
你需要管理属性值的完整性吗？你可以使用 domains，它是描述字段类型的合法值的规则。

Coded value domain
TSDataType

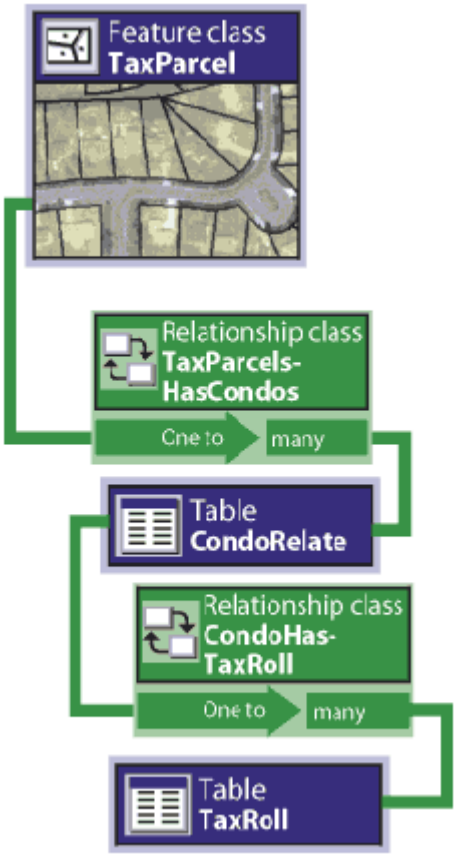
Description
Field type *Long integer*
Split policy *Default value*
Merge policy *Default value*

Code	Description
1	Instantaneous
2	Cumulative
3	Incremental
4	Average
5	Maximum
6	Minimum

你需要使用 subtypes 来帮助你管理一个要素类中要素子集吗？subtypes 允许你为每个 subclass 建立空间行为。它们可以为管理要素子类建立 default rules。例如，当在编辑过程中添加新的要素时，你可以使用 subtypes 自动地给定默认属性值，为每个连接其他要素的新要素设定空间完整性规则，添加其他要素行为。

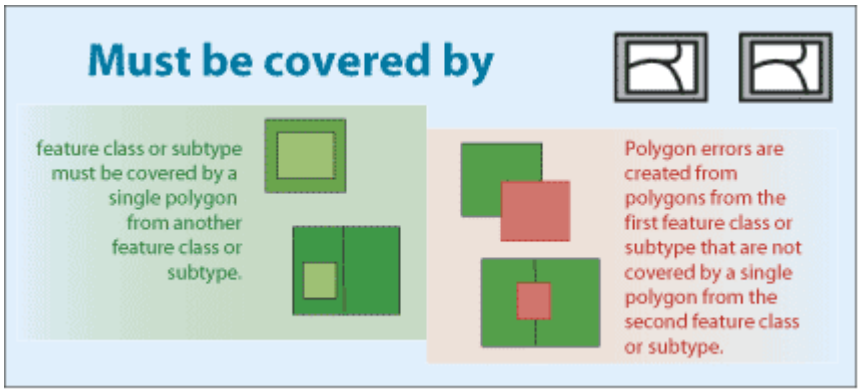


决定是否还有其他相关的表，你是否需要 relationship classes。relationship classes 允许你通过选择相关表中的要素来操作表中的要素，这是关系数据库的一个通用功能。



决定两个要素类中要素是否具有空间关系？例如，你的地块是否有公共的边界？你要保证道路段与另一段相连接或电力线在交叉点和开关处相接吗？你有 county 的边界套入了 states 边界是否不允许重叠？你是否有 vegetation classes 与其他的 environmental 层共享边界，如 slope, aspect, 和 soil type polygons?在这种情况下，topology 是有用的，事实上是基本的。

参与了任何 topology 的要素类必须组织为相同的要素数据集。可以看 topologies 了解在要素数据集中如何使用它，在编辑和更新操作中组织和管理拓扑关系的完整性。



Task4：为高级数据使用、分析模型（如网络分析和地理编码）和高级制图添加功能。

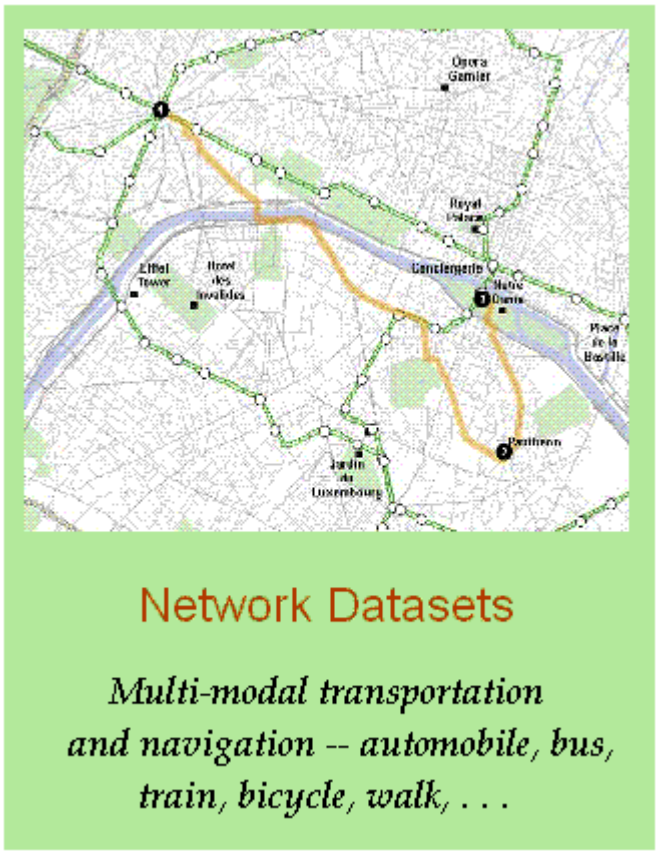
对每个数据集，你可以考虑添加另外的 geodatabase 功能，帮助你进一步利用每个数据集。大量的可选项可以利用，你可以应用其中的任何一些增添到你的 geodatabase。

你需要模拟和使用拓扑关系通过一个topology的整个nodes, edges, 和 faces吗？共享要素的几何形状会帮助你更实际的模拟你的要素吗？例如，大量地形数据层的多边形和线边

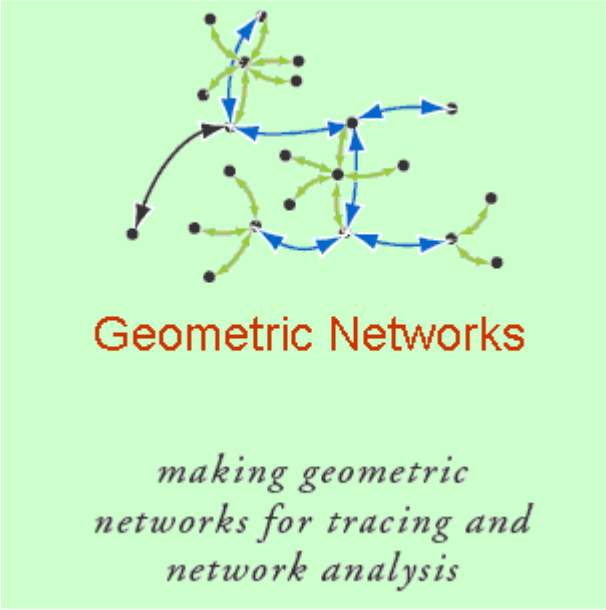
界，如 vegetation, slope, aspect, soil types, geology, water bodies, watersheds, ecological zones 要素类, 还有其他 environmental layers，一个嵌套一个。使用拓扑使公共的边界成一体能够帮助你建立更健壮和一致性的属性结合。这会大大影响 suitability/capability 模型，以及获得对问题真实看法的能力。拓扑也可以帮助你整合要素，如 parcel systems, census units, administrative boundaries,及其他的信息单元。GIS 用户有时称此为 GIS 数据层的垂直同一管理。

你想模拟一个运输网络吗？geodatabase 使用 network datasets 来模拟这种情形。一个 network dataset 是 edges, turns, 和 junctions 的集合，通过 network dataset 可以模拟航运、货物的流通、资源流动等。每个 network 都具有一套导航性能，包括沿着每条 edge（边）的花费，及转移到另一条边花费，还有模拟单行道、左转、其他旅行限制和多式联运网络（使用汽车、公交车和步行的总和性旅行）。

一个 network dataset 使用要素类做为 edges、junctions 和 turns 的数据源。你为每个要素类指定其在一个网络中的角色及导航特征。参与一个 network 的要素类必须组织在一个相同要素数据集中。



你想模拟一个 utility networks（效用网络）吗？在 geodatabase 中可以使用 geometric network（几何网络）来模拟水电的使用、暴风雪和下水道系统。几何网络包括网络要素和连接网络要素，边必须通过连接与其他边相连。用于模拟事物的流向，如电流、水流、天然气和暴风雨。每个要素类在几何网络中都被分派一个角色，如 edges 或者 junctions 的集合。网络的连接性由要素属性和几何一致性来定义。例如，阀门（它被存储为一个点要素类）与管线(以线要素存储)的端点相连接。如果阀门是打开的，水就可以沿着它具体指定的方向流动。



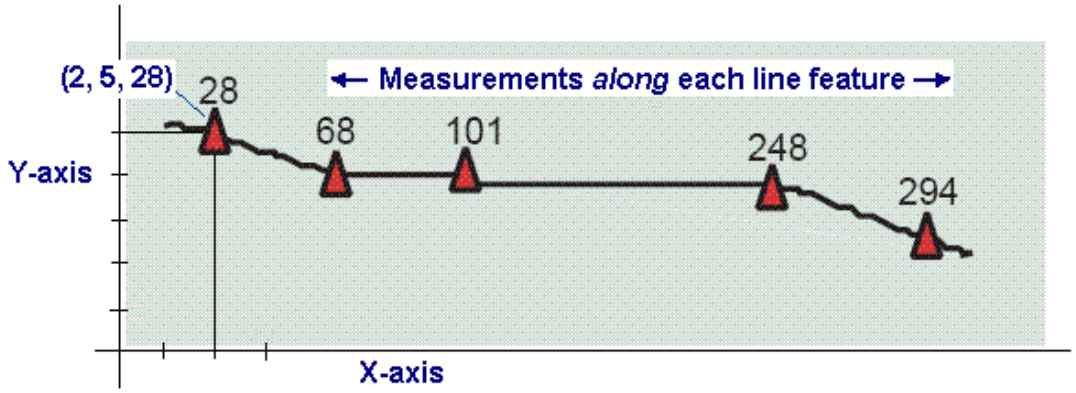
你想使用地理编码吗？对于地址编码，你添加 address locator 到你的 geodatabase 中。一个 locator 是一个或多个包含可设定地址的要素类（如街道中心线的地址信息）集合，一套地址样式和匹配规则。每个 locator dataset 作为源，用来匹配单个地址或者大量地址文件来查询 address locations。



可以创建 locators，并保存其与 geodatabase 独立的复本。这允许你和其他各类用户在他们自己的地理编码工作中可以共享你的 locator。

你想使用 linear referencing 来沿着交通运输线定位事件或设备吗？线性要素顶点也可以包含 m 值。一些 GIS 应用采用线性量度系统来沿着线性要素插入距离，如道路、河流线和管线。你可以给一个要素的每个顶点都赋一个 m 值。一个常见的例子就是公路的里程碑度量系统，被交通运输部门用于记录路况、速度限制、事故发生点和其他沿着道路的事件。两

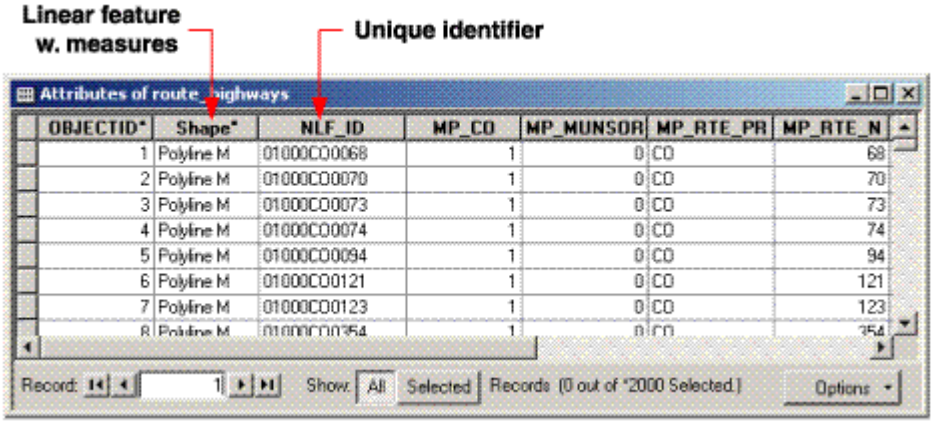
个常用的度量单位包括从一个固定位置的里程碑距离和从一个基准标记的距离。



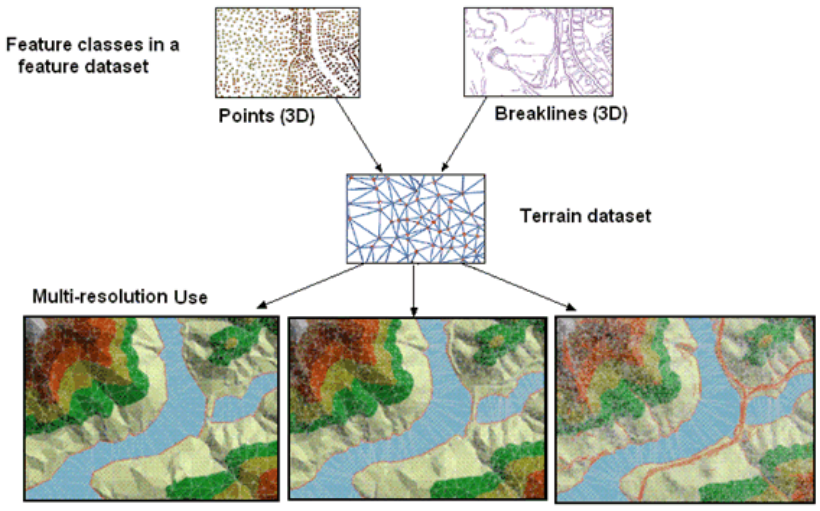
度量顶点可以用（x，y，m）或者（x，y，z，m）。

支持这些数据类型通常称为 linear referencing。沿着这些线性度量系统发生的地理定位事件的过程通常称为 dynamic segmentation。

在 ArcGIS 中线性参照执行的专业术语 route 用于任何线性要素，如城市街道，道路，河流或管线，每个都有一个唯一的标识码，沿着每个线性要素都有一个线性度量系统。具有一个通用度量系统的 routes 集合可以建立如下的一个线要素类。

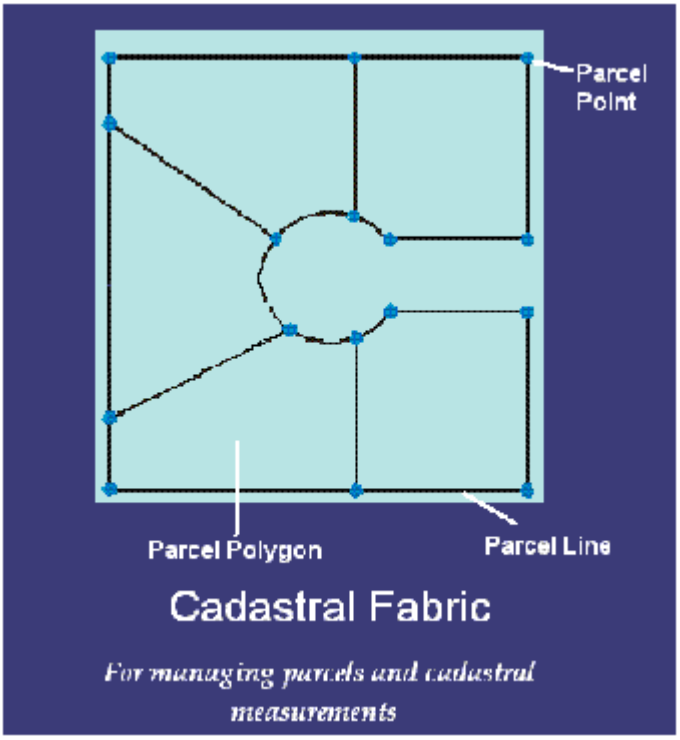


你需要使用不规则三角网模拟高程吗或者需要管理激光雷达点束集合吗？geodatabase 具有一个表面数据集来模拟表面。表面用于管理大量的 3D 点集合以及 3D 要素和生成 TIN。



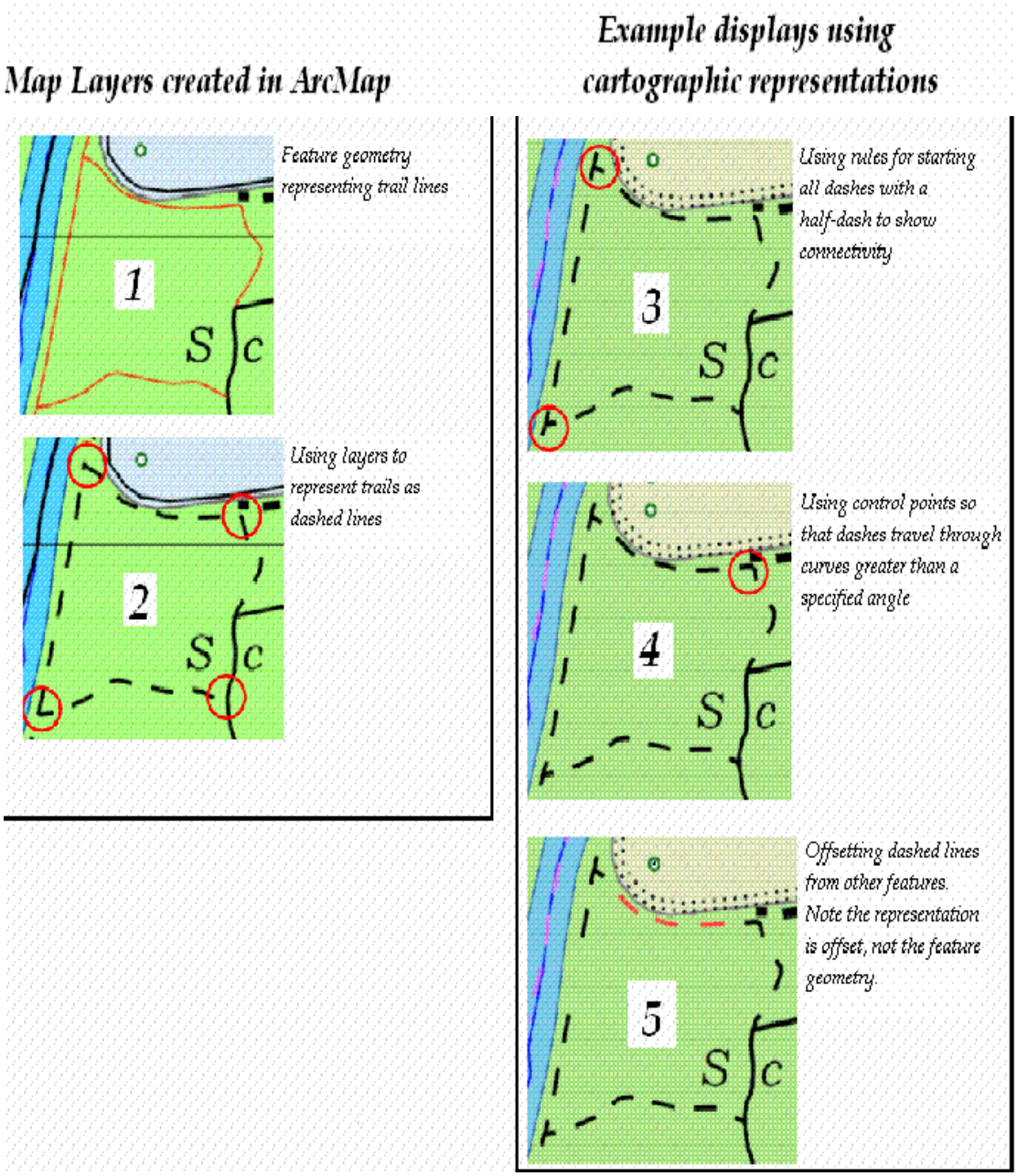
你想管理 parcels 或者 cadastral database 吗？一个 cadastral fabric 就是相连接的 parcels

数据集合。在一个 cadastral fabric 中，parcels 由 parcel 线要素、parcel 点要素和 parcel 多边形要素来表达。cadastral fabric 由 ArcGIS 的 Survey Analyst 扩展模块来创建和实现。



想在要素类中包括 cartographic representations（制图表达）和 rules 吗？cartographic representations 可以添加到一个要素类中，存储制图规则或者用于要素的地图显示的选择性的图形表达。在 GIS 中，大多数用户通过定义一系列地图图层来自动制图。一个地图图层是一系列规则，用于在一幅地图中如何用符号来表现和标注要素。有时候图层并不能足够恰当地传递信息。例如，可能有街道中心线在十字路口相连接。但是，如果你想表达 bridges, overpasses, tunnels,等等，不能容易地在地图上显示它们。

cartographic representations 可以使用户应用特殊的重载、规则和图形来保证地图表达是清晰明了的。例如，在一个地图显示中，道路符号被放大大小显示时，就会与其他要素引起冲突，如 streams 和 buildings。通过制图表达，你可以偏移一些要素来消除冲突——不用必须改变下面要素的地理位置。你可以从河流符号上偏移道路，从道路符号偏移建筑物。



3、6 使用 UML 设计 geodatabase 的注意事项

ArcGIS 支持使用 CASE 工具引入 Unified Markup Language（统一建模语言）来设计 geodatabase。但是，在 UML 数据模型中并不支持所有的地理数据类型、关联和行为。

尽管 UML 为 geodatabase 的 schema 的关联方面的文档化是一个很有用的工具，但是通常并不推荐仅使用 UML 来设计 geodatabase。

UML 在关系数据库设计方面很有用（如，要素类、属性表和其他 geodatabse 属性）。但是，UML 不能设计数据行为－topologies, networks, terrains, raster catalogs, map layers, map symbols, metadata, cartographic representations, semantic classifications, address locators, cadastral fabrics, linear referencing, 和 geoprocessing models。这些数据元素用于定义地理行为和关联。

这些丰富的 geodatabase 行为在 UML 设计中不能被普遍地表达。更为重要的是，在 UML 设计中特殊的 GIS 设计不能实现。绘制面向对象的一个层次类，子类和继承，在 UML 中不

能提供如何模拟你的地理数据的空间关系的设计。例如：

Parcel 边界线如何连接形成封闭的 parcel 多边形；

Parcel corners, boundaries, 和 areas 如何与另外的共享一致的几何形状；

在你的系统中你希望什么样的完整性规则作为地理表达的一部分而被存储。

通常，UML 把设计者从定义 use cases 注意力中分散出来了，这可以帮助你清楚表达关键的地理行为和空间关系。

当然，user communities 可以寻找其他的方法以 UML 来表达他们的地理数据元素。换句话说，使用 UML 可以文档化你的 geodatabase 设计方面的很多（但不是全部）。

另外的，许多 relational modelers 紧密依赖于 UML，并想让他们的 GIS 设计能与他们的其他 DBMS 设计互操作。在这种情形下，你们可以使用 UML 共享 geodatabase schemas 的部分。

概要：UML 是众多方法中的一种（如 entity-relationship 模型），它可以用于有效的关系和表的建模。但是，单独使用 UML 并不足够。UML 并不能代替 GIS 中所要求的地理数据建模必需工作－－空间行为的定义和空间关系的应用。设计步骤在帮助的设计部分已经描述，它们可以给你的 geodatabase 设计的其他方面提供指导。

一个有用的文档化 schema 的工具就是使用 graphic representations，这也是 ESRI 使用的。

3、7 设计技巧

Geodatabae 数据模型被设计为应用于大量的实际应用中。为了保证每一设计步骤易于理解和执行，每个数据模型都设计为支持容易从已存在的数据结构移植，被设计为可变的，可扩展的，容易被你的组织所采用。以下是一些设计技巧，帮助设计的实现。

在已存的 GIS 设计上创建

大多数已存在的数据库设计是适合进一步设计的。你可以在过去已经投入使用的数据库上开始创建，探索新的 geodatabase 功能来提高过去的功能。

使用通用可行的数据库类型

将普通数据结构和强大的 GIS 工具结合提供了最佳的解决方案，它是可伸缩，并支持多用户和多个应用程序。在工作中尽可能使用 ArcGIS 软件，使用定制化的 GIS 数据结构应作为最后的选择。

使用拓扑集成相关的要素类

继承从 ArcInfo 用户的 coverages 会有很多机会使用拓扑来集成要素类。学习如何使用 geodatabase 拓扑和它的规则。它会在编辑过程中创建真正的节约，最小化所需的用户定制工作，提供用户的效率。甚至一个小型的 GIS 组织会发现其数据维护工作将提高 40％。

将 GIS 设计概念和传统的关系数据库设计方法结合起来

对于一个好的 GIS 设计来说，DBMS 和 GIS 设计方法都是关键的。缺一不可。学习应用这两种技术。

原型化 geodatabase 设计

使用 ArcGIS 桌面程序的 file, personal, or ArcSDE Personal geodatabases 设计一个原型是容易的、有趣的和有效的。你会吃惊你在实验中所获得的见识，会使你的设计过程变得更效率更高。

在设计的最后阶段，你需要测试伸缩性和工作流 ，它们代表了你的组织将在 geodatabase 上执行的任务。使用这些技巧来调整你的设计。在最后的测试阶段，实际应用和调整设计是必需的。

4、4 操作 Geodatabase 数据集

4、4、1 表

4、4、1、1 表概要

地理对象属性和特性被存储和管理在 tables。表信息是地理要素的基础，允许可视化、查询和分析数据。

table 的基本信息：

Table 包含 rows。

所有的 row 有相同的 columns。

每个 column 存储一个具体的数据类型，如数值、日期或者字符类型。

一个 columns 的数据类型定义了它可以存储数据的种类，因此，属性的性质是 column 表达的。数据类型是对可能的值的可能在数据上进行操作的分类，以及 column 数据在数据库中存储方法。

Geodatabase 使用以下的数据类型来存储和管理属性：

Numeric—有四种类型的数据类型：

1、 Short integer

2、 Long integer

3、 单精度浮点型，称为'floats'

4、 双精度浮点型，称为'doubles'

Text—文字数字式字符

Date—日期和时间数据

BLOBs—大二进制对象用于存储和管理二进制信息，如符号和 CAD geometries。

Global identifiers—全球唯一标识符，globalID 和 GUID 数据类型，它们存储注册式样的字符串，它们唯一标识了一个或跨多个 geodatabase 中的一个 table row。

下面的图表是一个包含地块信息的典型 table 的例子。在 table 中每个 row 代表了一个单独的地块，它包含了和上面 row 或下面 row 相同的属性 column。在这些 columns 中，可以有 Parcel ID, Zone, 和 Property ID 这些属性。这些属性被存储为合适的数据类型。例如，Parcel ID 值被存储为 short integer，Zone 值被存储为 text。这个 table 组织了地块数据，并且使信息便于管理，容易获取。

在表上执行操作任务的例子

使用表数据可以执行很多制图、分析和数据管理的任务。

Tables 允许绘制地图，可视化显示数据。例如，可以分类属性，使用符号表示为一个层。可以使用人口数量来符号化城市，大城市使用比小城镇和村庄更大的符号。同样，可以指定不同的颜色来表达在每个地块层中每个土地使用类型。另外，也可以使用属性值来生成文本标注在每个地块要素上。在下图中，地块使用土地利用类型来符号化的，然后标注了它们 parcel ID 值。

属性数据帮助执行空间查询和分析。例如，可以用确定属性来调查要素的分布状态，通过让 ArcMap 选择含有需要调查属性的要素。在下面的图中，含 LAND_USE of UNK (unknown)的要素被用一个属性查询来选择。

当 geodatabase 中信息变化时，可以更新属性。例如，当 land uses 或 property ownership 改变时－－或未知值被分类。如果有一个 feature class 表达 pipe，它含有表示 diameter 的字段，当工人移走一个 8 英尺的 pipe，换上一个 6 英尺的 pipe 时，可以容易修改这个属性。可以在 table window 或 Attributes 对话框中编辑表的值，在 Attributes 对话框中仅显示独立选择的要素的属性。

Geodatabase 包含一些功能，它允许增强、维护和强化表数据的完整性。例如，通过建立属性域，可以设置规则指定 table 中 record 的有效值。例如，当更新 pipe 的 diameter 属性

时，可以使用属性域来确保这个 diameter 对这部分 pipe 是合适的。范围域（. Range domains）确保输入的值是在一个有效的范围内的。下图中表示了在编辑时，代码值域（coded value domains）的使用，它允许从一个预定义的列表中选择一个值，避免犯排版错误。

Geodatabase 也允许使用子类型（subtypes）在一个单独的表中创建和管理一组属性子类。每个 subtypes 可以有其自己的属性域集合，给定字段的默认值。例如，可以为 water mains 创建 subtypes，如 distribution mains 和 transition mains，每个 subtype 含有其自己的默认值集合和属性域。

在 tables 之间建立关联（relationships）允许通过一个公有的字段（称为关键字）联系一个 table 和另一 table 中的记录。可以用多种方法建立关联，包括在地图中临时连接或关联各表，或者在 geodatabase 中创建关联类（relationship classes），它可以存储很多永久的关连类。例如，可以使一个 parcel ownership 的 table 与 parcels layer 发生关联，因为它们共享一个 parcel ID 字段。

ArcGIS 也允许转换一个 table 数据到空间数据。例如，最普通的数据源之一就是使用 GPS 获得的一组坐标值。可以很容易地把这些 x、y 数据添加到 ArcMap 中显示它。下图显示了一组包含消防栓的 x、y 位置的文本文件，然后这些点显示在 ArcMap 中地图，最下面的表是 table window。

如果有一组地址，可以使用地理编码来匹配它们到已知的街道位置，创建点要素。另外，通过线性参照，可以仅用一个标识符和位置沿着一个线性要素来表示事件。

Tables 在数据模型的根部，它们是用于更好地模拟真实世界现象，建立 geodatabase 的模板。

表工作流

下面是一些与表相关的任务，可以在帮助系统中找到更多信息。

另外，在 ArcGIS 中有很多地理处理工具可以利用，它们允许执行如数据转换和管理的任务，还可以操作 tables 和 attribute data。不管使用那种 ArcGIS 的许可，都可以使用很多工具，尽管有些是需要 ArcEditor 或 ArcInfo 许可的。

4、4、1、2 在 geodatabase 中设计表

Table 设计步骤

在 geodatabase 中，描述性的属性通常存储在要素类表或一个关系表来描述单独的要素。

定义 geodatbase tables 的过程建立于基本的关系概念，一个及其相似的关系设计过程用于在 geodatabase 中组织 tables。

以下是 table 设计的基本步骤：

1、 识别需要在 table 中描述的对象类型。例如，它们可能是：

- 地块主人
- 土地类型和关于每个土地多边形的描述信息
- 特性和建筑物的描述信息
- 人口数量属性
- 调查区域的简单描述信息

2、 确定需要记录这些对象的属性字段集合。

3、 定义用于存储这些属性的字段类型和值。例如：

- Name 字段用于记录对象的名字，用大概 25 个字符和更少的。
- Road Class 字段用于分类道路。
- Various numeric 字段用于记录关于对象的度量值。
- Date 字段用于指定地块让与日期。

4、 给每个字段一个名字和数据类型。可以看 Data Types 了解更多的信息。

5、 组织这些字段到一个或多个 tables 中。

6、 在每个字段中为每个值定义规则。使用这些来定义 geodatabase 的域，它可以用于管理 geodatabase 中 table column 的字段值的完整性。

7、 在使用多于一个 table 的情况下，定义不同 tables 之间的关联属性。

4、4、1、3 创建表

Tables 代表了 GIS 中三种关键 dataset 的一种，其他两种是要素类和影像数据集（在 ArcGIS 中更多称为 *raster datasets*）。GIS 用户使用 tables 来存储描述性的熟悉。Tables 同样也是一种在 geodatabase 中存储要素类和栅格数据集的机制。

在 ArcGIS 中，一个 table 有一系列 rows 和 columns 用于存储描述信息。每个 row 有一系列带值的字段。相同的字段（columns）出现在每个 row 中。每个 column 有一个相关类型如 integer，含小数位的实数，文本字段，日期字段，shape 字段或二进制字段来存储，还有更复杂的结构，如 feature geometry 或 imagery，一个 XML field 等等。

大多数，用户称每个 row 为一个对象，字段存储每个对象的描述属性。对于要素类，每个 row 中对象就是一个 feature 和 shape 的列，存储 feature 的几何特征（如 point, line 或 a polygon）。对于一个非空间的 table（attribute table），它们有时称为"Aspatial 或 nonspatial 对象。

ArcGIS 用户操作包括多种格式和数据源的 table：

- Text files（用逗号或选项卡分隔的文本文件）
- Excel spread sheets
- dBASE tables
- personal, file, 和 ArcSDE geodatabases 的 table

这些 tables 都被组织在 geodatabase 中，作为 geodatabase 的一部分被管理。这是最基本的一种用于管理关于 feature 和 images 的描述信息的机制。

在 ArcCatalog 创建表

步骤：

1、 在 ArcCatalog 目录树，在需要创建新 table 的 database 上单击右键，指向 New，单击 Table，定义一个新 table 的向导会出现。

2、 输入 table 的名字。为了给这个 table 创建一个假名，输入假名。单击 Next。

3、 Note：如果 geodatabase 不是 file 或 ArcSDE geodatabasee，跳过这一步。

如果是在一个 file geodatabase 中创建一个新的 table，可以选择一个选项 configuration keyword，设置 table 来管理各种语言的文本字段。在大多数情况下，使用 DEFAULTS keyword 是明智的选择。

Note：如果需要存储亚洲字符如日语或汉语到 table 中文本字段，需要为 file geodatabase 选择 TEXT_UTF16 keyword 选项。

4、 需要添加一个字段到 table，单击 Field Name 列下面的一个空白行，输入字段名字。

5、 单击新字段名字边上的 Data Type 列，然后单击它的 data Type。

6、 为了给这个字段创建一个假名，单击 Alias，输入这个字段的假名。

7、 为了防止空值存储在这个字段，单击 Allow nulls，单击下来箭头，然后单击 No。

8、 为了给这个字段关联一个默认值，单击 Default value，输入数值。

9、 为了给这个字段关联一个域，单击 Domain，单击下拉箭头查看应用于这个字段的域值然后单击 domain 字段。

10、为了设置字段类型的其他属性，单击下拉列表内的属性和输入属性。

11、重复前面 8 个步骤，直到所有的表字段被定义。

12、单击 Finish。

Note：

- Geodatabase 中所有简单的 tables 都需要 ObjectID 类型字段。ObjectID 字段惟一标识每个存储在数据库中表的对象。
- 当创建一个新的 table 时，可以使用其他 table 作为一个模板。单击 Import，将鼠标移动至包含需要 copy 字段定义的 table 上，单击 OK，现在就可以编辑字段名字和其他数据类型。
- 如果已经输入一个字段，但是不想把它包含在新 table 中，选择它单击其格网标签然后按下 Delete 键。

使用 Geoprocessing 工具创建表

步骤：

1、 在 ArcToolbox 上，可以在名为 Data Management Tools 的 toolbox 上找到 Create Table 工具。它包含在 Tables toolset 中。

2、 指定将要创建新 table 所在的 geodatabase 的名字或指定输出工作空间的名字，新的 dBASE table 将在其中创建。

3、 给定新 table 的名字，可选 table，它的字段可以作为新 table 字段的模板。

4、 如果是在一个 file geodatabase 中创建一个新的 table，可以选择一个选项 configuration keyword，设置 table 来管理各种语言的文本字段。可以选择一个任意的 configuration keyword 来设置 table。在大多数情况下，使用 DEFAULTS keyword 是明智的选择。

Note：如果需要存储亚洲字符如日语或汉语到 table 中文本字段，需要为 file geodatabase 选择 TEXT_UTF16 keyword 选项。

4、4、1、4 创建属性索引

Note：添加和修改 ArcSDE geodatabases 的属性索引在 ArcView 中是不可用的。

属性索引可以提高对要素类或表的属性查询速度。属性索引是 ArcGIS 用于检索表中记录的一种可选途径。对于大多数属性查询类型，使用索引查询一个记录要比从第一个记录开始遍历整个表查询要快得多。

一旦一个要素类或表中含有数据，在频繁使用的字段上建立属性索引。应该建立那些真正需要的索引，这是因为每添加一个索引就会使对要素类的编辑稍微地变慢。每次编辑这个要素类，ArcGIS 都必须更新这些索引。如果需要频繁编辑一个字段，尽量避免在它上面创建索引。属性索引可以通过访问 ArcCatalog 中属性对话框或 geoprocessing 来创建。一旦一个索引被添加，它可以在任何时候被删除和再添加。

属性索引域可以用不同的方法创建。它们可以为单个或多个字段创建；它们可以是惟一的；对于一些 geodatabase，它们可以按升序或降序创建。

在 ArcCatalog 中创建属性索引

属性索引可以建立在一个 geodatabase 的要素类或表的单个或多个字段。在 ArcCatalog 目录树上，鼠标放在需要建立属性索引的表或要素类上，单击右键，访问 Properties 对话框。

1、 在 Catalog 目录树上，在需要建立索引的表或者要素类上，单击右键，单击 Properties。

2、 单击 Indexes 标签，进入 Indexes 选项卡。

3、 单击 Add 按钮，打开 Add Attribute Index 对话框。

4、 在 Name 文本框输入新索引名称。

5、 如果索引是惟一的，选中 Unique 复选框；如果索引降按升序，选中 Ascending 复选框。

6、 单击需要建立索引的字段。

7、 单击右箭头按钮，把选择字段移到 Fields selected 窗口。

8、 单击上下箭头，改变选择字段在索引中的顺序。

9、 单击 OK 按钮。

10、单击 Apply 按钮建立属性索引，或单击 OK 按钮建立索引，关闭 Properties 对话框。

使用 geoprocessing 工具创建属性索引

Data Management toolbox 中的 Indexes toolset（工具箱）提供了两种基本的工具建立和移动属性索引。

Add Attribute Index 为已存在的表，要素类或属性关联类添加一个单独或多个字段的索引。这对任何 ArcGIS 许可都是可用的。

Remove Attribute Index 从一个已存在的表，要素类或属性关联类移除单个或多个索引。这也是对任何 ArcGIS 许可都是可用的。

属性索引名字

当命名一个 ArcSDE geodatabase 中的索引时，给索引一个反映表或列的名字是一个良好的习惯做法。但是，如果被索引的表的名称改变了，索引名字就不能再显示哪个表被索引。一些机构发现给一个索引能显示它是一个索引的名字是非常有帮助的，如添加一个 IDX 到名字的开头或结尾。例如，一个地址表上的索引可以称为 ADRS_APK_IDX，ADRS 显示这个索引是在地址表上的，APK 表明这个列被索引，IDX 很明显表示这是一个索引。

与表名相似，在 ArcSDE geodatabases 中索引名：

- 在数据库中必须是惟一的
- 必须以字符开始
- 不能包含空格字符
- 不能包含保留字。

对于如何命名一个 file geodatabase 中的属性索引没有任何限制。Persoanl geodatabase 中的索引名字不能包含空格字符和保留字。

唯一索引

当建立一个索引时，会看到一个建立此索引为惟一的选项。如果在每个记录中这个属性都仅有一个惟一的值选择这个选项。针对这个属性，这将会大大加快查询执行速度，由于数据库在找到第一个匹配值后就停止搜索。

注意，但是，在一个 personal geodatabase 中不能编辑一个在自定义字段上含惟一索引的要素类。同时，也不能编辑一个要素类，它所在要素数据集中的另外一个要素类在自定义字段上含惟一索引。当试着启动编辑一个 personal geodatabase，ArcMap 会返回这样一个错误。"Could not edit any of the map's layers. Check to see if a layer or table contains a unique index on a user managed column."。

如果在 personal geodatabase 中有一个字段含有一个惟一索引，需要编辑它，使用 ArcCatalog 去掉惟一索引，重新作为一个非惟一索引来编辑。

升序 vs. 降序索引

当建立索引时，会看到一个建立这个索引为升序的选项，如果没有选择这个选项，就是按降序建立的。一个升序索引按上升的顺序存储。例如，城市的名称是 Athens, Berlin, London, and Paris 就是按照升序索引出现的，降序索引将是以 Paris, London, Berlin, and Athens 形式出现。

大多数情况下，索引的顺序方向对检索的速度影响很小甚至没有影响，因为对于大多数查询，索引向前和向后移动是相同效率的。

单列 vs.多列索引

在 personal geodatabase 或 ArcSDE geodatabase 中索引可以为单个列或多个列建立。如果

在一个查询中频繁指定两个或三个字段一起，Multicolumn indexes 是很有用的。在这种情况下，Multicolumn indexes 提供了比 2 个或 3 个单独的索引更快的查询性能。

字段在一个 Multicolumn indexes 中的顺序是重要的。在一个 Multicolumn indexes 中，A 列在 B 列前面，A 列将用于执行初始化查询。同时，这样的索引对仅包含 A 的查询比仅包含 B 的查询更有用。

决定是建立 multicolumn 或 single column 索引，还是这两种的混合很少是明显的。尽管通常有大量的解决方案可以操作。例如，如果有时仅查询 A 列，有时仅查询 B 列，有时两列都查询，可以选择以下的任何一种方法：

- 给 A 列和 B 列分别建立两个独立的索引。
- 给 A 列和 B 列建立一个 multicolumn 索引。这个索引将对包含两个列的查询效率更高。对于仅包含 A 的查询，这种索引要比仅对 A 的索引要慢很多。这种索引对仅包含 B 的查询几乎没有用。为了弥补，可以在 B 上建立另外一个索引。
- 建立三种索引：A 索引，B 索引，A 和 B 的 multicolumn 索引。如果三种查询类型是常见的，并且这个表的查询比更新更多，这种索引是很有意义的。

4、4、2 要素类

4、4、2、1 创建要素类概要

一个要素类是具有相同几何特征的地理要素的集合（如点，线或多边形），以及相同的属性字段。街道、井、地块、土地类型和人口普查范围都是要素类的例子。

相关的要素类通常组织在一个要素数据集中。如果需要使用拓扑来管理要素是如何共享几何图形，给效用层建立一个几何网络，给路径和最优化建立一个网络数据集，或建立一个表面（TIN 数据结构），需要组织要素类到一个要素数据集中。

在 geodatabase 中有四种基本的方法建立一个新的要素类：

- 使用 ArcCatalog
- 使用一个 geoprocessing tool 名为 Create Feature Class 的工具
- 在 ArcMap 中保存一个地图层中数据
- 转换一个外部数据源到一个 geodatabase 要素类（如转换一个 shapefile 或者一个 CAD file）。

在 AtcCatalog 中创建一个空的要素类。一个新的要素类可以作为一个要素数据集的部分或 geodatabase 中一个独立的要素类被创建。它们也可以包含在以下的数据结构中：shapefile, coverage, CAD, SDC, 和 VPF。

一旦创建了一个要素类，并用要素装载了它，可以通过创建属性索引来提高属性查询的速度。

4、4、2、2 定义要素类属性

当创建一个新的要素类时，必须指定一些要素类的属性，它们定义了要素类的结构。

在大多数情况下，最好的选择就是接受这些属性的默认值，由 Create Feature Class wizard 向导提供。但是，这部分描述了每个要素类属性，因此了解何时和为什么使用除了默认值外的其他值，还有如何改变这些值将会影响数据。

创建适合数据模型的一个合理的要素类依赖于以下的要素类属性。

Name/Alias

要素类名字是标识要素类的唯一句柄。输入的名字是不能包含空格，标点符号，或以数字开头。最流行的命名要素类的方式就是使用混合的或使用下划线，如"MajorRoads" or "Major_Roads"。

可以随意给要素类一个假名，它作为一个又名。不同与要素类的真名，假名可以包含空格、标点符号、可以以数字开头。例如，可以设置"MajorRoads"的假名为"Major Roads" or "5

Major Roads"。

在 ArcMap 中，假名是自动用于要素类的。但是，在 ArcCatalog 中，要素类通常是用它们的真名表达的。

Type

当创建一个新的要素类时，可以有几种不同类型的要素来选择。不同类型数据组成的要素类决定创建何种类型的要素。例如，well locations 被存储为点要素类，road 存储为线要素类，parcels 存储为多边形要素等等。

以下是可用要素类型的列表及何时使用它们：

Points—用来表达那些很小且不能用线或多边形来表示的地理要素（如 GPS 观测站）

Lines—用来表达那些长条形的，非常狭窄也不能用多边形表示的地理要素，如街道中心线和溪流。线也可以表达那些有长度却没有面积的要素，如轮廓线和边界。

Polygons—是一个封闭的图形，用来表示均质要素的形状和位置，如州、县、土地、土壤类型或者土地利用类型区等。

Annotation—地图文本，包括文本如何组织的属性；例如，除了每个注记的文本字符串。还包括其他属性，如放置文字形状，字体，字体大小和其他显示属性。注记也可以是 feature-linked，也可以包含子类。

Dimensions—一种特殊的注记，它显示具体的长度或距离；例如，为了显示一个建筑物或一块地的一条边的长度，或者两个不同要素之间的距离。维度经常用于 GIS 设计、工程和工具应用中。

Multipoints—由不止一个点组成的要素。Multipoints 通常用于管理大量点集合的数组，如激光雷达点串，它们可以包含几十亿个点。使用一个单独的列表示这样的点几何是不可行的。聚类这些多点行可以使 geodatabase 处理大块点集合。

Multipatches—一个 3D 的几何形状用于表示要素的外部表面或外形，它可以应用一个离散的区域或三维空间的体积。

Multipatches 是由平面的 3D 环和三角形组成，三角形是被用于联合模拟一个三维的外形。Multipatches 可以用于表达任何从简单对象如球和立方体，到复杂对象如建筑物。

Geometry 属性

当创建一个新的要素类时，可以有一个选项允许坐标包含度量值（m），或用于三维数据的 z 值。

是否需要使用 m 或 z 值取决于将要使用的数据类型。

数据中包含 m 值，允许属性值被存储在点坐标的顶点。在线性参照的情况下，m 值沿着线性要素存储度量值到顶点。这允许一个位置沿着这条线被查找。如果将使用线性参照或动态分割应用到数据，将需要坐标包括 m 值。

Z 值用于表示高程或其他给定表面位置的属性。在一个高程或地面模型中，z 值表示高程；在其他地面模型中，它表示密度或一个特定属性的数量如年降雨量，人口和其他表面度量值如果打算模拟高程，创建表面或处理任何三维的表面，都需要坐标包括 z 值。

坐标系统

当创建一个新的要素类时，必须或可能要选择一个坐标系统。坐标系统，还有容差值和精度值组成了一个要素类的空间参考系统。空间参考描述了要素在真实世界中的位置。

可以通过几种方法来定义新要素类的坐标系统：

- 选择由 ArcCatalog 提供的预定义的坐标系统中的一个。鼠标移动至一个地理或投影坐标系统，它能适当表达数据模型的区域。
- 导入另一个要素类使用的坐标系统参数。如果需要使用另一个要素类的坐标系统作为一个模板，可以有一个选项浏览并导入它。
- 定义一个新的，自定义的坐标系统。可以输入值来创建一个坐标系统满足需求。

如果选择包含 z 值的坐标系统，必须也指定一个垂直坐标系统。一个垂直坐标系统(VCS)参考 z 值，大多数用于表示高程。一个垂直坐标系统包括一个测量或垂直数据，一个线性度量单位，一个轴方向和一个纵向移动。

M 或度量值没有坐标系统。

如果没有数据的坐标系统信息或者不知道使用哪个坐标系统，可以选择一个 Unknown coordinate system。

修改选项允许复查或编辑一个坐标系统的属性。

Tolerance

一个空间参照也包括容差值，xy，z 和 m 坐标都有相关的容差值，它们反映了坐标数据的精度。容差值不同坐标之间的最小距离。如果一个坐标与另一个坐标距离小于容差值，这两个坐标被认为是在同一个位置。这个值用于关联和拓扑操作中，当决定两个是否足够近以至于被赋予相同的坐标值，或者它们足够远可以赋予每个它们自己的坐标值。

默认的容差是 0.001 米或它相等的地图单位。这是默认精度值的 10 倍，也是大多数情况下推荐值。最小允许的容差值是精度值的 2 倍。设置高的容差值会导致低精度，设置低的容差值会导致高精度。

Resolution 和 domain

要素类或要素数据集的所有坐标都是依据选择的坐标系统被地理引用，然后对齐到一个格网。这个格网是用精度定义的，它决定了坐标值的精度（如有效数字的位数）。这个精度建立了覆盖在要素类或者要素数据集上的最合适的格网大。

精度值与相关坐标系统是相同的单位。例如，一个空间参考是使用一个以米为单位的投影坐标系统，精度值也是被定义为米的。应该使用一个至少比容差值小 10 倍的精度值。

默认（推荐）的精度值是 0.0001 meters (1/10 mm)。

例如，如果一个要素类以 state plane feet 存储，默认的精度将是 0.0003281 feet (0.003937 inches)。如果坐标系统是 latitude-longitude，默认的精度将是 0.000000001 degrees。

对于未知的坐标系统，或者 M 值，将必须设置适合数据类型的精度值，不用设置度量单位。

Configuration keywords

一个数据库存储 configuration keywords 可以微调数据如何存储在一个 file geodatabase 或一个 ArcSDE geodatabase。configuration 参数被集合为一个或多个 configuration keywords，它们的中的一个就是默认的 configuration keywords，它指定了默认的存储参数。

选择 configuration keywords 并不被 personal geodatabase 支持。ArcSDE Personal 和 ArcSDE Workgroup geodatabases 仅支持默认的存储参数（DEFAULTS configuration keyword）。

当在一个 file geodatabase 或 ArcSDE geodatabase 中创建一个要素类时，可以告诉数据库使用哪一种 configuration keyword。在大多数情况下 Default keyword 应该被使用。在一些情况下，当创建特殊的数据集或数据类型时，需要指定可选择性的 configuration keywords，以取得最佳的性能或微调数据在数据库存储的某些方面。

以下是 configuration keywords 的一些例子及它们的使用：

DEFAULTS－它使用合理的默认 configuration，为大多数 geodatabase 使用存储设置。

MAX_FILE_SIZE_256TB－如果导入一个及其大的影像到一个 file geodatabase 中，可以指定 MAX_FILE_SIZE_256TB configuration keyword，它告诉 geodatabase 允许栅格数据集达到 256 terabytes。

SDO_GEOMETRY－如果需要添加一个栅格数据集到 ArcSDE 的 orcale geodatabase，可以指定 SDO_GEOMETRY configuration keyword，它告诉数据库以 Oracle GeoRaster 格式来存储栅格。

TEXT_UTF16－如果拷贝一个包含汉语字符的要素类到一个 file geodatabase，可以指定 TEXT_UTF16 configuration keyword，因此在属性列中的文本字符就被存储在 UTF-16，它存储汉语字符是效率比较高。

字段和字段属性

在 ArcCatalog 中创建一个新要素类时，可以指定许多字段被添加。也可以指定字段的属性，如字段类型，能在字段上存储的最大容量。每个字段类型具有特定的属性：

Alias—要素类字段的假名

不同于一个字段的真名，假名不受数据库的限制，可以包含空格，特殊字符，以数字开头。

Allow Nulls—如果不需要字段存储空值，设置是否允许为空属性为“no”。

Default Value—当在 ArcMap 中用编辑工具创建一个新要素或对象时，可以输入一个默认值，它会自动生成的。

Length—这是一个文本字段的属性，它决定了可以输入的字符的最大数量。

所有的要素类都有必需字段集合，这是记录要素类中特定对象状态所必需的。这些必需字段在建立新要素类时自动建立，并且不能被删除。必需字段可能有一些必需的属性如域属性，必需字段的必需属性是不能修改。例如，在一个多边形要素类中，OBJECTID 和 Shape 是必需字段，这些字段有可修改的属性，如假名、几何类型，但这些字段不能删除。

当创建一个新的要素类时，可以有选项导入另一个要素类或表的字段。这个选项可以使用其他的要素类或表作为你创建要素类或表的字段定义的一个模板。一旦导入这个字段，可以编辑字段名字，数据类型和属性。

在创建一个新要素类时导入字段，必需字段是不受影响的。例如，如果已经设置新要素类的几何类型属性为 Point，但是从一个要素类导入字段定义，这个要素类的 SHAPE 字段的几何类型属性是 Polygon，这不会重写 Point 属性。

4、4、2、3 创建要素类

关于创建要素类

在 geodatabase 中有四种基本的方法来创建一个新的要素类：

- 使用 ArcCatalog
- 使用 geoprocessing 中名为 Create Feature Class 的命令
- 保存 ArcMap 中一个地图图层的数据
- 转换一个外部数据源到一个 geodatabase 要素类（如转换一个 shapefile 或 CAD 件）

在 ArcCatalog 中创建一个空的要素类。一个新的要素类可以作为一个要素数据集的一部分或作为 geodatabase 中一个独立要素类被创建。它们可以包含在以下的数据结构中：shapefile, coverage, CAD, SDC, 和 VPF。

当创建一个要素类时，选择创建存储 polygons, lines, points, annotation, multipoint, multipatch，还是 dimension 要素。

一个数据集中的所有要素类必需使用相同的空间参考，当要素数据集在创建时空间参考就被定义了。一个例外就是 m 域，相同要素数据集中的要素类可以有不同的 m 域。当创建

一个独立的要素类时，必须定义它的空间参考。

如何创建要素类

创建一个要素数据集中的要素类

1、 在 ArcCatalog 目录树，鼠标放在需要建立新要素类的要素数据集，单击右键。

2、 鼠标指针指向 New，展开建立新数据库项快捷菜单。

3、 单击 Feature Class。

4、 在 Name 文本框输入要素类的名称，在 Alias 文本框输入要素类假名。

5、 从要素类型的下拉列表中选择存储要素类的类型。

6、 如果需要 m 或 z 值，选取相应的复选框。

7、 单击 Next。

8、 如果这个数据集中的任何要素类有度量值，输入 M 容差值或接受默认值。

9、 选取 Accept default resolution and domain extent 复选框，接受默认值。如果需要手动调整新新数据集的 resolution and domain extent 值，不要选择这个复选框。如果是在 ArcGIS9.2 以前的 geodatabase 建立要素类，或选择了未知的水平坐标系统，应该取消选择这个复选框，确认默认值为合适的。

10、如果已经取消选择 Accept default resolution and domain extent 复选框，在向导中会出现另外一个面板允许你更改 M 精度，还有最小和最大 m 值。

11、如果 geodatabase 不是 file 或 ArcSDE Enterprise geodatabase，跳跃至步骤 13，如果需要使用自定义存储关键字来建立表，单击 Use configuration keyword，然后指定需要的关键字。

12、单击 Next。

13、为了给要素类添加一个字段，单击 Field Name 列下一个空白行，输入新字段名。

14、在 Data Type 列，单击新字段对应的单元格，确定新字段的数据类型。

15、为了给新字段建立一个假名，单击 Alias 右边的单元格，输入新字段的假名。

16、为了防止在这个新字段存储空值，单击 Allow nulls 右边的单元格，单击下拉列表框，单击 No。

17、为了建立默认值与字段的关联关系，单击 Default value 的右边的单元格，输入值。

18、为了把一个域与这个字段关联，单击 Domain 右边的单元格，单击下拉列表箭头可以看到应用这个字段类型的一系列域，单击这个字段的域。

19、为了设置字段的其他属性，单击属性下拉列表或输入属性。

20、重复步骤 13 到 19 直到所有的要素类字段都被定义。

21、如果需要从另一个要素类或表导入字段定义，单击 Import。

22、单击 Finish。

提示：

当建立一个新的要素类时，可以使用另一个要素类作为一个模板。单击 Import，鼠标移动至要素类，它包含需要 copy 的字段定义，然后单击 OK。现在可以编辑字段名字和其数据类型。

Geodatabase 中所有的简单要素类都必需 ObjectID 和 geometry type 字段。不能删除默认的 ObjectID 和 geometry type 字段。

当添加一个要素类到一个要素数据集，空间参考按钮允许查看要素数据集的空间参考参数；但是，不能修改它们。一个例外就是 m 域；在相同要素数据集中的要素类可以有不同的 m 域。

创建一个独立要素类

1、 在 ArcCatalog 目录树中，鼠标放在需要建立独立要素类的数据库，单击右键。

2、 鼠标指针指向 New。

3、 单击 Feature Class。

4、 输入新要素类的名字。需要给这个要素类建立假名，输入假名。

5、 从下拉列表中选择要素类型，它将存储这个要素类。

6、 如果数据需要 m 或 z 值，选取相应的复选框。

7、 单击 Next。

8、 鼠标移动至需要使用的空间参考，或单击 Import，选择作为空间参考模板的要素类或要素数据集。

9、 如果需要修改所选择坐标系统的任何参数，单击 Modify 按钮。编辑所选择的坐标系统参数，单击 OK。

10、如果数据需要一个垂直坐标系统，可以从其他要素类或要素数据集导入；另外，选择 None。如果需要修改所选择坐标系统的任何参数，单击 Modify 按钮。编辑所选择的坐标系统参数，单击 OK。

11、输入一个 xy tolerance 或接受默认值。默认的 xy tolerance 在接近投影的地球表面上是 1mm。

12、如果要素数据集中的任何要素类有 z 值，输入 z tolerance 或接受默认值。

13、如果要素数据集中的任何要素类有度量值，输入 m tolerance 或接受默认值。

14、如果已经修改了 xy, z and/or m tolerances，需要恢复默认值，单击 Reset all tolerances 按钮。

15、选取 Accept default resolution and domain extent 复选框，接受默认值。如果需要手动调整新数据集的 resolution and domain extent 值，不要选择这个复选框。如果是在 ArcGIS9.2 以前的 geodatabase 建立要素类，或选择了未知的水平坐标系统，应该取消选择这个复选框，确认默认值为合适的。

16、如果你已经取消选择 Accept default resolution and domain extent 复选框，在向导中会出现另外一个面板允许你更改 xy, z 和 m resolution，还有最小和最大 z 和 m 值。

17、如果 geodatabase 不是 file 或 ArcSDE Enterprise geodatabase，跳跃至步骤 13，如果需要使用自定义存储关键字来建立表，单击 Use configuration keyword，然后指定需要的关键字。

18、单击 Next。

19、为了给要素类添加一个字段，单击 Field Name 列下一个空白行，输入新字段名。

20、在 Data Type 列，单击新字段对应的单元格，确定新字段的数据类型。

21、为了给新字段建立一个假名，单击 Alias 右边的单元格，输入新字段的假名。

22、为了防止在这个新字段存储空值，单击 Allow nulls 右边的单元格，单击下拉列表框，单击 No。

23、为了建立默认值与字段的关联关系，单击 Default value 的右边的单元格，输入值。

24、为了把一个域与这个字段关联，单击 Domain 右边的单元格，单击下拉列表箭头可以看到应用这个字段类型的一系列域，单击这个字段的域。

25、为了设置字段的其他属性，单击属性下拉列表或输入属性。

26、重复步骤 19 到 25 直到所有的要素类字段都被定义。

27、如果需要从另一个要素类或表导入字段定义，单击 Import。

28、单击 Finish。

使用 geoprocessing 工具创建一个要素类

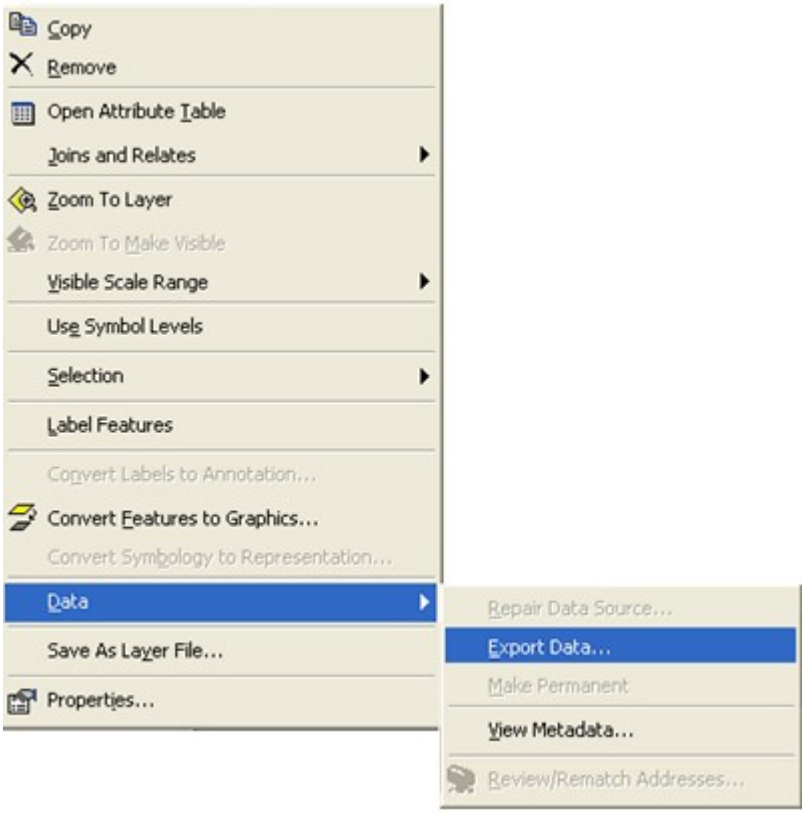
Geoprocessing tool 中 Creating feature class 命令位于 Data Management toolbox 的 Feature Class 下，和 ArcCatalog New Feature Class wizard 一样，它建立了一个空的要素类。

在 ArcMap 中通过保存一个地图数据层创建一个要素类

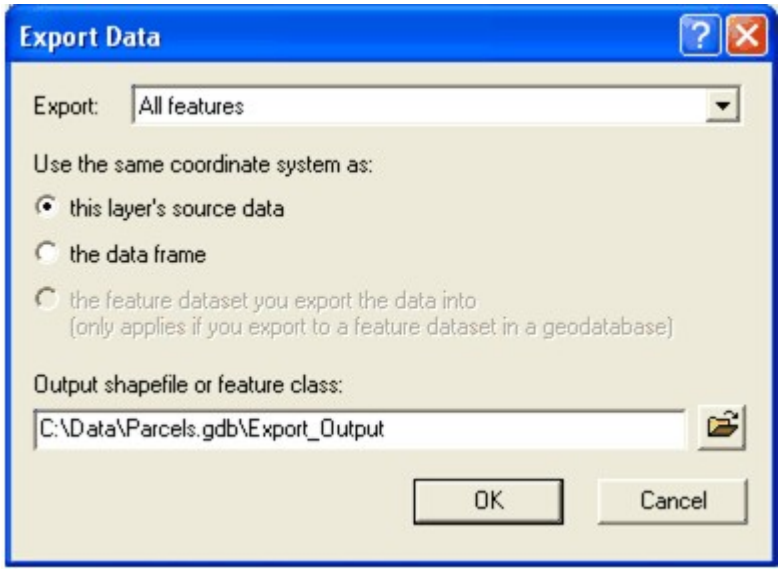
如果使用 ArcMap 处理一个数据集，可以导出它建立一个要素类或一个 shapefile 文件。

1、 在需要导出的数据集上，单击右键，展开一个上下文快捷菜单。

2、 鼠标放在 Data，选择 Export Data。



3、 打开 Export Data.对话框。



4、 从下拉列表中，选择是否需要导出所有的要素或仅是视图范围。

5、 选择以下三种单项按钮中的一种，确定导入数据集的坐标系统需要继承。

- This layer ’s source data
- The data frame
- The feature dataset you export data into

6、 指定需要建立的输出数据。

按照如下的所示的其中一种来做：

直接输入输出名字到字段。它将被保存到当前的输出位置。移动光标到 Browse 按钮，获取当前输出位置的信息。

或者：

如果保存输出到一个文件夹，它将被保存为一个 shapefile，在 shapefile 名字中可以使用空格。如果不指定 a .shp 扩展，ArcMap 将自动添加。

如果保存输出到一个 geodatabase，它将被保存为一个 geodatabase 要素类。可以在要素类的名字中使用空格。

转换一个外部数据源到一个 geodatabase 要素类

可以使用 ArcCatalog 或者 ArcToolbox 中导入工具来转换一个外部数据源到一个 geodatabase 要素类。

有两种方法可以导入一个要素类：

使用 ArcCatalog 中的 context sensitive menu；

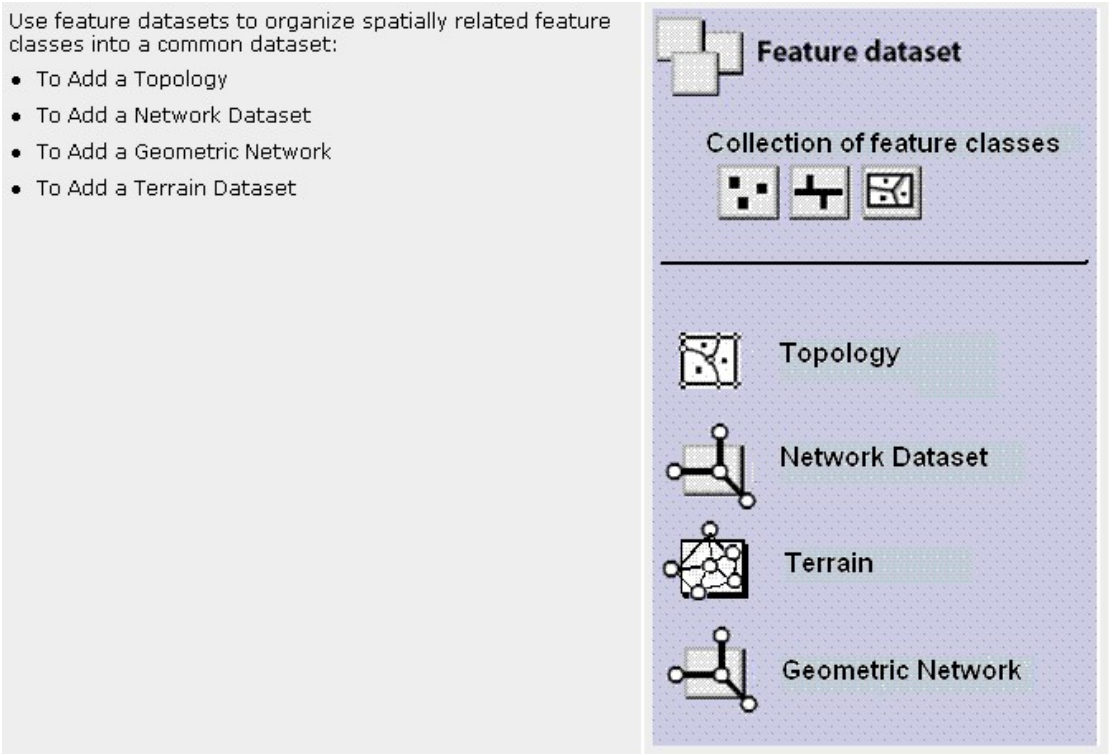
使用 ArcToolbox 中的 To Geodatabase 工具。

当导入的数据格式不是 geodatatabase 所使用的，ArcGIS 会自动转换它到 geodatabase 所使用的数据类型。

4、4、3 要素数据集

4、4、3、1 要素数据集概要

一个要素数据集是相关要素类的集合，这些要素类共享一个公共的坐标系统。要素数据集被用于在空间上或主题上集成相关的要素类。它们的基本目的就是组织相关要素类到一个公共的数据集来建立拓扑、网络数据集、表面数据集或几何网络。



还有一些其他情况，用户在他们的 geodatabase 中应用要素数据集：

组织主题上相关的要素类

有时，用户组织一个共同主题的要素类集合为一个独立的要素数据集。例如，用户可

能有一个 water 所要素数据集，其中包含 Hydro Points, Hydro Lines, 和 Hydro Polygons。

基于数据库的权限组织数据存取

有时，用户使用要素数据集来组织数据存取优先权。包含在一个要素数据集中的所有要素类具有相同的访问优先权。例如，用户可能需要不至一个的要素数据集分割一系列相关的要素类来解决用户之间的不同访问权限。每个组有对其中一个要素数据集和它的要素类的编辑访问权限，但对其他的则没有权限。

为数据共享组织要素类

有一些数据共享的情况，相互合作的组织可能与其他用户在共享数据集的共享模式上达成了一致意见。在这种情况下，用户可能将要素数据集作为一个文件夹来组织简单要素类的集合，以与其他人达到共享。

预先指定坐标系统

组织要素类到相同要素数据集的另一个设计原因就是使用一个空间参考的需要。因此，在组织要素类到相同要素数据集前，先定义每个要素类的坐标系统是很有意义的。

当建立一个新的数据集时，必须定义它的空间参考。这包括坐标系统－地理或者具体的投影系统－以及坐标单位和 xy、z 和 m 值的容差。在一个相同要素数据集中的所有要素类必须共享一个共同的坐标系统，要素的 xy 坐标应该落在一个相同的空间范围中。

4、4、3、2 创建一个要素数据集

建立一个要素数据集包括三个基本任务：

首先，建立一个新的要素数据集

添加即将存储到要素数据集中的要素类集合，装载它们的内容（如添加要素）

添加可选的 geodatabase 对象到要素数据集来模拟关联和添加丰富的行为。可选项包括添加一个关联类、一个拓扑、一个网络数据集、一个表面或一个几何网络。

在 ArcCatalog 中创建一个新的要素数据集

以下是建立要素数据集的步骤：

1、 在 ArcCatalog 目录树中，鼠标放在需要建立新要素数据集的 geodatabase，单击右键

2、 鼠标指针指向 New，单击 Feature Dataset，打开 Feature Dataset 对话框

3、 在 Name 窗口输入要素数据集名称

4、 移动鼠标至需要使用的空间参考。或者，单击 Import，移动鼠标至要素类或要素数据集，需要使用它们的空间参考作为一个模板。

5、 如果需要修改所选择坐标系统的参数，单击 Modify。编辑所选择的坐标系统参数，单击 OK。

6、 如果数据需要一个垂直坐标系统，可以从其他要素类或要素数据集导入一个；或者，选择 None。如果需要修改所选择坐标系统的参数，单击 Modify。编辑所选择的坐标系统参数，单击 OK

7、 输入 xy-tolerance, z-tolerance 和 m-tolerance 的值，或接受默认值，它相当于真实世界中 1mm。

Xy－、z－和 m－tolerances 用于设置坐标被允许在各种处理操作中调整的距离。

默认的 x、y－tolerances 被设置为 0.001 meters。例如，如果坐标系统是被记录为 feet，默认值是 0.003281 feet (0.03937 inches)。如果 x、y 坐标是被存储在 decimal degrees 默认值就是 0.0000000556。

在大多数情况下，使用默认值运行的及其良好，除非有一个极端情况，有令人难以置信精度的数据（如坐标精确性高于测量等级的坐标）。

如果已经修改了 xy-, z- and/or m-tolerances，想恢复默认值，单击 Reset all tolerances button。

8、 如果取消选择 Accept default resolution and domain extent 复选框，在向导中会有另外一个对话框允许你修改 xy-, z- and m-resolution，以及最大和最小的 z- 和 m-values。

9、 单击 Finish。

4、4、4 域

4、4、4、1 属性域概要

属性域是描述一个字段类型的合法值的规则，提供了加强数据完整性的一种方法。属性域用于限制在表、要素类的任何具体的属性字段内允许的值。如果一个要素类中的要素或一个表中的非空间对象被组织到一个子类型中，不同的属性域可以被分配到每个子类型。一个域就是可接受属性值的一个声明。无论何时，当一个域与一个属性字段建立联系后，只有在这个域内的值对这个字段才是有效的。换句话说，这个字段不会接受不在这个域内的一个值。使用域可以通过限制一个具体字段的值选择来保证数据完整性。

属性域可以在 geodatabase 的要素类、表和子类型之间共享。

例如，一个 water mains 要素类和 water laterals 要素类可以对 ground surface type 字段使用相同的域。

域属性

当建立或修改一个域时，必须编辑以下的属性：

1）Name and description

当建立一个新域时，指定一个名字，它将描述它所管理的参数。

当一个域建立后，当打开一个要素类或表的属性对话框，当选择一个域名关联一个给定字段时，这个域名将会在 domain 下拉框显示。

这是对域目的一个简小描述。

2）Field type

字段类型是属性字段的类型，域可以与属性字段建立联系。

字段类型可以设置为以下类型中的一种：

- Short—短整型
- Long—长整型
- Float—单精度浮点数类型
- Double—双精度浮点数类型
- Text (Coded domains only)—文字数字式字符
- Date—日期和时间数据

一旦字段类型被建立，域的名字会在属性对话框中的下拉列表中这种字段类型的任何字段上出现。

3）Domain type

当建立一个域时，必须指定需要使用那种类型的域。

有两种类型的属性域：

Range domains—一个范围域是数值属性指定值的有效范围。当建立一个范围域，输入一个最大和最小的有效值。范围域可以应用于 short-integer, long-integer, float, double, 和 date 类型。

例如，在 water mains 要素类中，可以有输水管、配水管和支路水管的子类型。配水管可以承受的压力范围为 50－75psi。对于一个有效的配水管，它输入的压力值必须在 50－75psi 之间的一些值。

范围值在编辑时被自动验证。

Coded domains—一个代码值域可以应用于任何属性类型，包括文本、数字、日期等。代码值域给一个属性指定有效的取值集合。

例如，水管可能被埋在不同类型的土地表面下，表示为 GroundSurfaceType 字段：pavement, gravel, sand, 或 none。代码值域包括两个值，一个是存储在数据库中的实际值（如代表 pavement 的 1），一个是说明数值意义的用户描述。

4）Split and merge policies

在编辑数据时，常常需要把一个要素分割成两个要素，或把两个要素合并成一个的要素。如，在一个土地数据库中，一个地块可能由于再分区被分割为两个独立的地块。相似的分区变化可能需要两个相邻的地块合并为一个地块。

分割和合并在要素的几何操作结果是容易理解的，但是其属性操作结果则不那么容易。当一个要素被分割时，属性值的分割由分割策略来控制。当两个要素合并时，属性值的合并由合并策略来控制。

每一个属性域都有分割和合并策略，当一个要素被分割或合并时，geodatabase 根据这些策略，决定其结果要素属性取值。

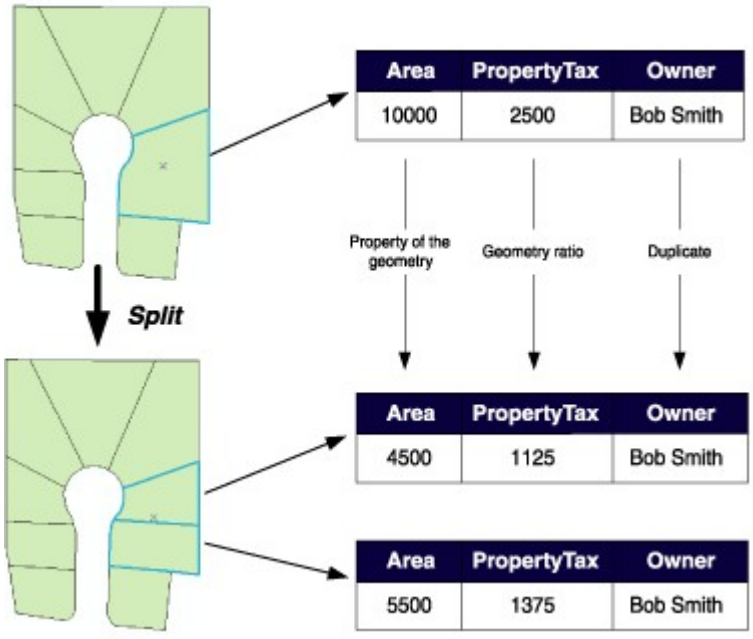
Split policies

对于一个给定的表、要素类或子类型，可以由下列 3 个分割策略中的任何一个，来控制分割结果要素的属性取值：

Default value（默认值）—分割结果要素的属性采用给定要素类或子类型属性的默认值。

Duplicate（复制）—分割结果要素的属性采用原始对象属性的拷贝值。

Geometry ratio（几何比例）－分割要素的属性值是对原始对象属性值的比例计算值。属性值计算比例是基于原始要素被分割的比例。如果对象被均等的分割，则每个分割要素获得原始对象属性值的一半。几何比例策略仅仅用于数字型字段。



在上面地块的例子中，当一个地块被分割时，Area 属性将作为结果要素的一个属性被自动分配。Owner 的值被拷贝到新要素（在这个数据库中，分割一个地块并不影响它的物主权）。PropertyTax 根据一个地块的面积、大小来计算。为了给每个新要素计算 PropertyTax，分割策略在新要素中根据它们的面积来按比例地分割原始地块的 PropertyTax。

Merge policies

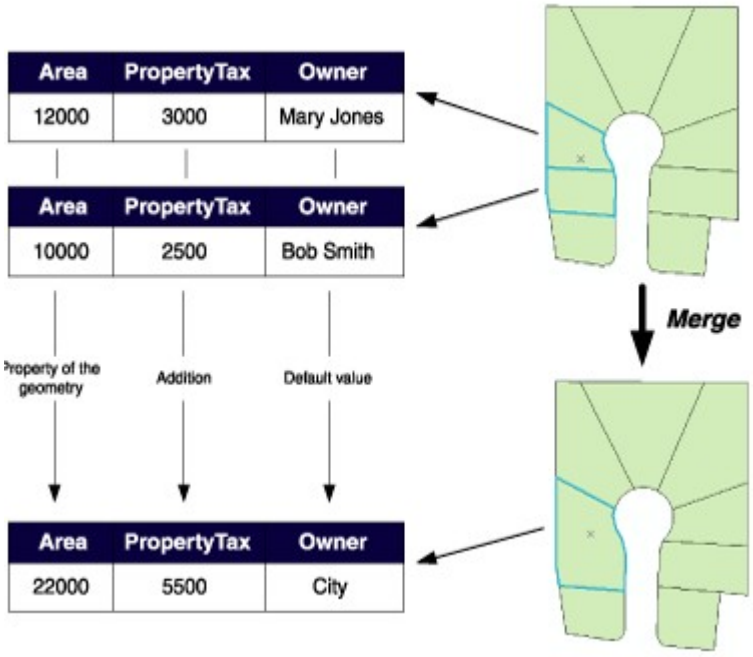
当两个要素合并成一个新的要素时，合并策略控制新要素的属性值。对任何给定的要素类或子类型的属性值，可以由下列 3 个合并策略中的任何一个确定：

Default value（默认值）—合并结果要素的属性采用给定要素类或子类型属性的默认值。

默认值策略仅仅用于非数字类型字段或代码值域属性。

Sum values（和值）——合并结果要素的属性采用原始要素属性值的和。

Geometry weighted（几何权重）－合并结果要素的属性是原始要素属性值的权重平均。这种平均是根据原始要素的几何比例的。



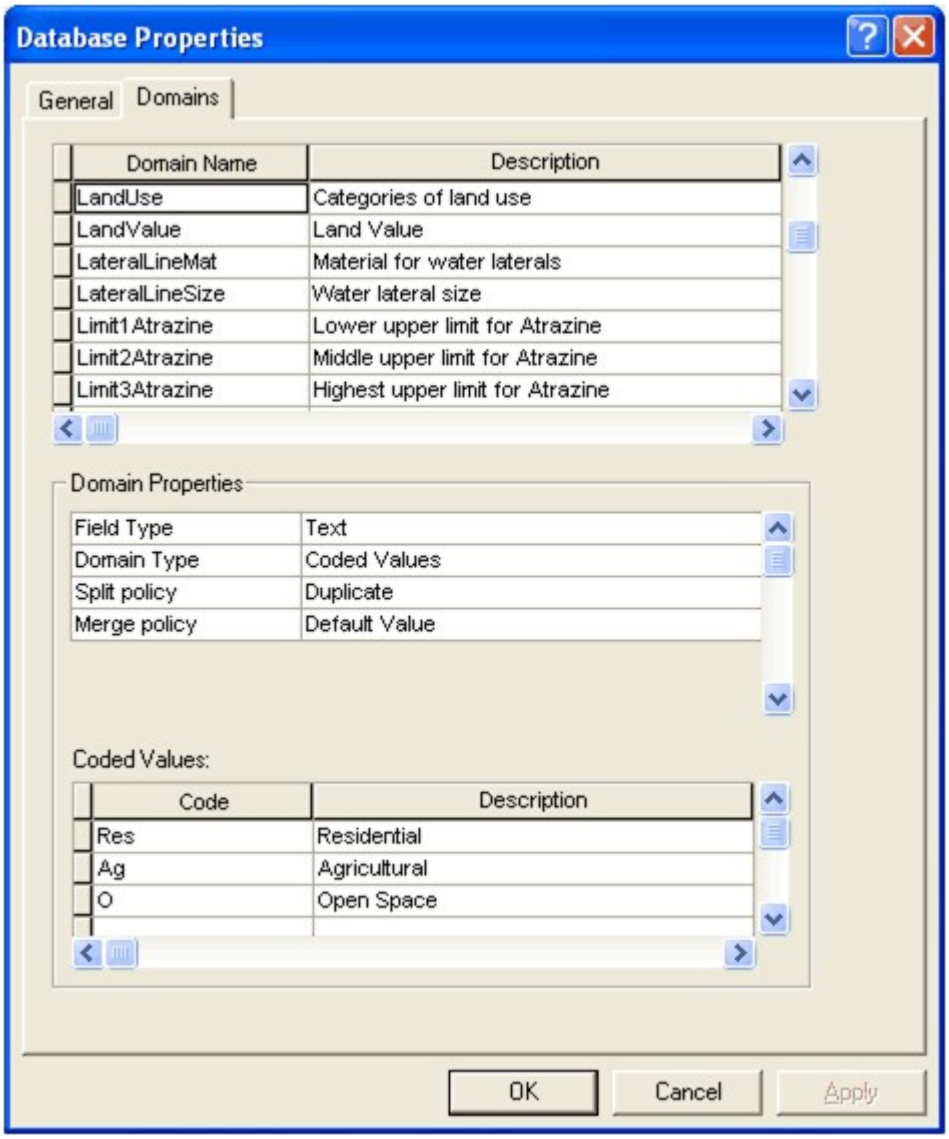
在上面的地块例子中，当两块地被合并时，Area 属性将作为结果要素的一个属性被自动分配。Owner 被赋值为默认值。合并结果要素的 PropertyTax 属性是原始要素的 PropertyTax 属性值的和，它的合并策略是取和值。

注意：当在编辑器中合并要素时，合并策略是不能被评价的。但是，开发人员在写自己的合并实现程序时，可以利用合并策略。

5）Coded values

代码值部分仅是对代码域可用。它包含一个域的代码值，以及说明这个数值意义的相关表示。

下面的图显示了使用缩略字作为代码值来表示一个要素类型的一种方法。在这个例子中，土地利用类型使用合适的缩略字来表表达。



当输入代码值，这个代码必须与相联的字段类型匹配。在上面的例子中，不能输入一个整型来表达居民地类型，因为土地使用域是与文本数据类型相关联的。

4、4、4、2 浏览 geodatabase 的属性域

关于浏览属性域

属性域存储在 geodatabase 中。一旦一个用户建立一个新属性域，那么所有的用户可以浏览这个域的属性，在一个要素素或表中使用这个域。

属性域使用域属性页来管理，它可以作为 geodatabase 的属性的一部分被显示，或者在要要素类、表的属性对话框中显示。

如何浏览属性域

浏览 geodatabase 的域

1、 在 ArcCatalog 目录树中，鼠标放在需要浏览的包含域的 file、personal 或 ArcSDE geodatabase 上，单击右键。

2、 单击 Properties 命令。打开 Database Properties 对话框。

在要素类或表中浏览 geodatabase 属性域

1、 在 ArcCatalog 目录树中，鼠标放在需要浏览属性域的要素类或表上，单击右键。

2、 单击 Properties 命令。

3、 单击 Subtypes 标签。

4、 单击 Domains，打开 Domains properties 对话框。

4、4、4、3 建立新的属性域

关于建立属性域

在 geodatabase 的生命周期，随时都可以借助 Domains properties 对话框来建立新的属性域。

可以建立新的属性范围域，或建立代码值域。

如果一个要素类含有子类型，属性域必须应用在子类型级别，而不是字段级别。

如何建立新的属性域

建立新的属性范围域

1、 在 ArcCatalog 目录树中，鼠标放在需要建立属性范围域的 geodatabase 上，单击右键

2、 单击 Domains 标签，

3、 单击 Domain Name 栏下的第一个空白字段，输入新属性域名。

4、 按 Tab 键，或单击新属性域的描述（description）字段，输入描述。

5、 在 Domain Properties 栏内单击 Field Type 项右边的下来箭头：选择属性域所应用的属性字段类型。

6、 单击 Domain Type 右边的下拉箭头，选择 Range 类型。

7、 对于范围域：单击最小值字段，输入最小值，单击最大值，输入最大值。

8、 单击分割（Split policy）字段右边的下拉箭头，选择分割策略或者合并策略。

9、 单击 Apply 按钮，建立新属性域范围，或单击 OK 按钮建立新的属性域范围，关闭对话框。

提示：

范围域不能应用于文本字段。它仅能应用于数字或日期字段。

建立新的代码值域

1、 在 ArcCatalog 目录树中，鼠标放在需要建立属性代码值域的 geodatabase 上，单击右键，单击 Properties 命令。

2、 单击 Domains 标签。

3、 单击 Domain Name 栏下的第一个空白字段，输入新属性域名。

4、 按 Tab 键，或单击新属性域的描述（description）字段，输入描述。

5、 在 Domain Properties 栏内单击 Field Type 项右边的下来箭头：选择属性域所应用的属性字段类型。

6、 单击 Domain Type 右边的下拉箭头，选择 Coded Values 类型。

7、 在 Coded values 选项卡，单击 Code 列第一个空白字段，输入新代码值。

8、 按 Tab 键，或单击新代码值右边的描述字段（Description），输入代码值的描述

9、 重复上面两个步骤，直到所有的有效代码值及其描述都被输入。

10、单击分割策略（Split policy）字段右边的下拉箭头，选择分割策略或者单击合并策略字段。

11、单击 Apply 按钮建立新的属性域的代码值范围，或单击 OK 按钮建立新的属性域的代码值范围，关闭对话框。

提示：

当添加一个新的值到一个域的代码值列表中，必须以添加用户描述。当编辑一个含有这个域的字段的属性值，这个描述值会出现在 ArcMap 的编辑器中。描述值帮助你选择正确的值。

代码值域仅支持默认值和复制分割策略。

代码值域仅支持默认值合并策略。

4、4、4、4 修改和删除属性域

关于修改属性域

在属性域属性对话框中，可以进行属性域删除或修改操作。

当一个新的属性域建立时，属性域的拥有者－也就是建立属性域的用户将被记录在数据库中。只有属性域的拥有者才能删除和修改属性域。

属性域可以与要素类、表、子类型的特定字段关联。当一个属性域被一个要素类或表应用时，就不能被删除或修改。

修改属性域可以简单通过在属性域对话框中选择它们，修改从域名到类型、有效值的任何东西。它可以按照建立一个新域的相同方式去做。

修改子类型的域必须在要素类属性对话框中的 subtype 标签中完成。

如何修改和删除属性域

1、 在 ArcCatalog 目录树中，鼠标放在需要修改属性域的 geodatabase 上，单击右键，单击 Properties 命令。

2、 单击 Domains 标签。

3、 在 Domain Name 栏下确定要删除属性域，单击域名左边的标签。

4、 在键盘上按 Delete 键，删除所标识的属性域。

5、 单击 Apply 按钮删除属性域，和单击 OK 按钮删除属性域关闭对话框。

提示：

只有属性域的拥有者才能删除和修改属性域。

4、4、4、5 关联属性域

关于关联属性域

一旦建立了一个属性域，就可以将其和其默认值与表或要素类的字段关联起来。属性域同一个要素类或表建立关联以后，在 geodatabase 中一个属性有效规则就被建立起来了。

同一个属性域可以与同一个表或要素类或子类型的多个字段关联，也可以与多个要素类或多个表的多个字段关联。

如何建立关联属性域

1、 在 ArcCatalog 目录树中，鼠标放在需要关联属性域的表或要素类上，单击右键。

2、 单击 Properties 命令。

3、 单击 Fields 标签，进入 Fields 选项卡。

4、 在 Fields Name 栏，单击需要建立默认值并把它关联到属性域的字段。

5、 在 Field Properties 栏，单击 Default value 右边的单元格，输入默认值。

6、 如果不需要关联一个域到这个字段，跳跃至步骤 9。

7、 单击 Domain 右边的下拉箭头，选择想用于这个字段的属性域

仅仅可以用于这个字段类型的域才能显示在列表中。

8、 重复步骤 4 到 7，直到将默认值与需要具有这些属性的所有字段都建立了关联。

9、 单击 Apply 按钮。

提示：

在一个表或要素类中，并不是所有对象都必须有相同的属性域或默认值应用到相同的字段。为了应用不同的属性域和默认值到一个独立的表或要素类的相同的字段，必须建立子类型。

4、4、5 子类型

4、4、5、1 子类型概要

子类型是一个要素类中的要素子集，或一个表中的对象子集，它们共享相同的属性。它们作为数据分类的一种方法被使用。

子类型允许执行如下的操作：

- 提高 geodatabase 性能，通过表达多种真实世界的对象为一个给定要素类的子集，而不是为每个对象建立一个新要素类。

例如，街道在街道要素类中可以被分类为三种子类型：*local streets, collector streets,和 arterial streets*。

- 设置一个默认值，它会在建立新要素时自动应用。

例如，*local street* 子类型可以被建立和定义，所以任何时候这个子类型添加到要素类时，它的速度限制属性会自动设置为 35 英里/时。

- 应用代码值域和范围值域到要素可以限制输入信息到一个值的有效集合。

例如，一个水分布网络，子类型水管道可以有一个建筑材料的代码值域来限制它们是由 *cast iron, ductile iron, 或 copper 组成*。

- 建立其他子类型和要素类之间的连接规则来维护网络的完整性。

例如，一个水分布网络，一个 *hydrant* 可以连接到一个 *hydrant lateral*，而不是一个 *service lateral*。

- 建立其他子类型和要素类之间的拓扑规则。

例如，除了在街道属于死胡同子类型的情况下，可能需要街道要素必须与其他街道要素在两者的末端相连接。

- 建立其他子类型、表和要素类之间的关联规则。

例如，在一个电力网络中，可以在铁支杆类的 A 变压器和木支杆的 B 变压器两个子类型之间建立关联规则。

- 编写代码在不同要素之间建立定制规则。

子类型可以有两种方法建立：

1、 在 ArcCatalog 中使用上下文菜单。

2、 通过 geoprocessing 使用 Subtypes toolset 工具箱。

4、4、5、2 操作子类型

子类型是一个表和要素类中的记录，它已经基于属性字段被分类。子类型通过建立代码值来实现，因此，它必须与短整型或长整型数据类型的字段相联系。例如，下面名为 RoadClass 子类型的代码表示了街道要素类的有效子类。

0 - Local Streets

1 - Secondary Streets

2 - Main Streets

每个子类型可以有其默认值集合。在上面的例子中，local streets 子类型有一个速度限制属性的默认值，设为 25 英里/小时，main streets 子类型有一个速度限制属性的默认值为 35 英里/小时。任何时候当一个 local streets 被添加到这个街道要素类，它的速度限制属性将会自动设置为 25 英里/小时。当一个 main street 被添加，它的速度限制属性将会自动设置为 35 英里/小时。

每个子类型同样对于一个给定的字段有其自己的范围或代码值属性域。例如，对于一个水管道要素类，水压力的范围属性域可以被实现。表达输水管的子类型可以承受的压力在 40－100psi 之间，表达配水管的子类型可以承受的压力在 50－75psi 之间。

每个子类型同样可以关联不同的连接规则、关联规则或拓扑规则。

要素类 vs 子类型

在进行 geodatabase 设计时，需要决定在什么地方适合使用子类型和在什么地方需要添

加要素类。当决定是建立子类型还是新要素类，紧记如下内容：

当需要通过默认值、属性域、连接规则、关联规则区分对象时，就需要对单一的要素类或表建立不同的子类型。

当根据不同的行为、属性、访问权限或对象的多版本来区分对象时，必须建立另外的要素类。

4、4、5、3 建立子类型

关于建立子类型

可以使用 ArcCatalog 来给要素类添加子类型，并为每个子类型设置默认值和属性域。

可以使用 geodatabase 的一个要素类或表的属性特征对话框，来管理每一个表或要素类的子类型。可以定义子类型字段、添加新的子类型，删除或修改已经存在的子类型。

如何建立子类型

1、 在 ArcCatalog 目录树中，鼠标放在需要添加子类型的表或要素类上，单击右键。

2、 单击 Properties 命令。

3、 单击 Subtypes 标签，进入 Subtypes 选项卡。

4、 单击 Subtype Field 右边的下拉箭头，单击可用的长整型字段列表中一个子类型字段。

5、 为了添加一个新子类型，单击 Code 栏下的第一个空白字段，输入新的子类型代码（整型数）。

6、 按 Tab 键或单击 Description 字段，输入新建立子类型的描述。

7、 对于每个字段，在 Default Values 下面的单元格输入默认值。

8、 为了将一个属性域关联到新子类型的字段，单击 Domain 下面的下来箭头，从列表中选择一个属性域。

仅仅可以用于这个字段类型的域才显示在列表中。

9、为了设置新建立的子类型为该要素类的默认子类型，单击 Default Subtype 窗口的下来菜单，选择新的子类型名。

10、重复步骤 5 到 8，添加其他子类型，可以在任何时候设置默认子类型。

11、添加一个新子类型，单击对话框左下方的 Use Defaults 按钮，让新子类型采用默认子类型的所有默认值和属性域。然后再修改它们中的一些。

12、当完成了建立子类型，选择默认子类型，单击 Apply 按钮建立新子类型，或单击 OK 按钮建立新子类型，关闭对话框。

提示：

子类型必须时长整型或短整型字段。如果没有可选子类型字段，不能添加子类型。

默认子类型提供两种目的：1）当建立一个新子类型时，单击 Use Defaults 按钮，子类型将继承默认子类型的所有默认值和属性域，然后修改满足新子类型的要求。当添加子类型时，默认子类型可以在任何时候被修改。2）当在一个没有指定子类型的要素类中建立新要素时，它会自动被指派为默认子类型。

4、4、5、4 修改和删除子类型

关于修改和删除子类型

可以用 geodatabase 中的一个要素类或表的属性对话框，来修改或删除一个要素类或表的子类型。可以修改一个子类型的任何方面包括它的描述、默认值和属性域。修改一个子类型的每个方面和建立一个新子类型是一样的方法。

如何修改和删除子类型

1、 在 ArcCatalog 目录树中，鼠标放在需要修改子类型的要素类或表上，单击右键。

2、 单击 Properties 命令。

3、 单击 Subtypes 标签，进入 Subtypes 选项卡。

4、 在 Subtypes 栏下确定要修改或删除的子类型，单击子类型名前面的标签。

5、 进行更改或按 Delete 键。

6、 单击 Apply 按钮或担架 OK 保存修改。

4、4、6 关联类

4、4、6、1 关联和 ArcGIS（仅对 ArcInfo 和 ArcEditor）

一个 GIS 集成空间的和非空间类型的实体信息，它们中许多可以相互关联。

- 空间实体和空间实体之间的关联。例如：一个建筑物可以与一个地块相关联。
- 空间实体和非空间实体之间的关联。例如，一个地块和其拥有者相关联。
- 非空间实体和非空间实体之间的关联。例如，一个地块拥有者和一个免税代码相关联。

ArcGIS 提供很多方法在一个 geodatabase 中建立要素和记录之间的相互关联关系。当在地理要素之间建立关系时，第一步就是模拟不同要素之间的空间关系。考虑可以如何使用 geodatabase 的拓扑、几何网络、共享边的编辑、编辑环境中几何图形捕捉和地理空间操作符。这些方法可以帮助有效地建立和维护数据。

如果需要建立关系很紧密，但在空间方面又有模糊的关系的要素之间的关系，不能仅仅使用空间关系。例如，一个垫式变压器可以为几栋建筑物提供电力服务，但是除非有第二条线被绘制，否则变压器和它服务的建筑物集合之间的关系就不清楚、明确了。

另外一个不能仅建立空间关系的情况就是关系包括一个非空间实体，如一个或多个地块拥有者。

对于这两个普通例子，可以在关联类中建立属性关系，使用 relate，或建立 joins。这些方法提供了一个类（要素类或表）中的记录与另一个类中记录的连接。

4、4、6、2 决定使用 relationship classes、joins、或者 relates (仅对 ArcInfo 和 ArcEditor)

Relationship classes 帮助确保参考完整性。例如，删除或修改一个要素可以删除或修改一个相关的要素。甚至，一个 relationship class 被存储在 geodatabase 中，它可以使这个 geodatabase 的任何用户都能访问。

On-the-fly relationships 也被称为 relates，被定义为一个 ArcMap 图层的一个属性。使用它们用于提高编辑性能。

Joins 最适合标注和符号化。在关系数据库中定义 joins，进行跨数据库及不同数据源的标准 SQL 查询。

4、4、6、3 relationship classes 的优点 (仅对 ArcInfo 和 ArcEditor)

Geodatabase 中的关系类与在数据库管理系统中建立的关系是相似的。关系类管理一个类中（要素类或表）的对象与另一个类中对象的关系。在关系的任何一端的对象可以是几何要素，或是表中的记录。

关系类支持各种基数－一对一、一对多和多对多，还可以有关于关系本身的属性。关系类提供了许多高级功能，但在 ArcMap 中 joins 和 relates 中却没有这些功能。

- 关系类有助于加强参考完整性。

当修改一个对象时，一个关系类可以建立，相关的对象自动更新。这包括物理上移动相关要素，删除相关对象或更新一个属性。例如，可以建立这样一个关系，即当任何时候移动一个 utility pole，attached transformers 和其他设备也会跟着移走。

关系类主动地维护相关类之间的参考完整性，即使它们中一个还没有被添加到 ArcMap session 中。

- 关系类有利于编辑，帮助降低维护成本。

通过提供自动更新相关对象，一个关系类可以避免执行另外的编辑操作。

关系类可以帮助在编辑的时候访问对象。可以选择一个对象，然后使用属性对话框或表

来查找相关对象。一旦已找到相关的对象，可以编辑其属性。不管相关联的链条有多深，所有相关联类都可以用于编辑。

由于关系类存储在 geodatabase 中，它们可以被管理为版本。版本允许多个用户在同一时间编辑一个关系中要素或记录。

- 关系类允许查询相关要素和记录。与 ArcMap join 相似，可以查询，执行分析和用关系类中的属性生成报表。

4、4、6、4 关系类属性（仅对 ArcInfo 和 ArcEditor）

一个关系类包含了一些熟悉，它们定义了源类中的对象如何与目标类中对象相关联。当建立关系类时需要指定以下属性。

- 类型：简单或复合
- 源类和目标类
- 主关键字和外关键字
- 基数：是一对一，一对多还是多对多关系？
- 消息通知方向：如果需要执行定制级联更新或删除行为时是可用的
- 是否需要存储每个关系的属性
- 名字
- 向后和向前标注

一旦已经建立了关系，可以指定规则精制基数。

简单 vs.复合

当建立一个关系类时，指定它是简单关系，还是复合关系。

简单关系中，相关对象可以相互独立存在。例如，在一个铁道运输网络中，铁路交叉口可以有一个或多个信号灯。但是，没有信号灯铁路交叉口也可以存在，信号灯也可以存在于一个没有交叉口的铁道运输网络中。

当在一个简单关系中删除一个源类，对于匹配的目标类的关键字的值被设置为 Null。删除一个目标类不会影响源类。

简单关系可以有一对一、一对多、多对多的基数。

与简单关系相同，当对象被删除时，复合关系也维护参考完整性，但是它们采用不同的方法。一个复合关系中，目标类不能独立于源类而存在，因此当源类被删除时，相关的目标类在一个被称为级联删除过程中也被删除了。

这个从属规则也被 ArcMap 的 Validate Features 命令强化－－这个命令是在一个编辑会话中运行来检查参考完整性。如果建立了一个目标类，但是还没有与源类关联，Validate Features 可以警告发生错误。

一个复合关系同样也帮助用户从空间完整性上维护要素；移动或旋转一个源要素会引起目标要素也跟着它移动或旋转。

复合关系通常是一对多，但也可以通过关系规则限制到一对一。

源类和目标类

当建立一个关系类时，选择一个类为源类，另一个为目标类。做到不混淆这两种类是很重要的。在复合关系中的级联删除行为中，这种重要性可能更明显。

在简单关系中，保证源类和目标类的正确性是关键的。这是因为当删除源类中的一条记录时，简单关系类查找目标类中相匹配的记录设置它们的关键字段的值为 Null。如果选择错误的源类，删除源类中的对象，会将错误引入到外关键字字段。下面的例子解释了这种错误是如何发生的。

Case1、Parcel to Zone（wrong）

这是常见的错误情况。Zone 表包含不同区域编码的具体描述，概念与 ArcInfo

Workstation 中的查找表相似。在这种情况下，Parcel 类是源类，Zone 类是目标类。这就是如何在 ArcInfo Workstation 中设置关系。问题就是，当删除一个 parcel，为了匹配 Zone 表中的记录，Zone 表中的关键字段的值被设置为 Null，没有一个含那个区域编码的 parcel 表在 Zone 表中有一个匹配。

Case 2: Zone to Parcel (correct)

为了解决这个问题，设置 Zone 表为源，删除一个 parcel（一个目标对象）将对 Zone 表没有影响，删除一个 Zone code（一个源对象）将简单设置匹配的 parcel 记录中的 Zone 字段值为 Null，这是应该的，因为它们不再有匹配的 Zone 表记录。

主关键字和外关键字

在一个关系类中，源类中的对象匹配目标类中是通过它们关键字字段的值。在下面的例子中，parcel 789 匹配 permits 2 和 3，因为这几个记录都含有相同的 parcel ID。

一个关系中源类中的关键字段称为主关键字，通常简写为 PK。不同与真正的主关键字，在一个关系中主关键字段的值不要求对每个对象是惟一的。

在目标类中的关键字段称为外关键字，通常简写为 FK。它包含匹配与源类中主关键字段的值。同样，这个关键字段的值不要求对每个对象是惟一的。

关键字段可以有不同的名字，但是必须是相同的数据类型，包含相同类型的信息，如 parcel IDs。除了 binary lage object (BLOB), date 和 raster，所有的数据类型的字段都可以是关键字段。当建立一个关系类时指定关键字段。

当决定采用哪个为主关键字段，方法之一就是使用行 ID 字段，通常称为 ObjectID 字段。当建立一个新的要素类或表或注册一个 ArcSDE 层或表时，ArcGIS 会自动添加 ObjectID 这个字段的。这个字段保证每条记录都惟一有一个 ID。它是由 ArcGIS 来维护的，用户不能修改它。

只要给定对象的 ObjectID 值存储在源类中，它的值就始终不会改变。如果这个对象是一个要素，不能分割它。如果分割它，一个新的 ObjectID 会被分配，破坏了它依赖的任何关系。

基于这个原因，建立和使用自己的主关键字段比依赖于 ObjectID 字段更好。下面描述了在执行这些操作时如何使用自己的主关键字来维护关系。

- 当导入记录到另一个要素类和表，新的 ObjectID 值会被分配，将会失去基于源 ObjectID 值的任何关系。如果，相反，建立基于另一个主关键字段关系，当导入记录时当主关键字段的值将不会改变。这允许在导入相关对象集合到新类时存储关系。

使用 copy/paste 功能时是一个特列。copy/paste 存储 ObjectID 值，因此如果打算使用这种方法来移动对象，可以使用 ObjectID 字段作为主关键字段。

- 当分割一个要素时，源要素类从要素类中被删除，两个新的要素被建立。结果，源 ObjectID 值被删除，两个新的 ObjectID 值被创建。如果有基于源 ObjectID 的一个关系，源要素和相关表的记录之间的关系将会被丢失，因为源要素已经不存在。但是，如果使用另一个字段作为主关键字段，当分割要素时，源要素的 ObjectID 值将会被拷贝到这两个新要素中。结果，相关表中的记录将会与这两个新要素相关－关系类使被建立为多对多。

如果不分割要素，一定要做到所有的对象存储在源类中，可以使用 ObjectID 作为它们的 IDs。如果不能保证这点，最好建立和使用自己的 ID 字段，不要依赖于 ObjectID 字段。

当合并两个要素时，新要素保留源要素中的其中一个的 ObjectID。如果打算合并要素，但并不从它们的类中移走要素或分割要素，可以使用 ObjectID 作为它们的主关键字。

基数

关系的基数描述了源类中有多少个对象与目标类中的对象关联。一个关系可以有以下三种基数中的一种：

一对一：一个源类对象仅能与一个目标类对象关联。例如，一个地块仅能有一个合法的描述。在 ArcGIS 中，这种基数同样适用多对一。一个多对一关系的例子就是多个地块相关于相同的合法的描述。

一对多：一个源类对象可以与多个目标类对象关联。例如，一个地块上有多个建筑物。在一对多的关系中，一的一边必须时源类，多的一边必须是目标类。

多对多：一个源类对象可以与目标类中的多个对象关联，相反，一个目标类对象可以与多个源类对象关联。例如，一个特定财产可能有多个业主，一个特定的业主可以有多个财产。

术语一和多可以被误解。一实际是零对一，多实际是零对多。因此，当建立地块和建筑物之间的一对多关系时，例如，这个关系允许下面所有的形式。

- 一个地块，零个建筑物
- 一个建筑物，零个地块
- 一个地块，多个建筑物

建立一个关系后，通过定义关系规则控制关系类型的基数。可以建立规则来指定源类中的哪些对象被允许与目标类中的哪些对象相关联。

Relationship rules

当建立一个关系类时，可以创建它的基数为：一对一，一对多，或多对多。

关系通常需要多个限制性的术语来定义。在地块和建筑物的关系中，例如，可能需要每个建筑物与一个地块相关联，或者一个地块包含一个最大数量的建筑物。需要防止用户忘记一个建筑物与一个地块相关联，或者多个建筑物与一个地块相关联。

如果有子类型，可以控制源类中的对象数量和类型，它与目标类中某一确定的对象类型相关联。例如，铁支杆类的 A 变压器，木支杆的 B 变压器。此外，可能需要为每个有效的子类型对指定允许的基数范围。例如，一个铁支杆能支持 0－3 级别的 A 变压器，一个木支杆能支持 0－2 级别的 B 变压器。

一旦已经建立了一个关系类，可以指定规则来强化参考完整性规则：

1、 在 ArcCatalog 中，在一个已存在的关系类上，单击右键，打开它的 Relationship Class Properties 的对话框，单击 Rules 标签。

2、 从源类中选择一个子类型，选择与源类中被选的子类型相关的目标子类型。

3、 选择 origin cardinality 和 destination cardinality 选项卡上的复选框。为这个规则设置合适的最小或最大的基数值。这个对话框防止设置一个大于最大值的最小值。因此先设置最大的基数值。

一旦一个关系规则被添加到一个关系类，那么这个规则就成为现存的惟一有效的关系。为了使其他关系结合，基数有效，必须添加其他的关系规则。

在下面的例子中，一个 HazMat landfill 可以与一个或两个 deep wells 相关联，或两个到七个 shallow wells 相关联。但是，如果一个 sanitary landfill 与一个 deep well 相关联，但是在这两个子类型之间没有建立规则，Validate Features command 将考虑这个关系的有效性。

建立这个规则后开始编辑，可以用 ArcMap Validate Features command 来测试它们。Validate Features command 命令会告诉当前被选择的要素是否违反关系规则。

Message notification direction

正如上面讨论的，当删除一个复合关系中的源类对象，相关的目标类对象会被自动删除。

不管操作简单关系，还是复合关系，可能需要要素更新的其他操作，这会引起它相关要素的更新。甚至，更新可以被要求在一个方向或另一个方向或两个方向。

- 当移动或旋转一个要素时，相关的要素也需要跟着移动或旋转。
- 当更新一个要素时，需要相关要素中的属性表也自动更新。
- 更新一个源类对象要求相关的目标对象也被更新。

- 更新目标对象要求相关的源类对象也被更新。

如果关系需要这种行为，可以让源和目标对象相互发送消息通知它们已经改变，允许相关的对象相应地更新。

为了完成这个，建立关系时设置消息通知方向。如果更新一个源对象要求相关目标对象也更新。设置消息通知方向为 Forward（向前），如果更新目标对象要求相关源对象也更新，设置消息通知方向为 Backward（向后）。如果需要两个都这样做，设置消息通知方向为 Both。一旦建立了关系，必须把这种行为赋给对象，它们才能收到消息，做出响应。

惟一的例外就是当复合关系的消息通知方向设置为 Forward。当建立一个 Forward 的复合关系时，移动或旋转源对象会引起相关目标要素自动跟着移动或旋转。假定设置了正确的关系，一旦建立了关系这种功能就会有效－不需要自定义编程。

对于另一种消息通知方向，自定义编程是必需的。除非已经建立了一个 Forward 的复合关系，或打算编程实现自定义行为，设置消息通知方向为 None。否则，每次执行一个编辑操作，消息都会不必要的生成，造成低性能。

当设置复合关系的方向时，紧记：复合关系中的源对象被删除时，目标类中的所有对象将自动被删除。不管消息通知方向设置为 Forwar、Backward、Both 或 None，这种情况都会发生。

Many-to-many relationships

在一对一和一对多关系中，源类中主关键字段的值直接与目标类中外关键字段的值相关联。

另一方面，多对多关系需要使用一个中间表类存储关系。结果，当建立一个多对多关系时，一个中间表自动建立。这个中间表存储来自源类的主关键字和来自目标类的外关键字。每一行关联一个源对象和一个目标对象。

当中间表建立时，仅有字段被生成。ArcGIS 并不知道哪些源对象与哪些目标对象相关联，因此必须在 ArcMap 中手动建立这些对象。生成这个表是建立关系中最耗费时间的一个部分。

Relationship attributes

多对多关系中的中间表可以提供另一个目的－存储关系本身的特性。例如，在一个地块数据库中在地块和拥有者之间可以有一个关系类，拥有者拥有地块，地块被拥有者拥有。每个关系的特性可以是用有权的百分比。如果需要存储这样一个特性，可以在建立这个关系时或建立后的任何时候添加它们到中间表。

尽管用处不大，当建立一对一或一对多关系时，也需要存储关系的特性。如果是这种情况，在建立关系时必须指定它，因此一个中间表才会被建立。对于多对多关系，中间表存储来自于源类的主关键字和来自于目标类的外关键字，允许存储每个关系的任何特性。

在 ArcCatalog 中可以预览中间表，查看它包含的数据。如果添加关系类到 ArcMap，它会以表的形式出现，可以打开和操作它。ArcGIS 没有为其他操作暴露这个中间表。例如，用户不能在 ArcCatalog 中打开它的属性来添加或删除字段，它也不支持默认值或域的使用。

Name

每个关系类有一个名字显示在 ArcCatalog 目录树中。为了使数据库结构容易理解，名字描述了这个关系。

以源要素类的名字为开头，接着是 Has 或 Have，以目标要素类的名字为结尾。例如，AddressHasZones 或 ParcelsHaveOwners。如果是多对一或多对多的基数，源要素类的名字要以复数形式表示，如果是一对多或多对多的基数，目标要素类的名字也要以复数形式表示。

使用这种方法，可以从它的名字中决定一个关系的基数。例如，两个要素类都是复数形式，ParcelsHaveOwners 暗示一个多对多的关系。

Forward and backward labels

向前和向后标注在 ArcMap 的 Attributes 和 Identify results 对话框中显示，帮助描述相关对象之间的关系。

一个关系类有两种标注：

- 向前标注描述了从源类到目标类的关系。在 pole-transformer 例子中，这个标注为 "supports"，意为这个 pole 支持这些 transformers。
- 向后标注描述了从目标类到源类的关系。在 pole-transformer 例子中，这个标注为 "is mounted on"，意为这些 transformer 被安装在这个 pole 上。

4、4、6、5 关系类的局限性 (仅对 ArcInfo 和 ArcEditor)

一个要素类或表可以参与不止一个的关系类。惟一的限制就是：

- 一个给定要素或表不能是两个复合关系类中的目标类。
- 如果在两个类之间已经存在一个复合关系类，那么在这两个相同的类之间不能再有其他的关系类被建立。
- 必须避免在一个几何网络中要素类之间建立一个关系类，因为这可能导致编辑数据时无法预料的结果。

可以在两个类之间建立多于一个的简单关系。如果需要依据所描述关系方向来执行不同的基数就可以这样做。

在一个相同要素类或表中的记录之间存在一个关系的递归关系在 ArcGIS 中并不被支持。

主关键字和外关键字可以各自仅基于一个字段；复杂的主关键字和外关键字在 ArcGIS 中也不被支持。

为了在一个要素数据集中建立一个关系类，或者源类或者目标类必须是在这个要素数据集中。

在不同的精度方面并没有限制。简单和复合关系类可以被不同精度的要素类组成。

4、4、6、6 创建和维护关系类概要 (仅对 ArcInfo 和 ArcEditor)

建立关系类

在建立一个关系之前，必须在源类中有一个主关键字段，在目标类中有一个外关键字段。

一旦主关键字段和外关键字段被定义，可以使用以下三种工具中的一种来建立一个关系类：

- 在 ArcCatalog 中，鼠标放在需要在其中建立关系类的 geodatabase 或一个要素数据集上，单击右键，鼠标指针指向 New，单击 Relationship Class 命令。使用面板向导设置关系属性。
- 在其中任何一个地理处理框架环境下，可以使用 Create Relationship Class 工具建立一个新的关系类。这个工具允许在一个对话面板中指定和 ArcCatalog 向导中相同的参数。它位于 Create Relationship Class> Relationship Classes toolset。
- 同样在 Data Management Tools toolbox > Relationship Classes toolse 也是一个 Relationship Class 工具。不同于前面两个工具，这个工具允许指定一个已存在的中间表，在一个多对多关系或含有属性的关系中，这是必须的。使用这个工具可以在两个表之间建立一个多对多关系。

如果源类和目标类都存在于同一个要素数据集，这个关系类将位于这个要素数据集中。否则，这个关系类直接位于这个 geodatabase 目录下。

建立一个关系包括几个额外的步骤，有不同的工作流，可以依次完成这些过程。用户所遵循的工作流应该根据数据和需要建立的关系类型来定。

- 如果源类和目标类是由要素或记录组成，并且源类中的主关键字段值与目标类中的

外关键字段值正确关联，完成以下步骤：

1、 使用 New Relationship Class 向导或 reate Relationship Class 工具来建立关系类。

2、 如果需要设置规则。

3、 如果设置规则或操作复合关系，使用 ArcMap 中的 Validate Features command 命令来测试参考完整性。

局限性：这种工作流不能应用多对对关系或含属性的关系。

- 如果源类和目标类是由要素或记录组成，在源类中有主关键字段值，但是或者在目标类中没有外关键字段值，或者源类中的主关键字段值与目标类中的外关键字段值没有正确关联。完成以下步骤：

1、 使用 New Relationship Class 向导或 reate Relationship Class 工具来建立关系类。

2、 如果需要设置规则。

3、 在 ArcMap 中交互式选择相关源对象和目标对象，用 Attributes 对话框将它们关联。

这一步设置可目标类中的外关键字段值，并与源类中主关键字段值相关联。如果在步骤 1 建立的是一个属性关系，这一步添加记录到 Intermediate Key 表。

4、 如果在步骤 1 建立一个属性关系，手动输入属性到 Intermediate Key 表。

5、 如果设置规则或操作复合关系，使用 ArcMap 中的 Validate Features command 命令来测试参考完整性。

局限性：这种工作流不适用于多对多关系，它在源类或目标类中不包括至少一个要素类。

- 为了在两个表之间建立一个多对多关系，或使用一个包含关系属性的已存的 Intermediate Key 表来建立一个关系，完成以下步骤：

1、 建立源表、目标表和 Intermediate Key 表，保证源表中主关键字段与目标表中外关键字段正确相关。

2、 使用 Table To Relationship Class 工具来建立关系类。

建立和维护关系

ArcMap 带有一些工具，可以帮助用户建立和维护一个关系。

如果源类和目标类中都有对象，但它们并没有相关联，可以任何时候在 ArcMap 手动建立。先在目标类中选择一个和多个对象，在源类中选择一个或多个对象，然后打开 Attributes 对话框，将这些对象关联。假定至少关系的任何一边包含要素时，用户才可以这样做。

可以选择一个对象，然后在一个相关类中建立一个相关对象，假定在一个表或一个要素中这是一个新记录。

可以使用 Attributes 对话框从一个关系中移除一个对象。

一旦完成编辑一个复合关系或含规则的关系，可以通过 Validate Features command 来核查你的操作。它会告诉任何违反参考完整性的编辑。

4、4、6、7 创建一个简单的关系类 (仅对 ArcInfo 和 ArcEditor)

关于建立关系类

使用 ArcCatalog 中的工具，在 geodatabase 中的任何要素类或表之间都可以建立新的关系类。这些工具可以用于建立简单关系类、复合关系类和属性关系类。

关系类显示在 ArcCatalog 目录树中，你可以查看它们的属性，以及任何特定要素类的关系。

下面的例子显示了如何在存储地块对象的一个要素类和存储拥有者对象的一个表之间建立一个关系类。这是一个简单、没有属性的关系类。在数据库中，一个地块可以被一个单独的拥有者拥有，一个拥有者可以拥有一个单独地块，因此这是一个一对一的关系。

如何建立一个简单关系类（仅能在 ArcInfo 和 ArcEditor）

1、 在 ArcCatalog 目录树中，鼠标放在需要在其中建立关系类的 geodatabase 或要素数

据集上，单击右键。

2、 鼠标指针指向 New。

3、 单击 Relationship Class。

4、 输入新关系类的名字。

5、 单击源表或要素类。

6、 单击目标表或要素类。

7、 单击 Next。

8、 选择 Simple (peer-to-peer) relationship 单选按钮：建立简单关系类。

9、 单击 Next。

10、输入向后和向前路径标注。

11、选择消息传递方向

12、单击 Next。

13、选择第一个基数选项。在这个例子中，一个拥有者可以拥有一个单独地块，一个地块只属于一个拥有者，因此这是一个一对一的关系类。

14、单击 Next。

15、单击 No。在这个例子中，关系类不需要属性。

16、单击 Next。

17、单击下拉列表箭头，查看源要素类或表的字段列表，单击要素类或表的主关键字。

18、单击下拉列表箭头，查看目标要素类或表的字段的列表。只有与步骤 17 中选择字段类型相同的字段才被列出。单击指向步骤 17 被选的主关键字的外关键字。

19、单击 Next。

20、检查为新关系类指定的选项。如果需要修改，单击 Back 按钮。

21、选项确定后，单击 Finish 按钮，建立新的关系类。

提示：

在这个例子中，没有给关系类添加属性，尽管任何关系类都可以有属性。

多对多关系类需要关系类在数据库中必须有其自己的表。可以随意添加属性到这个表中，或者允许 ArcGIS 来关系表的 schema。

默认情况下，简单关系的消息传递方向是 None。

4、4、6、8 创建一个复合关系 (仅对 ArcInfo 和 ArcEditor)

关于建立复合关系类

可以一个向导来建立一个复合关系类。下面的例子显示了在存储变电站（transformer banks）的一个要素类和存储变电单位（transformer units）要素类之间建立一个关系类。

数据库中 transformer units 的存在是依赖于 transformer bank 的存在。这个关系是一个复合关系，transformer bank 是源要素类。

这个关系将没有属性；复合关系被定义一对多关系。

建立一个复合关系包括很多与建立简单关系相同的步骤。这些步骤反映了两种任务的不同点，包括使用不同的源类和目标类。

如何建立一个复合关系类（仅能在 ArcInfo 和 ArcEditor）

1、 在 ArcCatalog 目录树中，鼠标放在需要在其中建立关系类的 geodatabase 或要素数据集上，单击右键。

2、 鼠标指针指向 New。

3、 单击 Relationship Class。

4、 输入新关系类的名字。

5、 单击源表或要素类。

6、 单击目标表或要素类。

7、 单击 Next。

8、 选择 Composite relationship 单选按钮：建立复合关系类。

9、 单击 Next。

10、输入向后和向前路径标注。

11、选择消息传递方向

12、单击 Next。

13、选择第二个基数选项。一个复合关系根据定义可以是一对多和一对一关系。

14、单击 Next。

15、单击 No。在这个例子中，关系类不需要属性。如果关系类需要属性，单击 Yes。

16、单击 Next。

17、单击下拉列表箭头，查看源要素类或表的字段列表，单击要素类或表的主关键字。

18、单击下拉列表箭头，查看目标要素类或表的字段的列表。只有与步骤 17 中选择字段类型相同的字段才被列出。单击指向步骤 17 被选的主关键字的外关键字。

19、单击 Next。

20、检查为新关系类指定的选项。如果需要修改，单击 Back 按钮。

21、选项确定后，单击 Finish 按钮，建立新的关系类。

提示：

建立一个一对多关系时，不管是简单关系还是复合关系，一的一边必须是源类，多的一边必须是目标类。

4、4、6、9 创建一个属性关系类 (仅对 ArcInfo 和 ArcEditor)

关于建立属性关系类

任何关系类，不管是简单的还是复合的，也不管有什么样的基数，都可以有属性。具有属性的关系类被存储在数据库的表中，这个表至少包括一个指向源要素类或表的外关键字，和一个指向目标要素类或表的外关键字。

具有属性的关系类可以包括多种属性。下面的例子显示了在存储水管（water laterals）的一个要素类和存储水龙头（hydrants）之间建立的一个简单关系。

水管对象有自己的属性，水龙头对象也有自己的属性，而关系类描述了哪个水管供应哪个水龙头。有时还希望存储关系类的一些信息，例如连接两者的材料类型等，可以把这些信息作为属性存储在关系类中。

如何建立一个属性关系类

1、 在 ArcCatalog 目录树中，鼠标放在需要在其中建立关系类的 geodatabase 或要素数据集上，单击右键。

2、 鼠标指针指向 New。

3、 单击 Relationship Class。

4、 输入新关系类的名字。

5、 单击源表或要素类。

6、 单击目标表或要素类。

7、 单击 Next。

8、 选择 Simple (peer-to-peer) relationship 单选按钮：建立简单关系类。

9、 单击 Next。

10、输入向后和向前路径标注。

11、选择消息传递方向。

12、单击 Next。

13、选择第一个基数选项。在这个例子中，一个拥有者可以拥有一个单独地块，一个地块只属于一个拥有者，因此这是一个一对一的关系类。

14、单击 Next。

15、单击第一个选项添加属性到关系类。

16、单击 Next。

17、为了添加一个字段，单击 Field Name 列下的一行，输入添加字段的名字。

18、在添加字段右边单击 Data Type 列，选择数据类型。

19、在下面的对话框中设置新字段的属性。

20、重复步骤 17 到 19，定义关系类的所有属性字段。

21、单击 Next。

22、单击下拉列表箭头，查看源要素类或表的字段列表，单击要素类或表的主关键字。

23、输入源表或要素类的外关键字。

24、单击下拉列表箭头，查看目标要素类或表的字段的列表。单击要素类或表的主关键字。

25、输入目标表或要素类的外关键字。

26、单击 Next。

27、检查为新关系类指定的选项。如果需要修改，单击 Back 按钮。

28、选项确定后，单击 Finish 按钮，建立新的关系类。

提示：

在一个属性关系中，关系表必须有字段扮演源和目标要素类或表的外关键字。这些外关键字与源和目标要素类或表的主关键字相关联。

4、4、6、10 创建关系规则(仅对 ArcInfo 和 ArcEditor)

关于属性规则

当建立一个关系类时，使用初始化的基数，如一对多或多对多。在真实的系统内，关系基数是特定的。一旦建立了关系类，可以通过建立规则来控制基数。关系规则允用于限制源要素或表中的对象类型，是否可以被连接到目标要素类或表中的一个确定类型的对象。

如何建立关系规则（仅能在 ArcInfo 和 ArcEditor）

1、 在 ArcCatalog 目录树中，在需要建立规则的关系类上，单击右键。

2、 单击 Properties 命令。

3、 单击 Rules 标签，进入 Rules 选项卡。

4、 如果源类有子类型，单击与关系规则关联的子类型。如果源类没有子类型，无须特别选择，关系规则则将应用于所有要素。

5、 如果目标类有子类型，单击源类中被选的子类型相关的目标子类型。如果目标没有子类型，无须特别选择，关系规则将应用于所有的要素。

如果关系类的一边或两边是多，可以限制基数的指定范围。在这个例子中，关系的源是一，因此不能修改基数范围。但是，目标边是多，因此可以修改其范围。

6、 选择复选框 Specify the range of …：指定每一个源对象相关的目标对象的范围。

7、 单击上或下箭头，增加或减少相关目标对象的最大或最小数目。

8、 重复步骤 4 到步骤 7，指定这个关系类的所有关系规则。单击 OK 按钮或 Apply 按钮建立关系规则。

提示：

一旦一个关系规则被添加到一个关系类，这个关系规则成为惟一可以存在的有效关系类。为了使其他关系结合，基数有效，必须建立其他的规则。

4、4、6、11 浏览和修改关系属性 (仅对 ArcInfo 和 ArcEditor)

关于关系属性的视图和修改

关系类一旦被建立，它就会显示在 ArcCatalog 目录树中，可以查看其属性，以及任何特定要素类的关系。

在建立关系类时指定的属性是不能被修改的，除了重命名它。可以随意设置和修改关系规则。如果需要修改其他属性，删除关系类和重建它都很容易的。但是，如果这个关系类有很多关系规则，重建它就比较枯燥。可以让这个过程变得容易些的一个方法就是用统一建模语言（UML）的模型图建立关系类，统一建模语言（UML）的模型图可以被修改和随时返回。

ArcCatalog 不允许修改关系类的属性，或在 ArcMap、ArcCatalog 中正在访问的任何 geodatabase 数据集。在修改前，先关闭所有 ArcMap 和 ArcCatalog 中实例。

如何修改关系类属性

在 ArcCatalog 中重命名一个关系类

1、 鼠标放在需要重命名的关系类上，单击右键。

2、 单击 Rename。

3、 输入关系类的新名字，按下键盘的回车键。

提示：

如果在 ArcCatalog 中注册了源类或目标类为版本化，这两个关系类和与它相关的类都被注册为版本。

在 ArcCatalog 中删除一个关系类

1、 鼠标放在需要删除的关系类上，单击右键

2、 单击 Delete 命令。

提示：

当在 ArcCatalog 中删除一个要素类或一个表时，如果这个要素类或表参与了一个关系类，这个关系类也将被删除。

4、4、6、12 在 ArcMap 中查找相关联的对象 (仅对 ArcInfo 和 ArcEditor)

关于在 ArcMap 中查找相关联的对象

通过 ArcMap，可以在 geodatabase 中查找与特定对象相关的对象。例如：

当在地图中标定一个要素时，可以在 Identify Results 对话框中看到与它相关的对象。如果在 Identify Results 对话框中选择这个相关的对象，这个相关对象又与其他关系类相关，可以继续选择与它相关的对象。

当在一个表中选择了一个或多个行或要素时，可以打开相关表选择相关对象。

如何查找相关对象

查找一个要素的相关对象

1、 在 ArcMap 中，单击 Identify 工具。

2、 在 Identify Results 对话框中，单击 Layers 下拉箭头，在需要标定的要素上单击其图层。

3、 单击地图中的要素。

4、 在 Identify Results 对话框中左面板双击要素。

5、 双击关系路径标注。

相关的对象被列在路径标注下面。

6、 单击需要查找属性的相关对象。

提示：

如果在 Identify Results 对话框中选择这个相关的对象，这个相关对象又与其他关系类相关，可以继续选择与它相关的对象。

查找表中一个对象的相关对象

1、 在 ArcMap 的内容表上，单击 Source 标签。

2、 在包含需要查找相关对象的表上，单击右键，单击 Open Attribute Table。

这个包含需要查找相关对象的表被打开。

3、 选择这个对象，它的相关对象需要查找。

4、 单击 Options，指向 Related Tables，然后单击关系的路径标注。

相关表的一个新对话框将打开。

5、 单击 Show Selected，仅显示这些与在第一个表中被选择对象相关的对象。

4、4、6、13 用 join 添加相关的字段 (仅对 ArcInfo 和 ArcEditor)

使用相关联字段

一个相关联类的字段不能自动连接到一个类的属性表。这是因为一对多和多对多基数不支持这种类型的可视化。同时，一个给定表或要素类可能参与多个关联类，这个关联类存取大量的相关字段，它们的中大多数可能没有特定任务的需要。

对于一对一和多对一（实际和一对一是相同的）的关联类，使用 ArcMap 连接一个具体的相关要素到一个表来天际相关字段。这个帮助避免了不需要字段所带来的混乱。可以基于已存的关联类来建立一个连接，这种方法比指定所有需要的属性来定义一个连接要快一些。一旦建立了连接，从相关要素类或表中的字段被添加到你的要素层中。可以使用这些字段来标注、符号化和查询要素。

Attribute 对话框提供访问一个类的所有相关字段，不管是否建立了一个连接。

在 ArcMap 中如何使用相关字段

1、 在 ArcMap 的内容表上，单击要素层。

2、 指向 Joins and Relates，单击 Join 命令。

3、 单击 Join Options 的下拉箭头，单击 Join data based on a predefined relationship class 命令。

4、 单击下拉箭头，看到一个关系类列表，单击关系类。

5、 单击 OK。

可以使用相关字段来标注、符号化和查询要素。

提示：

如果关联类是一对多或多对多，每个要素可以有多个相关对象。这种情况下，第一个相关对象的属性被连接到要素。

ArcMap 提供了编辑关联的工具。

4、4、6、14 关系类：最大化性能(仅对 ArcInfo 和 ArcEditor)

当建立一个新的关系类时，主关键字和外关键字是自动被建立为索引的（如果它们还没有索引）。当查找一个关系或编辑一个含消息的关系时，这些索引可以提高速度性能。

当在 ArcMap 中编辑一个要素类或含消息的表时，ArcGIS 打开相关联的类，可以通过移动或删除它本身或执行一些自定义的行为来响应消息。如果相关联的类并不在所操作的地图中 ArcGIS 将打开它，因此它可以响应消息，然后关闭它。在每个编辑中，ArcGIS 会再次打开和关闭相关联的类，降低了性能。为了防止这个发生，当在 ArcMap 中操作一个类时，在地图中打开所有相关联的类。这种方法，当这些要素类被添加到 ArcMap 时，相关联的类都会保持打开，直到这些类被移除地图或完成 ArcMap 会话。

对于许多 ArcInfo 的 coverage 数据模型，Feature Attribute table 几乎不包含什么项，要素类的许多属性被存储在一个相关的表中。这个可以用 geodatabase 的要素类处理，但是，在 geodatabase 中查找一个关系是比在 INFO 表中查找更费时的操作。在 INFO 环境下，在外部存储一个要素的符号是很平常的，相关联的表称为查找表。这个也可以在 geodatabase 中操

作，通过使用关系类和连接两个表；但是，符号化大量数据集时这种方法很慢，即使有主关键字和外关键字为索引。尽力存储符号的属性在要素类的表中。出于性能考虑，推荐符号信息存储在要素类中。

4、4、7 注记

4、4、7、1geodatabase 中的注记

Geodatabase 中注释被存储在注释要素类中。和其他要素类一样，注释类中的所有要素具有地理位置和属性，可以存储在一个地理要素数据集内部，也可以作为独立的注释类。每个注释要素有自己的符号，包括字体、大小、颜色，还有其他文本符号属性。注释不一定完全是文本，还可以包括图形－例如框和箭头－它需要其他符号类型。

在 ArcGIS 中存储文本主要有两种选项：动态标注和地图文档注释。

如果有多于几百页的文本和需要在许多地图中使用可编辑的文本，使用 geodatabase 注释。在 ArcMap 中使用 geodatabase 注释通常比使用地图文档注释更快。Geodatabase 注释可以被空间索引，意味着它将绘制和选择更快。

standard 和 feature-linked annotations

在 Geodatabase 中有两种类型的注释－标准的注释类和连接要素的注释类。标准注释类在 geodatabase 中不与要素相关联。标准注释类的一个例子就是地图上山脉的名称，只是标注一个区域，没有特定的对应要素。

连接要素的注释类与 geodatabase 中一个要素类的特定要素相关联，连接要素的注释文本反映了其所关联要素的一个字段或多个字段。例如，在一个供水网络中的给水总管可以用它们的名称来注释，它们被存储在给水总管要素类的一个字段中。

注释是通过带消息的复合关系来连接要素的。被注释的要素类在关系中是源类，注释要素类是目标类。对于其他复合关系，源要素控制着目标要素。如果源要素的一个属性值改变，基于这个属性连接的注释类将会自动更新反映这个变化。当源要素被移动或旋转，连接的注释类也会跟着它移动或旋转。当一个源要素从 geodatabase 中被删除，连接注释的要素也会被删除。

在供水网络例子中，一个水龙头可能距离一个十字路口太近，可能需要移动 50feet。当这个水龙头被移动时，它连接的注释类也将跟着它移动。在同一个网络中，一个输水管的名称可能修改。当这个名称字段的值被修改后，存储在它的连接的注释类要素中的文本也会自动更新为新名称。

在一个要素数据集中的连接要素的注释要素将连接同一个数据集中的一个要素类。相似的，独立的连接要素的注释要素类将连接同一个 geodatabase 中的独立要素类。

一个注释要素类只能连接惟一一个要素类，但是一个要素类可以有许多的连接注释要素类。

注释类

标准的和连接要素的 geodatabase 注释要素类包括一个或多个注释类。每个注释类包含属性，它们决定了一个注释子集如何在要素类中显示。

对于标准注释和连接要素的注释类，它们都有这些属性：

- 当建立新注释时应用的默认符号
- 可见比例范围

对于连接要素的注释，还可以有下面的属性：

- 基于连接要素类中的属性如何定义注释文本字符串。
- 在连接要素类中哪些要素将被注释类注释。
- 如何放置新注释

例如，如果有一个城市的注释要素类，可以有文本大小变化的注释类，以及管理小城市、

中等城市和大城市的比例范围的独立注释要素类。注释类避免定义和维护多个注释要素类。

可以建立和修改注释类。使用 ArcCatalog 建立一个空的注释要素类或当转换标注为注释都可以建立一个注释类。当转换标注时，每个标注类转换为一个注释类。一旦已经建立一个注释要素类，可以使用 Feature Class Properties 对话框来建立和修改注释类。

为了达到 ArcMap 的最佳化显示和查询的最佳性能，总是给每个注释类定义一个可视的比例范围，因此注释要素仅在放大到足够读取它们的文本时才绘制。

如果有多于一个的的注释类，注释类在注释要素类中以子类型被实现。为了给每个注释类建立一个新的注释，在 Editing 工具条上选择这个类为目标类。

可以在任何许可范围内建立和修改注释类，包括 ArcView。

文本符号

一个注释要素类包含定义的一个或多个文本符号集合。这个符号包含描述注释要素如何被绘制的属性，如字体、大小和颜色。例如，如果有一个小、中、大城市的注释，建立三种不同字体大小的文本符号分配给注释。由于每个注释要素不要存储它自己的符号属性，ArcGIS 可以减少存储需求，最优化显示和查询性能。提交一个有限的符号列表可以帮助提高所建立的任何新注释要素的标准。

无论何时建立一个注释类，都给它建立一个默认的文本符号。也可以使用 Feature Class Properties 对话框在任何时候给一个要素类建立其他的文本符号。一旦建立了符号，在 ArcMap 中编辑时就可以把它们赋给注释要素。

当在 ArcMap 中把符号赋给注释要素时，可能会发现所建立的文本符号不包含所需要的属性。例如，可能需要一个较小的字体大小类适合一个拥挤范围的注释。其中一个方法就是在 ArcCatalog 中使用属性来建立一个新的文本符号，然后在 ArcMap 中的编辑会话中将这个新文本符号赋给注释要素。

为每个需要的惟一属性集合建立新符号可能导致一个很长的符号列表，它将很难工作。ArcMap 允许在 feature-by-feature 基础上修改符号属性。当在 ArcMap 中编辑时，可以选择注释，修改它的任何符号属性。

在 feature-by-feature 基础上编辑特定属性会导致注释丢失集合中它相关的文本符号的参考。

当发生这种情况时，注释要素存储它的所有符号属性。这增加了存储需要，降低了注释显示和查询性能。

这些属性可以基于 feature-by-feature 的基础编辑，当继续维护集合中一个文本符号的参考。

- X 和 Y 偏移
- 水平和垂直定线
- 翻转角度
- 字体名字、大小、颜色、字符间距、字符宽度和背景符号
- 字体－粗体、斜体和下划线
- 字空间
- 主要属性

例如，如果减小一个注释要素的字体大小，从 12 到 8，注释要素仍然参考一个预先定义的符号；但是，它存储它自己的字体大小 8。如果修改了一个上面并未列出的属性，如给注释要素一个晕环，注释要素将失去它的相关文本符号的参考，存储所有的符号属性。

可以使用文本格式标签来修改一条注释的一部分格式。这可以让用户建立混合格式的注释，例如，一个句子中字可以加下划线。格式标签总是比注释要素的符号属性优先采用。

当决定哪一种符号存储在注释要素类中，在建立注释要素类时给每个都选择一个默认的

符号，然后添加含一般需要的属性的其他符号。对于很少被使用的属性，可以在 feature-by-feature 基础上应用它们。少数存储自己符号属性的要素会稍微影响存储要求，以及要素类的性能。

当在编辑注释要素类时，可以看见红色边框代替注释文本。这表示注释要素的 SymbolID 已经与它的文本符号断开。

这可以以两种方式发生：

- 注释已经从一个注释要素类拷贝、复制到另一个注释要素类，第一个注释要素类中的 SymbolID 在第二个注释要素类的符号集合中已经不存在。
- ArcCatalog 中注释要素类已经被编辑，SymbolID 指向的文本符号已经被删除。

为了解决这个问题，可以做以下其中之一。

- 建立新的符号到符号集合，计算注释要素的 SymbolID 给它。
- 计算注释要素的 SymbolID 给新的文本符号编码。

注释存储

在 ArcGIS9.2 版本中，注释要素的存储有所改变，由于新属性和地理空间参考。在以前的版本中，这个变化在需要以前的 geodatabase 编辑是不能进行的。注释要素类现在同样在 ArcGIS 应用中利用相同的功能－Save A Copy command，给已经编辑的 geodatabase 版本的注释要素保存合适的版本。因此，ArcGIS9.2 可以编辑 9.0 和 9.1 的 geodatabase 注释。在 8.3 或更早的 geodatabase 注释仍需要更新才能编辑。

4、4、7、2 创建注释概要

ArcCatalog 和 ArcMap 都具有建立标准注释和连接要素的注释的工具。建立注释需要建立一个要素类存储注释，同时要装载注释要素。可以采用以下三种方法中的一种：

- 如果已经花费时间和精力建立 ArcInfo Workstation coverage 或 PC ARC/INFO coverage, SDE 3, CAD 或 VPF 注释，可以在 ArcMap 中显示这些格式的注释，或导入它们到一个新的标准的或连接要素的注释要素类。

从这些格式导入到一个连接要素的注释要素类并不自动连接注释到要素。如果导入的注释要素类含有一个字段，与需要连接的要素类中的一个字段相关联，可以使用 SQL 来连接注释到它们的要素。如果没有这些字段，可以用 ArcMap 中的编辑工具手动连接导入注释到要素。

- 如果需要显示的文本被存储在一个要素类中的一个或多个属性中，可以在 ArcMap 中以标注来显示文本，转换它们到注释。这建立了一个新的标准或连接要素的注释要素类。
- 如果没有任何数据，或需要从头开始设计注释要素的 schema，在 ArcCatalog 中建立一个空的标准或连接要素的注释要素类。

组装一个空的标准注释要素类，可以在 ArcMap 中显示，转换标注到注释，或者使用 Annotation 工具条上工具手动添加注释。

组装一个空的连接要素的注释要素类，在 ArcMap 中选择要素，使用 Annotate Selected Features 命令生成注释。也可以在 ArcMap 中建立一个新的注释来添加新要素。

当通过转换标注或导入其他格式的注释在 ArcSDE geodatabase 建立注释，应该尽量先注册数据为版本，然后建立注释。如果数据不是版本化的，所有的数据将被直接加载到 base tables，数据库的压缩将并不是必需的。

如果建立的连接要素的注释被连接到一个几何网络要素类，应该在建立几何网络后再建立注释。当要素在几何网络建立过程中被捕捉时，它们的几何图形被修改，但是并没有传递消息给它连接的注释要素来更新。因此，在建立了几何网络之后，如果一个要素的几何图形在捕捉过程被修改，这个要素和它连接的注释可能不再正确对应。

4、4、7、3 在 ArcCatalog 中查找注释要素类

可以在 ArcCatalog 中建立和管理注释要素类。

注释要素类可以存在于要素数据集内部，也可以存在于 geodatabase 的根目录。连接要素的建立在要素数据集内部的连接要素的注释类，必须连接到同一数据集内的另一个要素类；同样地，建立在 geodatabase 的根目录的连接要素的注释类，必须连接到同一个 geodatabase 的根目录另一个要素类。

一个连接要素的注释要素类可以仅连接到一个单一的要素类。但是，一个要素类可以有多个连接的注释要素类。

在 ArcCatalog 目录树上查看注释要素类时，并不能直接看出是标准的注释类还是连接要的注释类。Annotation Feature Class Properties 对话框的 Annotation Classes 标签可以判断是标准的注释类，还是连接要素的注释类，连接要素的注释类有一个标注字段和表达式按钮，标准注释类却没有。

在 ArcCatalog 目录树中，也看不出连接要素注释类关联的是哪一个要素类。但是，Annotation Feature Class Properties 对话框的 Relationships 标签可以看到它连接的要素类。

4、4、7、4 管理注释要素类

在 ArcCatalog 中操作 geodatabase 注释要素类与在 geodatabase 中操作其他要素类是很相似的。可以建立、重命名、拷贝、粘贴或删除注释要素类，也可以查看和修改它们的属性。

管理要素类属性

为了在 ArcCatalog 中访问注释属性，双击一个注疏要素类，打开 Feature Class Properties 对话框。

管理连接要素的注释类

建立一个新的连接要素的注释要素类时，一个关联类被建立用于连接注释要素类和要素类，这个关联类可以和其他关联类一样在 ArcCatalog 中被管理。

不能先建立一个标准注释要素类，通过建立一个关联类让它变成一个连接要素的注释要素类。实际上，无论怎么做都不可能让一个标准注释类变成一个连接要素的注释要素类。但是，相反地却可以，如果删除关联类或一个给定的连接要素的注释类，它就变成了一个标准的注释要素类。

在 ArcCatalog 中拷贝和粘贴一个要素类时，任何连接的注释类，还有相关联的类，将都被拷贝和粘贴。反过来也是可行的：如果拷贝和粘贴一个连接要素的注释要素类，连接的注释类（相关联的类）也将被拷贝和粘贴。

复合注释要素类

为了避免维护多个注释要素类，可能需要将它们复合到单一的注释要素类中。例如，如果它们注释同一个要素类中的要素，或它们是标准注释要素类注释多个要素类中的相似要素，就可以这样做。当复合注释要素类时，来自每个要素类的注释变成一个新注释要素类中注释类。可以用 Append Annotation 要素类工具来复合注释要素类。

更新 ArcGIS8 的注释要素类

ArcGIS 9 添加了注释要素类的功能，在以前版本中并不可用。如果有 ArcGIS8 中建立的注释要素类，可以显示和查询它们。但是，为了编辑或修改它们的属性，必须升级它们。升级注释要素类也添加文本符号字段，它对于查询和编辑是很有用的。

4、4、7、5 管理注释要素类属性

Feature Class Properties 对话框可以用于设置假名、空间和属性索引，以及仅管理注释要素类的属性－注释要素、符号和设置注释要素如何被编辑。

可以通过双击 ArcCatalog 目录树中的一个注释要素类，打开 Feature Class Properties 对话框。注意一些属性是可以编辑的，其他的属性是只读的。

字段

在注释要素类中的 Fields 标签列出了字段。以下的是必需字段，不能被删除：

ObjectID

Shape

FeatureID

ZOrder

AnnotationClassID

Element

SymbolID

Status

Shape_length

Shape_area

SymbolID 字段的值通常大于或等于 0。这个值是文本符号在符号集合中编码。如果这个值是-1，注释要素被存储在显示属性的记录中，它在符号集合中不是参考一个文本符号。当 SymbolID 值是-1 时，当每个属性被 override，Override 字段值将是 0。

要素类中的其他字段参考注释要素的显示属性。这些字段可以被编辑，修改将会更新要素 BLOB element。同样地，通过 Attribute Editor 中的 Annotation 标签的任何注释要素的修改将被更新。如下列表的字段可以被删除：

TextString

FontName

FontSize

Bold

Italic

Underline

VerticalAlignment

HorizontalAlignment

XOffset

YOffset

Angle

FontLeading

WordSpacing

CharacterWidth

CharacterSpacing

FlipAngle

Override 字段是一个非编辑的字段，它由 ArcGIS 维护。当 override 值为 0 时，注释要素使用参考的 SymbolID 文文符号被显示，没有显示属性存储在记录中。当 override 值大于 0 时，注释要素使用 SymbolID 作为一个基显示，然后应用被 override 的属性到这个符号。下表是这些可能值的列表。

子类型

如果注释要素类有多个注释类，注释类可以用子类型来实现。Subtypes 标签列出了注释类。但是，为了管理注释类，使用 Annotation classes 标签。

关联

在 ArcCatalog 中，要素连接的注释类，Relationships 标签显示了注释要素类和它的连接要素类之间的关联类。

注释类

Annotation Classes 标签允许管理注释类。可以从这个列表中添加或移除注释类。为了修改或查看一个注释类的属性，在这个列表中选择注释类，访问任何属性。

其他属性

Annotation 允许修改注释要素类的基准标度。修改基准标度需要很长的时间完成，因此在仔细选择了新的比例后再修改。

Annotation 标签也允许设置属性，这些属性会影响注释如何被编辑。

符号集合列出了预定义的文本符号，当在 ArcMap 中建立或编辑注释要素时，可以赋值给它们。

4、4、7、6 编辑注释要素类属性

当建立一个注释要素类时，设置影响它如何被编辑的属性。注释要素类被建立后，可以通过 Feature Class Properties 对话框修改它的任何属性。

从符号表中选择必需的符号

这里有一个关于哪些文本符号属性可以在 ArcMap 中被编辑的限制。

如果选择了这个复选框，就仅可以编辑那些允许注释要素维护集合中参考它们的相关联的文本符号的符号属性。如果取消选择这个复选框，就仅能编辑如下任何注释要素的文本符号属性。

X 和 Y 偏移

水平和垂直定线

翻转角度

字体类型、大小、颜色、字符间距和背景符号

粗体、斜体和下划线

字空间

主要属性

例如，如果选取这个复选框，可以减小一个注释要素的字体大小：从 12 到 8。编辑后，这个注释要素仍然参考集合中的一个文本符号，但是，它存储它自己的字体大小：8。

如果没有选择这个复选框，可以编辑任何文本符号的属性。如果修改了上面未列出的属性，如给注释要素晕环，注释要素丢失集合中参考它的关联文本符号。当发生这个时，注释要素存储所有它的符号属性。这增加了注释的存储需求，降低了显示和查询性能。

如果注释不是文本，是由图片元素组成，如线、圆弧、多边形或标记符号，取消这个复选框的选择，否则，要素类将不能存储这些图片元素。

当新要素被添加时建立注释（仅连接要素的注释类）

如果需要一条新的注释在添加一个新要素到连接要素类时被建立，选择这个复选框。

如果需要在新要素被添加后建立注释，取消这个复选框。

当要素的形状被改变时更新注释（仅连接要素的注释类）

如果需要注释的位置在连接要素被修改时自动更新，选择这个复选框。

当选择了这个复选框，注释会根据修改的要素的新形状来自动重新配置，避免用户来移动它。但是，如果手动放置一条注释，仅仅是对连接要素的形状重置进行小小的编辑，相对连接要素，注释要素支持其默认的位置。因此，如果需要更大的控制注释的位置，取消选择这个复选框。这会保持注释在它原始的位置，如果需要可以手动移动注释。

4、4、7、7 创建注释要素类

关于建立注释要素类

可以建立一个空的注释要素类，然后建立它的注释，或者通过在 ArcMap 中转换动态的标注显示到一个新注释要素类来建立带注释的注释要素类。

在 ArcCatalog 中建立一个空的注释要素类，需要给它指定注释类。

如何建立注释要素类

建立在要素数据集内部的标准注释要素类

1、 在 ArcCatalog 目录树中，鼠标指针放在需要建立注释要素类的要素数据集上，单击右键。

2、鼠标指针指向 New，单击 Feature Class 命令。

3、输入名字。为了给注释要素类建立一个假名，输入假名。

4、单击下拉列表箭头，选择 ESRI Annotation Feature 为要素类型。

5、单击 Next。

6、输入一个基准标度。这个比例应该与注释能正常显示的比例相等。

地图单位是被自动设置的，这是由于注释要素类在一个数据集内部。这个单位与要素数据集的坐标系统相匹配。如果这个要素数据集坐标系统是未知的，这个单位默认为米。

7、为了设置从符号表选项中选择必需符号，可以看注释要素类编辑属性。

8、单击 Next。

9、对于文本符号，给第一个注释类设置默认文本符号属性。

10、指定注释类的可见比例范围。

11、如果需要添加一个另外的注释类，单击 New，指定注释类的名字。重复步骤 9 到 10 来设置它的属性。

12、重复步骤 11，指定所有注释类和它们的属性。

13、单击 Next。

14、如果建立的新注释要素类在 file 或 ArcSDEgeodatabase 中，需要使用自定义存储关键字，单击 Use configuration keyword,，然后从下拉列表中选择需要使用的关键字。如果不需要使用自定义存储关键字，保持默认。

15．单击 Next。

16、不需要修改任何字段属性。要素类在一个要素数据集内部，因此空间参考将自动设置要素数据集的。

默认情况下，有几个字段被添加到一个注释要素类。一些是必需的，其他一些是可选的。如果不需要可选字段，可以立即删除它们，或者在建立要素类后通过 Feature Class Properties 属性对话框来删除它们。

如果需要从其他要素类（geodatabase, coverage）来导入字段，单击 Import。或者可以手动添加字段到注释要素类。

17、单击 Finish。

建立在 geodatabase 根目录的一个标准注释要素类。

1、 在 ArcCatalog 目录树中，鼠标放在需要建立注释类的 geodatabase 上，单击右键。

2、 按照“建立在要素数据集内部的标准注释要素类”中步骤 2 到步骤 5 做。

3、 指定注释要素类的空间参考。一个空间参考由一个坐标系统，空间域和精度组成。

4、 在坐标系统窗口中选择一个坐标系统或单击 Import 或 New 来指定一个坐标系统。

Import 可以从一个已存在的要素类或要素数据集中导入一个坐标系统。

New 可以建立一个新的坐标系统

一旦已经选择了一个坐标系统，可以单击 Modify 来修改它。

5、 单击 Next。

6、 X/Y 域是基于在步骤 3 中指定的坐标系统来计算的。如果需要可以输入一个不同的 X/Y 域，但是默认应该是足够的。

7、 单击 Next。

8、 输入一个基准标度。这个比例应该与注释能正常显示的比例相等。

9、 单击 Map Units 的下拉列表箭头，选择注释存储的单位。这个单位应该与步骤 3 指定的坐标系统的单位匹配。

10、为了设置从符号表选项中选择必需符号，可以看注释要素类编辑属性。

11、单击 Next。

12、对于文本符号，给第一个注释类设置默认文本符号属性。

13、指定注释类的可见比例范围。

14、如果需要添加一个另外的注释类，单击 New，指定注释类的名字。重复步骤 12 到 13 来设置它的属性。

15、重复步骤 14，指定所有注释类和它们的属性。

16、单击 Next。

17、如果建立的新注释要素类在 file 或 ArcSDEgeodatabase 中，需要使用自定义存储关键字，单击 Use configuration keyword,，然后从下拉列表中选择需要使用的关键字。如果不需要使用自定义存储关键字，保持默认。

18、单击 Next。

19、默认情况下，有几个字段被添加到一个注释要素类。一些是必需的，其他一些是可选的。如果不需要可选字段，可以立即删除它们，或者在建立要素类后通过 Feature Class Properties 属性对话框来删除它们。

20、单击 Finish。

建立在要素集内部的连接要素的注释要素类

1、 在 ArcCatalog 目录树中，鼠标放在需要建立注释要素要素类的要素数据集上，单击右键。

2、 鼠标指针指向 New，单击 Feature Class 命令。

3、 输入名字，为了建立注释要素类的假名，输入假名。

4、 单击下拉列表箭头，选择 ESRI Annotation Feature 为要素类型。

5、 选中 Link the annotation to the …复选框，连接注释到要素类。

6、 单击下拉箭头，选择需要连接到新注释要素类的要素。这个要素类必须与建立的注释要素类再同一个要素数据集。不能修改属性。

7、 单击 Next。

8、 输入一个参考比例。这个比例应该与注释能正常显示的比例相等。地图单位是被自动设置的，这是由于注释要素类在一个数据集内部。这个单位与要素数据集的坐标系统相匹配。如果这个要素数据集坐标系统是未知的，这个单位默认为米。

9、 关于如何设置以下三个选项（从符号表中选择必需的符号，添加新要素时建立注释，当要素的形状被修改时更新注释）的信息可以看 Annotation feature class editing propertie。

10、如果没有安装 Maplex，选择一个标注引擎。单击 Properties 来指定标注属性。

11、单击 Next。

12、指定连接要素类的字段，它包含第一个注释类的文本。单击 Label Field 下拉列表框选择用于建立注释的要素字段，或单击 Expression 按钮，建立从多个字段获取注释文本的表达式。

13、设置注释文本的默认文本符号和放置属性。

可以一次一个属性来设置它们或单击 the Label Styles button 按钮加载一个已存的标注样式。

14、单击 Scale Range 来指定显示注释文本的比例范围。

15、单击 SQL Query 来指定连接要素类中的惟一确定要素要被这个注释要素类注释。

16、如果需要添加其他的注释类，单击 New，指定注释类的名字。重复步骤 12 到 15

设置它的属性。

17、重复步骤 16，指定所有注释类和它们的属性。

18、单击 Next。

19、如果建立的新注释要素类在 file 或 ArcSDEgeodatabase 中，需要使用自定义存储关键字，单击 Use configuration keyword,，然后从下拉列表中选择需要使用的关键字。

20、单击 Next。

21、不需要修改任何字段属性。要素类在一个要素数据集内部，因此空间参考将自动设置要素数据集的。

默认情况下，有几个字段被添加到一个注释要素类。一些是必需的，其他一些是可选的。如果不需要可选字段，可以立即删除它们，或者在建立要素类后通过 Feature Class Properties 属性对话框来删除它们。

如果需要从其他要素类（geodatabase, coverage）来导入字段，单击 Import。或者可以手动添加字段到注释要素类。

22、单击 Finish。

提示：

一个关联类被自动建立连接注释要素类和被注释的要素类。为了查看哪一个关联类与一个注释要素类关联，双击这个注释要素类，单击 Relationships 标签。

指定每个注释类的可见比例范围，提供在 ArcMap 中操作性能。也可以在 ArcMap 中指定一个可见比例范围。

建立 geodatabase 根目录的连接要素的注释要素类

1、在 ArcCatalog 目录树中，鼠标放在需要建立注释类的 geodatabase 上，单击右键。

2、按照“建立在要素数据集内部的连接要素注释要素类”中步骤 2 到步骤 7 做。

3、指定注释要素类的空间参考。一个空间参考由一个坐标系统，空间域和精度组成。

4、在坐标系统窗口中选择一个坐标系统或单击 Import 或 New 来指定一个坐标系统。

Import 可以从一个已存在的要素类或要素数据集中导入一个坐标系统。

New 可以建立一个新的坐标系统。

一旦已经选择了一个坐标系统，可以单击 Modify 来修改它。

5、单击 Next。

6、 X/Y 域是基于在步骤 3 中指定的坐标系统来计算的。如果需要可以输入一个不同的 X/Y 域，但是默认应该是足够的。

7、 单击 Next。

8、 输入一个基准标度。这个比例应该与注释能正常显示的比例相等。

9、 单击 Map Units 的下拉列表箭头，选择注释存储的单位。这个单位应该与步骤 3 指定的坐标系统的单位匹配。

10、按照“建立在要素数据集内部的连接要素注释要素类”中步骤 9 到步骤 20 做。

11、在最后一个面板，不需要修改任何字段属性。对于 Shape 字段，要素类被赋给与它连接的要素类相同的空间参考。

如果需要从其他要素类（geodatabase, coverage）来导入字段，单击 Import。或者可以手动添加字段到注释要素类。

12、单击 Finish。

提示：

一个关联类被自动建立连接注释要素类和被注释的要素类。为了查看哪一个关联类与一个注释要素类关联，双击这个注释要素类，单击 Relationships 标签。

指定每个注释类的可见比例范围，提供在 ArcMap 中操作性能。也可以在 ArcMap 中指

定一个可见比例范围。

4、4、7、8 转换标注到注释要素类

关于转换标注到注释要素

在 ArcMap 中，可以转换动态标注到标准或连接要素注释要素。如果不确定转换为标准注释，还是连接要素注释，可以看 geodatabase 中的注释。

可以从 geodatabase 的任何一个图层、coverage,、shapefile、或 CAD 要素类数据源转换标注到标准注释要素。可以建立一个新的标准注释要素类或添加新的注释到一个已存的注释要素类。

为了转换到连接要素注释类，被标注的图层必需有一个 geodatabase 要素类数据源。不同于转换到标准注释类，在这个过程中总是建立一个新连接要素注释要素类－不能添加新注释到一个已存的连接要素的注释要素类中。这个新注释要素类建立在与被标注的 geodatabase 要素类在相同的一个要素数据集内，或如果标注的要素类在 geodatabase 根目录，那么新注释要素类也要这个 geodatabase 根目录。

如何转换标转到注释要素

1、 通过确定合适的比例和标注属性，准备转换的标注。

2、 在 ArcMap 内容表中，鼠标指针放在需要转换标注的数据层上，单击右键。为了转换多个数据层，在数据组上，单击右键。

3、 单击 Convert Labels to Annotation 命令。

4、 对于 Store Annotation，选择 In a database。

5、 指定需要建立注释的要素。

6、 建立连接要素的注释，选取 Feature Linked 复选框。建立标准注释，取消这个复选框的选择。

7、 如果建立的是标准注释，需要添加注释到一个已存的标准注释要素类，选择 Append 复选框。

8、 如果建立的是连接要素的注释，单击新注释要素类的名字来修改。

9、 如果建立的是标准注释，单击打开文件夹的图标，指定将建立的新注释要素类的路径和名字，或者如果是添加，指定需要添加到的标准注释要素类的路径和名字。

10、如果是添加到一个已存的要素类，跳过至步骤 15。

11、单击 Properties 按钮。

12、关于如何设置“从符号表中选择必需的符号”“当新要素被添加时建立注释”（对标准注释不可用）或者“当要素的形状被修改时更新注释”等选项的信息可以看编辑注释要素类属性。

13、如果建立的新注释要素类在 file 或 ArcSDEgeodatabase 中，需要使用自定义存储关键字，单击 Use configuration keyword,，然后输入需要使用的关键字。

14、单击 OK。

15、一些标注在当前并没有显示在地图中，这是由于没有地方显示它们。为了转换这些标注，选中 Convert unplaced labels 复选框。这在注释要素类中保存了这些未放置的标注，可以在以后的 ArcMap 编辑会话中放置它们。

16、单击 Convert。

如果选中了 Convert unplaced labels 复选框，需要放置这些未放置的注释。

提示：

可以同时从多个数据层转换标注。在数据组上，单击右键，单击 Convert Labels to Annotation 命令，按照上面描述的步骤为每个数据层完成设置。

标注被转换后，新的注释类自动被添加到地图中，在 ArcMap 的内容表中显示。

如果转换多个标注类的数据层，每个标注类成为注释要素类中的一个注释类。

当转换到一个已存的注释要素类，保证当前地图比例匹配已存要素类的基准比例。如果这些比例不匹配会收到一个警告，将不能继续进行，直到它们匹配才可以。

当转换中有错误，一个错误信息和对话框将出现。可以打开对话框中的登陆文件了解更多的错误信息。

如果有转换错误，登陆文件才会被创建。

这个文件建立在用户临时文件夹目录中，它通常位于\Documents and Settings\user\Local Settings\Temp。

这个登陆文件的名字为 GL<feature class name>#.log。例如，如果目标注释要素类是 AlbanyAnno，第一个为这个目标注释建立的错误登陆文件的名字将是 GLAlbanyAnno0.log，第二个文件的名字为 GLAlbanyAnno1.log。

在登陆文件中错误列表是与加载或者要素修改相关的 geodatabase 错误。

如果标注一个几何网络要素类，应该在建立几何网络后，再转换这些标注到连接要素的注释类中。这是一种推荐的做法，这是因为当要素在几何网络建立过程中被捕捉，它们的几何图形被修改，连接的注释要素却不能被更新。

如果操作 ArcSDE geodatabase，在数据版本化之前，可能转换标注到注释。这样会减少解决冲突，提交编辑版本支持父版本的必需时间。

4、4、7、9 放置未放置的注释要素

关于放置未放置的注释要素

假定在 Conversion 转换对话框中选中了 Convert unplaced labels 复选框，那么在转换标注到注释要素时，未放置的标注被转换为未放置的注释，存储在目标注释要素类中。

以下这些步骤说明了在 ArcMap 的一个编辑会话中如何交互式查看和放置未放置的注释。

如何放置未放置的注释要素

1、 在 ArcMap 中，添加包含未放置注释的注释要素类到地图中，也可以添加可以帮助放置注释的数据层。

2、 在 Editor 工具条上，单击 Editor 命令，单击 Start Editing 命令。

3、 在 Annotation 工具条上，打开 Unplaced Annotation 窗口。

4、 在 Unplaced Annotation 窗口中，单击 Show 下拉箭头，选择含未放置注释的注释要素类。

5、 单击 Search Now ，列出未放置的注释。

6、 默认情况下，未放置注释并不显示在地图中。选择 Draw 复选框，可以实现拖动未放置注释。

7、 如果需要仅在一个特定范围操作注释，放大到这个范围，选取 Visible Extent 复选框，单击 Search Now 更新这个列表。

8、 在列表中双击这个文本，在未放置注释的位置使其闪烁。

9、 放大到所有未放置注释的区域，在列表中的文本上，单击右键，单击 Zoom to Annotation 或 Zoom to Feature 命令。

10、为了放置注释到地图中，在列表中文本上，单击右键，单击 Place Annotation 命令。单击地图中注释，把它拖到需要放置的地方。

11、一旦完成放置注释，单击 Editor 下的 Stop Editing 命令，保存编辑。

4、4、7、10 导入 coverage 或 CAD 注释到 geodatabase 注释

关于导入 coverage 或 CAD 注释到 geodatabase 注释

可以用 Import Coverage Annotation 工具导入一个或多个 coverage 注释要素类。同样地，

可以用 Import CAD Annotation 工具导入一个或多个 CAD 注释要素类。这两个工具都导入到一个新建立的标准或连接要素的注释要素类。从这些格式导入到一个连接要素的注释要素类不连接注释到要素，但是建立关联类。

导入的注释要素类中存储在规则文本属性的值也被转移到新要素类中。

导入到 geodatabase 的 Coverage 和 CAD 注释就如它们在 ArcMap 中显示的一样。基于这个原因，在导入前，应该仔细设置注释的显示。如果同时导入几个平铺的注释要素类，仅需要在 Import Coverage Annotation 或 Import CAD Annotation 工具中为第一个注释要素类设置符号。这个工具允许应用相同的符号到导入的其他所有要素类。

Coverage pseudo items 和 annotation appearance

当在 ArcMap 或 ArcInfo Workstation 显示一个 coverage 注释要素类，在 coverage 的文本属性表中的 pseudo item 字段决定了注释的如何显示。pseudo item 在 ArcMap 中与在 ArcInfo Workstation 用法不同，geodatabase 注释要素类并不存储 pseudo item 。因此，当导入 coverage 注释时要紧记一些规则：

- 由于以下的文本符号属性存储在 pseudo item，在 ArcMap 中不能修改它们：水平和垂直定线、x 和 y 偏移，还有文本大小如果$SIZE 不等于零。如果需要修改这些属性中的任何一个，在 ArcMap 中显示和导入这些注释前，先在 ArcInfo Workstation 中修改 pseudo item 属性。另一个方法就是在导入后在新注释要素类中修改这些属性。
- 如果需要在转换时也转移 pseudo item －－例如，为了查询－应该使用 ArcInfo Workstation 预先在存储这些属性的文本属性表中建立 regular items （不是 pseudo items）。

在 pseudo item 中$SIZE 的决定了导入的注释的文本大小是如何被设置的。如果这个值是零，ArcMap 文本符号大小被使用。这意味着在导入前必须用合适的文本符号大小显示注释。 所选择的大小是新注释在放大到这个比例时才会显示。

如果$SIZE 值不是零，在 ArcMap 中不能修改注释的大小。在这种情况下，$SIZE 值作为 ArcMap 文本符号的大小被使用。

如何导入 coverage 和 CAD 注释到 geodatabase 注释

1、 在 ArcMap 中，添加需要导入的 coverage 和 CAD 注释要素类。

2、 放大到可以正常看到需导入的注释。

3、 根据转换后需要的符号显示来符号化注释。

4、 打开 ArcToolbox 窗口，单击 Toolboxes，单击 System Toolboxes。单击 Conversion Tools 命令，单击加号展开到 To Geodatabase。

5、 如果导入的是 coverage 注释，双击 Import Coverage Annotation。如果导入的是 CAD 注释，双击 Import CAD Annotation。

6、 添加需要导入的数据层。

7、 对于 Output feature class，移动鼠标至新注释要素类的位置，指定它的名字。

如果导入到一个新的连接要素的注释要素类，必须指定相同要素数据集中的一个要素类为连接要素类，或者如果连接的要素类在 geodatabase 的根目录时，必须在这个 geodatabase 的根目录。

8、 指定基准比例。

9、 对于 Create annotation classes from levels，如果需要按照$LEVEL 字段定义来建立一个注释类，选中这个复选框。

如果需要综合到一个单一的注释类到输出要素类，取消这个复选框的选择。

10、如果导入的注释要素类集合都使用相同的符号和相同$SYMBOL 值参考每个符号，选中 Match symbols from first input 复选框。这避免多次将一个符号添加到新要素类的符号集合中。

11、对于'Require symbol to be selected from the symbol table' 选项，可以看编辑注释要素类属性

12、对于 Feature-linked，如果需要导入到一个连接要素的注释要素类中，选择这个复选框。

如果不需要导入到一个连接要素的注释要素类中，跳至步骤 15。

13、对于 Linked feature class，鼠标移动至已存的 geodatabase 要素类，这是新注释要素类将要连接的。

14、对于'Create annotation when new features are added'和'Update annotation when feature's shape is modified'选项，可以看编辑注释要素类属性。

15、单击 Environments。

16、展开 General Settings。

17、设置输出坐标系统。

18、展开 Geodatabase Settings。

19、如果导入到一个 file 或 ArcSDE geodatabase，需要使用自定义存储关键字建立要素类，输入这个关键字。

20、如果导入到一个 personal 或 ArcSDE geodatabase，已知数据的最佳空间索引网格大小，在地图单位中指定它。

21、如果导入到一个 ArcSDE geodatabase，有其他的格网大小，输入它们。

22、设置 x、y 域。

23、单击 OK。

24、在 Import tool 上，单击 OK 导入注释要素类。

提示：

也可以在 ArcCatalog 中运行 Import Coverage Annotation 工具。但是，所有导入的注释被赋予默认的字体 Arial，如果$SIZE 是零，默认的大小为 14 pt。

当转换中有错误，一个错误信息和对话框将出现。可以打开对话框中的登陆文件了解更多的错误信息。

如果有转换错误，登陆文件才会被创建。

这个文件建立在用户临时文件夹目录中，它通常位于\Documents and Settings\user\Local Settings\Temp。

这个登陆文件的名字为 GL<feature class name>#.log。例如，如果目标注释要素类是 AlbanyAnno，第一个为这个目标注释建立的错误登陆文件的名字将是 GLAlbanyAnno0.log，第二个文件的名字为 GLAlbanyAnno1.log。

在登陆文件中错误列表是与加载或者要素修改相关的 geodatabase 错误。

如果操作 ArcSDE geodatabase，在数据版本化之前，可能转换标注到注释。这样会减少解决冲突，提交编辑版本支持父版本的必需时间。

一个注释要素的行长度依赖于以下而变化：文本字符串的长度、是否为 curved text（curved text 占用更多空间），内部符号存储。对于文本注释，80-100 字节的行长度是典型的。以注释存储的图片需要 400 字节，或多于一行。

当导入到连接要素的注释，如果目标注释要素类被连接到一个几何网络要素类，，应该在建立几何网络后，再导入这些注释到连接要素的注释类中。这是一种推荐的做法，这是因为当要素在几何网络建立过程中被捕捉，它们的几何图形被修改，连接的注释要素却不能被更新。

4、4、7、11 导入 SDE 3, VPF, 或 PC ARC/INFO coverage 注释

关于导入 SDE 3, VPF, 或 PC ARC/INFO coverage 注释

可以导入一个 SDE 3, VPF, 或 PC ARC/INFO coverage 注释要素类到一个已存的标准或连接要素的 geodatabase 注释要素类。

要导入的注释要素类可以是空的或已经包含注释的。由于这些格式被符号化为与 ArcMap 中的 coverage 注释很相似，可以使用 Convert Coverage Annotation 命令导入它们。

如果导入的注释要素类包含属性，并且导入到一个空的 geodatabase 注释类中，这些属性将也会导入。

从这些格式导入到一个连接要素的注释要素类并不连接注释到要素，但是建立关联类。

如何导入 SDE 3, VPF, 或 PC ARC/INFO coverage 注释

在 ArcMap 中添加 Convert Coverage Annotation 工具

1、 在 ArcMap 中，单击 Tools 菜单，单击 Customize 命令。

2、 单击 Commands 标签。

3、 单击 Label category 命令。

4、 从 Commands 列表中拖放 Convert Coverage Annotation 命令到工具条或菜单上。

5、 单击 Customize 对话框中的 Close 按钮。

导入 SDE 3, VPF, 或 PC ARC/INFO coverage 注释

1、 打开 ArcMap，添加需要导入的注释要素类。

2、 按照导入后需要的显示方式来设置注释的显示。

3、 打印包含注释的地图，保证在执行步骤 4 之前文本符号是正确的。

4、 根据上面的详细步骤添加 Convert Coverage Annotation 工具到 ArcMap。

5、 确定有一个 geodatabase 注释要素类来存储导入的注释。可以在 ArcCatalog 中建立一个新的注释要素类。如果是导入到一个连接要素的注释要素类，在建立一个新注释类时，必须指定默认的文本符号和一个注释基准比例。这个文本符号应该与在 ArcMap 中使用的符号化注释的符号是相同。这个基准比例应该是导入注释的比例。

6、 单击 Convert Coverage Annotation 按钮。

7、 检查需要导入的注释。可以一次导入多个注释要素类，但是它们将导入到一个单一的 geodatabase 注释要素类。所有导入的注释将有相同的基准比例，这是因为所有在一个 geodatabase 注释要素类的注释具有相同的基准比例。

8、 单击 In a Database 选项。

9、 单击 Browse 按钮，移动鼠标至需要导入注释的一个已存的 geodatabase 注释要素类。

10、单击 Convert 命令。

11、单击 Yes 命令添加导入的注释到地图中。

12、单击 Close 命令关闭 Convert Coverage Annotation 对话框。

提示：

如果目标注释要素类有已存的注释要素，将出现一个对话框，询问是否需要删除所有已存的要素。单击 Yes 删除所有已存的要素，由导入的注释代替它们。单击 No 保留已存的注释，添加导入的注释到其中。单击 Cancel 保留已存的注释，退出此工具，不导入注释。

如果有转换错误，登陆文件才会被创建。

这个文件建立在用户临时文件夹目录中，它通常位于\Documents and Settings\user\Local Settings\Temp。

这个登陆文件的名字为 GL<feature class name>#.log。例如，如果目标注释要素类是 AlbanyAnno，第一个为这个目标注释建立的错误登陆文件的名字将是 GLAlbanyAnno0.log，第二个文件的名字为 GLAlbanyAnno1.log。

在登陆文件中错误列表是与加载或者要素修改相关的 geodatabase 错误。

如果操作 ArcSDE geodatabase，在数据版本化之前，可能转换标注到注释。这样会减少

解决冲突，提交编辑版本支持父版本的必需时间。

一个注释要素的行长度依赖于以下而变化：文本字符串的长度、是否为 curved text（curved text 占用更多空间），内部符号存储。对于文本注释，80-100 字节的行长度是典型的。以注释存储的图片需要 400 字节，或多于一行。

当导入到连接要素的注释，如果目标注释要素类被连接到一个几何网络要素类，，应该在建立几何网络后，再导入这些注释到连接要素的注释类中。这是一种推荐的做法，这是因为当要素在几何网络建立过程中被捕捉，它们的几何图形被修改，连接的注释要素却不能被更新。

4、4、7、12 重连接转换的连接要素的注释

关于重连接转换的连接要素的注释

当 coverage, CAD, SDE 3, VPF, 或 PC ARC/INFO 注释被导入到一个连接要素的注释要素类，同时一个关联类被建立。如果导入的这个注释要素类含有一个字段，这个字段与需要连接到要素类中的字段相关联，可以在这两个要素类之间执行一个 join，然后使用 SQL 来连接注释到它们的要素。

如何重连接转换的连接要素的注释

1、 在 ArcMap 中，加载导入的注释，和关联类需要连接的要素类。

2、 在注释要素类上单击右键，单击 Joins and Relates，然后单击 Join 命令。

3、 单击第一个下拉箭头，从一个表中单击 Join 属性。

4、 单击第二个下拉箭头，单击数据层中字段名字，join 将以这个表为基础建立。这个字段是与注释一起被导入的，这个注释参考被注释的要素。

5、 单击第三个下拉箭头，选择与这个数据层 join 的表。这个是被导入的要素类。

6、 单击第四个下拉箭头，单击表中的字段，这个表是以 join 为基础的。这个字段是与要素一起被导入的，这些要素指向注释要素。

7、 单击 OK。这个表的属性被添加到数据层的属性表中。

8、 打开注释要素类的属性表，在 FeatureID 字段上单击右键，打开 Field Calculator。

9、 在字段列表中，从被 join 的要素类中双击 OBJECTID 字段。

10、单击 OK。

11、移除 join。

4、4、7、13 转换 coverage 或 SDE3 注释到地图文档注释

关于转换 coverage 或 SDE3 注释到地图文档注释

ArcGIS 支持 ArcInfo Workstation coverage, CAD, SDE 3, VPF 和 PC ARC/INFO。可以在 ArcMap 中显示这些格式，但是它们是只读的。如果需要编辑这些注释，或者转换它到存储在 annotation group 中的地图文档注释，或导入它到一个 geodatabase 要素类。

如果仅有几百条注释，并且只打算在当前地图中使用这些转换的注释，那么转换它们到地图文档注释是一种有益的选择。

Pseudo items 和 annotation appearance

coverage 注释按照在 ArcMap 中的显示转换为地图文档注释。基于这个原因，在转换前，应该仔细设置 coverage 在 ArcMap 中的显示属性。在 ArcMap 中，可以在 Layer Properties 对话框中的 Symbols 标签上修改 coverage 注释的显示属性。或者选择一个新的文本符号或修改默认文本符号属性。

在 ArcMap 和 ArcInfo Workstation 中，coverage 注释的显示是由 coverage 的文本属性表（称为 pseudo items）中的特殊字段控制的。这些字段在 coverage 注释的数据层的属性表中是可见的。这些例子包括$ALIGN, $JUSTIFY, 和 $SIZE。Pseudo items 在 ArcMap 和 ArcInfo Workstation 的使用是不相同的。地图文档注释却根本不使用 pseudo items。因此，当在 ArcMap

中显示和转换 coverage 注释要紧记以下的重要规则：

coverage 注释中的 pseudo items 被 ArcMap 自动解译，coverage 注释的放置和显示基于 pseudo items 中的值被相应的调整。这意味着：当显示 coverage 注释时，对以下文本符号属性的修改不会影响地图中的注释：垂直和水平定线、x、y 偏移和文本大小（如果$SIZE 是非零）。

如果 coverage 注释中 pseudo item 的$SIZE 值等于零，修改文本符号的大小将影响地图中的注释。

Annotation size

pseudo item 中的$SIZE 存储 coverage 注释文本大小用 map units。由于 ArcGIS 存储注释文本大小用 page units（如 points），当转换的注释其$SIZE 不等于 0，ArcMap 将基于$SIZE 值为转换的注释自动计算文本符号的 page units 大小。在$SIZE 值等于 0 的情况下，在 Layer Properties 对话框中为 coverage 注释指定的文本符号大小将被转换的注释文本符号大小使用。

和其他注释一起显示，coverage 和 SDE3 注释的显示将不受数据组的基准比例的影响。这意味：不同于转换标注到注释要素，当转换 coverage 和 SDE3 注释时数据组的基准比例将不再重要。但是，当转换时，必须保证当前地图比例匹配目标注释组的基准比例。不同于转换标注到注释，如果这些比例不匹配将不会出现警告信息。

如何转换 convert ArcInfo Workstation coverage, CAD, SDE 3, VPF 或 PC ARC/INFO 注释到地图文档注释

添加 Convert Coverage Annotation 到 ArcMap

1、 在 ArcMap 中，单击 Tools 菜单，单击 Customize 命令。

2、单击 Commands 标签。

3、单击 Label category 命令。

4、从 Commands 列表中拖放 Convert Coverage Annotation 命令到工具条或菜单上。

5、单击 Customize 对话框中的 Close 按钮。

转换 convert ArcInfo Workstation coverage, CAD, SDE 3, VPF 或 PC ARC/INFO 注释到地图文档注释

1、 打开 ArcMap，添加需要转换为地图文档注释的 ArcInfo Workstation coverage, CAD, SDE 3, VPF 或 PC ARC/INFO 注释要素类。

2、 按照导入后需要的显示方式来设置注释的显示。

3、 根据上面的详细步骤添加 Convert Coverage Annotation 工具到 ArcMap。

4、 如果将转换注释到一个已存的注释组，放大到它的基准比例。

5、 单击 Convert Coverage Annotation 按钮。

6、 检查需要转换的 ArcInfo Workstation coverage, CAD, SDE 3, VPF 或 PC ARC/INFO 注释。可以一次转换多个注释类，但是它们将都被转换到一个单一的注释组。所有注释将具有相同的基准比例，这是因为在一个注释组的所有注释有相同的基准比例。

7、 单击 Into the Map 选项。

8、 为了转换注释到一个新的注释组，输入新注释组的名字。为了转换注释到一个已存的注释组，单击向下的箭头，单击这个注释组。

9、 单击 Convert。

10、转换完成，单击 Close 关闭 Convert Coverage Annotation 对话框，或重复上面的步骤转换更多的 ArcInfo Workstation coverage, CAD, SDE 3, VPF 或 PC ARC/INFO 注释。

提示：

如果指定一个已存的注释组为目标组，转换的注释将被插入到已存的注释组中。

可以使用这个工具导入到一个 geodatabase 注释要素类。但是，推荐使用 Import Coverage Annotation 工具导入 coverage 注释要素类到 geodatabase。

4、4、7、14 插入注释要素类

关于插入注释要素类

可以用 Append Annotation 要素类工具插入注释要素类。当插入注释要素类时，来自每个要素类的注释成为新注释要素类中的一个注释类。

如何插入注释要素类

1、 在 ArcCatalog 目录树中，单击 Toolboxes，单击 System Toolboxes，单击 Data Management Tools，单击加号展开 Feature Class。

2、 双击 Append Annotation 要素类工具。

3、 对于 Input features，移动鼠标至需要插入的要素类，添加它们到列表中。

4、 对于 Output feature class，移动鼠标至新注释要素类的位置，指定它的名字。

5、 对于 Reference Scale，提供了一个默认比例。为了改变它，输入一个新的或从下拉列表中选择一个。仔细选择 reference scale，由于选择后就不能修改了。

6、对于'Require symbol to be selected from the symbol table'选项，可以看编辑注释要素类属性。

7、如果插入到标准注释要素类，跳至下一步。

8、对于'Create annotation when new features are added'和'Update annotation when feature's shape is modified'选项，可以看编辑注释要素类属性。

9、单击 Environments。

10、展开 Geodatabase Settings。

11、如果导入到一个 file 或者 ArcSDE geodatabase，需要使用一个自定义存储关键字建立要素类，输入这个关键字。

12、对于 Output Spatial Grid 1，如果导入到一个 persoanl 或 ArcSDE geodatabase，并知道数据的的最佳空间参考索引网格大小，用 map units 指定它。

13、如果导入到一个 ArcSDE geodatabase，还有另外的网格大小，输入它们。

14、单击 OK。

15、在 Append Annotation 要素类工具上，单击 OK。

提示：

如果是插入到连接要素注释要素类，它们必须都被连接到同一个要素类。

不能插入一个使用 standard label engine 的连接要素注释要素类到使用 Maplex Label Engine 的注释要素类。所有输入的连接要素的注释要素类必须使用同一个标注引擎。

4、4、7、15 转换 geodatabase 注释到 coverage 注释

关于转换 geodatabase 注释到 coverage 注释

ArcMap 和 ArcCatalog 都可以转换存储于 geodatabase 注释要素类的注释到 coverage 注释。Geodatabase 注释要素类中的属性将被转换到一个相应的 coverage 注释子类，Text Attribute Table (TAT)。coverage 注释的 pseudo items ($ALIGN, $FIT, $ID, $JUSTIFY, $LEVEL, $OFFSETX, $OFFSETY, $SIZE, $SYMBOL, $TEXT, and $WORD)也将被生成。

每个注释要素的大小将从 unit points 转换为相应的 map units 大小。

ArcMap 提供了很多文本符号属性，在 ArcInfo Workstation 中是没有的，如填充样式，高级背景选项和阴影。因此，只有那些 ArcInfo Workstation 支持的属性被保留了。

在 ArcMap 中转换 geodatabase 注释的命令可以通过 Customize 对话框获得。ArcCatalog 同样有转换 geodatabase 注释到 coverage 注释的一个工具。这个工具还增加了批转换和合并注释要素类的功能。

如何转换 geodatabase 注释到 coverage 注释

添加 Convert Geodatabase Annotation 命令到 ArcMap

1、 在 ArcMap 中，单击 Tool 菜单，单击 Customize 命令。

2、 单击 Commands 标签。

3、 单击 Label category。

4、 从 Commands 列表中将 Convert Geodatabase Annotation 命令拖放到任何工具条或菜单上。

5、 单击 Customize 对话框的 Close 按钮。

在 ArcMap 中转换 geodatabase 注释到 coverage 注释（仅能在 ArcInfo 和 ArcEditor 中）

1、 添加需要转换的 geodatabase 注释要素类到地图。

2、 单击 Convert Geodatabase Annotation 命令（使用上面的步骤添加到 ArcMap）

3、 检查需要转换的注释要素类。可以转换多个 geodatabase 注释类到一个单一的 coverage （每个将被转换到一个相对应的注释子类中）。

4、 单击 Browse 按钮来浏览一个已存的 coverage。

如果没有已存的 coverage 注释子类，注释将被转换到一个新建的 coverage 注释子类。这个子类的名字将和输入的 geodatabase 注释要素类的名字相同（长度限制在 13 个字符）。

5、 单击 Convert。

6、 转换完成后，单击 Convert Geodatabase Annotation 对话框的 Close 按钮。

提示：

通过一个选择集或选中 Convert Geodatabase Annotation 对话框中的 Convert only selected annotation 选项，可以转换 geodatabase 注释要素类中要素子集。

当转换到一个已存的 coverage 注释子类，将会被询问是否需要删除已存的注释。单击 Yes，在转换前删除子类中所有的已存注释。单击 No，保留 coverage 注释，插入它到转换的 geodatabase 注释。单击 Cancel，保留已存的 coverage 注释，停止转换。

当输出 coverage 建立时，一个相应的文本文件也在相同的目录中建立，可以使用这个文本中信息，这个文本是为 ArcInfo Workstation (ArcPlot and ArcEdit)使用设置的。这个文件包含注释要素中惟一使用的符号列表。它为每个符号指定了字体、样式、颜色等，这个文件被命名为<output coverage>_<subclass>.sym。

可以在 http://arcscripts.esri.com/找到一个 AML 的例子，它使这个过程自动化，输入检索字"Annotation"，选择语言，AML and Software：ArcInfo Workstation 。

在 ArcCatalog 中转换 geodatabase 注释到 coverage 注释(仅对 ArcInfo)

1、在 ArcCatalog 中，在需要导出的注释要素类上，单击右键。

2、鼠标指向 Export。

3、单击 To Coverage。

4、添加其他需要导出的注释要素类。

5、移动鼠标至需要导出的 coverage。

6、单击 OK。

4、4、8 拓扑

4、4、8、1 拓扑概要

如果有一致性的要素（如共享相同的坐标位置，边界或者节点），使用一个 geodatabase 拓扑可以帮助更好地管理地理数据。

大多数用户都很关心数据的空间完整性，geodatabase 拓扑可以帮助更好地管理数据的完整性。使用拓扑提供了执行数据完整性检查的强有力机制，将帮助用户更好地验证和维护

geodatabase 中要素表达。

还有许多用户喜欢在各种分析操作中使用拓扑（查找相邻要素、处理要素之间的一致边界、沿着相连要素移动）。拓扑增强了 GIS 更丰富的分析功能。

拓扑是对点、线和多边形要素如何共享点或边的安排。使用拓扑的目的在于：

- 约束要素如何共享点或边。例如，相邻的多边形如地块有共享边；街道中心线和人口普查区共享点和边；相邻的土地多边形共享边等。
- 定义和强化数据完整性规则（如在两个多边形间不能有缝隙；要素间应该没有重叠等）。
- 支持拓扑关联查询和搜索（搜索要素的邻接性和连通性）。
- 支持复杂编辑工具（执行拓扑约束的数据模型工具）。
- 从无组织的几何图形建立要素（由线建立多边形）。

以下的主题包括关键的概念，ArcGIS 支持拓扑。还包括了在 ArcGIS 中建立和使用拓扑的许多通用工作流。

从已存数据建立一个拓扑的过程总结为如下的步骤：

4、4、8、2 拓扑基础 (仅对 ArcInfo ArcEditor)

一个 ArcGIS 拓扑是规则和行为的集合，它模拟了点、线和多边形如何共享一致的点或边。例如：

- 相邻要素，如两个县，在它们之间有一个公共的边界，它们“共享”这个条边。
- 在每个州的县多边形集合必须完全覆盖州多边形，与州边界共享边。

有很多拓扑规则和行为的例子普遍地用于管理 geodatabase 中几何特征的一致性。

Why topology?

拓扑一直都是数据管理和完整性的一个关键的 GIS 必需。通常，拓扑数据模型表示空间对象（点、线和面积要素）为拓扑简单图－nodes, faces, 和 edges。这些简图，加上它们彼此间的关联和所表示的要素的边界，通过用平面的拓扑元素表示要素几何特征来定义。

从根本上说，拓扑被用于保证数据质量和帮助数据编辑。拓扑也用于多种情况下分析空间关系－－如用相同的属性值来消除相邻多边形之间的边界或者在一个拓扑图形中沿着一个网络的元素穿越。

拓扑也可以用于模拟多个要素类如何集成，有时称这个为要素类的垂直统一管理（*vertical integration*）。

通常地，在以下情况下使用拓扑：

- 管理一致性的点或边（限制要素如何共享点或边）。例如，相邻的多边形如地块有共享边；街道中心线和人口普查区共享点和边；相邻的土地多边形共享边等。
- 定义和强化数据完整性规则（如在两个多边形间不能有缝隙；地块间应该没有重叠等，道路中心线应该与它们的端点相连接）。
- 支持拓扑关联查询和操作（如，提供了识别相邻和连接的要素的功能，查找共享边，沿着一系列相邻的边移动）。
- 支持复杂编辑工具，它执行拓扑约束的数据模型工具（如编辑一个共享边和更新共享公共边的所有要素的功能）。
- 从无组织的几何图形建立要素（如由线建立多边形的功能，有时称为"spaghetti"）。

在拓扑中要素共享几何要素的方式

在一个拓扑中要素可以共享点和边，以下是相邻要素的一些例子：

- 区域要素可以共享边界（polygon topology）
- 线要素共享端点(edge–node topology)

另外，共享的点和边可以使用一个 geodatabase 拓扑协作要素类管理。例如：

- 线要素可以与另一个线要素共享线段。如，地块可以嵌套在一个街区内：
- 区域要素可以与另一个区域要素有一致性。
- 线要素可以与其他点要素共享端点的顶点(node topology)。
- 点要素可以与线要素有一致性(point events)。

两种观点：要素和拓扑要素

以下的解释说明了一个多边形数据层是如何被描述和使用的：

- 作为地理要素的集合（points, lines, 和 polygons）；或
- 作为拓扑元素的一个图形（nodes, edges, faces, 和 their relationships）。

这意味着有两种选择处理要素，一种要素被坐标定义，另一种要素被它们的拓扑元素的顺序图表示。

Geodatabase 拓扑的演化

"Arc-node" 和 "Geo-relational"的来源

ArcInfo coverage 用户有一段很长的历史，高度评价了拓扑在维护数据的空间完整性方面所扮演的角色。

以下是 ArcInfo coverage 数据模型的元素：

在一个 ArcInfo coverage 中，要素边界和点被存储在一些主文件中，这些主文件是由 ArcInfo Workstation 管理和拥有的。"ARC"文件存储线性或多边形几何为拓扑边界，它被称为"arcs"。"LAB"文件存储点位置，它被作为多边形标注点或独立的点要素，如一个水井要素层。其他的文件用于定义和存储每个边界和多边形之间的拓扑关系。

例如，称为"PAL"文件("Polygon-arc list")列出了每个多边形中 arcs 的顺序和方向。在 ArcInfo 中，程序逻辑被用于组织显示、分析和查询操作的多边形的坐标。在 PAL 文件中边界的顺序列表被用于查询和组织存储在 ARC 文件中的边界坐标。

Coverage 数据模型的优点：

- 它使用一个简单的结构来维护拓扑
- 它使由多个要素共享的边界仅被数字化和存储一次
- 它可以表示巨大的多边形（含有数千个坐标）这是因为多边形是被定义为一个边界的顺序集（称“arcs”）
- Coverage 的拓扑存储结构是以人文本。它的物理拓扑文件可让 ArcInfo 用户容易理解

Coverages 的一些缺点：

- 一些操作比较慢，这是因为当需要使用要素时，许多要素被闲混装配。这包括所有的 polygons 和 multi-part 要素，如 regions（coverage 术语称为 multi-part polygons）和 routes（coverage 术语称为 multi-part line）。
- 拓扑要素（如 polygons、regions 和 routes）都是在拓扑建立好之后才可以使用的。如果边线被编辑，拓扑需要重建。一般地，当对一个拓扑数据集中的要素进行编辑，不管采用的是何种存储模型，必须执行几何分析运算来重建拓扑关系。
- Coverage 仅限于单用户编辑。由于需要保证拓扑图形的同步性，在同一时间只能有一个用户更新拓扑。

Shapefiles 和简单几何要素的存储

早在 20 世纪八十年代，coverages 被视为是以前基于多边形和线系统（多边形被存储为完整的环）的一个巨大进步。在过去的系统中，一个要素的所有坐标被存储在每个要素的几何特征中，在 coverages 和 ArcInfo 诞生以前，这些简单的多边形和线结构被使用。这些数据结构是简单的，但是有“数字化边界两次”的缺点。就是，具有共享边线的相邻多边形的坐标的两个拷贝将被存储在每个多边形的几何特征中。这个主要的缺陷是当时的 GIS 软件

不能存储共享边线的完整性。另外，存储成本也是巨大的，每个字节的存储空间都是相当珍贵的。在 80 年代，一块 300MB 的硬盘像一台洗衣机那么大，价格却是$30,000！存储坐标的两种或更多种表达是昂贵的，计算占用了太多的机器时间。因此，coverage 拓扑的使用具有真正的缺点。

在 20 世纪 90 年代，人们开始不断对简单几何特征结构感兴趣，这是因为磁盘存储和硬件价格普遍地下降，然后计算机的计算速度却提高了。同时，现存的 GIS 数据集比较容易获取。GIS 用户的工作从基本的数据编辑活动转变为包括数据使用、分析和共享等工作。

用户需要更快性能的数据使用（例如，不需要花费太多的计算机时间得到需要使用的多边形几何图形，可以尽可能快传递 1200 多个多边形的要素坐标）。所有要素几何特征的随意获取都很有效。可以有数千用户使用，大量的数据集随意可以获取。

在这个时候，ESRI 开发和颁布了其 ESRI Shapefile 格式。Shapefiles 使用一种简单的要素坐标的存储模型。每个 shapefiles 代表一个单独的要素类（点、线或面的），为要素的坐标使用一种简单的存储模型。Shapefiles 可以在 ArcInfo coverages，以及其他各种 GIS 系统中建立。它们作为一种事实标准被广泛使用，至今仍被大量使用和配置。

数年后，ArcSDE 倡导了一种在关系数据库表中使用的相似的简单的存储模型。一个要素表在每行存储一个要素，其中一列为几何特征，还有其他的要素属性列。

下面是一个州多边形的要素表实例。每一行代表一个州。Shape 为每个州存储了多边形几何特征。

这种简单要素模型很适合 SQL 处理引擎。由于关系数据库的使用，在没有降低性能的情况下，GIS 数据达到了空前的规模和用户数量。人们开始利用 RDBMS 进行 GIS 数据管理。

Shapefiles 变得普遍存在，使用 ArcSDE，这种简单要素机制成为 RDBMS 中基本的要素存储模型（为了支持互操作性，ESRI 是 OGC 和 ISO 简单要素规范的主要创作者）。

简单要素存储有很明显的优点：

- 每个要素的几全部几何特征被存储在一条记录中。不需要任何组装。
- 数据结构（物理 schema）是简单、快速和可升级的。
- 它很容易让程序员写各类接口，
- 它是互操作性的。许多写个简单的转换就可以从大量其他格式转换为这种格式，或者这种简单的格式转换为其他格式。Shapefiles 作为一种数据使用和交换格式而被广泛地使用。

它的缺点就是维护数据完整性，数据完整性可以由拓扑随时来提供的，但是对于简单要素来说拓扑却不容易实现。因而，用户总是使用一种数据模型来编辑和维护（如 coverages），使用另一种来部署（如 shapefiles 或 ArcSDE layers）。

用户开始使用这种混合的方法来编辑和数据部署，例如，用户使用 coverages、CAD 文件或其他格式来编辑数据。然后，他们转换数据为 shapefiles 来部署和使用。因此，尽管简单要素结构是一个优秀的直接使用格式，但是它不支持拓扑编辑和共享几何要素的数据管理。直接使用数据库将使用简单要素结构，但是其他的拓扑形式被用于编辑。这对于部署是有利的。但是其缺点就是数据变得过时，必须更新。它可以工作，但有一个滞后的时间用于信息更新。Bottom line—拓扑被丢失。

使用简单要素几何特征，并且使得拓扑可以被应用在这种简单要素、开放数据结构上是 GIS 必需的，也是 geodatabase 拓扑模型所实现的一种机制。这意味着：用户可以两全其美－－一个事务数据模型可以用于拓扑查询、共享几何要素的编辑、丰富数据建模和数据完整性，除此之外，还有一个简单、高度的可扩展性的数据存储机制，它是基于开放的、简单的要素几何特征的。

直接使用数据模型是快速、简单和有效的。它可以被许多用户同时直接编辑和维护。

ArcGIS 中的拓扑框架

有效地，拓扑应该不只是被考虑为一个数据存储问题。完整的解决方案包括：

- 完整数据模型（对象、完整性规则、编辑和验证工具、一个拓扑和几何要素引擎，它可以处理任何大小和复杂性的数据集，一个丰富的拓扑操作集合、地图显示和查询工具）。
- 一个开放的存储格式，为简单要素使用一套记录类型，使用一个拓扑接口来查询简单要素、检索拓扑要素、操作它们的空间关系（如查找相邻区域和它们的边线）。
- 提供要素（points, lines, and polygons）和拓扑元素（nodes, edges, and faces）以及它们相互之间的关联关系的功能。
- 可以支持以下内容的一种机制：

1）含数百万个要素的大数据集。

2）执行多个并发编辑的编辑和维护功能。

3）即可使用，对任何要素几何特征总是可用的。

4）支持拓扑完整性和行为。

5）一个为多个用户和多个编辑运行较快的系统。

6）一个灵活和简单的系统。

7）一个利用 RDBMS SQL 引擎和是事务框架的系统。

8）一个可以支持多编辑、长事务、历史归档和复制的系统。

在一个 geodatabase 拓扑中，验证过程鉴定要素之间的共享坐标（可以是同一个要素类，也可以是不同要素类之间）。聚类运算用于保证共享坐标具有相同的位置。这些共享的坐标被存储为每个简单要素的简单几何特征的一部分。

这可以得到较快地、可扩展的拓扑元素的查询（nodes, edges, and faces），还有另外一个好处，可以扩展利用 RDBMS 的 SQL 引擎和事务管理框架。

在编辑和更新过程中，当要素被添加时，它们是立即可用的。地图上被更新的区域称为“脏区域”，在对每个要素类进行更新时，这些更新区域将被标记和跟踪。任何时候，用户可以选择拓扑地分析，验证“脏区域”来生成干净的拓扑。只有“脏区域”的拓扑需要重建，以节省处理时间。

结果发现：拓扑元素(nodes, edges, faces) ,它们之间的关系，还有它们的要素可以有效地发现和组装。这样做有几个好处：

- 简单要素几何特征用于存储要素。这种存储模型是开放的，有效的，可以达到大容量和多用户的规模。
- 这种简单要素数据模型是事务和多用户的。相比之下，以前的拓扑存储模型不具有规模型，在支持多个编辑事务处理和大量其他 GIS 数据管理流方面有困难。
- Geodatabase 拓扑全面支持所有的长事务和 geodatabase 版本功能。Geodatabase 拓扑不需要被分割，许多用户可以并发编辑拓扑数据库－如果必要的话甚至编辑相同要素的独立的版本。
- 要素类可以增加到任何大小（亿万个要素），不影响性能。
- 这种拓扑执行是加成的。可以把这种添加到一个空间上相关的要素类的已存的 schema 中。另一种方式就是：必需重新定义和转换现有的要素类到存储拓扑元素的数据 schema 中。
- 只需要一种数据模型用于几何要素编辑和数据使用，不需要两种或更多。
- 它是互操作的，这是因为所有的几何要素特征的存储遵循 OpenGIS Consortium 和 ISO 简单要素规范。
- 数据模型是比较接近自然的，这是因为它是基于用户要素的（如地块、街道、土地类型和流域）而不是拓扑元素（如节点、边线和面）。用户考虑的实际要素的完整性规则和

行为，而不是拓扑元素的完整性规则。例如，地块行为是怎样的？这将使各类地理要素的建模更强。它将提高思考街道、土地类型、人口普查单位、流域、铁路系统、地质、森林、地形地貌、物理特征等等。

- Geodatabase 拓扑提供了与存储拓扑执行相同的信息内容－－要么存储了拓扑线图形元素，查找要素几何元素（如 ArcInfo coverages），要么存储要素几何元素，查找拓扑元素和关系（如 geodatabase）。

在用户需要存储拓扑元素的情况，建立和提交拓扑、以及它们的关系到表，用于各类分析和互操作目的是容易的（如用户需要提交要素到一个 Oracle Spatial warehouse，它是用于存储拓扑元素的表）。

从实效层面来说，ArcGIS 拓扑执行有效。它可以达到非常大的 geodatabase 和多用户系统的规模，而没有性能的损失。它包括丰富的用于建立和维护 geodatabase 中拓扑的验证和编辑工具。它包含丰富的和灵活的数据建模工具，它可以使用户组装实际的，可操作的系统，可以在文件系统上、任何关系数据库和许多 schema 上。

4、4、8、3 ArcGIS 中的拓扑

在 geodatabase 中，拓扑是对点、线和多边形要素如何共享一致性几何特征要素的安排。例如，街道中心线和人口普查街区共享点和边线，相邻的土地多边形共享边界。

拓扑定义和强化了数据完整性规则（如在两个多边形之间不能有空隙）。它支持拓扑关系查询和搜索（如，邻接的或连接的要素的搜索），支持复杂编辑工具，从无组织的几何图形建立要素（由线建立多边形）。

选择拓扑不仅仅提供了一种数据存储机制。在 ArcGIS 中，拓扑包括以下的六个方面：

1、geodatabase 包括一个拓扑数据模型，它使用简单要素的开放存储格式（如点、线和多边形要素类），拓扑规则和共享几何特征的要素之间按拓扑集成坐标。这个数据模型包括定义完整性规则和参与拓扑的要素类的拓扑行为的功能。

2、ArcGIS 包括 ArcMap 中的拓扑数据层，它们用于显示拓扑关系、错误和异常。ArcMap 还包括一套丰富的工具用于查询、编辑、验证和拓扑错误校正。

3、ArcToolbox 包括一套全面的处理工具用于建立、分析、管理和验证拓扑。

4、ArcGIS 包括高级应用逻辑，分析和查找点、线和多边形要素类中的拓扑元素。这包括一套丰富的工具用于验证、查找、标识、编辑和解决拓扑图形和要素坐标。这些工具广泛用于整个 ArcGIS 的许多工作流和任务。

5、ArcMap 包括一个丰富的编辑和数据自动化框架，它用于建立、维护和验证拓扑完整性，执行共享要素的编辑。

6、ArcGIS 软件逻辑在 Desktop、Engine 和 Server 中都是可用的，它可以用于拓扑关系，操作邻接性和连接性要素，组装这些元素为要素。例如，标识共享一条公共边的多边形；列出连接在一个结点上的边线；从当前位置沿着边线查找；添加一条新线，加入拓扑图形；在交叉点分割不同的线；建立结果结点、面和边线。

Geodatabase 拓扑元素

在 geodatabase 中，每个拓扑定义以下的属性：

- 新建拓扑的名字。
- 在拓扑处理操作中的聚类误差。聚类误差是用于涉及两种误差的一个术语：x、y 误差和 z 误差。默认的聚类误差是坐标精度的 10 倍。
- 要素类列表。首先需要一个需要将参与拓扑的要素类列表。所有的要素类必须是在同一个坐标系统中，并被组织在相同的要素数据集中。
- 每个要素类中坐标的相对精度级。如果一些要素类比其他要素类更精确，需要分配一个更高坐标精度级。这将用于拓扑验证和集成。一个低精度的坐标应该移动至较高精度坐

标的位置，当它们落入聚类误差中。最高精度的要素应该得到的值为 1，较低精度的要素类值为 2，精度更低的要素类值为 3，等等。

- 要素如何共享几何特征的拓扑规则列表。

ArcMap 中拓扑编辑和验证

在 ArcMap 中，有一个特殊地图数据层用于拓扑，它被用于显示拓扑编辑中的错误和拓扑状态（被编辑区域需要验证称为“脏区域”）。

在 ArcMap 中也有一个丰富的拓扑编辑环境和工作流。

拓扑规则

拓扑规则定义要素之间可允许的空间关系。定义的拓扑规则控制同一个要素类中要素之间的关系，不同要素类中要素之间的关系或者要素子类型之间的关系。

例如，"Must not overlap"规则用于管理同一个要素类中要素完整性。如果两个要素重叠，重叠部分被显示为红色（如下图显示的相邻多边形重叠的红色区域，两条线的重叠的红色线段部分）。

拓扑规则可以在要素类的子类型之间定义。例如，假定有街道线要素的两个子类型－normal streets（在两者的结点处连接其他街道）和 cul-de-sac streets（在一个结点处只有一个出路）。拓扑规则要求街道要素与其他街道要素在两者端点处相连接，除了街道属于 cul-de-sac 子类型外。

使用要素的空间关系和行为定义拓扑规则

空间关系表达随着空间表达的行为规则要素如何具体地共享一致性几何特征。例如，一些普通的空间关系和规则包括：

- 地块不能重叠。邻接的地块具有相同边界。
- 河流线不能重叠，并且必须在端点处相互连接。
- 邻接的县具有公共的边线。县必须完全覆盖一个州，不能嵌套。
- 邻接的人口普查区具有共享的边线。人口普查区不能重叠，普查区必须完全覆盖区集。
- 道路中心线必须在它们的端点连接。
- 道路中心线和普查区共享一致的几何特征（边线和节点）。

以下是一个概念上的空间关系和完整性规则的观念，它们使用拓扑来管理。在左边的第一列，是一些普通的关系和规则，它们可以用于相同要素类中要素之间关系的定义。在第三列，定义了两个要素类中要素之间的关系。

这些情况中的每一个都定义了一种潜在的使用拓扑规则来维护数据完整性。

验证性能和规则效果

拓扑验证是计算密集型的－－需要处理的顶点数量越多，验证处理完成所需的时间就越多（例如，对于含几十万个要素的数据集需要几分钟，对于含有一千万个要素的数据集则需要几个小时）。在拓扑中增加规则的数量不会显著地增加验证处理的时间。

初始的拓扑验证检查所有参与拓扑的要素类中要素。这种检查可能需要一些时间，但是接下来只检查编辑过的区域，这些区域称为“脏区域”。

聚类处理

建立拓扑关系包括集成要素顶点的坐标位置，使它们具有一致性。这可以出现在相同要素类中的要素之间，也可以是参与拓扑的不同要素类之间。

一个聚类误差用于集成顶点。所有落在聚类误差中的顶点在验证处理中会轻微移动。默认的聚类误差是最小的可能聚类误差，是基于数据集定义的精度。默认的聚类误差是 0.001 米，是 x、y 精度的 10 倍（它定义了存储坐标数值精度大小）。

两种聚类误差：x、y 误差和 z 误差

在 ArcGIS 中，一对聚类误差用于集成顶点：

X、y 误差用于发现彼此的水平距离落在误差中的顶点。

Z 误差用于辨别 z 高度或高程是否落在彼此的误差中，是否应该被聚类。

坐标是如何聚类

X、y 误差应该比较小，因此只有那些挨足够紧密的顶点（彼此距离在 x、y 误差中）才会被聚类当坐标在误差中，它们被认为是一致的，需要被调整到共享一个相同的位置。

这样，x、y 误差也定义了在聚类过程中一个坐标需要移动的距离。因此，如果坐标在 xy 误差中，不管是 x 维还是 y 维，坐标都可以被聚类。可以看下面的图表。如果记得几何学和勾股定理，聚类坐标中的最大距离等于根号 2 倍的 xy 的误差。

默认 x,y tolerance

默认的 x、y 误差倍设置为 0.001 米或数据集坐标系统单位的相等量。例如，如果坐标系统是以 feet 记录，默认值为 0.003281 feet (0.03937 inches)。默认值是 10 倍的默认的 xy 精度，在大多数情况下这是推荐使用的值。如果坐标是经纬度，默认 xy 误差是 0.0000000556 degrees。

有一个选项用于设置：当数据的精度比较小时设置大的误差值，精度很高的数据集设置比较小的误差查。

验证和聚类中使用的算法

当拓扑中一个要素的顶点落在了另一个要素的一条边的 xy 容限值，拓扑引擎会在这条边上建立一个新的顶点，允许要素在聚类过程中被几何集成。

拓扑验证中当聚类要素顶点时，了解要素的几何图形是如何被调整的是重要的。任何参与一个拓扑的要素类的所有顶点都可以潜在的被移动，如果它们落在了另一个顶点的 xy 容差值中。高级别要素的顶点并不移动到低级别的要素中，但是相等级别的要素在几何方面是平等的。

注意：x、y 容差值并是规定用于生成几何图形。相反，它是用于在拓扑处理中集成线和边界，这意味着：集成落在很小距离范围内的坐标。这是因为坐标可以在 x 和 y 坐标上像聚类误差移动差不多，处理数据集使用聚类误差的命令可以解决许多潜在的问题。

顶点最大移动距离

聚类处理是通过在地图上移动来实现的，标识落入彼此 x、y 容限值的坐标聚类。ArcGIS 使用这种运算来发现、清理和管理要素之间的一致性几何特征。这意味着：一致性几何特征元素的坐标是协同定位的（co-located）（被捕捉到相同的位置）。这是许多 GIS 操作和概念的基础。

聚类处理的结果，要素的顶点可以有两种方法潜在移动。

1、 容限值用于计算水平距离和垂直距离来找到在容限值内的坐标。一个坐标在聚类处理中可以移动到新位置的最大距离是根号 2 倍的 xy 容限值。

2、 聚类运算是迭代的。因此以下这种情况是可能的，即一些坐标已经移动，但它们又落入其他顶点的聚类误差中，它们还可以移动，最大距离是根号 2 倍的 xy 容限值。

Tips

以下是聚类误差的一些有用的提示：

1、 一般地，可以使用 x、y 容限值 10 倍的 x、y 的精度。

2、 典型 x、y 容限值是数量级小于数据捕捉的真实精度。例如，当要素的坐标可以精确到 2 米，默认的 x、y 容限值为 0.001 米。

3、 为了保持移动很小，保持 x、y 容限值很小。但是，x、y 容限值太小（如 2 倍的 xy 精度或更小）可能不能合适集成一致性边界的线要素。

4、 相反地，如果 x、y 容限值太大，要素的坐标可能彼此叠置在一起。这可能降低要

素边界表达的精度。

5、 x,y tolerance 决不能接近于地图精度标准。

6、 在拓扑中，可以设置每个要素类的坐标级别。

7、 通常，需要控制哪些要素类在聚类操作中最有可能被移动。

坐标级别

在 geodatabase 中每个要素类的坐标级别控制验证过程中要素顶点的移动。坐标级别帮助控制要素顶点在落入聚类误差时如何移动。要素顶点落入聚类误差被认为是具有相同的位置和共位。

当不同的要素类具有不同的坐标精度时，如当一个坐标是通过测量或者 GPS 收集的，另一个是通过一个精度稍低的数据源数字化得到的，坐标级别允许保证可靠精度的坐标顶点被放置，而较低精度的顶点被移走。相等级别的顶点位置在落入聚类误差时是被几何平均。

确保赋予合适顺序的坐标级别。最高精度的要素级别为，较低的级别为 2，依次类推。

下表提供了设置坐标级别的一个例子：

Z 聚类误差和级别

模拟 3 维表面或者建筑物的要素类具有一个 z-value，表达每个顶点的高程。与用 x、y 聚类误差和级别控制要素类在水平方向上如何被捕捉一样，如果一个拓扑中具有模拟高程的要素类，可以通过 z 聚类误差和级别控制垂直方向上一致点如何被捕捉。

Z 聚类误差定义了两个一致顶点之间的高程或者 z-value 的最小差异。具有 z-value 的顶点，在 Validate Topology 过程中，如果落入 z 聚类误差中，将被捕捉在一起。

如果是模拟城市建筑物，两个建筑物可能彼此相邻，共享一条公共的边界。

4、4、8、4 设计一个 geodatabase 拓扑

在 ArcGIS 中创建一个 geodatabase 拓扑前，先按照以下的步骤设计一个 geodatabase 拓扑：

1、 列出需要的共享几何图形的要素类；

2、 指定每个要素类的空间表达；

3、 列出将要共享几何图形和一起被编辑和维护的要素类。例如，如果在其中一个要素的进行几何编辑，那么另一个也将被更新。

4、 组织这些要素类到一个要素数据集中。

5、 指定每个单独要素类中的元素之间的拓扑规则。例如，地块可以是单部件或者多部件多边形。相邻的地块共享几何图形，地块不能重叠。

6、 指定要素类之间的拓扑规则。例如，定义地块多边形的线要素必须被地块边界线覆盖。

7、 确定每个要素类中的坐标精度级别。

8、 用要素的复本建立一个测试的 geodatabase 来测试设计的拓扑。

9、 改善和调整拓扑设计，直到具有一个良好的实现。

4、4、8、5 建立一个拓扑

创建一个拓扑

ArcGIS 提供了大量定义和创建拓扑的方法。主要的方法是使用 ArcCatalog 中的工具。一旦知道如何创建拓扑，通过使用一套 geoprocessing 工具将自动化这项工作。

使用 ArcCatalog 创建拓扑

1、 在需要添加拓扑的要素数据集上，单击右键，指向 New。从列表中选择 Topology，打开 New Topology 向导。

2、 在第二个面板中，命名新的拓扑，指定聚类误差。

3、 接下来，选择参与拓扑的要素类。将显示要素数据集中所有要素类的一个列表。根

据拓扑设计选择任何或者全部要素类。

4、 单击下一步，在这个对话框中设置拓扑中每个要素类的坐标精度级别。

5、 接下来，添加拓扑规则，它将帮助你结构化要素之间的空间关系，以及控制和验证要素如何共享几何图形。

6、 单击 Complete，完成建立拓扑的向导。

7、 现在已经添加拓扑到要素数据集中。将被询问是否需要现在验证要素数据集中拓扑。如果要素类中有数据，可以选择 Yes。

也可以使用 ArcCatalog、ArcMap 或者 Geoprocessing 工具来验证拓扑。

关于拓扑验证

验证拓扑包括以下内容：

- 根据聚类误差，找到共享几何图形（具有公共的几何坐标）的要素；
- 运行一系列完整性规则来检查违反定义的拓扑规则；
- 在要素数据集中建立一个潜在拓扑错误的错误日志。

在初始化验证后，它是针对要素数据集中的所有要素，接下来的验证可以仅针对要素数据集中变化的区域。在 ArcMap 中，可以验证整个范围的拓扑，地图可见范围的拓扑或者选择范围区域的拓扑。也可以在 ArcCatalog 和 geoprocessing 工具中验证整个的拓扑。

开始操作新建的拓扑

一旦已经创建了拓扑，并运行了初始化验证，就可以开始操作它了。

- 使用 ArcMap Topology Error Inspector 核查拓扑错误；
- 使用 ArcMap 修改拓扑错误或者标记为一个异常。

使用 geoprocessing 工具创建拓扑

ArcToolbox 包含一系列拓扑处理工具。Topology toolset 是位于 Data Management 工具箱中。

工具箱中的工具可以用于编写创建和修改 geodatabase 的拓扑的脚本。这些脚本对一些任务的自动化和重复工作流的创建是很有用的。

以下是用 geoprocessing 工具创建拓扑的步骤：

1、 进行拓扑设计。

2、 使用 Create Topology 工具创建一个要素数据集的拓扑。

3、 使用 Add Feature Class To Topology 工具添加要素类到拓扑，并设置它们的坐标级别。

4、 使 Add Rule To Topology 工具添加要素如何共享几何特征的拓扑规则。

5、 使用 Set Cluster Tolerance 工具设置聚类误差。

6、 一旦建立了拓扑定义和拓扑规则，使用 Validate Topology 工具来验证新建的拓扑。

4、4、8、6 验证拓扑

一旦已经建成拓扑，并加载了数据，那么在任何时候都可以对拓扑中的要素类数据进行拓扑验证。验证拓扑执行以下的处理的任务：

- 根据聚类误差，找到共享几何图形（具有公共的几何坐标）的要素；
- 插入公共坐标顶点到共享几何图形的要素；
- 运行一系列完整性规则来检查违反定义的拓扑规则；
- 在要素数据集中建立一个潜在拓扑错误的错误日志。

在 ArcMap 中编辑时，可以验证整个范围的拓扑，地图可见范围的拓扑或者选择范围区域的拓扑。也可以在 ArcCatalog 和 geoprocessing 工具中验证整个的拓扑。

验证和脏区域

一旦新建拓扑已经被验证，接下来编辑被标记为脏区域，它确定为需要验证的要素数据

集的子集。这将节约时间和提高性能，因为仅是需要重新验证的区域被处理。

在 ArcCatalog 中验证拓扑

在 ArcCatalog 中，选择目录树上的拓扑，单击右键，从上下文快捷菜单中选择 Validate。

如果是一个新建的拓扑，将验证整个范围的拓扑，在大多数情况下，仅在有更新的脏区域进行验证。

在 ArcMap 中验证

可以在 ArcMap 中的编辑过程进行拓扑验证。

1、 首先，启动编辑。

2、 通过单击 Toolbars 上的 View 菜单，单击 Topology 工具条来添加拓扑工具。

3、 通常有三个拓扑选项可用。

使用 geoprocessing 工具验证拓扑

ArcToolbox 包含一系列拓扑处理工具。Topology toolset 是位于 Data Management 工具箱中。

工具箱中的工具可以用于编写创建和修改 geodatabase 的拓扑的脚本。这些脚本对一些任务的自动化和重复工作流的创建是很有用的。

拓扑验证的技巧：

- 可以自由选择何时和多久来验证拓扑。通常，每个拓扑必须仅全面验证一次。所有其他的验证将仅是对脏区域的重新验证。
- 在编辑数据时，如果修改了拓扑定义，ArcGIS 将自动跟踪脏区域。当重新验证拓扑时，ArcGIS 将自动确定脏区域必须被验证。
- 应该多久验证一次拓扑呢？最好在每次对要素数据集编辑后重新验证一次拓扑。也可以在编辑过程中选择验证一部分数据集，然后可以发现和修改其中的拓扑错误。
- 如果修改拓扑定义，可能需要重新验证拓扑。例如，如果修改了拓扑规则集或者坐标级别，整个拓扑将被标记为脏区域，在下次运行验证时将被重新验证。
- 参与拓扑的要素类总是在 ArcGIS 中可用，不管拓扑是否被最近验证过。如果拓扑还没有被验证，一些要素可能没有遵循完整性规则，但是，它们可以使用－－无错或者含有错误。
- 当验证拓扑时，它生成一个错误报告，因此可以了解错误，修改严重的错误。在这种情况下，拓扑帮助了解数据集的完整性和在各类应用中的适宜性。

4、4、8、7 在 ArcMap 中绘制和显示拓扑

拓扑可以在 ArcMap 中绘制和使用，以一个要素数据集中的要素类集合或者一个拓扑数据层。可以通过添加一个地图数据层和选择拓扑来创建一个拓扑数据层。例如：

访问一个拓扑的显示属性和其他任何地图数据层一样－－在数据层名字上，单击右键，单击 Properties。然后，单击 Symbology 标签来修改拓扑的显示属性。

这个面板提供了基于拓扑中的数据和规则的错误、异常和脏区域的状态的选项。

4、4、8、8 编辑拓扑

ArcMap 编辑器包含大量高级的管理和编辑拓扑的工具和功能。以下是对拓扑编辑的一个简短描述：

1、 启动 ArcMap，添加需要编辑拓扑的要素类地图数据层。也可以添加需要编辑的拓扑的地图数据层。

2、 启动编辑，设置拓扑中的其中一个要素类作为编辑的目标图层。

3、 通过单击 Toolbars 上的 View 菜单，选择 Topology 工具条添加拓扑工具条。

4、 开始编辑任务。

5、 一旦完成编辑，保存编辑，验证拓扑。

6、 检查、诊断、修复任何拓扑错误。然后验证保存编辑结果。

4、4、8、9 拓扑和版本化 geodatabase

操作版本化 geodatabase 中的拓扑是相对简单的。但是，首先需要理解一些基于版本的工作流和在版本化环境中拓扑的行为。

修改一个版本化 geodatabase 的 schema

修改一个版本化 geodatabase 的数据集的任何 schema，首先注册它为非版本，修改完 schema，然后再次注册它为版本。

在一个版本化 geodatabase 中创建拓扑

在一个版本化要素数据集中建立一个新拓扑或者修改一个已存拓扑的 schema，需要按照以下的步骤进行。首先，必须反注册要素数据集为版本。然后，使用一般的创建步骤新建一个拓扑，最后，注册要素数据集为版本。

1、 在 ArcCatalog 中建立一个用户自定义工具条来操作版本化 geodatabase

- 在 ArcCatalog 的 Tools 菜单下，选择 Customize，，在 Customize 对话框中的 Toolbars 标签上选择 New，新建一个工具条。给新的 ArcCatalog 工具条一个名字。
- 单击 Commands 标签，然后，向下拉动滚动条，从 Categories 列表中选择 Geodatabase tools。你将在右边的 Commands 列表中看到一系列工具列表。
- 从 Customize 对话框中拖三个工具放到新工具条上。

  Register As Versioned

  Unregister As Versioned

  Compress Database
- 单击 Close 建立自定义工具条。

2、 反注册版本化要素数据集

- 首先，协调和提交 geodatabase 中这个要素数据集的默认版本。
- 运行 Compress Database 命令压缩所有的编辑到默认版本。
- 单击要素数据集，然后单击 Unregister As Versioned 命令。

3、 建立新拓扑。

4、 单击要素数据集，然后在自定义工具条上单击 Register As Versioned 命令。

版本化要素类中的脏区域

大量编辑人员可以同时编辑一个要素数据集和它的拓扑。

每个编辑人员将按照一个工作流验证他或者她编辑的拓扑，发现错误和修改错误（或者设置它们为异常）。

通常，多个编辑会话被合并到默认版本中。

值得注意的是当版本被协调时，新的拓扑错误可能发生，甚至每个编辑版本已经被验证，并且没有错误。

为了管理这些错误，版本化的拓扑具有特殊的错误处理和冲突检查规则，这将影响协调过程。

以下的部分描述了对脏区域、错误、异常和潜在冲突协调的结果。在每种情况下，结果都是基于父版本和子版本更新后的一个协调的。如果父版本在子版本被协调前没有被编辑，协调的结果将是子版本的数据。在每个例子中，Version2 是以 Version1 的一个子版本被创建的。然后两个版本都以例子中描述的方式被编辑；Version2 是被协调到 Version1 中。为了说明以下的例子，使用以下的图例：

在脏区域的协调结果可以被总结如下：

- 任何在父版本或者子版本出现的脏区域，但是它们在父版本和子版本创建前并不存在，将在协调结果中保持为脏区域。

- 在父版本中出现的任何脏区域和在子版本中被验证的脏区域将在协调结果为脏区域。
- 在父版本中任何引入和验证的脏区域，不管是否在子版本中出现，将在协调结果中保持为验证的脏区域。

根据上面显示，父版本的原始状态在协调后仍被保存。但是，其他的脏区域可能作为子版本更新的结果被创建。

下面的例子 2 和例子 3 说明了其他的情况。

- 任何子版本中进行的拓扑要素的编辑将导致在协调中产生一个脏区域。即使脏区域来自在版本中被验证的编辑。当源编辑没有产生脏区域，也是这种情况，如一个属性的更新。这将在下面的例子中解释：

脏区域作为协调过程的结果创建

有许多情况，协调过程可以导致产生新脏区域，它们在父版本或者子版本中并不存在，这些脏区域是由于验证过程中的聚类处理导致的。在下面的例子中，两个版本都包含多边形，它们共享拓扑中的边界。一个多边形在子版本中被分割，脏区域被验证。

分割一个多边形，删除源要素，代之的是两个新要素。当脏区域被验证，聚类处理引入新的顶点到相邻多边形的共享边界。当两个版本被协调时，所有在子版本中已经被修改的要素——分割的多边形和由聚类处理添加的新顶点的多边形——都被脏区域覆盖。

以下的例子解释了为什么这是必要的。假定在两个版本中新要素被创建，结果中的脏区域被验证没有错误。在协调中，脏区域必须被创建，以至于潜在的错误通过合并两个版本中的变化而被引入。在这个例子中，Version1 和 Version2 的要素互相叠加，这违反了“Must not overlap”的规则。

错误要素和版本化要素

错误要素和标记为异常的错误要素在如何版本协调过程中在如何对待它们方面具有特殊的行为。错误要素仅能通过修改错误或者标记为一个异常来更新。

父版本中的错误和异常的协调结果可以总结如下：

- 任何在父版本中创建的错误，不管是否它被标记为一个异常，将作为协调结果被引入到子版本中。（在以下的例子 1 和 2 中显示）
- 在父版本中任何被标记为异常的预存在的错误，在协调后，将在子版本中被标记为一个异常。
- 任何从父版本中删除的错误或者异常，在协调后，将从子版本中被删除。

子版本中的错误和异常的协调结果可以总结如下：

- 任何在子版本中创建的错误，在协调后，将被删除，通过定义，将被脏区域覆盖。这个错误然后在验证脏区域时再次被发现。
- 任何在子版本中创建的错误和标记为异常的，在协调后，将作为一个异常。通过定义，它将被一个脏区域覆盖。
- 在父版本中的已存错误和在子版本中被标记为一个异常，在协调后，将保留为一个异常，并被一个脏区域覆盖。但是，如果这个错误在父版本被改正，然后它将在子版本中保留修改后的。（如例子 3 和例子 4 所示）
- 在父版本中存在的一个异常和在子版本中被标记为一个错误，协调后，将保留为一个错误，并被脏区域覆盖。但是，入骨这个异常在父版本中被修改，它将在子版本中保留修改后的。（如例子 5 和例子 6 所示）
- 在父版本中已存的一个错误或者异常和在子版本中被改正，将保留修改的作为协调结果。（如例子 7 和例子 8 所示）

在一些情况中，当子版本被建立时，通过验证父版本中已存的一个脏区域，在两个版本

中创建相同的错误。如果这个错误在父版本或者子版本中被标记为一个异常，协调将导致复制错误要素。在这些情况下，当脏区域被验证时，错误要素将被脏区域覆盖，并变为一个单独的错误或者异常。（如例子 9 和 10 所示）

异常和版本化要素类

当一个错误被标记为一个异常，仅错误要素被修改。如果一个错误在子版本中被标记为一个异常，协调后它仍保留为一个异常。以下是这种行为的含义：

如果一个错误是作为子版本中要素更新的结果被创建，然后在子版本中标记为一个异常，协调过程中创建的脏区域将跨被创建、修改、删除或者验证的要素建立错误。通过定义，这个脏区域也将包含这个异常。

如果这个错误是作为验证子版本建立时一个已存的脏区域的结果被创建，这个脏区域将是源脏区域，因此，将包含一个异常。

当子版本创建时，如果一个已存的错误被标记为一个异常，一个仅包含这个异常的脏区域被创建。

版本冲突和拓扑要素

参与拓扑的要素在版本协调中产生的冲突方面没有任何特殊行为。如果要相同的要素在两个不同的版本中被编辑，这些版本被协调，然后它们将处于冲突。

最常见的由拓扑验证过程产生的冲突源是被顶点引入的，由于验证中聚类处理阶段集成要素产生。

以下的例子说明了冲突是如何从验证过程中产生的。

例子 1－－在每个版本中分割两个相邻的多边形

一个父版本中共享边界的拓扑多边形被子版本继承。在这种情况下，父版本中的一个多边形被分割，在子版本中一个相邻的多边形被分割，脏区域被验证。分割的多边形删除源要素，用两个新的多边形代替它们。当版本被协调时，两个源多边形被告知更新－删除冲突。换句话说，当子版本被验证时，父版本中被删除的要素通过聚类处理被更新，子版本中被删除的要素通过父版本中的聚类处理被更新。

例子 2－－分割通过一个大右通行的多边形共享边界的多边形

在这个例子中，由验证过程产生的顶点给一个土地记录数据库带来冲突。在这种情况下，被分割的土地多边形通过一个大右通行的多边形共享边界。

如上面所描述的冲突，可以通过结构化工作流和使用 geodatabase 的复制来控制用户能否编辑的范围来避免。另外，数据模型设计可以帮助减少在第二个例子中介绍的冲突类型。

4、4、8、10 一般拓扑任务

在一个拓扑上可以进行大量修改。在一些情况下，如命名一个拓扑，修改对拓扑的状态没有任何影响。但是，在另外一些情况下，修改可能需要拓扑被验证。例如，添加新要素类或者规则的修改或者修改聚类误差，可能创建新的脏区域和错误的要素，拓扑重新被验证是必要的。

查询和操作图形

ArcGIS 开发工具 ArcObjects 提供了程序员使用拓扑图形的脚本。以下是一些实例脚本可以用于操作 geodatabase 中拓扑图形：

拓扑查询命令－－这个自定义命令允许基于要素和其他要素的拓扑关系进行选择查询。这个命令允许查询，如选择与被选择要素共享边界的要素，或者选择没有左多边形或者右多边形的边线。

确定相邻的命令－－这个自定义的命令基于左多边形或者右多边形的 Object IDs 来计算一个线图层的属性值。这个字段将计算 coverage 中的 LPOLY 和 RPOLY 字段中的相同信息。Geodatabase 拓扑的优势就是就是能够计算不同多边形集合中的这些信息。

创建最短路径要素的任务－－这个自定义脚本创建一个新的多义线要素，它是两个输入点的最段路径。含权重的 Dijkstra 算法符合图形中两个结点之间的欧氏距离。在拓扑图形中使用平面图形。

用 Kirk Kuykendall 绘制相邻图形－－这个实例脚本绘制线分开具有相同属性的多边形。

用线新建拓扑

有时候，需要从线要素数据创建多边形。例如，可能有一个数字化的边界要素集合，需要将它转换为多边形。或者，可能仅能从数据供应者那里获得线要素。

在 ArcCatalog

1、在 ArcCatalog 中，在数据集上，单击右键，指向 New，然后单击 Polygon Feature Class From Lines。

2、输入新多边形要素类的名字。

3、修改聚类误差（可选地）。

4、核查在查找封闭区域时需要考虑的要素类。

如果核查多于一个的要素类，那么被任何要素类的线围绕的区域将变为多边形要素。

如果核查一个多边形要素类，多边形边界将被认为与线相同。

5、 随意选择一个点要素类，它将被用于给新多边形赋属性。封入一个点的多边形将被给予点的属性。

6、单击 OK。新要素类在要素数据集中被创建。

使用 geoprocessing

Feature to Polygon 工具从输入的线和多边形要素生成新的多边形要素。可以指定一个点要素类，它将提供新多边形要素的属性。

Feature to Polygon 工具位于 Data Management Tools 工具箱中。

使用拓扑中的子类型

当设计一个 geodatabase 时，应该考虑在拓扑中使用要素类子类型的选项。

关于子类型

拓扑中的子类型可以有不同的规则应用于每个子类型，也可以有不同的坐标级别。这允许创建仅用于具体子类型的规则。也可以应用不同的坐标级别给每个子类型。

修改拓扑

在一个拓扑上可以进行大量修改。在一些情况下，如命名一个拓扑，修改对拓扑的状态没有任何影响。但是，在另外一些情况下，修改可能需要拓扑被验证。例如，添加新要素类或者规则的修改或者修改聚类误差，可能创建新的脏区域和错误的要素，拓扑重新被验证是必要的。

添加新要素类到拓扑

使用 ArcCatalog

以下是使用 ArcCatalog 添加一个新要素类到一个已存的拓扑中的步骤。

1、 在 ArcCatalog 中，在拓扑上，单击右键，单击 Properties。

2、 单击 the Feature Classes 标签。

3、 单击 Add Class。

4、 单击需要添加到拓扑中的要素类，单击 OK。只有在该要素数据集中，并且当前没有参与拓扑或者几何网络的要素类可以被添加。

5、 设置添加要素类的坐标级别。

6、 在 rules 标签上添加其他的规则。

7、 单击 OK，新要素类和规则将被添加到拓扑。修改拓扑的 schema 意味着整个拓扑将被标记为一个 dirty。

8、 最后一步就是验证修改的拓扑。

使用 geoprocessing 工具

也可以使用 geoprocessing 的拓扑工具来添加要素类和规则到一个已存的拓扑，在 Data Management 中的 Topology 工具箱中有两个可用的命令：

Add Feature Class To Topology

Add Rules To Topology

移除一个要素类

在 ArcCatalog 中

1、 单击 Topology Properties 对话框中的 Feature Classes 标签。

2、 单击需要移除的要素类。

3、 单击 Remove。

浏览拓扑属性

1、 在拓扑上，单击右键，单击 Properties。

重新命名一个拓扑

在 ArcCatalog 中

1、 单击 Topology Properties 对话框中的 General 标签。

2、 单击 Name 文本框，输入一个新的名字。

3、 单击 OK。

修改聚类误差

在 ArcCatalog 中

1、 单击 Topology Properties 对话框中的 General 标签。

2、 单击 Cluster Tolerance 文本框，输入一个新的聚类误差值。

3、 单击 OK。

修改坐标级别

可以有许多方法修改拓扑中设置的坐标级别。

修改级别值

- 单击 Topology Properties 对话框中的 Feature Classes 标签。
- 单击 Number of Ranks 文本框，输入一个级别值。

一个拓扑能够支持最大的级别为 50。

修改要素类的级别

- 单击 Topology Properties 对话框中的 Feature Classes 标签。
- 单击和修改要素的当前级别。

操作拓扑规则

在一个拓扑中有很多操作拓扑规则的任务。以下所示：

- 添加规则到一个已存拓扑或者移除规则。
- 保存规则为一个规则文件，它可以被共享和再利用。
- 浏览规则描述和图表。

添加规则到一个拓扑

1、单击 Topology Properties 对话框中的 Rules 标签。

2、单击 Add Rule，从下拉列表中选择添加一个新规则。

3、可以选择 Show Errors textbox 复选框，浏览每个规则的图形示例。

4、单击 OK。

从一个拓扑中移除一个规则

1、 单击 Topology Properties 对话框中的 Rules 标签。

2、 单击需要移除的规则。

3、 单击 Remove。

保存拓扑规则为一个 Rule Set File

1、 单击 Topology Properties 对话框中的 Rules 标签。

2、 单击 Save Rules。

3、 移动鼠标至需要保存已经定义在拓扑上的规则的位置。

4、 输入 Rule Set 的名字。

5、 单击 Save 保存所有的拓扑规则到这个文件。

从一个 Rule Set File 加载拓扑规则

1、 单击 Topology Properties 对话框中的 Rules 标签。

2、 单击 Load Rules。

3、 移动鼠标至需要加载的 Rule Set 被保存的位置。

4、 单击这个 rule set。

5、 单击 Open。Load Rules Feature Class Mapping 对话框将显示。如果 rule set 是来自与定义的新拓扑中的要素类具有相同要素类名字的一个拓扑创建的，这个要素类在 rule set 的命名应该匹配新拓扑中的要素类。

如果名字不同，将需要匹配 rule set 中的要素类与对应的新拓扑中的要素类。

6、 对于不能匹配的每个源要素类，单击 Target 列，单击符合新拓扑的要素类。

7、 单击 OK。

8、 单击 Topology Properties 对话框中的 OK 加载 rule set.。

获得一个拓扑规则的描述

1、 单击 Topology Properties 对话框中的 Rules 标签。

2、 单击需要查看描述的规则。

3、 单击 Description。

总结拓扑错误

可以在 Topology Properties 对话框中查看一个拓扑中的错误数量总结。这个总结告诉你有多少个错误和异常在每个拓扑规则中存在。

可以保存这个总结为一个文本文件来建立某一个特定时间拓扑状态的一个记录。这对于文档化和监视一个较大的拓扑编辑项目是很有用的。

在 ArcCatalog 中总结拓扑错误

1、 在 ArcCatalog 目录树中，在拓扑上，单击右键，单击 Properties，打开 Topology Properties 对话框。单击 Topology Properties 对话框中的 Errors 标签，单击 Generate Summary。

2、 如果需要保存错误总结报表为一个文本文件，单击 Export To File。

在 ArcMap 中总结拓扑错误

ArcMap 包括许多确定和改正独立的拓扑错误。

与 ArcCatalog 相似的一个任务可以执行。在 ArcMap 中，在拓扑数据层上，单击右键，单击 Properties 查看错误总结。

或者，在 ArcMap 中一旦启动编辑，可以使用 Topology Editing 工具条和它的 Error Inspector 工具生成一个错误报表，集中在每个独立的错误上进行修改。

4、4、8、11 使用拓扑的有用提示

测试预提议的拓扑设计

用一个 file 或者 personal geodatabase 中要素类数据的复本来测试拓扑设计。一个复本是很重要的，这是因为你将需要针对真正的数据验证拓扑设计是否工作良好。验证将聚类坐标，修改基本的几何特征。一些初始的设计可能不能如预期的进行工作。在测试前将需要保证源数据创建了复本。

如果一个 ArcSDE geodatabase 是被许多人使用，注意在你的系统处于联机状态时，具有较小的灵活性来修改变化的 schema。同时，如果有一个版本化的 geodatabase，将需要在应用 schema changes 前，压缩版本。通过对拓扑设计的一个好的初始测试，可以将问题减小到最少。

同时，也可以保存拓扑规则和定义在以后其他 geodatabase 的 schema 中再使用。

Schema locking

当建立一个拓扑时，所有输入的要素类上需要一个 exclusive lock。如果任何输入的要素类上有一个 shared lock，拓扑将不能被创建。

如果一个拓扑中的任何要素类具有一个 shared lock 或者 exclusive lock，这个 lock 被传递给拓扑中其他所有的要素类。

Topology error files

单验证拓扑时，发现和记录拓扑错误是很重要的。

多边形区域的多边形通常以多边形或者线记录错误。

线和点要素区域通常以线和点记录错误。

可以在 ArcMap 中以数据层浏览这些错误，当改正后，错误记录为 0。

4、4、9 地址定位器

一个 address locators 包含用于地理编码的参考数据，和使标准化地址的和搜索匹配的位置的参数和建立输出的一个快照。

当地理编码一个地址表时，地址定位器中包含的地址属性和索引被用于转换非空间的位置描述到空间数据，它可以以地图中的要素被显示，如街道地址。

4、4、10 维数

4、4、10、1 关于维数（仅对 ArcInfo 和 ArcEditor）

对于许多应用来说，显示地图中要素的形状和一些描述注记并不足够显示显示精确的测量值或者距离。维数是用于在地图上显示这种长度或者距离的一种特殊的地图注记。一个维数可能显示一个建筑物一边的长度或者地块一边的长度，它也可能显示两个要素之间的距离，如一个消防栓到一个建筑物角落的距离。

ArcGIS 提供的工具允许在 geodatabase 中存储维数。也可以建立维数式样适应维数要素，因此它们与应用标准一致。ArcGIS 支持大量的维数式样，以及建立维数的方法。维数可以从要素中自动被建立或者使用 ArcMap 中的丰富的创建工具集。

4、4、10、2 geodatabase 中的维数（仅对 ArcInfo 和 ArcEditor）

维数是用于在地图上显示这种长度或者距离的一种特殊的地图注记。一个维数可能显示一个建筑物一边的长度或者地块一边的长度，它也可能显示两个要素之间的距离，如一个消防栓到一个建筑物角落的距离。

一个维数要素由几个部分组成，它们可能显示，也可能不显示，依赖于应用程序。以下是对一个维数要素的解释：

Dimension types

ArcGIS 支持两种类型的维数：aligned dimensions 和 linear dimensions。Aligned dimensions 取基线的平行线，表达起始点和终点之间的真正距离。

不同于 Aligned dimensions，linear dimensions 并不表达起始点和终点的真正距离。linear dimensions 可以是垂直、水平或者旋转的。一个垂直的 linear dimensions 表达起始点和终点

的垂直距离。一个水平的 linear dimensions 表达起始点和终点的水平距离。一个旋转的 linear dimensions 是指维数线与基线成一定角度，它的长度表达了维数线的长度，而不是基线的长度。

所有的维数的方向可以是向内或者向外的。Outward dimensions 具有指向要素外部的 dimension lines，表达两个边界之间的距离。Inward dimensions 具有从要素外部指向的两个箭头，表达这两个箭头之间的距离。

Dimension 要素类

在 geodatabase 中，dimensions 存储在 dimensions 要素类中。和 geodatabase 中的其他要素类一样，所有 dimensions 要素类中的要素具有一个地理位置和属性，可以在一个数据集内，也可以不在一个数据集内。和注记要素相同，dimensions 要素是图形要素，它们的符号被存储在 geodatabase 中。

Dimension styles

dimension styles 的集合与一个 dimension 要素类相关联。dimension styles 描述 dimension 要素的符号，绘制的组成部分和如何被绘制。每次新建一个 dimension 要素，它被赋予一个特定的样式。具有一个特定样式的所有 dimension 要素共享确定的特征，一些可以基于 feature-by-feature 原则进行修改。可以使用 ArcCatalog 来创建、拷贝和管理一个 dimension 要素类式样。使用 ArcMap 中的编辑功能，式样被赋予单独的 dimension 要素。

以下是一个 dimension 要素的属性，它可以基于一个样式被赋予：

ArcGIS 提供工具允许在 geodatabase 中存储 dimension。也可以建立 dimension 样式，应用到 dimension 要素，因此它们与应用标准相一致。ArcGIS 支持大量的 dimension 类型和创建 dimension 的方法。dimension 可以从要素中自动被建立或者使用 ArcMap 中的丰富的创建工具集。

一旦在一个 dimension 类中创建了式样，它将不能被修改。如果需要修改一个已存 dimension style 的属性。必须用新的属性建立一个新的 dimension style。可以基于一个已存的 dimension style 属性建立一个新的式样，或者从其他 geodatabase 中的 dimension 要素类导入 dimension style。

性能考虑

Dimension 是 geodatabase 注记的一种类型。和常规注记一样，只有在放大到能清楚显示 dimension 要素的比例时，dimension 要素所传递的信息才是有用的。和注记要素一样，dimension 要素从数据库中检索，绘制在地图中显示是代价昂贵的。当在 ArcMap 中操作 dimension 要素类时，应该应用比例约束以使 dimension 要素仅在可视的范围被绘制。

4、4、10、3 创建 dimension 要素类 (仅对 ArcInfo 和 ArcEditor)

关于创建 dimension 要素类

Dimension 要素被存储在 dimension 要素类中。创建一个 dimension 要素类与创建存储自定义要素的要素类是相似的。

创建一个 dimension 要素类前，必须至少为将要创建的 dimension 要素建立一个式样。可以自己指定这个式样的属性，或者从其他 dimension 要素类导入式样，或者向导将建立一个默认的式样。

一旦创建了一个 dimension 要素类，可以使用 ArcCatalog 来建立和导入其他的式样。

如何建立 dimension 要素类(仅对 ArcInfo 和 ArcEditor)

用默认的样式建立一个 dimension 要素类

1、 在 ArcCatalog 中，在需要建立新 dimension 要素类的 geodatabase 或者要素数据集上，单击右键。

2、 指向 New。

3、 单击 Feature Class。

4、 输入新 dimension 要素类的名字。为了建立这个要素类的假名，输入假名。

5、 单击第二个选项，存储要素类中的 custom objects。

6、 单击下拉列表箭头，单击 ESRI Dimension Feature。

7、 单击 Next。

8、 输入一个参考比例尺。如果这个 dimension 要素类已经在要素数据集中创建，参考比例尺单位将自动匹配空间参考的单位。如果这是一个独立的 dimension 要素类，应该选择稍后在向导中空间参考将被指定的单位。

9、 单击 Next。

10、如果是在一个要素数据集中建立这个 dimension 要素，根据“建立要素类”章节中，按照建立一个要素数据集中的一个要素类的步骤 6 到 14 来完成。如果建立一个独立的 dimension 要素类，按照建立一个独立要素类的步骤 2 到 12 来完成。

用一个自定义式样创建一个 dimension 要素类

1、 在 ArcCatalog 中，在需要建立新 dimension 要素类的 geodatabase 或者要素数据集上，单击右键。

2、 指向 New。

3、 单击 Feature Class。

4、 输入新 dimension 要素类的名字。为了建立这个要素类的假名，输入假名。

5、 单击第二个选项，存储要素类中的 custom objects。

6、 单击下拉列表箭头，单击 ESRI Dimension Feature。

7、 单击 Next。

8、 输入一个参考比例尺。如果这个 dimension 要素类已经在要素数据集中创建，参考比例尺单位将自动匹配空间参考的单位。如果这是一个独立的 dimension 要素类，应该选择稍后在向导中空间参考将被指定的单位。

9、 单击该选项建立自己的式样。

10、单击 New Style 打开 Dimension Style Properties 对话框。

11、单击 Dimension Style Properties 对话框，设置 dimension style 的属性。

12、单击 OK。

13、单击 Next。

14、如果是在一个要素数据集中建立这个 dimension 要素，根据“建立要素类”章节中，按照建立一个要素数据集中的一个要素类的步骤 6 到 14 来完成。如果建立一个独立的 dimension 要素类，按照建立一个独立要素类的步骤 2 到 12 来完成。

通过导入一个式样建立一个 dimension 要素类

1、 在 ArcCatalog 中，在需要建立新 dimension 要素类的 geodatabase 或者要素数据集上，单击右键。

2、 指向 New。

3、 单击 Feature Class。

4、 输入新 dimension 要素类的名字。为了建立这个要素类的假名，输入假名。

5、 单击第二个选项，存储要素类中的 custom objects。

6、 单击下拉列表箭头，单击 ESRI Dimension Feature。

7、 单击 Next。

8、 输入一个参考比例尺。如果这个 dimension 要素类已经在要素数据集中创建，参考比例尺单位将自动匹配空间参考的单位。如果这是一个独立的 dimension 要素类，应该选择稍后在向导中空间参考将被指定的单位。

9、 单击第三个选项，从一个已存的 dimension 要素类中导入一个式样。

10、单击 Import Style，浏览一个需要从其导入式样的 dimension 要素类。

11、单击 Browse 按钮，浏览 geodatabase，一旦这个 geodatabase 被选择，dimension 要素类和包含 dimension 要素类的要素数据集将被列在树视图上。

12、单击包含需要导入式样的 dimension 要素类。

13、单击需要导入的 dimension style。

14、单击 OK。

15、单击 Next。

16、如果是在一个要素数据集中建立这个 dimension 要素，根据“建立要素类”章节中，按照建立一个要素数据集中的一个要素类的步骤 6 到 14 来完成。如果建立一个独立的 dimension 要素类，按照建立一个独立要素类的步骤 2 到 12 来完成。

建立和管理 dimension styles(仅对 ArcInfo 和 ArcEditor)

关于建立和管理 dimension styles

一个 dimension styles 描述了一个 dimension 要素如何被显示，包括其符号，标注字体和标注文本。每个 dimension 要素类至少有一个 dimension styles。在一个 dimension 类中的 dimension 要素是与一特定式样相关联的。某一特定式样的所有 dimension 要素具有确定的属性，它们是相同的，其他的属性可可以基于 feature-by-feature 原则被重载。

dimension styles 在 ArcCatalog 中的 Feature Class Properties 对话框被创建和管理。可以创建、删除、重命名和导入 dimension styles，可以为一个 dimension 要素类指定默认的式样。

dimension styles 的属性在 Dimension Style Property 对话框中被定义。这个对话框又三个标签：Lines and Arrows 标签、Text 标签和 Fit 标签。Lines and Arrows 标签允许你设置 dimension lines、line arrows,和 extension lines 的属性。Text 标签允许你控制 dimension 的文本和符号。当 dimension 的长度太短而不能显示 extension lines 之间的箭头和文本时，Fit 标签允许定义如何调整 dimension 和 dimension 文本。

当创建 dimension 要素和赋予它们一个式样时，一些属性可以基于 feature-by-feature 原则被重载。可以给每个要素重载的属性是：

Dimension line 显示

Dimension line arrow symbol 显示

Dimension 文本值

Extension line 显示

当建立、重命名或者删除一个 dimension 要素类中的式样时，需要一个 exclusive lock。

如何建立和管理 dimension styles

创建一个新的 dimension styles

1、在 ArcCatalog 中，在 dimension 要素类上，单击右键。

2、单击 Properties。

3、单击 Dimension 标签。

4、单击 New。

5、输入新 dimension style 的一个名字。

6、单击下拉列表箭头，单击 dimension 要素类中的式样，它包含需要拷贝到新式样中的属性。

7、在 Dimension Style Properties 对话框中，修改需要改变的 dimension style 元素。

8、单击 OK。

9、单击 Apply。

导入 dimension style

1、 在 ArcCatalog 中，在 dimension 要素类上，单击右键。

2、 单击 Properties。

3、 单击 Dimension 标签。

4、 单击 Import。

5、 单击 Browse 按钮浏览一个 geodatabase。一旦一个 geodatabase 被选择，dimension 要素类和包含 dimension 要素类的要素数据集将在树视图中被列出。

6、 单击包含需要导入的式样的 dimension 要素类。

7、 摁住 Ctrl 键，单击需要导入的 dimension styles。

8、 单击 Import。

9、 无论是否导入成功，将显示一个对话框。

10、单击 OK。

11、重复步骤 6 到 10，从同一个 geodatabase 中导入更多的 dimension styles 或者步骤 5 到 8 从一个不同的 geodatabase 导入 dimension styles。

12、单击 OK。

重命名一个 dimension styles

1、在 ArcCatalog 中，在 dimension 要素类上，单击右键。

2、单击 Properties。

3、单击 Dimension 标签。

4、在需要重命名的 dimension style 上，单击右键。

5、单击 Rename。

6、输入新 dimension style 的名字，并输入回车键。

7、单击 Apply。

设置默认的 dimension style

1、在 ArcCatalog 中，在 dimension 要素类上，单击右键。

2、单击 Properties。

3、单击 Dimension 标签。

4、在需要设置为默认的 dimension style 上，单击右键。

5、单击 Set as Default。

6、单击 Apply。

删除一个 dimension style

1、在 ArcCatalog 中，在 dimension 要素类上，单击右键。

2、单击 Properties。

3、单击 Dimension 标签。

4、在删除的 dimension style 上，单击右键。

5、单击 Delete。

6、单击 Apply。